UNIVERSITY OF KWAZULU-NATAL

# PLC IMPLEMENTATION OF ONLINE, PRBS-BASED TESTS FOR MECHANICAL SYSTEM PARAMETER ESTIMATION

by

Vaughan Rampersad (BScEng)

Submitted in fulfilment of the requirements for the degree of Master of Science in the Faculty of Engineering, University of KwaZulu-Natal, Durban, South Africa.

As the candidate's Supervisor I agree/do not agree to the submission of this dissertation

Approved: _____

Bruce Burton, Supervisor

December 2009

I, Vaughan Rampersad, declare that

(i) The research reported in this dissertation, except where otherwise indicated, is my original work.
(ii) This dissertation has not been submitted for any degree or examination at any other university.
(iii) This dissertation does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.
(iv) This dissertation does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:
a) their words have been re-written but the general information attributed to them has been referenced;
b) where their exact words have been used, their writing has been placed inside quotation marks, and referenced.
(v) Where I have reproduced a publication of which I am an author, co-author or editor, I have indicated in detail which part of the publication was actually written by myself alone and have fully referenced such publications.
(vi) This dissertation does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the dissertation and in the References sections.

Signed: _____

Vaughan Rampersad

# ABSTRACT

This thesis investigates the use of correlation techniques to perform system identification tests, with the objective of developing online test methods to perform mechanical parameter extraction as well as machine diagnostics. More specifically, these test methods must be implemented on a Programmable Logic Controller (PLC) in combination with Variable Speed Drives (VSD).

Models for motor-based mechanical systems are derived and other documented methods for parameter identification of mechanical systems are discussed.

An investigation is undertaken into the principle that the impulse response of a system may be obtained when a test signal with an impulsive autocorrelation is injected into the system. The theory of using correlation functions to determine the numerical impulse response of a system is presented. Suitable test signals, pseudorandom binary sequences (PRBS) are analysed, and their generation and properties are discussed. Simulations are presented as to how the various properties of the PRBS test signals influence the resulting impulse response curve.

Further simulations are presented that demonstrate how PRBS-based tests in conjunction with a curve-fitting method, in this case the method of linear least squares, can provide a fair estimation of the parameters of a mechanical system. The implementation of a correlation-based online testing routine on a PLC is presented. Results from these tests are reviewed and discussed. A SCADA system that has been designed is discussed and it is shown how this system allows the user to perform diagnostics on networked drives in a distributed automation system. Identification of other mechanical phenomena such as elasticity and the non-linearity introduced by the presence of backlash is also investigated.

# ACKNOWLEDGEMENTS

I would like to express my gratitude and thanks to my supervisor Bruce Burton for introducing me to this research and providing the expertise and guidance that he did. Thanks to his passion for the field of motion control, automation systems, and machines, this work was made possible.

I would also like to thank fellow post-graduate Lance Benn for his guidance as well as humour.

I appreciate the support of the following individuals and would like to thank them:

- My parents for guiding me from a young age to "reach for the stars" and providing all the loving support and motivation to ensure that I achieve my goals
- My wife Satvika for allowing me to sacrifice time spent together to complete this research and dissertation, and also for always being there, patiently supporting me
- My fellow post-graduate and good friend Thessygan Moodley, for sharing ideas (and office space) and always being willing to assist when needed
- Professor E Boje for his open-door policy, and always being ready to offer advice and support, as well as a humorous comment

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1
# INTRODUCTION

## 1.1 General

The measurement of input-output transfer functions is a basic approach to the characterisation of a system, a process known as system identification. While the various known system identification techniques have their advantages, in some cases many of them prove to be impractical. This is the case when using test methods that require a system to be offline while under test. In most industries, machines need to be operational at all times other than during scheduled maintenance; thus going offline to perform tests on such machines is not always viable. Additionally, tests requiring the device under test to be decoupled from its load also pose a problem.

Since tests with a view to parameter extraction and diagnostics are useful, if not necessary, online test methods need to be investigated. Such tests should perform parameter extraction which can in turn be used for online adaptive tuning of controllers, as well as for condition monitoring and diagnostics.

In any industrial application, a sound knowledge of the system characteristics is crucial. The availability of detailed information about the components of the plant and what linear, non-linear, static, and dynamic characteristics are inherent, is necessary when determining the method of control, as well as the control design parameters [Ogata1].

## 1.2 Industrial Automation Systems and Drives

DC Motors are ubiquitous in many industrial applications, as it is easy to control the speed and position of a DC motor, due to its inherent linear characteristics. Consequently, an examination of a DC motor's behaviour constitutes a useful effort for the purpose of system identification in many practical applications. At the same time, induction motors are not to be ignored, since they are increasingly prevalent due to their low cost as well as the extensive amount of research that has been undertaken into the implementation of optimised speed control for induction motors. It will be seen that the initial modelling using a DC motor is justified for this research into the identification of mechanical parameters, since the method used herein is applicable independent of the electrical characteristics of the motor.

When modelling motors, such as the DC motor, the nonlinear effects are generally neglected and a linear transfer function is derived for the input-output response. This assumption produces satisfactory results as far as conventional control problems are concerned [Ogata1]. Sometimes, though, the nonlinearities and other mechanical imperfections are combined with time-varying parameters such as load inertia or stiffness, which in turn have a noticeable effect on the performance of the control system. Since speed and position control loops include the dynamics of the mechanical elements, including these mechanical imperfections, they have to be considered and the controller has to be adapted for the design of optimal dynamic control [Schütte1]. Thus is born the need for online identification schemes.

Online monitoring and diagnostics are now being applied to the early detection of broken rotor bars, insulation degradation of stator windings, and air-gap eccentricity problems, to name but a few. Moreover, vibration monitoring and other advanced analyses could be applied in the identification of certain problems. Parameter identification with algorithms such as linear least squares algorithms produce accurate estimates, but not so in the presence of nonlinearities. Methods that use so-called black box models do not reveal all the system parameters, thus restricting system identification. Likewise, neural-network based solutions are subject to a similar disadvantage [ODonovan1]. As demonstrated in [Beineke1], a combination of system identification methods including some based on extended Kalman filters, extended Luenberger observers, basis function networks, and other methods, is usually required to create a unified model of a system using purely online methods.

While there are numerous offline test methods that can parameterise a system completely, due to the downtime required with offline tests, and the associated costs, online tests are favoured.

This dissertation focuses on the development of online tests that are based on correlation techniques, particularly the result that the mathematical correlation between a system's input and output will produce the system's impulse response curve if the input has an impulsive autocorrelation. These tests are applied to the determination of the mechanical parameters of a drive system. While white noise has an ideal impulsive autocorrelation, it will be shown that the use of white noise as a test signal is not as practical as the use of pseudorandom binary sequences (PRBS), which approximate white noise and have the added advantage of being deterministic. The advantages of PRBS signals include easy generation, simplified

correlation algorithms, as well as the fact that noise does not correlate with the PRBS [Benn3].

## 1.3 Project Background and Objectives

This dissertation extends the techniques proposed in a Master's dissertation [Benn2], in which correlation-based tests using PRBS-signals were used to identify electrical characteristics, specifically stator resistance, in an induction motor. The structure of the method used in that work is shown in Figure 1-1.

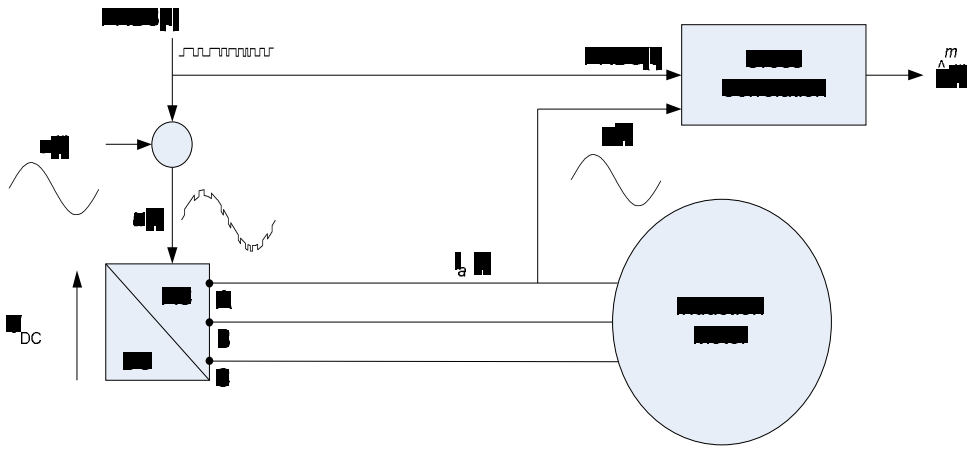



Figure 1-1 Method proposed for determining electrical characteristics of an induction motor [Benn2]

The objective of this dissertation is to investigate the implementation of the PRBS-based tests with the intention of identifying the *mechanical* parameters of a drive-based system. Specifically, this dissertation includes:

- Modelling of mechanical systems with a view to identifying certain parameters.
- A detailed study of the theory of correlation-based methods as well as the reasons for choosing PRBS signals as a viable input signal for correlation-based tests.

- The determination, by simulation, of the numerical impulse response from the abovementioned tests and how the impulse response may be used for parameter extraction.

- The application of this method to condition monitoring or diagnostics of systems. Specific cases of shaft elasticity and backlash are investigated.

- Practical verification of the theory by implementation on a PLC-based automation system.

## 1.4 Dissertation Layout

Chapter 2 discusses the modelling of a DC motor, with specific attention given to the mechanical system components. The mechanism of torque production in a DC motor is compared against that of torque production in an induction motor. The concept of field oriented control of an induction motor is briefly described and attention is drawn to the fact that an induction motor under field oriented control exhibits behaviour characteristic of a DC motor. Parameter estimation techniques are considered, with a differentiation made between offline and online techniques. Alternate methods of inertia identification are examined, as this will be shown to be necessary to verify the results of the practical implementation.

Chapter 3 introduces correlation theory and it is shown mathematically that by taking the correlation of a system's output and input, the correlation result effectively yields the system's response to the *autocorrelation* of the input signal. Therefore, by using a signal with an impulsive autocorrelation, the correlation between this input and the system output produces the numerical impulse response of the system. Pseudorandom binary sequences are shown to possess the characteristics required to determine the impulse response of a system using the correlation technique. The theory behind extracting a system's parameters from its impulse response is examined. A series of simulations are shown to verify the result that the correlation does indeed produce the impulse response, as well as to demonstrate that varying the characteristics of the PRBS sequence has an effect on the resulting numerical impulse response. This relationship between the PRBS sequence parameters and the impulse response necessitates proper scaling of the numerical impulse response in order to get the system's response to a unit impulse, and the scaling factors are outlined. The method of linear least squares is used to perform a first order curve fit before parameter estimation is performed using the least squares approximation. The error between the estimate and the

actual value is shown to be small enough to make this PRBS-based testing method suitable for the purposes of parameter estimation for this research.

Chapter 4 presents the results of simulations performed to verify the suitability of using the PRBS-based tests in order to identify the parameters of a mechanical system. The method of simulation is examined before the results are discussed. The simulations are classified into systems without a load (motor only) and systems with a load. Furthermore, experiments where the PRBS signal was added both as a torque perturbation and as a speed perturbation were simulated, and the differences between these two methods are outlined, as well as the results from both sets of simulations. The PRBS-based tests are then applied toward the detection of non-linear phenomena that may be present in the mechanical system, such as torque controller saturation, shaft elasticity, and backlash.

In Chapter 5, the implementation of these tests on an automation system comprising a PLC, a VSD, a speed encoder, and a motor-based mechanical system, is presented. The configuration of the test platform as well as the algorithms that were written to produce the PRBS signals, sample the system output, and perform the system correlation are also discussed in this chapter. Preliminary results are examined, and estimations of the system inertia under various inertial loads are presented. Measured data from systems with deliberately induced elasticity and backlash are displayed and compared to simulation results.

Chapter 6 summarises the work done in this research, presents conclusions that were deduced, as well as provides suggestions for future work in this field.

## 1.5 Summary

This dissertation investigates the use of correlation techniques to perform system identification tests, and applies these tests to mechanical parameter estimation as well as machine diagnostics. Detailed simulations are presented and analysed to verify the feasibility of this parameter estimation technique. A practical implementation of these tests is demonstrated on a test platform that, while built for the purpose of this research, was modelled after most industrial system configurations, thus lending to the flexibility as well as feasibility of using these methods in industry to identify mechanical parameters as well as in condition monitoring endeavours.

# CHAPTER 2
# IDENTIFYING MECHANICAL PARAMETERS

## 2.1 Introduction

In order to research test methods toward the identification of systems, it is imperative that one understands the systems themselves. This chapter begins by modeling a DC motor and uses the model to determine the state space equations that govern the dynamic behaviour of the DC motor. The mechanism of torque production in a DC motor is then compared to that of an induction motor and a few salient, yet critical, differences are identified. Nevertheless, it is shown that when an induction motor is operated under field oriented control, its behaviour approximates that of a DC motor, under the conditions expected within the scope of this research. An overview of various parameter estimation techniques is outlined, as well as alternate methods to determine the inertia of a motor as these methods will be used later in the practical implementation of the correlation-based tests.

## 2.2 DC Motor Model

For the purposes of simulation, it was necessary to derive the dynamic model for a motor. Since the DC motor is simpler to model than an induction motor, and the focus is not on the motor itself but on the mechanical parameters, the modeling of, and subsequent simulations with, a separately excited DC motor were performed.

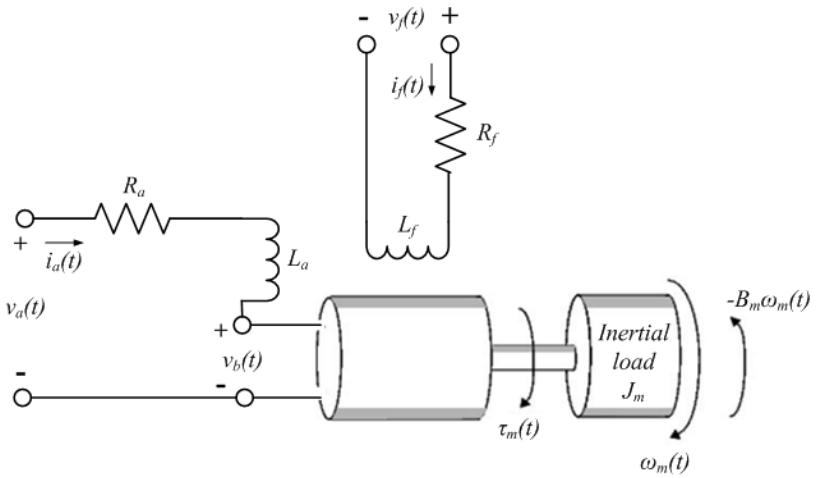



Figure 2-1 Diagram depicting electrical model of motor, with single mass load

In Figure 2-1, the following quantities are defined,

| | | | |
|---|---|---|---|
| $i_a(t)$ | - | current in the motor armature | (A) |
| $v_a(t)$ | - | voltage applied across motor terminals | (V) |
| $v_b(t)$ | - | back EMF | (V) |
| $R_a$ | - | resistance of motor armature | (Ω) |
| $L_a$ | - | inductance of motor armature | (H) |
| $i_f(t)$ | - | current in the field windings | (A) |
| $v_f(t)$ | - | voltage applied across field windings | (V) |
| $R_f$ | - | resistance of field winding | (Ω) |
| $L_f$ | - | inductance of field winding | (H) |
| $\tau_m(t)$ | - | torque produced by the motor | (Nm) |
| $J_m$ | - | inertia of the motor rotor and shaft | ($kgm^2$) |
| $B_m$ | - | viscous friction of the motor rotor and shaft | ($Nm.rad^{-1}s$) |
| $\omega_m(t)$ | - | angular velocity of the motor | ($rad.s^{-1}$) |

### 2.2.1 Electrical Equations

Kirchhoff's voltage law is applied to the electrical circuit of the DC motor, yielding,

$$v_a(t) = v_b(t) + L_a \frac{di_a}{dt} + R_a i_a(t) \tag{2.1}$$

The back EMF, is proportional to the angular velocity $\omega_m(t)$,

$$v_b(t) = k_e \omega_m(t) \tag{2.2}$$

where $k_e$ = electric constant, unit ($V.rad^{-1}s$), determined by the motor design and windings.

In this case,

$$k_e = k_f i_f$$

where $k_f$ is a constant determined by the design of the field windings, and the field current is assumed constant.

Equation (2.1) can be rewritten as,

$$\frac{di_a}{dt} = -\frac{R_a}{L_a} i_a(t) - \frac{k_e}{L_a} \omega_m(t) + \frac{1}{L_a} v_a(t) \tag{2.3}$$

Equation (2.3) is the first state differential equation for the DC motor.

### 2.2.2 Mechanical Equations

From Newton's Second Law applied to rotating bodies, the sum of all the torques on the motor shaft is equal to the inertial load multiplied by the angular acceleration,

$$J_m \frac{d\omega_m}{dt} = \tau_m - B_m \omega_m(t) \tag{2.4}$$

The torque produced by the motor $\tau_m(t)$ is proportional to the current $i(t)$ produced by the applied voltage $v_a(t)$,

$$\tau_m(t) = k_t i(t) \tag{2.5}$$

where $k_t$ = torque constant, unit ($NmA^{-1}$), and is a function of the square root of the armature resistance. Since the armature resistance is determined by the construction of the motor and *ideally* does not change during the lifetime of the motor, this factor is assumed to be constant.

Substituting the result from Equation (2.5),

$$\therefore J_m \frac{d\omega_m}{dt} = k_t i(t) - B_m \omega_m(t) \tag{2.6}$$

$$\therefore \frac{d\omega_m}{dt} = \frac{k_t}{J_m} i(t) - \frac{B_m}{J_m} \omega_m(t) \tag{2.7}$$

This second state differential equation describes the mechanical system behaviour for a DC motor and it will be shown that Equation (2.7) needs to be modified when a load is added to the motor.

### 2.2.3 State Space Representation

From Equations (2.3) and (2.7), it is possible to develop a state-space representation of the DC motor. The two states of the system are current *i(t)* and angular velocity $\omega_m(t)$. The input to the system is the applied voltage $v_a(t)$, while the output is the angular velocity $\omega_m(t)$. Grouping these two differential equations,

$$\begin{cases} \dfrac{di_a}{dt} = -\dfrac{R_a}{L_a} i_a(t) - \dfrac{k_e}{L_a} \omega_m(t) + \dfrac{1}{L_a} v_a(t) \\ \dfrac{d\omega_m}{dt} = \dfrac{k_t}{J_m} i_a(t) - \dfrac{B_m}{J_m} \omega_m(t) \end{cases}$$

The system can also be represented in state-space form, as follows,

$$\begin{bmatrix} \dfrac{di_a}{dt} \\ \dfrac{d\omega_m}{dt} \end{bmatrix} = \begin{bmatrix} -\dfrac{R_a}{L_a} & -\dfrac{k_e}{L_a} \\ \dfrac{k_t}{J_m} & -\dfrac{B_m}{J_m} \end{bmatrix} \begin{bmatrix} i_a(t) \\ \omega_m(t) \end{bmatrix} + \begin{bmatrix} \dfrac{1}{L_a} \\ 0 \end{bmatrix} v_a(t) \tag{2.8}$$

From the differential equations, a block diagram model of the DC motor can be derived. The derivation of the various block diagrams and transfer functions describing the DC motor is included in the following sections.

## 2.3 Mechanical System Modeling

### 2.3.1 DC Motor with No Load

The two differential equations that are used to model the DC motor, Equations (2.3) and (2.7), are represented in block diagram form in Figure 2-2.

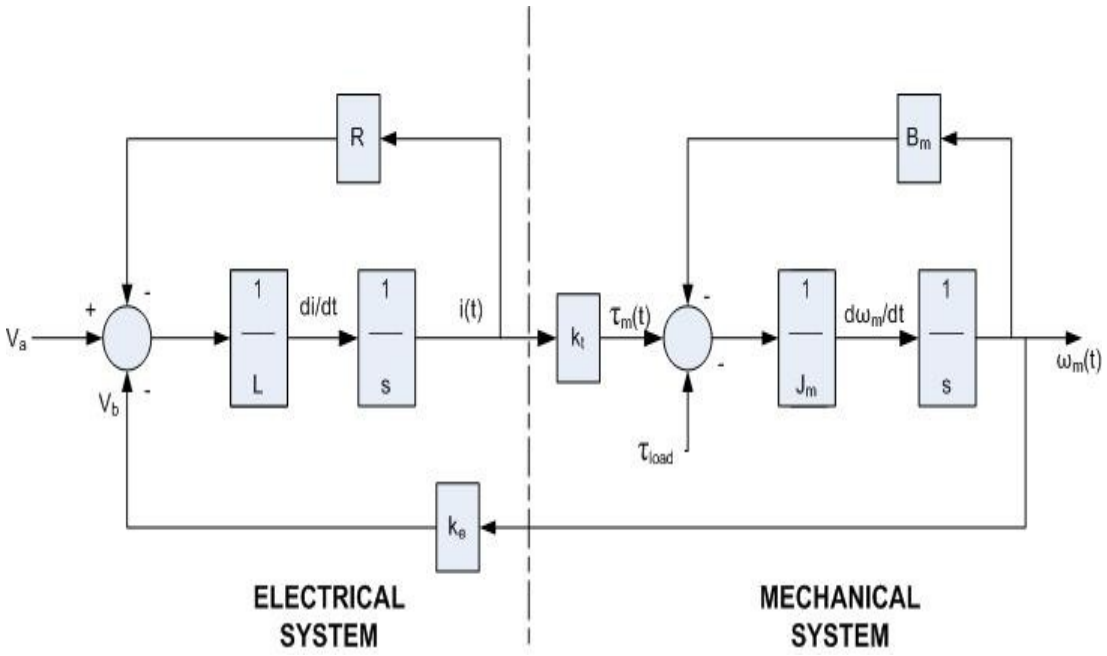



Figure 2-2 Complete DC Motor model (no load)

The block diagram in Figure 2-2 can be simplified further, with the resulting block diagram in Figure 2-3. From this model, the transfer functions from applied voltage, $v_a$, to speed, $\omega_m$, and torque, $\tau_m$, can be found. For the sake of completeness the derivation is included in the following two sections.

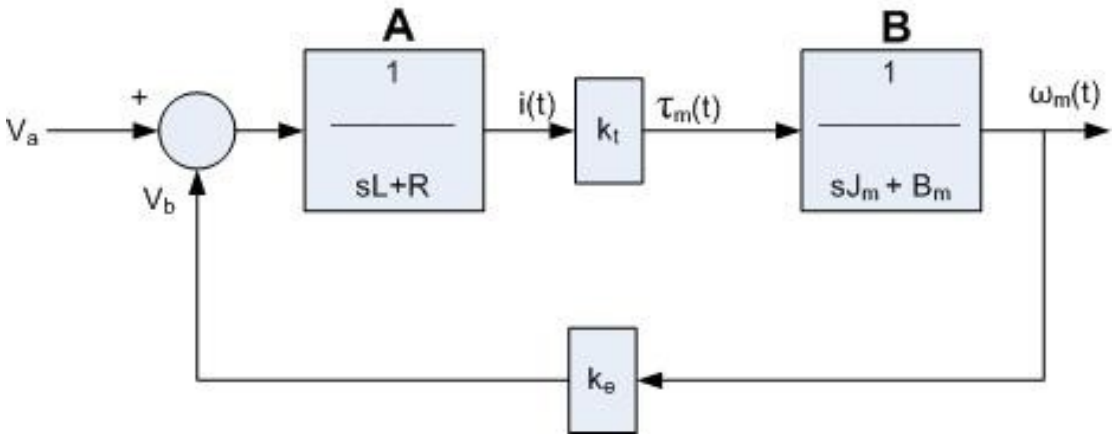



Figure 2-3 Simplified DC motor model

### 2.3.2 Transfer Function from Applied Voltage ($v_a$) to Motor Torque ($\tau_m$)

$$\begin{aligned}\tau_m &= k_t A(v_a - Bk_e\tau)\\ &= k_t A v_a - Bk_e k_t A\tau\end{aligned}$$

$$\begin{aligned}\tau_m(1+Bk_e k_t A) &= k_t A v_a\\ \frac{\tau_m}{v_a} &= \frac{k_t A}{1+Bk_e k_t A}\\ &= \frac{k_t\left[\dfrac{1}{sL+R}\right]}{1+k_t k_e\left[\dfrac{1}{sJ_m+B_m}\right]\left[\dfrac{1}{sL+R}\right]}\\ &= \frac{k_t(sJ_m+B_m)}{s^2(J_mL)+s(B_mL+RJ_m)+(B_mR+k_t k_e)}\\ &= T_{\tau/V}\end{aligned} \qquad (2.9)$$

### 2.3.3 Transfer Function from Applied Voltage ($v_a$) to Motor Speed ($\omega_m$)

$$\begin{aligned}\omega_m &= Bk_t A(v_a - k_e\omega_m)\\ &= Bk_t A v_a - BAk_t k_e\omega_m\end{aligned}$$

$$\begin{aligned}\omega_m(1+Bk_t k_e A) &= Bk_t A v_a\\ \frac{\omega_m}{v_a} &= \frac{Bk_t A}{1+Bk_t K_e A}\\ &= \frac{k_t\left[\dfrac{1}{sJ_m+B_m}\right]\left[\dfrac{1}{sL+R}\right]}{1+k_t k_e\left[\dfrac{1}{sJ_m+B_m}\right]\left[\dfrac{1}{sL+R}\right]}\\ &= \frac{k_t}{(sJ_m+B_m)(sL+R)+k_t k_e}\\ &= \frac{k_t}{s^2(J_mL)+s(B_mL+RJ_m)+(B_mR+k_t k_e)}\\ &= T_{\omega/V}\end{aligned} \qquad (2.10)$$

### 2.3.4 Transfer Function from Torque Reference to Torque Output

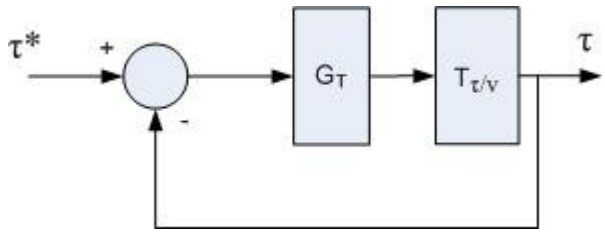



Figure 2-4 Transfer function from torque reference (t*) to torque output

In Figure 2-4, $T_{\tau/V}$ is the transfer function from applied voltage to motor torque, as defined in Equation (2.9). The torque controller is represented by $G_T$. Figure 2-5 depicts a closed loop torque control system, from torque reference to torque output.

The focus of this research is in developing online tests and as such, it is necessary to determine the effect of injecting perturbations into the reference inputs of the control functions implemented in variable speed drives. In this case, the transfer function from torque reference to motor torque produced is investigated, that is, with the torque controller taken into account. From here onward, the *m* subscript that was previously attached to $\tau$ and $\omega$ will be dropped, unless ambiguity could occur. Therefore, $\tau$ and $\omega$ refer to motor torque and motor speed, respectively.

$$\begin{aligned}\frac{\tau}{\tau^*} = T_{\tau/\tau^*} &= \frac{G_T T_{\tau/V_a}}{1 + G_T T_{\tau/V_a}} \\ &= \frac{G_T \left[ T_{\omega/V_a} (sJ_m + B_m) \right]}{1 + G_T \left[ T_{\omega/V_a} (sJ_m + B_m) \right]} \approx 1, \text{ for } G_T >> 1\end{aligned} \tag{2.11}$$

From this transfer function it can be seen that if $G_T$ is large, then the transfer function approximates to 1. This is expected for a well-tuned torque controller, since it is desirable that the torque output tracks the torque reference.

### 2.3.5 Transfer Function from Speed Reference to Speed Output

Similar to the investigation undertaken in the previous section, in this section the transfer function from the speed reference to the motor speed is derived. Note that previous approximations with regard to the transfer function from torque reference to torque speed is used in this derivation. The result helps aid understanding the effect that injecting a PRBS perturbation into the speed reference has on the motor.

In Figure 2-5, $G_S$ is the speed controller, while $T_{\tau/\tau^*}$ is the transfer function from torque reference to torque output, as defined by Equation (2.11), and which accounts for the torque controller. Figure 2-5 depicts a closed loop speed control system, from speed reference to speed output.

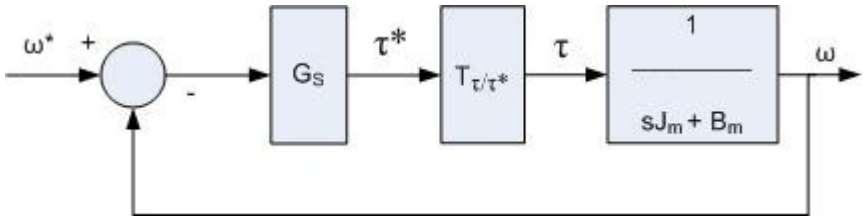



Figure 2-5 Transfer function from speed reference (ω*) to speed (ω)

$$\begin{aligned}\frac{\omega}{\omega^*} &= \frac{G_S T_{\tau/\tau^*}\left[\dfrac{1}{sJ_m + B_m}\right]}{1 + G_S T_{\tau/\tau^*}\left[\dfrac{1}{sJ_m + B_m}\right]} \\ &\approx \frac{G_S\left[\dfrac{1}{sJ_m + B_m}\right]}{1 + G_S\left[\dfrac{1}{sJ_m + B_m}\right]} \\ &= \frac{G_S}{sJ_m + B_m + G_S} = \frac{\left(\dfrac{G_S}{B_m + G_S}\right)}{s\left(\dfrac{J_m}{B_m + G_S}\right) + 1} \\ &= T_{\omega/\omega^*}\end{aligned} \tag{2.12}$$

### 2.3.6 Transfer Function from Torque Reference to Speed Output, Speed Loop Open

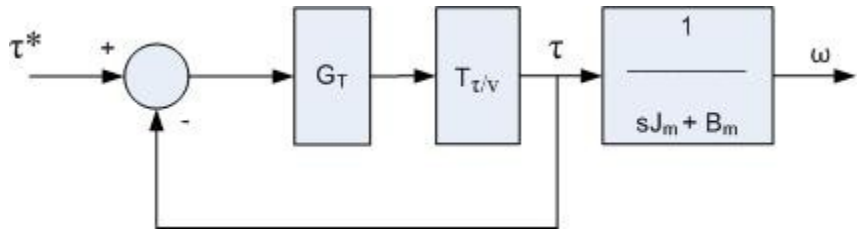



Figure 2-6 Transfer function from torque reference (τ*) to speed output (ω)

Equation (2.11) showed that for a well tuned torque controller ($G_T$), the transfer function from torque reference to torque controller output approximates to 1. For this to hold true, the torque controller would have to be an infinite bandwidth torque controller, which is clearly not possible in a practical implementation. This simplification is however used here for approximation purposes, because when applied to the block diagram in Figure 2-6, the following approximation results,

$$T_{\omega/\tau^*} = \frac{\omega}{\tau *} \approx \frac{1}{sJ_m + B_m} \tag{2.13}$$

Thus, if the PRBS was injected into the torque reference without closed-loop speed control, the resulting numerical impulse response when the PRBS is correlated with the speed would be directly related to the mechanical system's impulse response. In a practical implementation, the finite bandwidth of the torque controller will affect the performance of the parameter estimation, and later results will show this to be the case, e.g. as shown in Section 4.3.

### 2.3.7 Transfer Function from Additional Torque Reference to Speed Output, with Closed-Loop Speed Control

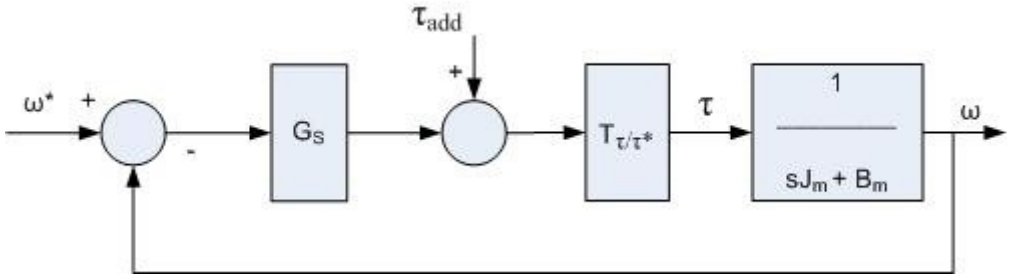



Figure 2-7 Transfer function from additional torque setpoint ($\tau_{add}$) to speed output ($\omega$), with the speed loop closed

$$\begin{aligned}\omega &= \left[\frac{1}{sJ_m + B_m}\right]\left[T_{\tau/\tau^*}\right]\left(\tau_{add} + G_S\left(\omega^* - \omega\right)\right) \\ &\approx \left[\frac{1}{sJ_m + B_m}\right]\left(\tau_{add} + G_S\left(\omega^* - \omega\right)\right) \\ &= \left[\frac{\tau_{add}}{sJ_m + B_m}\right] + \left[\frac{G_S\omega^*}{sJ_m + B_m}\right] - \left[\frac{G_S\omega}{sJ_m + B_m}\right]\end{aligned} \tag{2.14}$$

In order to simplify the above, assume $\omega^*=0$. Then, the following transfer function results,

$$\begin{aligned}\omega &= \left[\frac{\tau_{add}}{sJ_m + B_m}\right] - \left[\frac{G_S\omega}{sJ_m + B_m}\right] \\ \left(sJ_m + B_m\right)\omega &= \tau_{add} - G_S\omega \\ \frac{\omega}{\tau_{add}} &= \frac{1}{sJ_m + B_m + G_S} = \frac{\left(\frac{1}{B_m + G_S}\right)}{s\left(\frac{J_m}{B_m + G_S}\right) + 1}\end{aligned} \tag{2.15}$$

From Equation (2.15), if $G_S >> 1$, then the speed controller term, $G_S$, dominates the viscous friction coefficient term, $B_m$. The speed controller term must then be compensated for when determining the mechanical parameters – this term may be determined from the variable speed drive parameters.

If $G_S << 1$, then the transfer function approximates that of the mechanical system alone. Importantly, for any value of $G_S$, by applying the PRBS signal as an additional torque perturbation with the speed loop closed, the transfer function that will be approximated with this technique is that of the mechanical system.

## 2.4 Torque Production in an Induction Motor vs. Torque Production in a DC Motor

Historically, DC motor-based drive systems were the first choice for many industrial applications due to their ease of control over a wide speed range in combination with their relatively simple and well-documented dynamic characteristics. The disadvantages of DC motors are their size and cost, primarily due to their relatively complex construction.

On the other hand, the induction motor proves easier to produce due to its simple construction. Unlike the DC motor, there is no need for secondary power supplies for field excitation. However, the induction motor posed a problem when it came to controlling torque at low speeds, and its range of speed control was not as wide as the DC motor. With improvements in technology in recent years, more specifically solid-state electronics and computing power, high-performance dynamic control of the induction motor has become possible with many AC drives available in the market place today with highly dynamic, self parameterizing field oriented or "vector" torque controllers.

Some of the industrial drives available include, but are not limited to:

- ABB "Direct Torque Control" drives (ACS 600, ACS 800, and ACS 1000)
- Control Techniques "Rotor Flux Control" drives (Unidrive SP)
- Danfoss "Voltage Vector Control" drives (VLT 5000 and VLT 6000)
- Mitsubishi "Sensorless Vector Control" drives (FR-D 700 Series and FR-E 700 Series)
- Siemens "Vector Control" drives (MM140, G120, S120, Masterdrives VC, Masterdrives MC, and Simodrive)

While the purpose of this thesis is to investigate the mechanical models of motor-based systems, an understanding of the electrical dynamics of motors is required, as it is the electrical dynamics in a motor that produces mechanical torque.

In order to understand the justification of representing an induction motor under Field Oriented Control (FOC) as a DC motor when modeling its electrical dynamics, it is necessary to investigate the nature of torque production in both machines, as well as the underlying principles of operation, and the concept of Field Oriented Control.

### 2.4.1 Torque Production in a DC Motor

In a DC motor, a stationary field flux linkage space vector $\underline{\lambda}_f$ is created by the current from the field (stator) winding, which is fixed in space. The commutator, brushes, and compensating windings ensure that the armature (rotor) current produces an armature flux linkage space vector $\underline{\lambda}_a$ that is in quadrature with $\underline{\lambda}_f$ at all times. This is shown in Figure 2-8 and Figure 2-9.

