

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen



The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

ABSTRACT

VARIABLE GAIN CONTROL OF BRUSHLESS DC MOTOR CONTROL WITH LOW RESOLUTION SENSORS

by
Chung-Hsiang Wang

In this work, feasibility of using low cost, low resolution sensor for high performance brushless dc (BLDC) motor speed control is investigated. Conventional control, using a tachometer or high resolution encoder, suffers from drawbacks such as high cost, large physical volume, and high sensor processing bandwidth. On the other hand, sensorless BLDC motor, appealing in its hardware simplicity, does not provide sufficient fast performance. Using a standard low resolution sensor, such as a hall sensor or commutation encoder, a compromise between cost and performance can be obtained. However, the use of a low resolution sensor does pose a challenge to the control design: the sensor signal is discrete and speed dependent. Together with the nonlinear drive voltage/speed characteristic of the motor, control of the BLDC motor requires a more advanced algorithm than fixed gain control. This thesis presents a speed dependent control scheme to produce optimal performance. The characteristics of the control scheme is first assessed by numerical simulation, based on the mathematical model of the BLDC motor. This is followed by experimental verification of the BLDC motor. From the available data, it is concluded that speed dependent control provides significant advantages over fixed gain control when low resolution sensor is used.

VARIABLE GAIN CONTROL OF BRUSHLESS DC MOTOR
CONTROL WITH LOW RESOLUTION SENSORS

by
Chung-Hsiang Wang

A Master Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Department of Electrical and Computer Engineering

August 1999

APPROVAL PAGE

VARIABLE GAIN CONTROL OF BRUSHLESS DC MOTOR
CONTROL WITH LOW RESOLUTION SENSORS

Chung-Hsiang Wang

8/12/99

Dr. Timothy M. Chang, Thesis Advisor Date
Associate Professor of Electrical and Computer Engineering, NJIT

8/12/99

Dr. Edwin Cohen, Committee Member Date
Professor of Electrical and Computer Engineering, NJIT

8/12/99

Dr. Andrew Meyer, Committee Member Date
Professor of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Chung-Hsiang Wang
Degree: Master of Science in Electrical Engineering
Date: August 1999

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering,
New Jersey Institute of Technology, Newark, NJ, 1999
- Bachelor of Science in Electrical Engineering,
Tamkang University, Tamshui, Taiwan, 1993

Major: Electrical Engineering

To my family, friends and teachers

ACKNOWLEDGMENT

I would like to express my deepest gratefulness and respect to my advisor Dr. Timothy N. Chang for his support, his guidance and his assurance. I also like to thank Dr. Edwin Cohen for his generosity and patience that helps me through this process. Special thanks to Dr. Andrew Meyer for participating in my committee. And I like send my special appreciation to Xuemei Sun for her countless support. Also thanks to Biao Cheng and other fellow classmates. Finally, I would like to thank the NIST ATP Grant number 70NANB5H1092 for Precision Optoelectronics Assembly, for the financial support for this thesis.

TABLE OF CONTENTS

Chapter	Page
1 INTRODUCTION	1
1.1 Brushless DC Motor	1
1.2 Power Control in BLDC Motor	3
1.3 Standard BLDC Control Hardware	5
1.4 Literature Survey	5
1.5 Objective	7
1.6 Organization of Thesis	8
2 HARDWARE DESCRIPTIONS	9
2.1 BLDC Motor	9
2.2 Encoder	10
2.3 The Digital Hardware	11
2.3.1 Programming the DSP Board	11
2.3.2 Initialization	11
2.3.3 Analog Input	13
2.3.4 Analog Output	14
2.3.5 Sampling Rate Determination	14
2.4 Phase Converter	15
2.5 Motor Amplifier	16
3 SOFTWARE IMPLEMENTATION	20
3.1 Motor Drive	20
3.2 Speed Determination	22
3.3 Control Algorithms	23
3.4 Summary	25
4 INTRODUCTION TO SIMULINK AND MODEL DEVELOPMENT	26

Chapter	Page
4.1 Introduction to SIMULINK	26
4.2 The Mathematical Model Development	27
4.3 Simulation Model	28
4.3.1 Motor	28
4.3.2 Encoder	29
4.3.3 Drive Table and Speed Determination	30
4.3.4 PI Control	31
4.4 Parameter Identification	31
5 TEST RESULT AND DISCUSSION	35
5.1 Open Loop Linear Drive	35
5.2 Fixed Gain Control	35
5.3 Variable Gain Control	38
5.4 Comparisons of the Fixed Gain Control and the Variable Gain Control	41
6 CONCLUSIONS AND FUTURE RECOMMENDATIONS	52
6.1 Future Recommendation	52
APPENDIX A MOTOR PARAMETER SPECIFICATIONS AND ENCODER TERMINATION FUNCTIONS	54
APPENDIX B SAMPLE OF SOFTWARE	56

LIST OF FIGURES

Figure	Page
1.1 Cross-section View of the BLDC Motor Structure[2]	2
1.2 Waveforms of (A)Back EMF (B)Phase Current (C,D,E,F)Torque in Brushed or BLDC motor[2]	3
1.3 Torque-Speed Characteristic Curve	4
1.4 Switch Circuits of the PWM Inverter [2]	4
2.1 Experimental System of the BLDC Motor	9
2.2 The Picture of Model TS4073 Motor and RCH20 Encoder	10
2.3 Configuration of the RCH20 Output Signals[18]	12
2.4 Block Diagram of the Dalanco DSP Board	13
2.5 Configuration of the Phase Converter	17
2.6 Configuration of the Motor Amplifier	18
2.7 Frequency Response of the Motor Amplifier Circuit	19
3.1 The Waveform of the Linear Motor Drive	21
3.2 Test Result of the Linear Motor Drive Under No Load Condition	21
3.3 Actual Reading from C31	23
3.4 PI Control Loop for the Motor Speed Control	24
3.5 Flowchart of the Control Program	25
4.1 Mathematical Model of the BLDC Motor	28
4.2 Simulation Model of the BLDC Motor System	29
4.3 Simulation Model of the BLDC Motor	30
4.4 Simulation Model of the Encoder	31
4.5 Simulated Result of the Encoder	32
4.6 Simulation Model of the BLDC Motor PI Control System	33
4.7 Motor Speed Response (A)J=0.0000677 (B)J=0.00008124 (C)J=0.00005416 ($N - m - s^2$)	33

Figure	Page
4.8 Parameter Identification:(A)Experimental Result (B)Simulated Result .	34
5.1 The Test Results of the Open Loop Linear Drive	36
5.2 The Test Results of the Feedforward Gain K_f and the Polynomial Curve Fit Result	38
5.3 Block Diagram of Fixed Gain Control System	39
5.4 The Simulated Speed Response of the Fixed Gain Control ($K_p = 0.15$; $K_i=0.01$)	40
5.5 The Simulated Speed Response of the Fixed Gain Control ($K_p = 0.39$; $K_i=0.01$)	41
5.6 The Experimental Speed Response of the Fixed Gain Control ($K_p=0.15$; $K_i=0.01$)	42
5.7 The Experimental Speed Response of the Fixed Gain Control ($K_p = 0.39$; $K_i=0.01$)	44
5.8 The Motor Performance Under Different PI Gains at 400 RPM	45
5.9 The Motor Performance Under Different PI Gains at 600 RPM	46
5.10 The Optimal Gains at Different Speeds and the 4-degree Polynomial Curve Fit Result for (A) K_p and (B) K_i	47
5.11 Simulated Results using the Optimal Gain Control	48
5.12 Experimental Results Using the Optimal Gain Control	49
5.13 The Fixed Gain Control VS. the Variable Gain Control at 200 RPM . .	50
5.14 The Fixed Gain Control VS. the Variable Gain Control at 1000 RPM . .	50
5.15 The Fixed Gain Control VS. the Variable Gain Control at 2000 RPM . .	51
5.16 The Motor Performance Comparison Using the Fixed Gain and the Variable Gain Control	51
6.1 General Configuration of the BLDC Motor Drive Using Low Resolution Sensors	53

CHAPTER 1

INTRODUCTION

1.1 Brushless DC Motor

Conventional dc motors are highly efficient and their characteristics make them suitable for use as servomotors. However, a primary disadvantage is that conventional dc motors need commutator and brushes which can potentially cause maintenance problems. When the functions of commutator and brushes are implemented by solid-state electronic drives, maintenance-free motors are realized. These motors are known as brushless dc (BLDC) motors [1,2].

The BLDC motor is a synchronous motor with permanent magnet in the rotor and operates in self-controlled mode- using a rotor position sensor and an inverter to control current in the stator windings. Like inverted brushed dc motor, the BLDC motor's magnet is in the rotor and the armature is in the stator as shown in Figure 1.1. In order to get a unidirectional torque, the current in the conductors must reverse polarity when a magnet pole passes by. This procedure is called commutation. Figure 1.2 shows the waveforms of back EMF, phase current, torque in an elementary dc commutator motor or a BLDC motor. In the conventional brushed dc motor, the commutation is performed by the commutator and brushes. For the BLDC motor, the commutation is done by position sensor and solid-state switches in the inverter. Since the process of commutation is similar in both machines, their performance equations and speed/torque characteristics are almost identical (Figure 1.3).

BLDC motor has several advantages compared with conventional brushed motors: small rotor size and high power density, lower inertia, faster dynamic response, high speed and torque capability, almost no maintenance cost, high torque/inertia ratio, and better heat dissipation. These advantages enable BLDC motor to be widely used in applications such as blower motors, computer disk drive

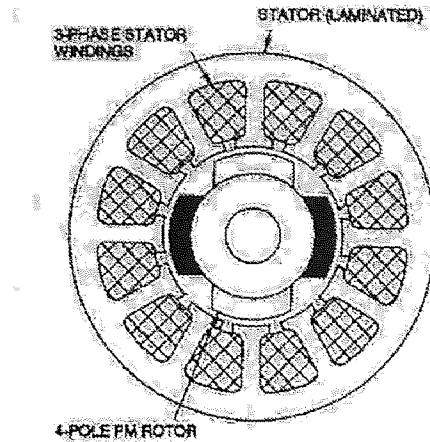


Figure 1.1 Cross-section View of the BLDC Motor Structure[2]

spindle motors, copiers and laser printers, direct-drive robotic joints, and in some extreme environments [11].

There are several different configurations of BLDC motors which use rotating permanent magnets and stationary phase coils. Generally speaking, they are interior-rotor motors, exterior-rotor motors, pancake or disc-type motors, and slotless motors. These designs provide all around choices for the users in wide range applications. For example, if an application requires rapid acceleration and deceleration of the load (as in servo systems) the torque/inertia ratio should be as high as possible. This indicates the use of an interior-rotor motor with high-energy magnets. On the other hand, if an application requires constant speed at medium to high speed it may use an exterior-rotor configuration with the rotating part on the outside of the wound stator. This type is sometimes used to drive fans and blowers. One important application for the exterior-rotor motor is the computer disk drives spindle motor. This application requires a very high and constant speed, and high inertia of the exterior rotor is a good choice to achieve this [2].

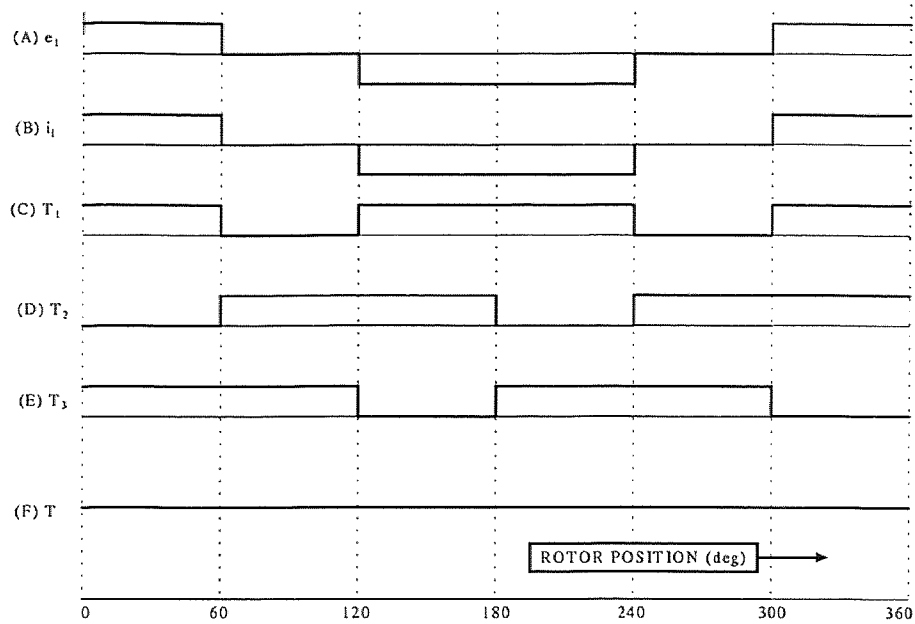


Figure 1.2 Waveforms of (A)Back EMF (B)Phase Current (C,D,E,F)Torque in Brushed or BLDC motor[2]

For other applications such as record players, VCR players, CD players, and floppy disc drives for computer which have a relatively low speed requirements and small space considerations, axial-gap or pancake motors have been designed for many these applications. Slotless motors have reduced core losses and are suitable for application at extremely high speeds, up to at least 100,000 RPM. Since the interior-rotor motor is used in this project, it will be the only type of motor discussed in this chapter.

1.2 Power Control in BLDC Motor

The BLDC motor is very similar to ac synchronous machine that the phase currents change polarity every half-cycle as the magnets (which also alternate in polarity) rotate pass them. If the waveform of the phase currents is sinusoidal, the magnetic-

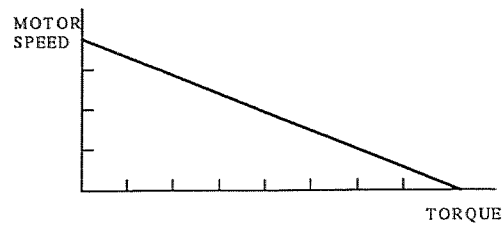


Figure 1.3 Torque-Speed Characteristic Curve

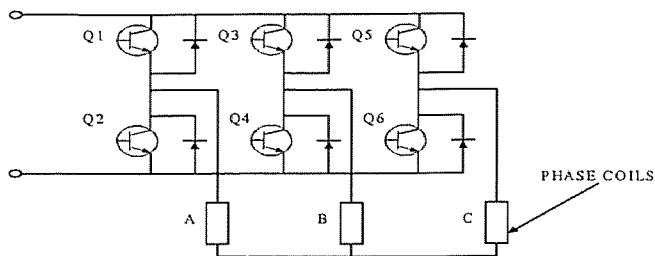


Figure 1.4 Switch Circuits of the PWM Inverter [2]

field distribution rotates continuously at synchronous speed. Sinewave phase currents can be generated by drive electronics which may be linear or pulse width modulated (PWM) [2]. For PWM drives, the supply voltage is switched on and off at a high frequency under the control of a current regulator that forces the transistors to switch the correct voltage polarity to the winding to make the current follow a sinusoidal reference waveform. A typical PWM of the six-step voltage-source inverter is shown in Figure 1.4.

