MODEL-BASED SPEED AND PARAMETER TRACKING FOR INDUCTION MACHINES

by

Kazuaki Minami

Bachelor of Engineering (Kogakushi) in Applied Mathematics and Physics
Kyoto University, Japan
(March 1980)

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1989

Signature of Author _____
Department of Electrical Engineering and Computer Science
May 12, 1989

Certified by _____
George C. Verghese
Associate Professor, Department of Electrical Engineering
Thesis Supervisor

Accepted by _____
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

Model-Based Speed and Parameter Tracking for Induction Machines

by

Kazuaki Minami

Submitted to the Department of Electrical Engineering and Computer Science on May 12, 1989, in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering.

Abstract

This thesis investigates the estimation of machine model parameters and the rotor speed of an induction machine from the stator voltages and currents. A major difficulty in estimation is the nonlinearity of the induction machine model. Since the dynamic structure in a small induction machine can be separated into two time scales, the estimation algorithm can also be separated into two time scales. In addition, since stator resistance is difficult to estimate and errors in this may corrupt the whole estimation, it should be estimated separately. Based on observations, three *linear* but coupled regression models are derived.

Utilizing the two-time scale property, the estimation is separated into a slow time scale and a fast time scale. Moreover, the slow time scale estimation is divided into two stages. One is a stator-resistance-only estimator. The other is the estimator of other machine parameters. The fast time scale estimator estimates rotor speed only. Variable forgetting factors and periodic covariance resets are used in order to eliminate estimator windup and provide good tracking characteristics.

Numerical simulations and off-line analysis of actual experiments are done to demonstrate the performance of the algorithm. The results show good ability to track parameter and speed variations.

Thesis Supervisor : Dr. George C. Verghese

Title : Associate Professor of Electrical Engineering

ACKNOWLEDGEMENTS

My years at MIT have been a revealing voyage of self-discovery through hardships and triumphs of graduate school.

I would like to express my appreciation and gratitude to my thesis supervisor, Professor George C. Verghese, not only for his valuable advice and constructive suggestion provided through this study but also for his friendship and guidance these years.

I thank Dr. Hirofumi Akagi (Associate Professor of Technological University of Nagaoka), who stayed at MIT for his sabbatical leave, for his advice about motor control. I also thank Ms. Dagmar Elten and Professor Dieter Filbert at Technische Universität Berlin, especially for Ms. Elten's courtesy in allowing me to use her excellent experimental setup and for her advice on machine parameter estimation.

I would like to thank Miguel Vélez-Reyes for his initial work on speed and parameter estimation and for his kind advice on my research; Louis Roehrs (now with Apple) for his data acquisition program; Karen Walrath for her help in dealing with the lab software. I thank other people in the lab — Alex, Carlos, Clem, Kathy, Kris, Larry, Mak, Peyman, Rae, Ray, and Xiaojun — for their moral support and friendship.

I am grateful for the funding provided by the Nippondenso Company, Limited. The experience at MIT is invaluable for me. The drive system was donated by Industrial Drives, and additional support for parts was provided by the MIT/Industry Power Electronics Collegium and the MIT Soderberg Chair in Power Engineering.

Finally, my appreciation goes to my wife, Mariko, and to my son, Ryoji, for their limitless patience and understanding of my hardship.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement and Approach

The advances during the past 20 years in the area of "Field-Oriented Control" of AC drives [4,23,24] have made possible the use of induction machine drives in high performance industrial applications where DC drives had been exclusively used. Because of their simple and robust structure, induction machine drive systems have many advantages over DC machines, especially in their lower cost and maintenance-free characteristics.

Field-Oriented Control requires an accurate knowledge of the parameters and the electrical and mechanical variables of the induction machine. Therefore, two significant problems arise in the control of an induction machine: first, the estimation or measurement of the electrical and mechanical variables; second, the estimation or measurement of the machine parameters.

The common approach in high performance control of induction machine drive systems involves the use of a speed sensor connected to the shaft of the machine. The

use of this sensor has two major drawbacks. First, it obstructs the mechanical interface between the motor and the load, interfering with the requirements of tight coupling or close spacing in the mechanical layout or imposing the application of undesirable gear trains or belts. Second, it generally requires the use of brushes or light sources that require maintenance, which represents a weak link in the system in terms of reliability. Therefore, the use of the sensor in the induction machine drive system spoils the mechanical simplicity and robust structure of the machine, which are two of the most important reasons to use these drives instead of DC machines [33].

High performance estimation can be realized by knowing the characteristics of the system. The dynamic analysis of a small induction machine and an estimation scheme which exploits the system characteristics are introduced in [33].

The present research work extends the work in [33], addressing the problem of estimating the rotor speed and machine parameters based on measurements of the stator voltages and currents. The research is aimed at eventual development of a low cost and simple Field-Oriented Control algorithm for induction machines that avoids using an expensive speed sensor. In this thesis the design and testing of an estimation algorithm for these quantities is presented. The approach used in this research is the following. Based on two-time-scale properties of small induction machine systems [33], three linear regression models are derived, relating measured quantities to those that need to be estimated. Then estimation using recursive least squares (RLS) with several practical modifications is proposed. The algorithm is verified using both simulated data and actual experimental data.

The thesis is organized as follows. Chapter 1 introduces the induction machine model. In Chapter 2 the requisite theoretical background is discussed. Chapter 3 shows three linear regression models for estimation. Then, in Chapter 4, a two-time-scale three-stage estimator algorithm using recursive least squares with practical modifications is proposed. In Chapter 5, the proposed algorithm is validated using synthesi

data. Chapter 6 shows the off-line estimation results using actual experimental data. Chapter 7 summarizes the thesis and addresses future work.

1.2 Induction Machine Model

The modeling of an induction machine is a well known area and [5,23,24] provide a good description of these models. A quick review is included here in order to establish the notation.

Using the assumption of magnetic linearity and sinusoidally distributed windings in the stator, applying conservation of energy, and exploiting the Park transformation [23,25], the induction machine model referred to stator-fixed coordinates is given by

$$\frac{d\lambda}{dt} = \left(-\mathbf{R}\mathbf{L}^{-1} + \begin{bmatrix} 0 & 0 \\ 0 & \omega_r \mathbf{J} \end{bmatrix}\right)\lambda + \mathbf{v} \tag{1.1}$$

$$\frac{d\omega_r}{dt} = p\left[-\frac{B}{H}\omega_r + \frac{(T_{em} - T_L)}{H}\right] \tag{1.2}$$

$$T_{em} = \frac{p}{2}\frac{M}{L_s L_r - M^2}\lambda^T \begin{pmatrix} \mathbf{0} & \mathbf{J} \\ -\mathbf{J} & \mathbf{0} \end{pmatrix}\lambda \tag{1.3}$$

$$\lambda = \mathbf{L}\mathbf{i} \tag{1.4}$$

where p denotes the number of pole pairs and

$$\mathbf{i} = \left(\mathbf{i}_s^T, \mathbf{i}_r^T\right)^T \tag{1.5}$$

$$\mathbf{v} = \left(\mathbf{v}_s^T, \mathbf{v}_r^T\right)^T \tag{1.6}$$

$$\mathbf{L} = \begin{pmatrix} L_s\mathbf{I} & M\mathbf{I} \\ M\mathbf{I} & L_r\mathbf{I} \end{pmatrix} \tag{1.7}$$

$$\mathbf{R} = \begin{pmatrix} R_s\mathbf{I} & 0 \\ 0 & R_r\mathbf{I} \end{pmatrix} \tag{1.8}$$

$$\mathbf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{1.9}$$

$$\mathbf{J} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \tag{1.10}$$

In the above model the rotor speed (ω_r) and the stator and rotor fluxes (λ) are state variables. The subscripts s and r stand for stator and rotor quantities and $\mathbf{i}$ and $\mathbf{v}$ denote the terminal current and voltage vectors. R_s and R_r are the stator and rotor resistance, and L_s, L_r and M are the stator, rotor and mutual inductance, respectively. Finally, B is the friction coefficient, H is the combined inertia of rotor plus mechanical load, T_{em} is the electromagnetic torque, and T_L is the load torque.

For a squirrel cage induction machine, the voltage vector takes the form $\mathbf{v} = \left(\mathbf{v}_s^T, \mathbf{0}^T\right)^T$. Using a standard isomorphism to represent two-axis real variables as a complex variable[29], (1.1) - (1.4) are reduced to

$$\frac{d\tilde{\lambda}}{dt} = \left(-\tilde{\mathbf{R}}\tilde{\mathbf{L}}^{-1} + \begin{bmatrix} 0 & 0 \\ 0 & j\omega_r \end{bmatrix}\right)\tilde{\lambda} + \begin{pmatrix} \tilde{v}_s \\ 0 \end{pmatrix} \tag{1.11}$$

$$\frac{d\omega_r}{dt} = p\left[-\frac{B}{H}\omega_r + \frac{(T_{em} - T_L)}{H}\right] \tag{1.12}$$

$$T_{em} = \frac{pM}{L_s L_r - M^2} Im(\tilde{\lambda}_s \tilde{\lambda}_r^*) \tag{1.13}$$

$$\tilde{\lambda} = \tilde{\mathbf{L}}\tilde{\mathbf{i}} \tag{1.14}$$

where

$$\tilde{\mathbf{L}} = \begin{pmatrix} L_s & M \\ M & L_r \end{pmatrix} \tag{1.15}$$

$$\tilde{\mathbf{R}} = \begin{pmatrix} R_s & 0 \\ 0 & R_r \end{pmatrix} \tag{1.16}$$

Under constant speed operation, (1.11) is a linear time invariant (LTI) model. Therefore, the transfer function from (complex) stator voltage to (complex) stator current

is

$$H_{\omega_r}(s) = \frac{\frac{L_r}{\sigma}s + \frac{L_r}{\sigma}(\frac{1}{T_r} - j\omega_r)}{s^2 + (\frac{L_sR_r+L_rR_s}{\sigma} - j\omega_r)s + \frac{L_rR_s}{\sigma}(\frac{1}{T_r} - j\omega_r)} \tag{1.17}$$

where

$$T_r = L_r/R_r \tag{1.18}$$

$$\sigma = L_rL_s - M^2 \tag{1.19}$$

Note that the rotor speed ω_r enters as a parameter in this transfer function. The use of this model to estimate the rotor speed and machine parameters was introduced in [29] and is refined and extended in this thesis.

Chapter 2

Theoretical Background

2.1 Hierarchical Structure

Two-time-scale properties in small induction machines are presented in [33]. The eigenvalues of a small induction machine, derived using a small signal model and assuming constant speed operation, are classified in three groups by use of participation factors [27]: the stator eigenvalues, the rotor eigenvalues, and the speed eigenvalue. The fast ones (those with larger real part) are usually associated with the electrical subsystem (stator and rotor), and the slow eigenvalue is associated with the mechanical subsystem (speed). Considering the evolution of the electrical variables in an interval of time much smaller than the mechanical time constant, the machine fluxes and currents changes significantly, while the rotor speed remains almost constant. For constant rotor speed, the electrical equations (1.1), (1.4) of the machine model constitute a linear time invariant (LTI) system, with stator voltage and current appearing as input and output for the model, and with the rotor speed appearing as a parameter of this model. Hence a parameter estimation algorithm based on the linear electrical equations, with convergence faster than the mechanical time constant, can estimate rotor speed as an

additional machine parameter. The algorithm must be designed to adapt to changes in the parameters, so that variations in speed and other parameters can be tracked.

2.2 Linear Regression Model

There are several ways to describe the dynamic relationship between the input and output signals. The linear regression models in our work take the form

$$y_0(k) = \mathbf{C}(k)\theta \tag{2.1}$$

$$\theta^T = \begin{pmatrix} a_1 & \cdots & a_n & b_1 & \cdots & b_m \end{pmatrix} \tag{2.2}$$

$$\mathbf{C}(k) = \begin{pmatrix} -y_1(k) & \cdots & -y_n(k) & u_1(k) & \cdots & u_m(k) \end{pmatrix} \tag{2.3}$$

with $y_i(k)$, $u_j(k)$ being the observed output and input variables at time $k\triangle$, where $\triangle$ is the sampling interval and $i = 0, 1, \cdots, n$, $j = 1, 2, \cdots, m$. The vector θ contains the parameters to be estimated. The components of $\mathbf{C}(k)$ are called regression variables or regressors.

For notational simplicity, $y(k)$ is used instead of $y_0(k)$ in the rest of this thesis.

2.3 Model-Based Parameter Estimation

One way to estimate the parameter vector of (2.1) is by measuring a "prediction error" or "innovation error" defined by

$$\eta(k) = y(k) - \mathbf{C}(k)\hat{\theta}(k-1) \tag{2.4}$$

where $\hat{\theta}(k-1)$ is the parameter estimate at time $(k-1)\triangle$. The prediction error $\eta(k)$ gets its name from the fact that, according to the model, $y(k) = \mathbf{C}(k)\theta$ if θ is the

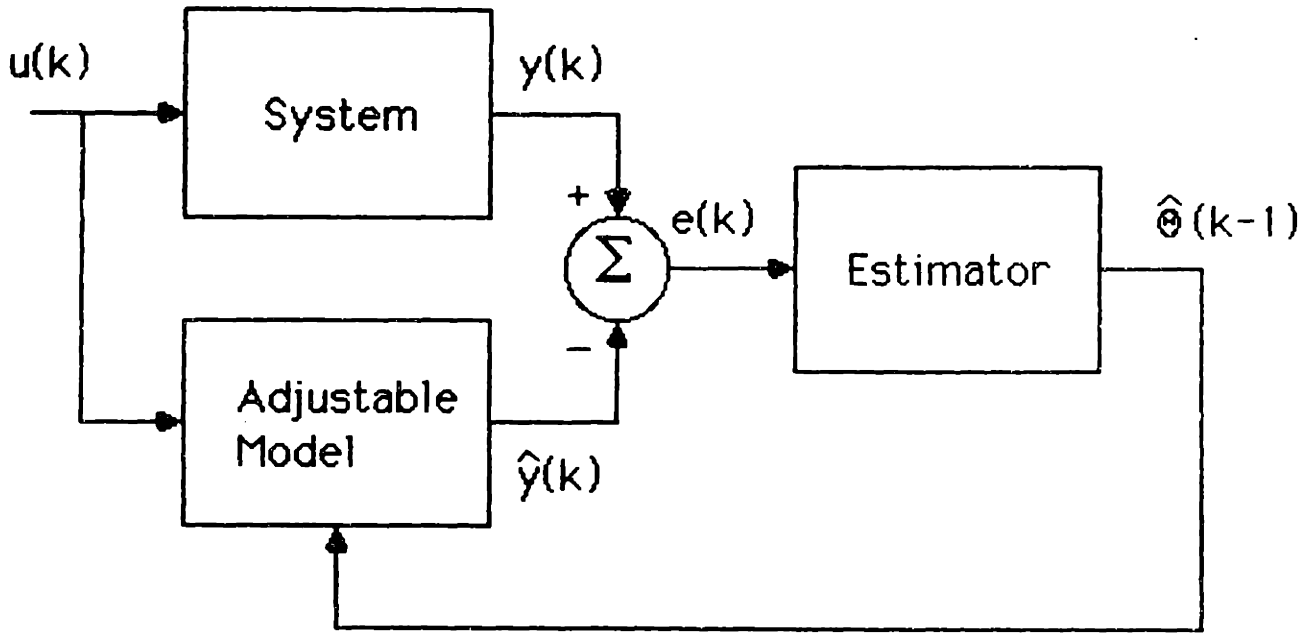


Figure 2.1: Structure of Model Based Parameter Estimation

"true" parameter. The estimation algorithm adjusts the model parameters to try and reduce the future error. This mechanism is illustrated in Figure 2.1.

The parameter estimate is updated via the equation:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \mathbf{K}(k)\left[y(k) - \mathbf{C}(k)\hat{\theta}(k-1)\right] \tag{2.5}$$

The challenge in such an estimation scheme is how to pick the estimator gain $\mathbf{K}(k)$ in (2.5) in order to obtain desirable behavior.

The notion of desirable behavior here is that the estimation algorithm should display good tracking of both the machine parameters and the rotor speed. As can be easily seen in (2.5), the gain $\mathbf{K}(k)$ should be kept big enough to track the variations of the parameters and small enough to get an averaging effect over time under noisy circumstances. There is a trade-off between these two performance requirements.

There are many ways to attempt to achieve these objectives. Some good references of are [26] and [34].

In order to get good tracking characteristics, it is necessary to discard old data. This

is done by use of a forgetting factor. However, unless care is taken, this introduces the possibility of estimator windup, where $\mathbf{K}(k)$ goes to 0. A variable forgetting factor [17] can be introduced to reduce the possibility of estimator windup. There are other methods to reduce or eliminate the possibility of estimator windup. Possible ways are by use of a so-called "periodic covariance reset" [19], through a "richness detector" [3,13] or an "error deadzone" [22]. Note that adjusting the forgetting factor and the richness detector do not guarantee the elimination of the estimator windup while the periodic covariance reset method does.

2.4 Recursive Least Squares Estimation

The basic idea of the Recursive Least Squares (RLS) estimation method is to compute the parameter vector θ in order to minimize the weighted quadratic cost function

$$J_N(\theta) = \sum_{i=1}^{N} \beta(N,i) \parallel e(i) \parallel^2 \tag{2.6}$$

where $\beta(N,i)$ is a weighting sequence such that

$$\beta(N,i) = \prod_{k=i+1}^{N} \alpha(k) \tag{2.7}$$

$$\beta(N,N) = 1 \tag{2.8}$$

and

$$e(i) = y(i) - c(i)\theta \tag{2.9}$$

Here $\alpha(k)$ is termed a forgetting factor. Picking $\alpha(k)$ to discard old data that might not contain sufficient information for estimating the present value of parameters will keep the gain $\mathbf{K}(k)$ from going to zero.

The recursive solution for this algorithm shown in [34] to be :

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \mathbf{K}(k)\left[y(k) - \mathbf{C}(k)\hat{\theta}(k-1)\right] \tag{2.10}$$

$$\mathbf{K}(k) = \frac{\mathbf{P}(k-1)\mathbf{C}^T(k)}{\alpha(k) + \mathbf{C}(k)\mathbf{P}(k-1)\mathbf{C}^T(k)} \tag{2.11}$$

$$\mathbf{P}(k) = \frac{1}{\alpha(k)}\left[\mathbf{P}(k-1) - \frac{\mathbf{P}(k-1)\mathbf{C}^T(k)\mathbf{C}(k)\mathbf{P}(k-1)}{\alpha(k) + \mathbf{C}(k)\mathbf{P}(k-1)\mathbf{C}^T(k)}\right] \tag{2.12}$$

$$\mathbf{P}(0) = \rho\mathbf{I} \tag{2.13}$$

where ρ is a positive number that reflects the uncertainty in our guess of the initial estimate $\hat{\theta}(0)$ of the parameter vector. This algorithm has the same structure as (2.5). The new parameter estimate is based only on the previous estimate $\hat{\theta}(k-1)$, the measurements at time $k\triangle$, $\alpha(k)$, and the previous covariance matrix[1] $\mathbf{P}(k-1)$.

[1]The term covariance is used in this context to point out the similarity between the structure of the

2.5 Variable Forgetting Factor

For nearly deterministic systems, a one-step forgetting factor chosen to maintain constant a scalar measure of the information content of the estimator enables the parameter estimates to follow both slow and sudden changes in the plant dynamics [17]. Furthermore, the use of a variable forgetting factor $\alpha(k)$ with correct choice of information bound can avoid one of the major difficulties associated with constant exponential weighting of past data - namely, "blowing-up" of the covariance matrix of the estimates.

The reasoning behind this scheme is as follows. For a nearly deterministic system the *a posteriori* error will at each step tell something about the state of the estimator. If the error is small, a reasonable strategy will be to retain as much information as possible by choosing a forgetting factor close to unity. If, on the other hand, the error is larger, the sensitivity to recent data should be increased by choosing a smaller forgetting factor. This will shorten the effective memory length of the estimator until the parameters are readjusted and the errors become smaller.

The proposed algorithm [17] related to estimation is :

$$\hat{y}(k) = \mathbf{C}^T(k)\hat{\theta}(k-1) \tag{2.14}$$

$$\varepsilon(k) = y(k) - \hat{y}(k) \tag{2.15}$$

$$\mathbf{K}(k) = \frac{\mathbf{P}(k-1)\mathbf{C}(k)}{1 + \mathbf{C}^T(k)\mathbf{P}(k-1)\mathbf{C}(k)} \tag{2.16}$$

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \mathbf{K}(k)\varepsilon(k) \tag{2.17}$$

$$\alpha(k) = 1 - \frac{[1 - \mathbf{C}^T(k)\mathbf{K}(k)]\varepsilon^2(k)}{\Sigma_0} \tag{2.18}$$

$$\mathbf{P}(k) = \frac{[\mathbf{I} - \mathbf{K}(k)\mathbf{C}^T(k)]\mathbf{P}(k-1)}{\alpha(k)} \tag{2.19}$$

If the right-hand side of (2.18) is less than some minimum value α_{min}, set $\alpha(k) =$

estimator given by (2.10) - (2.12) and the Kalman filter. This matrix does not represent the actual covariance of the estimator error unless the forgetting factor is always equal to unity.

α_{min}. The matrix Σ_0 is the variance of the measurement noise in the observation equation (2.1).

2.6 Periodic Covariance Reset

Goodwin [19] proved the convergence of a self-tuning regulator using ordinary recursive least squares with covariance resetting and proposed the following two algorithms.

The first one is :

$$\mathbf{P}(k) = \mu\mathbf{I} \tag{2.20}$$

$$\mathbf{P}(k) = \mathbf{P}(k-1) - \frac{\mathbf{P}(k-1)\mathbf{C}(k)\mathbf{C}^T(k)\mathbf{P}(k-1)}{1+\mathbf{C}^T(k)\mathbf{P}(k-1)\mathbf{C}(k)} \tag{2.21}$$

where (2.20) is for specific k's (say for $k = 10 \times i$ where i is integer) and (2.21) is for other k's.

The second one is :

$$\bar{\mathbf{P}}(k) = \mathbf{P}(k-1) - \frac{\mathbf{P}(k-1)\mathbf{C}(k)\mathbf{C}^T(k)\mathbf{P}(k-1)}{1+\mathbf{C}^T(k)\mathbf{P}(k-1)\mathbf{C}(k)} \tag{2.22}$$

$$\mathbf{P}(k) = \bar{\mathbf{P}}(k) \tag{2.23}$$

$$\mathbf{P}(k) = \bar{\mathbf{P}}(k) + \mathbf{Q}(k) \tag{2.24}$$

where Equation (2.23) is for the case the trace of $\mathbf{P}(k)$ exceeds a preset bound and Equation (2.24) is for other case, that is, the case that trace of $\mathbf{P}(k)$ is less than a preset bound or $0 \leq \mathbf{Q}(k) < \infty$. Since this second method checks trace of $\mathbf{P}(k)$ before updating $\mathbf{P}(k)$, it is similar to the method which is described in Section 2.8.

2.7 Richness Detector

Estimator windup in the absence of persistent excitation is characterized by increase of either norm of **PC** or that of ε. Therefore, proposed is the algorithm to stop the updating of the parameters and the covariance matrix when **PC** or ε is sufficiently small or in a preset bound [3,13,22]. The proposed algorithm using the norm of ε is

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \mathbf{K}(k)D[\varepsilon(k)] \tag{2.25}$$

$$\varepsilon(k) = y(k) - \mathbf{C}(k)\hat{\theta}(k-1) \tag{2.26}$$

where

$$D[\varepsilon(k)] = \begin{cases} 0 & \text{if } |\varepsilon(k)| \leq \delta \\ \varepsilon(k) & \text{if } |\varepsilon(k)| > \delta \end{cases} \tag{2.27}$$

2.8 Other Methods

There are other several ways to prevent or reduce the estimator windup. One is to keep the trace of **P** constant [3]. Another is to forget information only in the direction in which new information is gathered [3].

Chapter 3

Linear Regression Models for Estimation

3.1 Introduction

As is shown previously, the evolution of the electrical variables of the machine over an interval of time that is short relative to speed variations can be approximated by an LTI model. The transfer function from stator voltages to stator currents for this LTI model using complex variables was computed and given in (1.17).

In this chapter three linear regression models based on the transfer function will be derived. The estimation algorithm will estimate the parameters of these linear regression models. From these estimates, some of the machine parameters as well as the rotor speed can be estimated. Since there are five physical parameters (R_s, R_r, L_s, L_r, M) and rotor speed (ω_r), but only five independent equations, not all parameters can be estimated.

The possibility of different arrangements of the parameter estimation model is dis-

cussed in this chapter, and the use of state variable filters to avoid differentiation of measured signals is also described.

3.2 Linear Regression Model for Parameter Estimation

Let us for notational simplicity, rename the coefficients of the transfer function (1.17) in the following way :

$$a_1 = \frac{R_s L_r + R_r L_s}{\sigma} - j\omega_r \tag{3.1}$$

$$a_0 = \frac{L_r R_s}{\sigma}\left(\frac{1}{T_r} - j\omega_r\right) \tag{3.2}$$

$$b_1 = \frac{L_r}{\sigma} \tag{3.3}$$

$$b_0 = \frac{L_r}{\sigma}\left(\frac{1}{T_r} - j\omega_r\right) \tag{3.4}$$

Using this notation the transfer function from stator voltages to stator currents is given by

$$H(s) = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0} \tag{3.5}$$

Taking the inverse Laplace transformation, a second order differential equation relating stator voltages and currents can be obtained from (3.5). This equation is

$$\frac{d^2 i_s}{dt} + a_1 \frac{di_s}{dt} + a_0 i_s = b_1 \frac{di_s}{dt} + b_0 v_s \tag{3.6}$$

There are many ways to express this second order differential equation in a form that yields a regression model like (2.1). One way, assuming perfect knowledge of rotor speed (ω_r) and stator resistor (R_s), and writing two real equations instead of one

complex one, is

$$\begin{bmatrix} \frac{dv_\alpha}{dt} - R_s\frac{di_\alpha}{dt} + \omega_r(v_\beta - R_s i_\beta) \\ \frac{dv_\beta}{dt} - R_s\frac{di_\beta}{dt} - \omega_r(v_\alpha - R_s i_\alpha) \end{bmatrix} = \begin{bmatrix} \frac{d^2 i_\alpha}{dt^2} + \omega_r\frac{di_\beta}{dt} & \frac{di_\alpha}{dt} & -v_\alpha + R_s i_\alpha \\ \frac{d^2 i_\beta}{dt^2} - \omega_r\frac{di_\alpha}{dt} & \frac{di_\beta}{dt} & -v_\beta + R_s i_\beta \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \tag{3.7}$$

where p_1, p_2, and p_3 are defined by

$$p_1 = L_s\delta \tag{3.8}$$

$$p_2 = \frac{L_s}{T_r} \tag{3.9}$$

$$p_3 = \frac{1}{T_r} \tag{3.10}$$

and where $\delta = 1 - \frac{M^2}{L_r L_s}$.

Another form of solution is

$$\begin{bmatrix} \frac{d^2 i_\alpha}{dt^2} + \omega_r\frac{di_\beta}{dt} \\ \frac{d^2 i_\beta}{dt^2} - \omega_r\frac{di_\alpha}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{di_\alpha}{dt} & \frac{dv_\alpha}{dt} - R_s\frac{di_\alpha}{dt} + \omega_r(v_\beta - R_s i_\beta) & v_\alpha - R_s i_\alpha \\ -\frac{di_\beta}{dt} & \frac{dv_\beta}{dt} - R_s\frac{di_\beta}{dt} - \omega_r(v_\alpha - R_s i_\alpha) & v_\beta - R_s i_\beta \end{bmatrix} \begin{bmatrix} \bar{p}_1 \\ \bar{p}_2 \\ \bar{p}_3 \end{bmatrix} \tag{3.11}$$

where $\bar{p}_1$, $\bar{p}_2$, and $\bar{p}_3$ are defined by

$$\bar{p}_1 = \frac{1}{T_r\delta} \tag{3.12}$$

$$\bar{p}_2 = \frac{1}{L_s\delta} \tag{3.13}$$

$$\bar{p}_3 = \frac{1}{T_r} \tag{3.14}$$

Since the estimation algorithm is implemented in a microprocessor, the discrete time expression of the measurement equation is necessary. This equation at time $k\triangle$ where $\triangle$ is the sampling period is given by

$$\mathbf{y}_p(k) = \mathbf{C}_p(k)\theta_p \tag{3.15}$$

where $y_p(k)$ denotes the left-hand terms of (3.7) or (3.11) and $\mathbf{C}_p(k)$ is the first matrix and θ_p the second matrix of the right-hand terms of (3.7) or (3.11) respectively. Note that θ_p is the unknown parameter vector, which can be estimated using this linear regression model. By estimating the parameters of this model, some of the physical parameters can be obtained [10]. The subscript p is used here to show this model as parameter estimator.

3.3 Linear Regression Model for Stator Resistor Estimation

A different arrangement of (3.5) can be obtained when rotor speed ω_r and all machine parameters except R_s are known. In this case the following linear regression model for stator resistance estimation can be obtained :

$$\mathbf{y}_R(k) = \mathbf{C}_R(k) R_s \tag{3.16}$$

where

$$\mathbf{y}_R(k) = \begin{bmatrix} p_1 \frac{d^2 i_\alpha}{dt^2} + p_2 \frac{di_\alpha}{dt} + p_1 \omega_r \frac{di_\beta}{dt} - \frac{dv_\alpha}{dt} - p_2 v_\alpha - \omega_r v_\beta \\ p_1 \frac{d^2 i_\beta}{dt^2} + p_2 \frac{di_\beta}{dt} - p_1 \omega_r \frac{di_\alpha}{dt} - \frac{dv_\beta}{dt} - p_3 v_\beta + \omega_r v_\alpha \end{bmatrix} \tag{3.17}$$

$$\mathbf{C}_R(k) = \begin{bmatrix} -\frac{di_\alpha}{dt} - p_3 i_\alpha - \omega_r v_\beta \\ -\frac{di_\beta}{dt} - p_3 i_\beta + \omega_r v_\alpha \end{bmatrix} \tag{3.18}$$

and where all the quantities are measured at time $k\triangle$. The subscript R is used here to point out the difference between this model and the previous model. The quantities $\mathbf{y}_R$ and $\mathbf{C}_R$ can be written in terms of $\bar{p}_1$, $\bar{p}_2$, $\bar{p}_3$.

3.4 Linear Regression Model for Rotor Speed Estimation

Using a similar argument, a linear regression model where the speed is the only unknown parameter is

$$\mathbf{y}_\omega(k) = \mathbf{C}_\omega(k)\omega_r \tag{3.19}$$

and where

$$\mathbf{y}_\omega(k) = \begin{bmatrix} \frac{d^2 i_\alpha}{dt^2} p_1 + \frac{di_\alpha}{dt}(p_2 + R_s) + i_\alpha p_3 R_s - \frac{dv_\alpha}{dt} - v_\alpha p_3 \\ \frac{d^2 i_\beta}{dt^2} p_1 + \frac{di_\beta}{dt}(p_2 + R_s) + i_\beta p_3 R_s - \frac{dv_\beta}{dt} - v_\beta p_3 \end{bmatrix} \tag{3.20}$$

$$\mathbf{C}_\omega(k) = \begin{bmatrix} -p_1 \frac{di_\beta}{dt} - R_s i_\beta + v_\beta \\ +p_1 \frac{di_\alpha}{dt} + R_s i_\alpha - v_\alpha \end{bmatrix} \tag{3.21}$$

and where all the quantities are measured at time $k\triangle$. The subscript ω is used here to point out the difference between this model and the previous model.

3.5 State Variable Filters to Avoid Differentiation

The derivatives of stator voltages and currents that appear in the preceding regression equations are not available as measurable quantities. In addition, a direct analog or digital differentiation of these quantities causes problems because of the amplication of disturbance noise and quantization effects of the A/D-converters.

Though it is not possible to get derivative terms directly from measurements, some kind of signal processing can make it possible. A general treatment of this problem is presented in [15]. One possible method is using state variable filters presented in [33]. Note that the state variable filter should be in controllable canonical form to get the necessary derivative terms. Another possible method involves "modification functions"

[14]. State variable filters are used in this research because the main interest is eventual real-time application.

The Bode plots of the second order state variable filter used in both the simulations and the experimental setup are shown in Figures 3.1 - 3.3. Those plots show that at frequencies below cutoff the filter makes available the filtered output and its derivatives. Frequencies higher than the cutoff frequency should be eliminated by another filter in order for noise to be suppressed above cutoff.