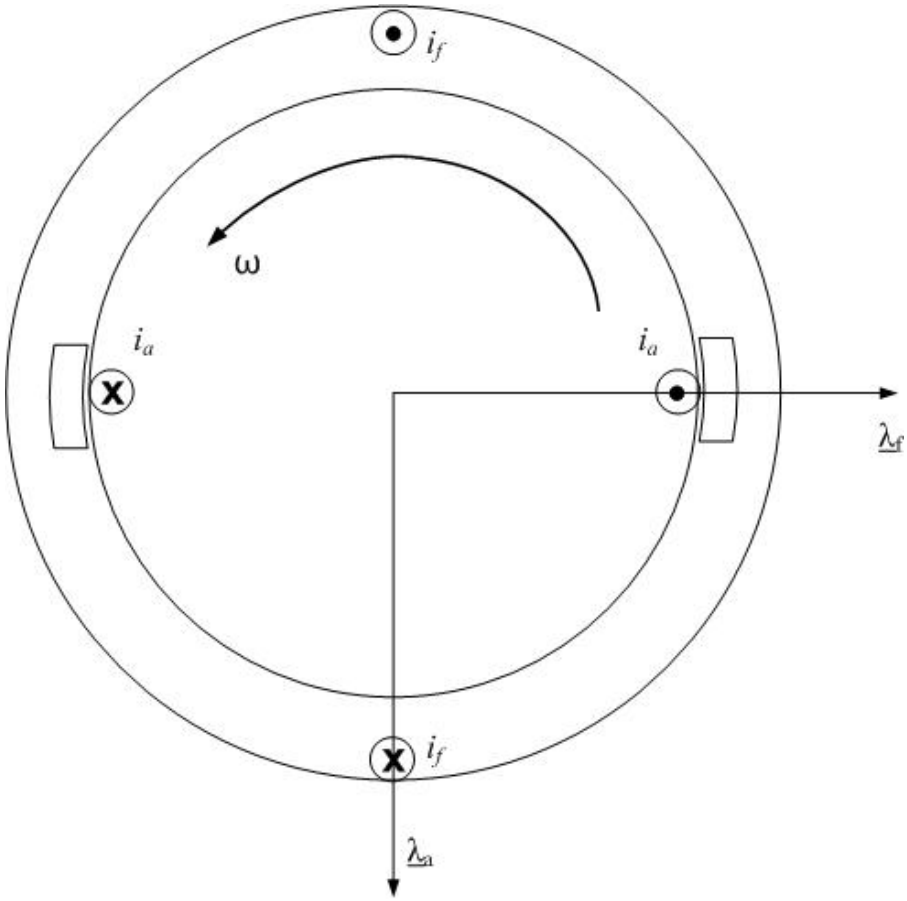



Figure 2-8 Schematic of a cross-section of a DC motor

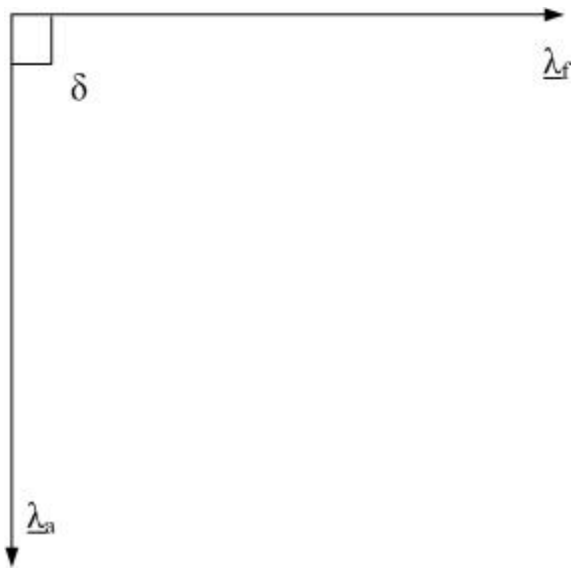



Figure 2-9 Vector representation for a DC motor

The torque produced in a DC motor $\tau_{DC}$ is defined by the external cross product of $\underline{\lambda}_f$ and $\underline{\lambda}_a$, and since these two vectors are orthogonal and flux linkage $\lambda$ is the product of the number of winding turns $N$ and the current in the windings $i$ it follows that the torque is proportional to the product of the field current $i_f$ and armature current $i_a$ assuming ideal commutation with no saturation. This is shown in Equation 2-9.

$$\begin{aligned} \tau_{dc} &= K_{dc} \left| \underline{\lambda_f} \times \underline{\lambda_a} \right| \\ &= K_{dc} \left| \underline{\lambda_f} \right| \left| \underline{\lambda_a} \right| \sin \delta \\ &= K_{dc} i_f i_a \end{aligned} \tag{2.16}$$

From this result it is noted that since the angle $\delta$ between the interacting fields which produce the torque is 90° for all operating conditions in a DC motor, the torque produced in a DC motor is maximum for all values of $\lambda_f$ and $i_a$.

Since $\lambda_f$ and $i_a$ are orthogonal, they are said to be decoupled, allowing for torque control by separately controlling each of the states, as they do not affect each other. The field winding has a large time constant when compared with the armature winding, and it desirable to minimise electrical losses by producing any desired torque using the smallest possible armature current, thus the torque is in most applications controlled by controlling the armature current ($\tau_{DC}$ α $i_a$) while keeping the field flux linkage $\lambda_f$ constant.

The simplicity of torque control and subsequently speed control of a DC motor, coupled with the fact that it is inherent in the design of the DC motor to realise maximum torque at all times for a given flux and given current, has long been recognised and has led to the popularity of the DC motor in applications requiring torque and speed control.

### 2.4.2 Torque Production in an Induction Motor

The physics of torque production is the same in the induction motor as it is in the DC motor, whereby torque is proportional to the stator flux and rotor flux (in the case of the separately excited DC motor, field winding flux and armature winding flux). However, torque control in the induction motor is more complicated because the rotor flux is induced by the stator flux.

The stator is excited by a *sinusoidal* supply, which generates a stator flux linkage space vector $\underline{\lambda}_s$ that rotates in the air gap with the synchronous air gap frequency $\omega_e$. The stator flux linkage space vector $\underline{\lambda}_s$ consists of a mutual or magnetising component $\underline{\lambda}_m$ as well as a leakage flux linkage component $\underline{\lambda}_{ls}$, due to imperfect coupling between the stator and rotor windings. Both these components rotate at $\omega_e$, thus

$$\underline{\lambda}_s = \underline{\lambda}_m + \underline{\lambda}_{ls} \tag{2.17}$$

When the rotor speed $\omega_{re}$ is less than the synchronous angular frequency $\omega_e$, the mutual flux linkage component $\lambda_m$ moves over the rotor surface at an angular frequency of $\omega_e - \omega_{re}$. This is known as the slip angular frequency, where

$$s\omega_e = \omega_e - \omega_{re} \tag{2.18}$$

The relative motion between $\underline{\lambda}_m$ and the rotor surface induces rotational voltages with frequency $s\omega_e$ into the rotor phases. In turn, these induced voltages cause currents to flow around the closed rotor circuits, creating a rotor flux linkage space vector $\underline{\lambda}_r$. With the rotor conductors rotating at angular frequency $\omega_{re}$, from Equation 2-18 the angular frequency of $\underline{\lambda}_r$ is given by $s\omega_e+\omega_{re}$. Therefore, at steady state both $\underline{\lambda}_s$ and $\underline{\lambda}_r$ rotate at $\omega_e$ and are stationary with respect to each other with the torque angle $\delta$ between them. The rotor flux linkage space vector also has a leakage flux linkage component, so it follows that

$$\underline{\lambda}_r = \underline{\lambda}_m + \underline{\lambda}_{lr} \tag{2.19}$$

where $\underline{\lambda}_{lr}$ is the rotor leakage flux linkage space vector.

The reaction components of the stator flux linkage and the rotor flux linkage cause the torque angle $\delta$ to be less than 90°.

Torque on the rotor is produced in the direction of the rotor movement due to the interaction of the stator and rotor flux linkage space vectors. Similarly to Equation 2-16, the torque produced is determined by the external cross product of $\underline{\lambda}_s$ and $\underline{\lambda}_r$, which may also be expressed as proportional to the product of the magnitudes of the two vectors and the sine of the torque angle $\delta$ between them.

$$\begin{aligned} T_{em} &= K_{im} \left| \underline{\boldsymbol{\lambda}}_r \times \underline{\boldsymbol{\lambda}}_s \right| \\ &= K_{im} \left| \underline{\boldsymbol{\lambda}}_r \right| \left| \underline{\boldsymbol{\lambda}}_s \right| \sin \delta \end{aligned} \tag{2.20}$$

where $K_{im}$ is the constant of proportionality for an induction motor, and depends on conditions in the induction motor, such as stator and rotor windings.

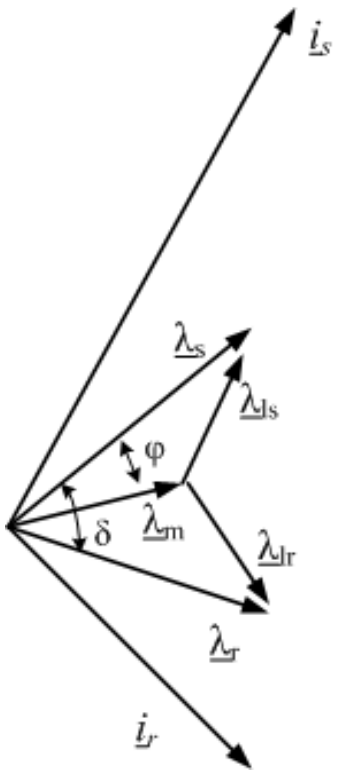



Figure 2-10 Vector representation for an induction motor

With reference to Figure 2-6, the torque produced is also proportional to the product of the magnitude of $\underline{\lambda}_m$ and $\underline{\lambda}_s$ and the sine of the angle $\varphi$ between these two vectors.

$$T_{em} = K_m \left|\underline{\lambda}_m\right| \left|\underline{\lambda}_s\right| \sin\varphi \tag{2.21}$$

where $K_m$ is the new constant of proportionality. Note that since $\varphi < \delta$, and $\underline{\lambda}_m < \underline{\lambda}_s$, for Equations (2.20) and (2.21) to yield the same value for torque $T_{em}$, the constant of proportionality $K_m > K_{im}$.

It is apparent that controlling the torque in an induction motor is more difficult when compared to the DC motor, for the following reasons:

1. The torque angle $\delta$ is not fixed at 90° to give maximum torque for any given $\underline{\lambda}_s$ and $\underline{\lambda}_r$.

2. $\underline{\lambda}_s$ and $\underline{\lambda}_r$ are coupled since $\underline{\lambda}_r$ is induced by $\underline{\lambda}_s$, which makes independent control of each of $\underline{\lambda}_s$ and $\underline{\lambda}_r$ impossible under dynamic conditions.

3. Both $\underline{\lambda}_s$ and $\underline{\lambda}_r$ rotate in space. Under dynamic conditions, $\underline{\lambda}_s$ and $\underline{\lambda}_r$ and $\delta$ vary according to the natural response, which is determined by the machine/load characteristics.

4. The torque angle $\delta$ is not directly measurable.

5. $\underline{\lambda}_s$ and $\underline{\lambda}_r$ depend on the magnitude, frequency, and phase angle of the stator currents.

6. The rotor currents are not easily measurable as in the case of the DC motor.

In order to achieve torque control in an induction motor, control strategies such as volts/hertz and later Field Oriented Control (FOC) have been developed and implemented.

## 2.5 Voltage/Frequency Control

Voltage/frequency, or volts/hertz torque control is based on a steady state analysis of the induction motor. The volts/hertz ratio is fixed so that at steady state the magnitude of $\underline{\lambda}_m$ remains at rated value for different stator frequencies or speeds. This method thus works best for *open loop* torque control applications.

However, the torque producing component of current and the flux producing component of current is not properly decoupled during *transient* conditions, and the volts/hertz torque control does not compensate for this.

## 2.6 Field Oriented Control

Field oriented control (FOC) or vector control addresses the limitations of the simple volts/hertz strategy by moving from a steady state analysis of the motor to an analysis based on instantaneous current values. The objective of FOC is to establish and maintain an explicit angular relationship between the stator current vector and the rotor flux that is produced within the air gap [Diana1].

While research has been done into FOC and the associated theory, it is beyond the scope of this research to include a detailed analysis of FOC for the induction motor. Reference may be made to ([Kleinhans1], [Diana1], [Rubin1]) for further information on FOC.

According to [Kleinhans1], with FOC, the decoupled control of the torque producing component and the flux linkage producing components of current makes the effective dynamics of the controller and induction machine appear as an equivalent DC machine. This conclusion is the basis for using a DC motor model for an induction motor, since the induction motor under system identification with the methods proposed herein are assumed to be in a variable speed drive (VSD) controlled application, with field oriented control properly implemented on the induction motor.

## 2.7 Parameter Estimation Techniques

In any industrial application, it is important to have a comprehensive knowledge of the system characteristics. Detailed information about the machine parameters assists in selecting the control philosophies as well as determines the design procedures undertaken.

Parameter estimation is invaluable for optimised torque control when using FOC. Linear torque control must be maintained in high precision applications such as robotics and position control applications. Furthermore, changes to certain machine parameters may be indicative of a pending malfunction.

When the system parameters are not readily available or are dependent on operating conditions or time, a system identification procedure or parameter estimation technique must be used to obtain these parameters. Many methods have been designed to estimate parameters, particularly for induction motors, due to their complexity. These methods may be classified as belonging to one of two categories: offline estimation techniques and online estimation techniques.

Offline estimation techniques require the machine to be removed from its normal operating conditions, such as disconnected from its power supply and at rest. Using either direct measurements or estimation techniques, or a combination of both, parameter identification is achieved.

The focus in this research is on online estimation tests due to the inherent inflexibility when using offline tests on a machine that needs to operate continuously. A number of online estimation techniques have been researched and implemented owing to this need to work on running systems. For the purpose of this dissertation, much research was done in the field of online parameter estimation techniques and some noteworthy methods may be found in ([Wertz1], [Beineke1], [Beineke2], [Schütte1], [Amann1], [ODonovan1], [Kara1]).

It soon becomes apparent that no single test may be considered as the most important or accurate, but comprehensive system identification results from using a combination of the methods available. The method that this research focuses on is envisioned as being one of the tools that one may utilise in the online identification of mechanical systems.

## 2.8 Inertia Estimation

Various estimation techniques have been proposed for the determination of the moment of inertia. It is possible to obtain an approximation of the inertia of a motor on its own if the dimensions of the rotor are known. For example, if the length $l_s$, radius $r_s$, and mass $m_s$ of the rotor shaft are known together with the length $l_c$, outside and inside radii ($r_{oc}$, $r_{ic}$, $r_s$), and mass $m_c$ of the rotor core, then one can use the formula $J=½m_s r_s^2+½m_c(r_{oc}^2+r_{ic}^2)$. However, this expression does not include the effects of the slots, windings, and the fan on the rotor [Vas1]. The inertia can be more accurately determined by experiment, and one of these experimental methods is described below.

### 2.8.1 Retardation Tests

This conventional method has the benefit of being applicable to any type of motor, as the results are independent of the electrical part of the motor. In a retardation test, the motor is initially run at some constant speed and then disconnected from the supply. As the speed decreases to zero, the angular rotor speed of the motor is monitored. Since the acceleration, $J\,d\omega/dt$, contains the inertia $J$, and the friction and windage torque, $B\omega$, contains the viscous friction coefficient $B$, it is possible to determine both $J$ and $B$ by solving two equations simultaneously, i.e. two retardation experiments need to be performed. For one test, the

machine deceleration characteristic, $\frac{d\omega_1}{dt}\omega_1$, is monitored, while for the second test an additional, known, inertia is added to the rotor. This inertia could be a disc with inertia $J_d$, where the inertia could be quite accurately determined using the formula $J=½m_d.r_d^2$. This new deceleration characteristic, $\omega_2$, is also measured and the data is then used along with the following force-balance equations:

$$J\frac{d\omega_1}{dt} + B\omega_1 = 0 \tag{2.22}$$

$$(J + J_d)\frac{d\omega_2}{dt} + B\omega_2 = 0 \tag{2.23}$$

Solving these two equations simultaneously yields values for *J* and *B*.

## 2.9 Summary

This chapter introduced the DC motor model and presented transfer functions for various motor-load scenarios. The nature of torque control in both the DC motor as well as the induction motor were discussed, and two types of commonly used torque control strategies were introduced, namely, voltage/frequency control and field oriented control. It was stated, with the aid of other references, that an induction motor under field oriented control exhibits behaviour characteristic of a DC motor model; hence it was concluded that using a DC motor model for all simulations was sufficient for this research, as the practical method proposed involves induction motors under field oriented control. A brief comparison between offline and online parameter estimation techniques was made, with references given to other publications which this author read through during the course of this research. One method in particular was presented in greater detail – the retardation test for estimation of the inertia parameter of a mechanical system.

Chapter 3 introduces the method of correlation and how correlation can be used in a system identification technique. The necessary theoretical background to this method will also be discussed in the next chapter.

# CHAPTER 3
# A CORRELATION-BASED METHOD FOR ONLINE SYSTEM IDENTIFICATION

## 3.1 Introduction

This chapter discusses the use of a correlation-based method to identify a mechanical system. Correlation-based methods require that the system under investigation be excited by a signal with special properties. These properties are discussed herein and it is shown that Pseudo Random Binary Sequences satisfy the criteria for an excitation signal that may be used for the correlation-based method.

The identification method involves superimposing a PRBS input on the system's input and then correlating the original PRBS signal with the system response. The result of this correlation is a scaled version of the system's impulse response. By using mathematical curve-fitting techniques, the system's parameters are identified from the impulse response curve.

In order to explain the theoretical basis of this method in more detail, it is necessary to discuss correlation theory.

## 3.2 Correlation Theory

The essence of correlation techniques to perform a system identification depends on the principle that the cross correlation of the input *u(t)* with the output *y(t)* of a system yields the system's impulse response, if *u(t)* has an impulsive autocorrelation [Jackson1]. Correlation functions thus play an important role in analysing signals in the time domain. In general, the correlation between two signals is a measure of similarity between the signals, as a function of their relative positions in time. As mentioned above, by carefully selecting the input to a system, the correlation between the input and output of a system can provide important information about the system, such as the impulse response. The following discusses how this is possible mathematically.

In Equation (3.1), $\varphi_{uy}(\tau)$ is the cross correlation between the input *u(t)* and the output *y(t)* of a system. From Equation (3.1) it is noted that the cross correlation computes the time average of the product of the two signals, separated in time by a period of $\tau$ seconds, where $\tau$ is a variable. Equation (3.2) shows how the cross correlation is computed when the signal order

is reversed. Two signals that are completely uncorrelated, or totally "dissimilar", yield a correlation of zero for all values of $\tau$. The autocorrelation function is the cross correlation of a signal with itself, as can be seen in Equation (3.3).

$$\varphi_{uy}(\tau) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} u(t)y(t+\tau)dt \tag{3.1}$$

$$\varphi_{yu}(\tau) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} y(t)u(t+\tau)dt \tag{3.2}$$

$$\varphi_{uu}(\tau) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} u(t)u(t+\tau)dt \tag{3.3}$$

The autocorrelation function is symmetrical about $\tau=0$ and is thus usually computed for $\tau\geq 0$. For a periodic signal, the autocorrelation is also periodic. Thus, the autocorrelation is usually computed for one period only for a periodic signal. Furthermore, since the autocorrelation is a special case of cross correlation (which indicates the degree of similarity between signals), the autocorrelation is at a maximum when $\tau=0$, when the signal is multiplied by itself. The autocorrelation function will be zero for $\tau$ values greater than the pulse duration. These properties of the autocorrelation function are shown in Figure 3-1 [Cremer1], where the autocorrelation function for a unit impulse is computed over four values of delay time $\tau$.

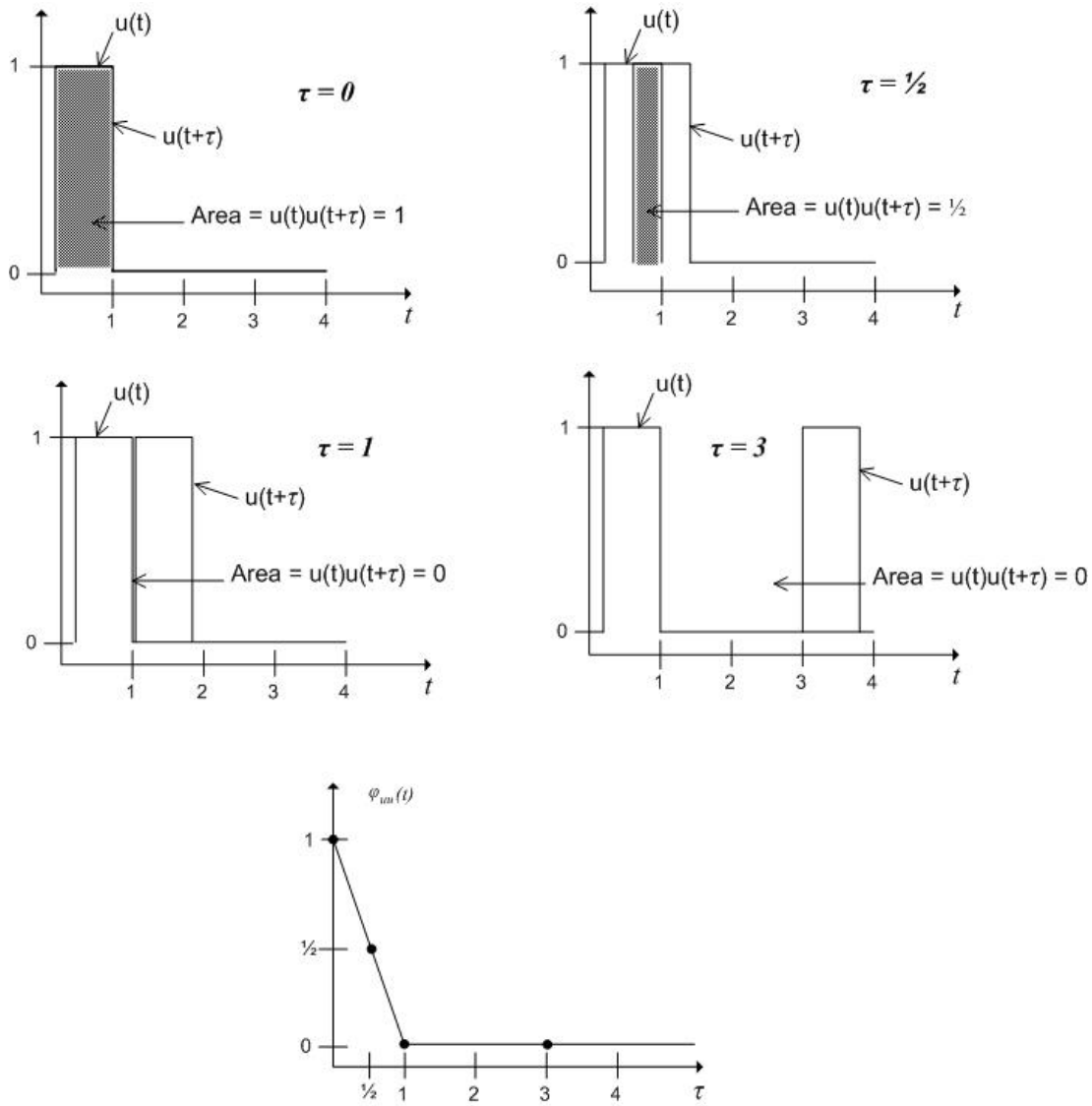



Figure 3-1 The Autocorrelation Function [Cremer1]

For the special case of a completely random signal such as white noise, which has ideally a flat power spectrum of infinite bandwidth, the autocorrelation function is simply an impulse function at $\tau=0$, and zero for $\tau\neq0$. While this satisfies the criteria for determining the impulse response of a system using correlation techniques, the use of white noise as a test signal poses other problems in practice and does not yield very good results. The main problem with using white noise is the long averaging times required to reduce statistical errors to acceptable levels [Benn1]. It is also difficult to generate a flat power spectrum at low frequencies as well as to produce delayed values of the white noise signal.

Consequently, it is necessary to use another test signal with an impulsive autocorrelation in order to determine the impulse response of a system using correlation techniques. The following discusses how the impulse response may be obtained using correlation techniques.

Given a system *p(t)* with input *u(t)* and output *y(t)*, the output *y(t)* can be determined using the convolution integral in Equation (3.4).

$$y(t) = \int_{-\infty}^{\infty} p(t)u(t-\lambda)d\lambda \tag{3.4}$$

The expression *p(t)* is sometimes referred to as the time response characteristic equation. Now, by performing a cross correlation between *u(t)* and *y(t)* using Equations (3.1) and (3.4), yields Equation (3.5).

$$\varphi_{uy}(\tau) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} u(t) \left[ \int_{-\infty}^{\infty} p(\lambda)u(t+\tau-\lambda)d\lambda \right] dt \tag{3.5}$$

Changing the order of integration results in Equation (3.6):

$$\varphi_{uy}(\tau) = \int_{-\infty}^{\infty} p(\lambda) \left[ \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} u(t)u(t+\tau-\lambda)dt \right] d\lambda \tag{3.6}$$

Note the appearance of the term

$$\left[ \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} u(t)u(t+\tau-\lambda)dt \right]$$

which, when compared to Equation (3.3) is the autocorrelation of *u(t)* as a function of delay *(t-λ)*, or $\varphi_{uu}(t-\lambda)$. Substituting this in Equation (3.6) yields Equation (3.7).

$$\varphi_{uy}(\tau) = \int_{-\infty}^{\infty} p(\lambda)\varphi_{uu}(t-\lambda)d\lambda \tag{3.7}$$

This takes the form of Equation (3.4), and it can be seen that $\varphi_{uy}(\tau)$, the cross correlation between the input and output of the system is numerically equal to the output that would result if $\varphi_{uu}(\tau)$ were input into the system in place of $u(t)$ [Benn1].

Therefore, if white noise were used as the input signal, $\varphi_{uu}(t-\lambda)$ reduces to $q$, where $q$ is the mean square value of the noise, the value of the impulse at $\tau=0$. Thus, Equation (3.8) results.

$$\varphi_{uy}(\tau) = qp(\tau) \tag{3.8}$$

In general, it is desirable to use an input with a simple autocorrelation function so that the right hand side of Equation (3.7) simplifies to an expression that can be easily manipulated. In the special case of an impulsive autocorrelation, the cross correlation between the system input and output is directly proportional to the time response characteristic equation $p(t)$, by the factor $q$. These considerations have led to the use of pseudorandom binary sequences as suitable test signals.

## 3.3 Pseudorandom Binary Sequences (PRBS)

PRBS signals may be generated by using linear feedback shift registers. The length of the sequence $L$ depends on the number of bits/stages $n$ of the shift register and the positions of the feedback paths. Figure 3-2 shows a four stage PRBS generator.

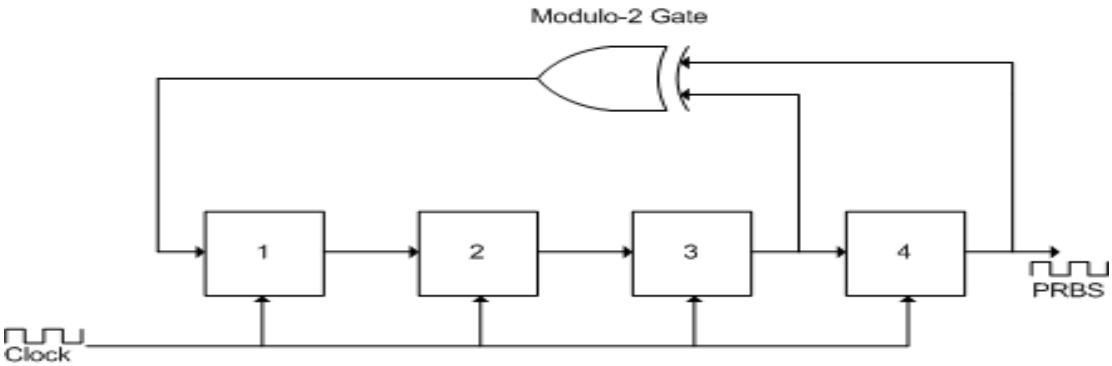



Figure 3-2 A four stage PRBS generator ($L=2^4-1=15$)

Looking at Figure 3-2, it can be seen that upon a clock transition, the information contained in each stage of the register is shifted by one stage to the right while the output of the modulo-2 gate is transferred to the first stage of the register. A modulo-2 gate performs the exclusive OR function, or XOR function. When generating a PRBS sequence, the input to the first register is taken from the output of a modulo-2 gate, which has as its inputs the

output of the last stage and the output of one other stage in the register. The choice of which intermediate stage this other "tapping" is taken from determines whether the sequence is maximal length PRBS. A maximal length PRBS sequence is one that has a length of $2^n-1$ bits. The PRBS generator in Figure 3-2 produces a signal of $2^4-1=15$ bits, with the all zero condition of the register being inadmissible.

With a constant shift register clock period of $T_B$ seconds and a sequence of length $L$, where $L=2^n-1$, the period of the sequence is $LT_B$ seconds, where $T_B$ is also known as the bit duration. If the binary levels of the PRBS sequence are assigned convenient positive and negative levels of equal amplitude ($\pm A$), the sequence is said to be bipolar and may appear as in Figure 3-3.

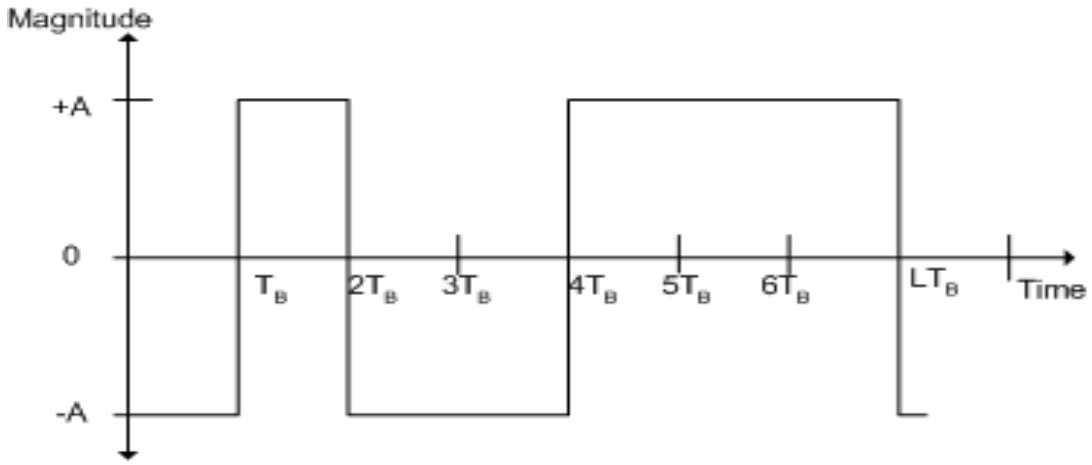



Figure 3-3 A bipolar PRBS signal with amplitude A, length L, and bit duration $T_B$

The autocorrelation of this type of amplitude PRBS signal takes the form shown in Figure 3-4.

From Figure 3-4, it can be seen that the autocorrelation function of a PRBS signal is periodic and, more importantly, impulsive. By making $L$ large and $T_B$ small, the autocorrelation will better approximate the "ideal" impulse function. The autocorrelation can be expressed as in Equation (3-9).

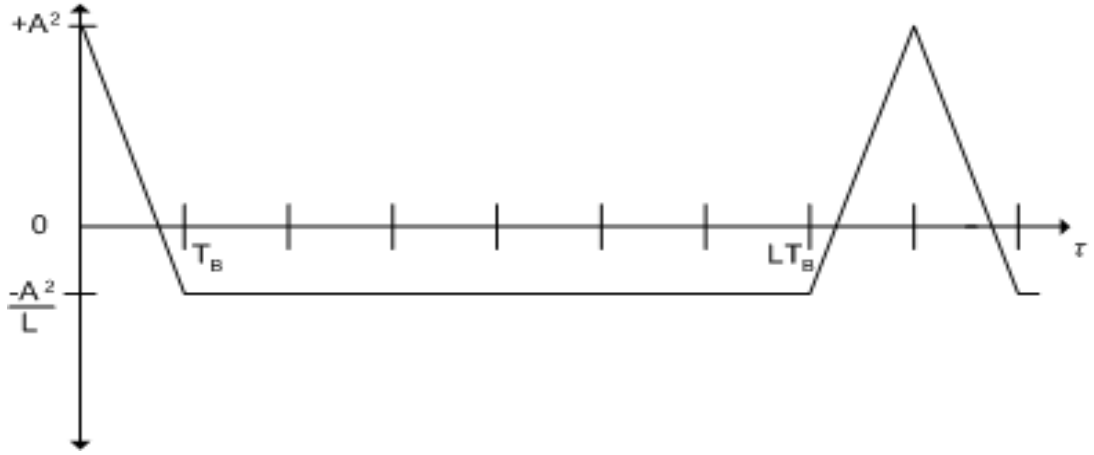



Figure 3-4 Autocorrelation of a PRBS signal

$$\varphi_{uu}(\tau)=\begin{cases}\dfrac{-A^2}{L}, \tau \neq nLT_B \\ +A^2, \tau = nLT_B\end{cases}; n=0,\pm1,\pm2,... \tag{3.9}$$

Due to the periodicity of the autocorrelation function, if the system input is a PRBS sequence it is only necessary to compute the cross correlation between the system's input and output over one period. Additionally, the binary nature of the PRBS signal makes the process of multiplication and delay generation (both required for computing a cross correlation), easy to implement on discrete/digital systems. In practice, the sequence length $LT_B$ must be greater than the significant part of the impulse response [Jackson1] and each bit of the signal must on average affect at least one bit of the sampled system output signal [Benn1].

It is necessary for $L=T_i/T_B$ to span the time interval of interest for the impulse response [Benn1]. Here, $T_i$ is the required time interval for the entire sequence. Since $L$ is determined by the number of stages $n$ in the shift register and $T_B$ is determined by the clock period, $T_i$ is usually chosen by using a suitable $L$ and $T_B$ combination. For large values of $L$, the PRBS signal will have less correlation with the system's normal working input and other noise. For smaller values of $T_B$, the PRBS-induced component of the system's output will be reduced.

One can effectively increase the period over which the cross correlation is computed, thus further removing uncorrelated noise, by repeating the PRBS sequence. The correlation is still computed over one period of the PRBS sequence, but with each iteration of the sequence the cross correlation is averaged with previous cross correlation computations. Any

correlation between the PRBS signal and the system's normal working input is mitigated by repeating the PRBS sequence.

While large values of *A* result in better impulse response curves, a trade-off needs to be made between how much the system can tolerate in terms of distortion from the PRBS-induced component and how clean the impulse response needs to be. Since this method ultimately requires perturbing the motor's torque control input with the PRBS signal, it is important to ensure the amplitude of the PRBS signal does not exceed the tolerable limit for the system, a factor which needs to be specified before testing begins.

## 3.4 Impulse Response

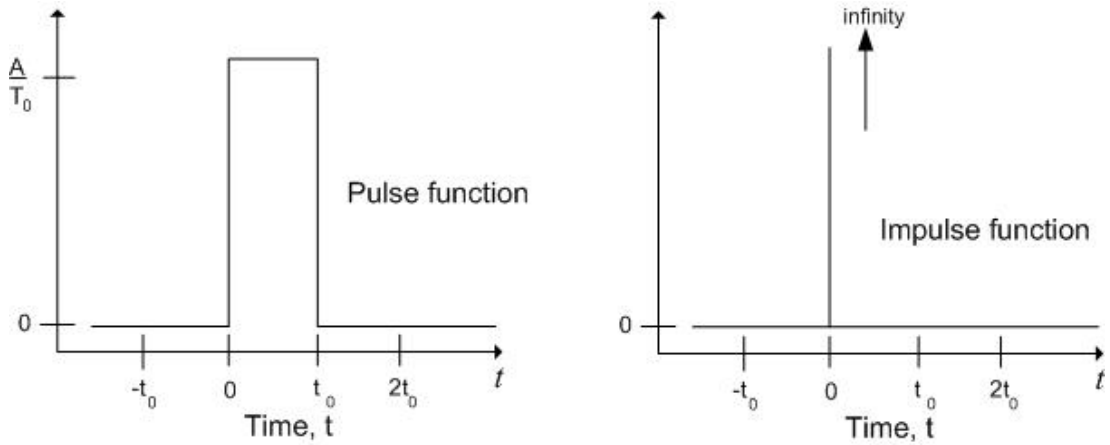



Figure 3-5 Pulse and Impulse functions

Given an impulse function, defined as in Equation (3.10)

$$h(t) = \begin{cases} 0, & t < 0 \\ \lim\limits_{t_0 \to 0} \dfrac{A}{t_0}, & 0 \le t \le t_0 \\ 0, & t_0 < t \end{cases} \tag{3.10}$$

The impulse function defined in Equation (3.10) has a height of $A/t_0$ and a duration of $t_0$. Thus, the area under the impulse is always equal to A. As the impulse duration $t_0$ approaches zero, the height of the impulse must approach infinity in order for a constant area to be

maintained. Comparing the impulse function to a pulse function enables one to determine the Laplace transform of the impulse function, using the Laplace transform of a pulse function.