Advantages of the PWM drive include low cost, high power density, and high reliability. For this project, however, the linear drive is used instead to minimize electromagnetic interference caused by the PWM switching.

1.3 Standard BLDC Control Hardware

The electronic circuit of the PWM inverter control system usually includes three basic blocks or sections [2]. The first is the power supply which consists of a power rectifier to convert the ac line voltage into a dc bus voltage. The second section is the inverter which includes the power switches (usually two per phase) and their circuit sensors. The third section is the control, which decodes shaft angle input data, controls dc voltage chopping (PWM), operates control loops such as speed and position.

The sequence of the controller functions are:

1. Switch the power transistors on and off to regulate the current to predetermined value called “current reference”.
2. Synchronize the conduction periods of power transistors so that the direction of current in each motor phase is coordinate with the rotor position.
3. Determine the reference current (I_r) as a function of speed error signal e (actual speed - desired speed). Because of the linearity of the BLDC motor, I_r will be proportional to e ($I_r \propto e$). The constant of proportionality is the *gain*. In servo systems normally two gains are used (PI control). The function of integral gain is to eliminate steady-state error. Sometimes the derivative gain (PID control) is also added.

1.4 Literature Survey

Bak [3] presented a method by using a solid-state circuit to replace the external position sensor in the BLDC motor commutation. For some particular applications where there is no sufficient room for the sensing system, or where the operating temperature is too hot or too cold for electronic devices, an external sensor cannot be used. In stead, three electronic comparators monitor the back EMF developed during

those interval when the motor coils are not energized. Because back EMF is proportional to rotor velocity and relative to rotor position, the waveform's continuous feedback can be used to energize the appropriate coils for maximum commutation efficiency. The comparators, six N-channel MOSFETs, a biasing integrator, and sequencing logic form the commutation circuit.

Iizuka [4] explained the method which permits the determination of the rotor position by the back EMF induced in the stator windings and the starting technique which use the motor as a synchronous motor at standstill in sensorless BLDC motor control. Kim [5] proposed the binary observer to alleviate a chattering phenomenon and enhance the estimation performance in a speed sensorless control of the BLDC motor. Matsui [6], Takeshita [7] presented sensorless drives for the BLDC motor with a sinusoidal flux distribution. The controller calculates the motor current by using the estimated position and back EMF of the motor model.

Blank [8] showed the shape of torque-speed characteristic can be varied by controlling the commutation angle. The commutation angle is defined as the angle between the applied voltage and the back EMF. If this angle is maintained at zero, the machine behaves in a manner similar to that of the conventional dc motor, its torque-speed characteristic being essentially linear especially at low speed. The results of torque-speed curves due to different commutation angles are presented. The degree to which the commutation angle is utilized should depend on the requirements of the application.

Huh [9] described a torque control strategy with low resolution sensor. The torque ripple due to motor's structure generates vibration, noise and losses on the servo system. The other mechanism of generating torque lies in the current control algorithm. The current controller demands fast response and high accuracy. But neither hysteresis nor PWM method can avoid high ripple factor on account of the freewheeling current and other factors. Therefore, an observer is added to the torque

control algorithm. The observer estimates position and speed at every sampling instant, which provides the torque control algorithm calculating desired one step ahead voltage to have rippleless torque.

From the literature survey three type of control schemes are used. The first and also the most commonly used type is the BLDC motor with position and speed sensors. Continuous data input is the advantage. The disadvantage includes more space required to install the sensor, and of course, more expensive. The second type is sensorless BLDC motor, using back EMF or observer to commutate and control. Its advantage is the disadvantage of the first one. The problem of sensorless motor is there is no back EMF while motor is at standstill. Therefore, more complex control scheme and circuits are required. The last type rarely used is the BLDC motor with low-resolution sensor (like Hall-effect sensor). This is the type of motor and control scheme discussed in this thesis.

1.5 Objective

The objective of this thesis is to investigate the feasibility of using low resolution sensors for high performance speed control of the BLDC motors. The linear motor drive has better performance than a PWM motor drive because due to its inherent high resolution and low noise characteristics. Furthermore, the DSP completes one execution in 0.1ms, the software-commutated time is much less than transistors turning on/off . This leads to less noise, less torque ripple, and faster response.

But the linear motor drive encountered a problem. Unlike other controllers which have current sensors or/and tachometers providing data feedback, the low-resolution sensor can degrade the performance of the linear motor drive when feedback control is applied. Fortunately, the simulation results indicate that the PI gain and motor speed have a well defined relationship. Trial-and-error is then used to find out the pattern of these control parameters. A look-up table is built

to overcome the slow data update caused by the low-resolution sensor. An optimal feedback control method is introduced and the results are presented.

1.6 Organization of Thesis

Chapter 2 of this thesis describes the hardware used in this project. Chapter 3 discusses the details of the software structures. A brief introduction of simulation software and mathematical and simulation model development are presented in Chapter 4. The simulated and experimental results are provided in Chapter 5. Chapter 6 derives conclusion from the work presented along with the future direction.

CHAPTER 2

HARDWARE DESCRIPTIONS

The experimental system for investigating the characteristics of variable gain control is shown in Figure 2.1. It consists of a TS4073 brushless dc(BLDC) motor by Tamagawa Seiki CO. LTD, a Renco Encoders RCH20 encoder, a Dalanco Spry Model 310 Digital Signal Processing Board, a motor drive amplifier, and interface electronics. The properties of these hardware components are described in this chapter.

2.1 BLDC Motor

The motor used in this project is a permanent-magnet, four poles, three-phase, Y-connect BLDC motor. TS4073 motor is capable of delivering 300W under a supply voltage 104V (line-to-line). The rated speed is 3000RPM, maximum to 4000 RPM. Rated torque of the TS4073 motor is 0.9516 N-m, peak torque up to 2.8547 N-m. The parameters including the rating of the motor is listed in Appendix A1.

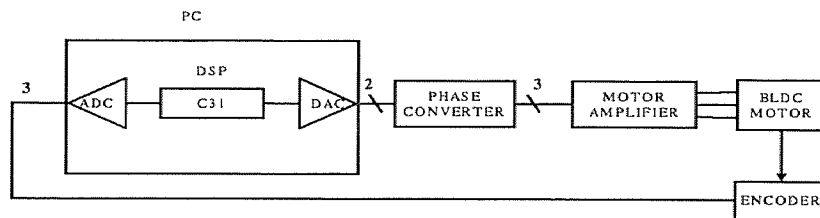


Figure 2.1 Experimental System of the BLDC Motor

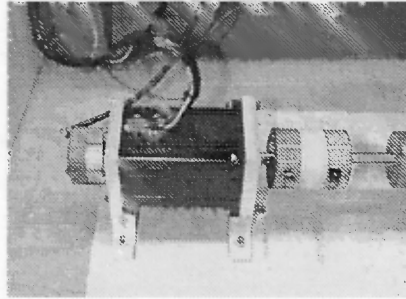


Figure 2.2 The Picture of Model TS4073 Motor and RCH20 Encoder

2.2 Encoder

The encoder used in this project is a RCH20 commutation encoder manufactured by Renco Encoders, INC. The optical encoder consists of a set of pairs of phototransistors and collimated light sources, used in conjunction with a metal encoder disc. The pattern of slits on the disc defines the frequency and waveform of the pulse trains which are produced by phototransistors.

The RCH20 commutation encoder is installed onto the motor shaft to provide position and velocity data. The encoder floats on the shaft allowing axial and radial movement of the shaft while providing high resolution capabilities. The resolution of RCH20 is 2 cycles/360 degree (2 pole pairs machine) in commutation channel, 2000 cycles/360 degree in data channels. The data channels are reserved for future effort and are not used in this project. The termination functions of RCH20 are shown in Appendix A2. And the output signal configuration is shown in Figure 2.3. [18]

In order to simulate the application of low-resolution sensor, only three commutation channels (S1, S2, S3) are used in this project. The commutation channels S1, S2, and S3 each produces 2 pulses in one revolution of the encoder. Therefore, the frequency of the each of three commutation channels is twice as the motor speed

frequency. The following equation shows the relationship between the motor speed and the encoder commutation signals :

$$n = 30 * f \quad (2.1)$$

where

n = motor shaft speed (RPM).

f = frequency of the encoder commutation channel signals (H_z).

2.3 The Digital Hardware

The DSP system used for this project is the Dalanco Spry Data Acquisition and Signal Processing Board Model-310B [17]. The DSP board has a Texas Instruments' TMS320C31 DSP chip running at 50MHz, two 12 bit DAC, a 14bit ADC with a four channel multiplexer, and 128k words of memory. The DAC is capable of outputting at a maximum rate of 140kHz. The ADC has a maximum conversion rate of 300kHz. The voltage ranges for the ADC and the DAC are $\pm 5V$. Figure 2.4 is a block diagram of the Dalanco Spry DSP Board. The DSP can be programmed in 'C' as well as Assembly code. It is however necessary to create an application programming interface so that all the coding could be done in 'C', with the user completely insulated from the architecture of the board.

2.3.1 Programming the DSP Board

The following sections describe the three function calls which form the entire user interface to the signal processing board. To make use of these calls it is necessary to include the file '*d310biov.h*' as a header at the start of the program.

2.3.2 Initialization

The 310B Board needs initialization at startup. The ADC connects to the serial port on the DSP. So the serial port and timer must be set up, and also the latch on

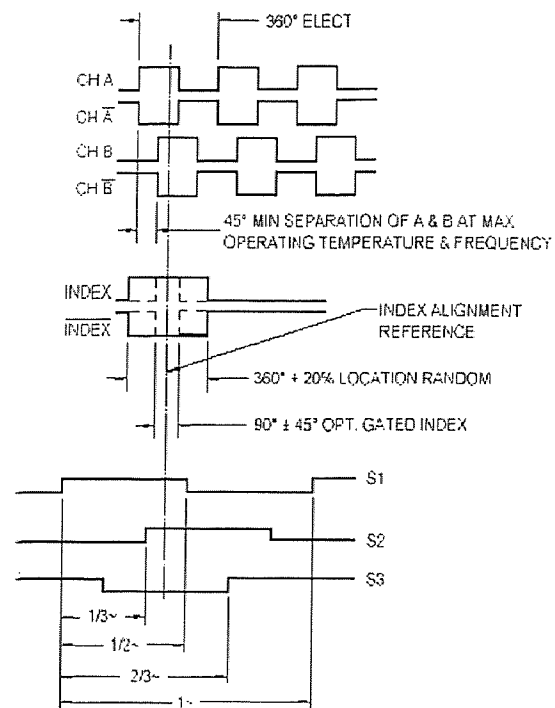


Figure 23 Configuration of the RCH20 Output Signals[18]

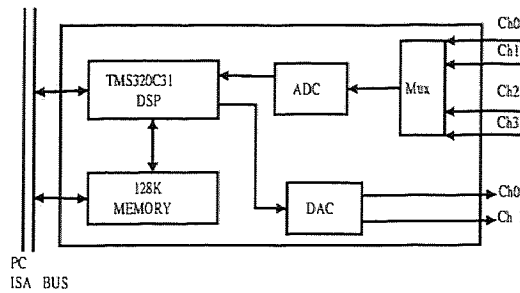


Figure 2.4 Block Diagram of the Dalanco DSP Board

the Dalanco Spry DSP Board must be set up. For this the *InitDSP()* function is implemented. The function call prototype is 'void *InitDSP*(void)'.

2.3.3 Analog Input

Pulses from the low resolution encoder are read by the ADC in the experimental system to facilitate rapid system diagnostic. In general implementation, the encoder pulses can be connected directly to the external interrupts of the C31. To input digitized signals from the ADC requires the function *ReadAdc()*. The prototype for this function is 'int *ReadAdc*(int channel)'.

This function reads the value from the ADC output buffer for the voltage applied on the specified channel. The function returns values ranging from -2047 to 2047 for voltages ranging from $-5V$ to $5V$. The integer data type 'int' on the C31 DSP is 32 bits, while the conversion result is 12 bits. The necessary sign extension is performed internally, transparent to the user. To read from more than one channel multiple calls to *ReadAdc* are necessary. Since the ADC on the card has four channels, legal values for channel are from 0 to 3. If the voltage at the ADC input is to be calculated then the ADC output is simply multiplied by the scaling factor $5/2047$. The function begins by writing the channel (and the default gain of unity) to the latch. Once the latch is set the function waits for the conversion to be

triggered. Whenever the count is complete the pin goes high, stays high for a clock period and then goes low. This event is used for triggering the ADC in hardware. This also is used to synchronize the software to a time source. The timer runs off the DSP clock, in the timer mode, and its accuracy is determined by the DSP clock. This is the source of timing in all the control programs written with this library.

2.3.4 Analog Output

The output to the analog channels is written via the DAC. The function call for this is *WriteDAC()*. The C prototype for this is 'int *WriteDAC*(int value, int channel)'.

This outputs the value to the DAC channel specified. Since the DAC has two channels, legal values for channel are 0 and 1. The value written out on these two channels ranges from -2047 to 2047. If the value is greater than 2047 it is clamped off to 2047, and if less than -2047, is also restricted to -2047. This clamping is done by the function *WriteDAC* automatically. This is to avoid problems associated with 'roll over'. If the value is greater than 2047, then it cannot be properly expressed in 12 bits and leads to wrong interpretation of the value. Thus if a value of 4096 is output to the DAC with an intended output value of 10V, it gets clamped to 5V only, since the DAC output is restricted to 5V. Similar clamping occurs on the negative side.

2.3.5 Sampling Rate Determination

To use the above function calls include the file '*d310bio.h*' at the start of the program, and it is also necessary to define the sampling rate for the system. This is done by defining the constant *TIMPER0*. The value of *TIMPER0* is calculated from following equation:

$$TIMPER0 = \frac{\textit{System Clock}}{\textit{samplingfrequency} * \textit{numcalls} * 8} \quad (2.2)$$

where

SystemClock=50MHz.

numcalls=the total number of function calls to the function *ReadAdc* in one execution of the control loop.

Take the program used in this project as an example, the sampling frequency is 10kHz, and the control loop has three *ReadAdc*, then the value of *TIMPER0* equals to 208.33.

Since *TIMPER0* is to be used in the header file *d310biov.h* to set the sampling period for the board it is essential to initialize *TIMPER0* prior to the '#include *d310biov.h*' statement.