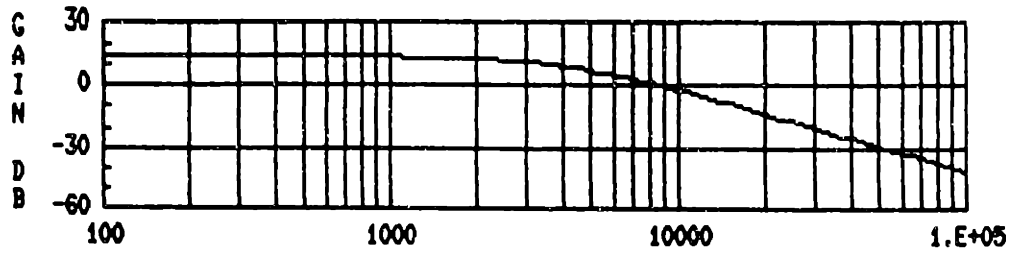


Figure 3.1: Bode Plots of State Variable Filter (X-axis is radian/sec) : Filter Output

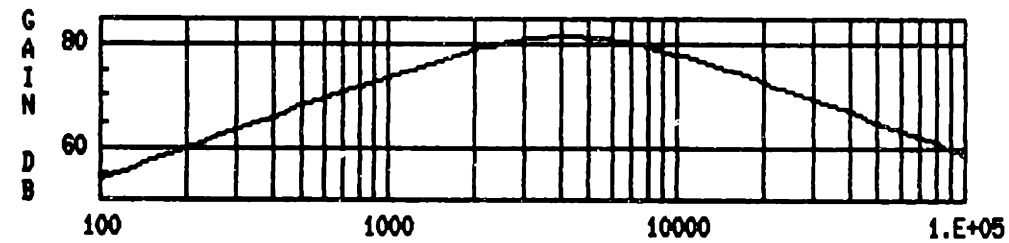


Figure 3.2: Bode Plots of State Variable Filter (X-axis is radian/sec) : Derivative of Filter Output

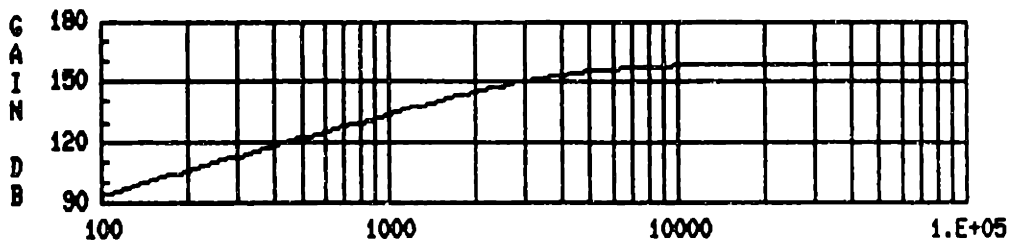


Figure 3.3: Bode Plots of State Variable Filter (X-axis is radian/sec) : Second Derivative of Filter Output

Chapter 4

Estimator Design

4.1 Introduction

In Chapter 3 three linear regression models that relate rotor speed, stator resistance and other physical parameters to stator voltages and stator currents were presented. In this chapter an algorithm to estimate all of the three groups of quantities using those models is described. The algorithm involves the use of Recursive Least Squares (RLS) estimation. Various ad hoc practical modifications introduced in Chapter 2 to overcome implementation problems are also incorporated.

4.2 Two-Time-Scale Structure

According to the discussion about hierachical structure in Chapter 2, it is natural to separate the estimator algorithm into two portions. One is a fast estimator portion which estimates the rotor speed, and the other is a slow estimator which estimates machine parameters. By using this separation, the whole algorithm can track parameter

and speed variations, which occur at different time scale.

There is another advantage to using this separation scheme. If both the parameters and the rotor speed are estimated at the same time, an expensive hardware with fast computation power is required in order to get good tracking, especially for rotor speed changes. Since the expensive part of the estimation (i.e. the slow estimator which requires more computation time) can be executed less frequently, however, one can assign the slow estimator to background mode, and allow the rotor speed (or speed-only) estimator to use hardware resources. This idea is shown in Figure 4.1 for the case where the speed estimate is updated twice as often as the parameter estimate. How to synchronize the two-time-scale estimator is shown in Figure 4.2.

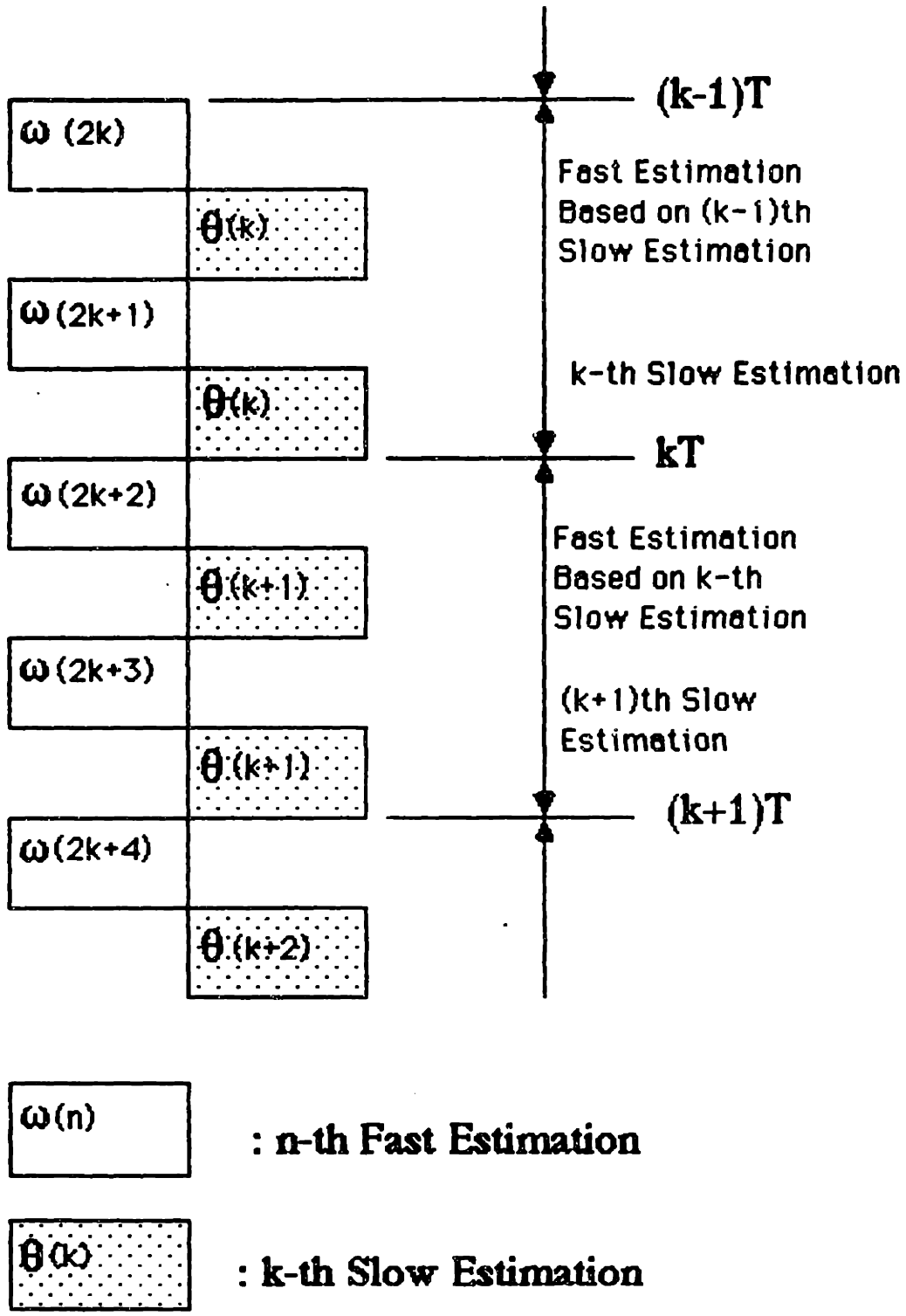


Figure 4.1: Structure of Two-Time-Scale Estimator

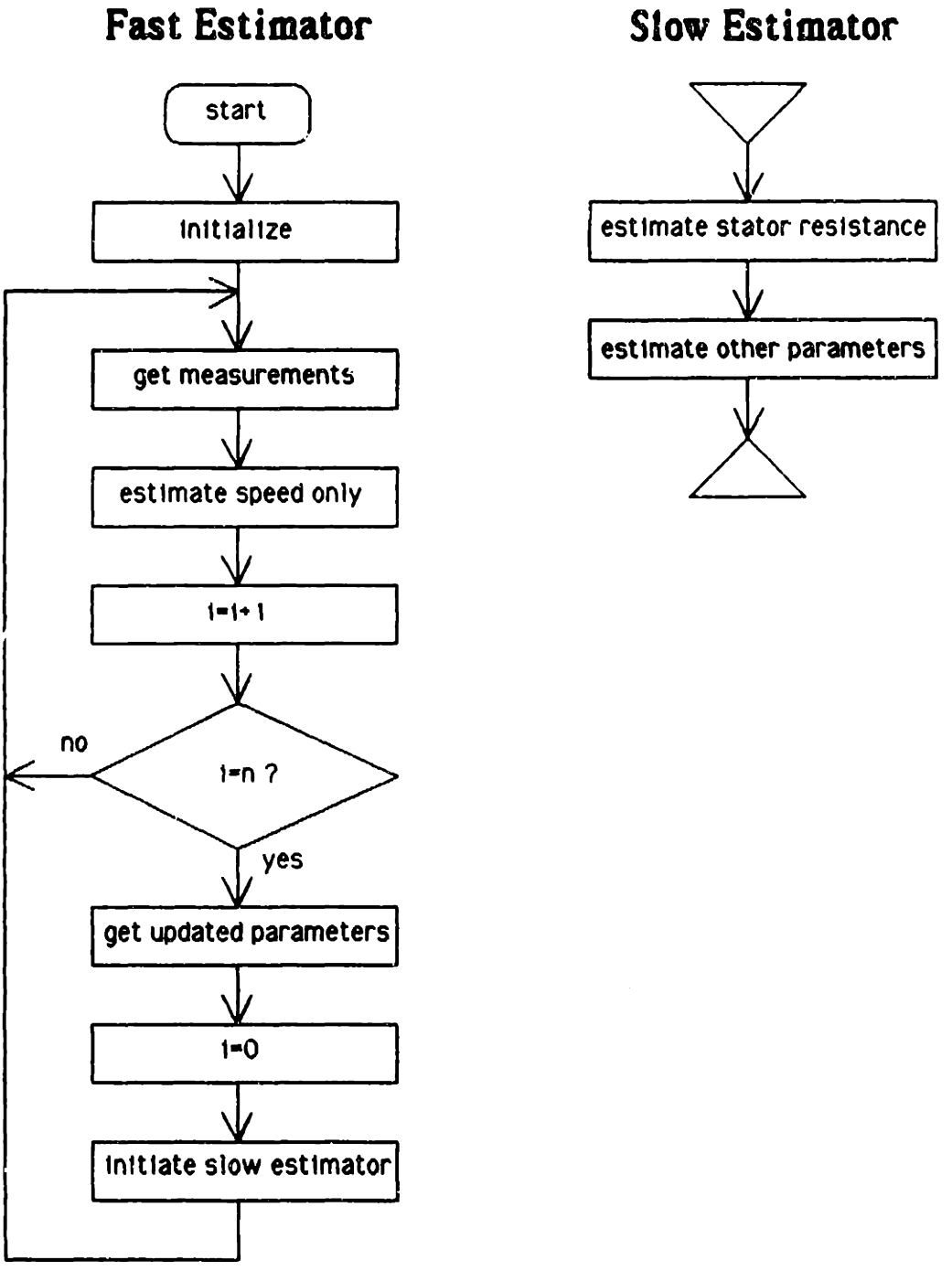


Figure 4.2: Flowchart of Synchronization

4.3 Two-Stage Slow Estimator

As mentioned in Section 3.1, the stator resistance is very difficult to estimate if left in with the other machine parameters, and its inclusion can jeopardize the whole estimation. Therefore the stator resistance estimator is implemented as an additional stage of the slow estimator.

The structure in the slow estimator is as follows. Treating the rotor speed estimate from the speed-only estimator and the most recent parameter estimates as "true" values, the estimator estimates stator resistance, assuming that this is the only unknown quantity. After updating the stator resistance estimate, the estimator will estimate other machine parameters, but now assuming the rotor speed and the stator resistance are exactly known. Then the slow estimator transfers the machine parameter values to the speed-only estimator to renew those values for its speed estimation task.

The structure of whole estimation algorithm is shown in Figure 4.3.

4.4 Practical Considerations

In order to get good tracking and eliminate or reduce the possibility of estimator windup, the methods introduced in Chapter 2 should be used in the estimator algorithm. Since the goal of this research is to develop a real time implementable algorithm for parameter and speed estimation of induction machines, the whole algorithm should be as simple as possible, taking account of the desire for microprocessor implementation. The rotor speed estimator should be especially simple because it must be updated very fast to track relatively fast variations in speed.

Fortunately the linear regression model for rotor speed estimation is simple (see Section 3.4). Instead of the usual matrix inversion, it requires only scalar division,

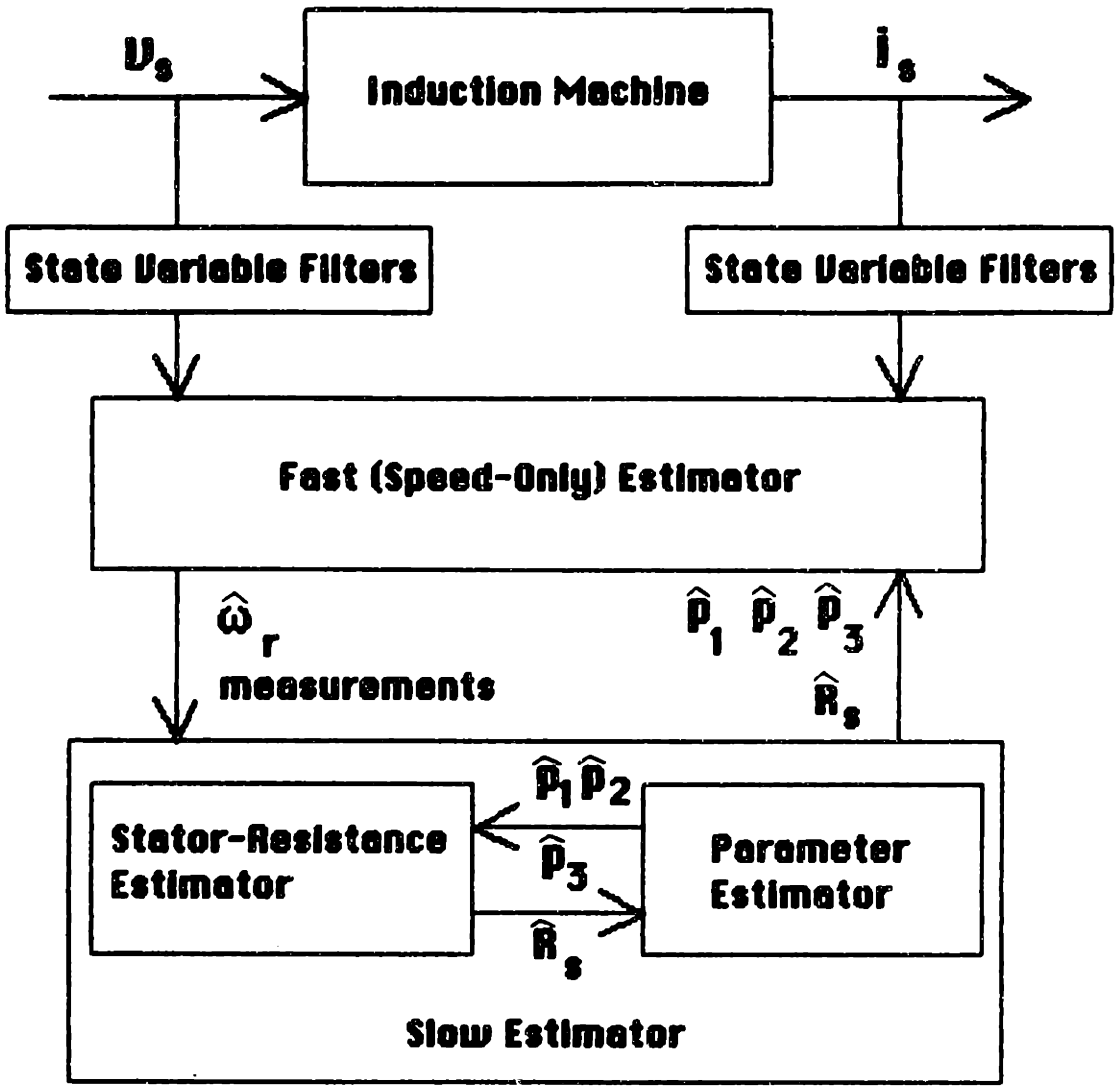


Figure 4.3: Block Diagram of the Two-Time-Scale Three-stage Estimator

since the number of estimated quantities is one. Similarly, the linear regression model for stator resistance estimation is simple too. Therefore the only complicated model is for estimating the other parameters.

In this research both variable forgetting factor and periodic covariance reset methods are applied. The periodic covariance reset scheme used here resets the covariance matrix $\mathbf{P}(k)$ to a preset value $\mu\mathbf{I}$ if the trace of $\mathbf{P}(k)$ exceeds a preset value. The reason that both are used is as follows. A variable forgetting factor may cause estimator windup, even though it gives a good tracking possibility. Periodic covariance reset may not give good tracking characteristics, but it eliminates the possibility of estimator windup. By combining both methods, good parameter tracking without estimator windup can be realized.

Another practical feature used is to hold the previous covariance matrix if at least one of the diagonal elements of the covariance matrix becomes negative. One more practical method used is to set the limit of one-step parameter change.

Note that these practical methods are applied to the most complicated linear regression model only in order to make the algorithm as simple as possible. For the other models, constant forgetting factors are used.

Chapter 5

Simulation Studies for the Estimator

5.1 Introduction

The analysis of the proposed estimator is very difficult because of the nonlinearity, time-varying characteristics, and the combination of three mutually dependent regression equations in the algorithm. In this chapter the performance of the estimator is studied through simulations.

This chapter is structured as follows. First, the simulation model used for data generation is given. Second, the estimation results under noise-free situations are presented. Third, the estimation results for a noisy enviroment are presented.

Parameter	Value
L_s	0.0293 H
L_r	0.0293 H
M	0.0277 H
R_s	0.3700 Ω
R_r	0.1260 Ω
Inertia	0.0065 Kgm^2
Rated Speed	2,500 rpm
number of pole pairs	2

Table 5.1: Machine Parameters of a 3.1 Horsepower Induction Machine from Blocked Rotor Test and No Load Test

5.2 Simulation Model for Data Generation

In this section the closed-loop scheme of induction machine drive scheme which is used for simulation is presented. The data was generated using the $SYSTEM - BUILD$ part of the $MATRIX_X$ software package.

The actual machine used as a basis for simulation is a 3.1 horsepower induction machine. The machine parameters of the induction machine using the traditional "Blocked Rotor Test" and "No Load Test" [16] are shown in Table 5.1, though actual parameters in operation will be expected to take different values.

Using these machine parameters the machine model is simulated using the closed-loop $V/f = constant$ and slip frequency control scheme with PI controller shown in Figure 5.1, [5]. A 3-pulse Pulse Width Modulation (3-pulse-PWM) scheme is used for the driving wave form in order to generate a sufficiently rich signal in current

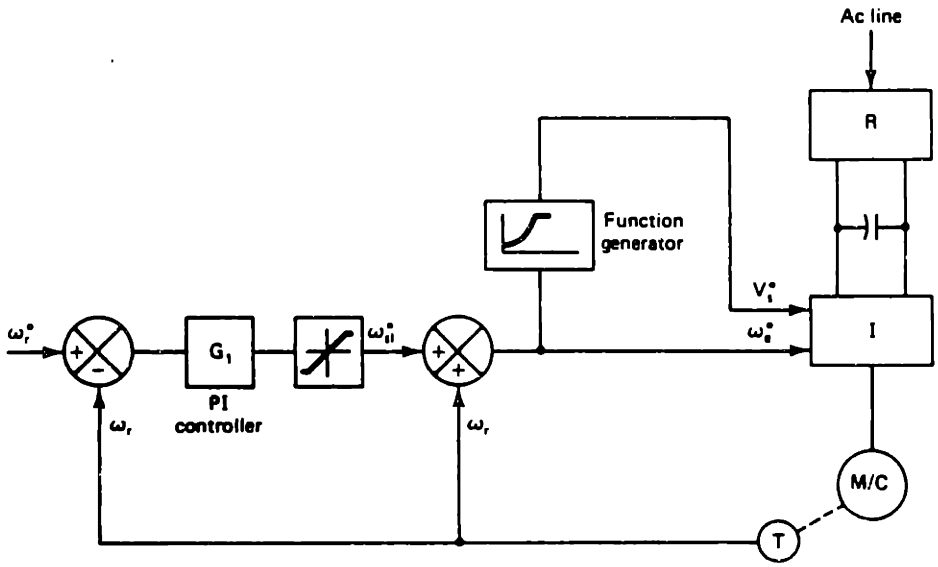


Figure 5.1: Block Diagram of Control Scheme

measurements. In this simulation the motor driving circuit was simulated in an abc frame while the motor was simulated in the $\alpha\beta$ (two-axis)-frame. See Appendix B for the details of the control scheme and the data generation model.

All stator currents and voltages are fed to second order Bessel-type anti-aliasing filters and state variable filters. The outputs of the state variable filters are sampled at 1 msec interval. The magnitudes of the measurements in peak-to-peak value are :

Non-derivative term of currents : 10^2

1st-derivative term of currents : 10^5

2nd-derivative term of currents : 10^8

Non-derivative term of voltages : 10^3

1st-derivative term of voltages : 10^6

Considering the through rate and the input and output voltage range of actual operational amplifiers, it is very difficult to realize similar characteristics to this digital

realization built in $MATRIX_X$, even if scaling is applied. Therefore the actual implemetation of state variable filter should be deliberately reviewed. One way to avoid this problem in real time implementation is to install state variable filters digitally same as the simulations. But in this case another microprocessor may be needed for this job and produce more time delay. Also note that in this case measurements should be sampled much faster to get good differential characteristics [14].

5.3 Estimation Results without Noise

Using the data generated by the drive scheme simulated in the previous section, the estimator tracking capabilities under noise-free circumstances have been studied. Representative results are presented here. The initial values of parameters for this estimation run are as follows :

$$\hat{\omega}_r = 0 \tag{5.1}$$

$$\hat{R}_s = 0.7 \times (\textit{true value}) \tag{5.2}$$

$$\hat{p}_1 = 0.9 \times (\textit{true value}) \tag{5.3}$$

$$\hat{p}_2 = 0.9 \times (\textit{true value}) \tag{5.4}$$

$$\hat{p}_3 = 0.9 \times (\textit{true value}) \tag{5.5}$$

The forgetting factors for the speed-only estimator and the stator-resistance estimator are constant (see α_ω and α_R respectively) while the forgetting factor for parameter estimation is a variable forgetting factor corresponding to noise variance Σ_0 :

$$\alpha_\omega = 0.8 \tag{5.6}$$

$$\alpha_R = 0.98 \tag{5.7}$$

$$\Sigma_0 = 1.6 \times 10^9 \tag{5.8}$$

In the process of choosing an appropriate noise variance, the error caused by incorrect parameter estimates was assumed to be nearly equal to the error caused by noise. Then Σ_0 was chosen for the forgetting factor to be around 0.99 when the parameter estimates converge to certain values and to be around 0.95 when the parameter estimates change their values.

The fast estimator (the speed-only estimator) operates every 1 msec while the slow estimator (the stator-resistance estimator and the parameter estimator) operates every

10 msec. The periodic covariance reset was not implemeted because there was no sign of estimator windup. The richness detector was implemented just for monitoring. Since there was neither sign of estimator windup nor the estimated parameter deviation from true value when the norms of ε and $\mathbf{PC}$ were small, the stopping of updating the parameters and the covariance matrix was not implemented.

The estimation results are shown in Figure 5.2 - 5.4. The results show that the algorithm exhibits a good tracking performance for speed variation, except for low speed transients.

One possible reason of poor speed estimation in low speed transients is following. The assumption that speed is almost constant in the evolution of electrical variables may not hold. Moreover the signal at low speed transient may not be so rich as that at high speed transients. The combination of these two might cause such poor estimation result. In order to get good tracking at low speed transient, nolinear estimator might be required to omit the assumption of *constant speed operation*.

Note that the parameter associated with higher order derivative terms (say $p1$) is tracked better than that associated with lower order ones (say $p3$). This behavior was also seen in [33] and is attributed to the richness of the corresponding regressor variables. This can be verified easily by checking the sensitivity of the model. Also note, however, that sensitivity of the parameters is very dependent on the scaling of the measurements.

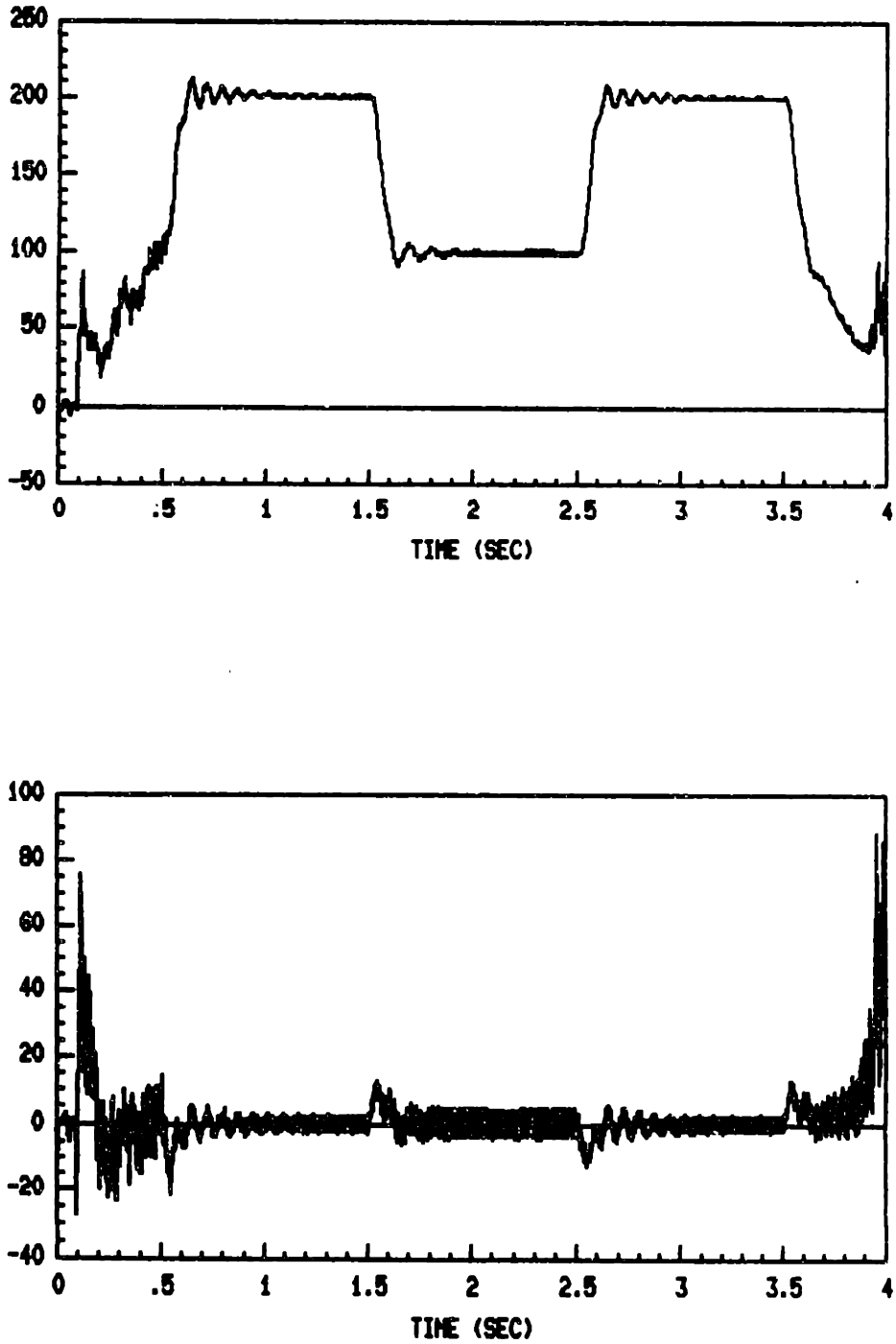


Figure 5.2: Performance of Estimator : (upper) Estimated Speed in rad/sec, (lower) Error in Speed Estimate in rad/sec

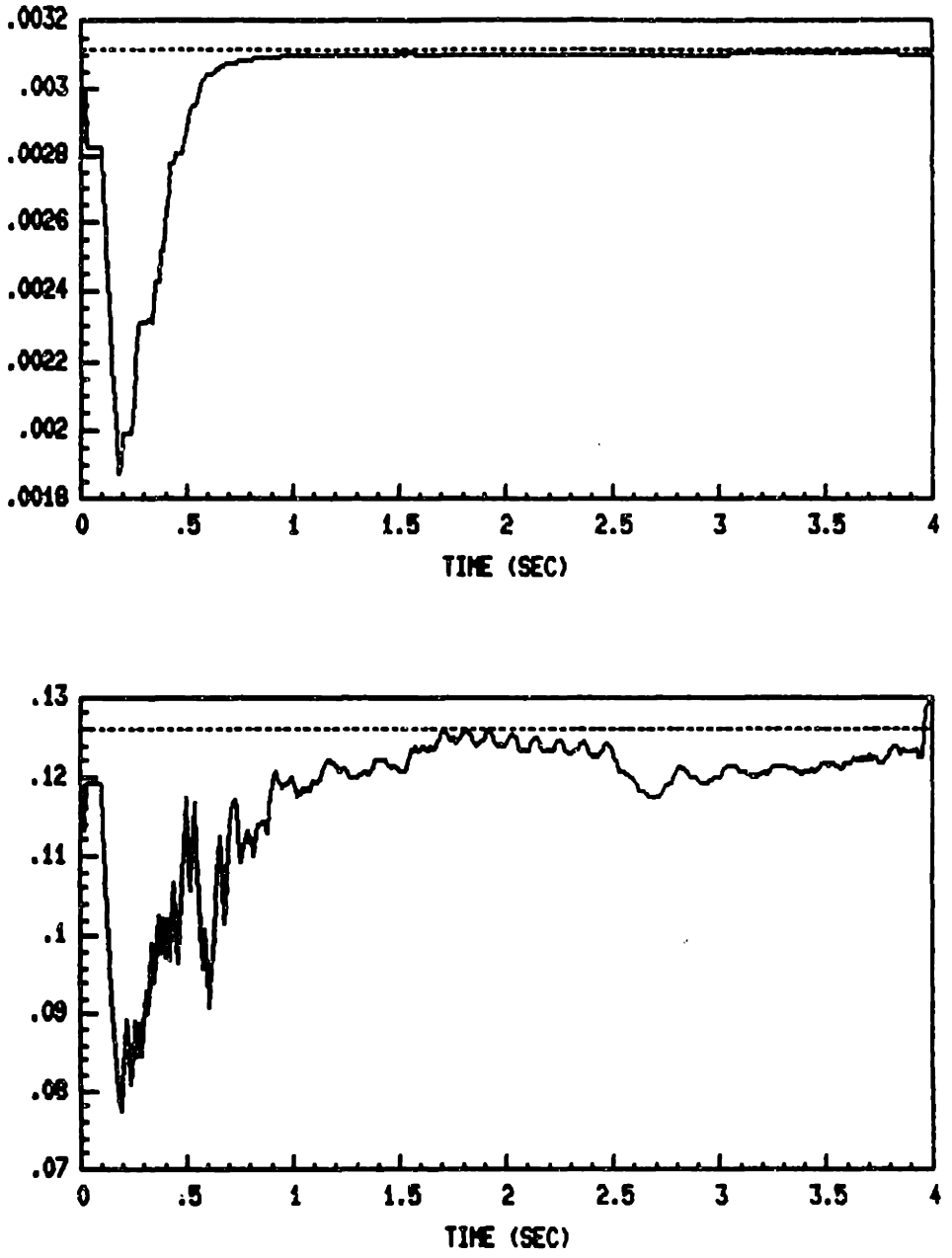
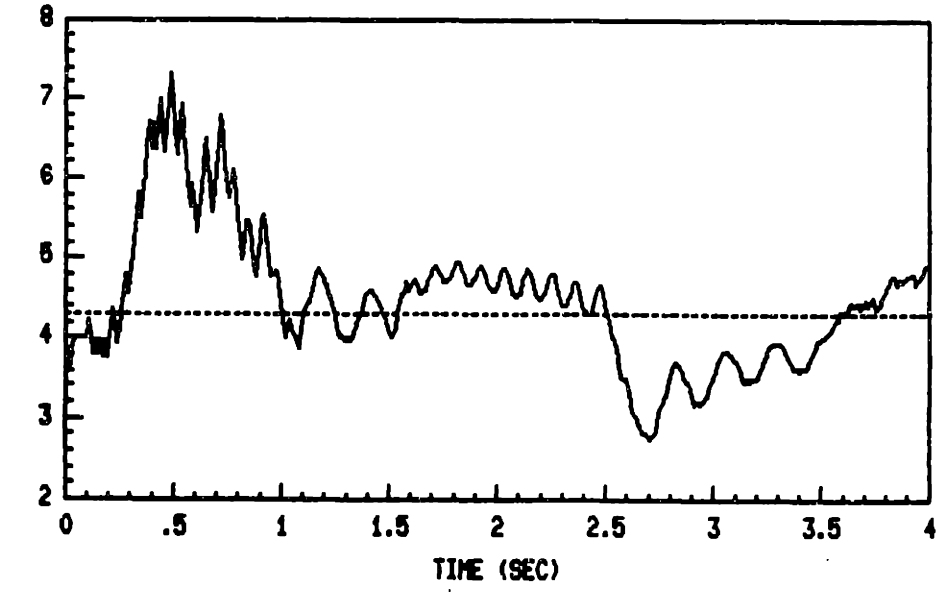


Figure 5.3: Performance of Estimator : (upper) Estimated p_1 and True p_1, (lower) Estimated p_2 and True p_2



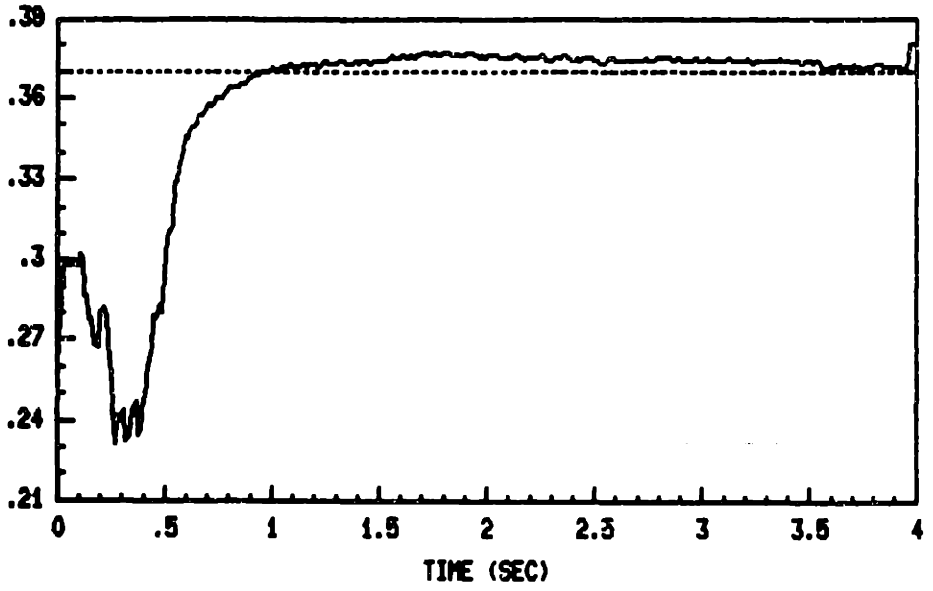


Figure 5.4: Performance of Estimator : (upper) Estimated p_3 and True p_3, (lower) Estimated R_s and True R_s

5.4 Estimation Results with Noise

In the previous section it was shown by simulation that the estimation algorithm has the ability to track the rotor speed and the parameters. In this section the case where the measurements of stator voltages and currents are corrupted by noise is studied.