The pulse function in Figure 3-5 may be defined as follows

$$d(t)=\begin{cases}0, & t<0\\ \dfrac{A}{t_0}, & 0\le t\le t_0\\ 0, & t_0<t\end{cases} \tag{3.11}$$

Analysis of the Laplace transform of *d(t)* is made simpler by considering the pulse function to be the superposition of a step function of height $A/t_0$ beginning at time $t$=0 and a second step function, this time negative, with a height also $A/t_0$ and beginning at time $t=t_0$. Thus,

$$d(t)=\frac{A}{t_0}\sigma(t)-\frac{A}{t_0}\sigma(t-t_0) \tag{3.12}$$

Here, $\sigma(t)$ represents the unit step function and is defined as,

$$\sigma(t)=\begin{cases}0, & t<0\\ 1, & t\ge 0\end{cases} \tag{3.13}$$

The Laplace transform of d(t) is shown as,

$$\begin{aligned}L[d(t)]&=L[\frac{A}{t_0}\sigma(t)]-L[\frac{A}{t_0}\sigma(t-t_0)]\\ &=\frac{A}{st_0}-\frac{A}{st_0}e^{-st_0}\end{aligned} \tag{3.14}$$

$$\therefore D(s)=\frac{A}{st_0}(1-e^{-st_0}) \tag{3.15}$$

By comparison, the Laplace transform of the impulse function is given by,

$$
\begin{aligned}
L[h(t)] &= \lim_{t_0 \to 0} L[d(t)] \\
&= \lim_{t_0 \to 0} \left[ \frac{A}{st_0}(1 - e^{-st_0}) \right] \\
&= \lim_{t_0 \to 0} \left[ \frac{\dfrac{d}{dt_0}\left[ A(1 - e^{-st_0}) \right]}{\dfrac{d}{dt_0}\left[ st_0 \right]} \right] \\
&= \frac{As}{s} = A
\end{aligned}
\tag{3.16}
$$

Equation (3.16) shows that the Laplace transform of an impulse response is equal to the area under the impulse. For the special case of a unit impulse response where $A$=1, the area is also unity. Thus, for a system *P(s)* excited with a unit impulse input, the Laplace transform of the output *Y(s)* of the system is

$$Y(s) = D(s)P(s) = P(s) \tag{3.17}$$

Now, taking the inverse Laplace transform on both sides of Equation (3.17) yields

$$
\begin{aligned}
&L^{-1}[Y(s)] = L^{-1}[P(s)] \\
&\therefore y(t) = p(t)
\end{aligned}
\tag{3.18}
$$

By noting that *y(t)* is actually the response of the system to a unit impulse (and is therefore called the impulse response function) it is shown in Equation (3.18) that the Laplace transform of *y(t)* gives the system transfer function *P(s)*. Consequently, the impulse response and the system transfer function contain the same information about the dynamics of the system, and it is thus possible to identify the system dynamics by measuring and analysing the impulse response of the system. It has been shown that the mathematical correlation between a system's input and output will produce the system's impulse response curve if the input has an impulsive autocorrelation. It has also been shown that PRBS signals have an impulsive autocorrelation, hence they are suitable signals for determining a system's impulse response and from this the system dynamics.

## 3.5 Effect of varying PRBS parameters on numerical impulse response

An investigation was undertaken into what effect varying each of the PRBS parameters *L* (length of sequence), *N* (number of repetitions), and *A* (amplitude), has on the numerical impulse response that results from the correlation.

In working toward a practical technique for determining the impulse response of a system using correlation techniques, simulations were performed using Matlab and its associated simulation software Simulink. Refer to Appendix B for relevant Matlab script files and Simulink models.

The system under investigation is defined by the transfer function *P(s)*, where

$$P(s) = \frac{10}{s+1} \tag{3.19}$$

The Simulink model used in the simulations is shown in Figure 3-6.

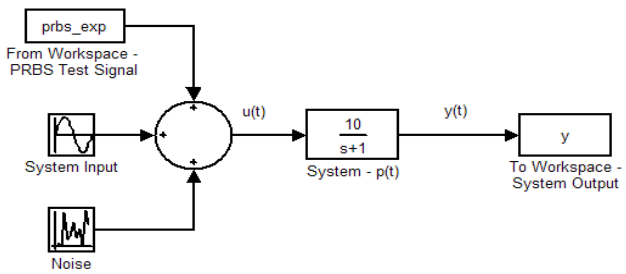



Figure 3-6 Simulink model used for simulation of PRBS-based tests to determine numerical impulse response

The traces in Figure 3-7 show the normal system working input, the PRBS input, the PRBS superimposed on the system input and the system output. The values used in this simulation were exaggerated in order to display clearly the system input with the superimposed PRBS component. When testing on a real system, the PRBS signal amplitude is limited so as to not cause such a noticeable effect.

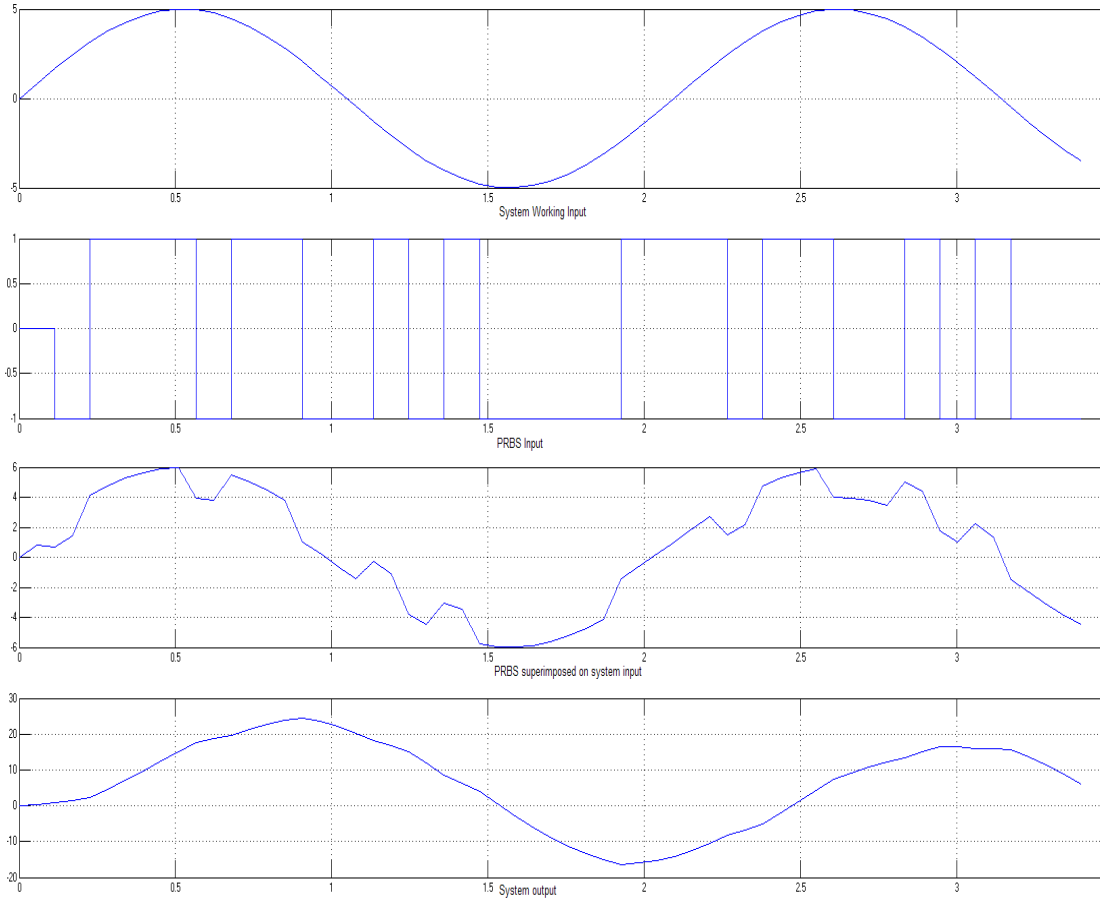



Figure 3-7 Oscilloscope traces for simulations.

The results for various values of *L* and *N* are shown in Figure 3-8 and Figure 3-9. For the following simulation results, all the amplitudes of the PRBS and random noise (where added) are given in percentage. This refers to the percentage amplitude of the system's normal working input.

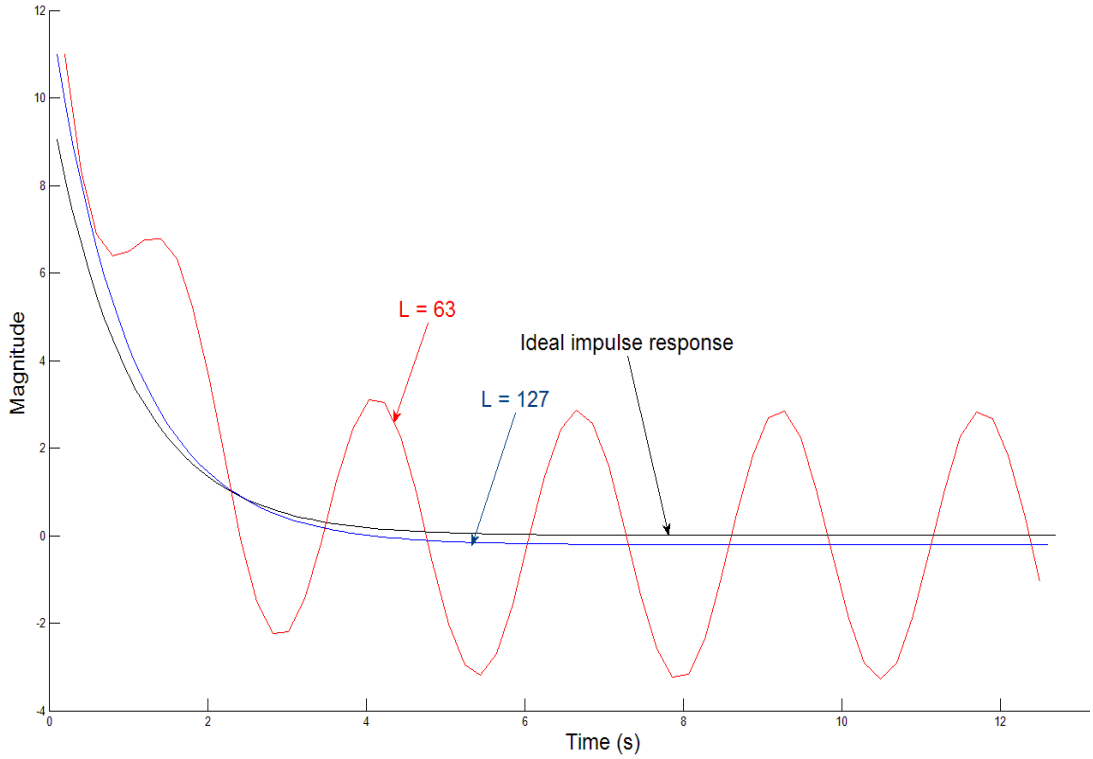



Figure 3-8 Effect as L is changed on impulse response curve

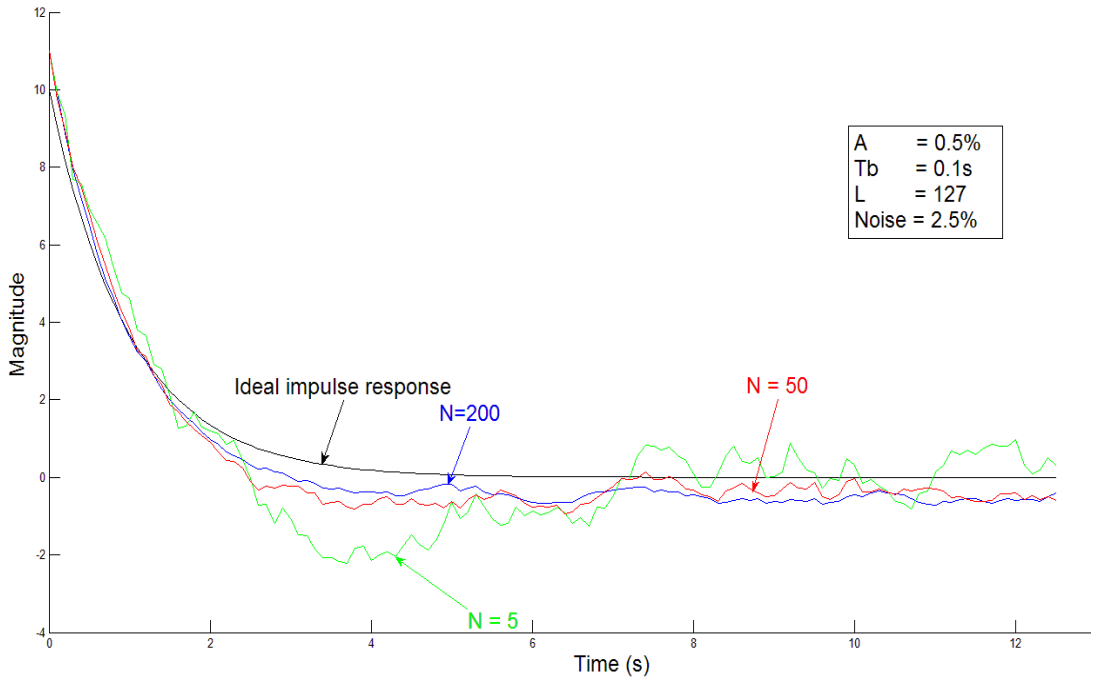



Figure 3-9 Effect of changing N on impulse response curve (L=127)

From Figure 3-8 it can be seen that as $L$ is increased, the numerical impulse response curve better approximates the theoretical curve. A similar phenomenon occurs as $N$ is increased, as seen in Figure 3-9. The disadvantage of increasing $L$ and/or $N$ is that it takes longer to capture the data as well as to compute the cross correlation.

By increasing $A$, the response also improves, as shown in Figure 3-10.

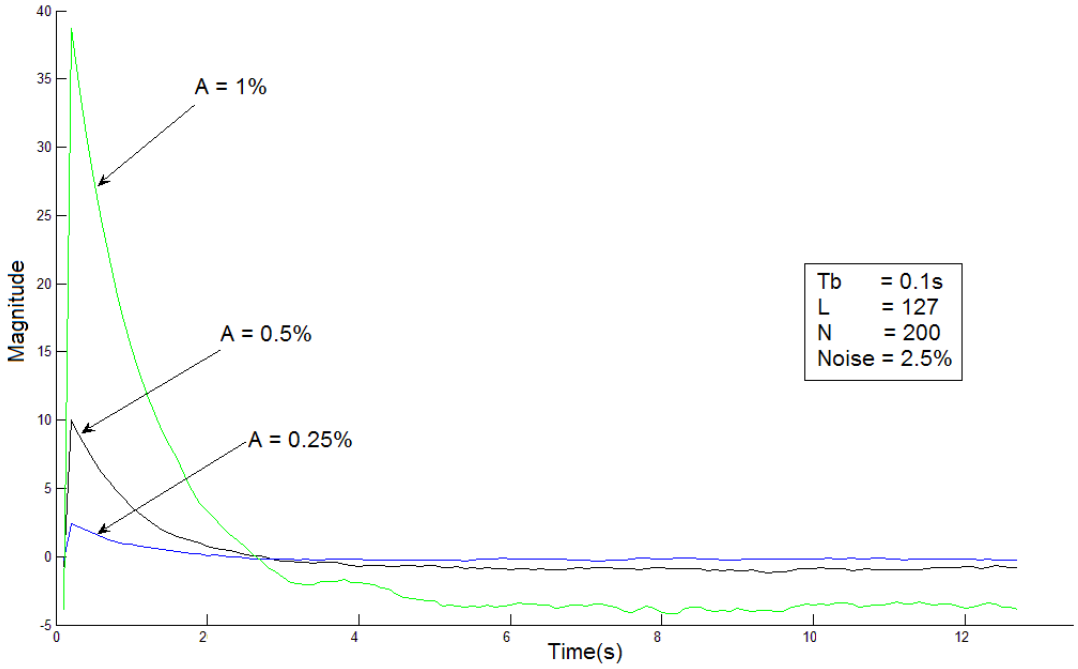



Figure 3-10 Effect of changing A, the PRBS amplitude.

Note that as $A$ is increased, the peak of the impulse response curve increases by a factor of $A^2$, since the peak of the autocorrelation of the PRBS signal increases by the same factor. It is therefore imperative that the impulse response that is calculated from the correlation be correctly scaled, especially if it is being used for parameter extraction.

## 3.6 Scaling the Numerical Impulse Response

Since the impulse response of interest in this method is that of the response to a unit impulse, it must be noted that the area under a unit impulse is one. However, from Figure 3-4 it can be seen that the area under the autocorrelation of a PRBS sequence is $A^2T_B$. Also, the PRBS sequence has an offset of $-A^2/L$ where the impulse function has zero value. This offset must be taken into account and added to the numerical impulse response. The numerical impulse

response must also be divided by the $A^2T_B$ factor in order to get an approximation of the system's actual impulse response. With regard to the offset factor, this is a particularly useful feature of the PRBS signal because it is possible to determine the DC gain of the system from the numerical impulse response, since the offset acts as a constant DC signal. The proof of this is shown later. After the necessary scaling is performed, curve fitting techniques may then be applied to the numerical impulse response curve in order to determine the system parameters.

One such curve fitting technique is the Linear Least Squares Method.

## 3.7 Linear Least Squares

The method of linear least squares involves fitting a vector of parameters (**θ**) to measured data in a matrix ***M***, such that:

$$y^{(n)} = M \bullet \theta \tag{3.20}$$

Given $y(t_k)$ = output values (with measurement noise) and $u(t_k)$ = input values for $k=1,2,\ldots,N$, it is possible to calculate the derivatives $y^{(i)}(t_k)$ and $u^{(i)}(t_k)$. For example, the first derivative may be determined as follows:

$$\left.\frac{dy}{dt}\right|_{t_k} \approx \frac{y(t_k) - y(t_{k-1})}{t_k - t_{k-1}} \tag{3.21}$$

By iterating as above, the following vector equation is attained:

$$\begin{aligned} & y^{(n)}(t_k) \\ & = (-y_k^{(n-1)}, -y_k^{(n-2)}, \mathrm{K}, -y_k, u_k^{(m)}, u_k^{(m-1)}, \mathrm{L}, u_k^{(0)}) \bullet (a_{n-1}, a_{n-2}, \mathrm{K}, a_0, b_m, b_{m-1}, \mathrm{K}, b_0)^T \end{aligned} \tag{3.22}$$

Here, if $a_n=1$, which is the conventional way of writing $P(s)$, we get,

$$y_k^{(n)} = M_k \bullet \theta \tag{3.23}$$

where, $y_k^{(n)} = y^{(n)}(t_k)$,

$M_k = (-y_k^{(n-1)}, -y_k^{(n-2)}, \mathrm{K}, -y_k, u_k^{(m)}, u_k^{(m-1)}, \Lambda, u_k^{(0)})$,

$\theta = (a_{n-1}, a_{n-2}, \mathrm{K}, a_0, b_m, b_{m-1}, \mathrm{K}, b_0)^T$

The error from the linear least squares approximation is given by

$$e = M\theta - y^{(n)} \tag{3.24}$$

The line that minimises the sum of the errors is considered to produce the most accurate fit to a set of data; hence in the method of linear least squares $\boldsymbol{\theta}$ is solved so as to minimise the squared error; or ($\boldsymbol{e^T e}$) must be minimised. This is achieved by letting:

$$\theta_{ls} = (M^T M)^{-1}(M^T y^{(n)}) \tag{3.25}$$

Important to note in this method is that the order of the linear least squares algorithm needs to be provided. The order of the algorithm should be chosen so as to minimise errors, such as noise and round-off errors, and errors from numerical integration or differentiation. At the same time, care must be taken not to choose so great an order that the linear least squares method provides a solution that fits the measured data, rather than the system.

The general order plant P(s) may be represented in the time domain by the following:

$$\sum_{i=0}^{n} a_i y^{(i)}(t) = \sum_{j=0}^{m} b_j u^{(j)}(t) \tag{3.26}$$

More specifically, a second order system may be written as:

$$\ddot{y} + a_1 \dot{y} + a_0 y = b_1 \dot{u} + b_0 u \tag{3.27}$$

Solving for *y(t)* requires integrating twice in this case; hence differentiation is avoided, since the presence of noise may create large errors when differentiating. For this reason, integration was preferred and, after integrating, and setting initial conditions to zero, the following results:

$$y(t) = (-\int_0^t y(\tau)d\tau, \quad -\int_0^t\int_0^\tau y(\tau_2)d\tau_2 d\tau_1, \quad \int_0^t u(\tau)d\tau, \quad \int_0^t\int_0^{\tau_1} u(\tau_2)d\tau_2 d\tau_1) \bullet (a_1, a_0, b_1, b_0)^T \tag{3.28}$$

Thus, $\mathbf{M} = [-\int_0^t y(\tau)d\tau, \quad -\int_0^t\int_0^\tau y(\tau_2)d\tau_2 d\tau_1, \quad \int_0^t u(\tau)d\tau, \quad \int_0^t\int_0^{\tau_1} u(\tau_2)d\tau_2 d\tau_1]$

$\boldsymbol{\theta} = (a_1, a_0, b_1, b_0)^T$

Likewise, for a first order system ($a_n=a_1=1$) it can be shown that

$$y(t) = \left( -\int_0^t y(\tau)d\tau, \int_0^t u(\tau)d\tau \right) \bullet (a_0, b_0)^T \tag{3.29}$$

$$\mathbf{M} = \left( -\int_0^t y(\tau)d\tau, \int_0^t u(\tau)d\tau \right)$$

$$\boldsymbol{\theta} = (a_0, b_0)^T$$

Using the above analytical technique, and given a set of input-output data for a system, it is possible to fit a curve to the data that minimises the sum of the squares of the errors between the actual data and a model-based data set. Since the method generates a model-based data set, it needs to be provided with the model order of the system. Models for extracting the mechanical parameters of a system were derived in Section 2.3.

Algorithms were written in Matlab to extract parameters from first and second-order systems, since these were the orders of the systems of interest to this research. Integration of the input *u(t)* is required in the analysis involving the method of linear least squares. In order to simplify the integration, the input of the system (for the least squares curve) was taken to be a step input, which is defined as in Equation (3.13). However, the measured values were for the system's impulse response since this is the resultant response for a PRBS-based test. In order to compare the two responses and analyse the data so as to produce a least squares fit, the step response of the system under investigation was required. Using the property that the impulse response curve can be determined by differentiating the step response curve, in this case the numerical impulse response was integrated to produce the step response. This step response is then analysed by the least squares algorithm to produce a best-fit curve and, from the $\boldsymbol{\theta}$ vector, the system's transfer function parameters.

For the plant defined by Equation (3.19), the numerical impulse response underwent a least squares identification with the resultant fitted curve shown in Figure 3-11. It is observable that this method when used in conjunction with the PRBS-based identification method produces more accurate results when compared to the numerical impulse response . Refer to Table 3-1 for the percentage error in the approximation.

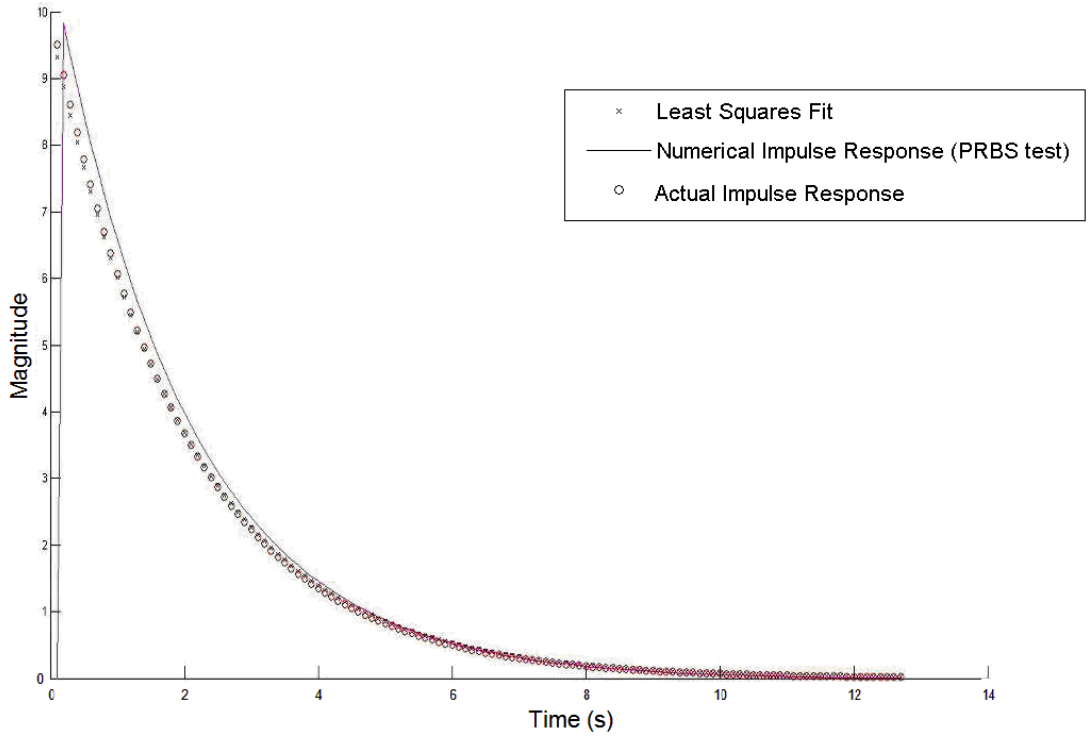



Figure 3-11 Least Squares fit to numerical impulse response and compared to actual system impulse response

## 3.8 Determining the Parameters from the Numerical Impulse Response

For a first order system, it is possible to read the parameters of interest directly off the impulse response curve. Given a first order system with transfer function *G(s)* in the generalised form

$$G(s) = \frac{k}{s\tau + 1} \tag{3.30}$$

The response to a unit impulse is given by

$$g(t) = \frac{k}{\tau} e^{\frac{-t}{\tau}} \tag{3.31}$$

where k = DC gain of the system

$\tau$ = time constant of the system

The value ($k/\tau$) is the peak of the impulse response. Furthermore, as mentioned earlier, numerical impulse response curves determined using a PRBS-based identification method have an offset, i.e. the impulse response settles to some negative value rather than zero. This is due to the negative offset in the PRBS sequence itself. Also, this offset has been shown to act as a DC injected signal. Therefore, this can be used to determine the DC gain of the system as follows.

$$\begin{aligned} \text{Final Offset Value} &= \text{Input Signal Offset} * \text{DC Gain} \\ \therefore \text{DC Gain} &= \frac{\text{Final Offset Value}}{\text{Input Signal Offset}} \\ \therefore \text{k} &= \frac{\text{Final Offset Value}}{\left(\frac{\text{A}^2}{\text{L}}\right)} \end{aligned} \tag{3.32}$$

Since this research focuses on the mechanical system parameters, the transfer function from torque to speed is of particular interest. It is shown in Equation (2.13) in Section 2.3.6 that this transfer function is the first order transfer function

$$T_{\omega/\tau^*} \approx \frac{1}{sJ_m + B_m} = \frac{\frac{1}{B_m}}{s\left(\frac{J_m}{B_m}\right)+1} \tag{3.33}$$

From this, and comparing to Equations (3.30), (3.31) and (3.33) it can be seen that the peak of the impulse response curve gives

$$\begin{aligned} peak &= \frac{k}{\tau} = \frac{\frac{1}{B_m}}{\frac{J_m}{B_m}} = \frac{1}{J_m} \\ \Rightarrow J_m &= \frac{1}{peak} \end{aligned} \tag{3.34}$$

Also, using the method described by Equation (3.32),

$$\frac{1}{B_m} = \frac{\text{Final Offset Value}}{\frac{A^2}{L}}$$
$$\Rightarrow B_m = \frac{\frac{A^2}{L}}{\text{Final Offset Value}} \tag{3.35}$$

Figure 3-12 shows that this method of reading the values off the scaled numerical impulse response should provide a fairly accurate estimation of parameters, since the scaled numerical impulse response fits the actual impulse response quite well.

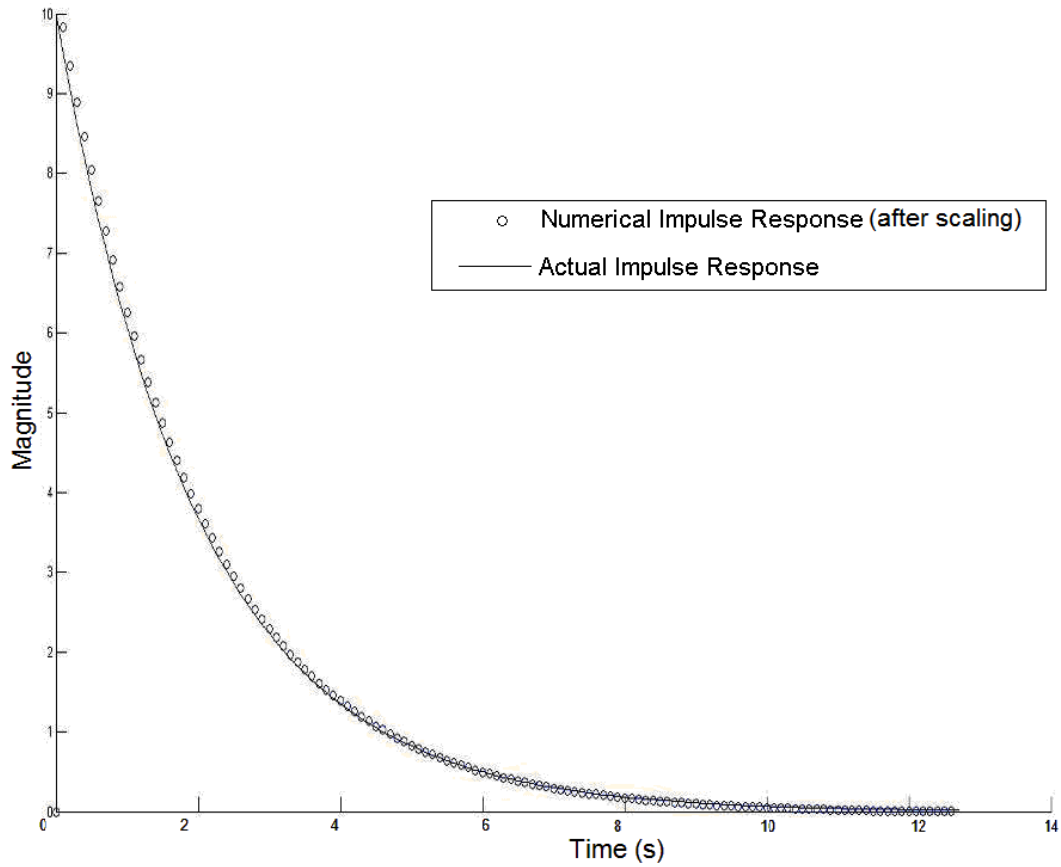



Figure 3-12 Plot showing the numerical impulse response after relevant scaling has been applied, as compared to actual impulse response

However, the DC gain cannot be determined using the scaled numerical impulse since, as may be observed from Figure 3-12, there is no offset once the response is properly scaled and adjusted. Therefore, *before* the response curve is adjusted to cancel out the offset, as in the case of Figure 3-13, the DC gain must be determined.

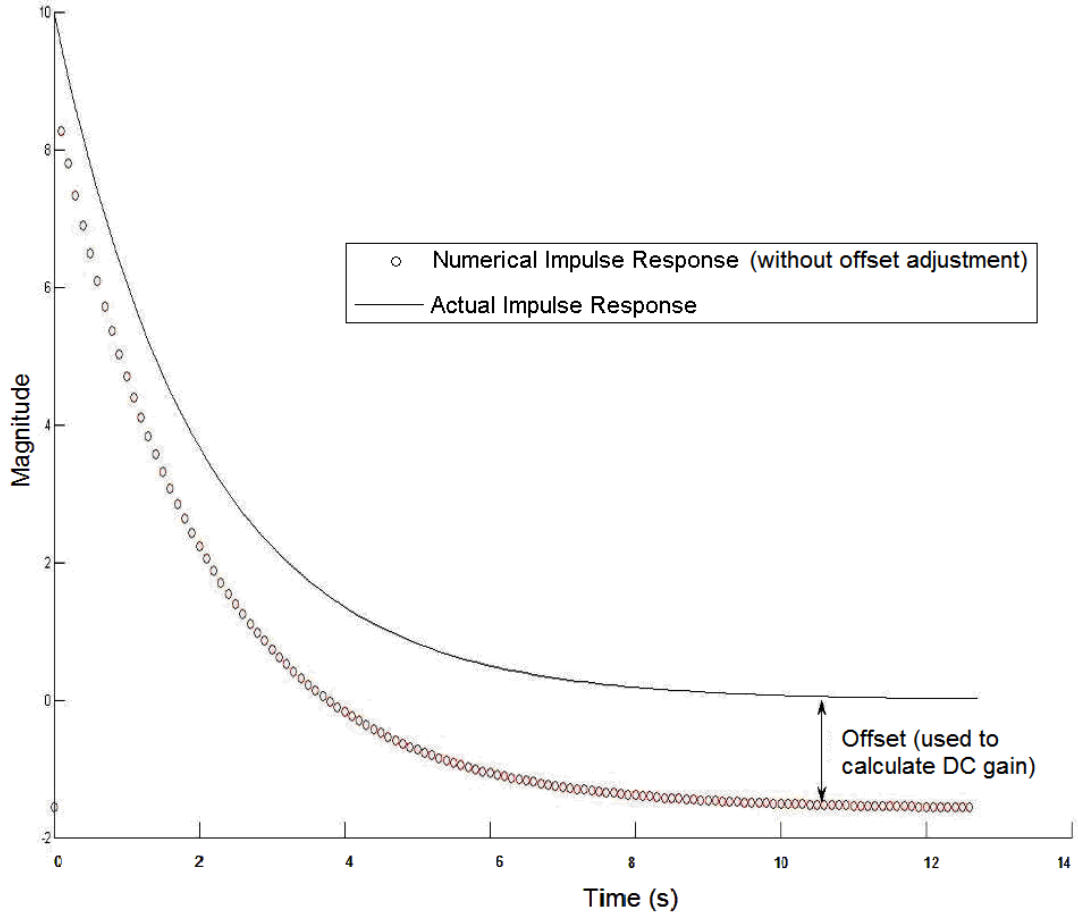



Figure 3-13 Plot showing the numerical impulse response before the offset has been removed, as compared to the actual impulse response

These two methods of parameter estimation, namely linear least squares and parameter identification from the peak and offset in the numerical impulse response, were used to identify the parameters of the system defined by Equation (3.19). Note, too, that by comparing Equation (3.23) to Equation (3.19), $J=0.1$, $B=0.1$. The system is assumed to be a simple mechanical system, therefore the experiment is said to determine the $J$ and $B$ parameters, in this case for a system that has $J=B$. The results are summarised in Table 3-1.

Table 3-1 Comparison between two methods of parameter estimation

| Parameter Estimation Method | % Error in J | % Error in B |
|---|---|---|
| From the curve | 1.7 | 1.2 |
| Linear Least Squares | 2.2 | 0.4 |

From the results in Table 3-1, it can be seen that the error in the estimation for *J* is greater when using the linear least squares estimation method. However, the preferred method is that of least squares estimation, since this method attempts to mitigate to some extent the effect of noise in the measured data, which will be quite noticeable when data from real systems is measured.

## 3.9 Summary

This chapter introduced the correlation-based method of system identification using an input signal called a PRBS signal. The properties of a PRBS sequence were discussed and it was shown why a PRBS signal can be used to determine the numerical impulse response of a system. It was also shown what effect varying the parameters of the PRBS signal has on the numerical impulse response.

Once the numerical impulse response has been determined by correlation, it was shown that scaling was necessary in order to approximate the system's response to a unit impulse. After scaling, two methods of parameter estimation were proposed and the results from each method were compared. It was decided that future parameter extraction would be performed with the method of linear least squares.

# CHAPTER 4
# SIMULATED IMPLEMENTATION

## 4.1 Introduction

It has been shown in Chapter 3 that the PRBS-based tests do indeed provide reasonable results for parameter identification. This chapter extends the work demonstrated in the previous chapter by looking at the simulated response of a DC motor model to a PRBS signal applied to its inputs. An investigation as to whether applying the PRBS signal as a torque or speed perturbation is undertaken, and the results shown. The simple DC motor model is then extended to include a load, with the aim of being able to perform parameter estimation with this additional complexity. Research into other mechanical effects such as elasticity and backlash is also demonstrated.

This chapter deals with the simulated implementation of the abovementioned tests, while the following chapter will contain the results of a practical implementation. Following on from the simulation work shown in the previous chapter, Simulink was used for modelling and simulation. Script files and Simulink models have been included in Appendix A.

## 4.2 PRBS as a Torque Perturbation vs. PRBS as a Speed Perturbation

With the enhanced functionality of most modern drives, one has the choice of introducing a test signal almost anywhere in the control system. The decision thus needs to be taken as to whether it is preferable to introduce the PRBS signal as a torque perturbation or as a speed perturbation. Equation (2.13) in Section 2.3.6 showed that, for an infinite bandwidth torque controller, if the PRBS input signal is applied as a torque perturbation with machine speed being the output measured and with the speed loop open, the transfer function thus identified is approximately $\frac{1}{sJ+B}$. It was also stated that degradation of the parameter estimation is expected in a practical implementation, due to the finite bandwidth of the torque controller.