2.4 Phase Converter

Since the three phase drive is phase locked, it can be generated by using two D/A channels along with a phase converter. The signals generated by the D/A converters are V_1 and V_2 which are fed to the phase converter to produce the required three phase drive signals V_a , V_b , and V_c according to the following equations:

$$V_1 = V \sin \theta \quad (2.3)$$

$$V_2 = V \cos \theta \quad (2.4)$$

$$V_a = V_1 \quad (2.5)$$

$$\begin{aligned} V_b &= V \sin\left(\theta - \frac{2\pi}{3}\right) \\ &= V\left[\sin \theta \cos \frac{2\pi}{3} - \cos \theta \sin \frac{2\pi}{3}\right] \\ &= V[(-0.5) \sin \theta + (-0.866) \cos \theta] \\ &= -0.5V_1 - 0.866V_2 \end{aligned} \quad (2.6)$$

$$\begin{aligned} V_c &= V \sin\left(\theta - \frac{4\pi}{3}\right) \\ &= V\left[\sin \theta \cos \frac{4\pi}{3} - \cos \theta \sin \frac{4\pi}{3}\right] \\ &= V[(-0.5) \sin \theta + (0.866) \cos \theta] \end{aligned}$$

$$= -0.5V_1 + 0.866V_2 \quad (2.7)$$

From equation (2.3) and (2.4), V_b and V_c can be reconstructed by one sine wave and one cosine wave. The phase converter can turn these two sinusoidal signals into desired three phase signals.

In the implementation circuit, one adder, one subtractor, and three inverters are used to build the phase converter circuit. Two potentiometers are used in two of those three inverters to create the precise gains (-0.5 and -0.866). The third one has unit gain only to reverse the phase of V_b . The complete circuit configuration is shown in Figure 2.5.

2.5 Motor Amplifier

The Motor Amplifier used in this project is based on the National Semiconductor LM12. The LM12 is a power op-amp capable of driving ± 25 at $\pm 10A$ while operating from $\pm 30V$ supplies. The monolithic IC can deliver 80W of sine wave power into 4W load with 0.01% distortion. Power bandwidth is 60k Hz[18].

As shown in Figure 2.6, the motor amplifier consists of three individual circuits. Each of them has one input connected to the phase converter and one output connected to the motor winding. Since each circuit has the same configuration and can operate separately, only the top side of the circuit will be discussed here. In this sub-circuit, one LM12 is used and two diodes and two capacitors are placed to create virtual ground. Upper side of this-circuit are two parallel components- one capacitor and one resistor. The resistor can be changed to modify the gain of output signal depending on the requirement of application. In this case, the resistor is $15K\Omega$ and the gain is 14.75.

The complete circuit configuration is shown in Figure 2.6. And Figure 2.7 shows the frequency response of this amplifier circuit.

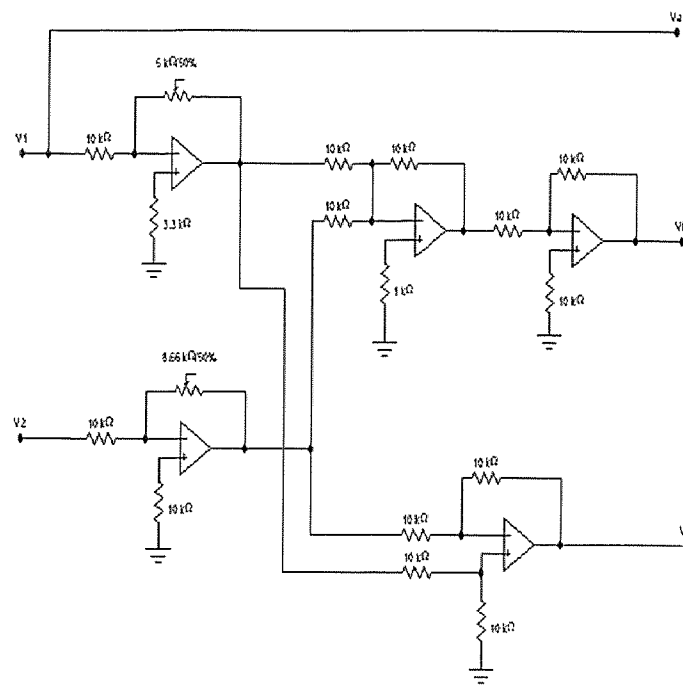


Figure 2.5 Configuration of the Phase Converter

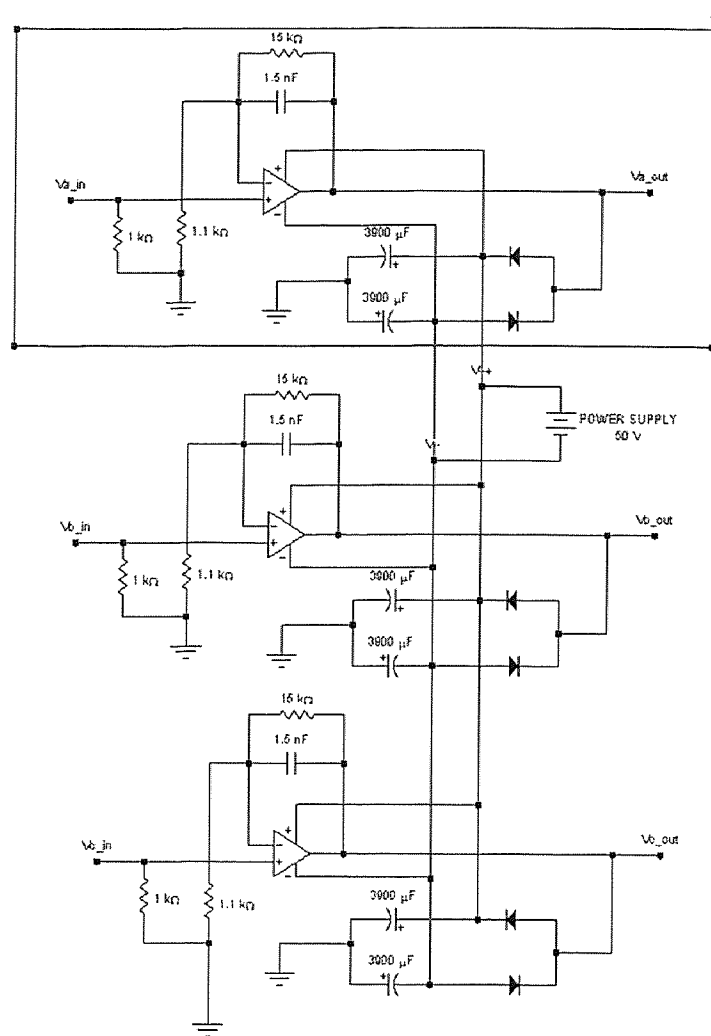


Figure 2.6 Configuration of the Motor Amplifier

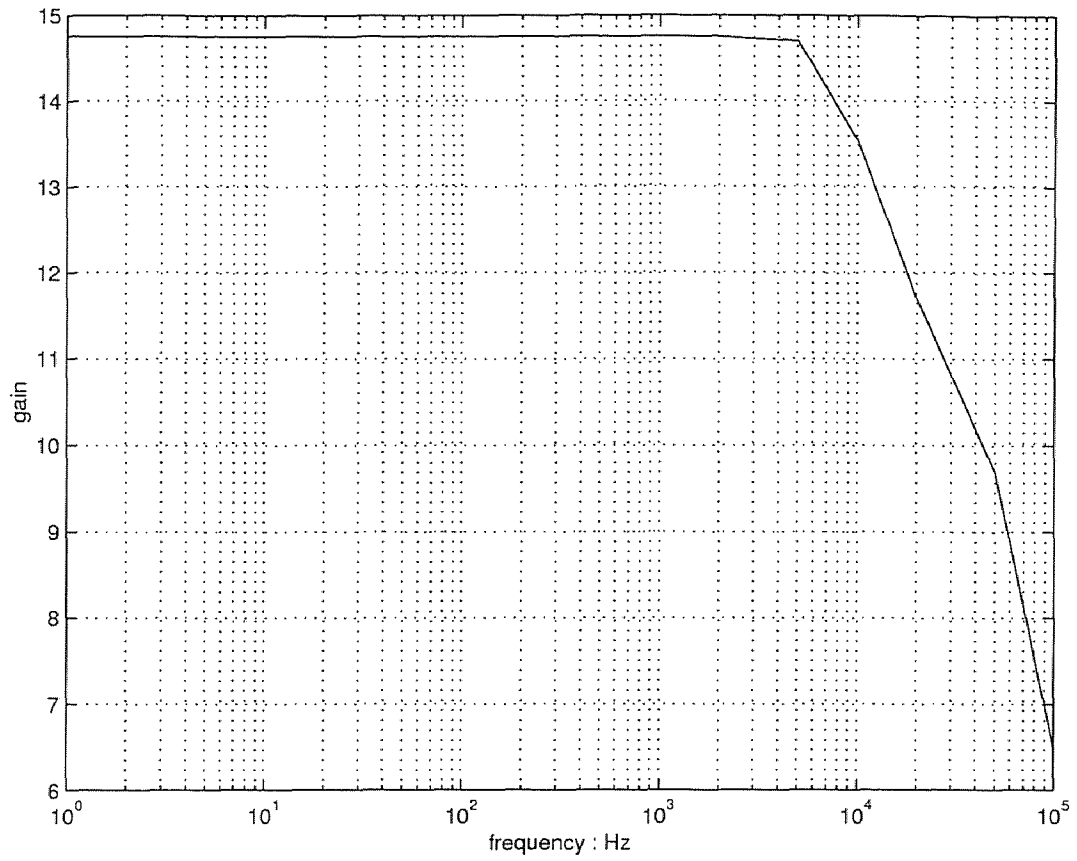


Figure 2.7 Frequency Response of the Motor Amplifier Circuit

CHAPTER 3

SOFTWARE IMPLEMENTATION

3.1 Motor Drive

In Pulse Width Modulation (PWM) inverter drive, the commutation is done by turning the switches on/off according to the rotor position signal from sensor. This section will introduce another way of commutation which is done all by software. As shown in previous chapter, rotor positions are provided by encoder. Physically, it is not necessary to align the encoder with the rotor shaft perfectly. The alignment actually can be done by software. By adding rotor position test, the phase between rotor and encoder can be found.

In the experimental system, linear motor drive is used instead of PWM for the following reasons:

- Linear drive is “quiet” with low switching noise.
- Linear drive simplifies system analysis and diagnostic.

For general implementation, either PWM or linear drive can be used with the proposed variable gain control. Due to the low resolution of the sensor, it is impossible to supply the sinusoidal voltages to the windings. Instead, a modified sinusoidal voltage has been generated to drive the motor. This is because the motor is a 4-pole machine; its speed is half the frequency of the winding voltage. And the encoder signal is two pulses per revolution, which makes the encoder signals have the exact same frequency as the winding voltages. When considering all three channels of the encoder S1, S2, and S3, six different output signals from the encoder are made during one cycle. If '1' indicates high of the encoder output, '0' indicates low, there are 6 combinations coming out of S1, S2, and S3: (1,0,1), (1,0,0), (1,1,0), (0,1,0), (0,1,1), (0,0,1), in one clockwise revolution. These 3 signals are read by ADC0, ADC1, and ADC2 of C31 to update the winding voltages. Therefore, the

sinusoidal waveform is modified as 6-step waveform shown in Figure 3.1. DAC0 and DAC1 write the corresponding voltages to the converter circuit. This code segment performs the linear drive generation of the BLDC motor. The motor drive table is listed in table 3.1. For example, if C31 reads (0,0,1) from encoder, then C31 writes $(A \sin \frac{2\pi}{3}, A \cos \frac{2\pi}{3})$ to the converter. The larger the A is, the faster the motor would be. The characteristic of A is shown in Figure 3.2. Figure 3.3 shows the actual reading from C31. The first 3 graphs represent the signals of 3 A/D Converters of C31 read from encoder S1, S2, and S3, respectively. Last two graphs are the waveforms written by DAC0 and DAC1 to the converter. i.e. Y-axis shows the actual scaling of C31. The value multiplying factor $\frac{5}{2047}$ is the actual voltage outside the C31.

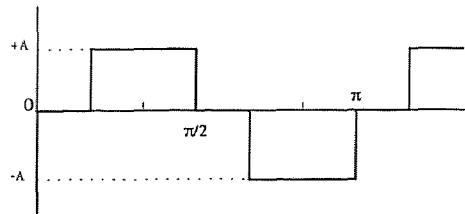


Figure 3.1 The Waveform of the Linear Motor Drive

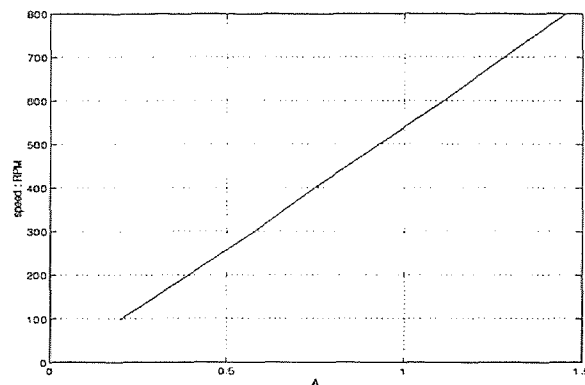


Figure 3.2 Test Result of the Linear Motor Drive Under No Load Condition

Table 3.1 Motor Drive Table

ENCODER			C31	
S1	S2	S3	DAC0	DAC1
1	0	0	$A \sin 2\pi$	$A \cos 2\pi$
1	0	1	$A \sin \frac{\pi}{3}$	$A \cos \frac{\pi}{3}$
0	0	1	$A \sin \frac{2\pi}{3}$	$A \cos \frac{2\pi}{3}$
0	1	1	$A \sin \pi$	$A \cos \pi$
0	1	0	$A \sin \frac{4\pi}{3}$	$A \cos \frac{4\pi}{3}$
1	1	0	$A \sin \frac{5\pi}{3}$	$A \cos \frac{5\pi}{3}$

3.2 Speed Determination

As shown in previous section, the motor speed is half of the encoder signal frequency. Here the channel S1 of the encoder has been chosen as speed determination sampler. When S1 goes through one cycle, (one complete high and one complete low voltage output from channel S1, or equivalently, 180° mechanical rotation of the motor shaft.) the total time during this process multiplies by two and takes reciprocal as the motor rotating frequency. This value can be displayed as rad/sec by multiplying a factor 2π . In DPS programming, time is expressed in sampling points. Now since the sampling rate is fixed, motor speed can be determined by counting the number of S1 pulses in a fixed time period. Therefore, counters are added in each of six situations of the drive table program. When C31 reads (1,0,1), it not only writes desired voltages to the converter circuit, but also adds one to counter1. After 30 mechanical degrees, the encoder signals turn to (1,0,0), counter1 stops counting and counter2 starts to count and writes second voltage out. After another 30 mechanical degrees, counter2 stops counting and counter3 starts to count. Till after all six counters are finished, the total counts of one cycle of channel S1 is obtained by adding them all. This value is transformed to the motor speed, and all those counters are set to zero to record next cycle. The sampling rate of this program is $10k H_z$, so the motor speed (n) is calculated by the following equation:

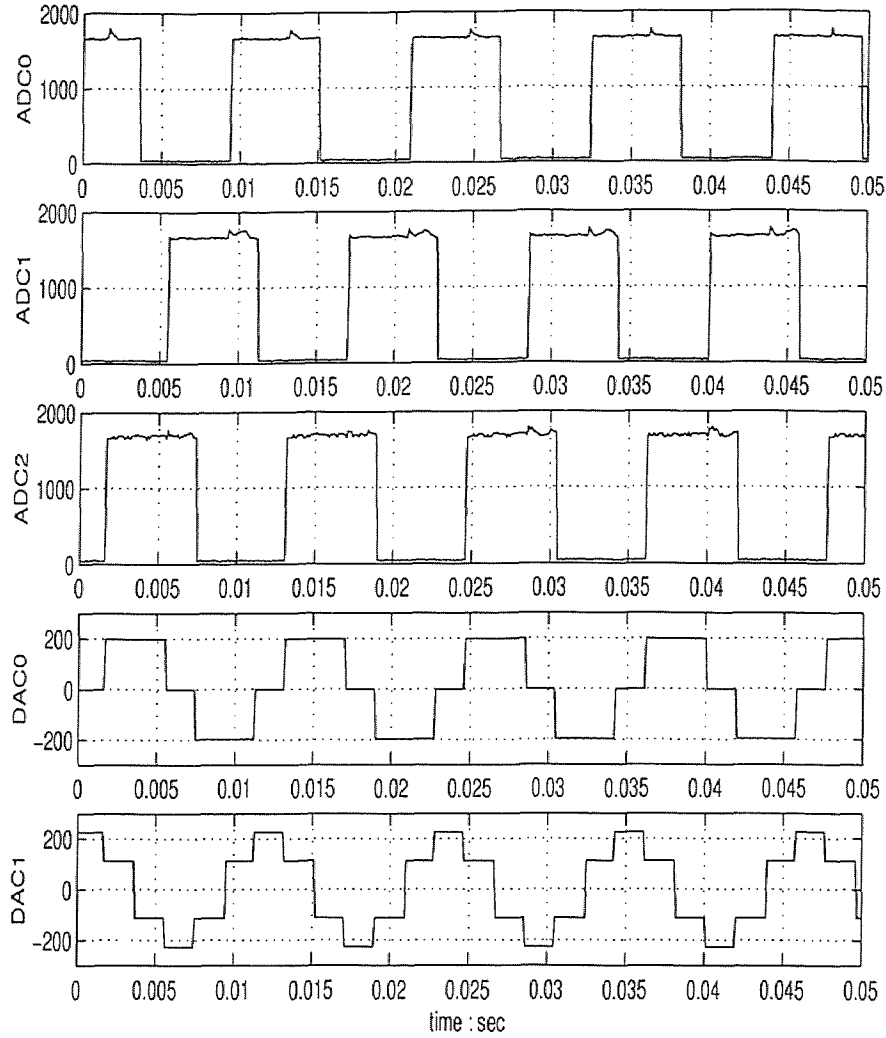


Figure 3.3 Actual Reading from C31

$$speed : n = 30 \times \frac{10000}{totalcount} (RPM) \quad (3.1)$$

3.3 Control Algorithms

The Proportional and Integral (PI) control is selected for speed control in this project, because it is well-known, easy to apply and working well in industrial motor control. Figure 3.4 shows the block diagram of the PI control loop. In Figure 3.4, K is a scale factor which converts the speed (RPM) into voltage. The feedback speed n_m is the

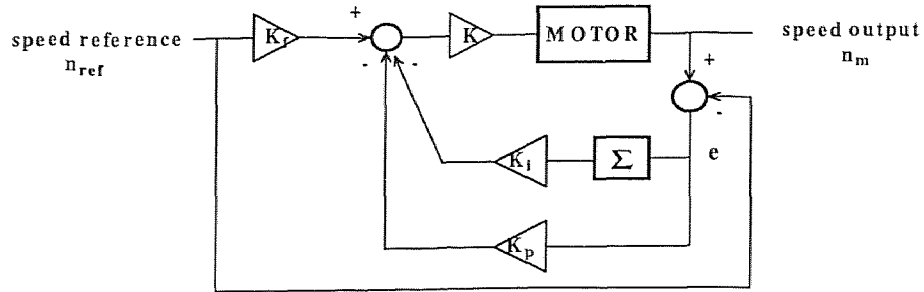


Figure 3.4 PI Control Loop for the Motor Speed Control

speed calculated by Speed Determination inside the program, the speed command n_{ref} is the speed reference; hence, the speed error $e = n_m - n_{ref}$. The input U then becomes:

$$U(k) = K_f n_{ref} - K_p e(k) - K_i \sum_{i=0}^k e(i) \quad (3.2)$$

where

K_f = feedforward gain
 K_p = proportional gain
 K_i = integral gain

Equation(3.2) is a fixed-gain, PI control algorithm. For variable-gain PI control, it is modified as:

$$U(n_{ref}) = K_f(n_{ref})n_{ref} - K_p(n_{ref})e - K_i(n_{ref}) \sum_{i=0}^k e(i) \quad (3.3)$$

where

$K_f(n_{ref})$ = speed dependent feedforward gain
 $K_p(n_{ref})$ = speed dependent proportional gain
 $K_i(n_{ref})$ = speed dependent integral gain

3.4 Summary

The motor speed control program starts with *include* directives and variable declarations, followed by the body of the actual program. The *include* directives is used to include the *d310birov.h*. This library file contains definitions of *InitDSP()*, the *ReadADC()*, and the *WriteDAC()* functions required to run the Dalanco DSP Board in 'C' language. The variables in this program includes all three control gains- K_p , K_i , and K_f , speed reference, and several counters for speed determination uses. The body of the program is an infinite loop which consists of reading the output signals of encoder, computing and writing out the resultant voltage to the converter. Figure 3.5 is the flowchart of this control loop.

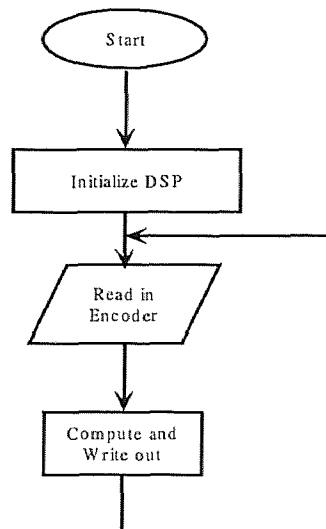


Figure 3.5 Flowchart of the Control Program

CHAPTER 4

INTRODUCTION TO SIMULINK AND MODEL DEVELOPMENT

In this chapter, a mathematical model of the motor system is derived, along with the parameter identification for the motor inertia. This model is then simulated using MATLAB/SIMULINK to assess the open loop and closed loop characteristics of the motor system.

4.1 Introduction to SIMULINK

SIMULINK is a toolbox extension of the MATLAB program. It is designed for simulating dynamic systems. Besides generating the simulated data, an interactive environment such as SIMULINK further offers a number of advantages:

- All dynamic states are readily available.
- Control structure can be easily re-configured.
- Simulation can run in parallel and in super real-time.

The simulation process consists of a number of steps:

1. model building
2. simulation engine selection
3. run condition optimization

A fixed-step solver with step size of 0.0001 second (sampling rate= $10\text{k}H_z$) were chosen to simulate the DSP environment. ODE1(Euler) was chosen as the simulation engine because it was found to be the fastest among all five available engines and does not has significant distortion.

Table 4.1 Nomenclature

V_{an}, V_{bn}, V_{cn}	electric voltages across phase a,b,and c windings, respectively
I_a, I_b, I_c	electric currents of phase a,b,and c windings, respectively
R	resistance of motor winding
L	capacitance of motor winding
T	electromagnetic torque
J	rotor moment of inertia
k	number of magnet pole pairs in the rotor
K_e	motor constant
ω	motor shaft angular velocity
θ	motor shaft angular position

4.2 The Mathematical Model Development

The motor used in this project is a 4-poles, Y-connected, BLDC motor. The nomenclature of the parameters of the motor is listed in Table 4.1. The windings of the motor are arranged in three phases : phase-a, phase-b, phase-c. These three phase windings are distributed sinusoidally in the stator with a separation of $\frac{2\pi}{3}$ angle from one another. The rotor has 2 pairs of permanent magnets which provide sinusoidally distributed magnetic flux in the air gap between the stator and the rotor. The voltage equations for the motor windings are:

$$V_{an} = RI_a + L \frac{dI_a}{dt} + K_e \omega \sin k\theta \quad (4.1)$$

$$V_{bn} = RI_b + L \frac{dI_b}{dt} + K_e \omega \sin(k\theta - \frac{2\pi}{3}) \quad (4.2)$$

$$V_{cn} = RI_c + L \frac{dI_c}{dt} + K_e \omega \sin(k\theta - \frac{4\pi}{3}) \quad (4.3)$$

The torque generated by the motor can be written as

$$T = K_e [I_a \sin k\theta + I_b \sin(k\theta - \frac{2\pi}{3}) + I_c \sin(k\theta - \frac{4\pi}{3})] \quad (4.4)$$

The equation describing the dynamics of the mechanical motion is

$$T = J \frac{d\omega}{dt} + T_L \quad (4.5)$$

Equation (4.1), (4.2), (4.3), (4.4), (4.5) combined together yield the mathematical model of the BLDC motor. Applying Laplace transformation to the above equations forms the motor block diagram shown in Figure 4.2[11,12].

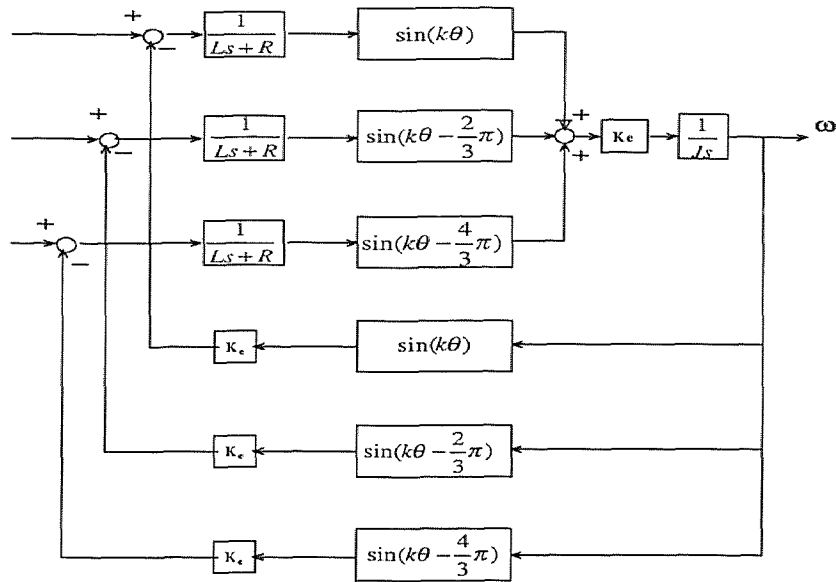


Figure 4.1 Mathematical Model of the BLDC Motor

4.3 Simulation Model

Figure 4.2 is a simulation block diagram of the BLDC motor control system. Each function block will be discussed in detail in the following subsections.

4.3.1 Motor

The simulation model of the motor is developed in terms of the mathematical model given by equations (4.1) to (4.5). The simulation diagram is shown in Figure 4.3

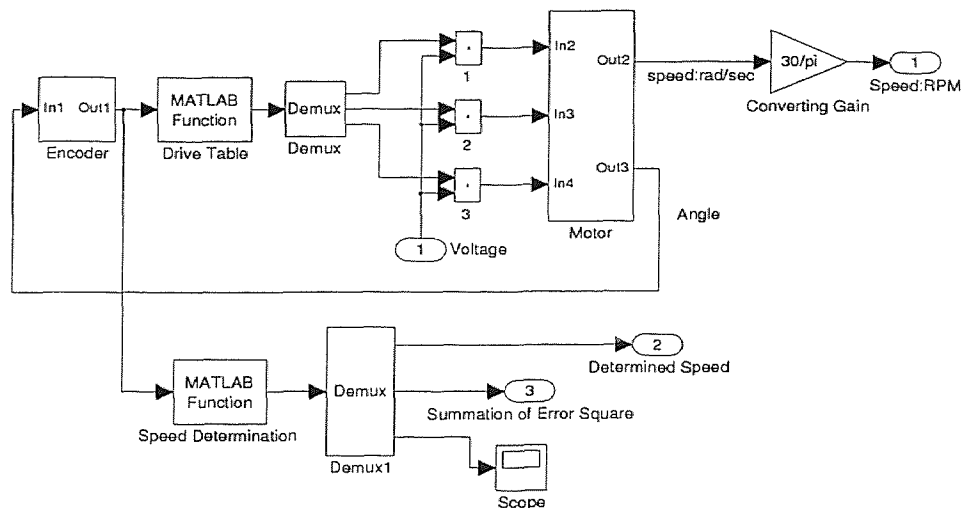


Figure 4.2 Simulation Model of the BLDC Motor System

4.3.2 Encoder

The encoder is a device that turns motor shaft angle into voltage pulse train. It can be simulated by using 3 MATLAB function blocks (Figure 4.4). The functions in each of three blocks are, $[\text{rem}(u,\pi) \leq \pi/2]$, $[\text{rem}(u,\pi) > \pi/3 \ \& \ \text{rem}(u,\pi) \leq 5\pi/6]$, $[\text{rem}(u,\pi) > 2\pi/3 \ + \ \text{rem}(u,\pi) \leq \pi/6]$, respectively. In these function blocks, 'u' is the input signal as motor shaft angle which is obtained by adding an integrator right after the speed output. According to (4.6), this integrated variable represents the motor shaft angle. 'rem(u,pi)' means the input variable 'u' is divided by π . If the remainder falls in $[0,\pi]$, then the MATLAB function block outputs 1, otherwise it outputs 0. This would give us the simulated encoder S1 output. The rest of two

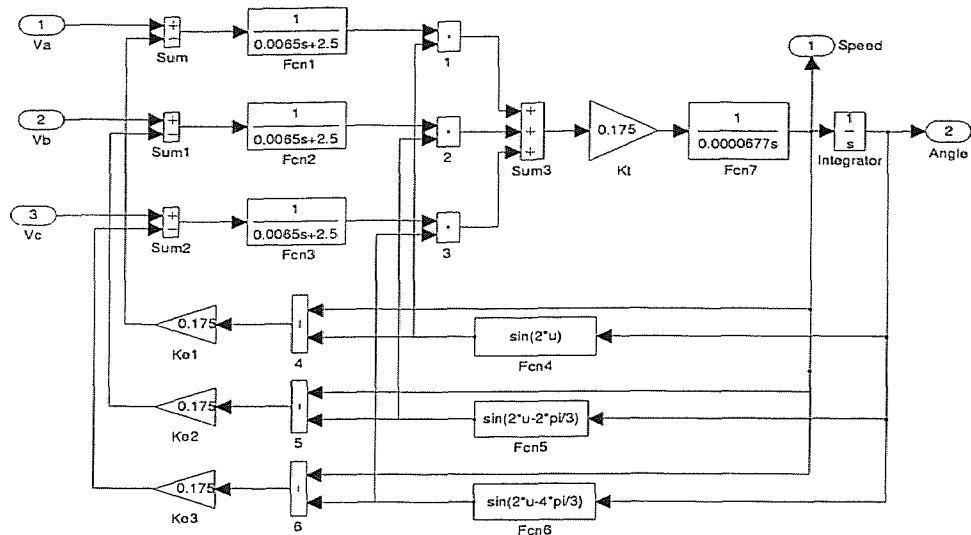


Figure 4.3 Simulation Model of the BLDC Motor

blocks represent the S2 and S3, respectively. The simulated results are shown in Figure 4.5.

$$\omega = \frac{d\theta}{dt} \quad (4.6)$$

where

ω = motor speed (rad/sec)
 θ = motor shaft angle (rad)

4.3.3 Drive Table and Speed Determination

In these two blocks, the MATLAB function blocks have been used to create three-phase 6-steps waveforms to drive the motor and calculate the output speed. The algorithm is the same as discussed in the previous chapter. Since 'C' language is

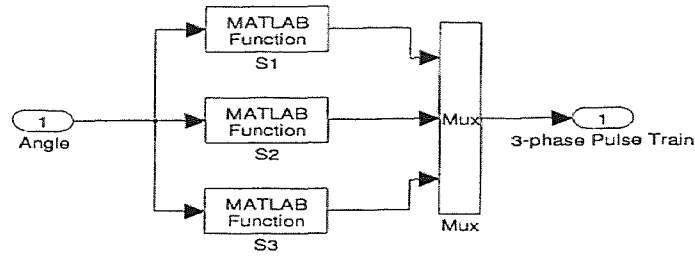


Figure 4.4 Simulation Model of the Encoder

used in both C31 and MATLAB, it's very convenient to implement without many changes. These two M-file are listed in Appendix B.