In the usual least squares estimator the uncertainty is restricted to the measurements $\mathbf{y}(k)$. However, since in this linear regression model case not only $\mathbf{y}(k)$ but also $\mathbf{C}(k)$ is dependent on the measurements obtained from the state variable filters, the uncertainty is present in both of them. If the uncertainty on $\mathbf{C}(k)$ is small, a good estimate of θ should be expected based on the least squares criterion. Otherwise the estimates based on the least squares may be expected to be poor. This fact makes the validation of estimation in a noisy enviroment important in the present algorithm.

The estimation results with around 2 percent white noise on stator voltages and currents are shown in Figures 5.5 - 5.7. The initial values of parameters are same as the previous section. The simulation results shows the estimator can track the rotor speed. Though it is not a rigorous validation, it is enough for our purposes.

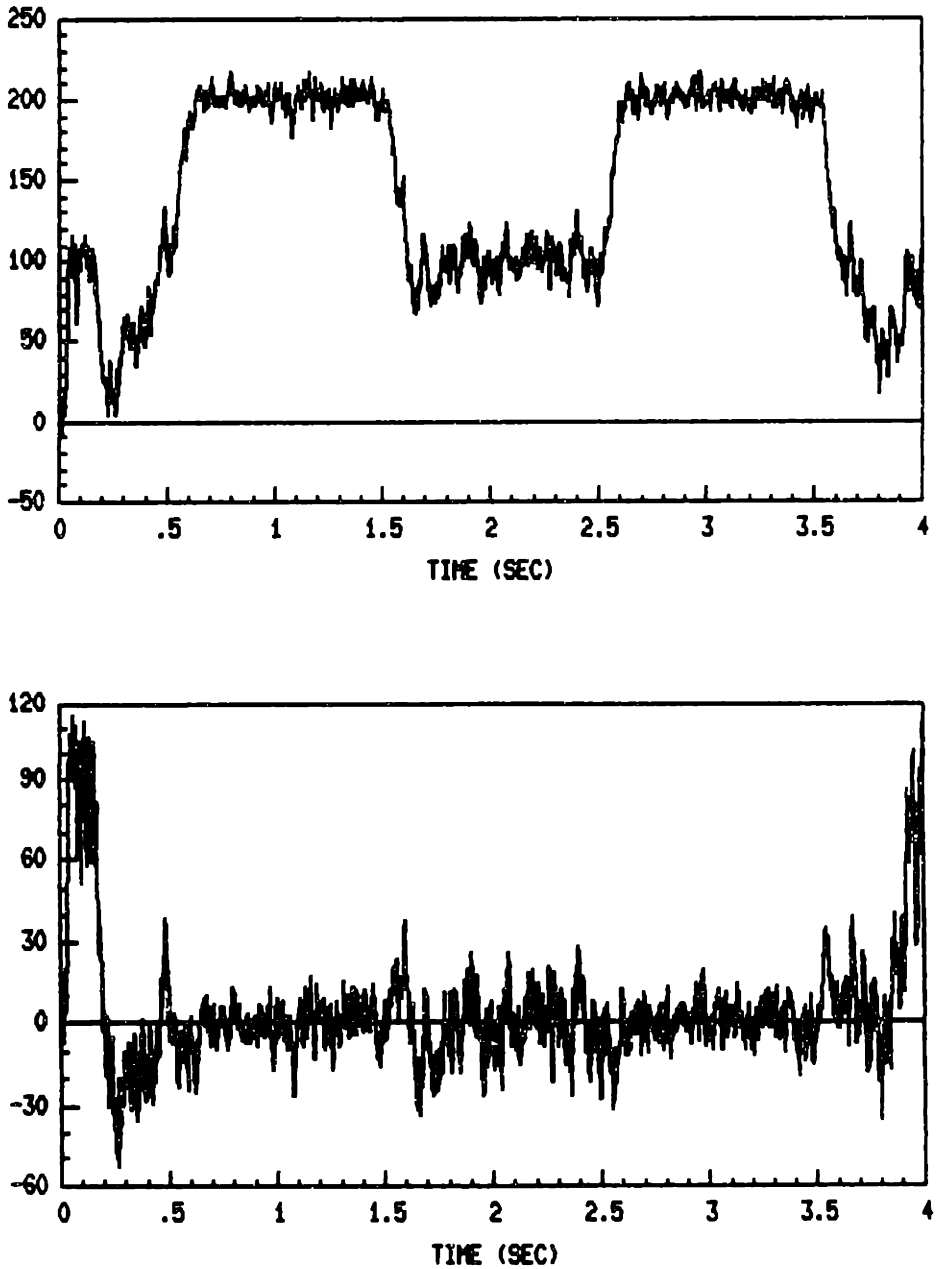


Figure 5.5: Performance of Estimator with Noisy Measurements : (upper) Estimated Speed in rad/sec, (lower) Error in Speed Estimate in rad/sec

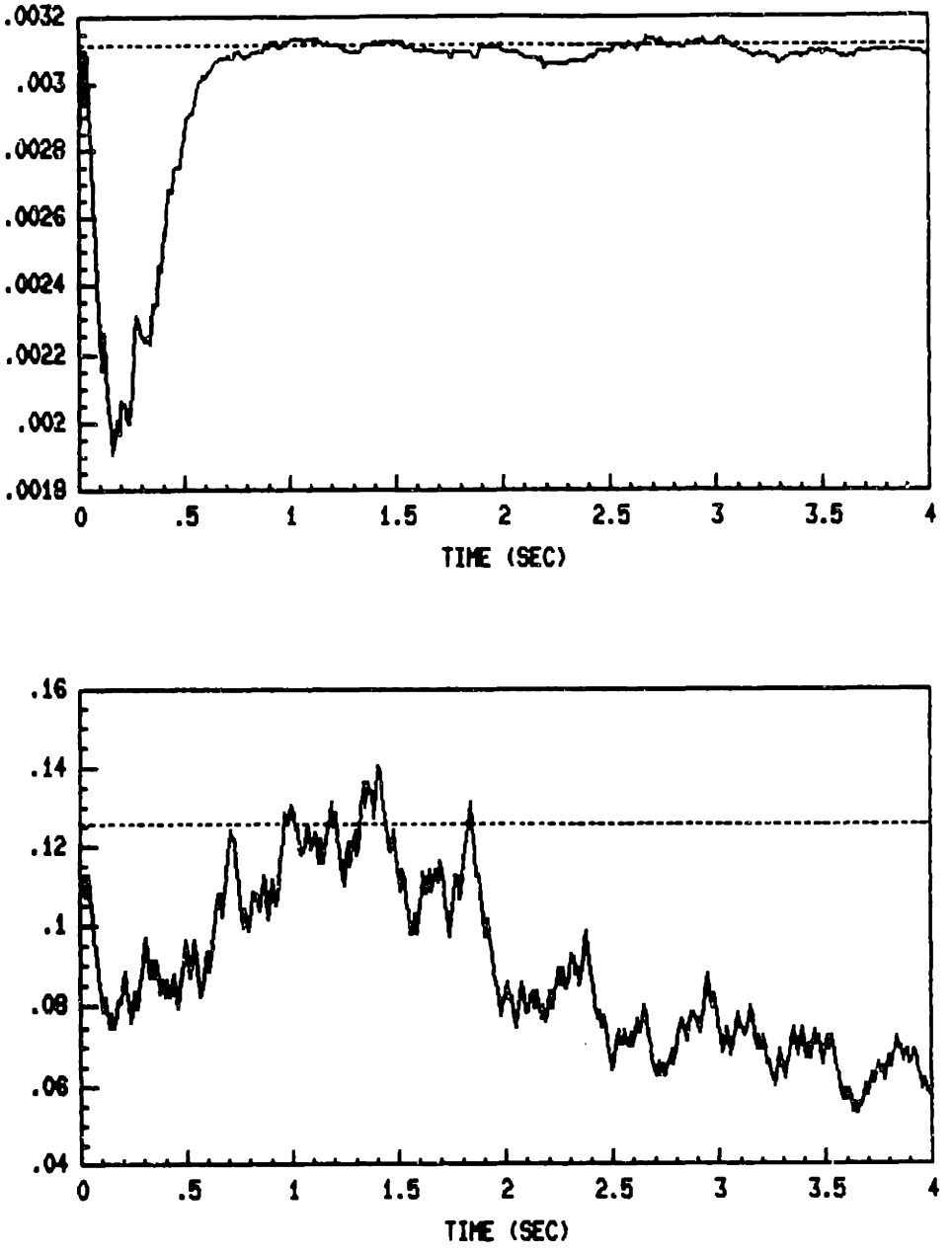


Figure 5.6: Performance of Estimator with Noisy Measurements : (upper) Estimated p_1 and True p_1, (lower) Estimated p_2 and True p_2

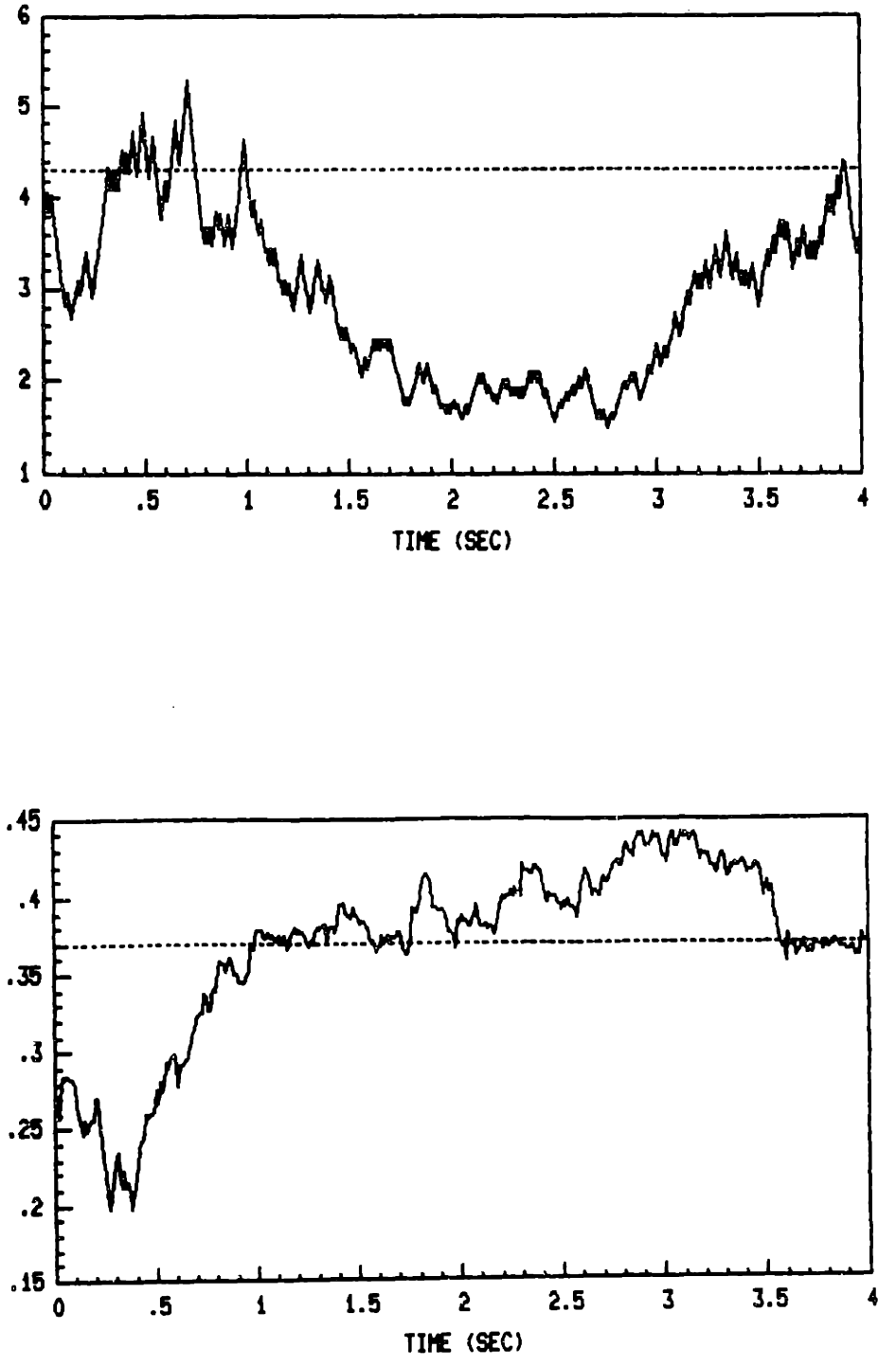


Figure 5.7: Performance of Estimator with Noisy Measurements : (upper) Estimated p_3 and True p_3, (lower) Estimated R_s and True R_s

Chapter 6

Off-Line Analysis of Experimental Data

6.1 Introduction

The estimator simulations presented in the previous chapter were useful for the initial analysis of the estimator algorithm. The next step to evaluate this algorithm is to test it off-line on data taken from an actual induction machine drive system. In order to test the estimator with real data, the test bench at the Technische Universität Berlin (TUB), built by Ms. D. Elten who is currently a Ph.D. candidate of TUB, is used. The system was designed to provide phase currents and line voltages under the open-loop voltage-fed control scheme. The rotor speed is also measured by a tachometer to compare with its estimated value. The induction machine used for this experiment was a 300-Watt 2-pole-pair induction machine.

Machine parameters of this induction machine obtained from "Blocked Rotor Test" and "No Load Test" [16] are shown in Table 6.1.

Parameter	Value
L_s	0.2731 H
L_r	0.2700 H
M	0.2550 H
R_s	5.50 Ω
R_r	7.00 Ω
number of pole pair	2

Table 6.1: Machine Parameters of a 300 Watt Induction Machine from Blocked Rotor Test and No Load Test

6.2 Experimental Set-Up

The data for off-line analysis is acquired using an Analog Data Acquisition Machine (ADAM) by René & Maurer, Switzerland. The acquired data is then transferred to an IBM-PC. From there it is downloaded to a VAX 11/750 computer, where it is processed using the $MATRIX_X$ software package on which antialiasing and state variable filters are implemented, followed by the estimator algorithm. Figure 6.1 shows a block diagram of the experimental system. All of the experimental data analyzed in this chapter was taken from this system.

In this experimental set-up, an open-loop voltage-fed control scheme is used. The stator voltage and frequency can be changed but not be controlled automatically[1] in closed loop.

In the transformer and filter unit the phase currents are picked up using Hall sensors

[1]This experimental set-up was originally designed for fault detection of induction machines under steady operating conditions.

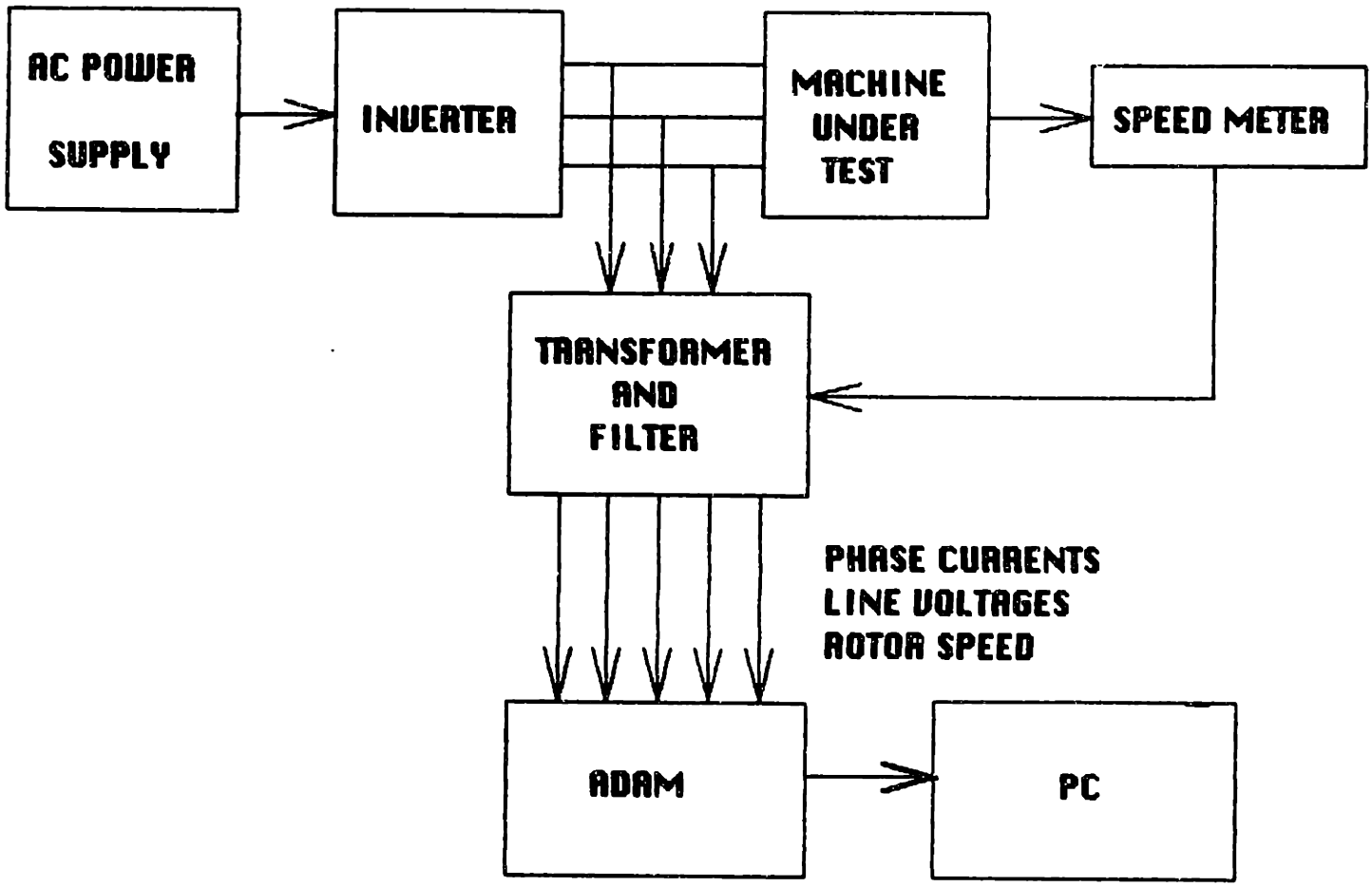


Figure 6.1: Experimental Set-Up

and the line voltages are picked up using resistors. The speed signal is picked up by a frequency/voltage converter after a 100Hz low pass filter. All measurements are scaled to an appropriate voltage range and sampled at 20kHz. Finally two phase currents, two line voltages and speed are obtained for a maximum of 6.4 seconds in a special format (not ASCII format). This time constraint and the format are imposed by the particular data acquisition system that was used.

6.3 Data Processing before Estimation

Since the data is not in ASCII format, it should be transfered to ASCII format using the CONTOA.C program written by Ms. D. Elten (see Appendix C). After that it can be fed to a VAX. The next step is adding header data for $MATRIX_X$ to load it.

The data is then fed to antialiasing filters state variable filters implemented using the $SYSTEM - BUILD$ function of $MATRIX_X$. The outputs of the state variable filter are subsampled (at 10 kHz) and scaled appropriately and transformed from the abc-frame to the $\alpha\beta$-frame, also using $SYSTEM - BUILD$ functions. Note that $MATRIX_X$ can accept a maximum of 100,000 variables. Also note that in order to get good differential characteristics, the sampling frequency of measurements should be 20 times higher than the cut-off frequency [14].

Before proceeding with the estimation, it is worth examining the waveforms of line voltages and phase currents shown in Figures 6.2 - 6.3 at ω =50Hz. Also power spectra are shown in Figure 6.4. The power spectra show promising characteristics for estimation because they have clean low-order harmonics. Note that a second order Bessel-type filter is used for the antialiasing filter.

Finally, the waveform of the voltages and currents in the $\alpha\beta$-frame are shown in Figure 6.5.

The transformation from phase currents in the abc-frame to those in the $\alpha\beta$-frame is shown in (6.1).

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{3}{2}} & 0 \\ \sqrt{\frac{1}{2}} & \sqrt{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \end{bmatrix} \tag{6.1}$$

Similarly the transformation from line voltages in the abc-frame to phase voltages in