However, it may not always be possible to introduce a torque perturbation on an online system, especially so with a closed loop speed controlled application, and so an attempt is made to estimate the mechanical parameters when the PRBS signal is applied either to the torque control input or to the speed control input of a drive system, with the speed loop closed. From Equations (2.12) and (2.15) it is expected that the system identified when applying the PRBS input signal to the torque or to the speed control input with the speed

loop closed will include the effects of the speed controller, and these need to be compensated for when estimating the parameters of the mechanical system. This means that one needs to know the parameters of the speed controller $G_S$ if these tests are performed on a closed loop speed-controlled system.

## 4.3 DC Motor With No Load

The sequence of performing the simulations and subsequent analyses is outlined below.

The values for *A, $T_B$, L, and N* were entered into the "prbs_gen_export_simulink.m" file. The values for the motor constants $k_e$ and $k_t$, as well as inductance $L_m$ and resistance $R_m$ were chosen at the outset and kept constant for all simulations, to make the relevant comparisons of the motor inertia $j_m$, motor viscous friction $b_m$, load inertia $j_L$, and load viscous friction $b_L$ were entered into the "prbs_gen_export_simulink.m" file. These parameters were imported into the Simulink model which was then run.

When the simulation completed, another Matlab file was used to perform the cross correlation, as well as the necessary scaling of the correlation, that is, to divide by $A^2*T_B$, as well as to add the offset that is introduced by the negative offset in the PRBS autocorrelation. While Simulink includes cross correlation blocks, the data was exported since it was desirable to write a custom cross correlation algorithm in Matlab. This algorithm was written in such a manner that it could be easily converted into PLC code, the objective of this exercise.

Applying the method explained in Section 3.8, parameter estimation was performed using the numerical impulse response, that is, the impulse response achieved from the cross correlation.

A linear least squares algorithm then processed the data from the numerical impulse response to provide an estimated transfer function, which was compared to the transfer function of the mechanical system to provide estimates for $J_m$ and $B_m$.

The ideal impulse response, numerical impulse response, and impulse response of the transfer function estimated by the linear least squares algorithm were then all plotted to verify the accuracy of the two methods (numerical impulse response and linear least squares).

### 4.3.1 PRBS as a torque perturbation, with speed loop open

As a starting point, simulations were performed for a DC motor with no load, with only the motor inertia and motor viscous friction to identify.

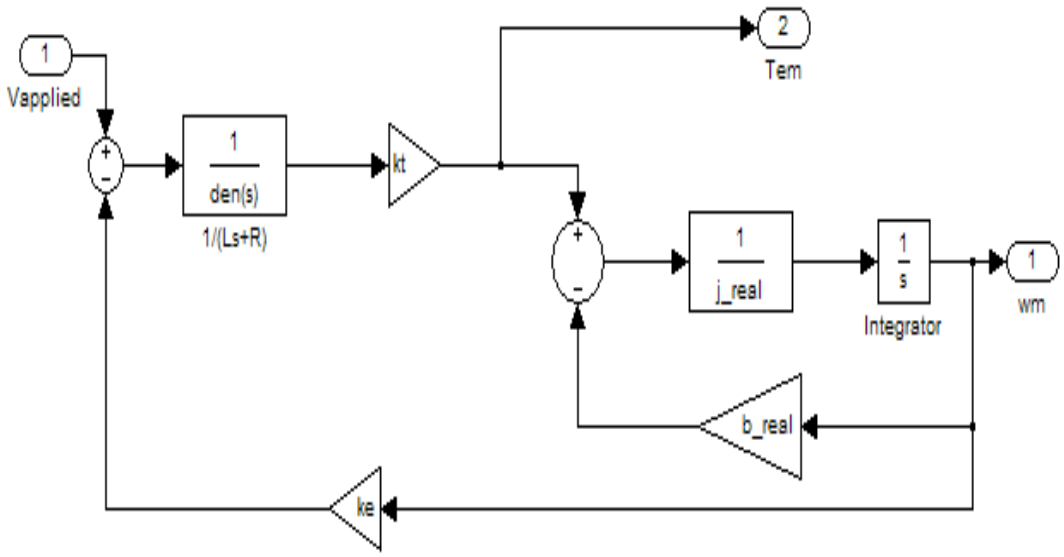



Figure 4-1 Simulink model of DC motor with no load

The model shown in Figure 4-1 is the Simulink equivalent of the block diagram in Figure 2-3. This model was then embedded in the subsystem (DC Motor Model) shown in Figure 4-2.

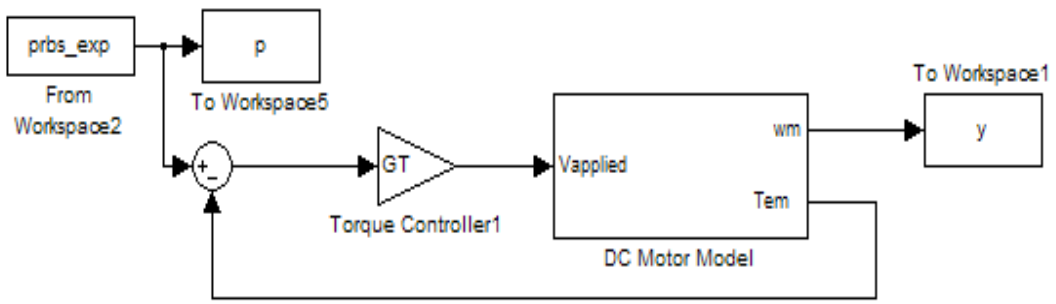



Figure 4-2 Simulink model incorporating DC motor model, with PRBS perturbation applied as a torque reference, and speed loop open

The simulated speed output $\omega_m$ was output to the workspace, after which it was correlated with the PRBS input, which was applied to the torque control input of the torque controller $G_T$. The scaling techniques discussed in Section 3.8 as well as the method of linear least

squares were applied to the correlation result to yield an impulse response curve which, according to Equation 2.13, may be used to determine parameter estimates for $J_m$ and $B_m$.

Figure 4-3 shows numerical impulse responses with varying values of $T_B$, and compares these against the ideal impulse response for the transfer function with the *known* values for $J_m$ and $B_m$, as determined using the *impulse* function in Matlab.

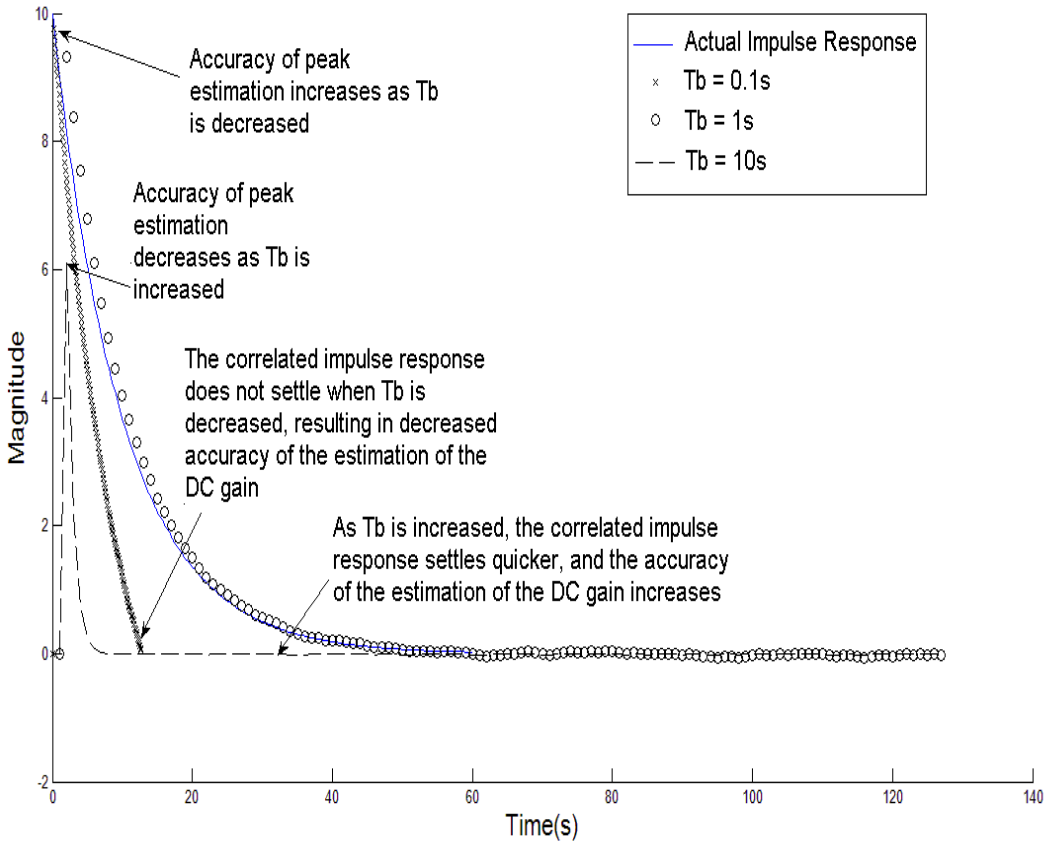



Figure 4-3 Comparison of numerical impulse responses as $T_B$ is varied

Simulations were performed on models with various values of $J_m$ and $B_m$. The results of two of these tests are shown in Table 4-1 and Table 4-2.

Table 4-1 Summary of accuracy of parameter estimations for DC motor model ($J_m$=0.01, $B_m$=0.01), with varying values of $T_B$

| $T_b$ | Numerical Impulse Response | | Linear Least Squares | |
|---|---|---|---|---|
| | $J_m$ error (%) | $B_m$ error (%) | $J_m$ error (%) | $B_m$ error (%) |
| 0.01 | 1.7 | >100 | 4.9 | 96.7 |
| 0.1 | 8.9 | -19.7 | 15.7 | 83.24 |
| 1 | 66.5 | -12 | >1000 | 60.5 |
| 10 | >100 | 5.3 | >1000 | 0.7 |

Table 4-2 Summary of accuracy of parameter estimations for DC motor model ($J_m$=0.1, $B_m$=0.01), with varying values of $T_B$

| $T_b$ | Numerical Impulse Response | | Linear Least Squares | |
|---|---|---|---|---|
| | $J_m$ error (%) | $B_m$ error (%) | $J_m$ error (%) | $B_m$ error (%) |
| 0.01 | 2.5 | >1000 | 3.7 | >1000 |
| 0.1 | 3.0 | >100 | 5.5 | 81.4 |
| 1 | 7.3 | 4.3 | 8.8 | 8.4 |
| 10 | 66.5 | 3.1 | 58.8 | 5 |

From Table 4-1 and Table 4-2 it may be seen that, in these simulated tests, the linear least squares approximation yields a less accurate estimation of parameters when compared to the estimation using the numerical impulse response. When approximating $B_m$, the results shown indicate that this parameter is more accurately determined using the numerical impulse response. Section 3.8 discussed how the parameters $J_m$ and $B_m$ may be determined from the numerical impulse response.

Although it appears that the method of linear least squares is less accurate than using the numerical impulse response, when tests are performed on real machine systems the method of linear least squares is critically important in providing accurate parameter estimations in the presence of measurement noise, as well as other non-linear effects, such as backlash and elasticity.

Also, when looking at the results in Tables 4-1 and 4-2 as well as Figure 4-3, it may be seen that when $T_B$ decreases, the accuracy of the estimation of $J_m$ increases while that of $B_m$ decreases; and when $T_B$ is increased, the accuracy of the estimation of $J_m$ decreases while that of $B_m$ increases.

This may be attributed to the fact that for a large value of $T_B$, the large bit period of the PRBS input signal fails to excite the system sufficiently to extract the mathematical impulse response of the system. While a large value of $T_B$ fails to produce a numerical impulse response that provides an accurate estimation of $J_m$, the large bit period of the input PRBS signal does excite the system under test for sufficiently long intervals of time to obtain a more accurate estimation of the DC gain, and subsequently $B_m$, when compared to smaller bit periods.

So far, the tests have been performed by applying the PRBS perturbations to the input of the torque controller. However, this is not always practical, particularly when the intent is to perform online tests on a closed loop speed-controlled application. The next section investigates the results attained when simulating the application of the PRBS signal to such a closed loop system.

### 4.3.2 PRBS on a System with Closed Loop Speed Control

In a closed loop speed-controlled application, applying the PRBS technique to estimate the mechanical parameters requires knowledge of the parameters of the speed controller. Equations (2.12) and (2.15) show that both the transfer function from reference speed to

speed output ($T_{\omega/\omega^*}$) and the transfer function from additional torque reference to speed output ($T_{\omega/\tau^*}$), respectively, contain terms dependent on the speed controller $G_S$. This is discussed in further detail in Section 4.3.2.1 and Section 4.3.2.2.

Therefore, it is imperative that the parameter extraction must compensate for the additional terms introduced by the speed controller. For the purpose of simplicity, the speed controllers used in this research were configured as pure gain, or proportional, controllers. Needless to say, most speed controllers in industry take the form of more complex controllers, such as proportional-integral or proportional-integral-derivative controllers, and compensating for the additional effects that such controllers introduce to the numerical impulse response needs to be considered. For now, this is left as an aspect that may be considered in further research.

For the case of a proportional controller with a gain of $G_S$, two potential testing methodologies are proposed for a system under closed loop speed control: the first is to introduce the PRBS perturbation as an additional input at the *torque* controller's input summing junction (this corresponds with the input $\tau_{add}$ in Figure 4-4); while the second method is to introduce the PRBS perturbation as an additional input at the *speed* controller's summing junction, in Figure 4-7 this would be at the point labelled $\omega^*$. An investigation into the corresponding transfer functions that would be identified by using these two methods was completed in Section 2.3.5 and 2.3.7, and the results are repeated here for convenience.

#### 4.3.2.1 PRBS added as a torque perturbation

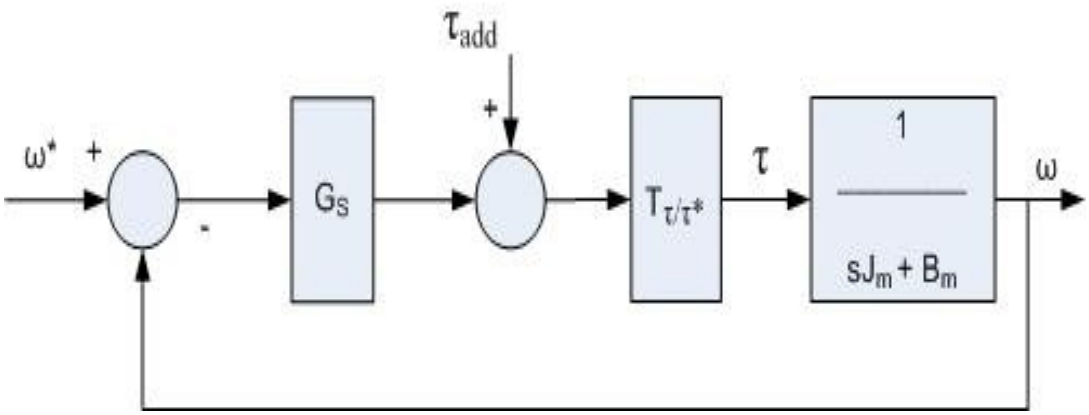



Figure 4-4 Block diagram model of mechanical system with speed controller ($G_S$)

In this case, the transfer function that one would identify by adding the PRBS as a perturbation on the torque loop (at the input labelled $\tau_{add}$) while the speed loop is closed, is given by:

$$\frac{\omega}{\tau_{add}} = \frac{1}{sJ_m + B_m + G_S} = \frac{\left(\frac{1}{B_m + G_S}\right)}{s\left(\frac{J_m}{B_m + G_S}\right) + 1} \tag{4.1}$$

Immediately it is seen that, as expected, the speed controller parameters are part of the system response. From the transfer function it is observed that the time constant of this system now features the $G_S$ term. In practice this translates to a much faster system response since most well-tuned speed controllers have a proportional term greater than 1. While this is not always be the case, this was the simplifying assumption taken for further tests. With a decreased time constant, it is necessary to decrease $T_B$ for viable results. Furthermore, when estimating parameters from the linear least squares approximation, the terms introduced by the speed controller must be removed before estimating $J_m$ and $B_m$. In Equation (3.31), a transfer function for a first order system was given as,

$$G(s) = \frac{k}{s\tau + 1}$$

where k = DC gain of the system

τ = time constant of the system

It was also shown in Equation (3.34) that the peak of the impulse response curve can be used to determine the term $k/\tau$ as follows,

$$peak = \frac{k}{\tau} = \frac{\frac{1}{B_m}}{\frac{J_m}{B_m}} = \frac{1}{J_m}$$

$$\Rightarrow J_m = \frac{1}{peak}$$

Applying the above method to Equation (4.1),

$$peak = \frac{k}{\tau} = \frac{\frac{1}{B_m + G_S}}{\frac{J_m}{B_m + G_S}} = \frac{1}{J_m}$$
$$\Rightarrow J_m = \frac{1}{peak} \tag{4.2}$$

The result in Equation (4.2) shows that when the PRBS signal is added as a torque perturbation with the speed loop closed, the identification of the inertia $J_m$ is independent of the speed controller $G_S$. However, the same is not true for the identification of $B_m$. Equation (3.35) is shown below,

$$k = \frac{1}{B_m} = \frac{\text{Final Offset Value}}{\frac{A^2}{L}}$$
$$\Rightarrow B_m = \frac{\frac{A^2}{L}}{\text{Final Offset Value}}$$

In this case, as shown by Equation (4.1), the DC gain term $k$ includes the speed controller term $G_S$,

$$k = \frac{1}{B_m + G_S} = \frac{\text{Final Offset Value}}{\frac{A^2}{L}}$$
$$\Rightarrow B_m = \frac{\frac{A^2}{L}}{\text{Final Offset Value}} - G_S \tag{4.3}$$

Therefore, in order to estimate parameter $B_m$, the speed controller term $G_S$ needs to be known, and compensated for by performing the subtraction shown in Equation (4.3).

Similarly, when using the method of linear least squares to estimate a transfer function that fits the impulse response from the correlation result, it is important to note that the estimated transfer function corresponds to the transfer function in Equation (4.1).

As shown in Equation (3.29), the method of linear least squares generates a parameter estimate vector $\boldsymbol{\theta}$, where

$$\boldsymbol{\theta} = (a_0, b_0)^T$$

The coefficients $a_0$ and $b_0$ may be written in the transfer function form, for a first order transfer function, as follows,

$$G(s) = \frac{b_0}{s + a_0} \tag{4.4}$$

Comparing Equation (4.4) with Equation (4.1),

$$\begin{aligned} \frac{\omega}{\tau_{add}} &= \frac{1}{sJ_m + B_m + G_S} \\ &= \frac{\left(\frac{1}{J_m}\right)}{s + \left(\frac{B_m + G_S}{J_m}\right)} \\ &= \frac{b_0}{s + a_0} \\ \therefore b_0 &= \left(\frac{1}{J_m}\right), a_0 = \left(\frac{B_m + G_S}{J_m}\right) \\ \therefore J_m &= \left(\frac{1}{b_0}\right), B_m = \left(\frac{a_0}{b_0}\right) - G_S \end{aligned} \tag{4.5}$$

It can be seen from Equation (4.5) that with the method of linear least squares, the speed controller term $G_S$ needs to be subtracted in the estimation of $B_m$ but is not needed for the estimation of $J_m$. This also shows that applying the PRBS signal as a torque perturbation to an open loop or closed loop system does not affect the peak of the impulse response. However, $G_S$ is inversely proportional to the time constant, as Equation (4.2) shows.

This is confirmed in Figure 4-5 where the time constant of the numerical impulse response is seen to decrease as $G_S$ is increased.

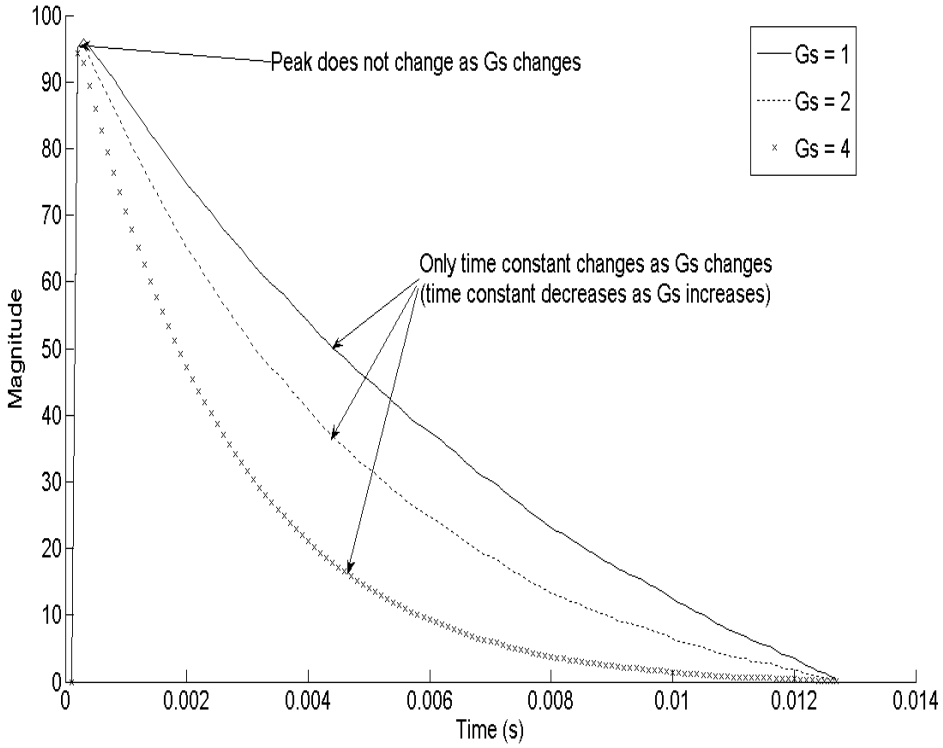



Figure 4-5 Comparing numerical impulse response for different values of $G_S$ (PRBS applied as a torque perturbation with speed loop closed)

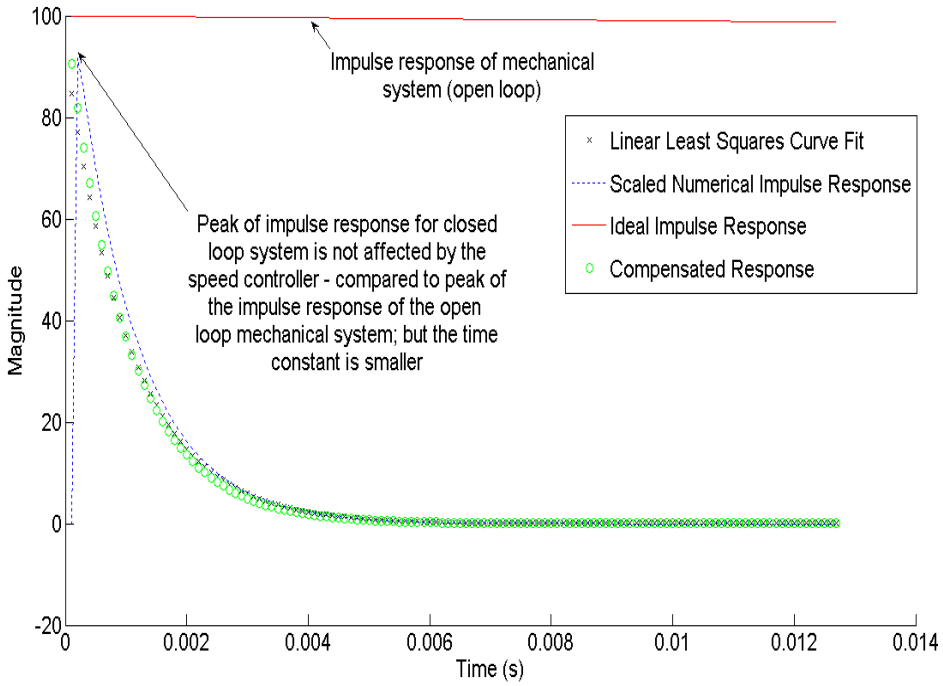



Figure 4-6 Simulated response with the PRBS signal applied as a torque perturbation, with the speed loop closed

With the PRBS signal applied as a torque perturbation and the speed output used as the system output, with the speed loop closed, the numerical impulse response that the test yields is for the system with the transfer function from Equation (4.1). This system includes the speed controller term, $G_S$.

Figure 4-6 clearly shows the difference between the open loop mechanical system response (the solid line marked *ideal impulse response*) and the numerical impulse response, due to the effect of the speed controller; the impulse response is seen to have a far smaller time constant due to the speed controller term.

Table 4-3 Summary of accuracy of parameter estimations for DC motor model with speed loop closed and PRBS applied as a torque perturbation ($J_m$=0.01, $B_m$=0.01), with varying values of $T_B$

| $T_b$ | Numerical Impulse Response | | Linear Least Squares | |
|---|---|---|---|---|
| | $J_m$ error (%) | $B_m$ error (%) | $J_m$ error (%) | $B_m$ error (%) |
| $1x10^{-5}$ | 3.7 | >10000 | -2.9 | >10000 |
| $1x10^{-4}$ | 9.2 | >1000 | 8.6 | >1000 |
| $1x10^{-3}$ | 59.6 | >1000 | 49.8 | >1000 |

An area of concern was the large errors in estimation of the $B_m$ term, even with increasing $T_B$. It was then found that by increasing $N$, the number of repetitions of the PRBS input, the accuracy of the estimation of $B_m$ increased. It was noted in Chapter 3 that a large value for $N$ is required to reduce the effects of noise, and it must also be noted here that increasing the number of repetitions of the PRBS sequence produced more accurate parameter estimation for $B_m$. This is shown in Table 4-4.

Table 4-4 Summary of accuracy of parameter estimations for increasing number of PRBS sequence repetitions ($J_m$=0.01, $B_m$=0.01, $T_B$=0.001)

| N | $J_m$ error (%) | $B_m$ error (%) |
|---|---|---|
| 10 | 59.6 | 1139 |
| 100 | 59.4 | 109.6 |
| 1000 | 59.3 | 5.5 |

These results show that while increasing $N$ does not affect the approximation for $J_m$, it provides a far better estimation for $B_m$.

#### 4.3.2.2 PRBS added as a speed perturbation

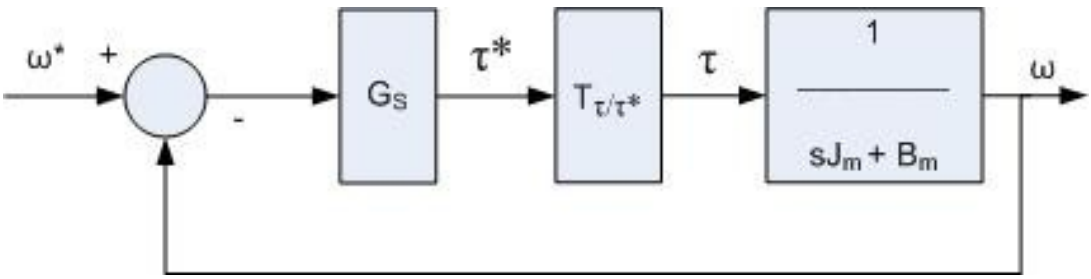



Figure 4-7 Block model for DC motor under closed loop speed control

In Figure 4-7, the block diagram for a system under closed loop speed control is displayed. In this figure, the block labelled $G_S$ represents the speed controller, while $T_{\tau/\tau^*}$ is the closed loop torque control system with torque controller $G_T$. For a well-tuned torque controller, $T_{\tau/\tau^*} \approx 1$.

The transfer function from speed reference $\omega^*$ to speed $\omega$ may be determined as follows,

$$\begin{aligned}\frac{\omega}{\omega^*} &= \frac{G_S T_{\tau/\tau^*}\left[\frac{1}{sJ_m + B_m}\right]}{1 + G_S T_{\tau/\tau^*}\left[\frac{1}{sJ_m + B_m}\right]} \\ &\approx \frac{G_S\left[\frac{1}{sJ_m + B_m}\right]}{1 + G_S\left[\frac{1}{sJ_m + B_m}\right]} \\ &= \frac{G_S}{sJ_m + B_m + G_S} = \frac{\left(\frac{G_S}{B_m + G_S}\right)}{s\left(\frac{J_m}{B_m + G_S}\right) + 1}\end{aligned} \tag{4.6}$$

Equation (4.6) thus shows that the transfer function that one would identify by adding the PRBS as a perturbation on the speed loop (at the input labelled $\omega^*$) while the speed loop is closed, is given by:

$$\frac{\omega}{\omega^*} = \frac{G_S}{sJ_m + B_m + G_S} = \frac{\left(\frac{G_S}{B_m + G_S}\right)}{s\left(\frac{J_m}{B_m + G_S}\right) + 1}$$

As with the case of applying the PRBS as a torque perturbation, when the PRBS is added as a speed perturbation with the speed loop closed, the speed controller parameters are part of the system response. From the transfer function it is observed that the time constant as well as the DC gain of this system now features the $G_S$ term. In practice this translates to a much faster system response since most well-tuned speed controllers have a proportional term greater than 1. While this is not always be the case, this was the simplifying assumption taken for further tests. With a decreased time constant, it is necessary to decrease $T_B$ for viable results. Furthermore, when estimating parameters from the linear least squares approximation, the terms introduced by the speed controller must be removed before estimating $J_m$ and $B_m$.

Following on from Equations (4.2) and (4.3)

$$peak = \frac{k}{\tau} = \frac{\left(\frac{G_S}{B_m + G_S}\right)}{\left(\frac{J_m}{B_m + G_S}\right)} = \frac{G_S}{J_m}$$
$$\Rightarrow J_m = \frac{G_S}{peak} \tag{4.7}$$

The result in Equation (4.7) shows that when the PRBS signal is added as a speed perturbation with the speed loop closed, the identification of the inertia $J_m$ is dependent on the speed controller $G_S$.

The $B_m$ term may be determined using Equation (4.8),

$$k = \left(\frac{G_S}{B_m + G_S}\right) = \frac{\text{Final Offset Value}}{\frac{A^2}{L}}$$
$$\Rightarrow \left(\frac{B_m + G_S}{G_S}\right) = \frac{\frac{A^2}{L}}{\text{Final Offset Value}} \tag{4.8}$$
$$\Rightarrow B_m = G_S\left[\left(\frac{\frac{A^2}{L}}{\text{Final Offset Value}}\right) - 1\right]$$

Therefore, in order to estimate parameter $B_m$, the speed controller term $G_S$ needs to be known, and compensated for by performing the computation shown in Equation (4.8).

Similarly, when using the method of linear least squares to estimate a transfer function that fits the impulse response from the correlation result, it is important to note that the estimated transfer function corresponds to the transfer function in Equation (4.6).

Comparing Equation (4.4) with Equation (4.6),

$$\begin{aligned}\frac{\omega}{\omega^*} &= \frac{G_S}{sJ_m + B_m + G_S} \\ &= \frac{\left(\frac{G_S}{J_m}\right)}{s + \left(\frac{B_m + G_S}{J_m}\right)} \\ &= \frac{b_0}{s + a_0} \qquad (4.9) \\ \therefore b_0 &= \left(\frac{G_S}{J_m}\right), a_0 = \left(\frac{B_m + G_S}{J_m}\right) \\ \therefore J_m &= \left(\frac{G_S}{b_0}\right), B_m = G_S\left[\left(\frac{a_0}{b_0}\right) - 1\right]\end{aligned}$$

It can be seen from Equation (4.10) that with the method of linear least squares, the speed controller term $G_S$ needs to be subtracted in the estimation of $B_m$ and it is also needed for the estimation of $J_m$, unlike in the case where the PRBS signal was applied as a torque perturbation. In this case, with the PRBS signal applied as a speed perturbation, the peak and the time constant of the impulse response changes as $G_S$ was varied. $G_S$ is inversely proportional to the time constant, and directly proportional to the peak of the response.

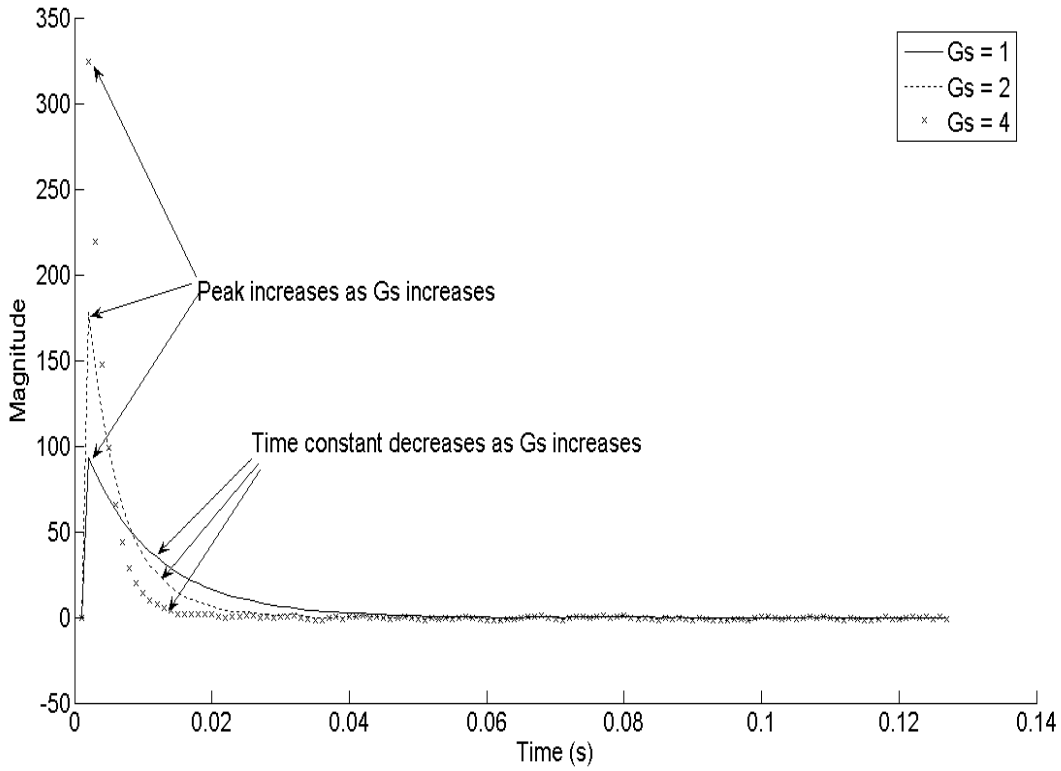



Figure 4-8 Comparing numerical impulse responses for different values of $G_S$ (PRBS applied as a speed perturbation with speed loop closed)

Figure 4-8 compares the numerical impulse responses as the speed controller gain parameter is changed. From this figure it can be seen that the speed controller affects the peak as well as the time constant of the response when the PRBS signal is applied as a speed perturbation.

With the PRBS signal applied as a speed perturbation and the speed output used as the system output, with the speed loop closed, the numerical impulse response that the test yields is for the system with the transfer function from Equation (4.6). This system includes the speed controller term, $G_S$, which affects both the time constant and the DC gain.

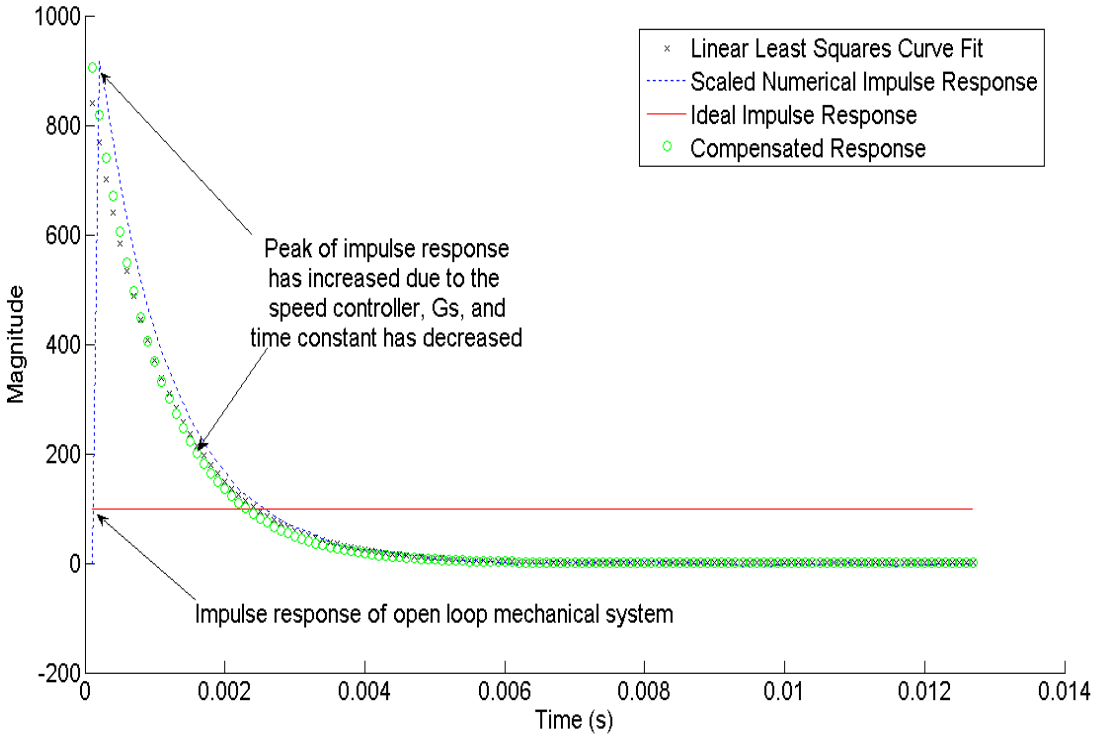



Figure 4-9 Simulated response with the PRBS signal applied as a speed perturbation, with the speed loop closed

Figure 4-9 clearly shows the difference between the open loop mechanical system response (the solid line marked *ideal impulse response*) and the numerical impulse response, due to the effect of the speed controller; the impulse response is seen to have a far smaller time constant due to the speed controller term. Unlike the case displayed in Figure 4-6, where the PRBS signal was applied as a torque perturbation, when the PRBS signal is applied as a speed perturbation, the peak of the numerical impulse response for the closed loop system is different from that of the peak of the response for the open loop system. The result from Equation (4.7) shows that the peak has changed by the factor $G_S$.