4.3.4 PI Control

For the convenience, those function blocks in Figure 4.3 are grouped as subsystem 'BLDC Motor' in Figure 4.5. The output signal coming from BLDC Motor function block is the motor speed (n_m). n_m minus reference speed (n_{ref}) becomes speed error (e). e extends into two parallel lines: one multiplies K_p as proportional control input; the other passes through an integrator then times K_i to form integral control input. These two lines are subtracted by n_{ref} to complete the speed input. This speed input multiplies converting(scale) factor (K) would convert to voltage input to the motor drive system.

4.4 Parameter Identification

To complete the simulation model, the parameter identification is applied. According to simulation model, the motor speed response is affected primarily by rotor inertia (J). Figure 4.7 shows the motor speed responses at no load and their corresponding J based on the PI control ($K_p = 0.4$ and $K_i = 10^{-6}$). The rated inertia of model TS4073 is $0.0000677 \text{ N} - \text{m} - \text{s}^2$ which is shown in Figure 4.7 (A). Figure 4.7 (B) and (c) show the results by increasing 20% of J ($J=0.00008124 \text{ N} - \text{m} - \text{s}^2$) and decreasing 20% of J ($J=0.00005416 \text{ N} - \text{m} - \text{s}^2$), respectively.

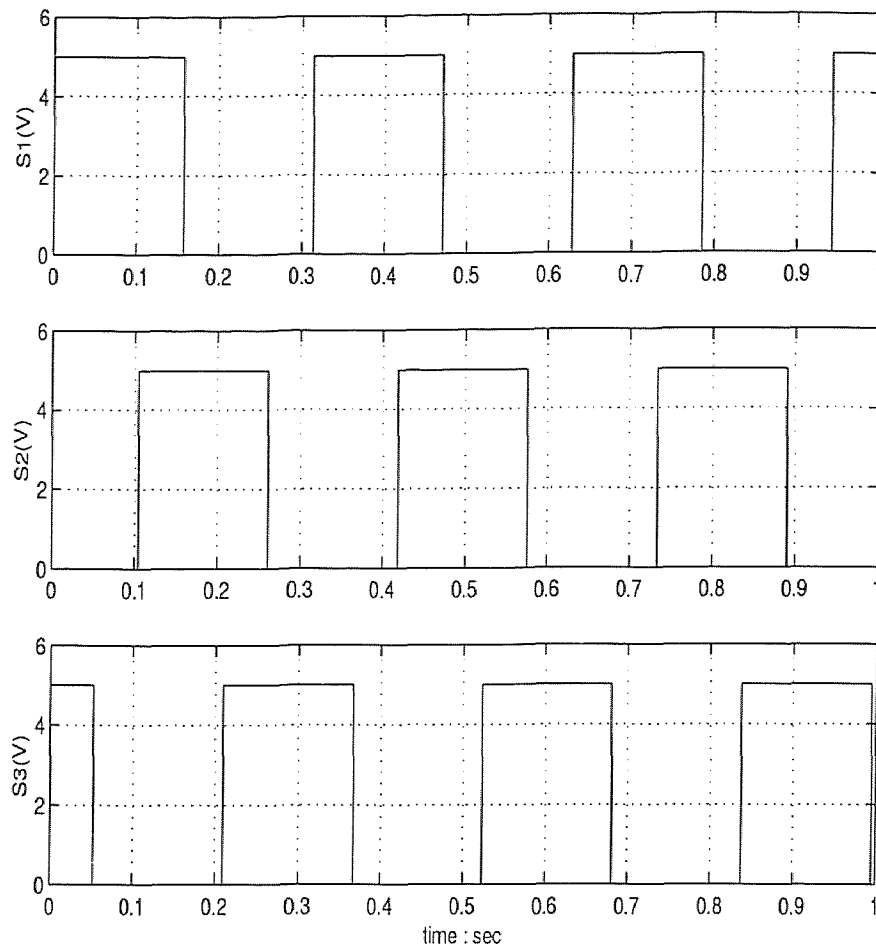


Figure 4.5 Simulated Result of the Encoder

The experimental result of motor speed response is obtained by implementing the same control conditions as in simulation. The value of J in simulation model is adjusted based on the comparison between experimental result and simulated result. At final value ($J = 0.0000775N - m - s^2$) the experimental result and simulated result show very good agreement with each other (Figure 4.8).

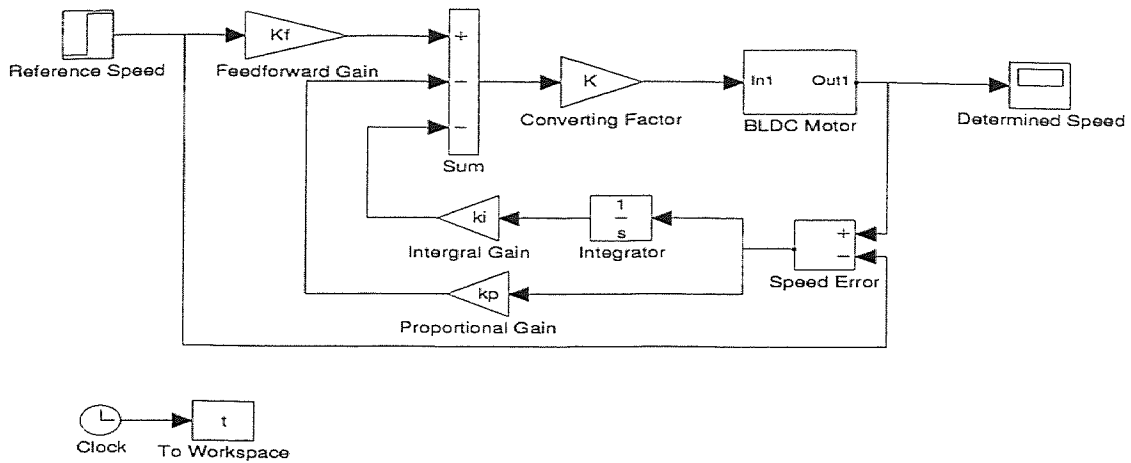


Figure 4.6 Simulation Model of the BLDC Motor PI Control System

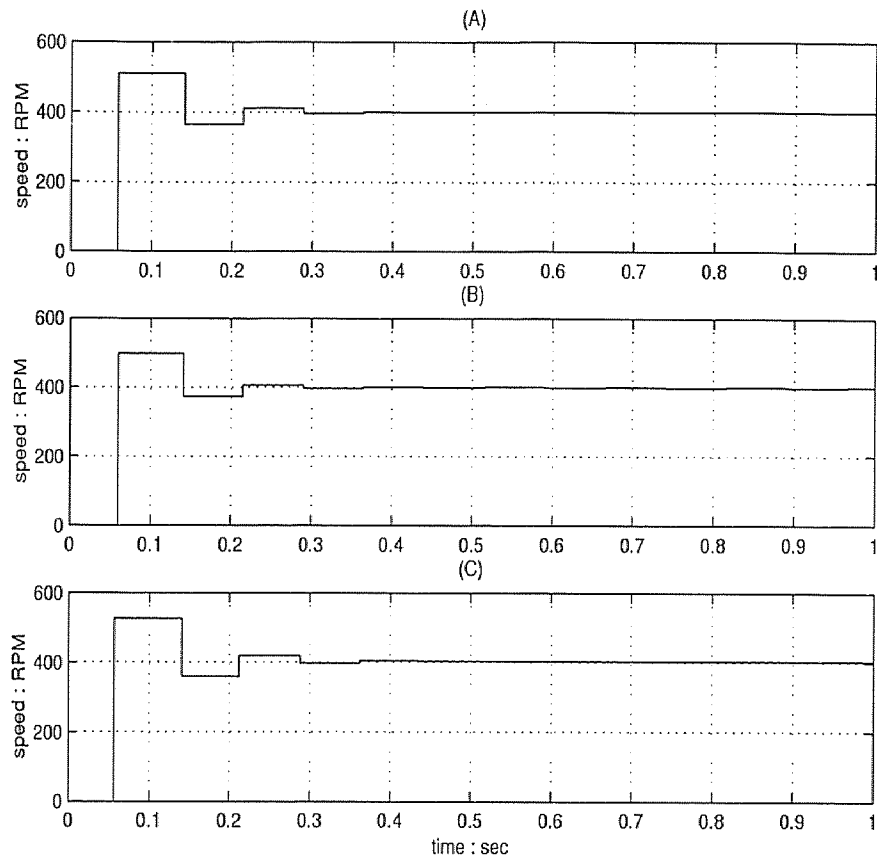


Figure 4.7 Motor Speed Response (A) $J=0.0000677$ (B) $J=0.00008124$ (C) $J=0.00005416$ ($N - m - s^2$)

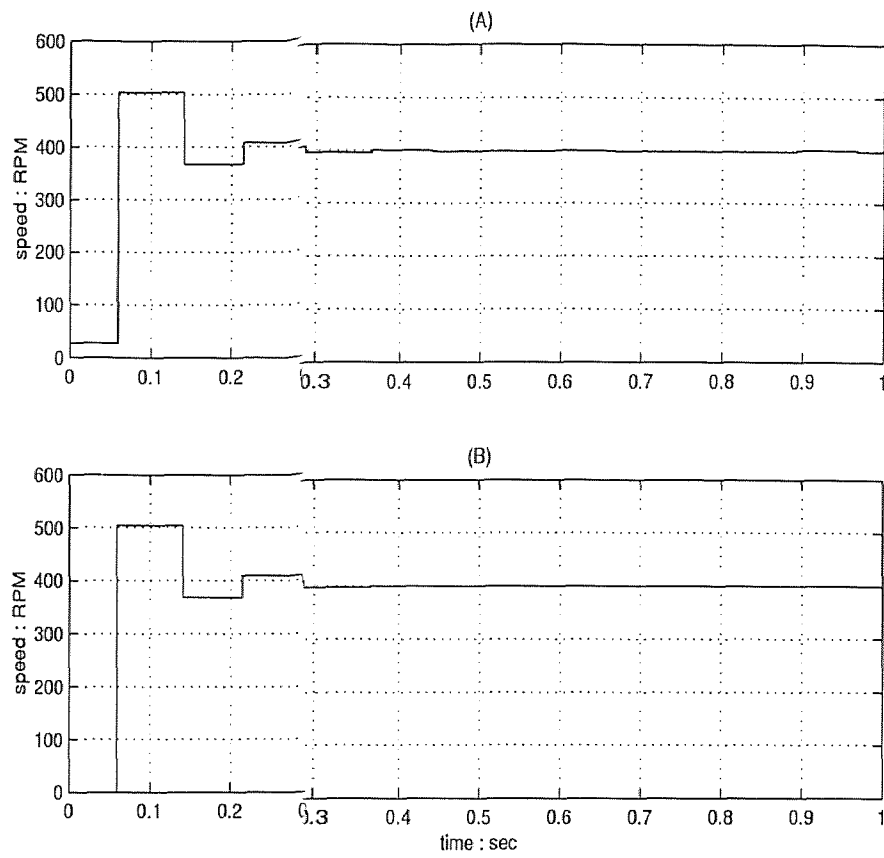


Figure 4.8 Parameter Identification: (A) Experimental Result (B) Simulated Result

CHAPTER 5

TEST RESULT AND DISCUSSION

In this chapter, results of motor speed control using low resolution encoder feedback are presented. Both fixed gain control and variable gain control are discussed and compared. The results were first obtained from simulation, followed by experimental verification.

5.1 Open Loop Linear Drive

In the open loop drive test, as discussed in the Chapter 3, the scale factor K converts speed reference into motor supply voltage. As depicted in Figure 5.1, the motor drive voltage versus speed curve is nonlinear, further suggesting the use of speed dependent control gains.

In order to implement the linear drive system, we add a feedforward gain to it. Therefore, the scale factor K stays constant and the feedforward gain K_f would change corresponding to speed reference. Figure 5.2 shows the relationship between converting factor K and speed reference n_{ref} , and 3-degree polynomial curve fit result. The equation for the K_f using 3-degree polynomial curve fit is given by following equation:

$$\begin{aligned} K_f(n_{ref}) = & 1.71695 \times 10^{-10} \times n_{ref}^3 - 1.54822 \times 10^{-7} \times n_{ref}^2 \\ & + 3.89123 \times 10^{-4} \times n_{ref} + 0.87266 \end{aligned} \quad (5.1)$$

5.2 Fixed Gain Control

In open loop drive system, the motor speed response is slow and exhibits steady-state error. The PI fixed gain control is commonly used in motor speed control to

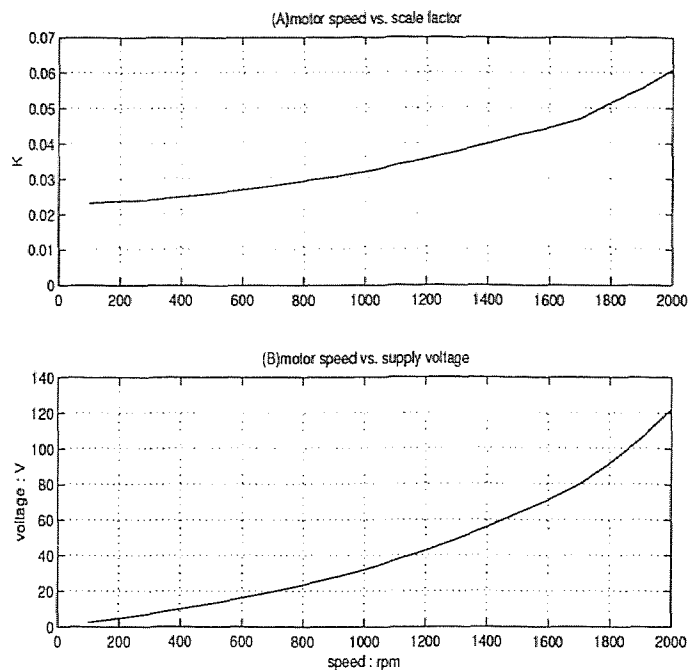


Figure 5.1 The Test Results of the Open Loop Linear Drive

improve the performance of the BLDC motor. The simulation setup of fixed gain control system is shown in Figure 5.3. And the test results are shown in Tables 5.1, 5.2 and Figures 5.4, 5.5. Two criteria: rise time and overshoot are used to judge the performance of the motor. Figures 5.6, 5.7, and Table 5.3 are the experimental results of the fixed gain control using the optimal gains of (A) 200RPM and (B) 600RPM.