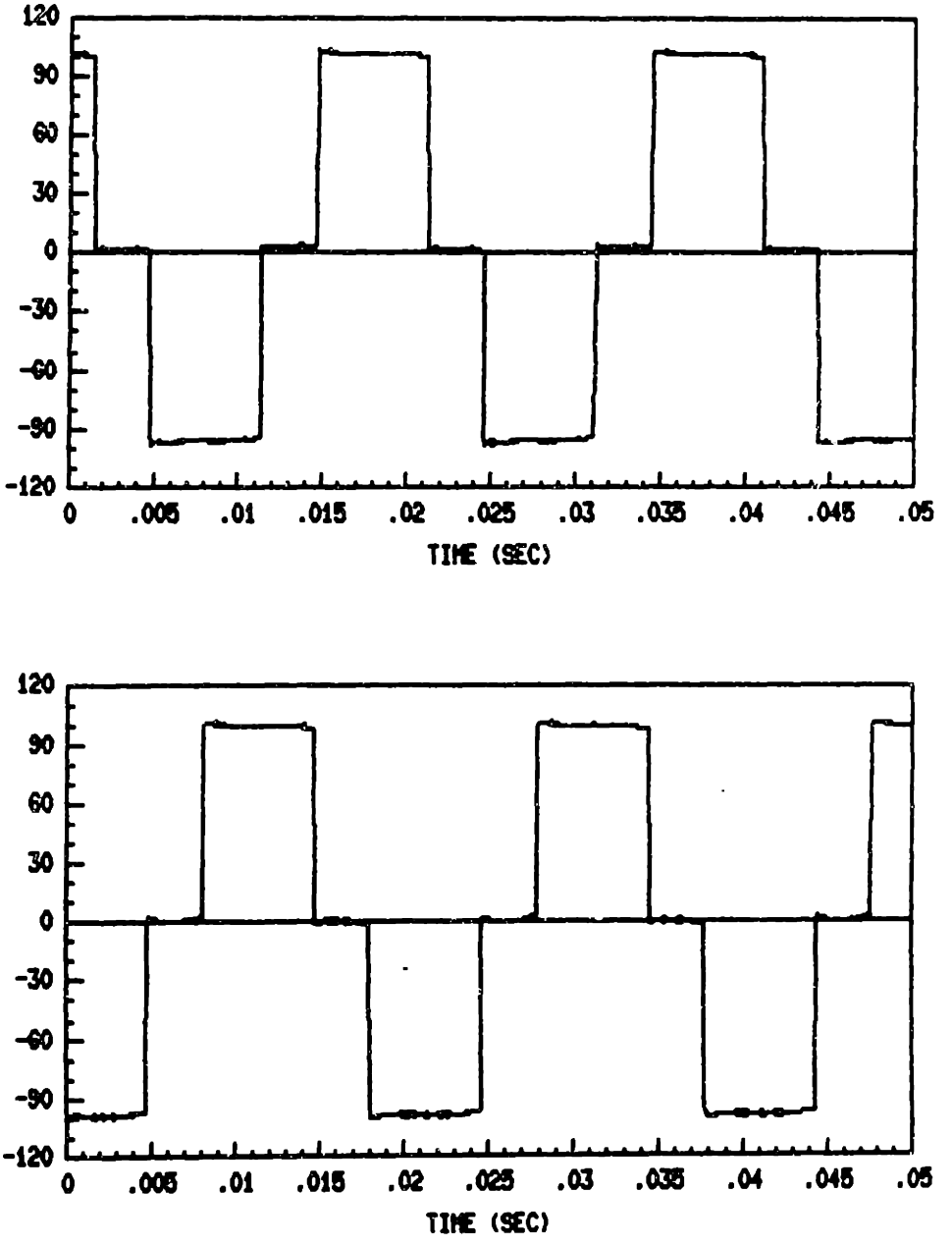


Figure 6.2: Line Voltage Waveform : (upper) V_{ab}, (lower) V_{b}

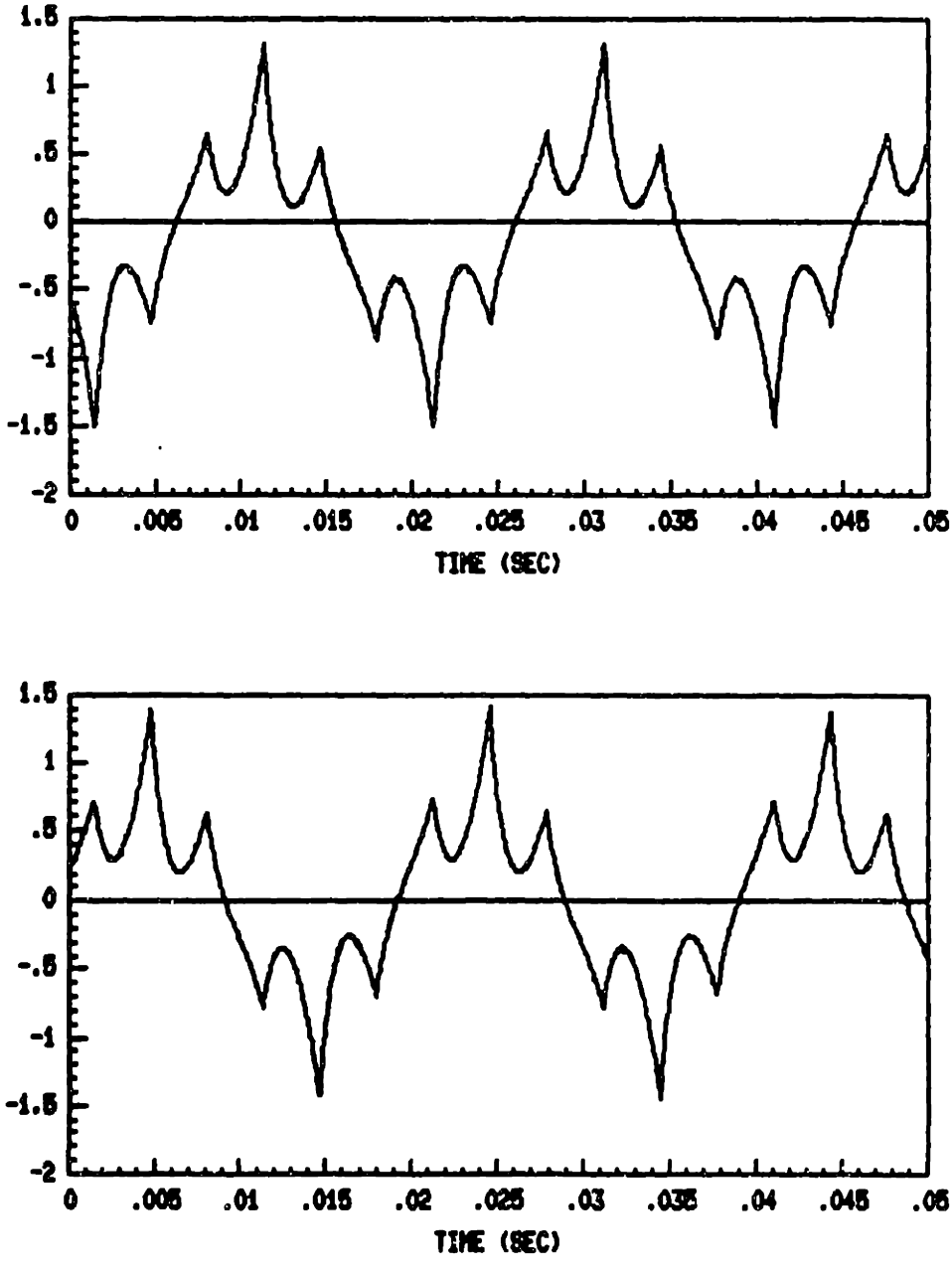


Figure 6.3: Phase Current Waveform : (upper) I_a, (lower) I_b

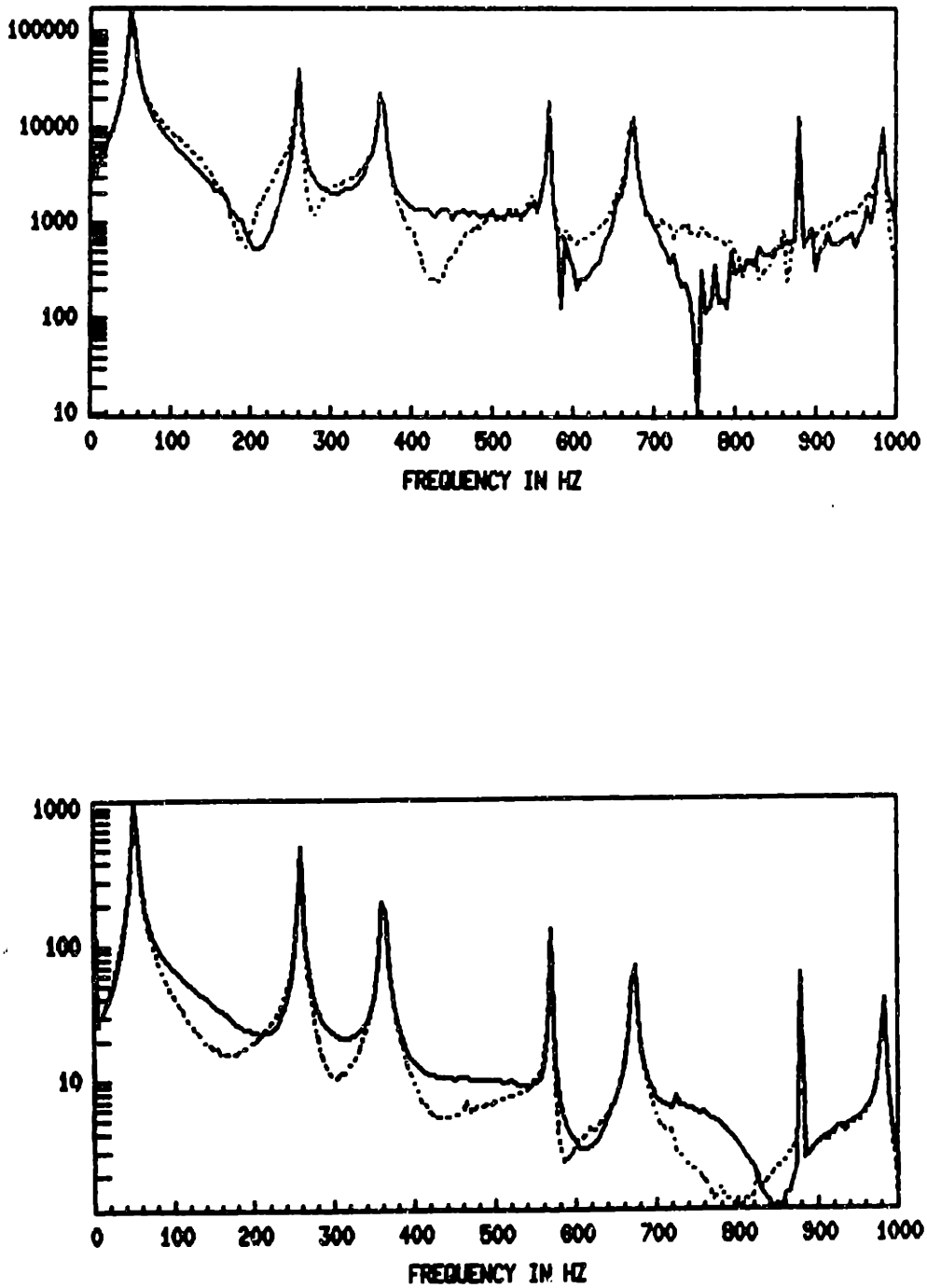


Figure 6.4: Power Spectrum : (upper) Line Voltage, (lower) Phase Current

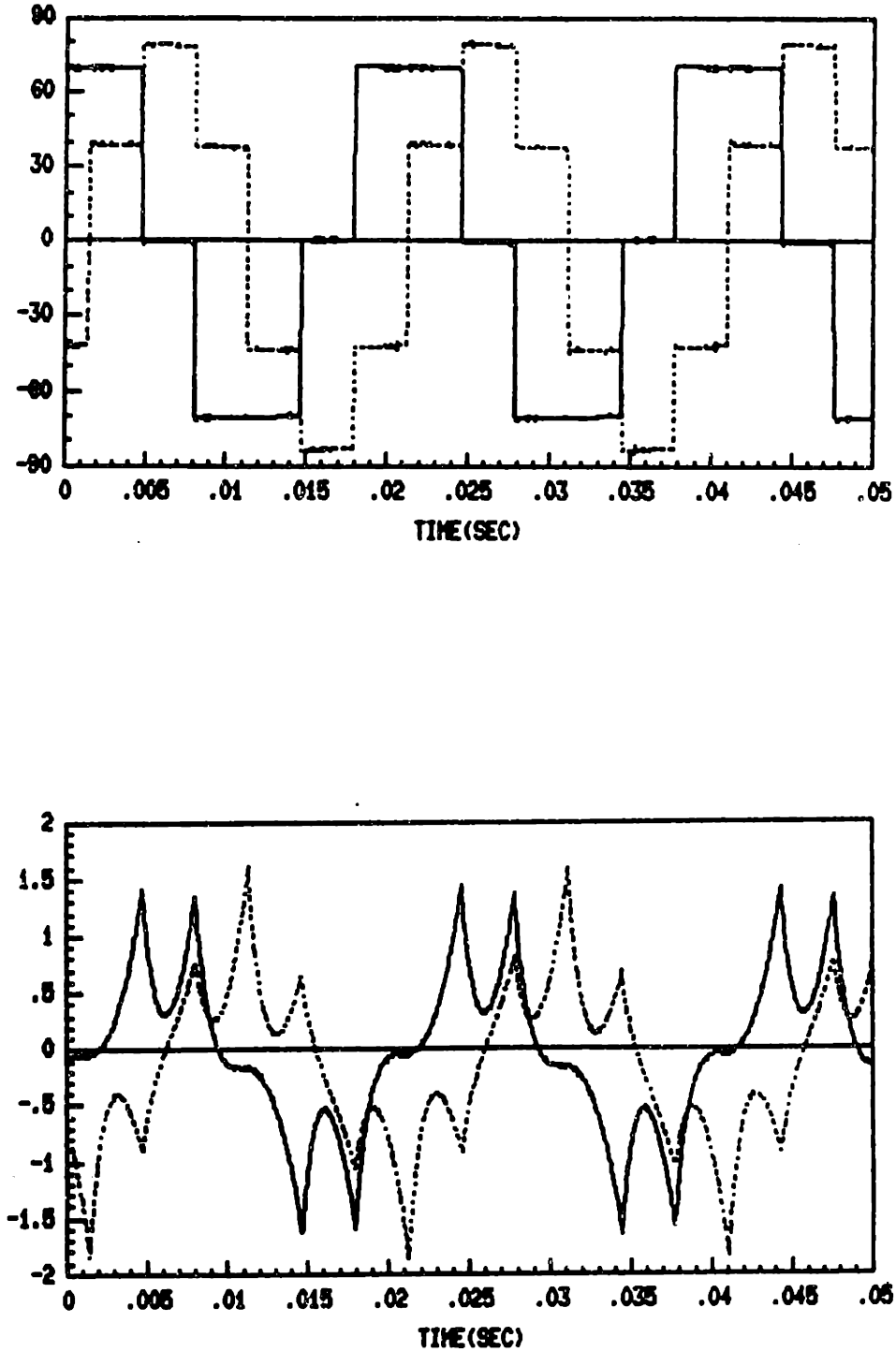


Figure 6.5: Waveform of Voltages and Currents in $\alpha\beta$-Frame : (upper) Voltages, (lower) Currents

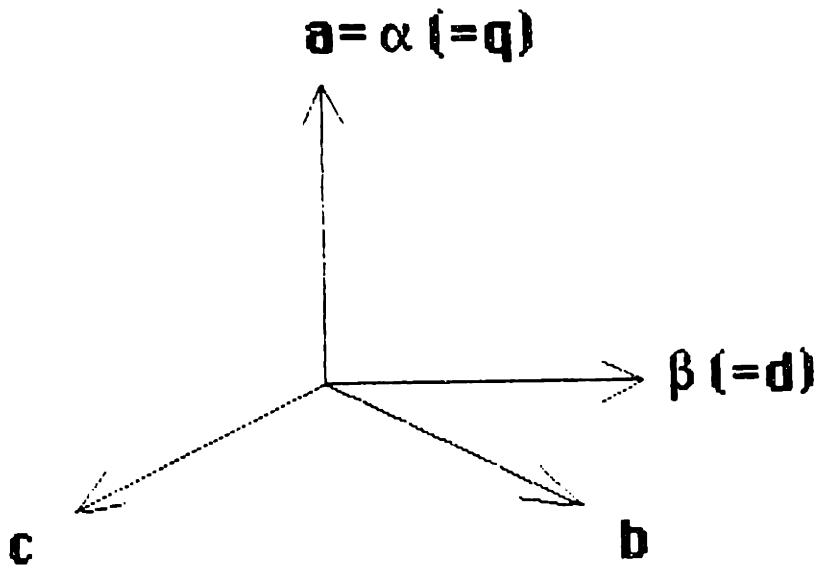


Figure 6.6: Relation between abc-Frame and $\alpha\beta$-Frame

the $\alpha\beta$-frame is shown in (6.2).

$$\begin{aligned} \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} &= \begin{bmatrix} \sqrt{\frac{2}{3}} & \sqrt{\frac{1}{6}} \\ 0 & \sqrt{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} -v_{ab} \\ -v_{bc} \end{bmatrix} \\ &= \begin{bmatrix} \sqrt{\frac{3}{2}} & 0 \\ \sqrt{\frac{1}{2}} & \sqrt{2} \end{bmatrix} \begin{bmatrix} v_a \\ v_b \end{bmatrix} \end{aligned} \tag{6.2}$$

where the relation between the abc-frame and the $\alpha\beta$-frame is shown in Figure 6.6.

6.4 Estimation

In this section two types of estimation are studied. First, different model estimation possibilities are studied using one-step batch least squares estimation. Second, recursive least squares estimation using the proposed algorithm is presented.

6.4.1 Model Comparison by Batch Least Squares Estimation

In this batch least squares estimation the complex transfer function model is used for computational simplicity. Since one of the coefficients in the transfer function (1.17) is not complex but real, this causes undesirable compensation in imaginary part which should be identically zero.

The first thing examined here is the possibility of estimation with unknown stator resistance. Table 6.2 shows the estimation results when the lefthand term in the linear regressor (2.1) is $\frac{d^2 i_s}{dt^2}$. Table 6.3 shows the estimation results when this term is $\frac{dv_s}{dt}$. Since rotor speed measured by a speed sensor is the most reliably available term in machine parameters and rotor speed, it is natural to evaluate the estimation performance by evaluating the rotor speed estimate. It is also checked that no parameter can take negative values. Neither of the above arrangements of the regressor seems to produce good estimation results.

Parameter	Data 1	Data 2	Data 3
$(R_s L_r + R_r L_s)/(L_r L_s \delta)$	720.73	727.59	794.93
ω_r	308.15	382.06	329.47
$R_s/(L_s T_r \delta)$	92512.	166060.	98716.
$\omega_r Rs/(L_s \delta)$	-42714.	164340.	144230.
$1/(L_s \delta) \cdots$(a)	37.5517	37.5496	37.4603
$1/(L_s \delta T_r) \cdots$(b)	2541.2	1374.4	1785.9
$\omega_r/(L_s \delta) \cdots$(c)	12337.	15599.	16627.
ω_r by speed sensor	316.7	379.8	442.8
ω_r by (c)/(a)	328.53	415.42	443.86
T_r by (a)/(b)	0.01478	0.02732	0.02098

Table 6.2: Estimated Parameters Based on Equation for $\frac{d^2 i_{ds}}{dt^2}$, Assuming R_s is Unknown

Parameter	Data 1	Data 2	Data 3
$L_s\delta$	0.0240	0.0240	0.0238
$R_s + (L_s/T_r)$	19.6106	19.4732	21.3899
$L_s\delta\omega_r$	4.3418	7.2090	6.0155
R_s/T_r	4650.0	5197.1	9868.4
$R_s\omega_r$	2440.8	2452.4	1495.8
$1/T_r \cdots$(a)	81.866	34.842	46.337
ω_r	236.41	316.08	335.46
ω_r by speed sensor	316.7	379.8	442.8
T_r by 1/(a)	0.01222	0.02870	0.02158

Table 6.3: Estimated Parameters Based on Equation for $\frac{dv_t}{dt}$, Assuming R_s is Unknown

The next thing to be examined is the possibility of estimation with known stator resistance. Table 6.4 shows typical estimation results when the lefthand term in the linear regressor (2.1) is $\frac{d^2 i_{sA}}{dt^2}$. Table 6.5 shows these estimation results when the lefthand term in the linear regressor is $\frac{dv_{sA}}{dt} - R_s \frac{di_{sA}}{dt}$. The estimated speed is compared against the measured rotor speed. It is also checked that no parameter takes negative values.

According to these results, the model with known stator resistance (=5.5Ω) can estimate better than the model with unknown stator resistance. Moreover, the model whose lefthand term is $\frac{dv_{sA}}{dt} - R_s \frac{di_{sA}}{dt}$ is better than the other.

Therefore, the empirically known fact that stator resistance is very difficult to estimate is observed here again. In addition the model whose lefthand term is $\frac{dv_{sA}}{dt} - R_s \frac{di_{sA}}{dt}$ turned out to be most promising in the estimation. Taking account of this result, a two-stage estimator for slow (parameter) estimation (one for stator-resistance estimation, the other for estimating the remaining parameters) is proposed.

Other batch least squares estimations with different stator resistance values (= 2.75Ω, 11Ω) were done. Their results show that the estimated rotor speed is quite robust to the stator resistance variations, in other words, quite decoupled from the stator resistance variations, though detailed results are not presented here.

Parameter	Data 1	Data 2	Data 3
$1/(T_r\delta)$	475.012	579.205	562.422
ω_r	171.816	255.187	210.178
$1/(L_s\delta)\cdots$(a)	36.8985	37.4569	36.563
$1/(T_rL_s\delta)\cdots$(b)	1438.4	1818.4	1734.2
$\omega_r/(L_s\delta)\cdots$(c)	10868.	13422.	14923.
ω_r by (c)/(a)	294.54	358.33	408.14
T_r by (a)/(b)	0.02565	0.02060	0.02509
ω_r by speed sensor	316.7	379.8	442.8

Table 6.4: Estimated Parameters Based on Equation for $\frac{d^2i_s}{dt^2}$, Assuming $R_s = 5.5\Omega$

Parameter	Data 1	Data 2	Data 3
$\delta L_s\cdots$(a)	0.0252	0.0248	0.0250
L_s/T_r	13.021	15.652	15.641
$\delta L_s\omega_r\cdots$(b)	5.9147	8.6193	8.8300
$1/T_r\cdots$(c)	41.275	51.173	50.940
ω_r	301.27	367.56	423.22
ω_r by (b)/(a)	234.71	347.55	353.20
T_r by 1/(c)	0.02423	0.01954	0.01963
ω_r by speed sensor	316.7	379.8	442.8

Table 6.5: Estimated Parameters Based on Equation for $\frac{dv_s}{dt} - R_s\frac{di_s}{dt}$, Assuming $R_s = 5.5\Omega$

6.4.2 Recursive Least Squares Estimation

In this section some of the recursive estimation results are presented. The values of Table 6.1 are used as initial estimates of the parameters. It is important to remember that the parameter estimates based on Blocked Rotor Test and No Load Test might not be very good because the actual operating condition is different from these test conditions. Note that stator resistance value can be obtained just by checking the resistance value between terminals.

Two estimation results are presented. One is a speed-up transient, shown in Figures 6.7 - 6.9. The other is a speed-down transient, shown in Figures 6.10 - 6.12.

The results show good estimation in steady state, but the estimate displays a highly undesirable "kick" in the wrong direction at the start of the transient. The possible reasons for this undershoot are explored via simulations in the next section.

The oscillation in the speed estimates at lower speed is correlated with the fundamental frequency of measurements. Since there is no oscillation of speed estimate at higher speed, too many harmonics included in measurements might cause this oscillation. Taking the moving average over one electrical cycle will surely reduce this bounding. This method give us a more smooth estimate, but the tracking ability is degraded. This tradeoff between noise insensitivity and tracking ability always exists in estimation.

Sharp changes in estimated values might be caused by the insufficient suppress of high frequency harmonics in measurements. These can be eliminated or reduced by installing shaper low pass filters before measurements are fed to state variable filters.

In this estimation richness detector was implemented only for monitoring the richness of the corresponding regressor variables. But since there was neither symptom of estimator windup nor the estimated value deviation from true value when the norms

of ε and **PC** were small, the stopping of updating the parameters and the covariance matrix was not executed.

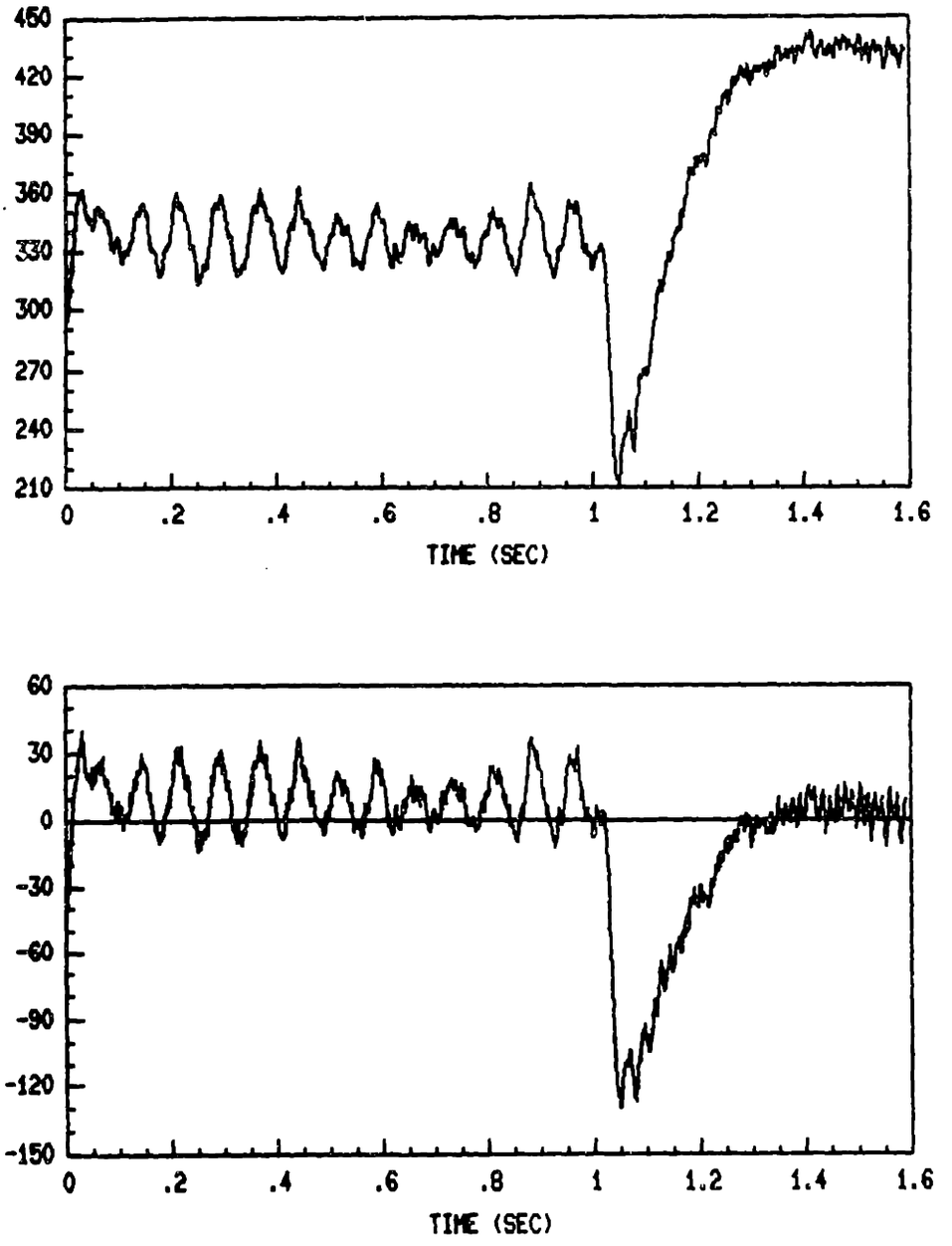


Figure 6.7: Speed Estimation : (upper) Estimated Speed in rad/sec, (lower) Speed Error in rad/sec

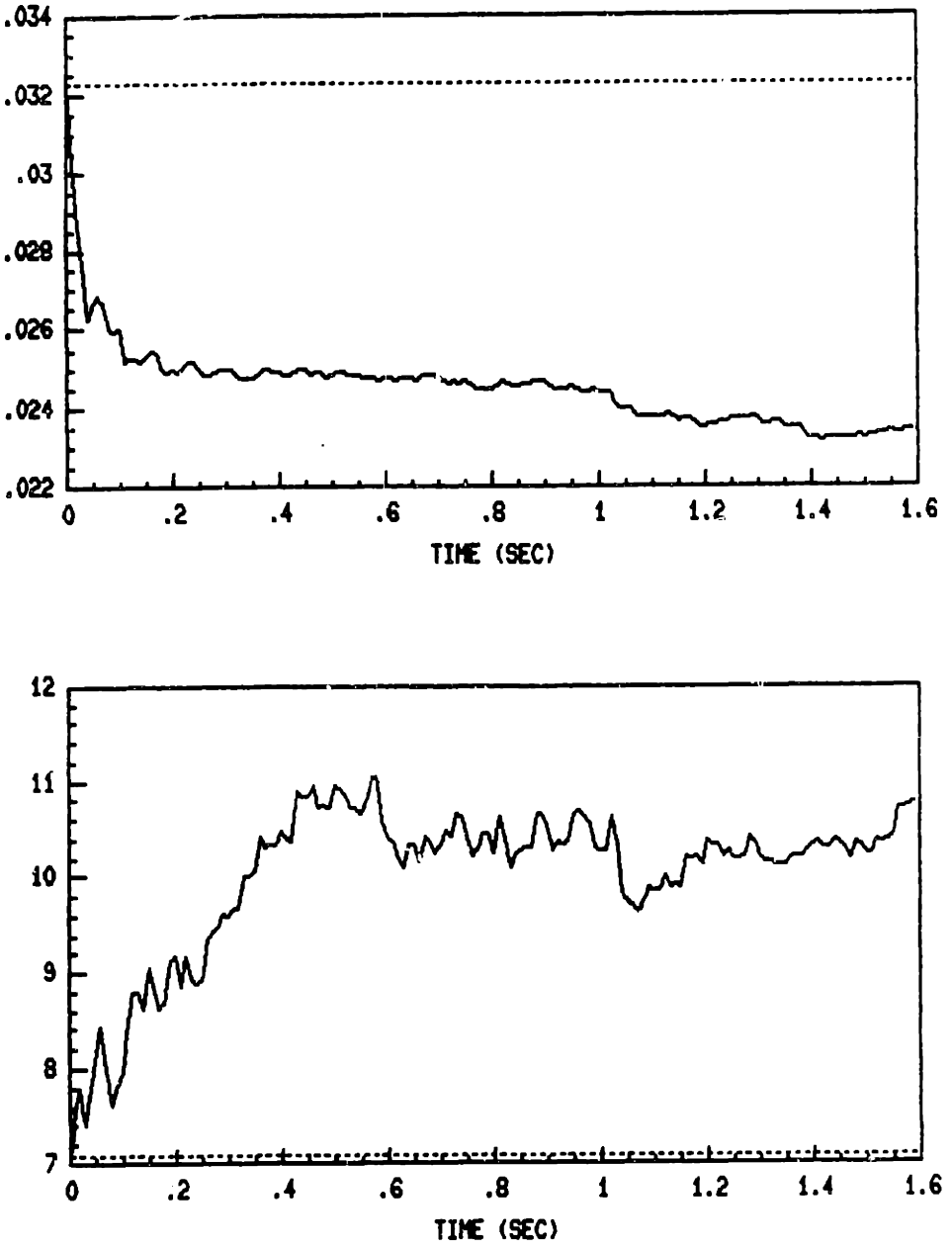
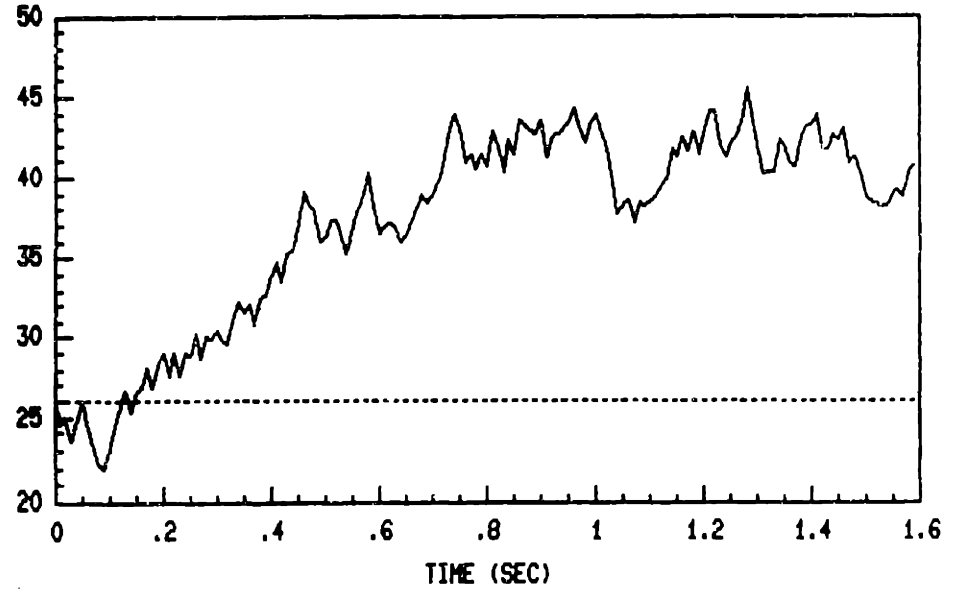


Figure 6.8: Parameter Estimation : (upper) p_1, (lower) p_2



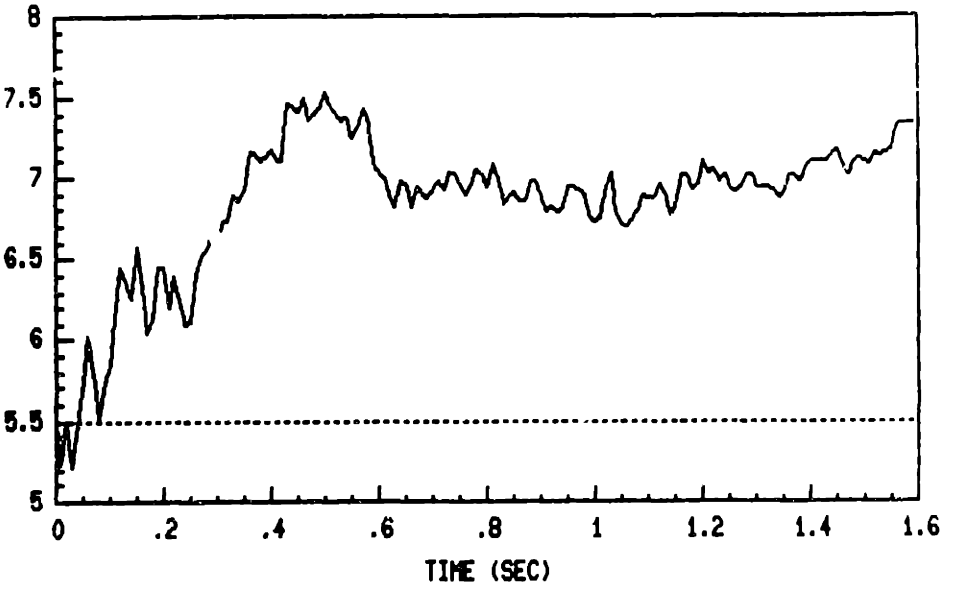


Figure 6.9: Parameter Estimation : (upper) p_3, (lower) R,

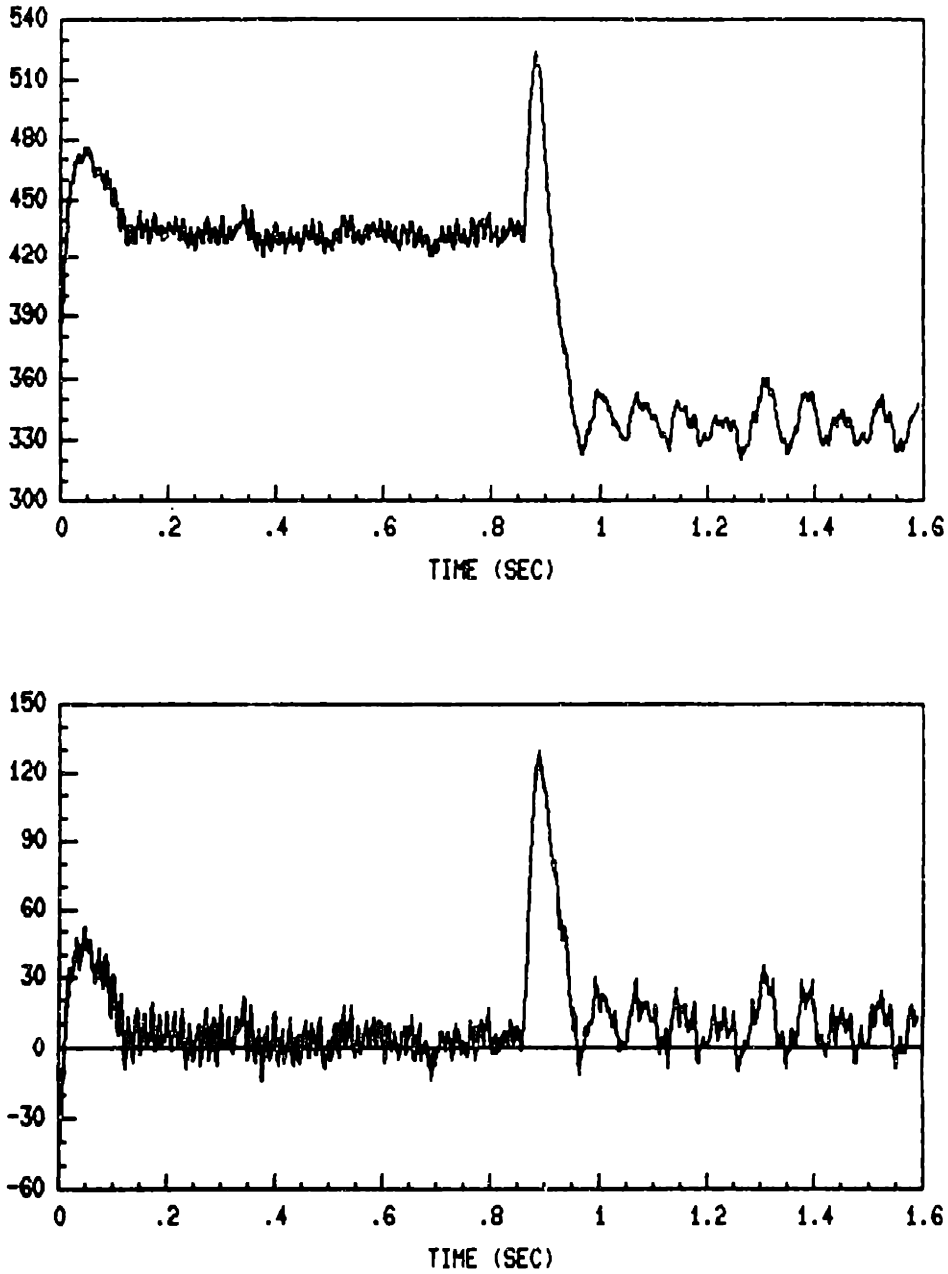


Figure 6.10: Speed Estimation : (upper) Estimated speed in rad/sec, (lower) Speed Error in rad/sec

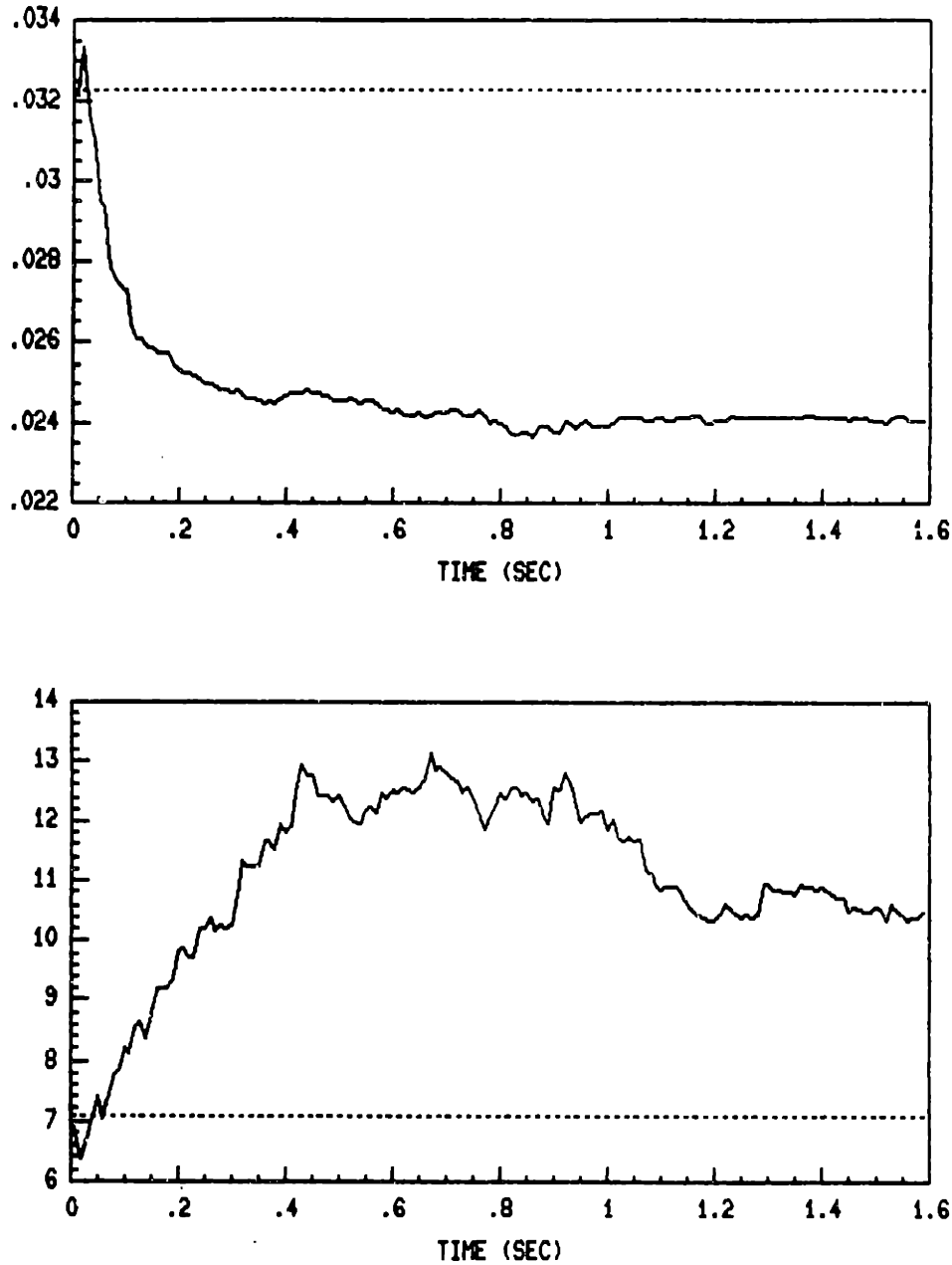
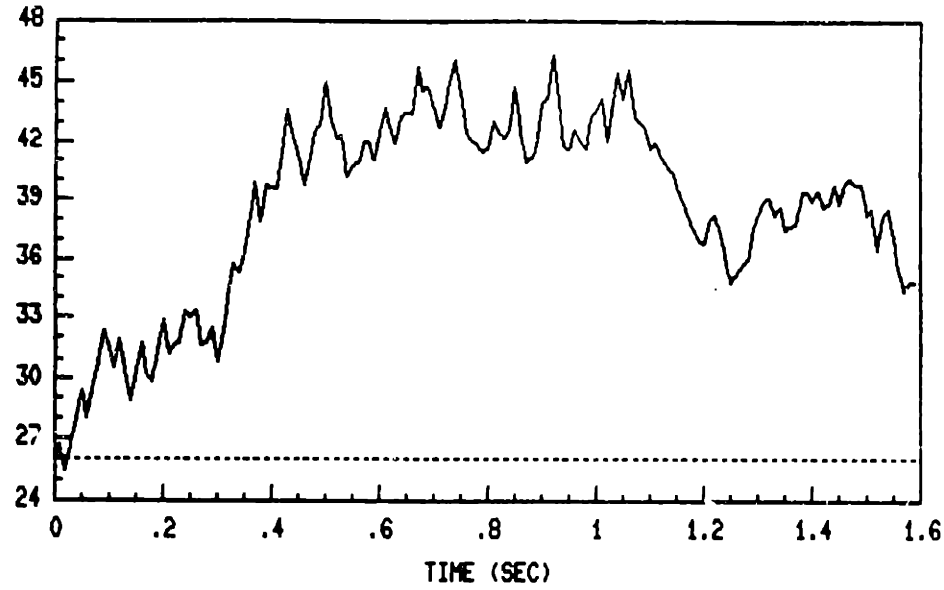


Figure 6.11: Parameter Estimation : (upper) p_1, (lower) p_2



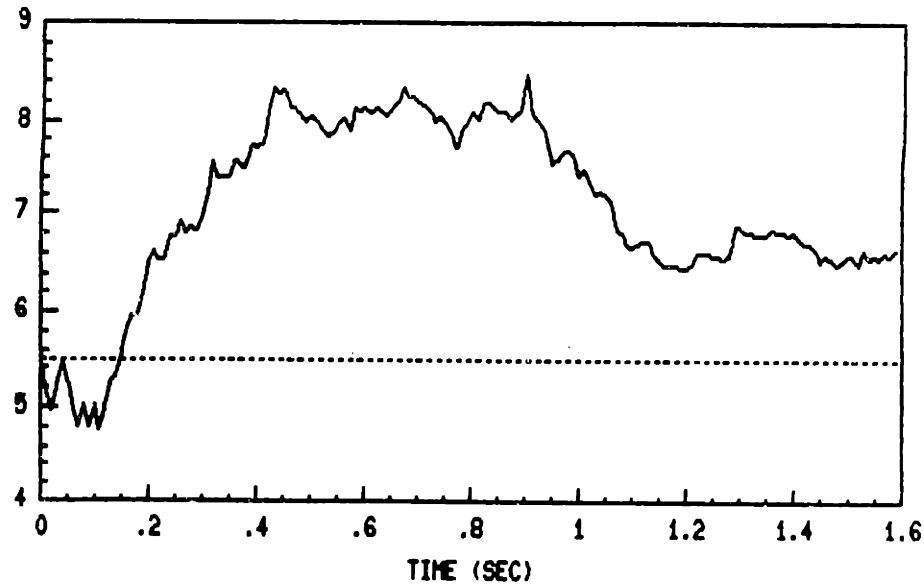


Figure 6.12: Parameter Estimation : (upper) p_3, (lower) R_s

6.5 Simulated Data

In order to analyze the undershoot in the transients displayed in Figure 6.7 and 6.10 (similar to the step response of a system containing a right-half-plane zero), simulated data *under noise-free environment* were generated. Data for a speed-up transient shown in Figure 6.13 - Figure 6.15 displays good similarity between actual data and simulated data, and thereby providing some confidence in the quality of the simulation.