Table 4-5 Summary of accuracy of parameter estimations for DC motor model with speed loop closed and PRBS applied as a speed perturbation ($J_m$=0.01, $B_m$=0.01), with varying values of $T_B$

| $T_b$ | Numerical Impulse Response | | Linear Least Squares | |
|---|---|---|---|---|
| | $J_m$ error (%) | $B_m$ error (%) | $J_m$ error (%) | $B_m$ error (%) |
| $1x10^{-5}$ | 3.7 | >10000 | 3.0 | >10000 |
| $1x10^{-4}$ | 9.2 | >1000 | 8.4 | >1000 |
| $1x10^{-3}$ | 59.6 | >1000 | 49.6 | >1000 |

From Table 4-5, it can be seen that as the bit period $T_B$ is decreased, the accuracy of the estimation for $J_m$ increases. When comparing Table 4-4 and Table 4-5, the results obtained are very similar. This is expected, since the same open loop system is used in both cases, and although both cases involve the PRBS signal being applied at different points in the closed loop system, the speed controller term $G_S$ is known, and is compensated for in both instances, using Equations (4.5) and (4.9)

As the number of repetitions of the PRBS sequence is increased, the accuracy of the estimation for $B_m$ increases, as may be seen in Table 4-6.

Table 4-6 Summary of accuracy of parameter estimations for increasing number of PRBS sequence repetitions ($J_m$=0.01, $B_m$=0.01, $T_B$=0.001)

| N | $J_m$ error (%) | $B_m$ error (%) |
|---|---|---|
| 10 | 59.6 | 1139 |
| 100 | 59.4 | 109.4 |
| 1000 | 59.3 | 5.3 |

Based on the results shown in Table 4-3, Table 4-4, Table 4-5, and Table 4-6 it was concluded that the two methods of applying the PRBS signal, as either a torque or speed perturbation, were interchangeable in that they both produced estimations for $J_m$ and $B_m$ of similar accuracy.

In both cases, the speed controller parameters need to be determined and compensated for in the parameter extraction algorithm, and the method to do this was shown in Equation (4.5) and Equation (4.9).

Modern VSDs allow the user or control system to access various parameters and inputs in the speed and torque control loops, an example of which is shown in Figure 4-10.

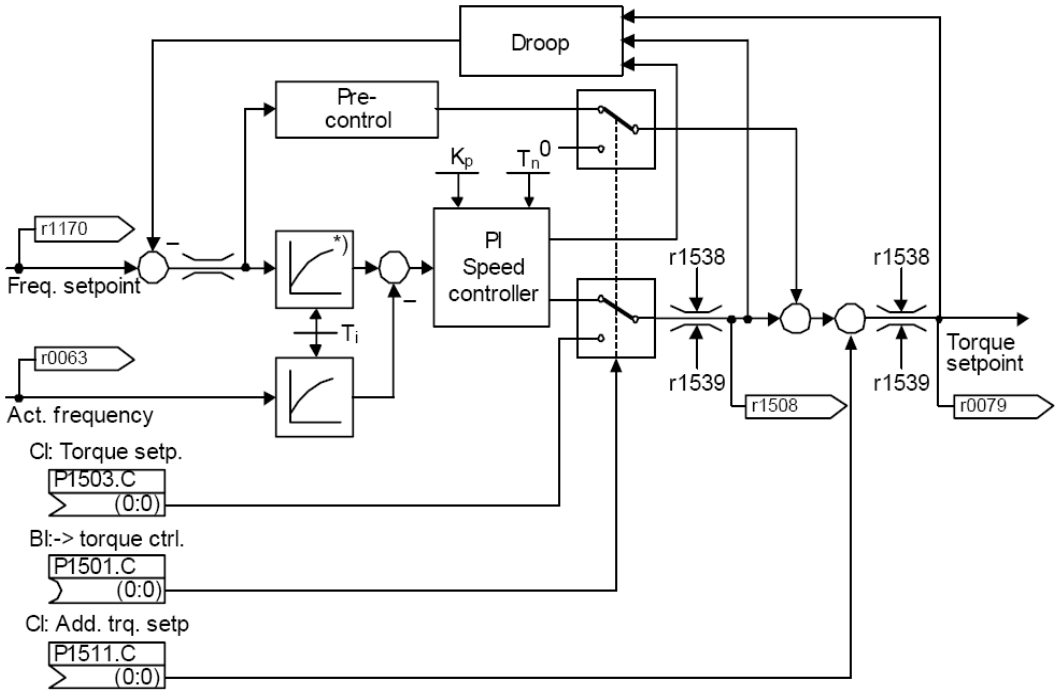



Figure 4-10 Block diagram representation of the speed control loop in a Siemens Micromaster VSD

In Figure 4-10, it can be sent that a parameter exists (in this case P1511.C) which allows one to input a control signal as an additional torque setpoint. Note that this torque signal is added to the torque control signal output from the speed controller. Such an additional torque input may be used to apply a PRBS signal as a torque perturbation in a closed loop speed controlled system. Likewise, a provision is also made in this VSD for an additional speed setpoint, and this may be used to apply the PRBS signal as a speed perturbation.

In addition to the control signal inputs that modern VSDs make provision for, one can also read the speed and torque controller parameters from the VSD. In Figure 4-10, this is shown as the $K_p$ and $T_n$ parameters above the block labelled *PI Speed Controller*.

These drive features make this method of online parameter identification a reality, and these control inputs were used in the practical implementation discussed in Chapter 5.

Even where the controller parameters are not contained in the VSD, these parameters may be found either in the PLC where the control loop is programmed, or in the documentation developed and maintained throughout the life cycle of the plant in which the machine under test resides.

## 4.4 DC Motor With Load

The previous section discussed simulations performed on models of systems that were under no load conditions, where only the motor inertia and motor viscous friction coefficient were identified. In most applications, the presence of a load with its own inertia and friction must be accounted for as the load usually considerably contributes to the total inertia and friction of the machine.

### 4.4.1 DC Motor with Drive Train

With the addition of a drive train, namely, a load and coupling to the motor, the system block diagram needs to be modified to include the additional elements.

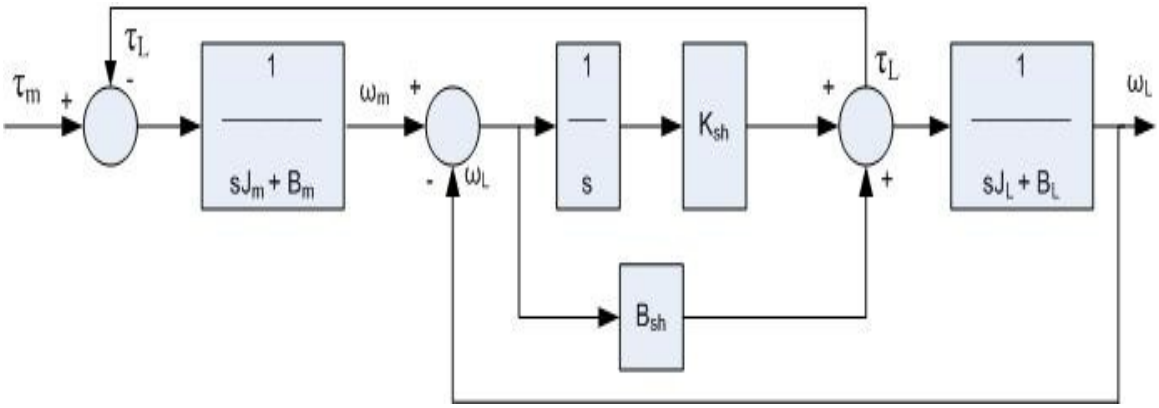



Figure 4-11 Complete model of the mechanical system, with shaft and load taken into account

When comparing Figure 4-11 and Figure 2-2, additional parameters $K_{sh}$, $B_{sh}$, $J_L$, and $B_L$ are noted. Here,

| | | | |
|---|---|---|---|
| $K_{sh}$ | = | Shaft/Coupling elasticity/stiffness constant | [Nm.rad$^{-1}$] |
| $B_{sh}$ | = | Viscous friction coefficient of shaft/coupling | [Nm.rad$^{-1}$s] |
| $J_L$ | = | Inertia of the load | [kgm$^2$] |
| $B_L$ | = | Viscous friction coefficient of load | [Nm.rad$^{-1}$s] |
| $\tau_L$ | = | Torque opposing motion due to load | [Nm] |
| $\omega_L$ | = | Angular velocity of the load | [rad.s$^{-1}$] |

$$\begin{aligned}\omega_m &= \left[\frac{1}{sJ_m + B_m}\right](\tau_m - \tau_L) \\ \tau_L &= \left(\frac{K_{sh}}{s} + B_{sh}\right)(\omega_m - \omega_L) \\ &= \left(\frac{K_{sh}}{s} + B_{sh}\right)\left(\omega_m - \left[\frac{1}{sJ_L + B_L}\right]\tau_L\right)\end{aligned} \tag{4.10}$$

$$\tau_L = \frac{\left(\frac{K_{sh}}{s} + B_{sh}\right)}{\left(1 + \frac{\frac{K_{sh}}{s} + B_{sh}}{sJ_L + B_L}\right)}\omega_m \tag{4.11}$$

Substituting the result from Equation (4.11) into Equation (4.10),

$$\omega_m = \left[\frac{1}{sJ_m + B_m}\right]\tau_m - \left[\frac{1}{sJ_m + B_m}\right]\left[\frac{\left(\frac{K_{sh}}{s} + B_{sh}\right)}{1 + \frac{\left(\frac{K_{sh}}{s} + B_{sh}\right)}{\left(sJ_{_L} + B_L\right)}}\right]\omega_m$$

$$\therefore \omega_m\left[1 + \frac{\left(\frac{K_{sh}}{s} + B_{sh}\right)}{\left[sJ_m + B_m\right]\left[1 + \frac{\left(\frac{K_{sh}}{s} + B_{sh}\right)}{\left(sJ_{_L} + B_L\right)}\right]}\right] = \left[\frac{1}{sJ_m + B_m}\right]\tau_m$$

$$\therefore \frac{\omega_m}{\tau_m} = \frac{\left[\frac{1}{sJ_m + B_m}\right]}{\left[1 + \frac{\left(\frac{K_{sh}}{s} + B_{sh}\right)}{\left[sJ_m + B_m\right]\left[1 + \frac{\left(\frac{K_{sh}}{s} + B_{sh}\right)}{\left(sJ_{_L} + B_L\right)}\right]}\right]}$$

$$= \frac{1}{\left[sJ_m + B_m\right] + \frac{\left(\frac{K_{sh}}{s} + B_{sh}\right)}{\left[1 + \frac{\left(\frac{K_{sh}}{s} + B_{sh}\right)}{\left(sJ_{_L} + B_L\right)}\right]}}$$

$$= \frac{1}{\left[sJ_m + B_m\right] + \frac{\left(\frac{K_{sh}}{s} + B_{sh}\right)\left(sJ_{_L} + B_L\right)}{\left[\left(sJ_{_L} + B_L\right) + \left(\frac{K_{sh}}{s} + B_{sh}\right)\right]}}$$

$$= \frac{\left[\left(sJ_{_L} + B_L\right) + \left(\frac{K_{sh}}{s} + B_{sh}\right)\right]}{\left[sJ_m + B_m\right]\left[\left(sJ_{_L} + B_L\right) + \left(\frac{K_{sh}}{s} + B_{sh}\right)\right] + \left(\frac{K_{sh}}{s} + B_{sh}\right)\left(sJ_{_L} + B_L\right)}$$

$$\therefore \frac{\omega_m}{\tau_m} = \frac{s^2 J_{_L} + s[B_L + B_{sh}] + K_{sh}}{s^3[J_m J_L] + s^2[J_m(B_L + B_{sh}) + J_L(B_m + B_{sh})] + s[B_{sh}(B_L + B_m) + K_{sh}(J_m + J_L) + (B_m B_L)] + [K_{sh}(B_m + B_L)]} \quad (4.12)$$

From the above result, it is clear that the transfer function no longer describes a first order system, once the shaft and load dynamics have been taken into account.

In order to determine the parameters of the system, a reasonable approach would be to compare the above transfer function with a generalised third-order continuous-time transfer function of the form,

$$P(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0} \tag{4.13}$$

By comparing Equation (4.12) to the generalised transfer function in Equation (4.13), the following system of equations results,

$$a_2 = \frac{B_L + B_{sh}}{J_L} + \frac{B_m + B_{sh}}{J_m} \tag{4.14}$$

$$a_1 = \frac{B_{sh}(B_L + B_m) + K_{sh}(J_m + J_L) + (B_m B_L)}{J_m J_L} \tag{4.15}$$

$$a_0 = \frac{K_{sh}(B_m + B_L)}{J_m J_L} \tag{4.16}$$

$$b_2 = \frac{1}{J_m} \tag{4.17}$$

$$b_1 = \frac{B_L + B_{sh}}{J_m J_L} \tag{4.18}$$

$$b_0 = \frac{K_{sh}}{J_m J_L} \tag{4.19}$$

By fitting a linear least squares estimate to the third order system, the estimated *a* and *b* parameters that result can be used to solve this system of equations, in order to determine $J_m$, $J_L$, $B_m$, $B_L$, $K_{sh}$, and $B_{sh}$. However, the solution is very sensitive to small changes in the parameters especially since mechanical parameters are usually of the order of $10^{-3}$ or smaller when using SI units [Tallfors1]. The viscous friction coefficients are very small and are thus especially sensitive to variation in the estimated parameters. A simplifying approximation approach is to ignore $B_m$ and $B_L$. This new, approximate system is given by,

$$\frac{\omega_m}{\tau_m} \approx \frac{s^2 J_{_L} + sB_{sh} + K_{sh}}{s^3[J_m J_L] + s^2[B_{sh}(J_m + J_L)] + s[K_{sh}(J_m + J_L)]} \tag{4.20}$$

Now, comparing to the generalised third-order transfer function, with $a_0 = 0$,

$$a_2 = \frac{B_{sh}(J_m + J_L)}{J_m J_L} \tag{4.21}$$

$$a_1 = \frac{K_{sh}(J_m + J_L)}{J_m J_L} \tag{4.22}$$

$$b_2 = \frac{1}{J_m} \tag{4.23}$$

$$b_1 = \frac{B_{sh}}{J_m J_L} \tag{4.24}$$

$$b_0 = \frac{K_{sh}}{J_m J_L} \tag{4.25}$$

The resulting over determined system (four unknown parameters and five equations) can be used in conjunction with the method of linear least squares to determine the unknown parameters.

Since the viscous friction coefficients were ignored in the above approximation, it remains for these to be estimated. One way of doing this follows from [Tallfors1], where the following relationship is used,

$$\tau_m = (B_m + B_L)\omega_m + \tau_0 \tag{4.26}$$

where $\tau_0$ = idle motor torque, the torque required when motor speed is zero to keep the load stationary.

By taking a number of readings of $\tau_m$ and $\omega_m$, and extrapolating the resulting straight line curve when $\tau_m$ is plotted against $\omega_m$, the gradient of the line will yield $B_{tot} = B_m + B_L$, while the y-intercept is at $\tau_0$.

This method allows only the total viscous friction, $B_{tot}$, to be determined. Some further assumptions are required in order to determine the individual coefficients. [Tallfors1] calculates the individual coefficients based on the relative size of the corresponding inertia, as follows,

$$B_m = \frac{B_{tot} J_m}{J_m + J_L} \tag{4.27}$$

$$B_L = \frac{B_{tot} J_L}{J_m + J_L} \tag{4.28}$$

For the purposes of this research, the individual viscous friction coefficients are insignificant – the total viscous friction coefficient will suffice.

So far, the approximation was made based on the fact that the viscous friction coefficients were negligible, and could be ignored at first. However, another simplifying assumption may be made in an attempt to simplify the original transfer function (Equation 4.12). In the case of a rigid shaft coupling the load to the motor, the shaft elasticity constant has a large numerical value. Therefore, terms including $K_{sh}$ in the transfer function will dominate, such that,

$$\frac{\omega_m}{\tau_m} \approx \frac{K_{sh}}{sK_{sh}(J_m + J_L) + K_{sh}(B_m + B_L)} = \frac{1}{s(J_m + J_L) + (B_m + B_L)} \tag{4.29}$$

From this result it can be deduced that for a rigidly coupled system, the dynamics of the entire mechanical system can be modelled as a first-order system, where the inertia term is now the total system inertia, $J_T = J_m + J_L$, and the viscous friction coefficient is the total coefficient, $B_T = B_m + B_L$.

The purpose of the following simulations is to test the above assumption, that the total system inertia and total system viscous friction coefficient may be equated to the sum of the respective motor and load parameters. It must be stated that this applies strictly to a rigidly coupled system because, in the event of a non-rigid coupling, other non-linear effects need to be taken into account, and these will be considered later.

### 4.4.2 PRBS applied as a torque perturbation, with speed loop open

The same tests as in the previous sections were repeated on a system with a load. Figure 4-12 shows the Simulink model that was used in the simulations of a motor with load attached. This model is based on the block diagram from Figure 4-11.

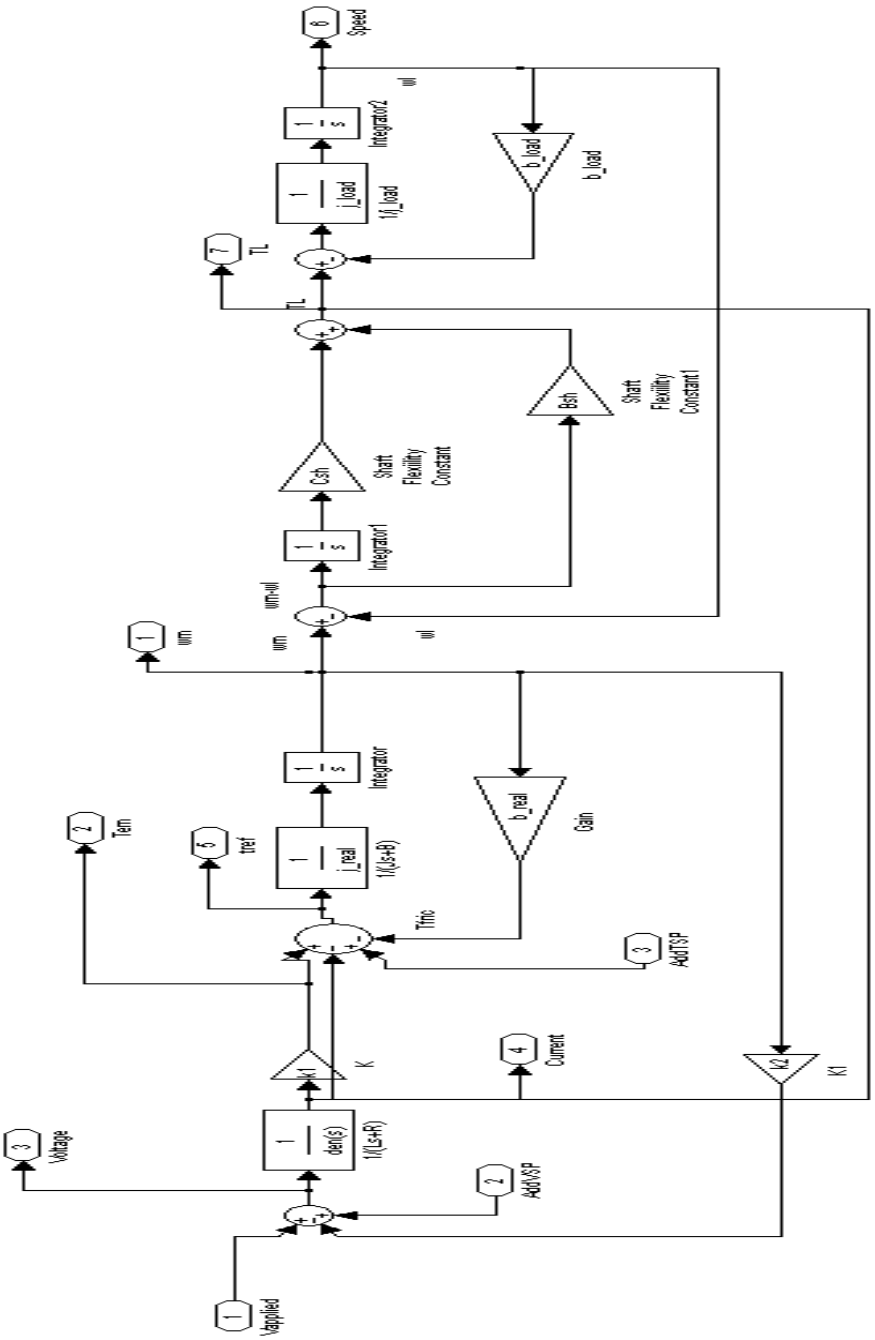



Figure 4-12 Simulink model for DC motor with attached load

First, the PRBS was added as a torque perturbation while the speed loop was open. Note that a high value for shaft flexibility $C_{sh}$ was chosen, of the order of $>10^6$, in order to simulate a rigid coupling.

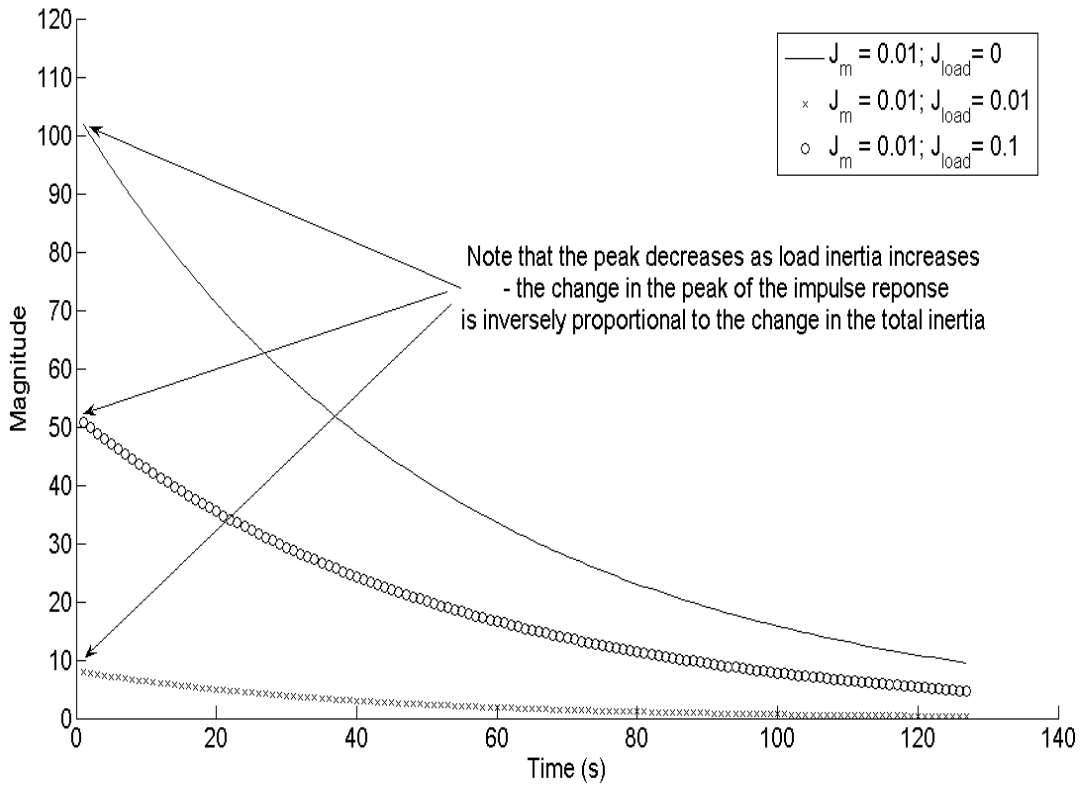



Figure 4-13 PRBS added as a torque perturbation to a system with an additional mechanical load

Figure 4-13 shows the comparison between the impulse responses for three different load conditions. Note that in all three instances, the motor inertia was kept constant at $J_m$=0.01kgm$^2$ and only the load inertia $J_L$ was changed. When comparing the instances where the load inertias are $J_L$=0.01kgm$^2$ and $J_L$=0.1kgm$^2$, the total inertia $J_T$ is 0.02kgm$^2$ and 0.101kgm$^2$, respectively.

Therefore, the ratio of the total inertia in the first instance (where $J_T$ =0.02kgm$^2$) to the second instance (where $J_T$ =0.101kgm$^2$) is approximately 0.2, which is the inverse of the ratio of the peak of the impulse response for the first instance (≈50) to the peak of the impulse response for the second instance (≈10). Thus, it can be seen that as the *total* inertia increases, the peak of the impulse response decreases proportionally.

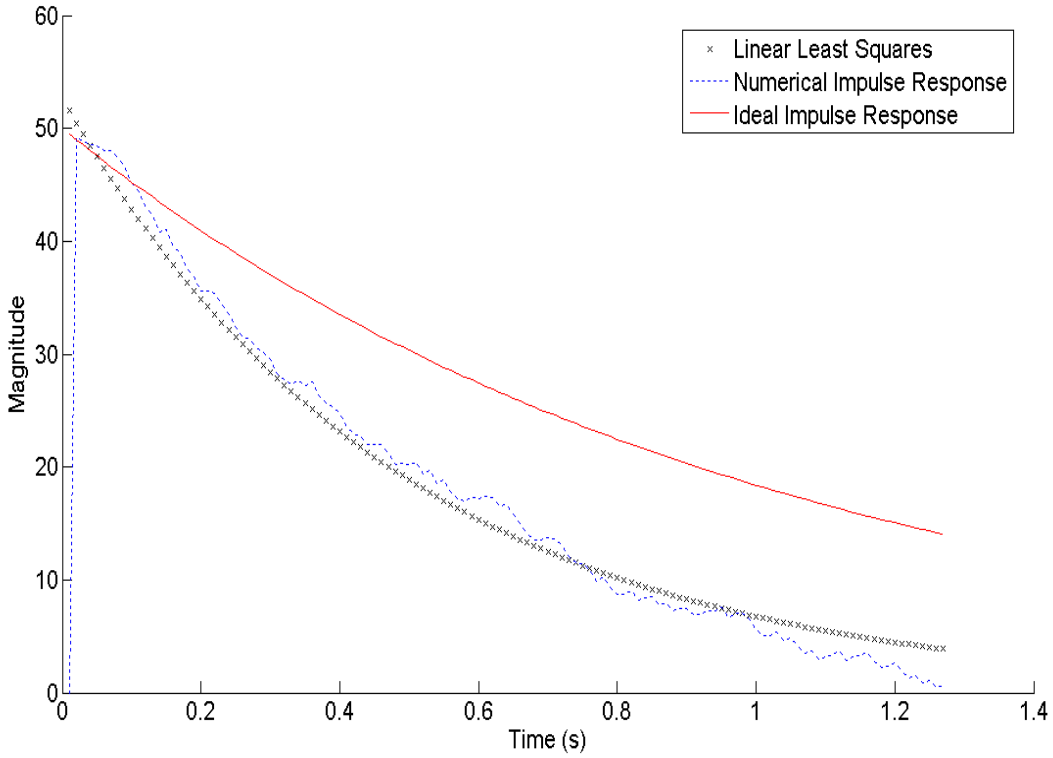



Figure 4-14 PRBS added as a torque perturbation to a system with an additional mechanical load

The system used in simulation to produce the results shown in Figure 4-14 had the following parameters: $J_m$=0.01kgm$^2$, $B_m$=0.01Nm.rad$^{-1}$s, $J_L$=0.01kgm$^2$, $B_L$=0.01Nm.rad$^{-1}$s,

In Figure 4-14, the linear least squares estimation is compared to the impulse response of the following transfer function:

$$\frac{1}{sJ_T + B_T} = \frac{1}{s(J_m + J_L) + (B_m + B_L)} \tag{4.30}$$

In order to test the assumption that the motor and load inertia may be lumped together into a total inertia term, a comparison was done between a motor with no load but with motor inertia $J_m$=0.02kgm$^2$, and the above described system where the sum of the motor and load inertias yields $J_T$=0.02kgm$^2$. The resulting impulse response is shown in Figure 4-15. It can be seen in this figure that the two results are only slightly different, which may be attributed to the terms from Equation 4.12 which were assumed to be negligent, but which have an effect on the numerical impulse response.

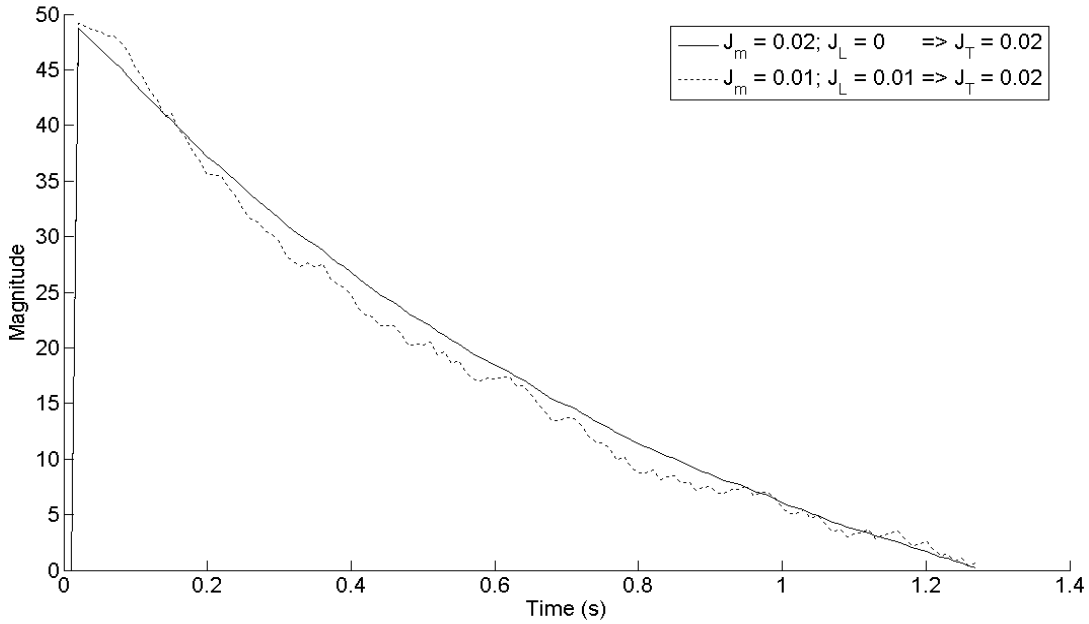



Figure 4-15 Comparing a motor only and motor-load system that have the same *total* inertias ($J_T$ = 0.02 kgm$^2$)

The same test was repeated for a different system, with parameters $J_m$=0.01kgm$^2$, $B_m$=0.01Nm.rad$^{-1}$s, $J_L$=1.0kgm$^2$, $B_L$=0.01Nm.rad$^{-1}$s.

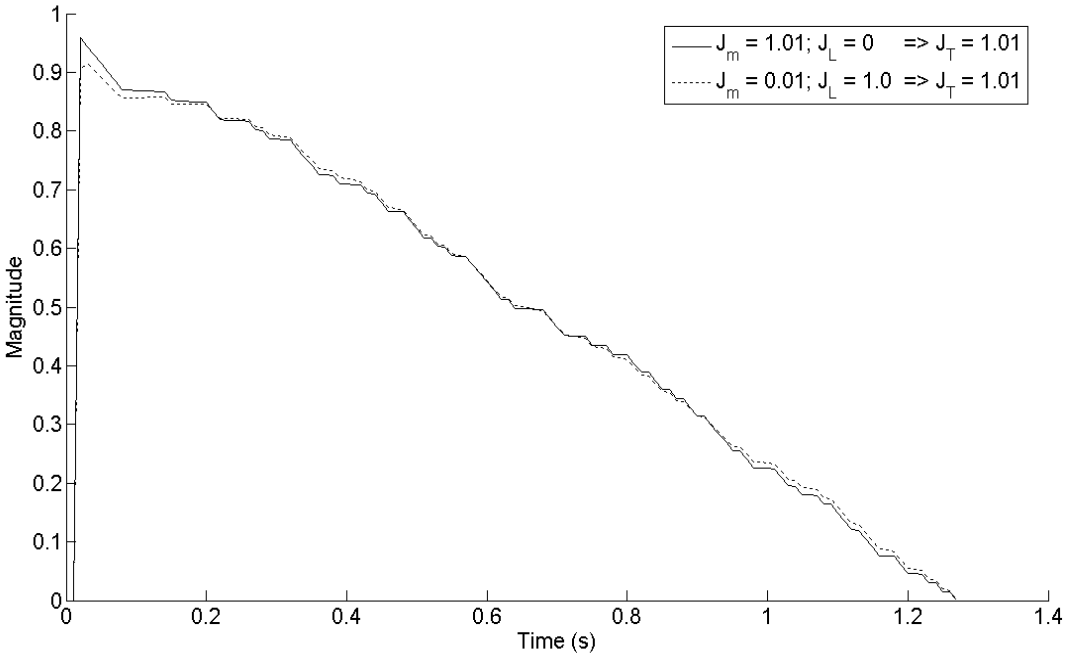



Figure 4-16 Comparing a motor only and motor-load system that have the same *total* inertias ($J_T$ = 1.01 kgm$^2$)

Figure 4-16 confirms what was also seen in Figure 4-15 - that adding a load to a motor increases the total inertia of the system by the amount of the load inertia, and when the method of applying a PRBS perturbation to the system is used, it is this *total* inertia $J_T$ that is estimated.

Table 4-7 Summary of parameter estimations for the simulations involving an additional mechanical load; PRBS applied as a torque perturbation in an open loop speed controlled system

| Actual Values | | | | | | Numerical Impulse Response | | Linear Least Squares | |
|---|---|---|---|---|---|---|---|---|---|
| $J_m$ | $B_m$ | $J_L$ | $B_L$ | $J_T$ | $B_T$ | $J_T$ error (%) | $B_T$ error (%) | $J_T$ error (%) | $B_T$ error (%) |
| 0.01 | 0.01 | 0.01 | 0.01 | 0.02 | 0.02 | 2.5 | 6.0 | 3.7 | 37.7 |
| 0.02 | 0.02 | 0 | 0 | 0.02 | 0.02 | 2.3 | 6.0 | 3.4 | 37.7 |
| 0.01 | 0.01 | 1.0 | 0.01 | 1.01 | 0.02 | 1.9 | 25.4 | 3.3 | 7.9 |
| 1.01 | 0.02 | 0 | 0 | 1.01 | 0.02 | 1.8 | 25.4 | 3.2 | 7.9 |

The results in Table 4-7 show that when a load was coupled to the motor in the simulations, the numerical impulse response generated by the PRBS-based test, as well as the linear least squares approximation of the results, produced estimates to within 10% accuracy of the *combined* motor and load inertia and viscous friction coefficient. This supports the assumption that for a rigidly coupled motor-load system, the motor and load inertia and friction terms may be lumped into *total* system inertia and friction terms.

It was concluded that the following assumptions were correct for the purpose of this research: that the total machine inertia may be approximated by the sum of the motor inertia and the load inertia, and likewise that the total machine viscous friction may be approximated by the combination of the motor viscous friction and load viscous friction. This applied to a mechanical system with a rigid coupling between the motor and load.

### 4.4.3 PRBS as a Speed Perturbation with speed loop closed

For the purpose of completeness, the PRBS signal was applied to a motor-load system as a speed perturbation with the speed loop closed. In the previous section it was shown that it is possible to apply the PRBS signal as either a torque or a speed input in a closed loop speed controlled system; and since both yield similar accuracies, it was decided to only show the results of a test where the PRBS was applied as a speed perturbation. Figure 4-11 shows the results of this test.