Table 5.1 The Simulation Results of the Fixed Gain Control ($K_p=0.15$; $K_i=0.01$)

Test Speed (RPM)	Rise Time (sec)	Overshoot (%)	RMS
200	0.1306	6.6	40.4721
400	0.1417	0.35	64.1588
600	0.095	1.83	90.0909
800	0.0714	2.75	108.9495
1000	0.0578	0.24	161.8642
2000	0.029	1.35	218.9561

Table 5.2 The Simulation Results of the Fixed Gain Control ($K_p = 0.39$; $K_i=0.01$)

Test Speed (RPM)	Rise Time (sec)	Overshoot (%)	RMS
200	0.1098	25.1	53.4664
400	0.0575	10.95	58.3733
600	0.0395	0.83	85.2687
800	0.0666	3.31	102.557
1000	0.0539	4.1	132.2497
2000	0.0278	2.75	208.2486

Table 5.3 Experimental Results of the Fixed Gain Control (A) $K_p=0.15$; $K_i=0.01$ (B) $K_p=0.39$; $K_i=0.01$

Test Speed (RPM)	Type	Rise Time (sec)	Overshoot (%)	RMS
200	A	0.134	22	59.42
	B	0.123	4	66.24
400	A	0.112	1.35	73.3
	B	0.102	6.5	68.6
600	A	0.103	12.4	91.7
	B	0.097	15.1	87.3
800	A	0.0401	6.41	112.3
	B	0.0354	6.25	109.1

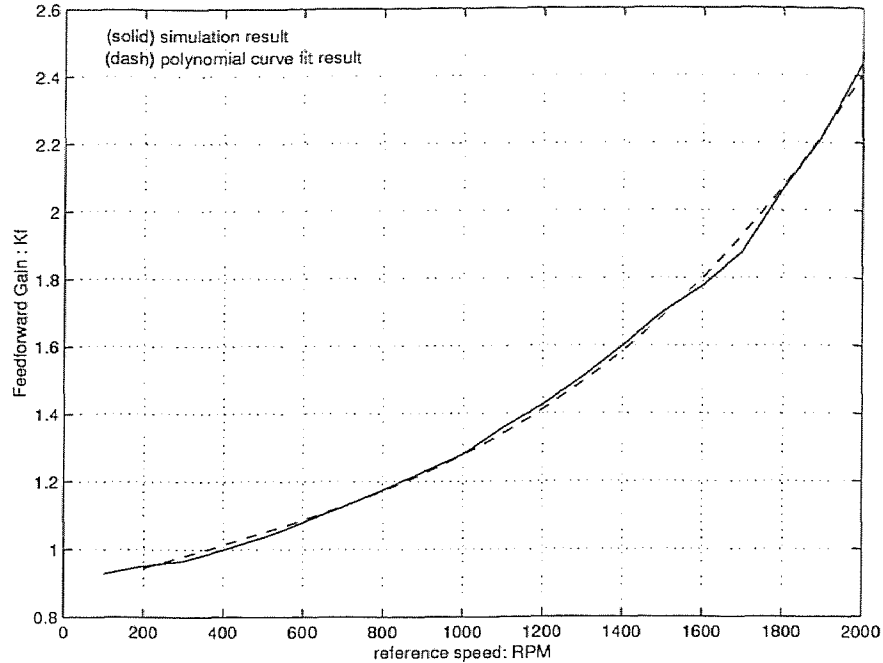


Figure 5.2 The Test Results of the Feedforward Gain K_f and the Polynomial Curve Fit Result

5.3 Variable Gain Control

From Table 5.1 and 5.2, the the performances between two different fixed gain controls show wide range variations in every tested speed. Getting fast response may suffers large overshoot and reducing overshoot may lose response time. It does not exits a specific group of gains for every speed in the linear motor drive. Therefore, the variable gain control is designed to optimized the motor speed control. First step is to find out what kind of conditions would be the best for the motor performance. Here the 'best' is decided by '*LeastErrorSquare*' criterion (Equation 5.2).

$$\text{RootMeanSquare}(RMS)\text{of error} = \sqrt{\frac{1}{N} \sum_{i=0}^N e^2(i)} \quad (5.2)$$

where

$$e = \text{speed error(RPM)}$$

Figure 5.8 and Figure 5.9 show the example at 400 RPM and 600 RPM, respectively. The RMS value of error is changing along with the the proportional gain (K_p) and

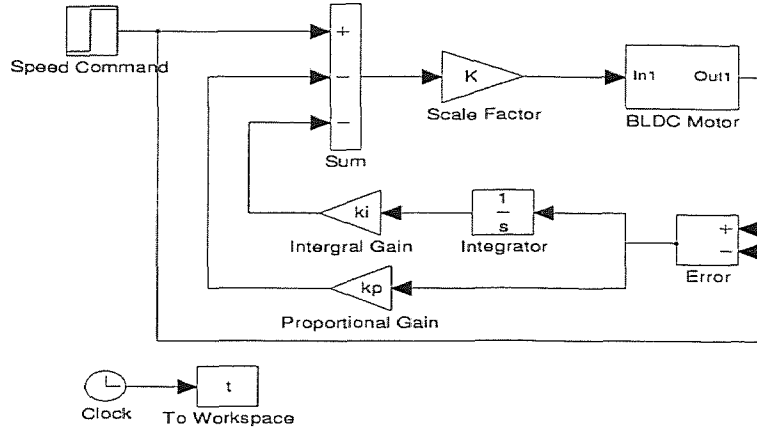


Figure 5.3 Block Diagram of Fixed Gain Control System

the integral gain (K_i). The optimal gains are obtained by trial-and-error. First, set K_i equals to zero and increase the value of K_p gradually. Thus, the corresponding value of K_p (0.34) of the minimum RMS (58.1) can be found. Next, set K_p equals to 0.34 and change the value of K_i in each test. The corresponding value of K_i (0.001) of the minimum RMS (56.5) can also be found. Repeat the first step by setting K_i equals to 0.001 and new pair of values can be found ($K_p = 0.36$ and $RMS=56.3$). Repeat this procedure till the RMS value of the error stabilizes. For $K_p = 0.36$ and $K_i = 0.01$, the RMS reaches its minimum at 56.1. As a result, the $K_p = 0.36$ and $K_i = 0.01$ are the optimal gains for speed 400 RPM. The rest of the optimal gains for each speed can be found in the same manner. The result and their 4-degree polynomial curve fit results are shown in Figure 5.10. The following equations are the 4-degree polynomial curve fit results for K_p and K_i :

$$K_p(n_{ref}) = -8.33957 \times 10^{-13} \times n_{ref}^4 + 3.87314 \times 10^{-9} \times n_{ref}^3$$

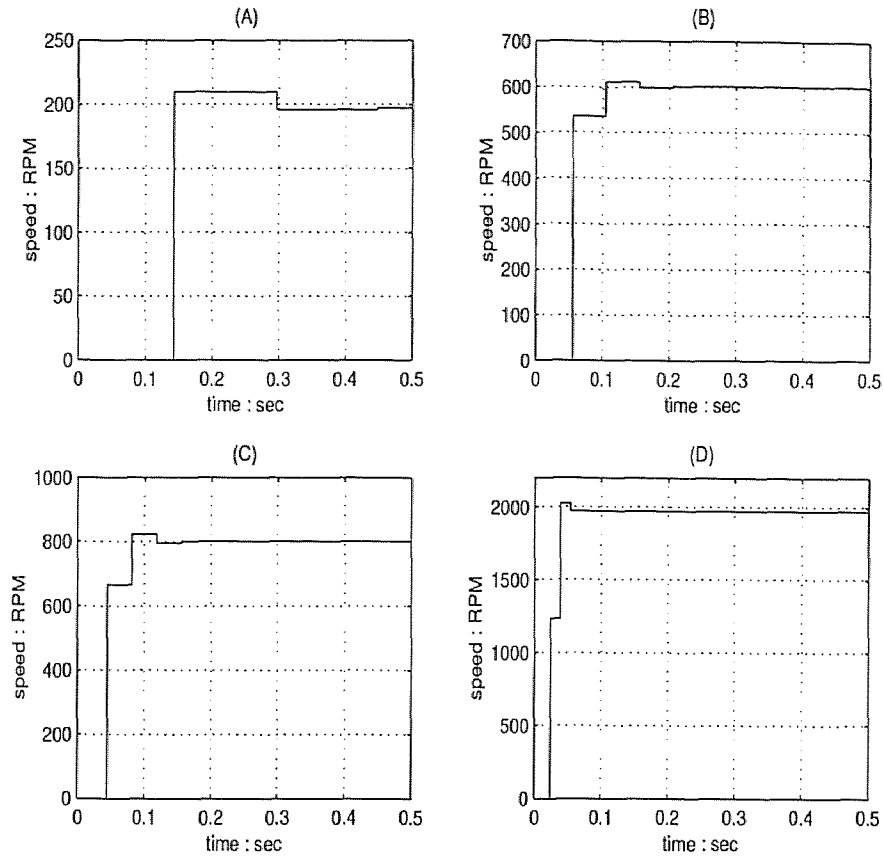


Figure 5.4 The Simulated Speed Response of the Fixed Gain Control ($K_p = 0.15$; $K_i=0.01$)

$$-5.59248 \times 10^{-6} \times n_{ref}^2 + 3.34716 \times 10^{-3} \times n_{ref} - 0.315516 \quad (5.3)$$

$$K_i(n_{ref}) = -6.95027 \times 10^{-15} \times n_{ref}^4 - 3.83148 \times 10^{-11} \times n_{ref}^3 \\ + 1.93072 \times 10^{-7} \times n_{ref}^2 - 1.32687 \times 10^{-4} \times n_{ref} + 0.03061 \quad (5.4)$$

where

$K_p(n_{ref})$ =optimal proportional gain in speed(n_{ref})
 $K_i(n_{ref})$ =optimal integral gain in speed(n_{ref})
 n_{ref} =desired speed reference(RPM)

To verify simulated results, those optimal gains were brought back to the motor control system. Figure 5.11 shows the simulated results of the motor speed response

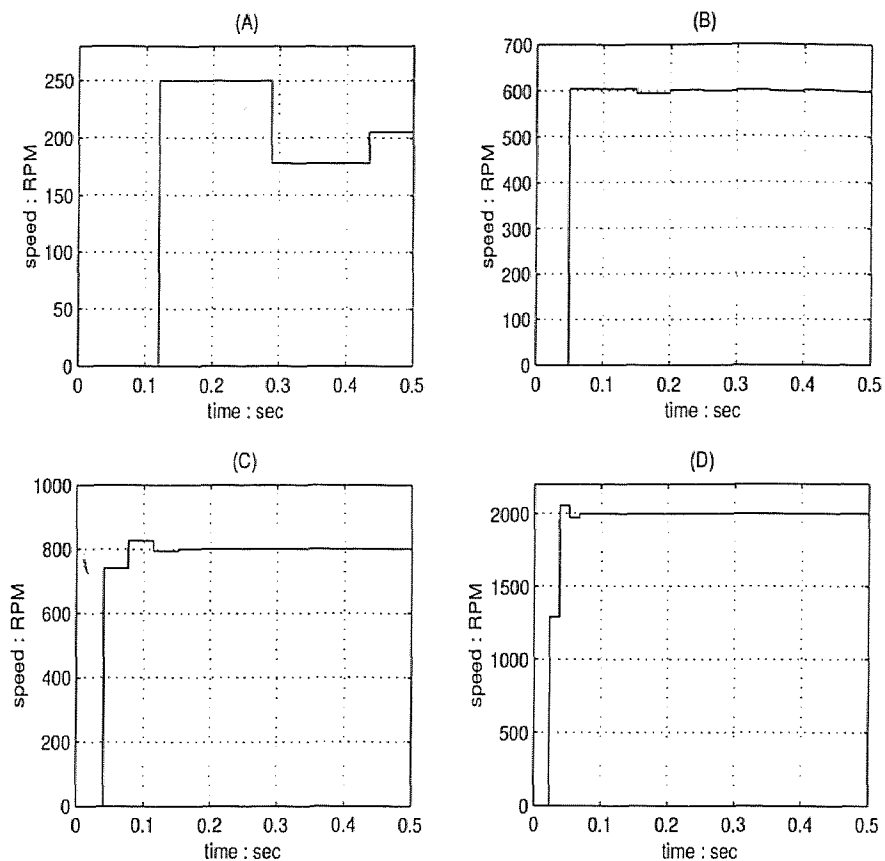


Figure 5.5 The Simulated Speed Response of the Fixed Gain Control ($K_p = 0.39$; $K_i=0.01$)

under optimal gains at 4 different speeds. And Figure 5.12 shows the motor actual speed response under same conditions. There are no major differences between simulated and experimental results. This comparison verifies the simulation and the feasibility of the variable gain control.