The inertia of machine is assumed $0.0002 Kgm^2$. The machine parameters used in simulation are from Table 6.1.

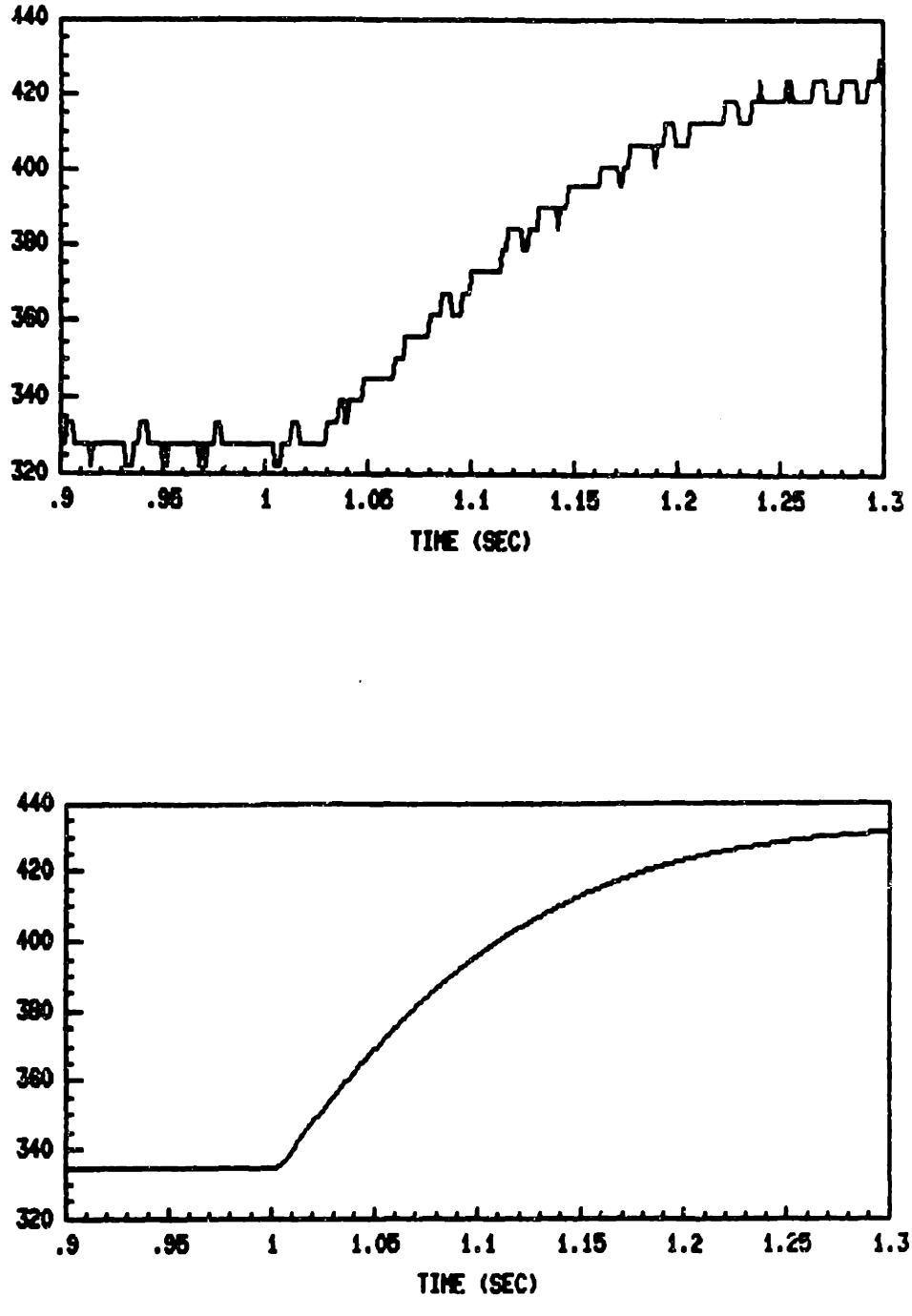


Figure 6.13: Measured Speed in rad/sec : (upper) Actual, (lower) Simulated

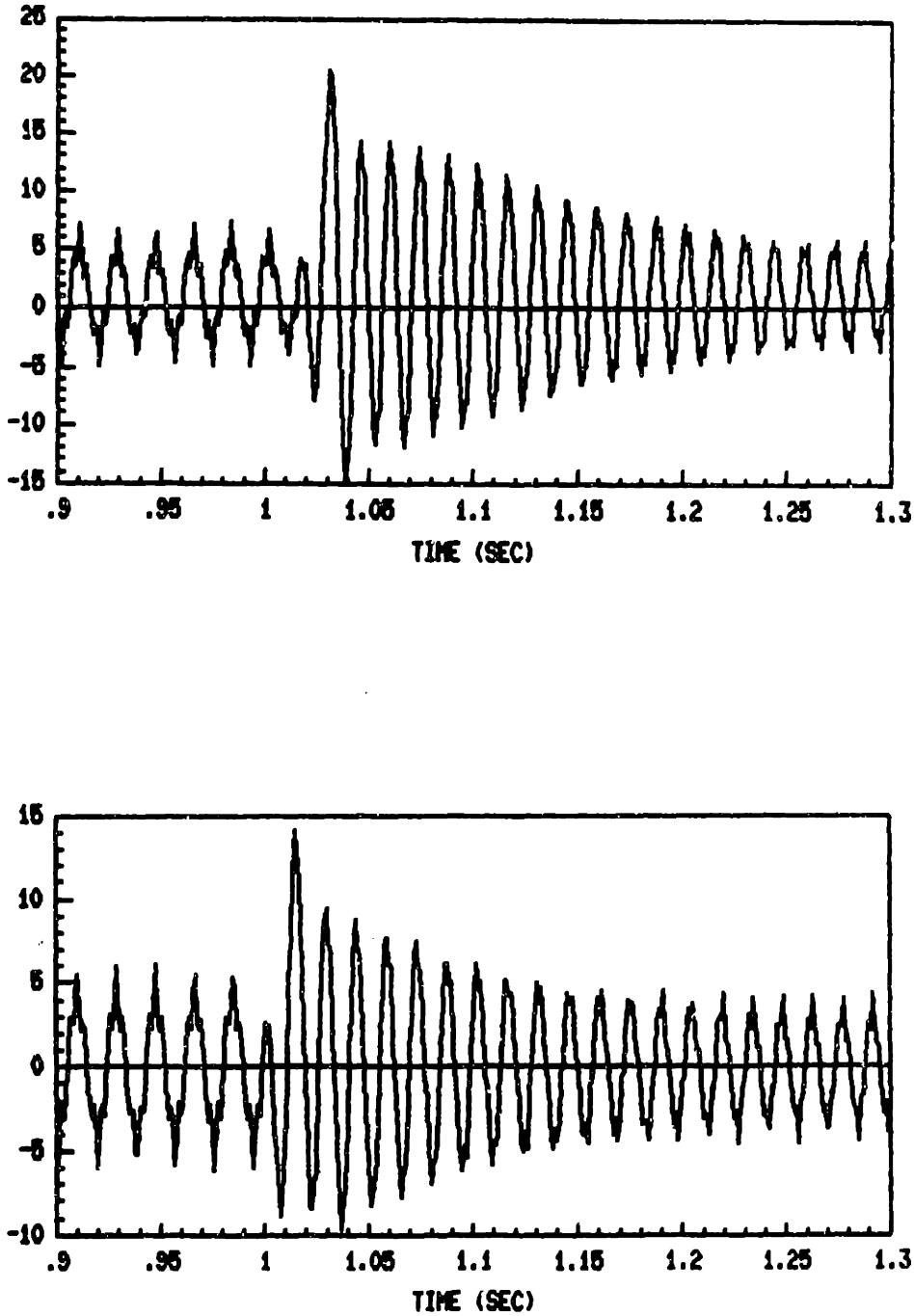


Figure 6.14: Stator Current i_α : (upper) Actual, (lower) Simulated

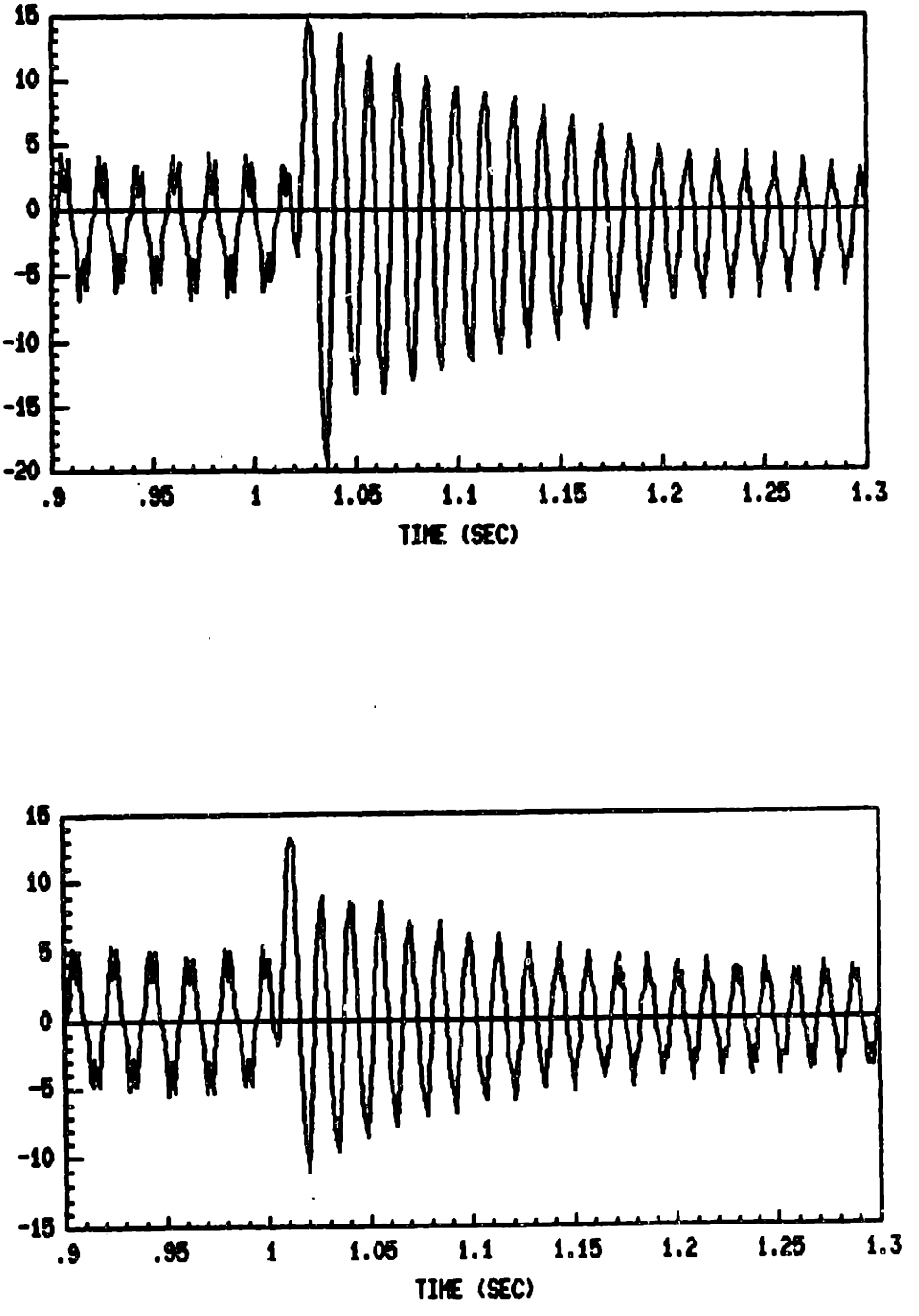


Figure 6.15: Stator Current i_β : (upper) Actual, (lower) Simulated

The estimation algorithm was checked using the simulated data. The result is shown in Figure 6.16 - Figure 6.18. It has a small sign of undershoot phenomena at the start of the transient state. Compared with other simulation results (though they are not presented here), the undershoot phenomena is more significant as the parameter estimation is poorer. *This phenomena is quite significant when the estimated rotor resistance is bigger than the true value,* though there might be other combinations of wrong (biased) estimated parameters.

One possible reason why the phenomena is so significant in the estimation using actual data is that the actual measurements at the transient state is not so rich as those at the steady state considering the noisy environment. Note that simulated data were generated *under noise-free* environment. Lack of the richness may cause biased estimator in parameters and therefore cause the undershoot phenomena.

When the measurements at the transient state are rich, another possible reason of this phenomena is that the speed estimation at the transient state might be more suceptable to poor machine parameter estimation than the steady state since at the steady state speed is estimated quite well.

Other possible reasons of this phenomena are skin effect, core loss, magnetic nolinearity, and unsymmetric windings which are negelected in the modeling procedure.

According to Figure 6.17 and 6.18 the estimated values of p_2, p_3, and R_s do not converge to true values. The estimated parameters p_2 and p_3 seem to compensate each other to get good estimate for the most sensitive parameter p_1 and this result seems to affect the estimate of R_s. According to [3], this type of biased estimator might not be a severe problem in adaptive control.

Some analysis of the effect of parameter number reduction in estimation using this simulated data is presented in Appendix E.

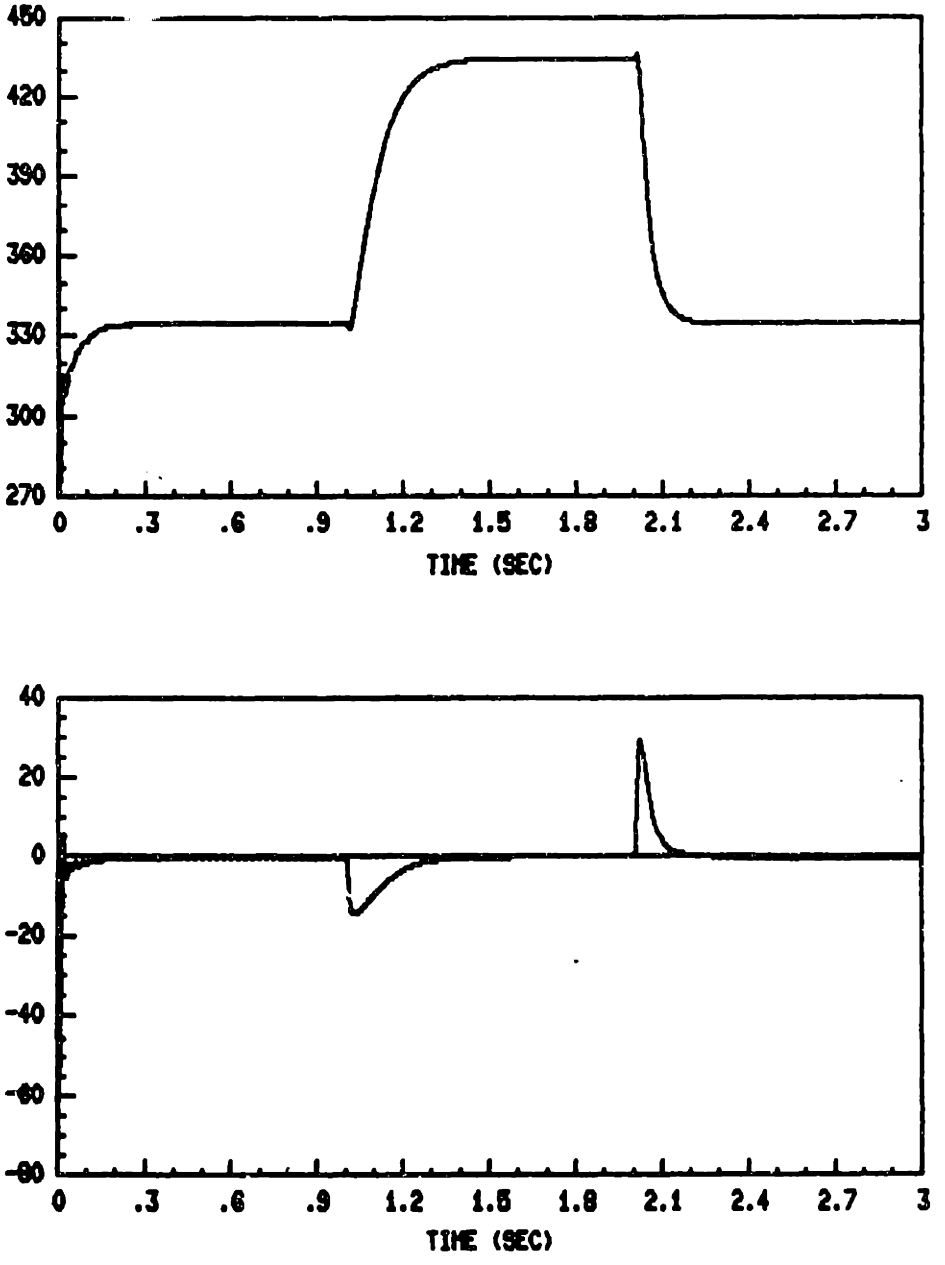


Figure 6.16: Speed Estimation : (upper) Estimated speed in rad/sec, (lower) Speed Error in rad/sec

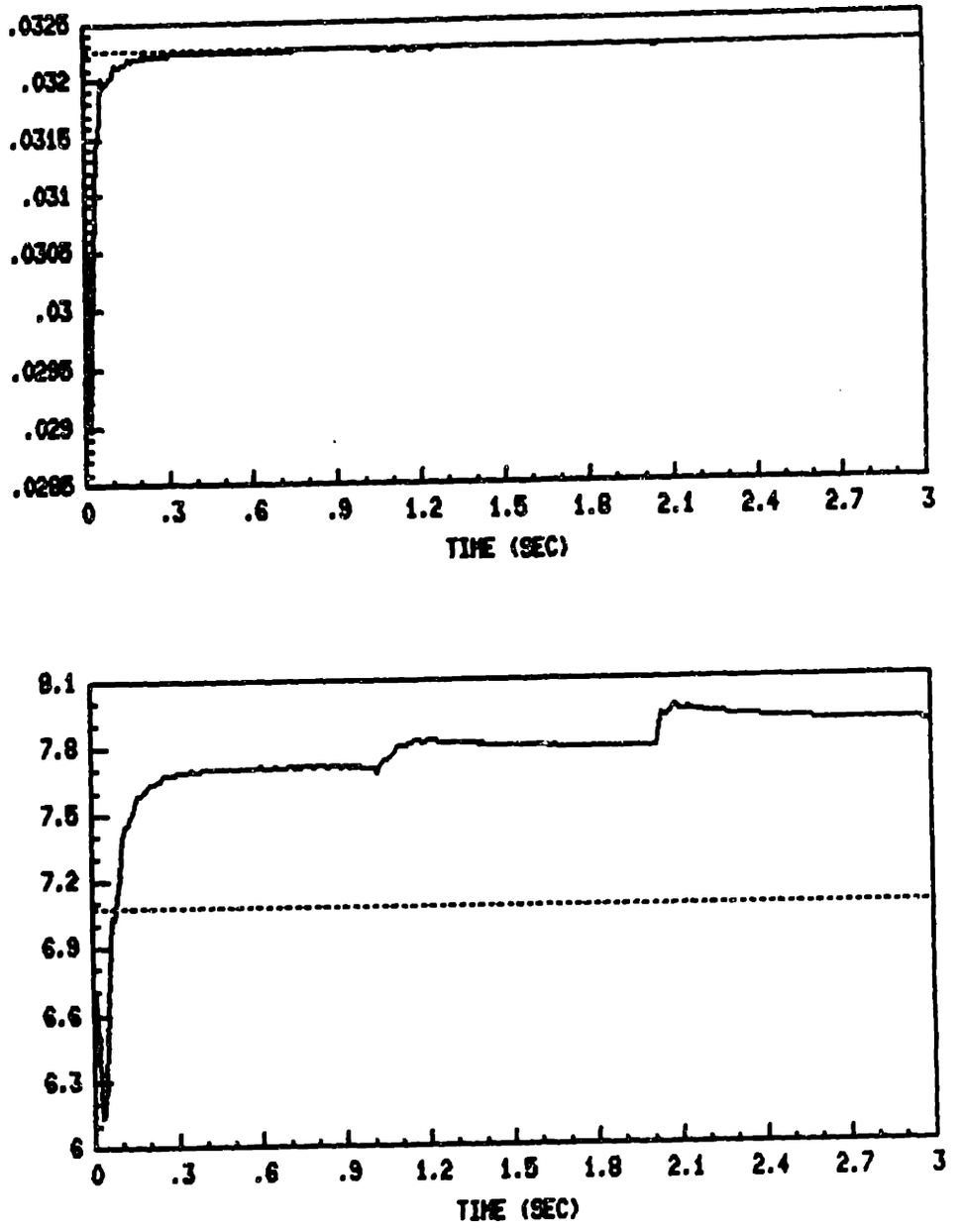
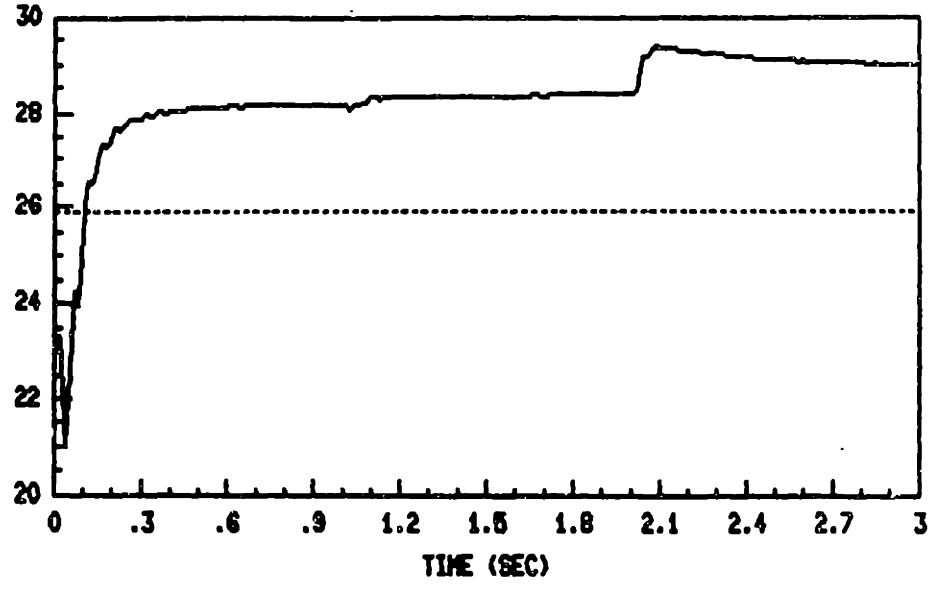


Figure 6.17: Parameter Estimation : (upper) p_1, (lower) p_2



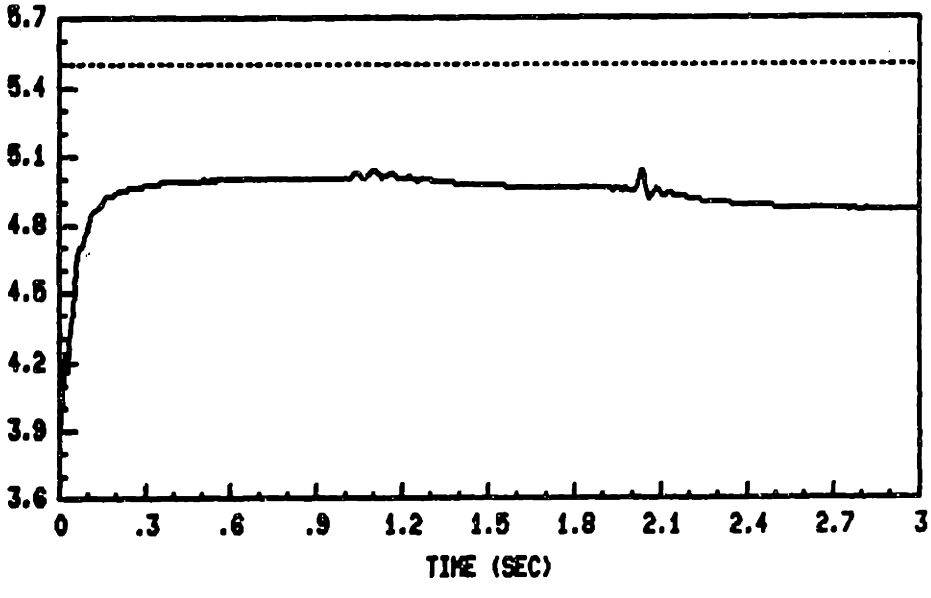


Figure 6.18: Parameter Estimation : (upper) p_3, (lower) R,

Chapter 7

Summary and Future Work

7.1 Summary

This thesis addressed the problem of estimating the rotor speed and machine parameters based on measurements of stator voltages and currents. The ideas and tools of parameter estimation theory were combined with the properties of induction machines, empirical ideas were used to improve the estimation, and modifications were made to enable the estimation algorithm to (eventually) operate in real time. The contribution of this research work is the methodology used to derive a real time implementable estimator.

Chapter 2 described theoretical background which is necessary to develop the estimation scheme, including practical issues for real time estimation.

In Chapter 3, three linear regression models were derived. One relates rotor speed to the stator voltages and currents, and their derivatives. Another relates stator resistance to the stator voltages and currents, and their derivatives. The third relates other machine parameters to the stator voltages and currents, and their derivatives. An

approach to avoiding taking the derivative terms of stator voltages and currents was also discussed.

Chapter 4 described how to organize the whole estimation scheme using the three linear regression models derived in Chapter 2. First the structure of model-based estimation was presented. The estimation algorithm is based on a recursive least squares approach. Two-time-scale separation was applied between the speed estimation and the machine parameter estimation. Two-stage separation was applied between stator resistance estimation and the estimation of the other machine parameters. Then, to take account of tracking capability and eliminate or reduce so-called estimator windup phenomena, variable forgetting factor and periodic covariance reset methods were used.

In Chapter 5 the performance of the proposed estimator was studied by numerical simulations. The measurement data is generated by a three-pulse PWM voltage-fed inverter under closed-loop slip frequency control. The estimator showed good tracking performance except for low speed transients. The performance with observation noise was also presented.

Estimation results using an actual 300 watt induction machine were presented in Chapter 6. The control scheme used in this experiment was an open-loop constant-voltage frequency-fed drive. The results showed that the proposed estimator can converge to appropriate values once a machine reaches a steady state, even though it may not give good results during the transient. The analysis of the undershoot phenomena at the beginning of the speed transient has not completed yet. This result might be caused by lack of richness in signal and/or poor machine parameters estimation, especially poor rotor resistance estimation. Another possible reason is the assumption made in modeling process.

7.2 Future Work

First of all, the undershoot phenomena at the beginning of the speed transient should be analyzed completely because it spoils the control when this estimator is implemented in closed-loop control. If necessary, a nonlinear estimation algorithm should be developed.

The next thing to be done is to increase the order of the antialiasing filter. The sharper the slope, the smoother the estimation result might be.

The real-time implementation of this algorithm is the next issue. The implementation of the state variable filter is especially important. Since it was implemented in $MATRIX_X$ in this research, there was no need to worry about the constraint of an actual circuit. If it is implemented in hardware, the throughput and scaling of the signal should be considered. One way to avoid this constraint is to install a digital filter instead of an analog circuit. This causes more time delay than an analog filter, however.

After all the work of tuning up the estimator, there are still issues related to the implementation of an adaptive controller using the proposed estimator. For this purpose, it is necessary to develop an algorithm to get physical parameter values from estimated parameters. By getting physical parameters a robust speed control can be realized.

Appendix A

Two-Axis Machine Model in Stator-Fixed Coordinates

A.1 Introduction

In this Appendix, the two-axis ($\alpha\beta$-frame) machine model in stator fixed coordinates is presented. First, the electrical machine model is presented, both without core loss and with core loss. Second, the mechanical model is presented.

A.2 Electrical Model

The electrical model is derived using the equivalent circuit per axis shown in Figure A.1.