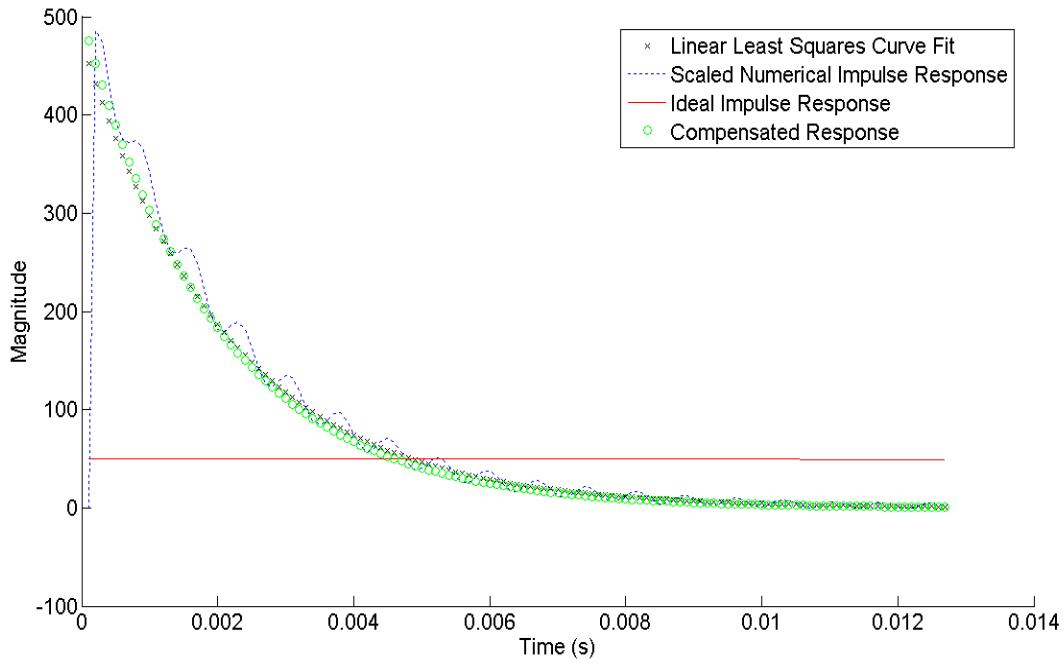



Figure 4-17 PRBS added as a speed perturbation to a closed loop speed-controlled system, with additional mechanical load ($J_m$=0.01kgm$^2$, $B_m$=0.01Nm.rad$^{-1}$s, $J_L$=0.01kgm$^2$, $B_L$=0.01Nm.rad$^{-1}$s)

Table 4-8 Summary of parameter estimations for the simulations involving an additional mechanical load; PRBS applied as a speed perturbation in a closed loop speed controlled system

| Actual Values | | | | | | Numerical Impulse Response | | Linear Least Squares | |
|---|---|---|---|---|---|---|---|---|---|
| $J_m$ | $B_m$ | $J_L$ | $B_L$ | $J_T$ | $B_T$ | $J_T$ error (%) | $B_T$ error (%) | $J_T$ error (%) | $B_T$ error (%) |
| 0.01 | 0.01 | 0.01 | 0.01 | 0.02 | 0.02 | 2.8 | 6.4 | 3.4 | 38.3 |
| 0.02 | 0.02 | 0 | 0 | 0.02 | 0.02 | 2.5 | 6.0 | 3.4 | 37.7 |
| 0.01 | 0.01 | 0.1 | 0.01 | 0.11 | 0.02 | 5.2 | 6.7 | 4.5 | 12.2 |
| 0.11 | 0.02 | 0 | 0 | 0.11 | 0.02 | 5.0 | 6.0 | 4.4 | 11.8 |

By looking at the error columns in Table 4-8 it can be seen that whether the motor and load inertias and viscous friction coefficients are treated separately, or if they are lumped together into a total inertia and total viscous friction coefficient (as if there is no load but merely a motor with these total parameters), the errors that this method yields is similar in both instances.

As was demonstrated in Section 4.3.2.2, it was necessary to compensate for the speed controller parameter $G_S$ in the estimation algorithm.

As expected, even with the speed loop closed, the motor and load inertia and friction may be lumped together into total machine inertia and total machine friction respectively.

## 4.5 Detection of Mechanical Imperfections

Up to now, the systems under investigation have been those of rigidly coupled systems. As was mentioned at the outset, an area of interest in this research is how the numerical impulse response is affected by the presence of non-linear effects, such as systems that have an elastic coupling between the motor and load, as well as systems with gear play in the couplings (backlash).

The purpose of the following simulations is not to estimate the system parameters in the presence of such effects, but merely to identify any patterns that may be reflected in the numerical impulse response when these phenomena are present.

### 4.5.1 Varying Shaft Elasticity

Earlier simulations did not account for systems with non-rigid couplings. An investigation was also undertaken into how an increase in the elasticity of the coupling between the motor and load may be identified in the numerical impulse response.

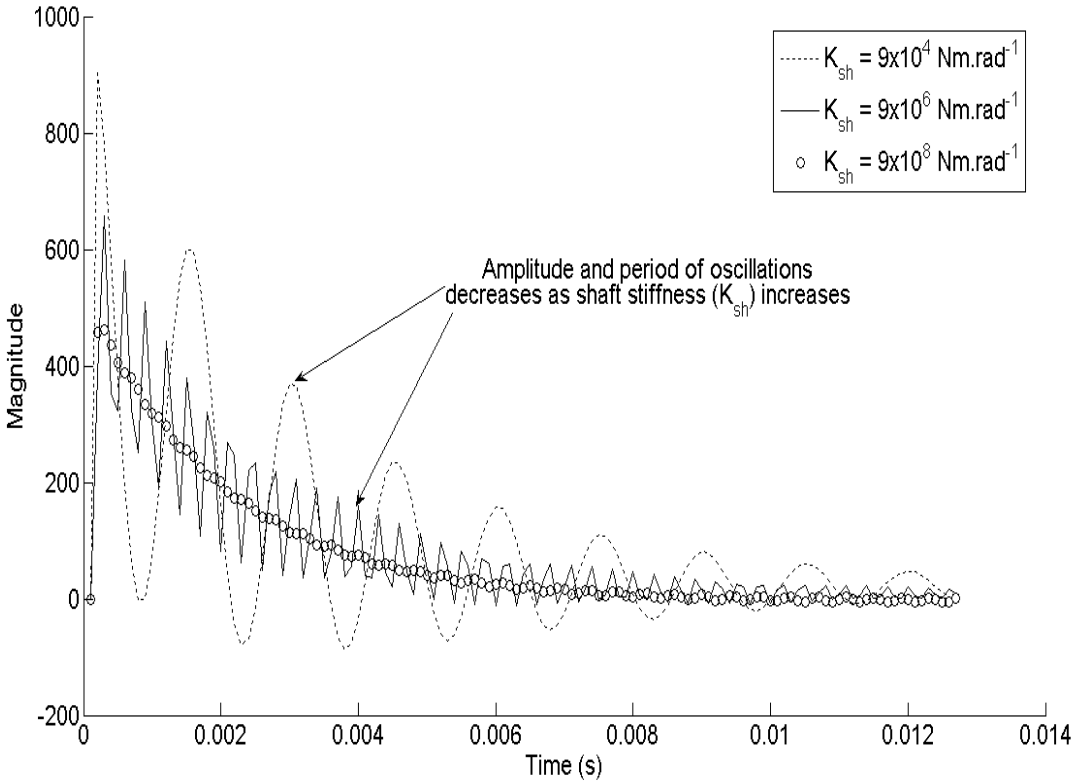



Figure 4-18 Comparison of systems with various degrees of shaft flexibility or elasticity

Figure 4-18 depicts the numerical impulse response for varying degrees of elasticity in the coupling. Following on from the information presented in Section 4.4.1 and as represented in the block diagram in Figure 4-11, the degree of elasticity may be varied by varying the shaft flexibility constant $K_{sh}$, where increasing $K_{sh}$ results in an increase in the shaft, or coupling, stiffness and hence decreases the degree of elasticity. As the elasticity in the coupling between the motor and load increases, oscillations are noted in the impulse response. The amplitude and period of these oscillations increase with an increase in elasticity. Due to these changes in the impulse response, the error in the approximation for $J_T$ and $B_T$ increases with increasing elasticity.

### 4.5.2 Backlash

Another common non-ideal effect found in machine systems, particularly those with geared couplings, is the presence of play in the gearing, termed backlash. Backlash is generally an undesirable effect, because the dead band introduced by backlash results in periods where a purely linear controller may not be in control of the plant. Ideally, one should be able to monitor machine systems to identify if the backlash in the system is increasing, an indication that the coupling is deteriorating and may require maintenance.

In order to ascertain what effect the presence of backlash has on the numerical impulse response, experiments were performed using the dead zone component in Simulink, effectively introducing a dead band into the system. This follows on the method demonstrated in [ODonovan1] for simulating backlash in a mechanical system.

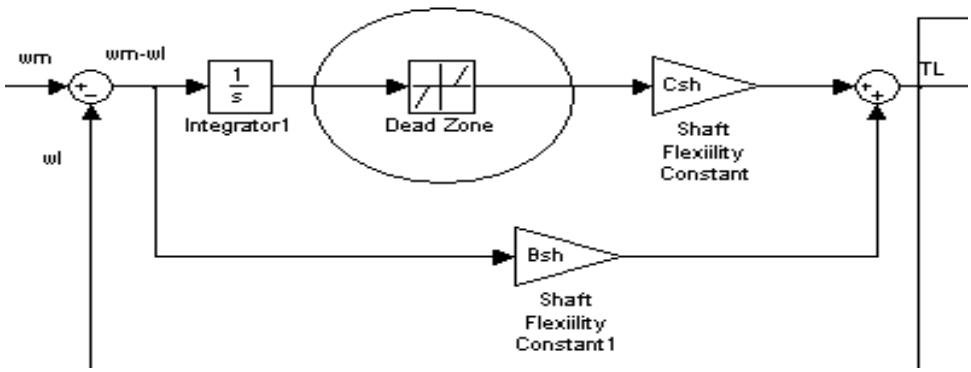



Figure 4-19 Part of Simulink model showing where Dead Zone component was used in the model to simulate backlash in the coupling

Figure 4-19 may be compared with Figure 4-12 to ascertain where in the Simulink model the dead zone component was used to simulate backlash.

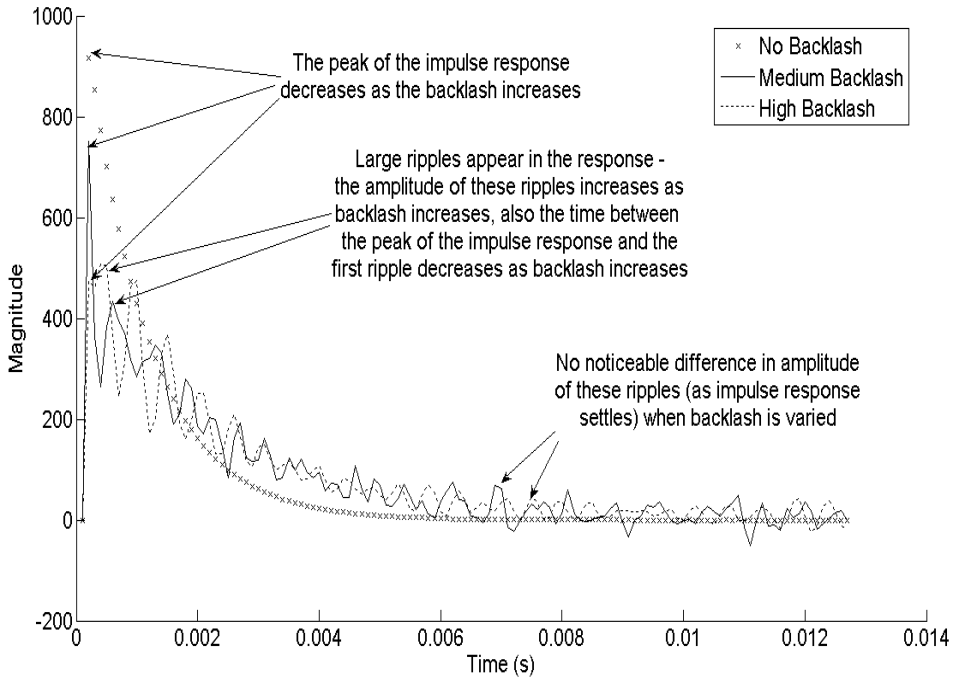



Figure 4-20 Comparison of impulse responses for systems with varying degrees of backlash or gear play in the coupling between motor and load

Figure 4-20 clearly shows that the presence of backlash is easily seen in the numerical impulse response. Note the presence of secondary peaks, or ripples, in the impulse response, other than the initial peak, when backlash is present. The amplitude of these peaks, which only appear during the initial part of the impulse response, increases as backlash increases. It is also seen that the initial peak of the impulse response decreases with an increase in backlash. Also to be noted is the fact that the ripples in the response as the response settles are of approximately the same amplitude irrespective of the degree of backlash. The presence of these oscillations in the impulse response results in inaccurate estimations for the $J_T$ and $B_T$ parameters.

## 4.6 Summary

This chapter has discussed the simulated implementation of the PRBS-based tests to identify mechanical system parameters. The simulation procedure and subsequent analyses were discussed. Results from the application of the PRBS signal as either a torque or a speed perturbation were compared, and conclusions were drawn about the responses that either method produces.

Results were displayed and discussed for systems under no load (motor only) as well as under load conditions. These simulations were performed using either of the PRBS application alternatives: as a torque perturbation with the speed loop open, as a torque perturbation with the speed loop closed, and as a speed perturbation with the speed loop closed. It was concluded that these testing methodologies are all viable, with the closed loop tests being necessary for the tests to be administered on an online system; however this method requires *a priori* knowledge of the speed controller parameters. An example was given of a Siemens VSD (see Figure 4-10) that allows the user or control system to read from or write to the aforementioned controller parameters, such as the proportional, integral, and derivative terms of the speed and torque controllers.

The simulated responses under no load conditions produced estimations for $J_m$ and $B_m$ with less than 10% errors in all cases. These errors were shown to decrease as one varied $T_B$, with small values of $T_B$ providing a high resolution of data points around the peak of the impulse response, and hence high accuracies for the estimation of $J_m$; while large values of $T_B$ were shown to produce high accuracies in the estimation of the gain of the system, and subsequently for the $B_m$ term.

Simulating a mechanical load was discussed in Section 4.4.1. When a load was added to the system in simulations, it was shown that the assumption was correct that the motor inertia and load inertia may be lumped into a total system inertia parameter $J_T$, and likewise that the motor viscous friction coefficient and load viscous friction coefficient may be added together as the total system viscous friction coefficient $B_T$.

An investigation into the presence of mechanical imperfections and their effect on the numerical impulse response was also undertaken. Qualitative results were shown for the patterns that one can identify from the impulse response when the motor-load coupling is flexible, and for cases where there is excessive gear play in the coupling. While this part of the investigation was qualitative, it is hoped that future research may look into the possibility

of enhancing the work done here to identify quantitatively the degree of elasticity or backlash, as well as provide accurate parameter estimations for $J_T$ and $B_T$ even in the presence of these phenomena.

The following chapter will discuss a practical implementation of these PRBS-based tests on a drive-based mechanical system.

# CHAPTER 5
# PRACTICAL IMPLEMENTATION

## 5.1 Introduction

Chapter 4 discussed various simulations that were performed to verify the viability of the PRBS-based tests to identify the parameters of a mechanical system, as well as to apply these tests to the identification of effects such as shaft flexibility and backlash. This chapter discusses the work that was done to implement these tests on a practical system.

Practical implementation of this method should involve a system that is comprised of the same components that one would find in most industrial machine systems, namely, a VSD, a PLC, and speed feedback from the motor such as from a quadrature encoder.

## 5.2 Automation System Configuration

The test setup used in this research included an induction motor under the torque and speed control (field-oriented control) of a VSD, a quadrature encoder to measure shaft speed, and a PLC to generate the PRBS signal as well as to perform the correlation computation.

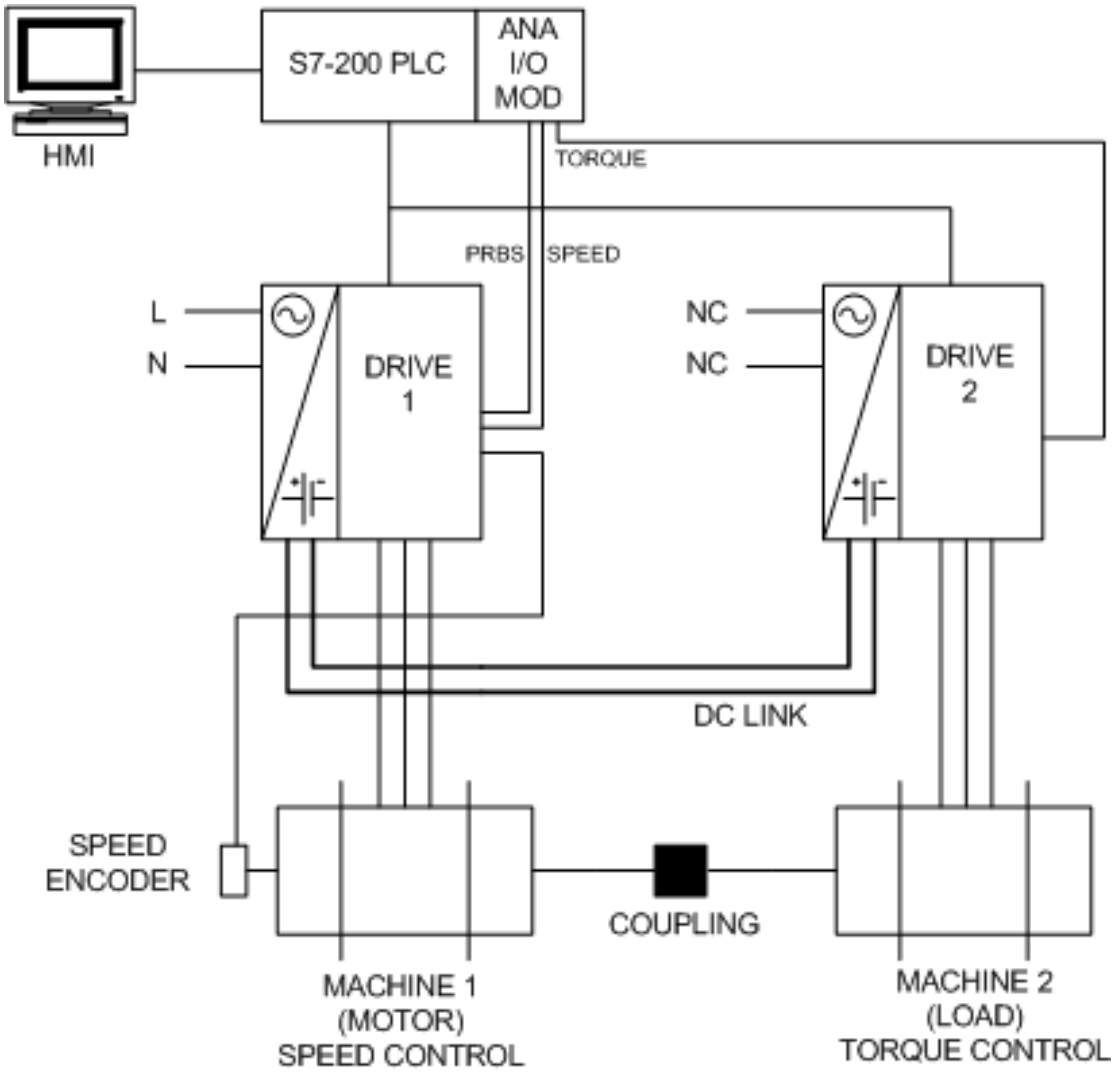



Figure 5-1 Diagrammatic representation of the practical rig used to perform measurements

The focus of the research is estimation of mechanical parameters for a motor and load system, as well as monitoring phenomena such as backlash and shaft elasticity. Toward the development of tests for such parameter extraction, the test rig shown in Figure 5-1 was constructed.

In this system, machine (1) is the driving motor. The shaft of machine (1) is attached with a coupling to the shaft of the load, machine (2). During testing, this load machine was removed to allow other inertial loads to be coupled to machine (1). These loads will be discussed later in this chapter.

The variable speed drives for the motors are Siemens MICROMASTER 440 VSDs. The analogue I/O channels in the PLC are used to transmit the generated PRBS signal from the PLC to the drive as well as to sample the motor speed.

The drives themselves are programmable and contain function blocks to facilitate programming.

Figure 5-2 is a depiction of one of the technology blocks that are accessible in the drive, namely the speed/torque control loop block. Figure 5-2 is the same as Figure 4-10, and has been included for ease of reading. In order to apply the PRBS input into the torque control loop, the drive is set to transfer what it receives on one of its analogue input channels to parameter P1511. The analogue output from the PLC with the PRBS signal is physically attached to this analogue input on the drive.

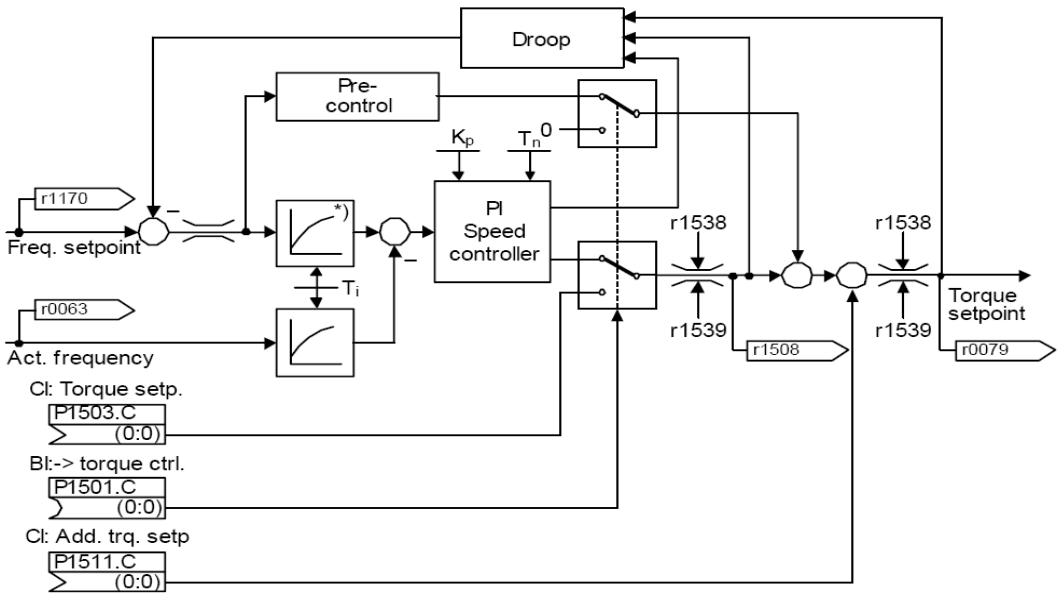



Figure 5-2 Block diagram representation of the speed control loop in a Siemens Micromaster VSD

A similar technology block is available to add an additional speed set point to the speed controller's summing junction. The following results were obtained by simplifying the test and adding the PRBS as a torque perturbation when parameter estimation was performed, and not as a speed perturbation. This was done so as not to have to take into account the

speed controller parameters, as discussed in Section 4.3.2.1 and Section 4.3.2.2. However, for the identification of elasticity and backlash, the PRBS was added as a speed perturbation, as the focus here was not on parameter identification, but rather to use the increased sensitivity introduced when the speed loop is closed to detect these phenomena more readily.

It may also be observed in Figure 5-2 that the speed controller's parameters are accessible to the user, and this will be necessary in possible future work to compensate for the effect of the speed controller. Furthermore, when adding the PRBS signal as a speed perturbation for the identification of elasticity and backlash, the speed controller was configured as a proportional only controller (refer to discussion in Section 4.3.2).

## 5.3 Experimental Procedure and Algorithm Overview

After simulating and testing the PRBS generation and correlation algorithms as discussed in Chapter 3, these routines were then programmed in ladder logic for a PLC. The type of PLC used was a Siemens S7-226XM PLC, which belongs to the S7-200 family of PLCs. Additionally, an analogue I/O module was used with four analogue input channels and one analogue output channel. The use of this additional module was required to facilitate high speed analogue signal generation and sampling.

The analogue output channel was used to output the generated PRBS signal. This output was connected to an input on the VSD, which was further configured in the drive's software to be treated as an additional torque set point to the torque controller.

Since the encoder was wired to the VSD, a requirement for high precision field oriented control, an output from the VSD was configured to transmit the motor speed data to one of the analogue input channels on the PLC.

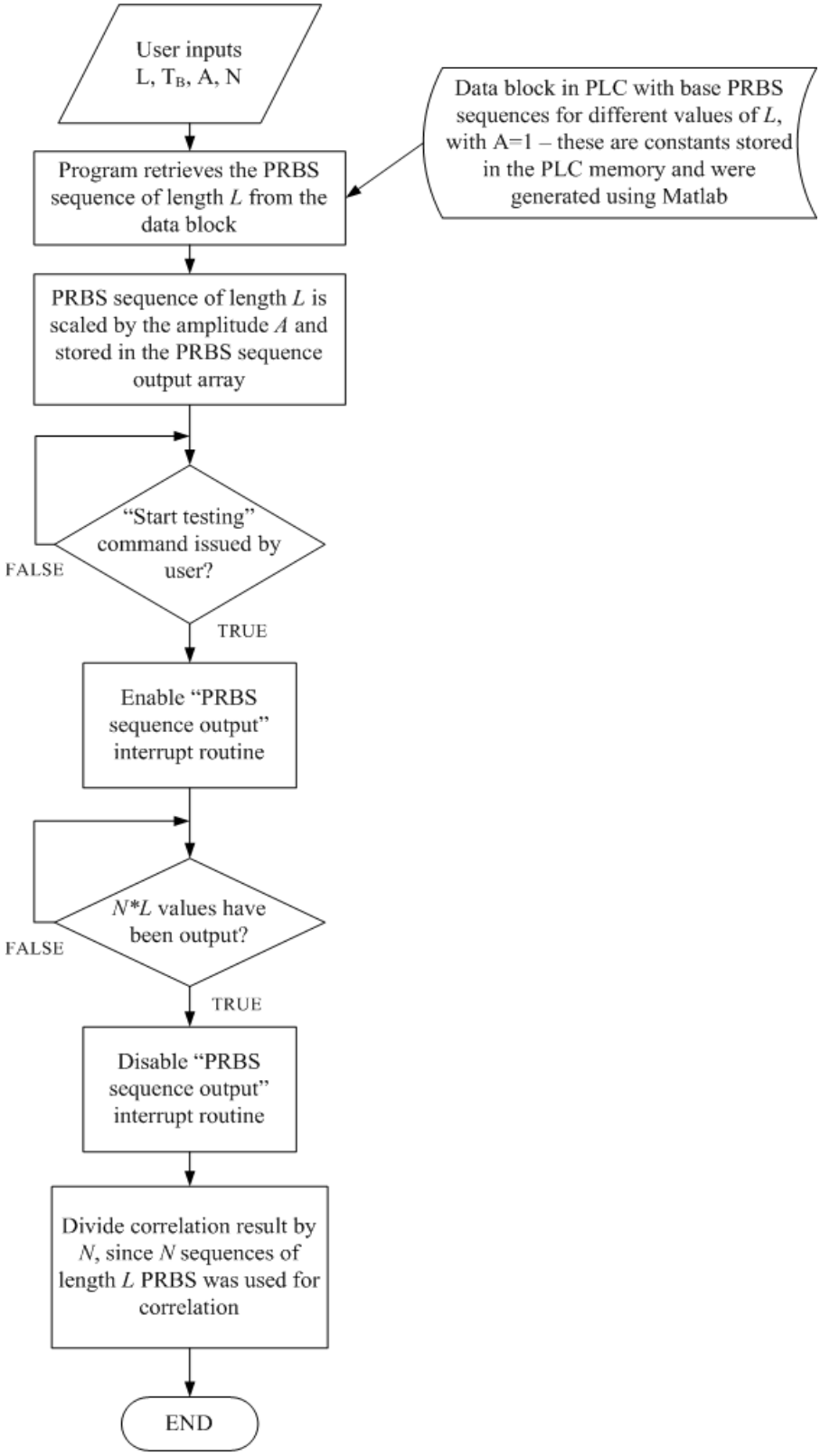



Figure 5-3 Flowchart algorithm for the program that was used for testing

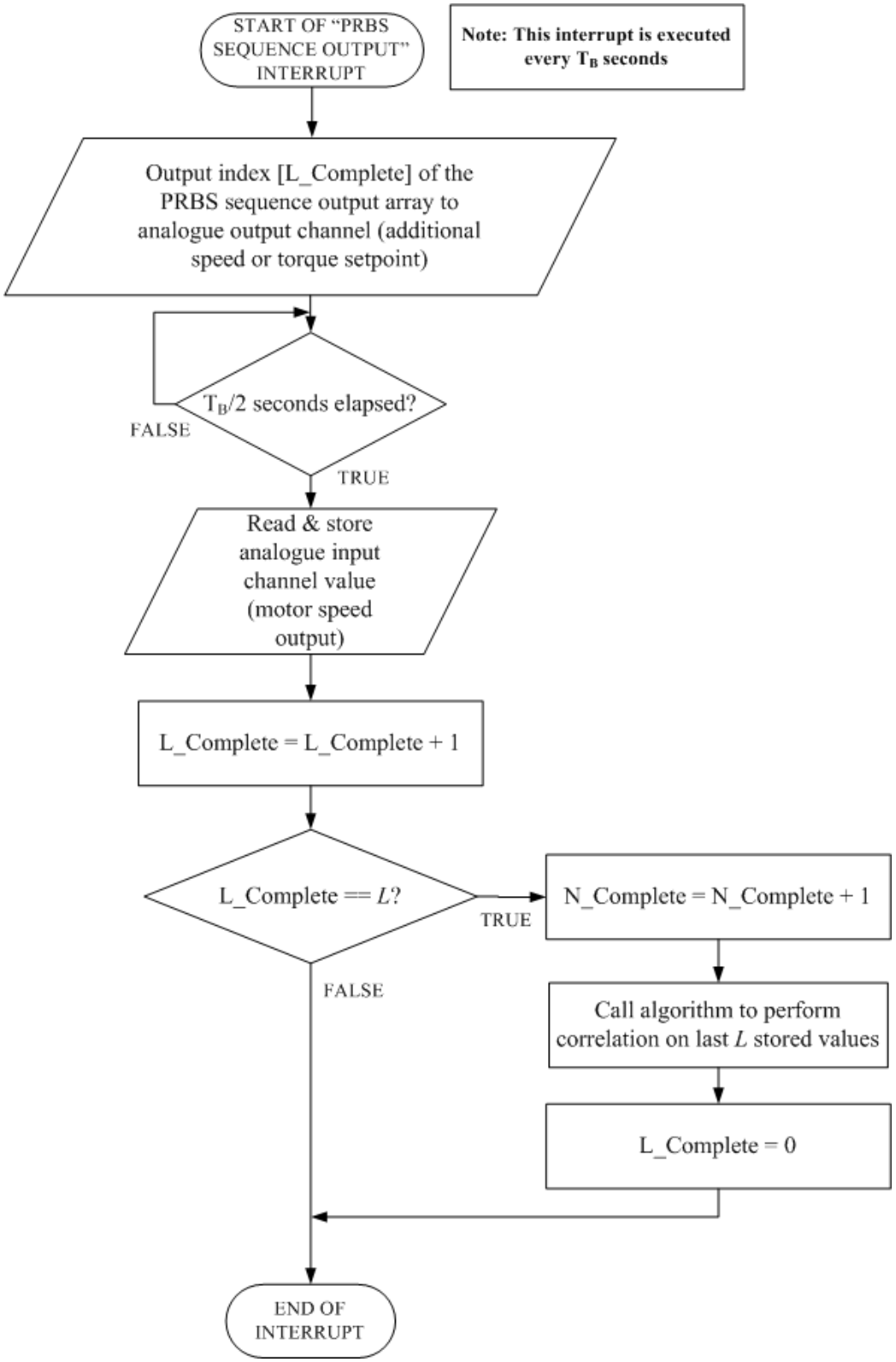



Figure 5-4 Flowchart algorithm for interrupt routine that outputs PRBS sequence and reads motor speed

The execution of the program is interrupt driven. The user specifies $L$, $T_B$, $A$, and $N$. Base sequences for different values of $L$ ($L$=15, 31, 63, 127, 255) were generated in Matlab, and then programmed as constants in a data block in the PLC. This base PRBS sequence, which has an amplitude of 1, is then scaled by $A$, and this $L$-length PRBS sequence with amplitude $A$ is stored in a PRBS output sequence array.

When the user issues the command to start the testing process, an interrupt routine is activated that executes every $T_B$ seconds. When this interrupt routine executes it outputs the next value of the PRBS sequence to the analogue output channel.

After $T_B/2$ seconds, the motor speed is read and stored. After each set of $L$ values have been outputted, the sequence starts again from the first value in the PRBS sequence. Concurrently, the correlation computation begins. After the $L$-length sequence has been output $N$ times and the $N^{th}$ correlation computation has completed, the testing routine completes. Since at each correlation computation the result is added to the previous result, when the routine completes the correlation result is averaged over $N$ times.

Only the correlation data was then exported to MATLAB for further analysis. The analysis on the data included scaling, performing the least squares fit to the data, computing total inertia and total viscous friction coefficients, as well as plotting the impulse response. Note that the actual correlation is done in the PLC.

## 5.4 Inertia Calculations

This section contains the measurements and calculations that were used to get an initial value for the inertias of the various elements used in the practical implementation of the PRBS-based tests.

### 5.4.1 Load Type 1 – Steel Disk

It was first necessary to determine the inertia of this disk-shaped load, as it was used to perform the inertia identification for the induction motor. The disk and its dimensions are displayed in Figure 5-5.

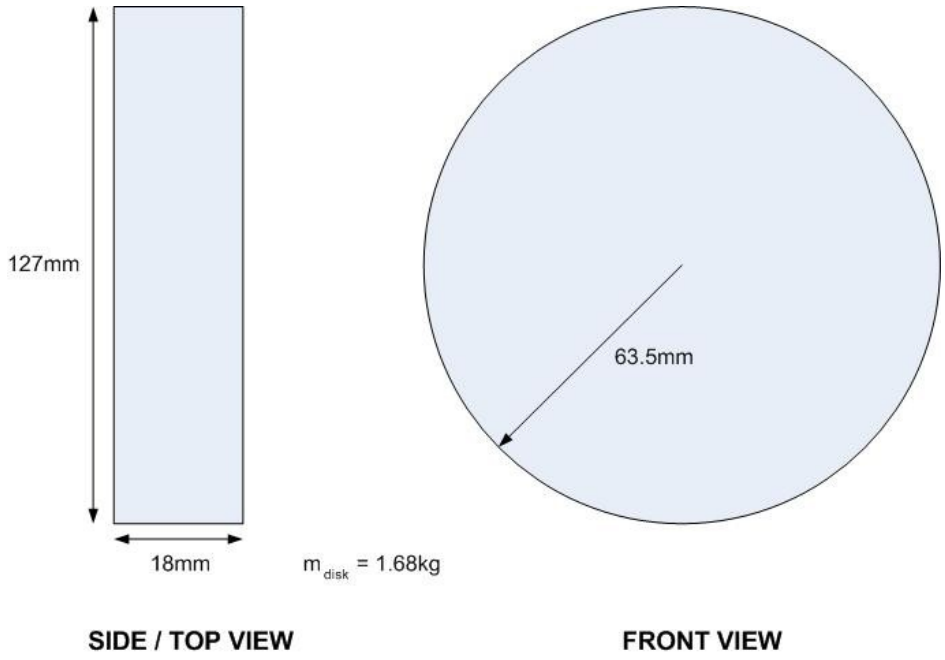



Figure 5-5 Top and Front Views of Steel Disk

Using the formula for the inertia of a cylindrical object,

$$\begin{aligned} J_{disk} &= \frac{1}{2} m_{disk} r_{disk}^2 \\ &= \frac{1}{2}(1.68kg)(63.5\times10^{-3}m)^2 \\ &= 3.387\times10^{-3}kg.m^2 \end{aligned} \tag{5.1}$$

Also, since the disk was uniformly shaped, it was decided to use its mass and volume to get an estimate of the density of the alloy used, as the density was required to establish the density of the second, less uniformly shaped load attachment, which was made of the same alloy.

$$\begin{aligned} \rho &= \frac{m}{v} \\ &= \frac{m}{\pi r^2 h} \\ &= \frac{1.68kg}{\pi(63.5\times10^{-3}m)^2(18\times10^{-3}m)} \\ &= 7367.83kg.m^{-3} \end{aligned} \tag{5.2}$$

### 5.4.2 Induction Motor

The retardation, or run-down, test discussed in Section 2.8.1 was used to determine the inertia of the motor. The motor was run at a steady-state speed of 45rad.s$^{-1}$ before being disconnected from the supply. The motor speed as it slowed down was sampled by one of the analogue inputs. The test was repeated after attaching the disk load described in Section 5.4.1. The two sets of speed data ($\omega_1$=$\omega_2$=45rad.s$^{-1}$) were plotted against each other and then the simultaneous equations discussed in Section 2.8.1 were solved.