5.4 Comparisons of the Fixed Gain Control and the Variable Gain Control

Figure 5.13 ~ Figure 5.15 are the motor speed response comparisons of the fixed gain and the variable gain control at 3 different speeds, Table 5.4 and Table 5.5 show the numerical comparisons of motor performance between the fixed gain and the variable gain control in the simulation and the experiment, respectively. The

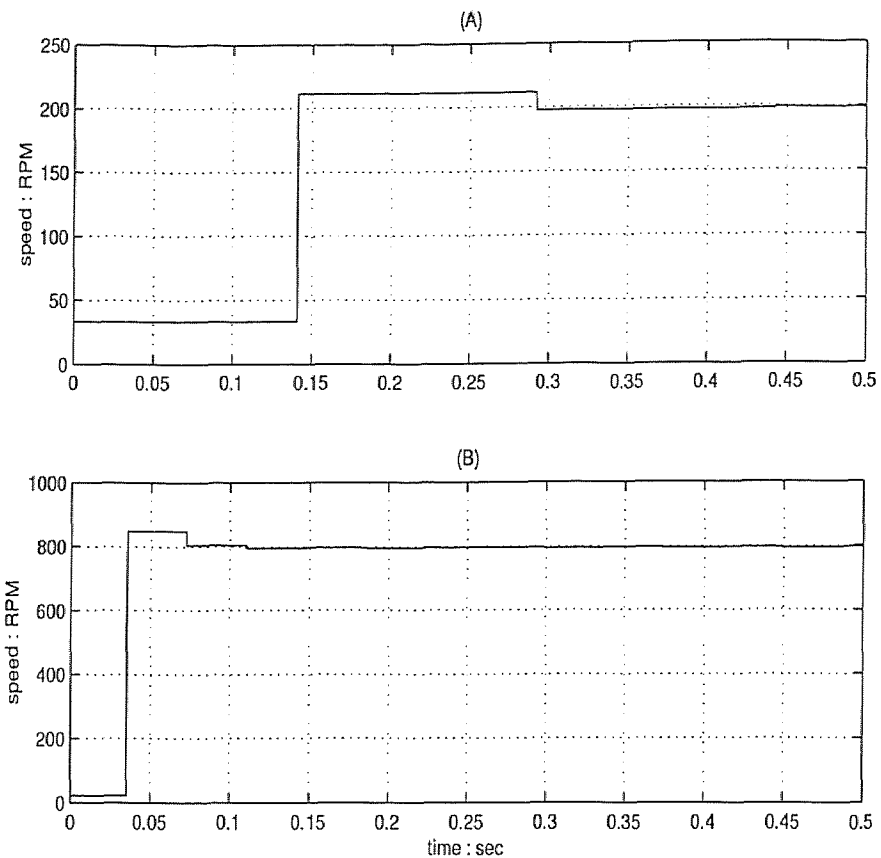


Figure 5.6 The Experimental Speed Response of the Fixed Gain Control ($K_p=0.15$; $K_i=0.01$)

fixed gain control is based on the optimal parameters at 400 RPM ($K_p = 0.36$ and $K_i = 0.01$). The advantages of the variable gain control over the fixed gain control is evident. Figure 5.16 is the overall motor performance comparison. The difference of the RMS value of error is getting larger when motor speed goes higher. According the test results that the variable gain control has bigger edge than the fixed gain control in the motor speed performance, especially in the high speed situation.

Table 5.4 Comparison Between the Fixed Gain and the Variable Gain Control- Simulated Result

Test Speed (RPM)	Type	Rise Time (sec)	Overshoot (%)	RMS
200	Fixed	0.114	25	32.7109
	Variable	0.134	9.5	15.4919
600	Fixed	0.094	16.7	63.4035
	Variable	0.091	11.1	62.9285
800	Fixed	0.074	8.21	88.0341
	Variable	0.061	12.5	84.3801
1000	Fixed	0.053	1.41	112.6943
	Variable	0.04	0.36	100.995
2000	Fixed	0.046	5.34	209.1172
	Variable	0.035	5.26	198.7712

Table 5.5 Comparison Between the Fixed Gain and the Variable Gain Control- Experimental Result

Test Speed (RPM)	Type	Rise Time (sec)	Overshoot (%)	RMS
200	Fixed	0.124	22	28.653
	Variable	0.152	6.2	21.149
600	Fixed	0.114	26.4	75.45
	Variable	0.103	21.41	73.22
800	Fixed	0.083	17.8	92.43
	Variable	0.077	14.25	89.81

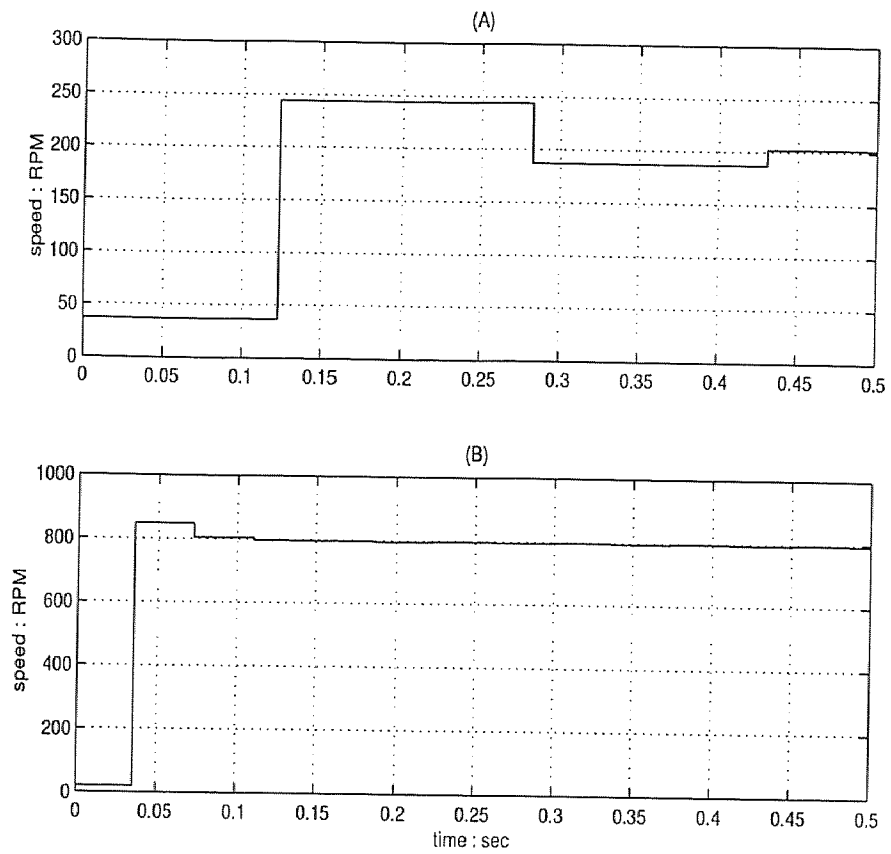


Figure 5.7 The Experimental Speed Response of the Fixed Gain Control ($K_p = 0.39$; $K_i = 0.01$)

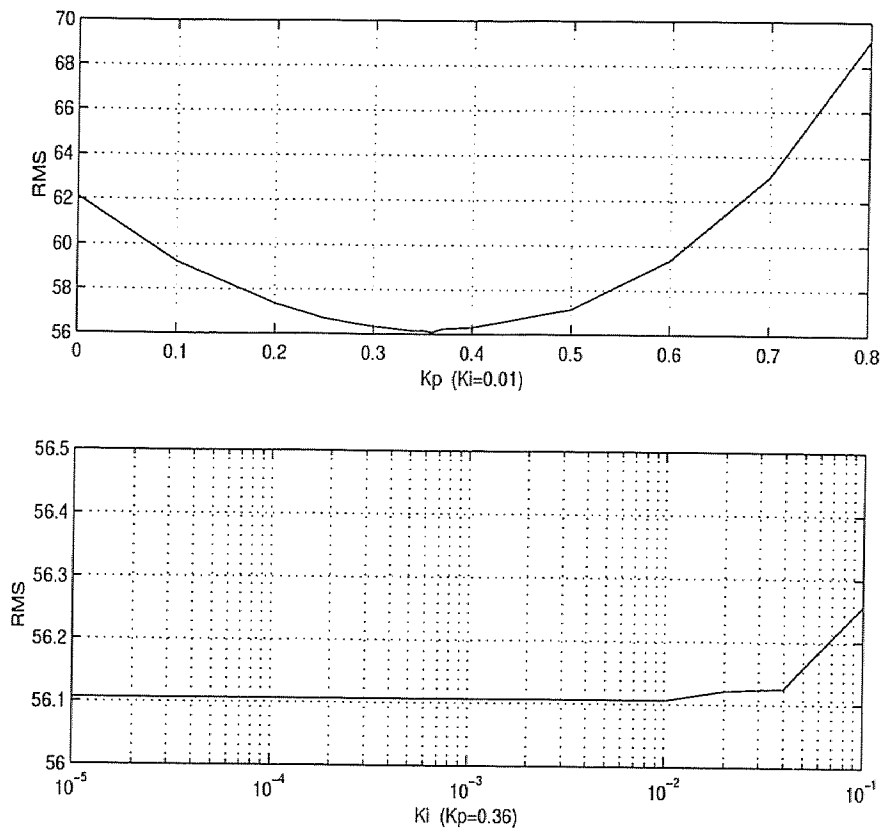


Figure 5.8 The Motor Performance Under Different PI Gains at 400 RPM

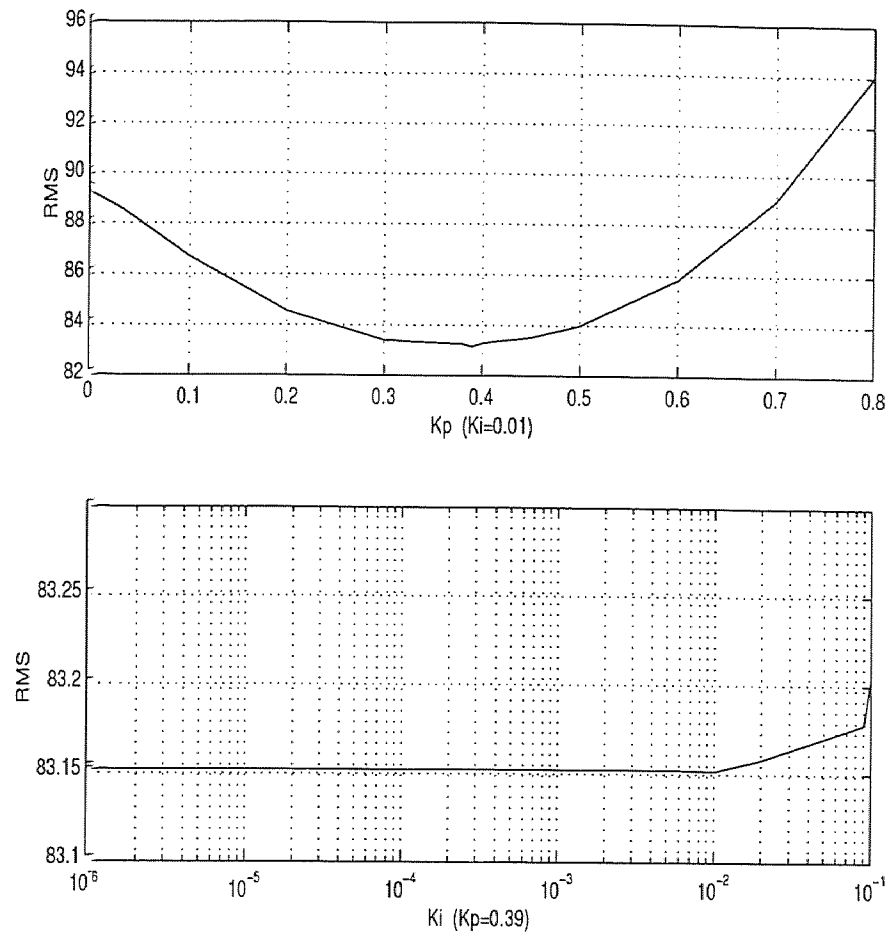


Figure 5.9 The Motor Performance Under Different PI Gains at 600 RPM

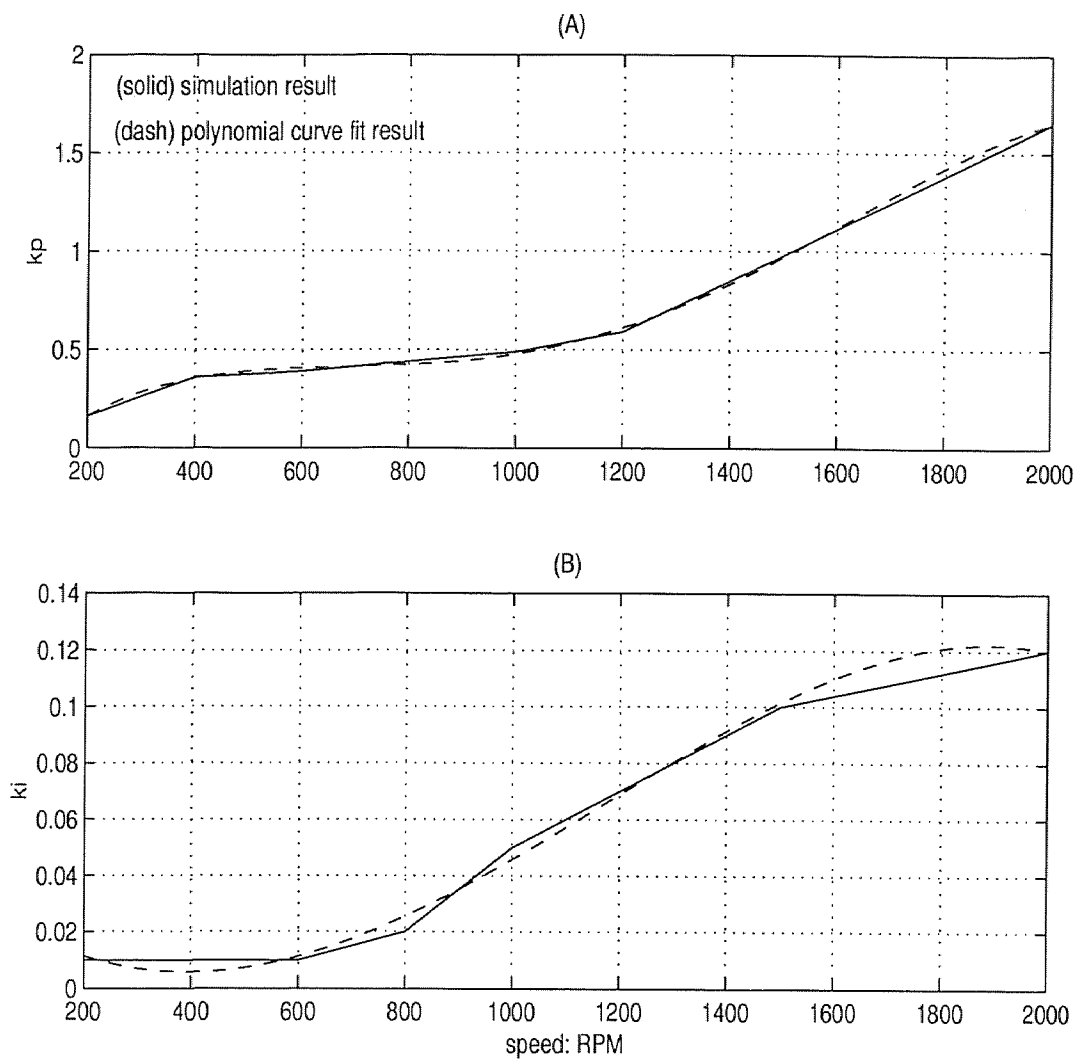


Figure 5.10 The Optimal Gains at Different Speeds and the 4-degree Polynomial Curve Fit Result for (A) K_p and (B) K_i