Since for a squirrel cage induction machine the rotor circuit is short circuited, the following equations are obtained.

$$\mathbf{v}_s(t) = R_s\mathbf{i}_s(t) + \frac{d\lambda_s(t)}{dt} \tag{A.1}$$

$$\mathbf{0} = R_r \mathbf{i}_r(t) + \frac{d\lambda_r(t)}{dt} - \mathbf{J}\omega_r \lambda_r(t) \tag{A.2}$$

where the flux equations are :

$$\lambda_s(t) = L_{ss}\mathbf{i}_s(t) + \lambda_m(t) \tag{A.3}$$

$$\lambda_r(t) = L_{sr}\mathbf{i}_r(t) + \lambda_m(t) \tag{A.4}$$

and for the model without core loss :

$$\lambda_m(t) = L_m(\mathbf{i}_s(t) + \mathbf{i}_r(t)) \tag{A.5}$$

while for the model with core loss

$$\frac{d\lambda_m(t)}{dt} = -\frac{R_{fe}}{L_m}\lambda_m(t) + R_{fe}(\mathbf{i}_s(t) + \mathbf{i}_r(t)) \tag{A.6}$$

The relation to the notation used in Chapter 1 is :

$$L_r = L_{sr} + L_m \tag{A.7}$$

$$L_s = L_{ss} + L_m \tag{A.8}$$

$$M = L_m \tag{A.9}$$

Note that $H_{\omega_r}(s) = \mathbf{i}_s(s)/\mathbf{v}_s(s)$ is the input admittance assuming ω_r is constant.

A.3 Mechanical Model

The modification to a two-axis model does not make any difference in the mechanical equation. Therefore

$$\frac{d\omega_r}{dt} = p\left[-\frac{B}{H}\omega_r + \frac{T_{em} - T_L}{H}\right] \tag{A.10}$$

$$T_e = \frac{p}{2}\frac{M}{L_s L_r - M^2}\lambda^T \begin{bmatrix} \mathbf{0} & \mathbf{J} \\ -\mathbf{J} & \mathbf{0} \end{bmatrix} \lambda \tag{A.11}$$

where p means the number of pole pairs and

$$\lambda = \begin{bmatrix} \lambda_s \\ \lambda_r \end{bmatrix} \tag{A.12}$$

$$\mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{A.13}$$

and the notation is :

ω_r : rotor speed

B : friction coefficient

H : inertia (rotor+load)

T_L : mechanical load

T_{em} : electromagnetic torque

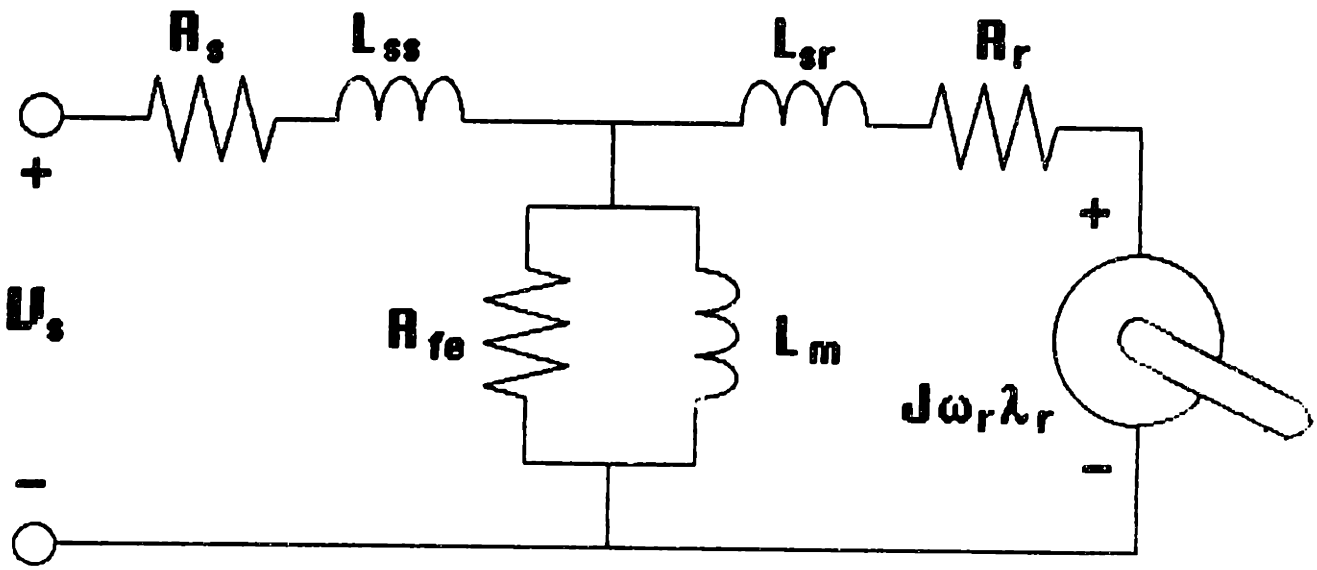


Figure A.1: Equivalent Circuit Per Axis

Appendix B

Closed-Loop Control Scheme

B.1 Introduction

In this appendix the organization of the simulated machine model is presented. First, the basic idea is presented. Second, the detailed mechanization built in $MATRIX_X$ is presented. Note that the model is described in stator-fixed coordinates.

B.2 Basic Idea

The drive scheme simulated is a voltage-fed constant V/f one, under slip frequency control, and using 3-pulse PWM (Pulse Width Modulation) in order to get sufficient low order harmonics in stator voltages and currents. This control scheme is shown in Figure 5.1.

The driving circuit is realized in the abc-frame with respect to ground. Then, it is translated to the abc-frame with respect to the neutral point. Next, the frame transformation from the abc-frame to the $\alpha\beta$ (two-axis)-frame is applied. This driving

voltages are supplied to the motor model described in the $\alpha\beta$-frame. The rotor speed is fed back to the driving circuit to close the control loop. A PI-controller is installed in the driving circuit to get good control characteristics. P-gain and I-gain for this PI-controller are 1 and 0.6 respectively. One possible way to determine the gains of a PI-controller is presented in [35]. The machine parameters used in this simulation is shown in Table 5.1.

B.3 Implementation Details

An overview of mechanization without noise is shown in Figure B.1. Individual blocks are shown in Figures B.2 - B.7. The driving voltage waveforms with in the $\alpha\beta$-frame at 262 rad/sec operation are shown in Figure B.8 for reference.

A brief explanation of some blocks is following.

SYSTEM is composed of six blocks. First one is CNTRL super-block which describes inverter control scheme. Second one is MOTOR super-block which describes motor model. Third one is SVF-V super-block which includes antialiasing filters and state variable filters for voltage measurements. Fourth one is SVF-I which includes antialiasing filters and state variable filters for current measurements. LOAD block which describes mechanical load is fifth one. LDCHG block which give the opportunity to change the load in the course of operation is sixth block.

CNTRL super-block is organized from six blocks. SPDCMD block gives the reference speed command. Second one generates the error between reference command and true speed. PI-controller is installed in PICNT block. Fourth one is LIMIT block which restricts saturation limit of controlled slip. In fifth block true speed and controlled slip are summed. The output of fifth block is fed to V/FCNT super-block which realizes constant V/F voltage-fed inverter.

V/FCNT super-block has six blocks. AMP block generates the mapping from controlled frequency to the peak-to-peak voltage of fundamental frequency with low speed compensation. In SIN-1F three phase sinusoidal waves with fundamental frequency are generated. TRW-6F generates triangle wave with 6-th order frequency. In PV-G super-block both of the outputs of SIN-1F and TRW-6F are compared and three phase voltages in the abc-frame with respect to ground are generated. Then they are transformed the voltages with respect to neutral point. Furthermore, they are transformed to the $\alpha\beta$-frame in ABCDQ.

MOTOR super-block is composed of three blocks. ELECT super-block describes electrical model of machine. Mechanical subsystem of machine is described by SPEED and EMTRQ. SPEED gives the state equatation of speed while EMTRQ generates electromagnetic torque.

ELECT super-block is composed of 5 blocks. The state variables are flux vectors which are implemented in INTGR and EXPRES. In F=>C current vector is generated. RSSTR gives the resistor values of stator and rotor. RESCHG offers the opportunity to chage those values in the course of operation.

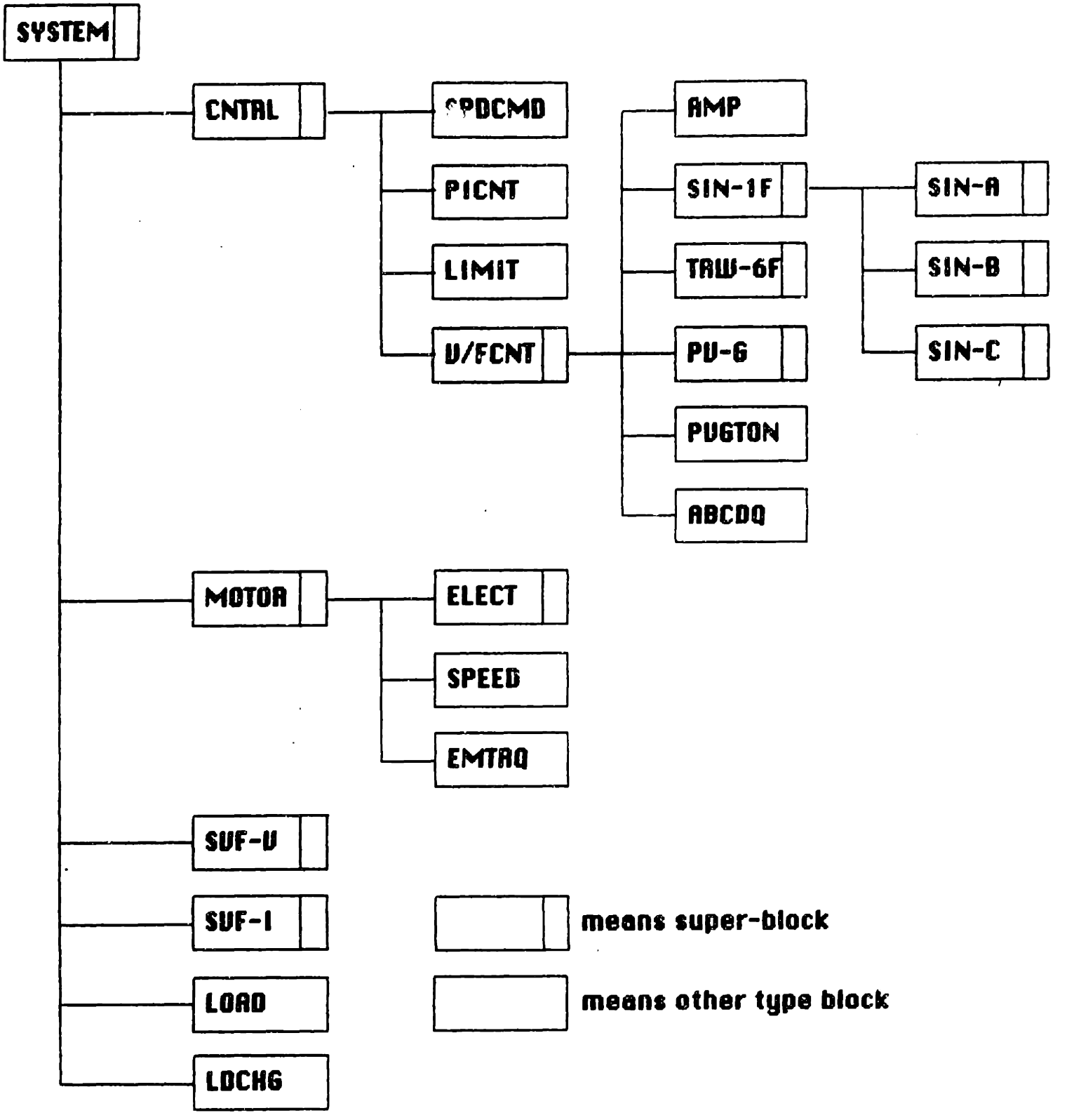


Figure B.1: Overview of Implementation

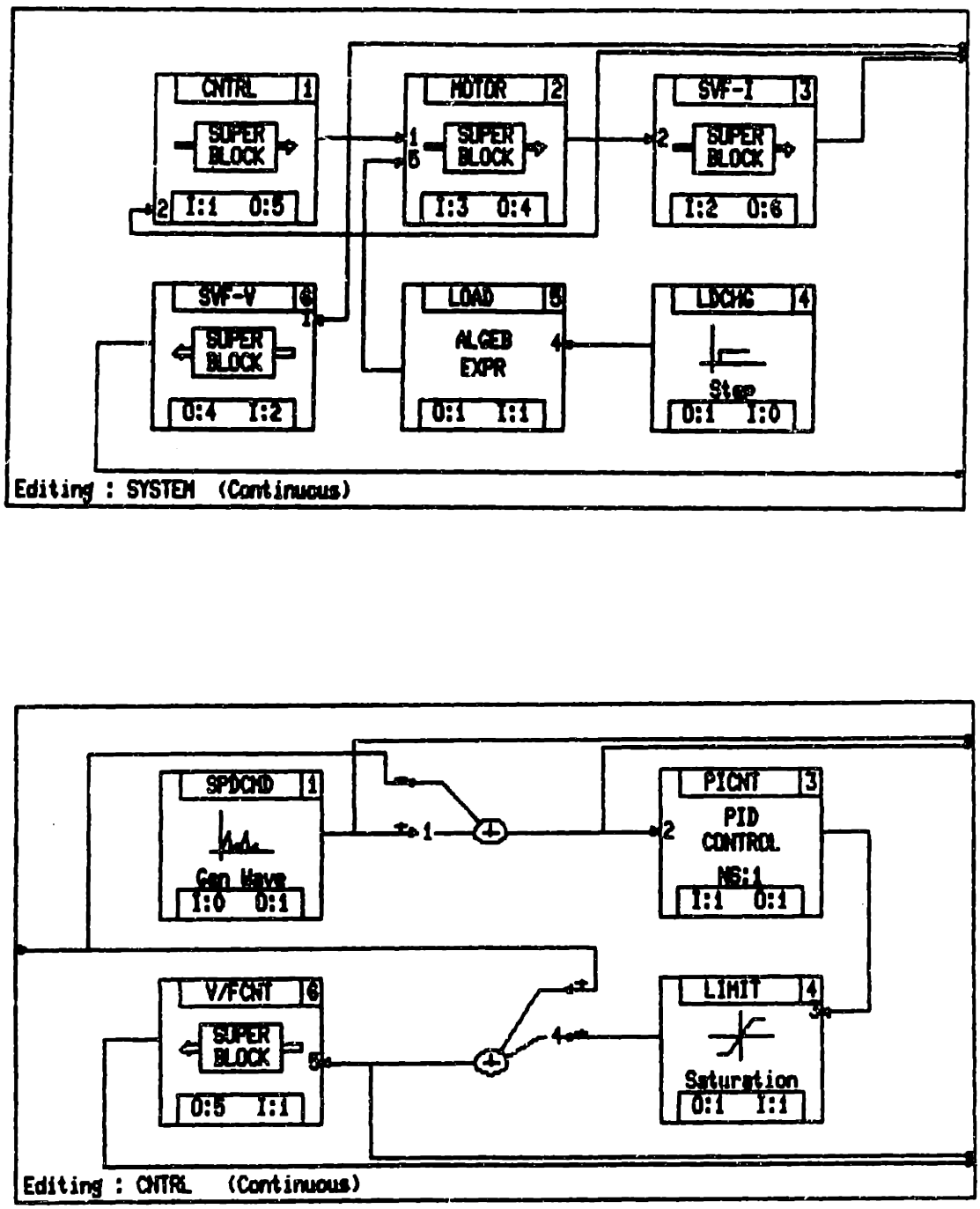
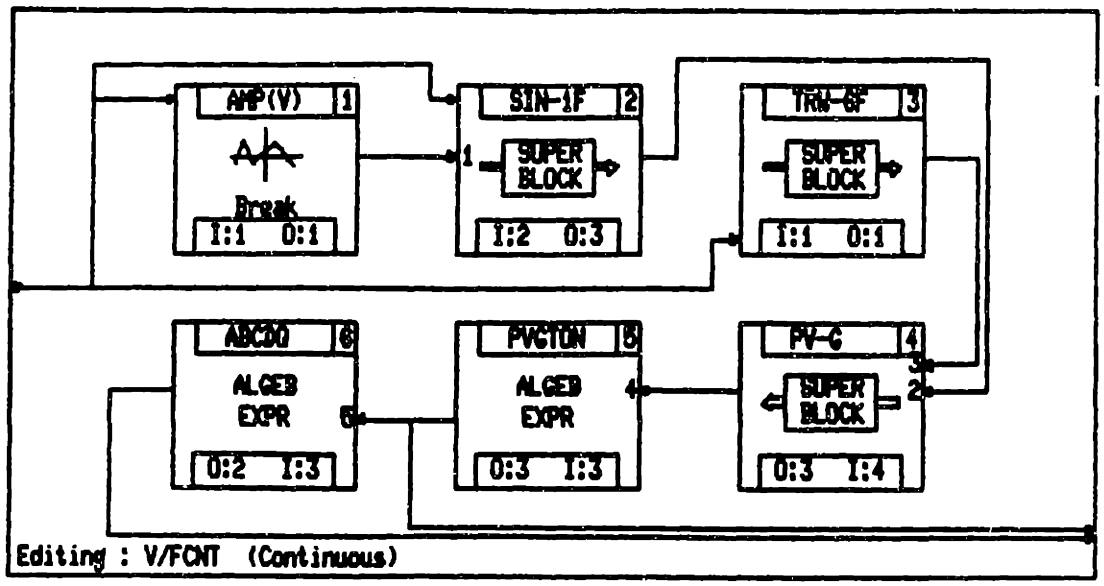


Figure B.2: Individual Block of Simulation Model-1



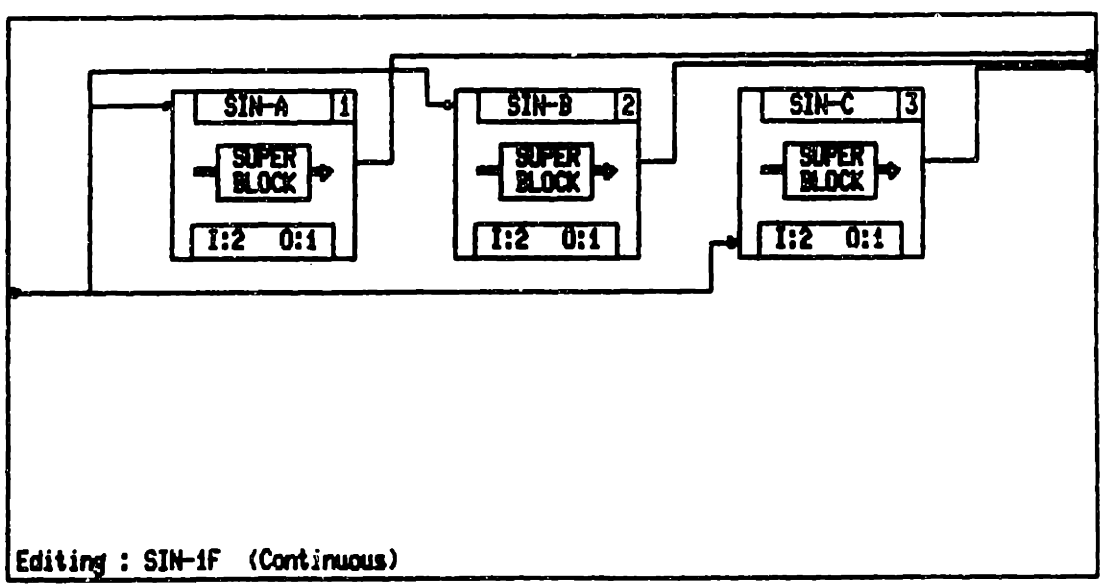
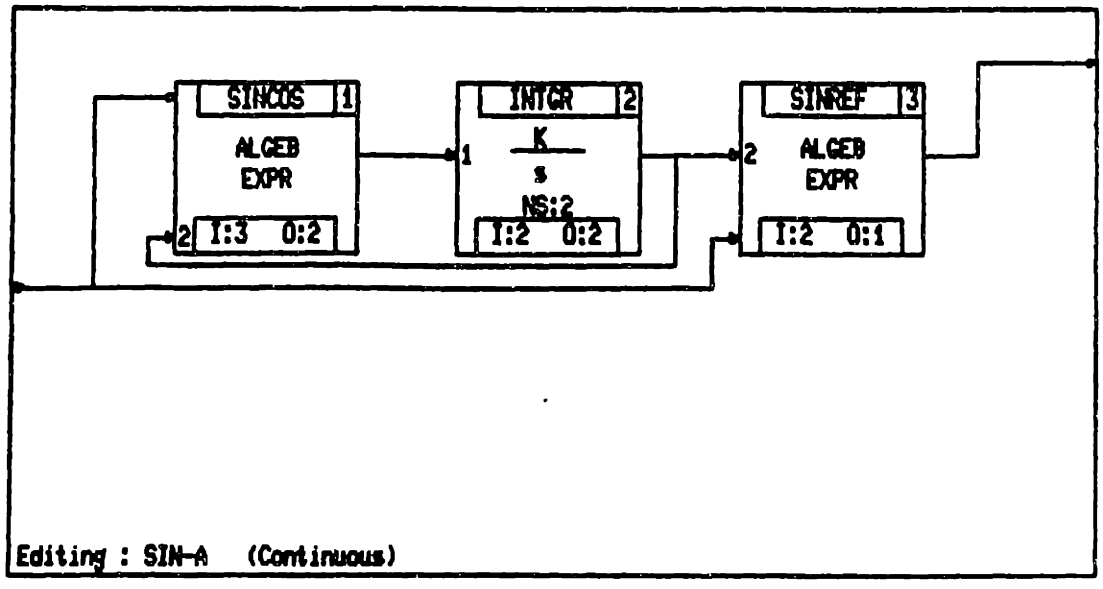


Figure B.3: Individual Block of Simulation Model-2



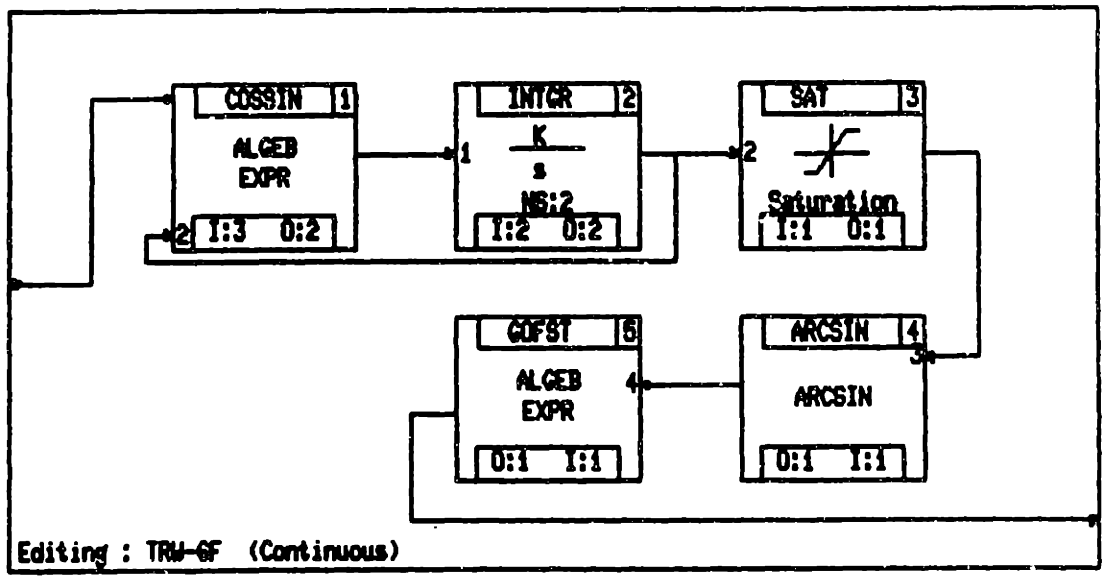
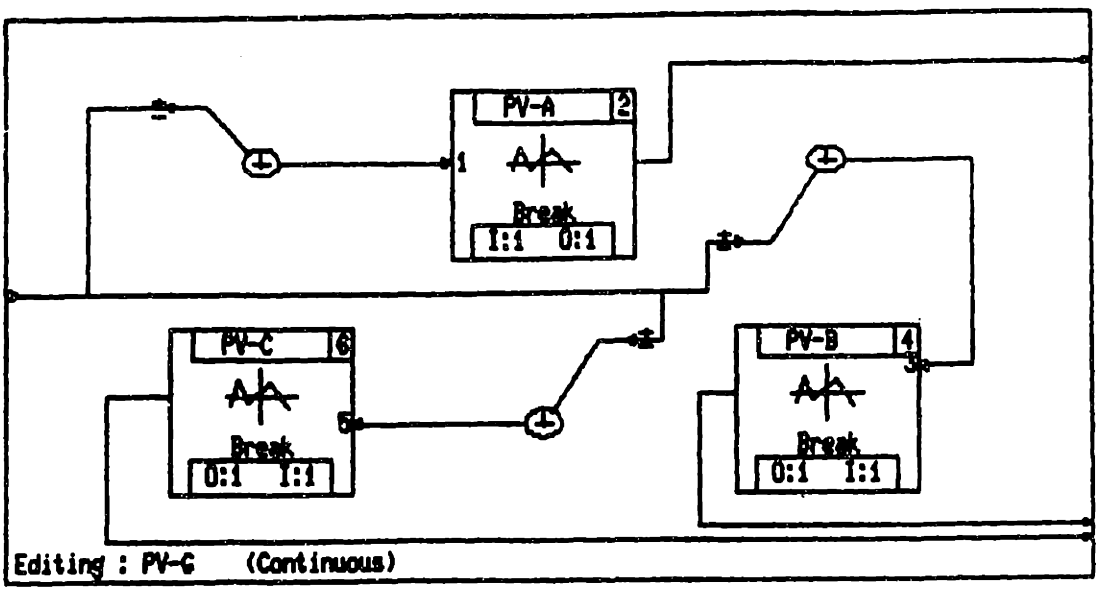


Figure B.4: Individual Block of Simulation Model-3



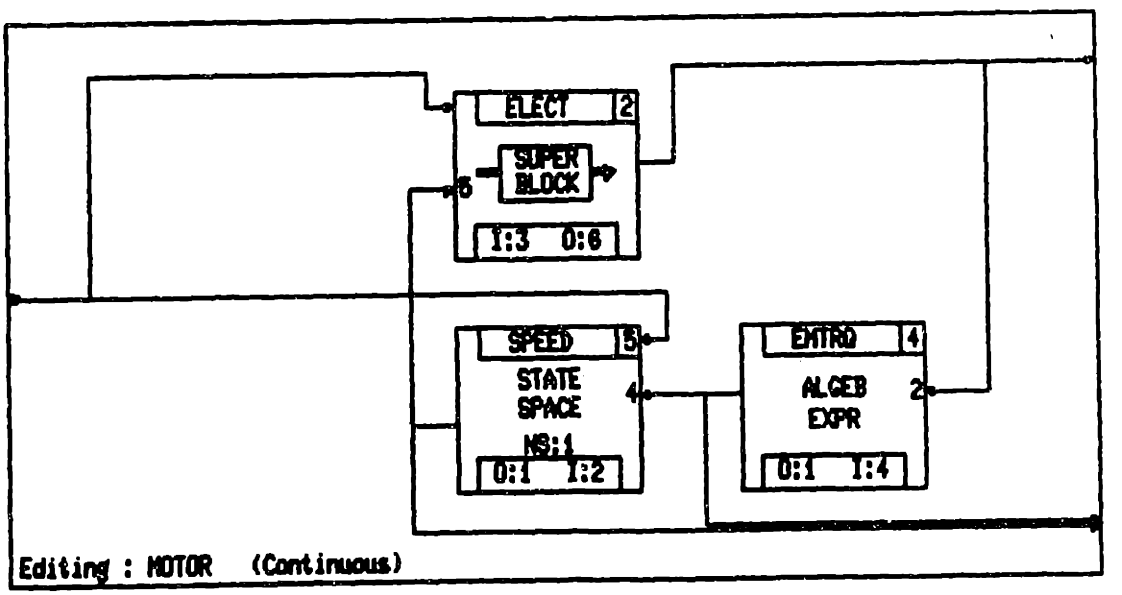
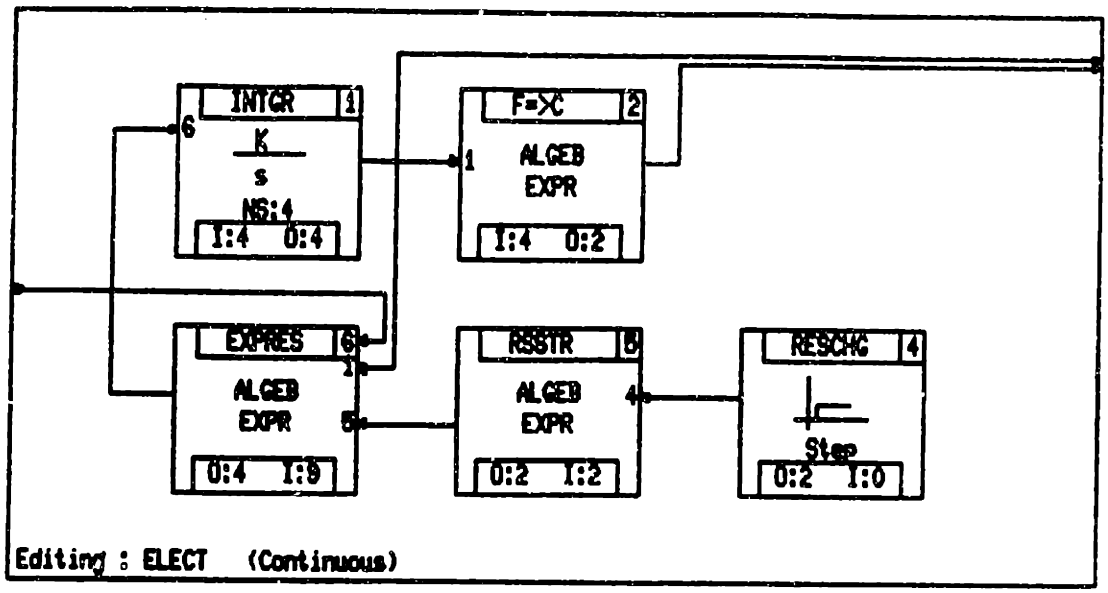