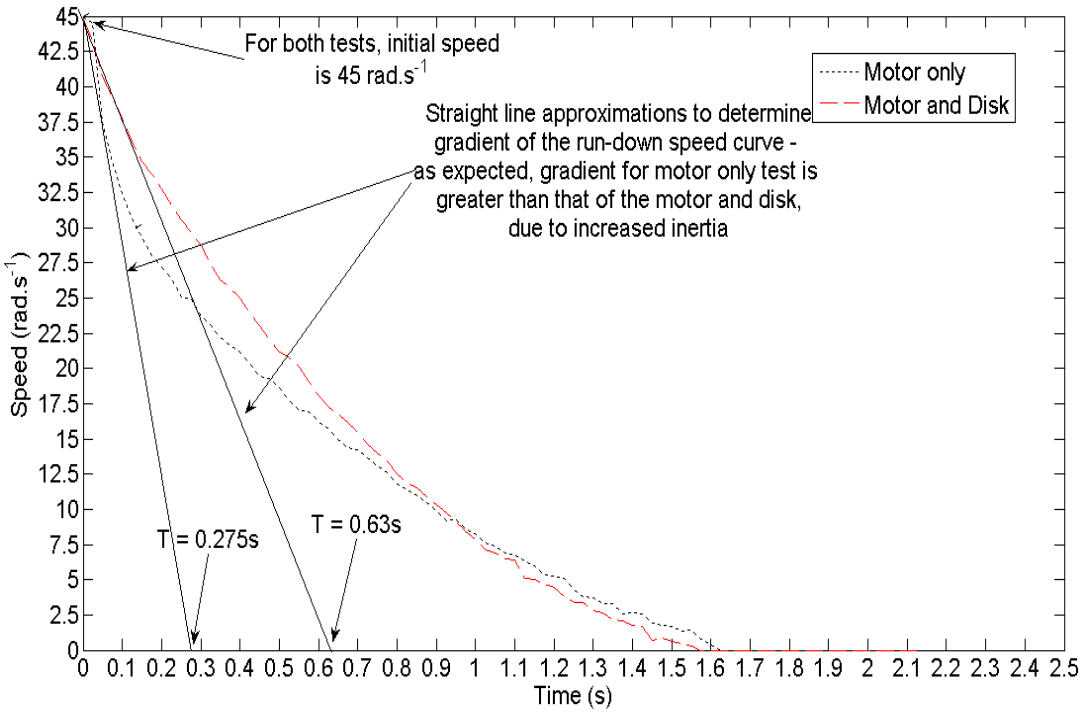



Figure 5-6 Retardation test results

It was approximated that

$$\frac{d\omega_1}{dt_1} = \frac{45-0}{0-0.275} = -163.64\omega.s^{-1}$$

and

$$\frac{d\omega_2}{dt_2} = \frac{45-0}{0-0.63} = -71.43\omega.s^{-1}.$$

Solving Equation (2.22) and Equation (2.23) using the value $J_{disk} = 3.387 \times 10^{-3} kg.m^2$, the inertia of the motor was found to be

$$\begin{aligned} J_{motor}\frac{d\omega_1}{dt_1} + B\omega_1 &= 0 \\ (J_{motor} + J_{disk})\frac{d\omega_2}{dt_2} + B\omega_2 &= 0 \\ J_{motor}\frac{d\omega_1}{dt_1} - J_{motor}\frac{d\omega_2}{dt_2} - J_{disk})\frac{d\omega_2}{dt_2} &= 0 \qquad (5.3) \\ J_{motor}(-163.64) - J_{motor}(-71.43) - J_{disk}(-71.43) &= 0 \\ (-92.21)J_{motor} &= (-71.43)J_{disk} \\ J_{motor} &= 2.624 \times 10^{-3} kg.m^2 \end{aligned}$$

### 5.4.3 Rigid coupling

The rigid coupling that was used when coupling two motors together was also considered to contribute to the total inertia of the system.

Figure 5-7 Photograph of rigid coupling used to couple the two identical motors together

Although the shape of the coupling was not completely uniform, it was approximated as a hollow cylinder with an outside diameter of 25mm and an inside diameter of 16mm. Its mass was measured and found to be 1.33kg.

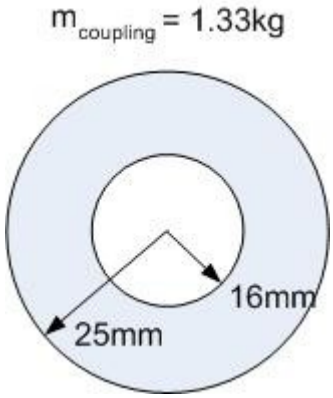



Figure 5-8 Diagrammatic representation of rigid coupling as a hollow cylinder with thick walls

For a hollow cylinder, the inertia is given by the expression

$$\begin{aligned} J_{coupling} &= \frac{1}{2} M_{coupling} (r^2{}_{inner} + r^2{}_{outer}) \\ &= \frac{1}{2}(1.33)((16\times10^{-3}m)^2 + (25\times10^{-3}m)^2) \\ &= 5.86\times10^{-4} kg.m^2 \end{aligned} \qquad (5.4)$$

### 5.4.4 Load Type 2

Load type 2 is best described by referring to Figure 5-9.

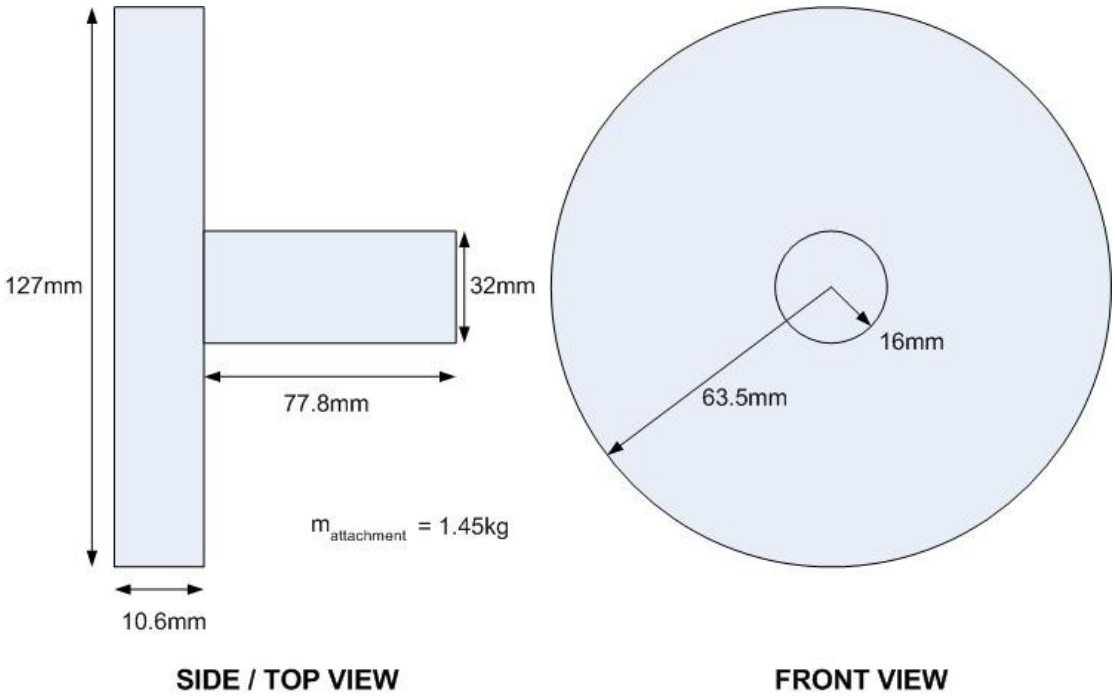



Figure 5-9 Top and Front View for Steel Attachment

For the purpose of determining its inertia, this attachment to the disk described in Section 5.4.1 may be considered as a combination of two cylindrical masses, one with radius 16mm, and the other with radius 63.5mm. Although the attachment was weighed and its mass was identified, in order to ascertain the inertia of the attachment, the masses of each of the two "sub-masses" that constitute this attachment is required. Therefore, using the density for this steel as established in Equation (5.2), the mass for each sub-mass was calculated.

$$\begin{aligned} m_1 &= \rho_1 v_1 \\ &= \rho_1 \pi r_1^2 h_1 \\ &= \pi (7367.83 kg.m^{-3})(16\times10^{-3} m)^2 (77.8\times10^{-3} m) \\ &= 0.461 kg \end{aligned} \tag{5.5}$$

$$\begin{aligned} m_2 &= \rho_2 v_2 \\ &= \rho_2 \pi r_2{}^2 h_2 \\ &= \pi(7367.83 kg.m^{-3})(63.5\times10^{-3}m)^2(10.6\times10^{-3}m) \\ &= 0.989kg \end{aligned} \tag{5.6}$$

The results for $m_1$ and $m_2$ are satisfactory, as their sum agrees with the combined mass of the attachment. Then, in order to work out the inertia of the attachment, the inertia of each sub-mass was established and summed.

$$\begin{aligned} J_{attachment} &= \frac{1}{2}m_1 r_1^2 + \frac{1}{2}m_2 r_2^2 \\ &= \frac{1}{2}(0.461kg)(16\times10^{-3}m)^2 + \frac{1}{2}(0.989kg)(63.5\times10^{-3}m)^2 \\ &= 2.05\times10^{-3} kg.m^2 \end{aligned} \tag{5.7}$$

### 5.4.5 Combination of motor and load

Since tests were run while coupling both the disk as well its attachment to the motor, it is left to show the total inertia against which the PRBS-based test estimate was compared. Note that in this case, the rigid coupling for which the inertia was calculated in Section 5.4.3 was not used to couple the motor to the load. The coupling used in this instances was considered to have a negligible effect on the total inertia.

$$\begin{aligned} J_{bigload} &= J_{motor} + J_{disk} + J_{attachment} \\ &= (2.624\times10^{-3} + 3.387\times10^{-3} + 2.05\times10^{-3}) kg.m^2 \\ &= 8.06\times10^{-3} kg.m^2 \end{aligned} \tag{5.8}$$

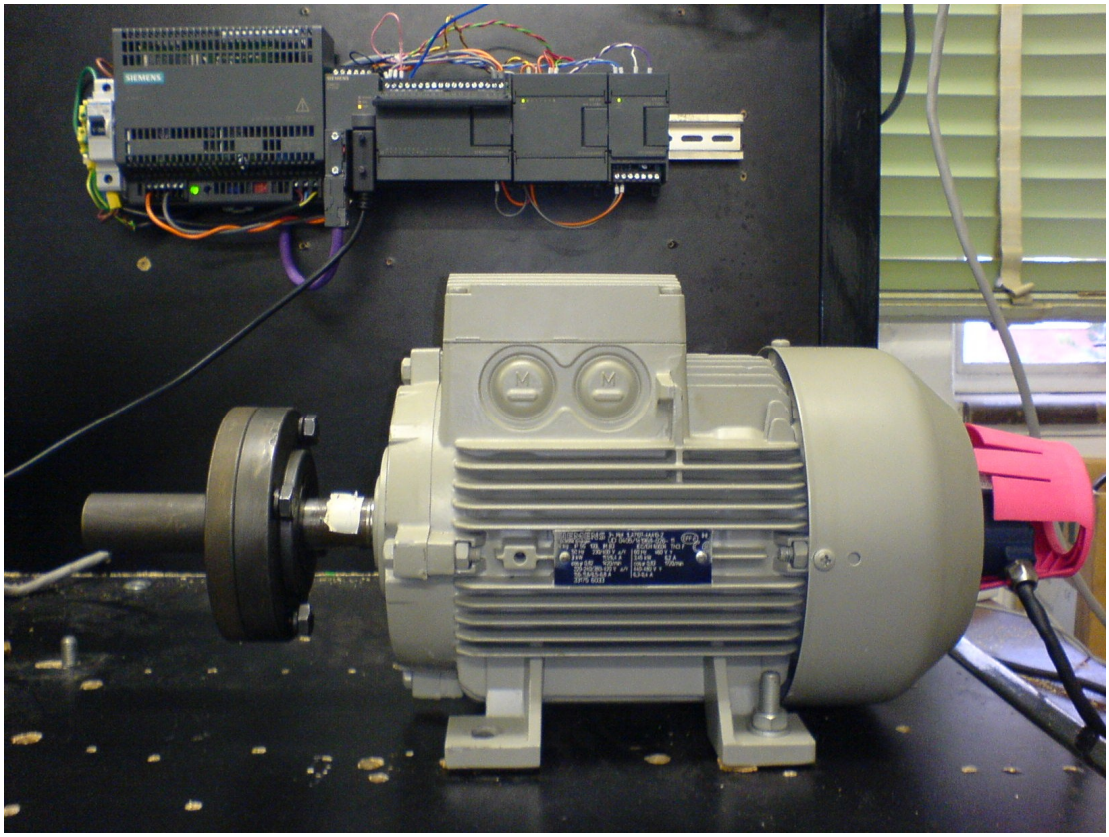


Figure 5-10 Photograph of motor and attached load. Also visible is the quadrature encoder for speed feedback as well as the PLC.

## 5.5 Results of Implementation on PLC

The following documents the results from testing on the practical rig using the PRBS-based tests. In order to verify the accuracy of the results produced by the PRBS-based tests, it was important to have an initial estimate for the motor and load inertias, and this was achieved by using an off-line test method – in this case the retardation test method discussed in Section 2.8.1, and for which calculations were done in Section 5.4.2, was used. The PRBS-based tests were then performed on the rig under no-load as well as with two different mechanical loads, and the mechanical parameters were estimated in each case. Important to note is that the focus in these practical tests was to identify the inertia in each test, as the inertia estimate could be compared with the estimate that results from the retardation tests. Due to difficulty in measuring the viscous friction coefficients $B_m$ and $B_L$ using other methods, the focus from here onward is the estimation of the total system inertia $J_T$. Initially, a rigid coupling was used between the motor and the load. Thereafter, the rigid coupling was replaced with a

coupling that was flexible, and further with couplings that had been tampered with to deliberately introduce backlash in the coupling.

As discussed, the PRBS signal was generated by the PLC and added as a torque perturbation to the system. The speed of the motor was sampled by an encoder and the correlation was computed in the PLC. The scaling of the numerical impulse response was done in MATLAB.

### 5.5.1 Motor With No Load

The first tests were done on a stand-alone induction motor, the inertia of which was identified as $J_{motor} = 2.624 \times 10^{-3} kg.m^2$. The method used to identify the inertia was discussed in Section 5.4.2. The scaled impulse response is displayed in Figure 5-11.

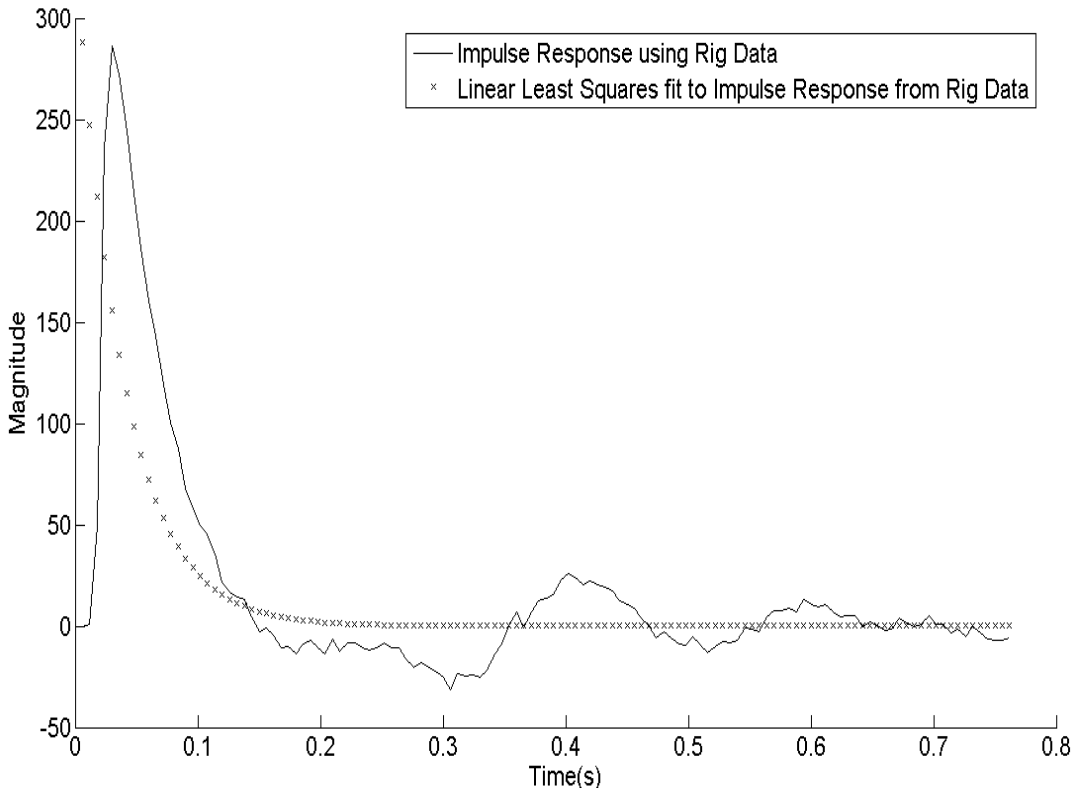



Figure 5-11 Impulse response of motor with no load

The parameter estimation algorithm estimated $J_m$ as $2.97 \times 10^{-3} kg.m^2$, an error of 13.2%.

### 5.5.2 Motor Coupled to Load Type 1 (Identical Motor)

Tests were then run with loads coupled to the motor. The first load that was used was another motor, of identical model to the first motor. In theory, this motor has the same inertia as the first motor.

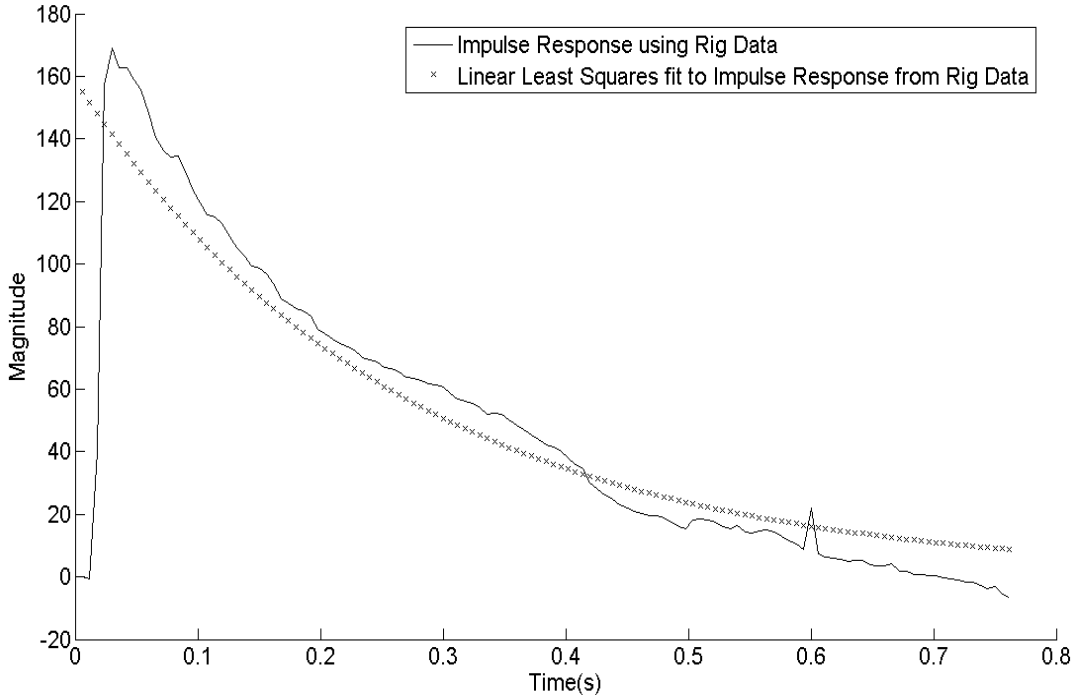



Figure 5-12 Impulse response of motor with an identical motor as an additional load

The resulting impulse response may be seen in Figure 5-12. The parameter estimate for $J_{total}$ was $6.4\times10^{-3} kg.m^2$. When comparing this estimate to the anticipated total inertia $J_T$ ($5.248\times10^{-3} kg.m^2$), the error was found to be 22%. However, this assumption neglected to account for the rigid coupling between the two motors.

After incorporating the calculated estimate for the inertia of the coupling (see Section 5.4.3), the combined inertia of the two motors and the coupling is approximately

$$\begin{aligned} J_{two\_motors} &= J_{motor1} + J_{motor2} + J_{coupling} \\ &= 2.624\times10^{-3}\,kg.m^2 + 2.624\times10^{-3}\,kg.m^2 + 5.859\times10^{-4}\,kg.m^2 \\ &= 5.83\times10^{-3}\,kg.m^2 \end{aligned} \tag{5.9}$$

By comparing the estimate from the numerical impulse response ($6.4\times10^{-3}\,kg.m^2$) to $J_T$ from Equation (5.9), the error in the estimate for $J_T$ was found to be 9.8%.

### 5.5.3 Motor Coupled to Load Type 2

The last test to identify inertia was performed using a single motor and a load comprising a disk as well as an attachment as described in Section 5.4.4, and illustrated in Figure 5-9. The impulse response is shown in Figure 5-13.

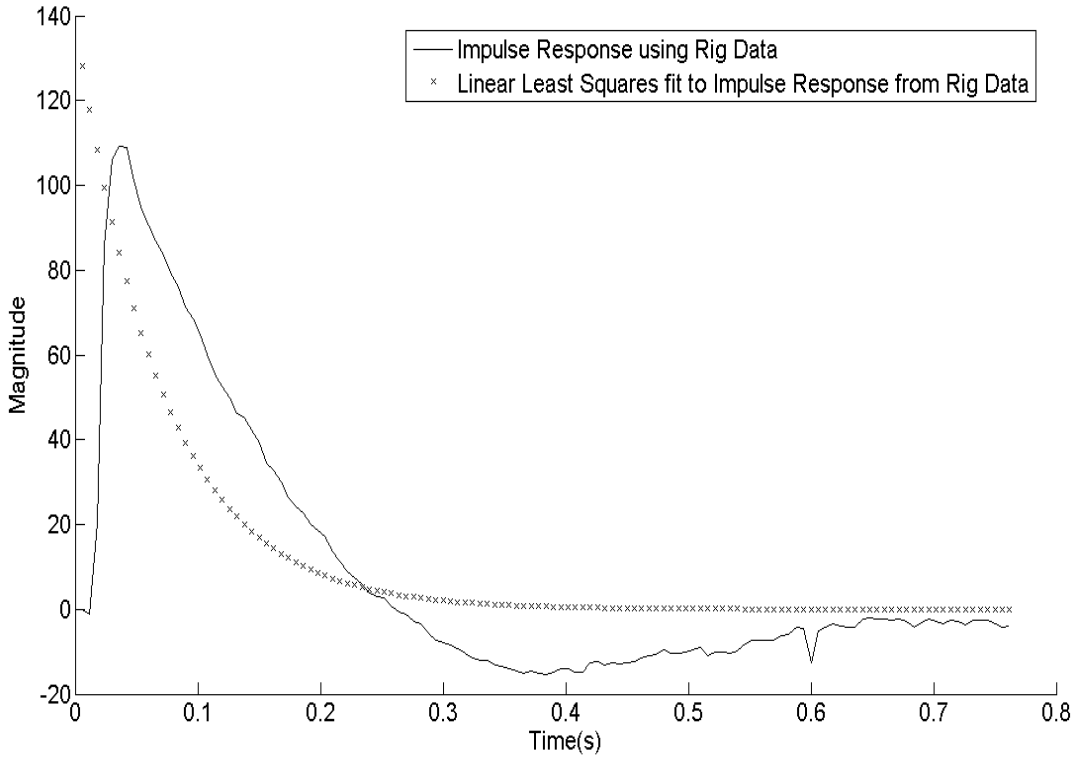



Figure 5-13 Impulse response of motor coupled to large load

The average value for the inertia estimate after numerous tests on this system with different values for *A*, $T_B$ and *N*, was $J_T = 8.31\times10^{-3} kg.m^2$. The calculated total inertia was determined as follows,

$$\begin{aligned} J_{bigload} &= J_{motor} + J_{disk} + J_{attachment} \\ &= (2.624\times10^{-3} + 3.387\times10^{-3} + 2.05\times10^{-3}) kg.m^2 \\ &= 8.06\times10^{-3} kg.m^2 \end{aligned} \qquad (5.10)$$

When comparing the estimated inertia from the numerical impulse response with the calculated inertia, as calculated in Equation (5.10), the error in the estimate was found to be 3.1%.

Table 5-1 Summary of results obtained using the PRBS-based test on a practical system

| **Load Type** | **$J_T$ error (%)** |
|---|---|
| No load | 13.2 |
| Second motor | 9.8 |
| Large disk | 3.1% |

The results shown in Table 5-1 show that, on a practical mechanical system, the PRBS-based tests produce satisfactorily accurate (≈10%) estimates for the total inertia of a mechanical system. The inertia parameter has received much focus since an alternative method to calculate inertia was readily available.

In Chapter 4, simulation results were shown that described the effects that certain non-linear phenomena such as shaft elasticity, as well as backlash have on the numerical impulse response.

A similar investigation was undertaken on the practical system to see how the results correlate with those examined in the simulations.

### 5.5.4 Non-rigid coupling –Shaft Elasticity

In order to include the effect of shaft elasticity on the numerical impulse response, a tire coupling was used to couple two motors – a flexible rubber coupling that deliberately introduced elasticity into the system. Figure 5-6 is a photograph of this coupling.

Figure 5-14 Photograph of tire type coupling used to emulate a coupling with elasticity or flexibility

The resulting impulse response curve is displayed in Figure 5-15. Note that no scaling was done on the impulse response data, as the aim of this test was not to estimate the parameters, but rather to identify any characteristic signatures in the impulse response curve that would correspond to the increase in the elasticity in the system.

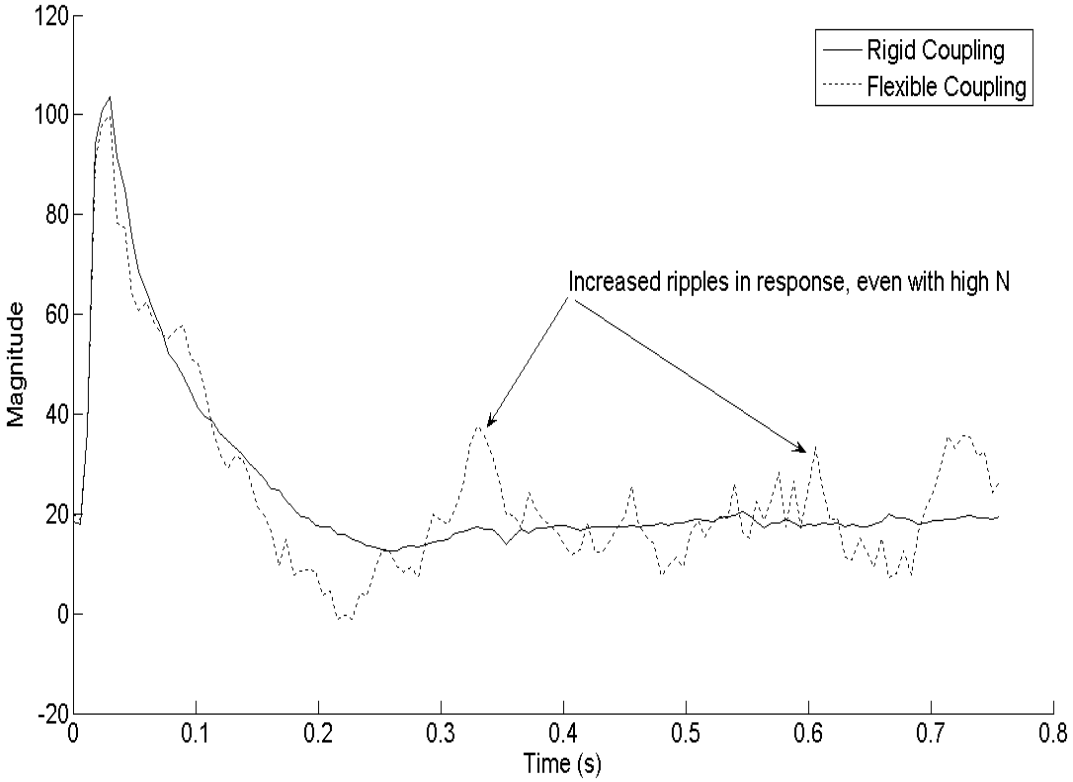



Figure 5-15 Comparison between impulse responses of system with a rigid shaft coupling and with a flexible coupling

In Section 4.5.1, the simulations presented showed that as the shaft elasticity increased, ripples appeared in the impulse response.

When comparing the impulse response from the practical implementation (shown in Figure 5-15) with the corresponding simulation, the results of which are displayed in Figure 4-18, a similar result is exhibited. Therefore, this could be a characteristic to look out for when performing these tests to allude to the presence of increasing elasticity in a mechanical system.

### 5.5.5 Backlash in the Coupling

In order to analyse the effect of backlash in the coupling on the impulse response, two motors were coupled with a coupling that would allow for play in the coupling. This coupling was metal with a rubber insert. As supplied by the manufacturer, the rubber insert provided a close coupling, with virtually no play. However, the rubber was then shaped, or cut, to ensure that it provided only a loose fit between the metal halves of the coupling, thus

introducing backlash into the system. Refer to Figure 5-16 and Figure 5-17 for photographs of the inside of this coupling, both with a new rubber insert, as well as with the shaped insert.

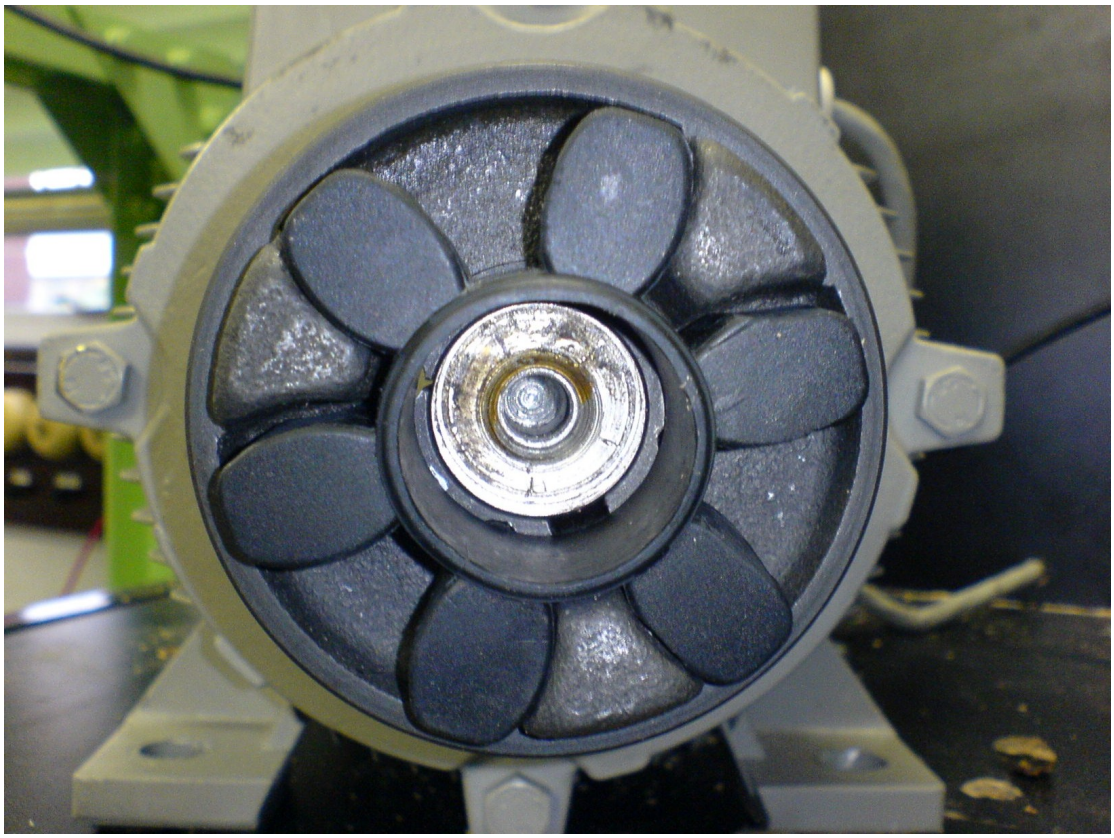


Figure 5-16 Photograph showing metal coupling with a rubber insert, with new rubber insert that provides for minimal gear play

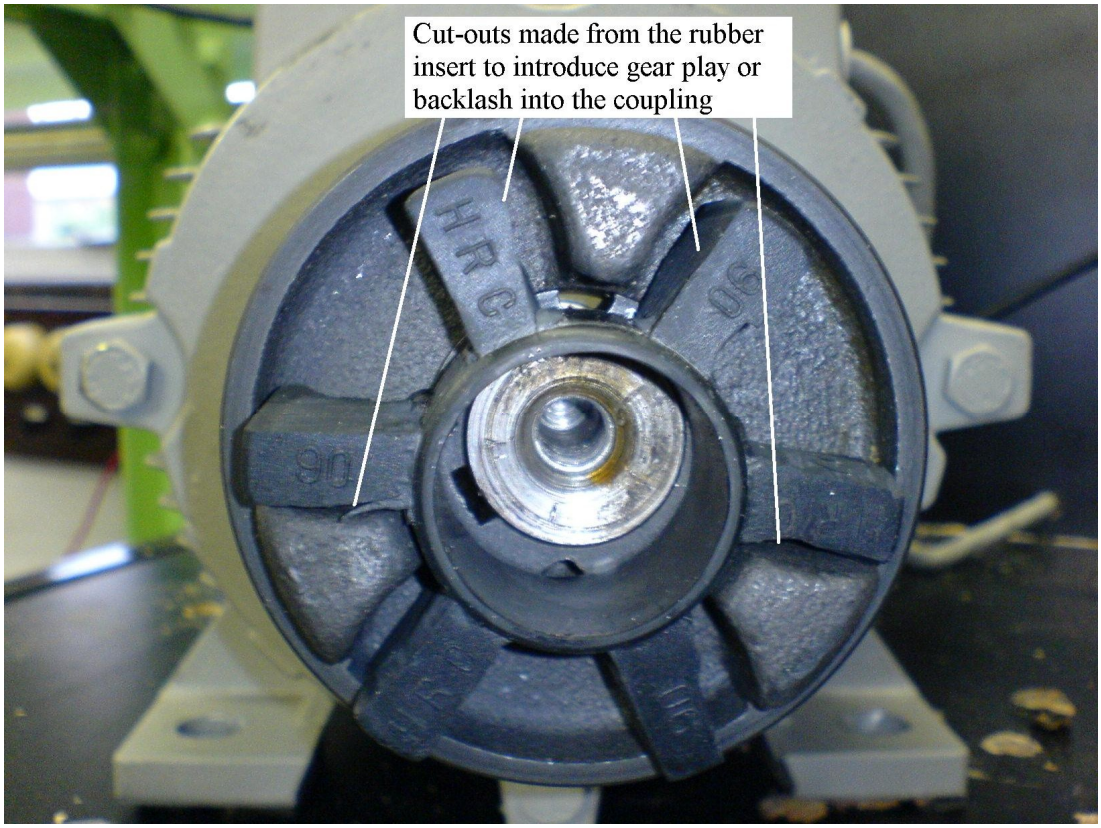



Figure 5-17 Photograph showing the cut-outs made from the rubber insert to deliberately introduce backlash in the coupling

Two such inserts were shaped, to produce two instances of backlash, one that is medium and one that is severe. In the case of the coupling that exhibits severe backlash, the rubber insert was deliberately shaped so as to cause an extreme deterioration in the coupling, in order to enhance the results.

The impulse response for the coupling with severe backlash was plotted against the impulse response for a rigid coupling, and is shown in Figure 5-18.