Figure B.5: Individual Block of Simulation Model-4



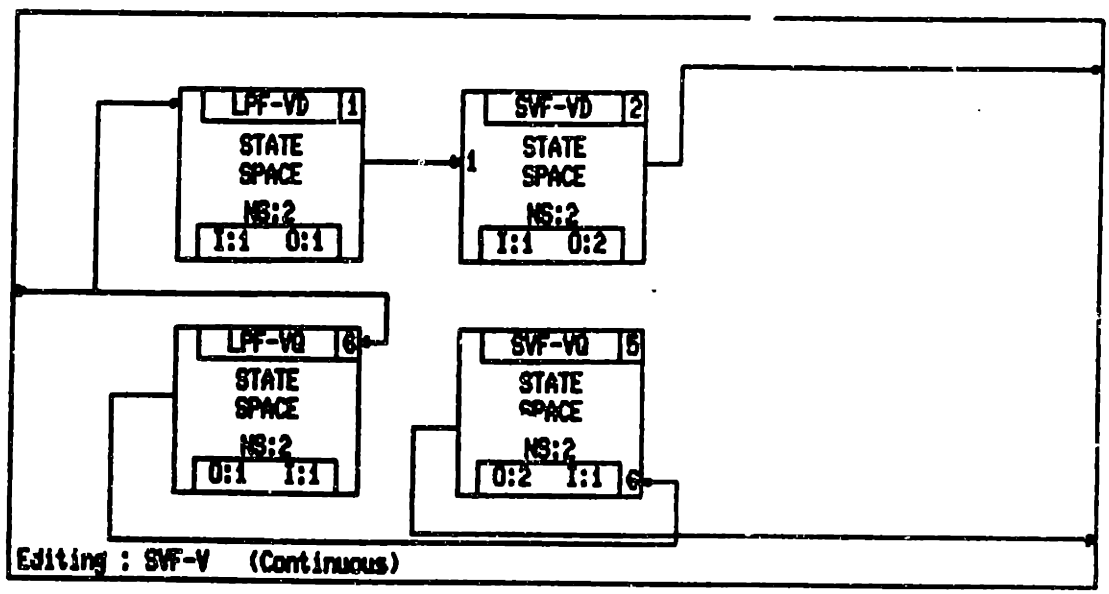


Figure B.6: Individual Block of Simulation Model-5

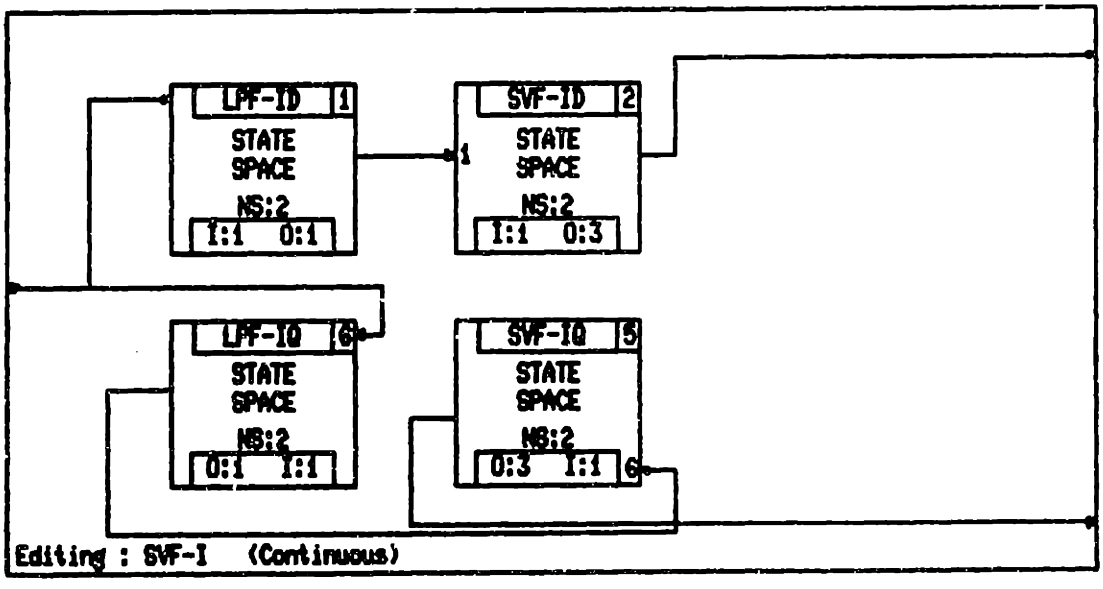


Figure B.7: Individual Block of Simulation Model-6

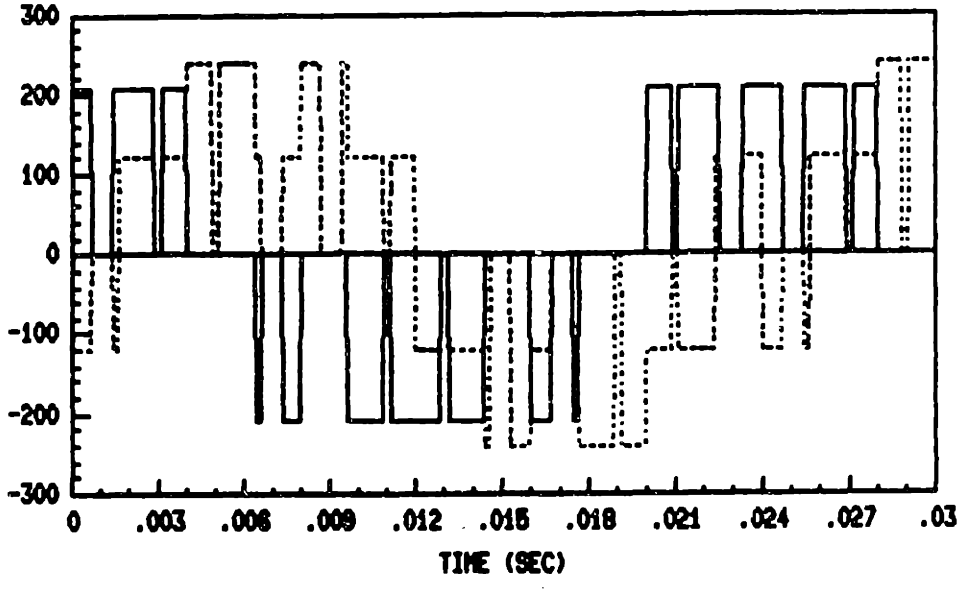


Figure B.8: Driving Voltage Waveforms in $\alpha\beta$-Frame at 262 rad/sec

Appendix C

Program for Data Conversion to ASCII Format

```
/*************************************************************/
/*   CONTOA.C            02.11.88   Dagmar Elten             */
/* This program converts the hex-data of the files           */
/* procuced by the transient recorder 'ADAM' into            */
/* data files in ASCII format                                */
/*      input :  "data.hex"                                  */
/*      output:  "data.asc"                                  */
/*************************************************************/

#include "stdio.h"
main()
{
int  n, range, tsample, channel, block, mask, creat(), open();
int  infile;
long int i;
char ccc, buffer[2], ascibuf[10];
unsigned int  intbuf;
short sss;
unsigned int count=2, fd;
double value, vrange;
FILE  *fopen(), *outfile;

outfile=fopen("data.asc","w");
infile = open("data.hex",BREAD);
```

```
if  ( infile<0)
        {
        printf("Opening of input file 'data.hex' was not possible");
        exit();
        }
printf("******* DATA - CONVERSION :  data.hex ===> data.asc *******\n");
printf("=================================================\n\n");

while ( (read(infile,buffer,count)) == count)
        {
        if ( buffer[0] == 'W' )
                {
                i = 0;
                channel = buffer[1]+1 ;
                fprintf(outfile,"\n*\n* \tCannel no. %d\n*",channel);
                printf("channel = %d\t",channel);
                }
        if ( i == 24)
                {
                tsample = buffer[0]*256 + buffer[1]; /*  us */
                printf("tsample = %d us\t",tsample);
                fprintf(outfile," \tt.sample = %d us\n*",tsample);
                }
        if ( i == 25 )
                {
                mask = buffer[0] | 0xff00;
                printf("mask = %4x\t",mask);
                }
        if (i == 9)
                {
                block = buffer[0] ;
                printf("block = %d\t",block);
                fprintf(outfile,"\tblock length = %d\n*",block);
                }
        if ( i == 30 )
```

```
                {
                range = buffer[1];
                switch(range)
                        {
                        case 21:
                                vrange=12;
                                break;
                        case 24:
                                vrange = 25;
                                break;
                        default:
                                printf("no common range!!");
                                exit();
                        }
                printf("range = %.0f V\n",vrange);
                fprintf(outfile,"\tRange of Voltage = %.0f V\n*\n",vrang
                }
        if ( i == 64 )   n = 0;
        if (i>64)
                {
                strncpy(&intbuf,buffer,2);
                intbuf = intbuf & mask;
                value = ( (double)intbuf/64000.0 - 0.5 ) * vrange;
                fprintf(outfile,"%7.3f  ",value);
                n++;
                if ( n == 8 )
                        {
                        n = 0;
                        fprintf(outfile,"\n");
                        }
                }
         i++;
         }
fclose(outfile);
}
```

Appendix D

Recursive Least Squares Estimation Program

```
//
// Recursive Least Squares Estimation Program
//
//   by Kazuaki Minami
//
//********Given Initial Value*****************
//*Lr,Ls,M,Rr,Rs,wr        ;                 *
//*last,st,set             ; set=st*integer  *
//*var                     ;                 *
//*ps0,psr0,pf0            ;                 *
//*******************************************
//
//*****************************************
//* INITIALIZE                            *
//*****************************************
clear zeta;clear lambda;clear wr;
Ls=0.0293;Lr=0.0293;M=0.0277;Rs=0.370;Rr=0.126;
sigma=Ls*Lr-M**2;delta=sigma/(Lr*Ls);Tr=Lr/Rr;
p1=Ls*delta;p2=Ls/Tr;p3=1/Tr;
wr(1)=y(1,11)*0.9;
wrma(1)=wr(1);
wrerr(1)=wrma(1)-y(1,11);
rc=Rs;
Rs=Rs*0.7; // initial value of Rs
r(1)=Rs;
zc=[p1 p2 p3]';// Constraint of zeta
p1=0.9*p1;p2=0.9*p2;p3=0.9*p3;
zetas=[p1 p2 p3]';
```

```
//
lct=st+1; // Loop Counter for sample Timing
lcs=1; // Loop Counter for Slow estimation
lcf=1; // Loop Counter for Fast estimation
lcc=1; // Loop Counter for periodic Cov. reset
//lambda(lcs)=lambda0;
ps0=10*eye(3);//
ps=ps0;        // Cov. matrix of slow estimation
ps11(1)=ps(1,1);ps22(1)=ps(2,2);ps33(lcs)=ps(3,3);
psr0=10;       //
psr(1)=psr0; // Cov. matrix of Rs estimation (scaler)
ipsr=1/psr0;  //
pf0=10;        //
pf(1)=pf0;        // Cov. matrix of fast estimation (scalar)
ipf=1/pf(1);      //
zeta(:,lcs)=zetas; // store initial zeta
//*****************************************
//* MAIN                                  *
//*****************************************
while lct+set<=last+st,... // set;Slow Estimate Time
...//*****************************************
...//* UPDATE FAST (SPEED ONLY) ESTIMATE     *
...//*****************************************
  for i=st:st:set,... // st;Sampling Timing
    lcf=lcf+1;...
    yfr=[y(lct,3) y(lct,2) y(lct,1) y(lct,8) y(lct,7)];...
    yfi=[y(lct,6) y(lct,5) y(lct,4) y(lct,10) y(lct,9)];...
    yf=[yfr;yfi]*[p1 p2+Rs p3*Rs -1 -p3]';...
    cfr=[y(lct,2) y(lct,1) y(lct,7)];...
    cfi=[y(lct,5) y(lct,4) y(lct,9)];...
    cf=[-cfi;cfr]*[p1 Rs -1]';...
    ipf=0.8*ipf+cf'*cf;... // forgetting factor=0.8


    if ipf<0,ipf=1/pf(lcf-1);... // periodic covariance reset
      elseif ipf<0.01,ipf=1/pf0;...
    end,...
    errwr=yf-cf*wr(lcf-1);...// prediction error
    pf(lcf)=1/ipf;... // covariance windup check
    rdw1(lcf-1)=norm(cf/ipf);... // richness detector
    rdw2(lcf-1)=norm(errwr);...  //
    wr(lcf)=wr(lcf-1)+cf'*errwr/ipf;...
    if wr(lcf)>wr(lcf-1)+10,wr(lcf)=wr(lcf-1)+10;end,...
    if wr(lcf)<wr(lcf-1)-10,wr(lcf)=wr(lcf-1)-10;end,...
    if lcf<4,wrma(lcf)=wr(lcf);...
      else wrma(lcf)=(wr(lcf-3)+wr(lcf-2)+wr(lcf-1)+wr(lcf))/4;...
    end;...
    wrerr(lcf)=wrma(lcf)-y(lct,11);...
...// transfer data for slow estimate
    if i=st,...
       wrt=wrma(lcf);...
       y1t=y(lct,1);y2t=y(lct,2);y3t=y(lct,3);y4t=y(lct,4);y5t=y(lct,5);..
       y6t=y(lct,6);y7t=y(lct,7);y8t=y(lct,8);y9t=y(lct,9);y10t=y(lct,10);
    end,...
  lct=lct+st;...
  end,...
```

```
...//*****************************************
...//* UPDATE SLOW (PARAMETER) ESTIMATE      *
...//*****************************************
...//
...// Rs estimation assuming parameter & speed are known
...//
    ysr=[(p1*y3t)+(p2*y2t+p1*wrt*y5t)+( -y8t)+(-p3*y7t-wrt*y9t);...
         (p1*y6t)+(p2*y5t-p1*wrt*y2t)+(-y10t)+(-p3*y9t+wrt*y7t)];...
    csr=[(-y2t)+(-p3*y1t)+(-wrt*y4t);...
         (-y5t)+(-p3*y4t)+( wrt*y1t)];...
    ipsr=0.98*ipsr+csr'*csr;... // forgetting factor=0.98
    if ipsr<0,ipsr=1/psr(lcs);... // periodic covariance reset
      elseif ipsr<0.01,ipsr=1/psr0;...
    end,...
    errrs=ysr-csr*r(lcs);...// prediction error
    psr(lcs+1)=1/ipsr;... // covariance windup check
    rdr1(lcs)=norm(csr/ipsr);... // richness detector
    rdr2(lcs)=norm(errrs);...    //
    r(lcs+1)=r(lcs)+csr'*(ysr-csr*r(lcs))/ipsr;...
    if r(lcs+1)>r(lcs)*1.05,r(lcs+1)=r(lcs)*1.05;end,...// one step chang
    if r(lcs+1)<r(lcs)*0.95,r(lcs+1)=r(lcs)*0.95;end,...// limit
    if r(lcs+1)>4*rc,r(lcs+1)=4*rc;end;...
    if r(lcs+1)<rc/4,r(lcs+1)=rc/4;end;...
    Rs=r(lcs+1);...
...//
...// parameter estimation assumig wr & Rs are known
...//
    cs=[ [y3t+wrt*y5t]  y2t  [Rs*y1t-y7t] ; ...
         [y6t-wrt*y2t]  y5t  [Rs*y4t-y9t] ];...
    ys=[ y8t-Rs*y2t+wrt*[y9t-Rs*y4t];y10t-Rs*y5t-wrt*[y7t-Rs*y1t] ];...
    zetad=[p1 p2 p3]';...
    s=eye(2)+cs*ps*cs';invs=inv(s);...
    ks=ps*cs'*invs;...
    e(:,lcs)=ys-cs*zetad;...// prediction error
    delta=ks*e(:,lcs);...
...//
    rds1(lcs)=norm(ps*cs');...  // richness detector
    rds2(lcs)=norm(e(:,lcs));...//
```

```
...// one step parameter chage limit
    if delta(1)>p1/20,delta(1)=p1/20;end,...
    if delta(1)<-p1/20,delta(1)=-p1/20;end,...
    if delta(2)>p2/20,delta(2)=p2/20;end,...
    if delta(2)<-p2/20,delta(2)=-p2/20;end,...
    if delta(3)>p3/20,delta(3)=p3/20;end,...
    if delta(3)<-p3/20,delta(3)=-p3/20;end,...
...//
    zetad=zetad+delta;...
    lambda(lcs)=1-e(:,lcs)'*invs*e(:,lcs)/var;...
    if lambda(lcs)<0.95,lambda(lcs)=0.95;end,...
    if lambda(lcs)>1,lambda(lcs)=1;end,...
    if ps(1,1)+ps(2,2)+ps(3,3)>10,lambda(lcs)=1;end,...
    psold=ps;... // save previous ps
    ps=(ps-ps*cs'*invs*cs*ps)/lambda(lcs);...// Var. Forgetting Fact.
    if ps(1,1)+ps(2,2)+ps(3,3)>100,ps=ps0;end,...// Periodi Cov. Reset
    if ps(1,1)<0,ps=psold;...
       elseif ps(2,2)<0,ps=psold;...
       elseif ps(3,3)<0,ps=psold;...
    end,...
    ps11(lcs+1)=ps(1,1);... // covariance windup check
    ps22(lcs+1)=ps(2,2);... // covariance windup check
    ps33(lcs+1)=ps(3,3);... // covariance windup check
...//
...// Project Estimate Parameter Space & Estimate Regulation
...//
    ztemp=[zetad(1) zetad(2) zetad(3)]';...
    for j=1:3,...
      if ztemp(j)<zc(j)/4,ztemp(j)=zc(j)/4;end,...
      if ztemp(j)>zc(j)*4,ztemp(j)=zc(j)*4;end,...
    end,...
    p1=ztemp(1);p2=ztemp(2);p3=ztemp(3);...
    zetas=[p1 p2 p3]';...
  lcs=lcs+1;...
  zeta(:,lcs)=zetas;...
end
return
```

Appendix E

Effect of Parameter Number Reduction in Estimation

E.1 Introduction

This appendix is written to show the difference between parameter estimation for on-line rotor speed estimation and parameter estimation for off-line failure detection.

In parameter estimation for on-line rotor speed estimation, there is no speed sensor in the measurement system. On the other hand, it is possible to use a speed sensor to reduce the number of parameters in parameter estimation for off-line failure detection.

In failure detection of induction machines, stator resistance and rotor speed are often assumed as known parameters. The assumption of known stator resistance might be possible only for short time steady state operation.

The comparison below is done using simulated data with the parameters obtained from the Blocked Rotor Test and No Load Test.

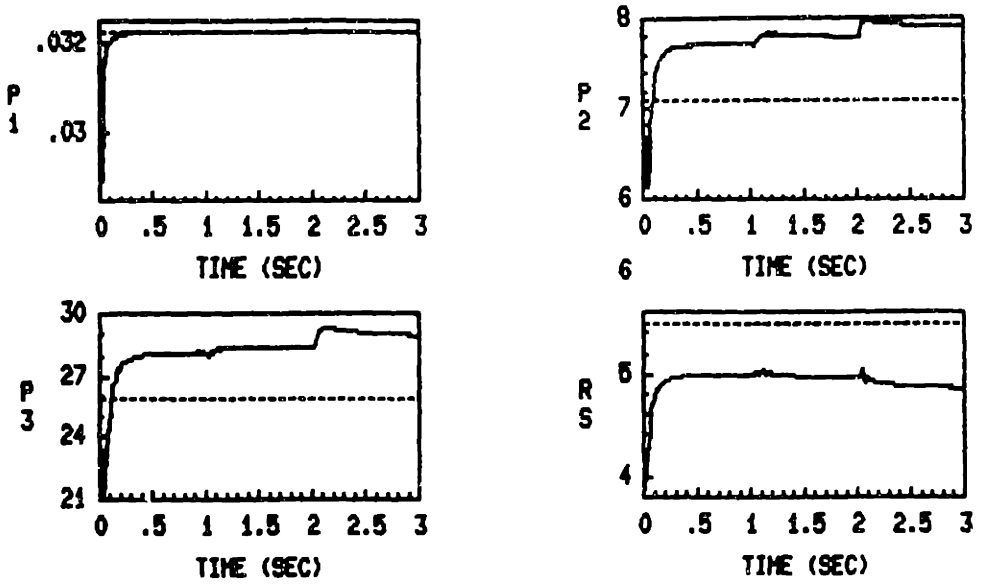


Figure E.1: Estimated Parameters with No Assumption

E.2 No Assumption for Stator Resistor and Rotor Speed

The estimation results with no assumption regarding stator resistance and rotor speed is shown in Figure E.1. The results show that the estimated parameters converge to wrong (biased) values, though the estimated speed shows good tracking characteristics, see Figure E.2.

E.3 Stator Resistor as a Known Parameter

The estimation results with known stator resistance are shown in Figure E.3. The results show that the estimated parameters do converge to true values, though the convergence speed is not so fast.

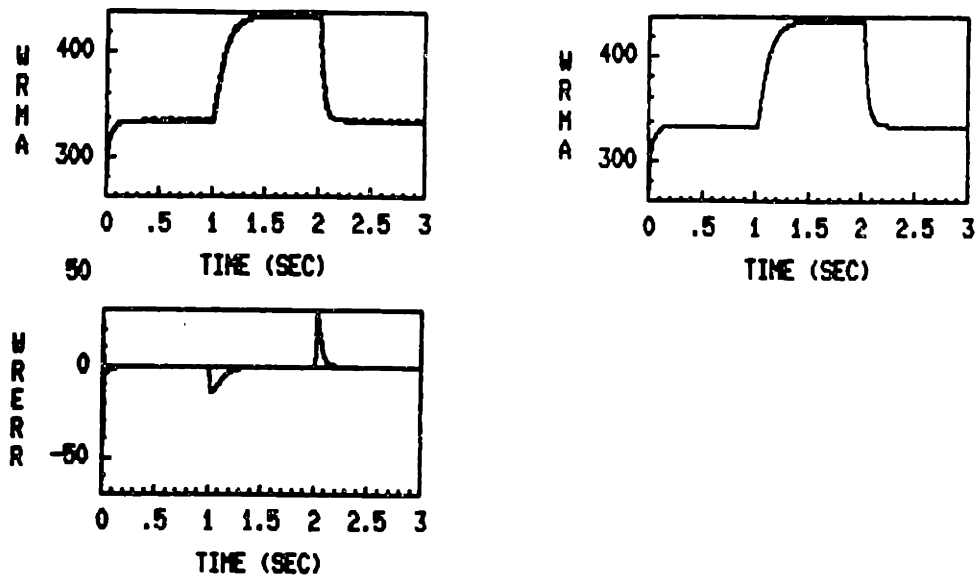


Figure E.2: Estimated Speed in rad/sec with No Assumption

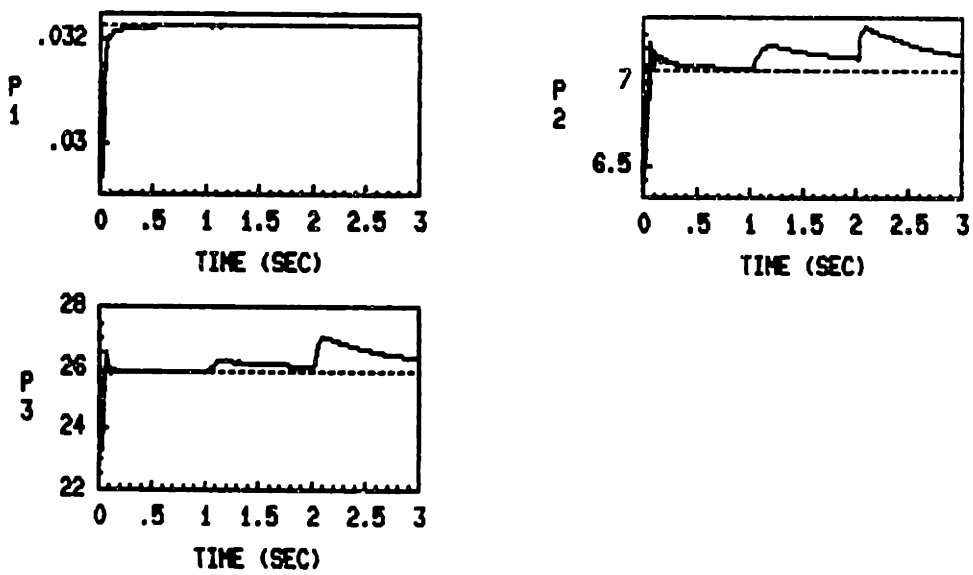


Figure E.3: Estimated Parameters with Known Stator Resistor

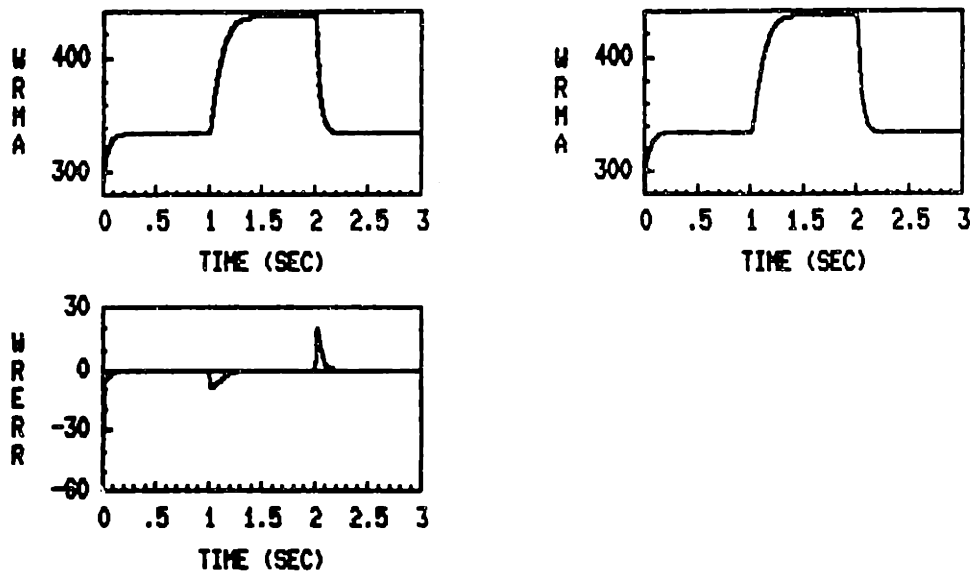


Figure E.4: Estimated Speed in rad/sec with Known Stator Resistor

E.4 Stator Resistor and Rotor Speed as Known Parameters

The estimation results with known stator resistance and rotor speed is shown in Figure E.3. The results show that estimated parameters do converge to true values very quickly. According to these results, failure detection setting should give more stable estimation results than in on-line rotor speed estimation. A possible reason for these results is the reduction of the number of parameters, especially the elimination of the most sensitive parameter (stator resistance).

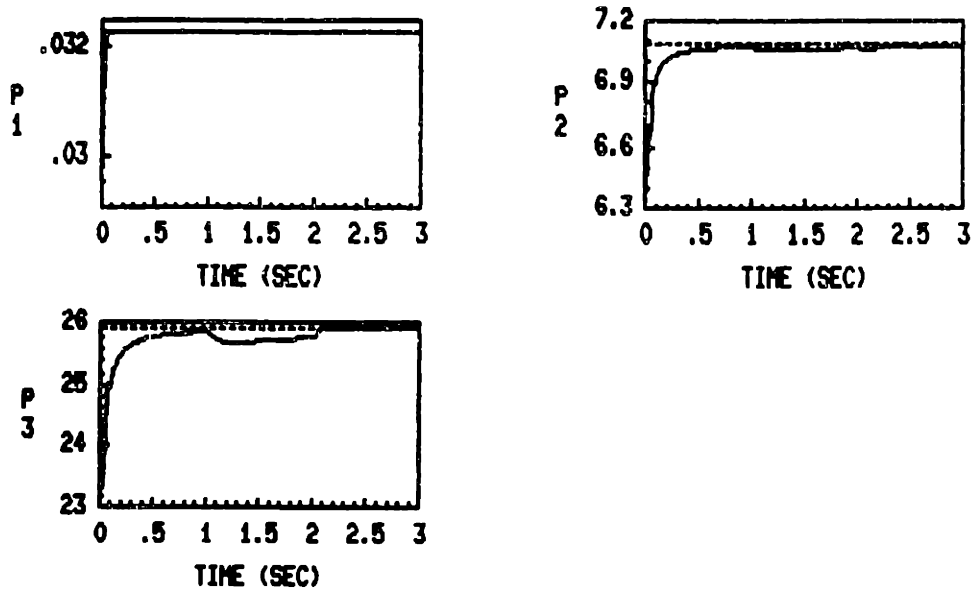


Figure E.5: Estimated Parameters with Known Stator Resistor and Rotor Speed

E.5 Comments

The idea that the fewer the number of parameters, the less difficult the estimation, is illustrated very clearly here. Parameter estimation for off-line failure detection under constant speed operation is much easier than parameter estimation for on-line rotor speed estimation.

Appendix F

Present Status of Experiment at MIT

F.1 Introduction

In order to test the estimator at MIT, a test bench using Industrial Drive's ASC-3 induction machine drive system has been designed. The system uses a voltage-fed Pulse-Width-Modulation (PWM) . The test bench was designed to provide outputs of the state variable filters as inputs to the estimation algorithm. The rotor speed is also measured by a tachometer, to compare with its estimated value.

The data for off-line experimentation is acquired using the data acquisition system DAS-20 by MetraByte Co. which is installed on Compaq-386 computer. The acquired data is downloaded to a VAX11/750 computer where it is processed using the $MATRIX_X$ software package on which simulations and estimations were carried out. Figure F.1 shows the block diagram of the experimental system.

F.2 Drive System

The induction machine drive system (model ASC-3) donated by Industrial Drives Inc. includes a three phase rectifier, an inverter, a controller board, and a 3.1 horsepower induction machine. As supplied, the drive system was equipped with a tachometer to provide speed measurement, and Hall effect sensors with required buffering circuitry to provide phase current measurements. A resistively loaded DC generator was used to provide a mechanical load to the machine.

F.3 Signal Processing Board

The overview of the signal processing board is shown in Figure F.2. The obtained measurements are filtered phase currents, and their first and second derivatives, as well as filtered line voltages and their first derivatives. Every measurement is scaled to $\pm 10V$ in order to maximize the resolution.

Since voltage signals are not provided by the ASC-3 system, two isolation amplifiers are used to get line voltages. These isolation amplifiers provide the isolation from the high voltage output of inverter and scale the 300 V signals to voltage levels that can be handled by linear IC's. Figure F.3 shows the detailed circuit using Analog Devices' AD210.

The antialiasing filter circuit is shown in Figure F.4. This filter was designed using a second order Bessel type low pass filter with cutoff frequency of 500 Hz. The bandwidth was chosen at 500 Hz because the maximum frequency of the fundamental components of currents and voltages is 150 Hz (at 4000 rpm). A standard 741 was used to implement this filter.

The state variable filter circuit is shown in Figure F.5. This filter was also designed

using a second order Bessel type low pass filter with cutoff frequency of 500 Hz. Burr-Brown's universal filter IC (UAF41) was used to implement this filter in a compact way.