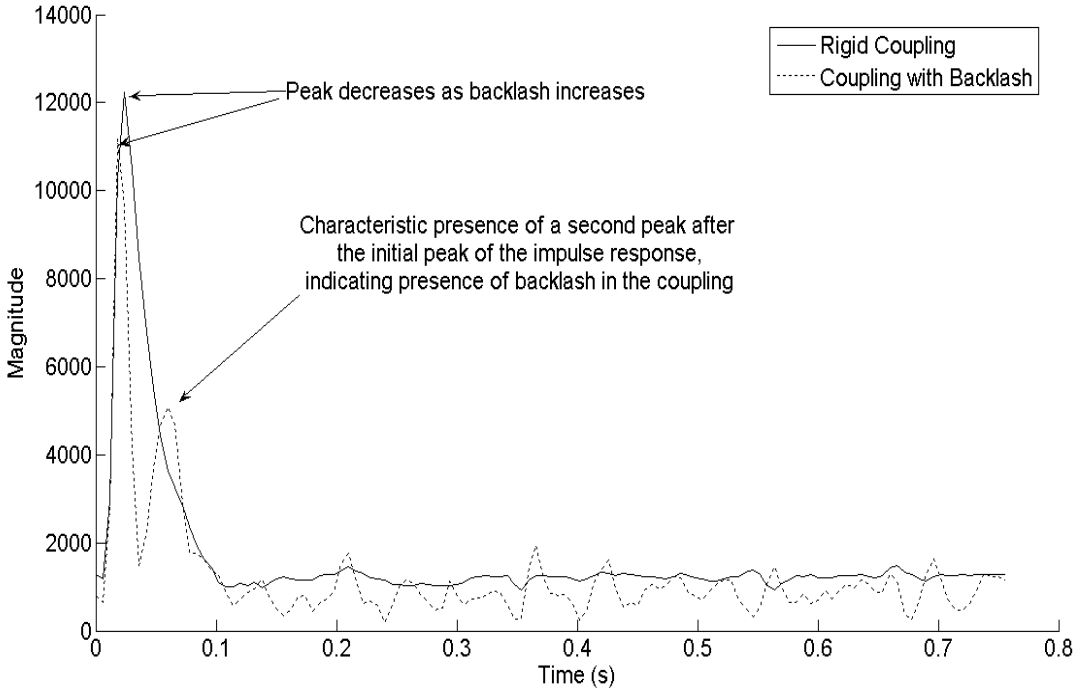



Figure 5-18 Impulse responses for a rigid coupling, and for a severe case of backlash

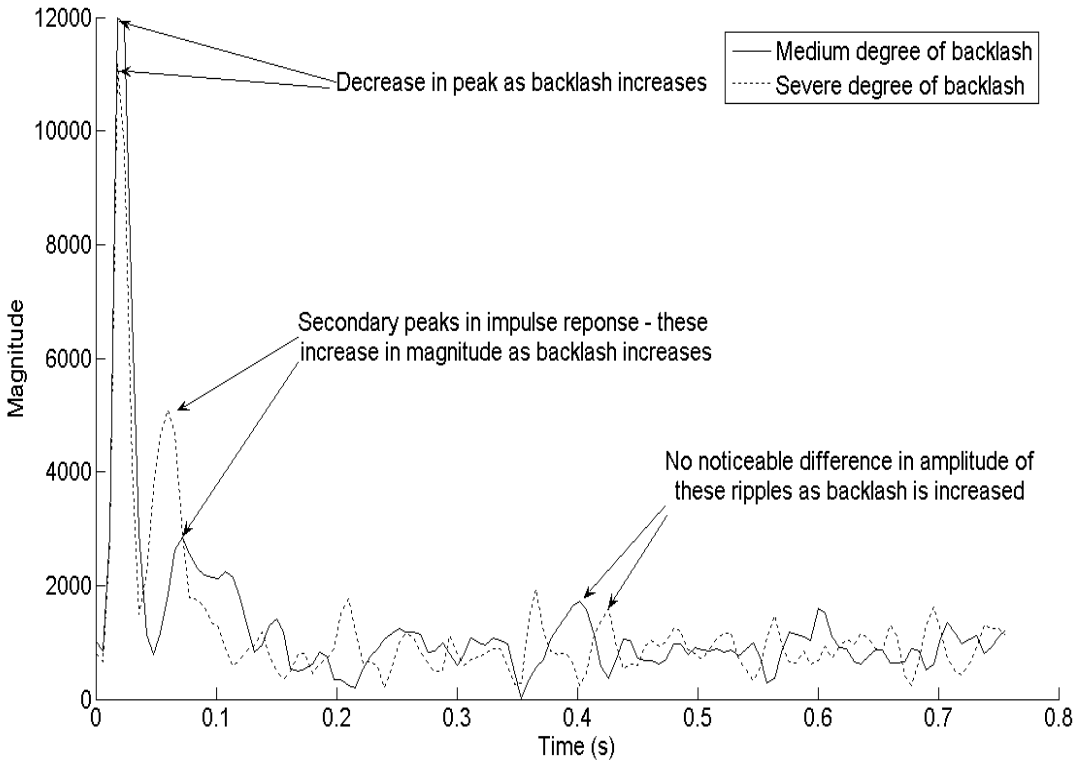



Figure 5-19 Comparison of impulse response for a coupling that has exhibits medium backlash, and one that exhibits severe backlash

Figure 5-19 compares the impulse responses for the coupling with medium backlash and the coupling with severe backlash.

Once again, the impulse responses from the practical implementation concur with those that were obtained by simulation (see Figure 4-20); note the presence of secondary peaks, or ripples, in the impulse response, other than the initial peak, when backlash is present. The amplitude of these peaks, which only appear during the initial part of the impulse response, increases as backlash increases.

It is also seen that the initial peak of the impulse response decreases with an increase in backlash. Also to be noted is the fact that the ripples in the response as the response settles are of approximately the same amplitude irrespective of the degree of backlash.

These factors may be used to identify the presence of backlash on an online system by using the PRBS-based tests discussed in this research.

## 5.6 Summary

This chapter has discussed the implementation of the PRBS-based tests on a practical motor-based system. The configuration of the system, as well as the algorithms used to run the on-line tests, to perform the correlation, to scale the resulting impulse response, and finally to estimate the mechanical system parameters, namely inertia, have been outlined. Due to difficulty in determining the viscous friction coefficient $B_T$ from other methods, so as to use these results to compare the estimated parameters against, it was decided to focus primarily on providing parameter estimations for the inertia of the system.

The results from this practical implementation were shown to be satisfactory within the scope of this research. This research may be taken further by trying to apply these tests to provide more accurate estimates for the mechanical system parameters. Furthermore, the identification of signature effects in the impulse response curve in the presence of elasticity and backlash was discussed and the results were shown to agree with observations made from simulations.

# CHAPTER 6
# CONCLUSION

## 6.1 General

As mentioned at the outset, a sound knowledge of a system's characteristics is essential. Numerous parameter estimation techniques are available to identify systems, but certain tests such as offline ones are not feasible in industrial applications that require availability all the time. It was suggested that in order to comprehensively and accurately perform online system identification, a few online tests should be chosen that complement each other in providing an overview of the system under test.

One such test that will provide invaluable information is the correlation-based test using PRBS signals as input perturbations to the system under test. The main objective of this dissertation was to investigate the use of correlation techniques to perform system identification tests, with particular application to mechanical parameter estimation as well as machine diagnostics. Detailed simulations were presented and analysed to verify the feasibility of this parameter estimation technique. The method was then tested on a practical test platform, and the results were compared to those of the simulations performed.

The conclusions of each of the main sections in this dissertation are summarised below.

## 6.2 Mechanical System Modelling

Chapter 2 discussed the modelling of a DC motor, with specific attention given to the mechanical system component. The mechanism of torque control in a DC motor was compared against that of torque control in an induction motor, with important differences between the two motor types highlighted. The concept of field oriented control for an induction motor was described and attention was drawn to the fact that an induction motor under field oriented control exhibits behaviour characteristic of a DC motor. Parameter estimation techniques were considered, with a differentiation made between offline and online techniques. Alternate methods of inertia identification were examined, as these were later implemented during the practical implementation of the correlation-based tests, and used to provide a benchmark against which to compare the result produced by the correlation-based tests.

## 6.3 A Correlation-Based Method for Online System Identification

Chapter 3 introduced correlation theory and it was shown mathematically that by taking the correlation of a system's output and input, the correlation result effectively yields the system's response to the autocorrelation of the input signal. It was proven that, by using a signal with an impulsive autocorrelation, the correlation between input and the system output produces the numerical impulse response of the system. Pseudorandom binary sequences were shown to possess the characteristics required to determine the impulse response of a system using the correlation technique.

The theory behind extracting a system's parameters from its impulse response was examined. A series of simulations were shown to verify the result that the correlation does indeed produce the impulse response, and it was also demonstrated that varying the characteristics of the PRBS sequence has an effect on the resulting numerical impulse response. This relationship between the PRBS sequence parameters and the impulse response necessitated proper scaling of the numerical impulse response in order to get the system's response to a unit impulse, and these scaling factors were outlined.

The method of linear least squares was used to perform a first order curve fit before parameter estimation was performed. The error between the estimate and the actual value was shown to be small enough (<10%) to make this PRBS-based testing method suitable for the purposes of parameter estimation for this research.

## 6.4 Simulated Implementation

Chapter 4 included the results of simulations performed to verify the suitability of using the PRBS-based tests in order to identify the parameters of a mechanical system. The simulations were classified into systems without a load (motor only) and systems with a load. Furthermore, experiments where the PRBS signal was added both as a torque perturbation and as a speed perturbation were simulated, and the differences between these two methods were outlined, as well as the results from both sets of simulations.

Results were displayed and discussed for systems under no load (motor only) as well as under load conditions. These simulations were performed using either of the PRBS application alternatives: as a torque perturbation with the speed loop open, as a torque perturbation with the speed loop closed, and as a speed perturbation with the speed loop closed. It was concluded that these testing methodologies are all viable, with the closed loop tests being necessary for the tests to be administered on an online system; however this method was shown to require *a priori* knowledge of the controller parameters. It was

explained that this is possible because modern VSDs allow access to the aforementioned parameters.

The simulated responses under no load conditions produced estimations for $J_m$ and $B_m$ with an error of less than 10% in all cases. These errors were shown to decrease as one varied $T_B$, with small values of $T_B$ providing a high resolution of data points around the peak of the impulse response, and hence high accuracies for the estimation of $J_m$; while large values of $T_B$ were shown to produce high accuracies in the estimation of the gain of the system, and subsequently for the $B_m$ term.

When a load was added to the system it was shown, by means of simulations, that the assumption was correct that the motor inertia and load inertia may be lumped into a total system inertia parameter $J_T$, and likewise that the motor viscous friction coefficient and load viscous friction coefficient may be added together as the total system viscous friction coefficient $B_T$. This was done by modelling a coupling and mechanical load, and comparing the impulse response of the simulated motor-load implementation, against a motor only implementation, where the motor and load inertias and viscous friction coefficients were lumped into the motor.

An investigation into the presence of mechanical imperfections and their effect on the numerical impulse response was also undertaken. Qualitative results were shown for the signatures that one can identify from the impulse response in the event that the coupling is excessively flexible, and for cases where there is excessive gear play in the coupling.

## 6.5 Practical Implementation

In Chapter 5, the implementation of these tests on an automation system comprising a PLC, a VSD, a speed encoder, and a motor-based mechanical system, was presented. The configuration of the test platform as well as the algorithms that were written to produce the PRBS signals as well as sample the system output and perform the system correlation were also discussed. Preliminary results were examined, and estimations for the system inertia under various inertial loads were presented. Measured data from systems with deliberately induced elasticity and backlash were displayed and compared to simulation results.

The results from this practical implementation were shown to agree with the results obtained in the simulated implementation. Furthermore, the identification of signature effects in the

impulse response curve in the presence of elasticity and backlash was discussed and the results were shown to correlate with simulations.

## 6.6 Suggestions for Future Work

This dissertation presents only the initial work done toward mechanical system identification using PRBS signals as input signals. A number of simplifying assumptions were made in order to limit the scope of the research, and future work should attempt to address these assumptions. Some of the aspects that future work could include are:

- Including PRBS generation in the PLC so that various length PRBS sequences may be used for testing purposes – it was stated in Chapter 5.3 that *L*-length base PRBS sequences were generated using Matlab and then stored as constants in a data block in the memory of the PLC – these base sequences were then scaled by the amplitude *A* in the PLC, and outputted during testing *N* times.

- Including the parameter estimation algorithms in the PLC so that the correlation data need not be exported to a PC for computation – while the actual correlation was done in the PLC, the correlation result needed to be exported to Matlab to run the linear least squares algorithm on the data, and to do the necessary scaling.

- Including a Human Machine Interface (HMI) panel so that tests may be performed without the need for a PC, thus making the test rig more mobile, such as for use in online testing in industrial applications

- Accurately estimating inertia and the viscous friction coefficient even in the presence of elasticity and backlash

- Providing a quantitative measure of the degree of elasticity or backlash in the mechanical system

# REFERENCES

[Amann1] N. Amann, J. Böcker, F. Prenner, "Active Damping of Drive Train Oscillations for an Electrically Driven Vehicle", *IEEE/ASME Transactions on Mechatronics, Vol. 9, No.4,* Dec 2004

[Beineke1] S. Beineke, F. Schütte, H. Grotstollen, "Comparison of Methods for State Estimation and Online Identification in Speed and Position Control Loops", *EPE 1997*, Trondheim, Norway

[Beineke2] S. Beineke, H. Wertz, F. Schütte, H. Grotstollen, N. Fröhleke, "Identification of Nonlinear Two-Mass Systems for Self-Commissioning Speed Control of Electrical Drives", *Paper ID: SS-001/3*

[Benn1] L. C. Benn, B. Burton, R. G. Harley, "PWM Spectrum Analysis and Optimization for Model Based Condition Monitoring and Control", *IEEE Industry Applications Society Annual Meeting*, Chicago, Illinois, Oct. 2001, pp CDROM.

[Benn2] L. C. Benn, B. Burton, R. G. Harley, "Online Converter fed Induction Motor Impulse Response Identification and Parameter Extraction using Pseudorandom Modified PWM", *Conf. Rec. IEEE SDEM*PED, Aug. 2003, pp 70-75.

[Benn3] L. C. Benn, "Advanced PWM Algorithm Development for Online VSI Fed Induction Motor Parameter Estimation", *Masters Thesis - University of Natal*, Durban, South Africa, March 2003

[Boje1] University of KwaZulu-Natal, School of Electrical, Electronic, Computer Engineering Practical Handout for course Control Systems 2 (DNE4CN1), "*CN7 – System Identification"*

[Burton1] B. Burton, "Analysis and Practical Implementation of a Continually Online Trained Artificial Neural Network to Identify and Control VSI fed Induction Motor Stator Currents", *Masters Thesis - University of Natal*, Durban, South Africa, July 1995

[Cremer1] K. Cremer, M.M. Michaelis and M. Kuppen, "Random-Modulation Continuous Wave LIDAR", Proceedings of the International Workshop on Modern Physics and Laser Applications, Harare, Zimbabwe, Sep. 1993.

[Diana1] G. Diana, R. G. Harley, "An Aid for Teaching Field Oriented Control Applied to Induction Machines", *IEEE Transactions on Power Systems, Vol. 4, No. 3*, Aug. 1989, pp 1258-1261

[DeKock1] J. A. de Kock, F. S. van der Merwe, H. J. Vermeulen, "Induction Motor Parameter Estimation Through an Output Error Technique", *IEEE Transactions on Energy Conversion, Vol. 9, No. 1,* March 1994, *pp 69 - 75.*

[DeGraauw1] Th. De Graauw, B. P. Th. Veltman, "Pseudo-Random Binary Sequences for Multiplex Codes", *Applied Optics, Vol. 9, No. 12*, Dec. 1970, pp 2658-2660

[Eitelberg1] E. Eitelberg, "Control Engineering – Course Notes", *NOYB Press*, 2001

[Jackson1] P. A. Jackson, "PRBS Cross-Correlation Measurements by Hybrid Computational Techniques", *The Computer Journal*, Vol. 14, No.1, Dec. 1969, pp 49-54

[Kara1] T. Kara, İ. Eker, "Experimental Nonlinear Identification of a Two Mass System"

[Kleinhans1] C. E. Kleinhans, "Simulation and Practical Implementation of Field Oriented Control on the Current Source Inverter-Fed Induction Machine", *Masters Thesis - University of Natal*, Durban, South Africa, May 1995

[ODonovan1] J. G. O'Donovan, R. C. Kavanagh, J. M. D. Murphy, M. G. Egan, "Modular Approach to Parameter Estimation in Geared and Linear Resonant Systems", *Control Engineering Practice 12 (2004)*, pp 75-86

[Ogata1] K. Ogata, "Modern Control Engineering", *Prentice Hall Inc USA*, 1990

[Rampersad1] V. Rampersad, B. Burton, "PLC Implementation of Online, PRBS-Based Tests to Identify Impulse Response of a Mechanical System", *Fifteenth Southern African Universities' Power Engineering Conference*, Durban, Jan. 2006 pp CDROM

[Rubin1] N. P. Rubin, "Slip Estimation for Induction Motors in Speed Control Applications with Particular Reference to Sensorless Field Oriented Controlled Drives", *Masters Thesis – University of Natal*, Durban, South Africa, September 1990

[Saloma1] C. Saloma, A. Jose de Vera, "Photoacoustic by Cross-Correlation Using a GaAs Light Emitting Diode", *Applied Optics, Vol. 30, No. 17,* 10 June 1991, pp. 2393-2397

[Schütte1] F. Schütte, S. Beineke, A. Rolfsmeier, H. Grotstollen, "Online Identification of Mechanical Parameters Using Extended Kalman Filters", *IEEE IAS Annual Meeting,* New Orleans, Louisiana, Oct 1997

[Siemens1] "S7-200 Programmable Controller System Manual", *Siemens AG,* June 2004

[Tallfors1] M. Tallfors, "Parameter Estimation and Model Based Control Design of Drive Train Systems", *Kungliga Tekniska Högskolan, Stockholm, PhD Thesis,* 2005

[Taub1] H. Taub, D. L. Schilling, "Principles of Communication Systems", *McGraw-Hill International Edition,* 1986

[Vas1] P. Vas, "Parameter Estimation, Condition Monitoring, and Diagnosis of Electrical Machines", *Oxford University Press,* 1993, pp 279-288

[Wertz1] H. Wertz, F. Schütte, "Self-tuning Speed Control for Servo Drives with Imperfect Mechanical Load", *IEEE IAS Annual Meeting*, Rome, Oct. 2000

# APPENDIX A

# Matlab Program Listings and Simulation MODELS

## A.1 General

This appendix contains the Matlab program listings as well as Simulink models that were used to perform the simulations used in the research for this dissertation. Table A-1 summarises the names of the script files as well as their associated descriptions.

Table A-1 Script / Model filenames and descriptions

| Section | Program / Model Name | Description |
|---|---|---|
| A.2 | prbs_gen.m | Used to generate PRBS sequences for larger order sequences. Used under permission of original author [Benn3]. |
| A.3 | prbs_gen_export_Simulink.m | Generates the PRBS sequence as per the parameters configured by the user, as well as configures the motor and load parameters for the Simulink model. Executed before each simulation. |
| A.4 | corr25nov.m | Function that performs the mathematical correlation between the two input vectors passed to the function. |
| A.5 | analyse_results.m | Script to call the correlation routine, scale the resultant data appropriately, and then estimate the parameters. Furthermore, performs linear least squares curve fitting on the data and then estimates the parameters again to the fitted data. If the system under test is closed loop, this script file provides the necessary compensation before |

| | | |
|---|---|---|
| | | performing parameter extraction |
| A.6 | DC_motor_no_load.mdl | Model used for a simple DC motor only |
| A.7 | DC_motor_with_load.mdl | Extended model to include the effect of adding a load to a motor. Includes the inertia and viscous friction coefficient of the new load, as well as the elasticity and viscous friction coefficient of the shaft or coupling. Was also extended to include the backlash (deadband) element for the backlash simulations. |
| A.8 | PRBS_in_Torque.mdl | Model showing how the PRBS signal is added as a torque perturbation |
| A.9 | PRBS_in_Speed.mdl | Model with PRBS added as a speed perturbation, with the speed loop closed. |

## A.2 prbs_gen.m

```
%*****************************************************

%This function was authored by Lance Benn and used by

%Vaughan Rampersad under the authorisation of the

%original author


function [prbs,L]=prbs_gen(a,n,fb)

reg0=ones(1,n);

L=2^n-1;
```

```
reg=reg0;

for i=1:L

  in=xor(reg(n),reg(fb));

  [reg,dummy]=shift(in,reg);

  prbs(i)=in;

end

prbs=2*a*prbs-a;



function [reg_new,out]=shift(in,reg)

n=length(reg);

reg_new=zeros(1,n);

reg_new(1)=in;

reg_new(2:n)=reg(1:n-1);

out=reg(n);

%*****************************************************
```

## A.3 prbs_gen_export_simulink.m

```
%*****************************************************

% this file generates a PRBS vector of amplitude A,

% length L, bit duration Tb. N is used to repeat the sequence

% The motor mech parameters are imported from the workspace as
b_real and j_real

% The load parameters are imported from the workspace as b_load and
j_load
```

```
% The coupling elasticity and viscous friction coefficient are
imported

% from the workspace as Csh and Bsh respectively

% The motor elec parameters are imported from the workspace as
L_real and R_real

% The motor constants are imported from the workspace as k1 and k2

% The controller parameters are imported from the workspace as GT
and GS



% The PRBS parameters are configured below

A=1;

N=500;

L=127;

Tb=0.1;

Ti=Tb*L;

Ls=N*L;

Ts = N*Tb*L;

if L==3

    prbs_in = [1 -1 1];

elseif L==7

    prbs_in = [-1 1 -1 -1 1 1 1];

elseif L==15

    prbs_in =  [-1 1 1 1 -1 1 1 -1 -1 1 -1 1 -1 -1 -1];

elseif L==63
```

```
    prbs_in=prbs_gen(1,6,1);
elseif L==127
    prbs_in = prbs_gen(1,7,1);
end
prbs_seq = [];
for count=1:N
    prbs_seq = [prbs_seq,prbs_in];
end
prbs_t=[];
for interval =1:(length(prbs_seq))
    prbs_t = [prbs_t,interval];
end
prbs_exp=[Tb*prbs_t',A*prbs_seq'];
%*******************************************************
```

## A.4 corr25nov.m

```
%*******************************************************
% Calculating cross correlation
function corr_xy=corr25nov(y,x,NNN,LL)

if length(y)<length(x)
    y((length(y)+1):length(x))=0;
end
```

```
lx=length(x);

corr_xy(1) = sum(y.*x);


for i=1:NNN-1


    corr_xy(i+1)=sum(y.*[x(lx-i+1:lx),x(1:lx-i)]);

end


corr_xy=corr_xy/LL;

%*******************************************************
```

## A.5 analyse_results.m

```
%*****************************************************************

% This program is used to perform the correlation on the simulated
data and

% then the necessary scaling to achieve the numerical impulse
response. A

% linear least squares curve is then fitted to the data, and
parameter

% estimation is performed against the fitted data.  The accuracy of
the
```

```
% estimation is determined by comparing the estimated parameters
with the

% known parameters.  In instances where the speed loop is closed, a

% comparison to the compensated system is made, i.e. the comparison
takes

% into account the affect of the controller parameters on the system

% response



% Do Correlation

corr_xy=zeros(N,L);

corr_main=zeros(1,L);



aa=[];

bb=[];



for loopc = 0:(N-1)

    aa=yy(loopc*L+2:loopc*L+L+1)';

    bb=p(loopc*L+2:loopc*L+L+1)';

    corr_xy (loopc + 1,:)= corr25nov(aa,bb,L,L);

end



if N~=1

  corr_main=sum(corr_xy)/N;

end
```

```
if N==1

  corr_main=corr_xy;

end


clc


%Determine ideal impulse response


ppp=tf([1],[j_real b_real]);




tt=0:Tb:Tb*L;

imp=impulse(ppp,tt);




%Scaling the numerical impulse




scaled_imp_fin=corr_main./(A^2*Tb);
```

```
scaled_imp_fin=scaled_imp_fin + abs(scaled_imp_fin(1));



%Extract parameters before curve fitting

[peak3 peak_index3]=max(scaled_imp_fin);



j_real;

j_fin = 1/peak3;

settling_fin = mean(corr_main(L-31:L));

b_real;



b=(-A^2/L)/settling_fin;

if c2==1
   b=(-A^2/L)/settling_fin - GS;
elseif c2==3
```

```
   j_fin = GS/peak3;

   b=GS*( (-A^2/L)/settling_fin - 1);

end









j_error = ( (j_fin-j_real)/j_real )*100

b_error = ( (b-b_real)/b_real)*100




% Do the linear least squares algorithm


clear M


clear tetha




time_vect=Tb:Tb:Tb*L;

dt=Tb;


response=scaled_imp_fin';
```

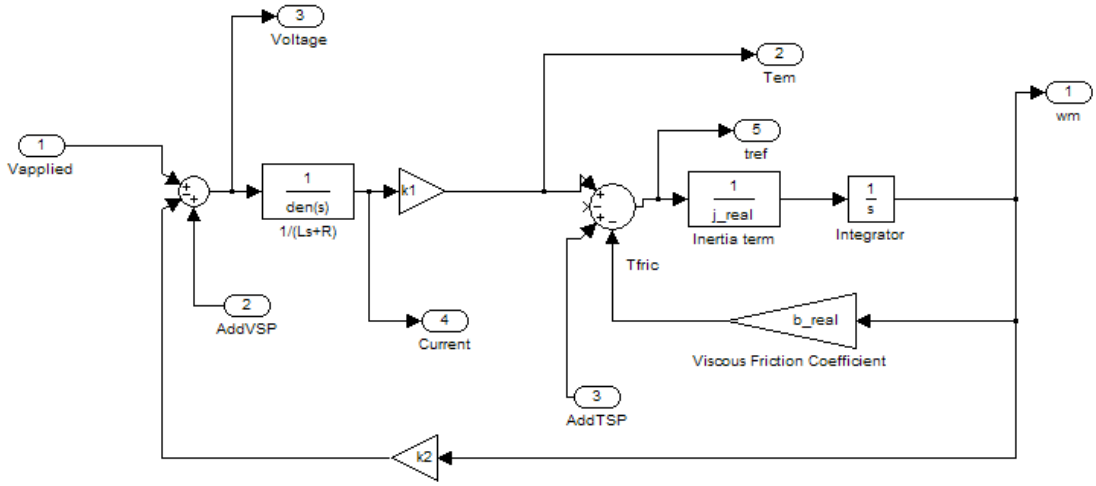
```
Step_Response = (cumtrapz (response).*dt);


FirstInt = -(cumtrapz (Step_Response).*dt); % function used to
numerically integrate


for index=1:length (FirstInt)                % Single integration,
and begin to populate M matrix
    M (index,1)= FirstInt(index);
    M (index,2)= (index-1)*dt;
end


Tetha = (M'*M)\(M'*Step_Response);           % Determine the tetha
parameter vector


num_lls = [Tetha(2)];
den_lls = [1 Tetha(1)];
p_lls = tf(num_lls, den_lls);
[Step_LLS, TVect]=impulse(p_lls,time_vect);


Imp_Plant = impulse( tf([1],[j_real b_real]), time_vect);


% Create compensating plants to compare curve fitted data against
```

```
if c2==1

    Comp_Plant   =   impulse(   tf([1],[j_real   (b_real   +   GS)]),
time_vect);

elseif c2==3

    Comp_Plant   =   impulse(   tf([GS],[j_real   (b_real   +   GS)]),
time_vect);

elseif c2==4

    Comp_Plant = impulse( tf([1],[j_real (b_real)]), time_vect);

end



%Parameter estimation taking into account controllers' parameters



%j_real

if c2==4

j_lls = 1/Tetha(2);

j_lls_error = (j_lls-j_real)/j_real*100



%b_real

b_lls = Tetha(1)/Tetha(2);

b_lls_error = (b_lls - b_real)/b_real*100



elseif c2 == 1

    j_lls = 1/Tetha(2);

    %j_lls = (GT/(1+GT))/Tetha(2);
```

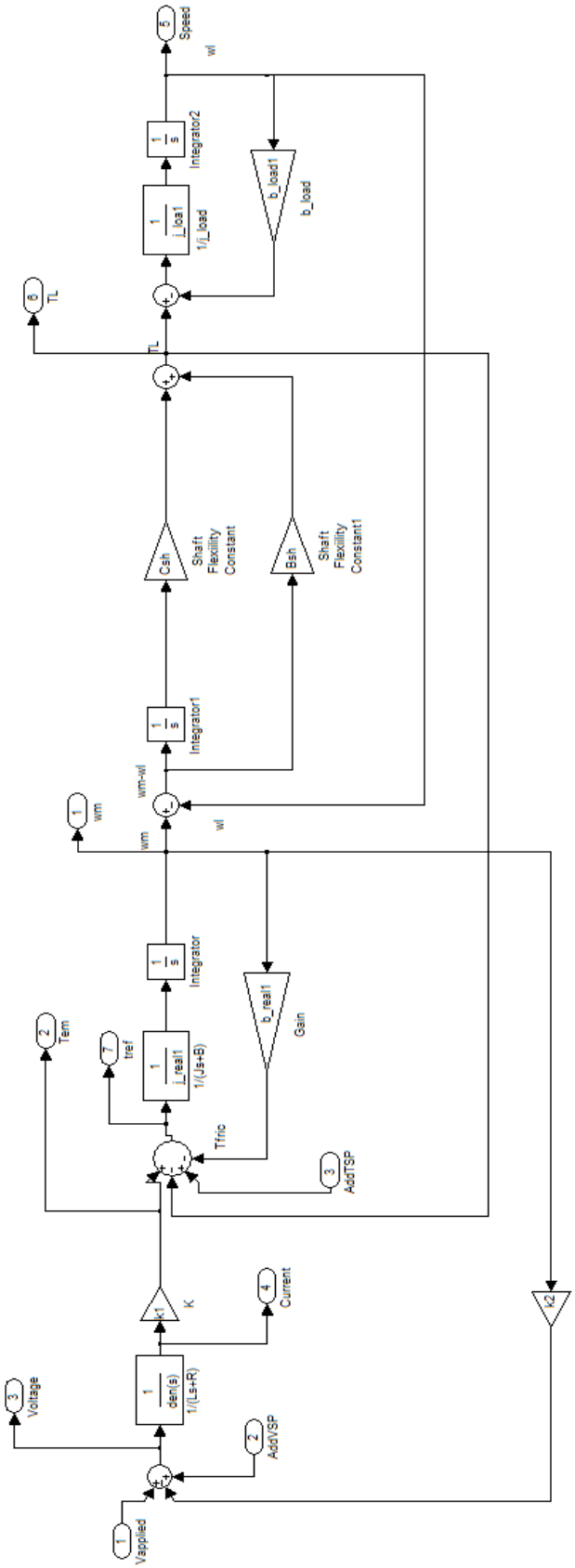
```
    j_lls_error = (j_lls-j_real)/j_real*100


%b_real
    b_lls = Tetha(1)/Tetha(2) - GS;
  % b_lls = (Tetha(1)/Tetha(2))*(GT/(1+GT)) - GS;
    b_lls_error = (b_lls - b_real)/b_real*100


    elseif c2 == 3
    %j_lls = GS/Tetha(2);
    j_lls = GS*GT/(Tetha(2)*(1+GT));
    j_lls_error = (j_lls-j_real)/j_real*100


%b_real
    %b_lls = GS*(Tetha(1)/Tetha(2) - 1);
    b_lls = (GS*GT/(1+GT))*(Tetha(1)/Tetha(2) - 1);
    b_lls_error = (b_lls - b_real)/b_real*100


elseif c2 == 0
     j_lls = 1/Tetha(2);
     j_lls_error = (j_lls-j_real)/j_real*100


%b_real
    b_lls = Tetha(1)/Tetha(2);
```

```
    b_lls_error = (b_lls - b_real)/b_real*100


end

%*********************************************************************
```

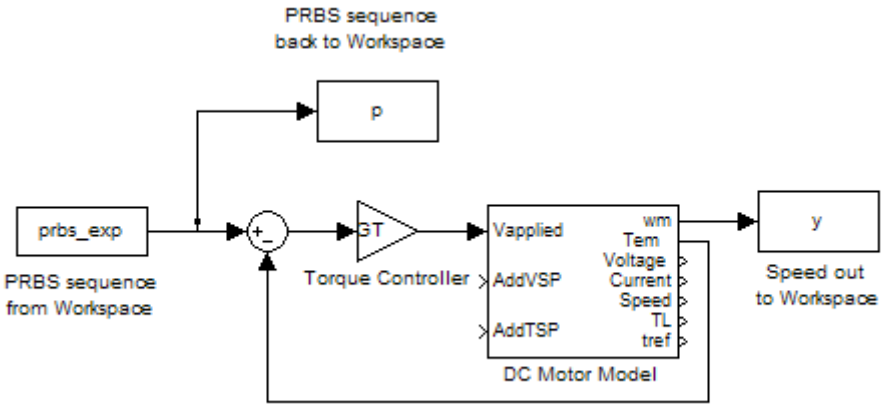
## A.6 DC_motor_no_load.mdl



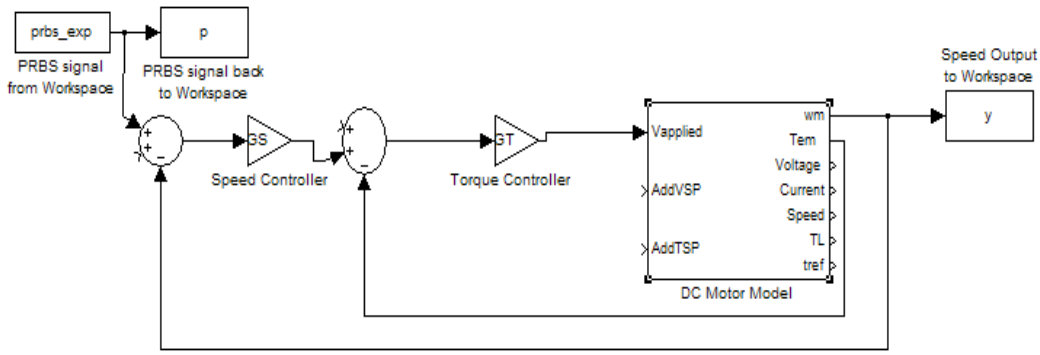

## A.7 DC_motor_with_load.mdl


5
Speed
wl
1/s
Integrator2
1/(Js+B)
1/J_load
b_load1
b_load
6
TL
TL
Csh
Shaft Flexility Constant
Bsh
Shaft Flexility Constant1
Integrator1
wm-wl
1
wm
wm
wl
Integrator
b_real1
Gain
2
Tem
7
tref
1/(Js+B)
J_real1
Tfric
3
AddTSP
K1
K
4
Current
K2
3
Voltage
1/(Ls+R)
den(s)
2
AddVSP
1
Vapplied

## A.8 PRBS_in_Torque.mdl




## A.9 PRBS_in_Speed.mdl



# APPENDIX B
# PLC PROGRAM LISTINGS

The following pages include the program listings for the main program and subroutines used in the PLC.

# APPENDIX C

# INDUCTION MOTOR PARAMETERS

Table C-1 Induction Motor Parameters

| **Parameter** | **Value** |
|---|---|
| Rated Power | 3kW |
| Rated Speed | 1420rev/min |
| Rated Current | 11.1A |
| Power Factor | 0.82 |

$$\text{Rated Speed} = \frac{1420 rev}{\min} \times \frac{1 \min}{60 s} \times \frac{2\pi rad}{rev}$$
$$= 148.7 rad.s^{-1}$$

$$\text{Rated Torque} = \frac{\text{Rated Power}}{\text{Rated Speed}}$$
$$= \frac{3 \times 10^3 W}{148.7 rad.s^{-1}}$$
$$= 20.17 Nm$$

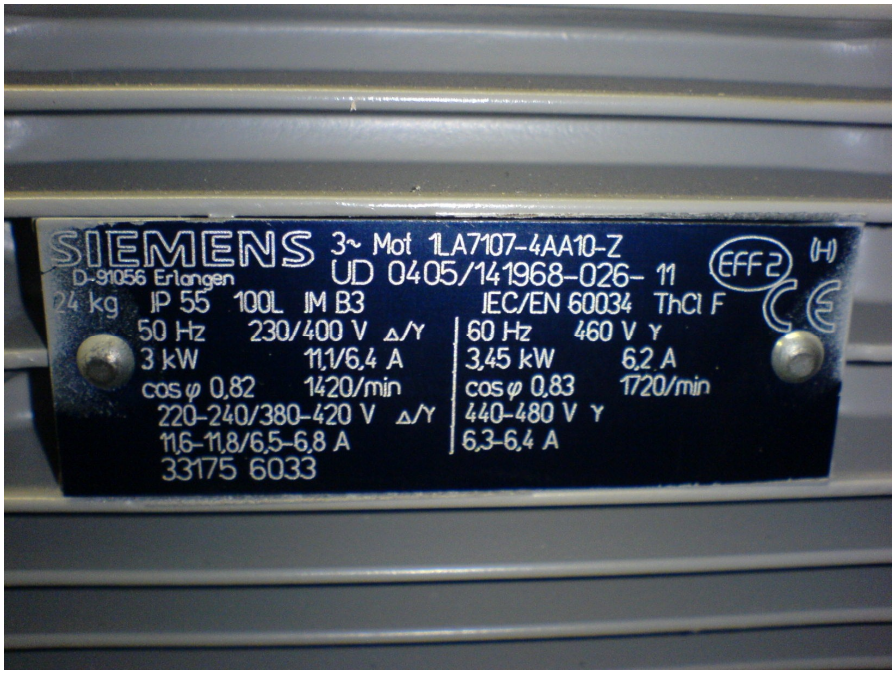



Figure C-1 Induction motor faceplate