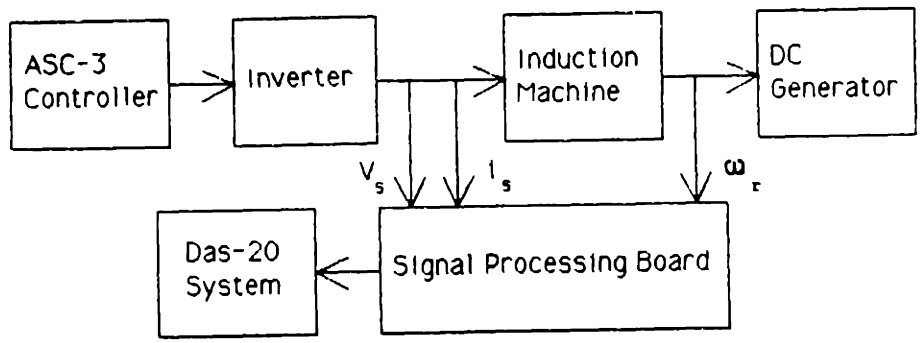


Figure F.1: Block Diagram of the Experimental System at MIT

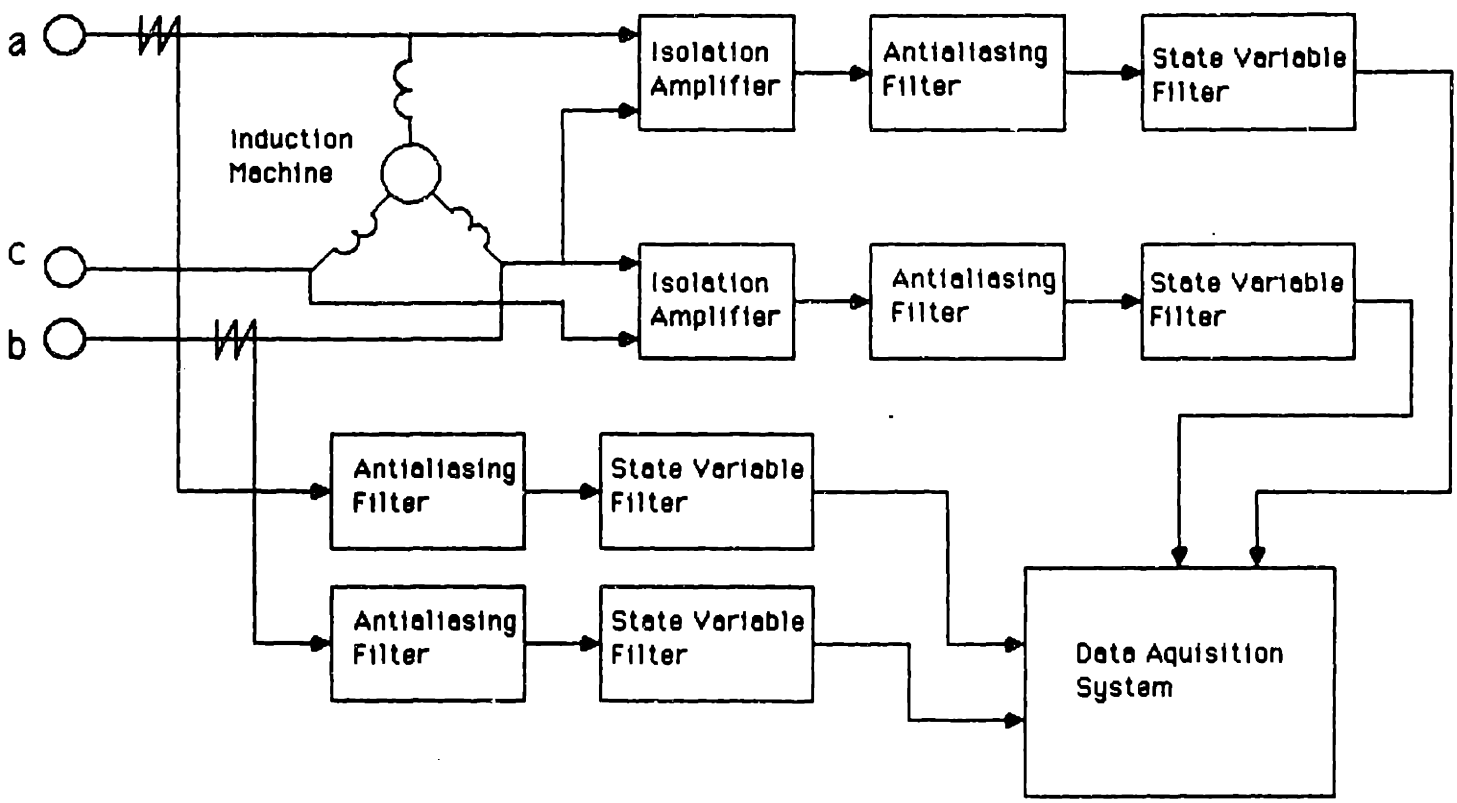


Figure F.2: Block Diagram of Signal Processing Board

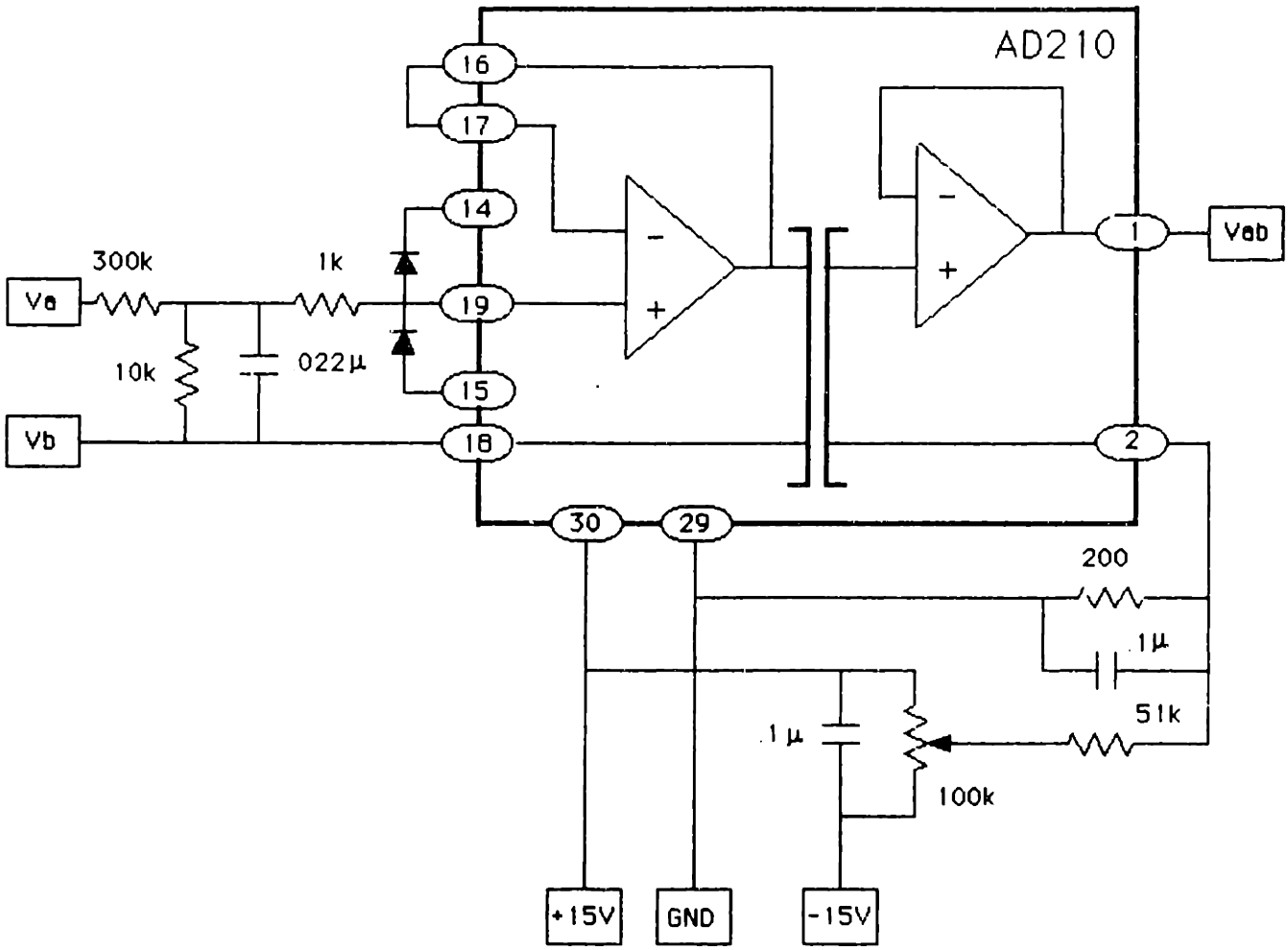


Figure F.3: Isolation Amplifier Circuit

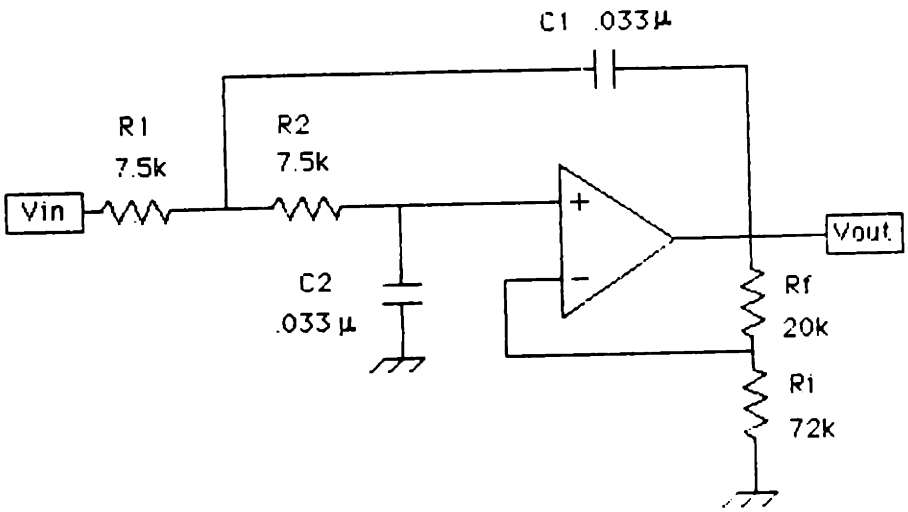


Figure F.4: Antialiasing Filter Circuit

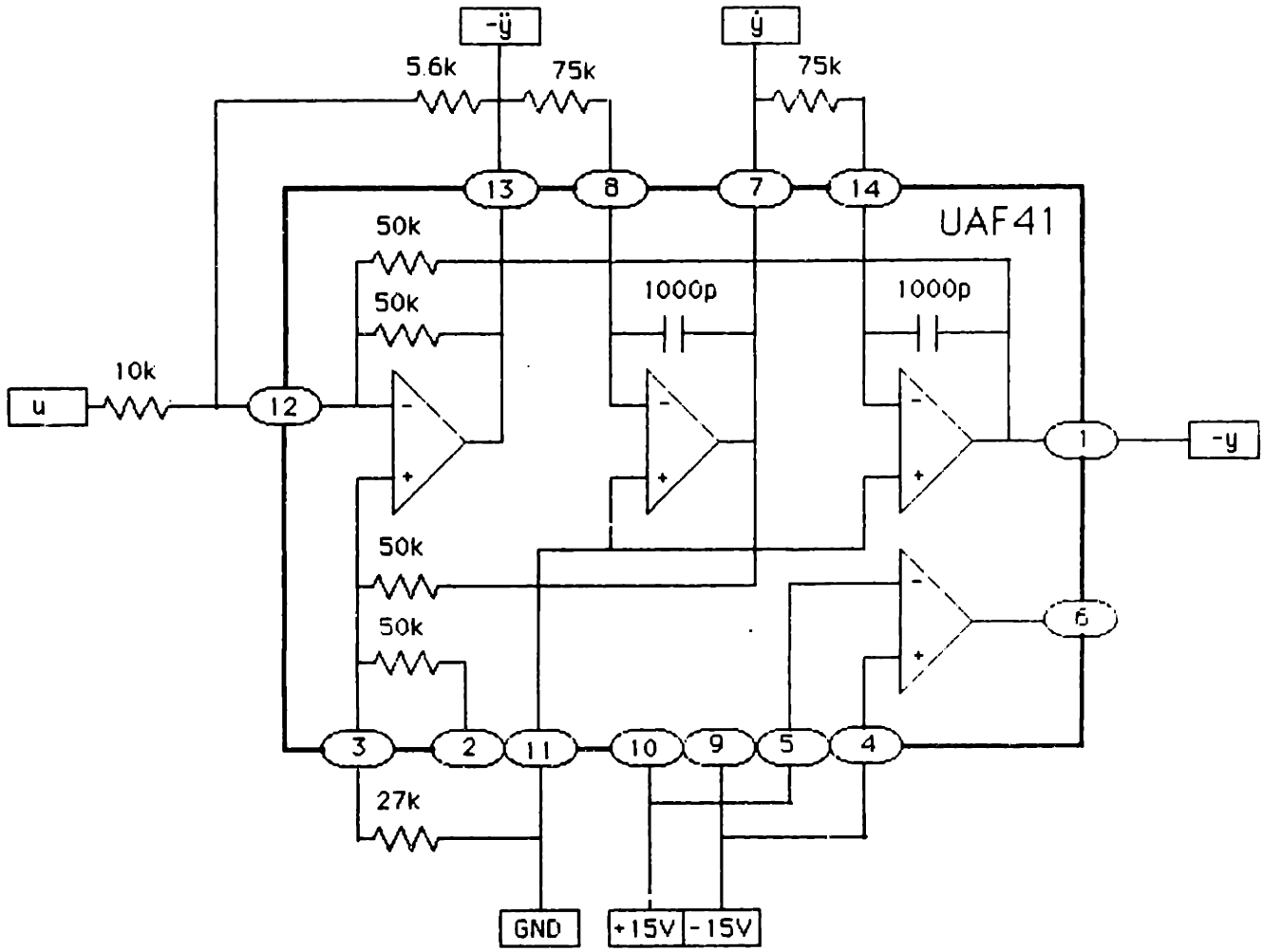


Figure F.5: State Variable Filter Circuit

F.4 Data Acquisition System

For data acquisition MetraByte's DAS-20 data acquisition system was used. MetraByte's model DAS-20 is a multifunction high speed Analog/Digital I/O expansion board for IBM-PC compatible machines that can be used for data acquisition and signal analysis. It provides 16 analog input channels but only 11 channels are required for our experiment. The input voltage range for each hannel is $\pm 10V$. Using SAMPLE.BAS [28] program written in BASIC on the Compaq-386, 11 channel data can be sampled up to 2 kHz and stored up to 1000 samples. See [9] for further reference on different features of the DAS-20 system.

F.5 Scaling and Frame Transformation

After the data has been collected, it should be rescaled to its original value because the data is scaled by the signal processing board. Then it is transformed from the abc-frame to the $\alpha\beta$-frame since the estimator program deals with $\alpha\beta$-frame data. The transformation matrix from phase currents in the abc-frame to phase currents in the $\alpha\beta$-frame is :

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} -\sqrt{\frac{3}{2}} & 0 \\ \sqrt{\frac{1}{2}} & \sqrt{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \end{bmatrix} \tag{F.1}$$

The transformation matrix from line voltages in the abc-frame to phase voltages in the $\alpha\beta$-frame is :

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} -\sqrt{\frac{2}{3}} & -\sqrt{\frac{1}{6}} \\ 0 & \sqrt{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} v_{ab} \\ v_{bc} \end{bmatrix} \tag{F.2}$$

F.6 Spectrum of Current and Voltage

In order to check the harmonics of the signals, a spectrum analysis of phase current and line voltage is done. Figure F.6 shows the power spectrum of the line voltage. Figure F.7 shows the power spectrum of the phase current acquired by a current probe. Figure F.8 shows the power spectrum of the phase current at the ASC-3 phase current output. Since both Figure F.7 and Figure F.8 are quite similar, the phase current output of ASC-3 can be used for our measurement.

As can be seen very clearly, the waveforms for 60 Hz operation have much more harmonics than 50 Hz operation. This means that the richness of signals at steady state highly depends on the operating point. Therefore the performance of the estimator may change dramatically according to the operating point. For transient and other load conditions, the richness of the signals should be different, and therefore further analysis will be required.

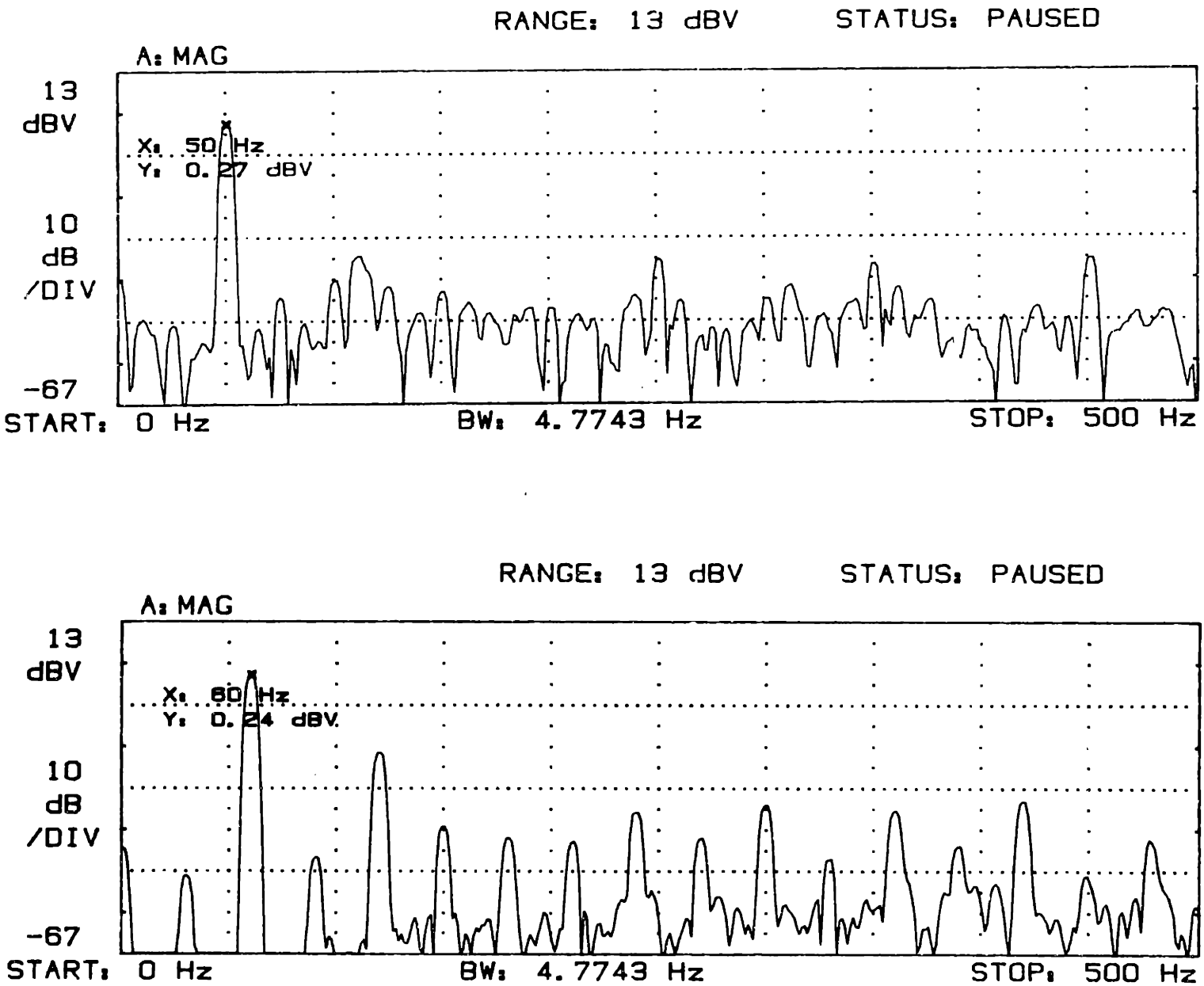
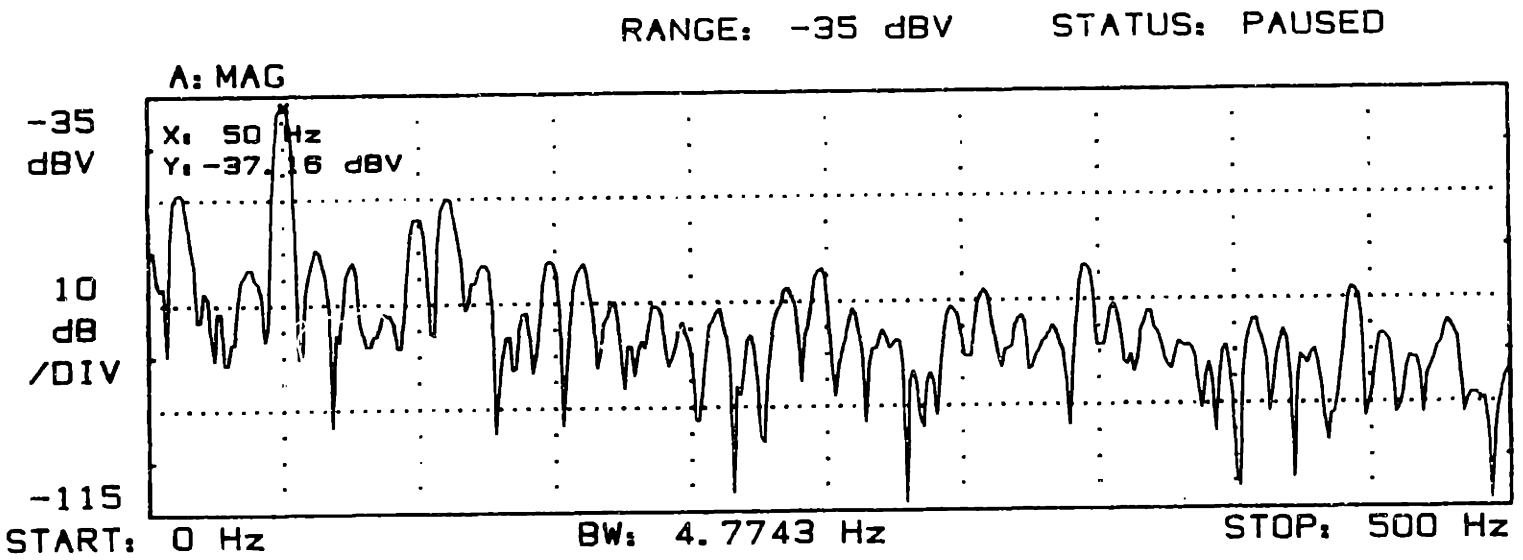


Figure F.6: Power Spectrum of Line Voltage : (upper) Electrical Frequency=50 Hz, (lower) Electrical Frequency=60 Hz



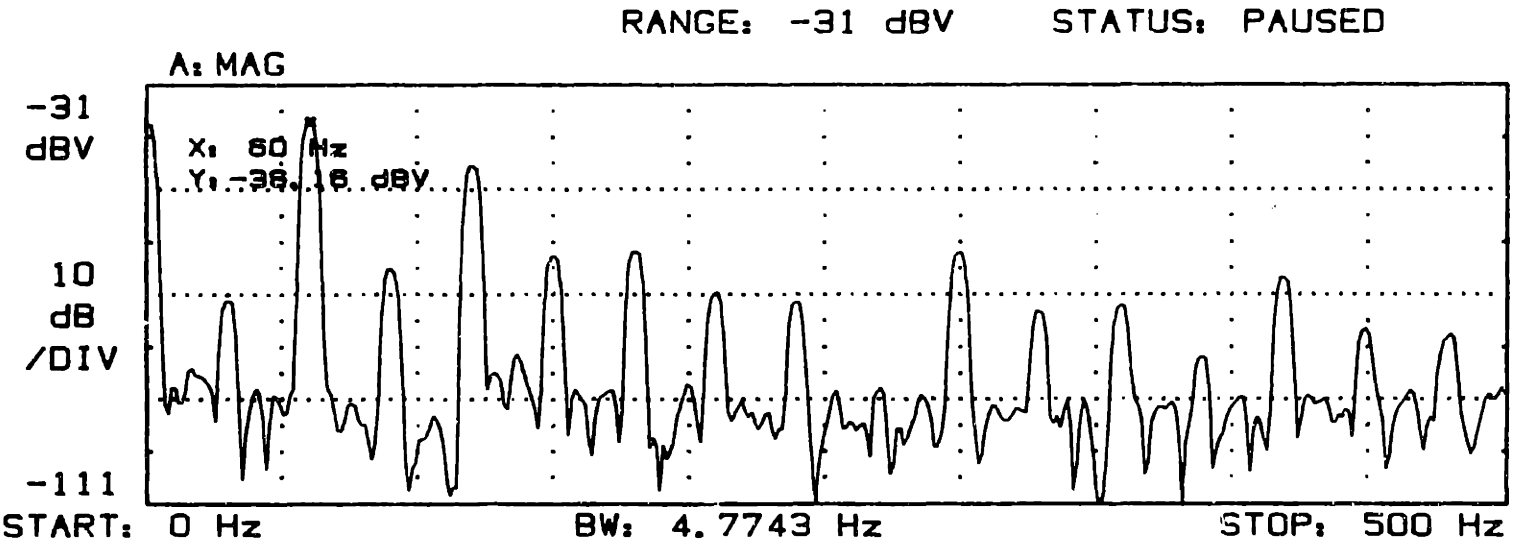
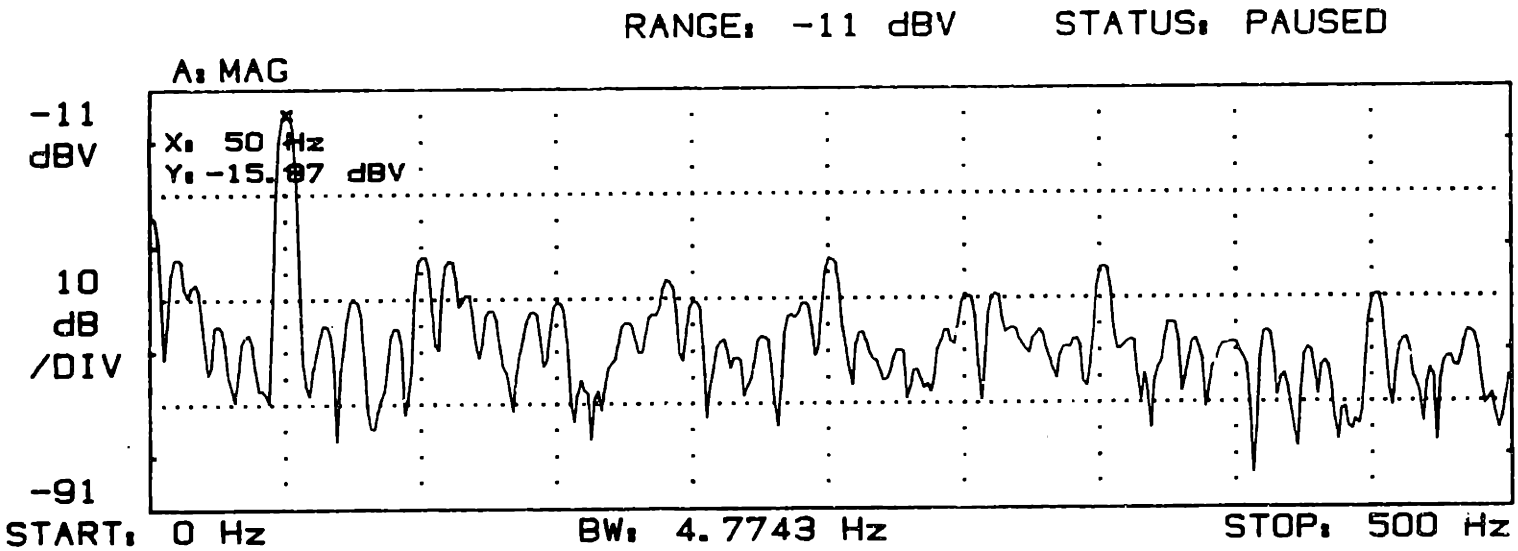


Figure F.7: Power Spectrum of Phase Current Measured by Current Probe : (upper) Electrical Frequency=50 Hz, (lower) Electrical Frequency=60 Hz



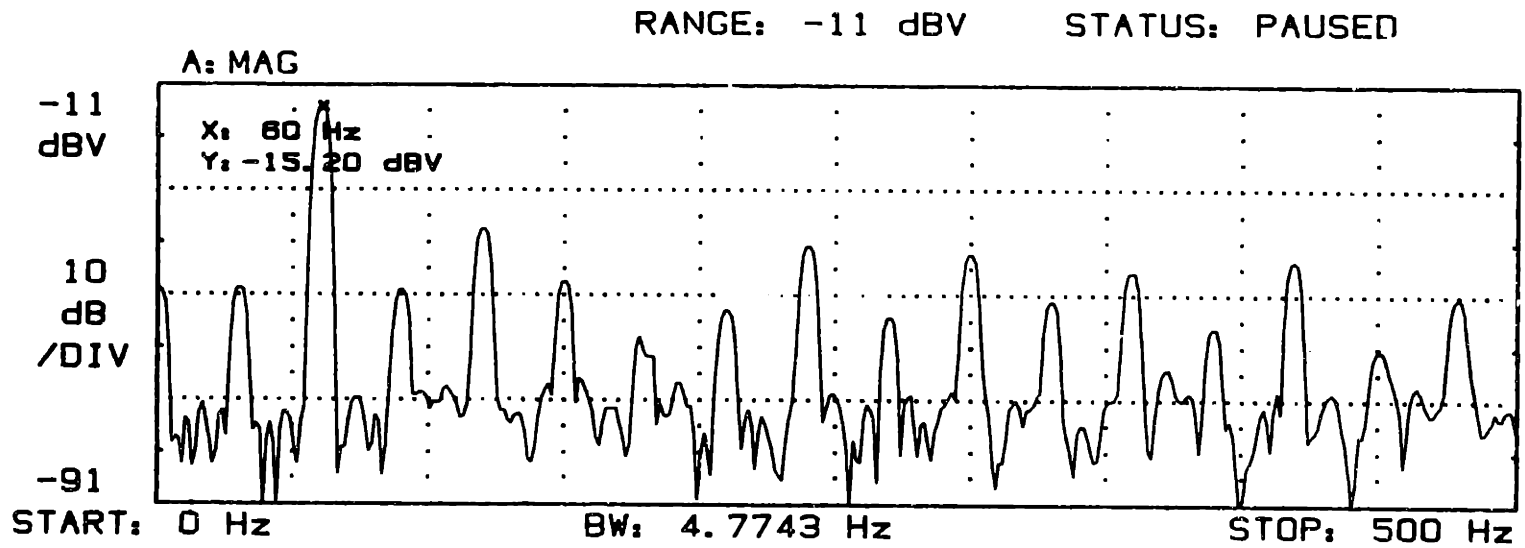


Figure F.8: Power Spectrum of Phase Current Measured by ASC-3 Phase Current Output : (upper) Electrical Frequency=50 Hz, (lower) Electrical Frequency=60 Hz

F.7 Suggestion for Future Work

Since the experiment at MIT has just started, there are plenty of things to do. First of all, the isolation amplifier's characteristics should be checked using the spectrum analyzer. The power spectrum of the output of the isolation amplifier should be compared with that of the raw line voltage below cutoff frequency, because the isolation amplifier uses a certain excitation frequency to transfer the signal from its input to output. If this frequency is not high enough, the signal is distorted badly. This can cause lack of richness of the processed signal.

The next thing to be considered is implementing a scaling circuit at every stage of the state variable filter. As can be realized easily, there is large difference in the magnitudes of signals among filter outputs. This makes certain outputs sensitive to noise. Moreover the order of output magnitudes is reversed from the simulated (digital) state variable filter outputs in $MATRIX_X$. The reason seems to be that in an actual circuit there are limitations such as through rate, input voltage range, output voltage range, etc. These limitations do not hold in $MATRIX_X$. If scaling does not make things better, it is reasonable to implement a digital state variable filter instead of an analog filter, though it may require another microprocessor and much higher sampling rate of measurements [14].

Another thing to try is no-load operation. This may generate richer signals. Also it may be useful to increase the order of antialiasing filter, to reduce unnecessary high frequency signal.

Once good estimation results are obtained, the next step is to close the loop, i.e. to install an adaptive controller in real time. I expect the continuing research will be successfully done in a couple of years.

Bibliography

[1] K.J. Åstrom, "Theory and Application of Adaptive Control - A Survey", IFAC, 1983.

[2] K.J. Åstrom and B.J. Wittenmark, *Computer Controlled Systems - Theory and Design*, Prentice-Hall, 1984.

[3] K.J. Åstrom and B.J. Wittenmark, *Adaptive Control*, Addison-Wesley, 1989.

[4] I. Boldea and S.A. Nasar, *Electric Machine Dynamics*, Macmillan Publishing Company, 1986.

[5] B.K. Bose, *Power Electronics and AC Drives*, Prentice-Hall, 1986.

[6] A. Brickwedde, "Microprocessor-Based Adaptive Speed and Position Control for Electrical Drives", *IEEE Transaction on Industry Applications*, Volume IA-21, Series 5, September/October 1985.

[7] A. Consoli, L. Fortuna, and A. Gallo, "Microcomputer System for Estimating the Parameters of an Induction Motor Model", IFAC Symposium on Control in Power Electronics, Lausanne, Switzerland, 1983.

[8] A.O. Cordero and D.Q. Mayne, "Deterministic Convergence of a Self-Tuning Regulator with Variable Forgetting Factor", *IEE Proceedings*, Volume 128, January 1981.

[9] *DAS-20 Manual*, MetraByte Corporation.

[10] S. Dasgupta, "Adaptive Identification of Systems with Polynomial Parameterization", *IEEE Transactions on Circuits and Systems*, Volume 35, May 1988.

[11] B. de Fornel, J.M. Farines, and J.C. Hapiot, "Numerical Estimation of the Speed of an Asynchronous Machine Supplied by a Static Converter", IEEE-IAS Annual Meeting, 1979.

[12] B. de Fornel, J. Faucher, and A. Sague, "A Method for Estimating the Flux and Slip of a Voltage Supplied Asynchronous Machine", IEEE Industrial Electronics Conference (IECON), 1986.

[13] B. Egardt, *Stability of Adaptive Controller*, Springer-Verlag, 1979.

[14] D. Elten, "Parameter Identification Methods for Induction Machines", *VDI Berichte*, Nr. 644, 1987.

[15] P. Eykhoff, *System Identification - Parameter and State Estimation*, JohnWiley & Sons, 1974.

[16] A.E. Fizgerald, C. Kingsley, and S.D. Umans, *Electric Machinery*, McGraw Hill, 1983.

[17] T.R. Fortescue, L.S. Kershenbaum, and B.E. Ydstie, "Implementation of a Self-Tuning Regulators with Variable Forgetting Factors", *Automatica*, Volume 17, Series 6, 1981.

[18] G. Geiger, "On-Line Fault Detection and Localization in Electrical DC-Drives Based on Process Parameter Estimation and Statistical Decision Methods", *Proceedings of the IFAC Symposium on Identification and System Parameter Estimation*, 1983.

[19] G.C. Goodwin, H. Elliott, and E.K. Tech, "Deterministic Convergence of a Self-Tuning Regulator with Covariance Resetting", *IEE Proceedings*, Volume 130, Part D, Series 1, January 1983.

[20] F. Hillenbrand, "A Method for Determining the Speed and Rotor Flux of the Flux of the Asynchronous Machine by Measuring the Terminal Quantities Only", *Proceedings of IFAC Symposium on Control in Power Electronics and Electrical Drives*, 1984.

[21] R. Joetten and G. Maeder, "Control Methods for Good Dynamic Performance Induction Motor Drives Based on Currents and Voltages as Measured Quantities", *IEEE Transactions on Industry Applications*, Volume IA-19(3), 1983.

[22] C.R. Johnson Jr., *Lectures on Adaptive Parameter Estimation*, Prentice-Hall, 1988.

[23] P.C. Krause, *Analysis of Electrical Machinery*, McGraw Hill, 1986.

[24] W. Leonhard, *Control of Electrical Drives*, Splinger-Verlag, 1985.

[25] X.Z. Liu, G.C. Verghese, J.H. Lang, and M.K. Onder, "Extending the Blondel-Park Transformation to Generalized Electrical Machines : Necessary and Sufficient Conditions", *Proceedings of the 12th IMACS World Congress*, 1988, also to appear in *IEEE Transactions of Circuits and Systems.*

[26] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*, MIT Press, 1983.

[27] I.J. Perez-Arriaga, G.C. Verghese, and F.C. Schweppe, "Selective Modal Analysis with Applications to Electrical Power Systems, Part 1 and Part 2", *IEEE Transactions on Power Apparatus and Systems*, PAS-101, 1982.

[28] L. Roehrs, *Microprocessor Real-Time Induction Machine Speed Estimation* (Bachelor's Thesis), MIT, 1988.

[29] S.R. Sanders, *State Estimation in Induction Machines* (Master's Thesis), MIT, 1986.

[30] T.J. Shan and T. Kailath, "Adaptive Algorithms with an Automatic Gain Control Feature", *IEEE Transaction on Circuits and Systems*, Volume 35, Series 1, January 1988.

[31] H. Sugimoto and S. Tamai, "Secondary Resistance Identification of an Induction-Motor Applied Model Reference Adaptive System and Its Characteristics", *IEEE Transaction on Industry Applications*, Volume IA-23, Series 2, March/April 1987.

[32] H. Unbehauen, *Method and Applications in Adaptive Control*, Springer-Verlag, 1980.

[33] M. Vélez-Reyes, *Speed and Parameter Estimation for Induction Machines* (Engineer's Thesis), MIT, 1988.

[34] P. Young, *Recursive Estimation and Time-Series Analysis*, Springer-Verlag, 1984.

[35] A. Zuckerberger and A. Alexandrovitz, "Controller Design for CSIM Drive Operating under Slip Frequency Strategy", *Electric Machines and Power Systems*, Hemisphere Publishing Corporation, 1988.