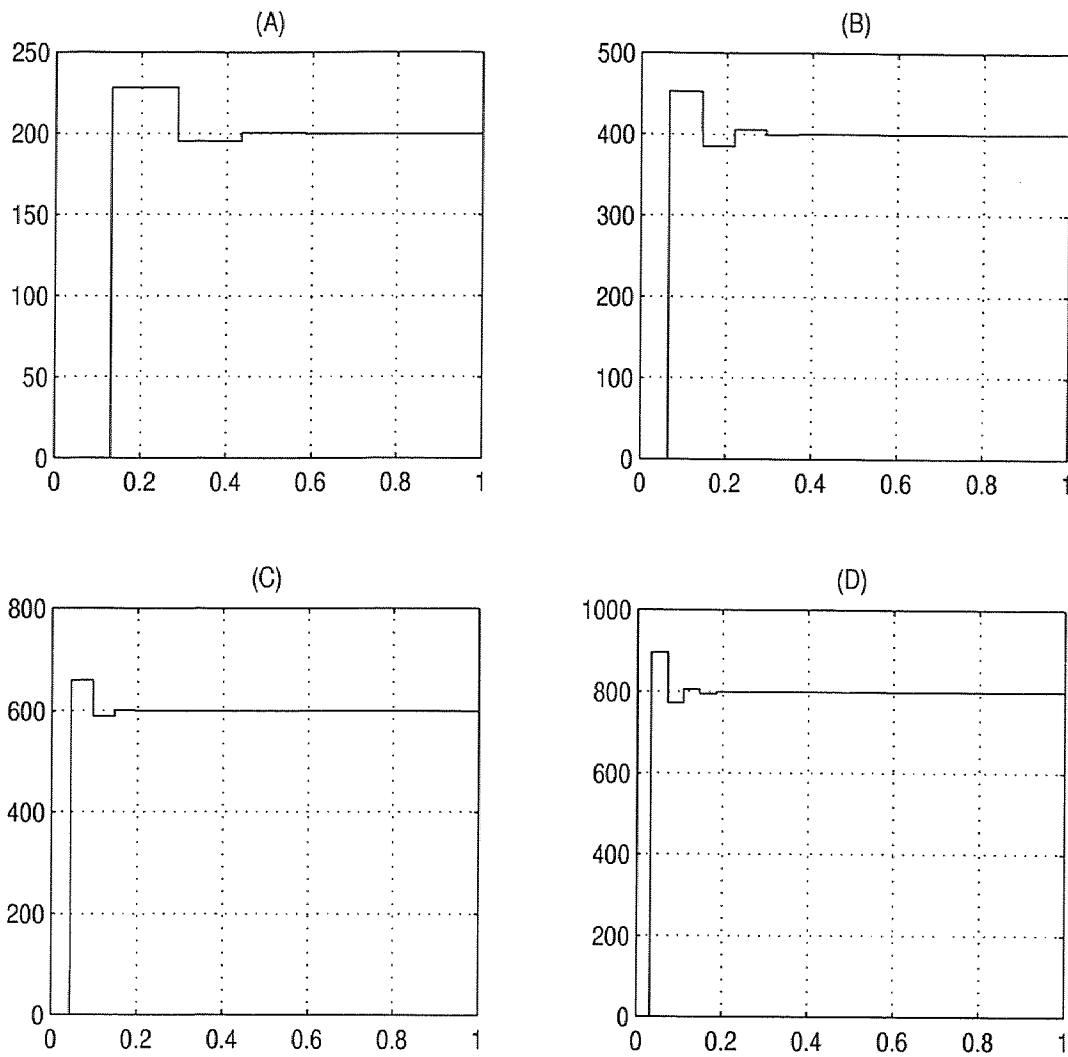


Figure 5.11 Simulated Results using the Optimal Gain Control

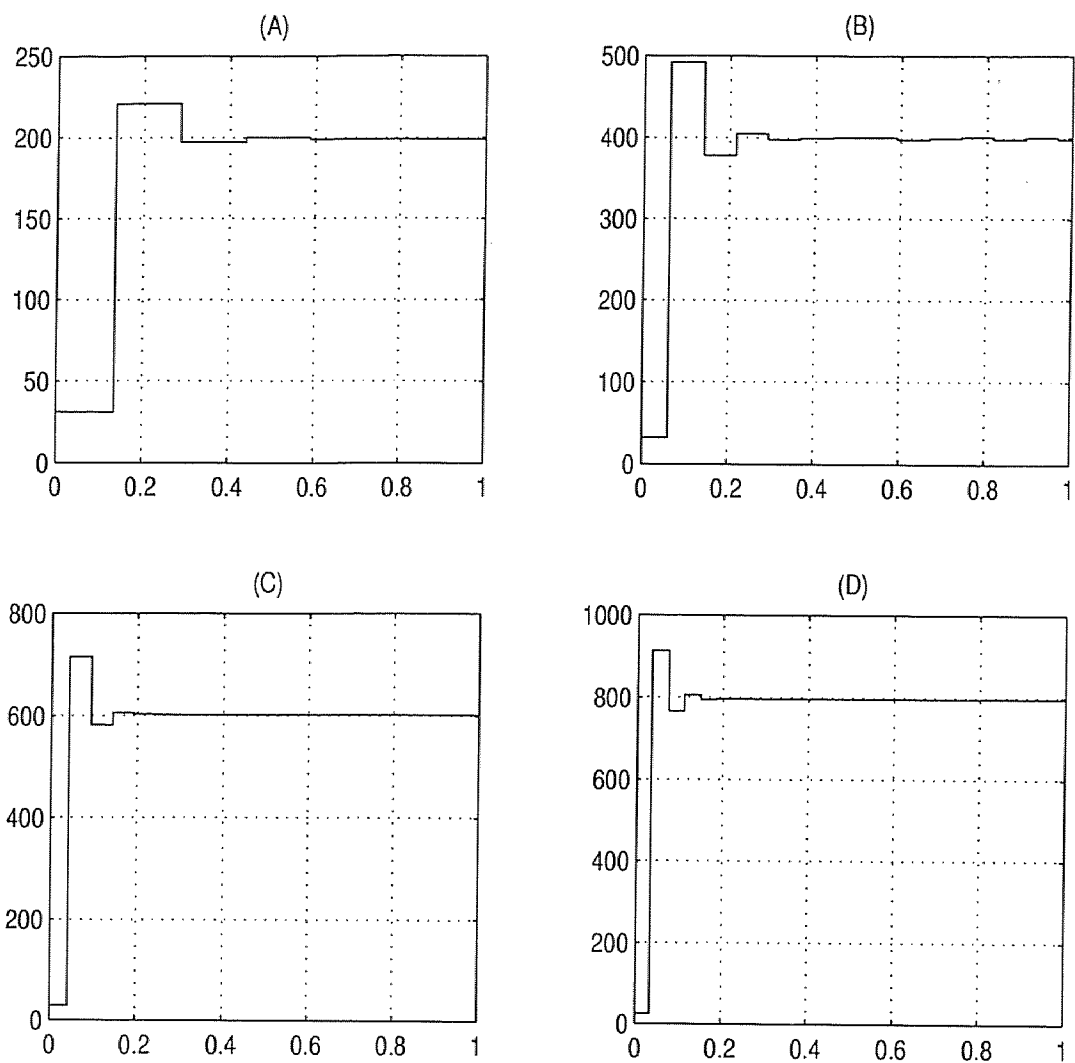


Figure 5.12 Experimental Results Using the Optimal Gain Control

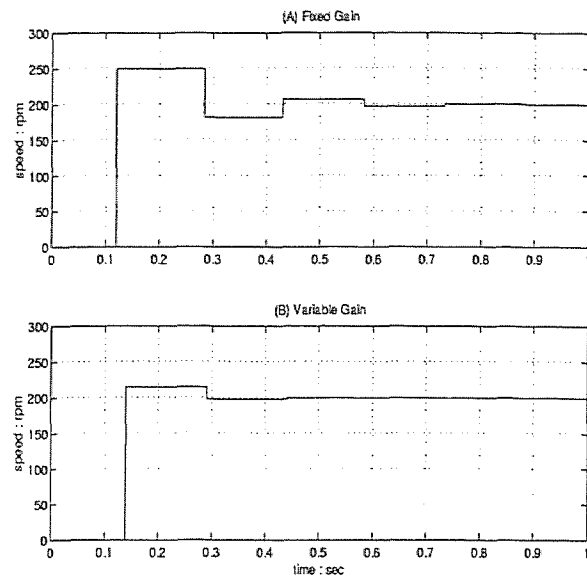


Figure 5.13 The Fixed Gain Control VS. the Variable Gain Control at 200 RPM

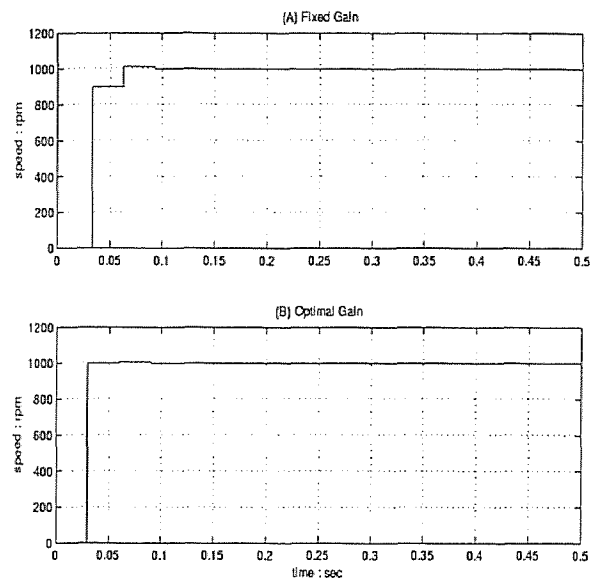


Figure 5.14 The Fixed Gain Control VS. the Variable Gain Control at 1000 RPM

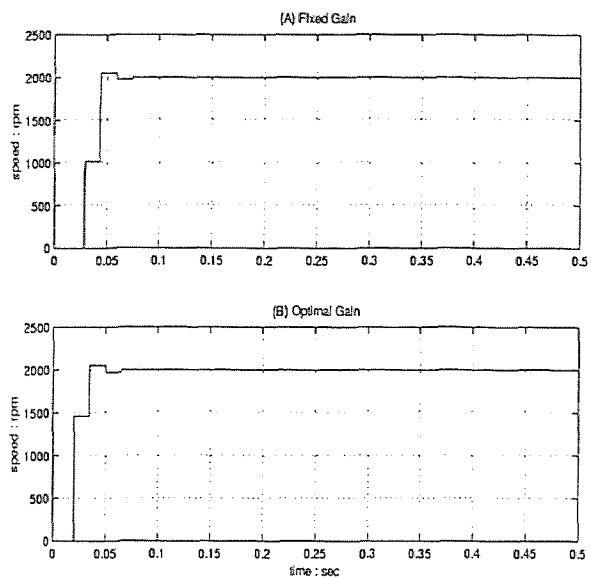


Figure 5.15 The Fixed Gain Control VS. the Variable Gain Control at 2000 RPM

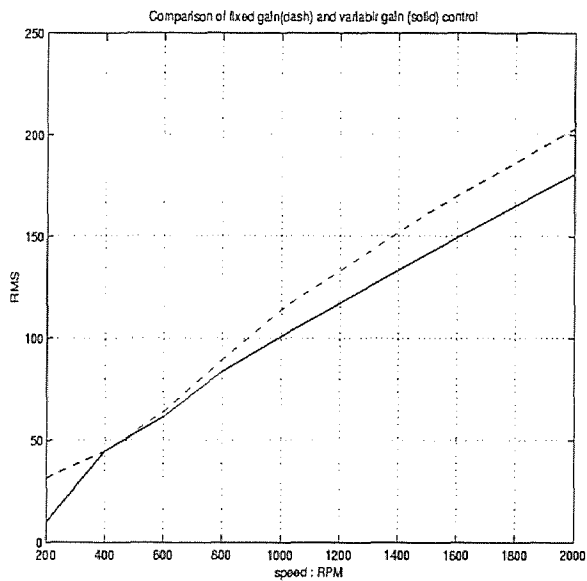


Figure 5.16 The Motor Performance Comparison Using the Fixed Gain and the Variable Gain Control

CHAPTER 6

CONCLUSIONS AND FUTURE RECOMMENDATIONS

In this work, speed control of the brushless dc motor using low resolution sensor is investigated. low resolution sensing poses a unique problem in speed control problems since the data pulses are discrete and speed dependent. Therefore, a variable gain control algorithm is proposed in this thesis. Results from simulation study and experimental work confirm the superiority of the variable gain control over the conventional fixed gain algorithm, and further establishes the feasibility of using low resolution sensor for speed control.

6.1 Future Recommendation

The study successfully demonstrates the application of the variable gain control in the BLDC motor speed control using low resolution sensing devices. Future directions include the following:

1. Increase motor inertia for robustness test. All of the results presented in this work are under no load situation. Adding different load on motor shaft will provide the opportunity to assess the robustness of the variable gain control.
2. Implement input shaping and other control schemes for comparison.
3. Improve motor amplifier for higher voltage output range. The $\pm 25\text{V}$ output voltage limitation of the LM12 restricts the experimental speed to 800RPM or less. A new motor amplifier capable of delivering $\pm 100\text{V}$ is desirable for future experiment.
4. Design a more advanced control and speed determination scheme to ameliorate the motor performance.

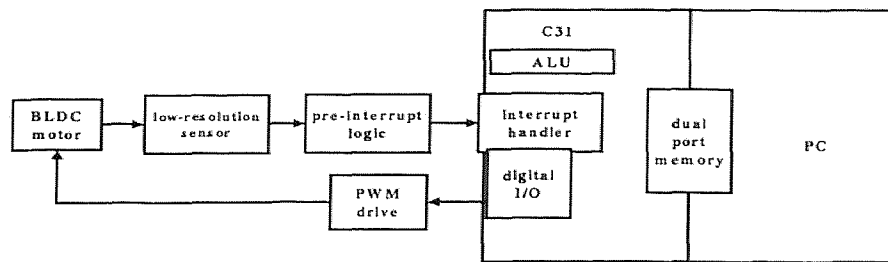


Figure 6.1 General Configuration of the BLDC Motor Drive Using Low Resolution Sensors

To implement more complex control scheme, using the interrupt of the digital hardware is taking into account. Figure 6.1 is a general configuration of the BLDC motor drive using low resolution sensors. The use of interrupt can speed up the data access which makes the speed control more efficient.

APPENDIX A

MOTOR PARAMETER SPECIFICATIONS AND ENCODER TERMINATION FUNCTIONS

A.1 Motor Parameter Specifications

Form	Unit	4-pole,3-phase, Y-connect
Weight	Kg	2.5
Rated Output	W	300
Rated Torque	N-m	0.9516
Peak Torque	N-m	2.8547
Rated Speed	RPM	3000
Maximum Speed	RPM	4000
Rated Terminal Voltage (V_{L-L})	V	104
Rated Armature Current	A	1.9
Torque Constant (K_t)	N-m/A	0.175
Voltage Constant (K_e)	V/rad/s	0.175
Armature Resistance (R)	Ω	2.5
Armature Inductance (L)	mH	6.5
Rotor Moment of Inertia (J)	N-m-s ²	0.0000677
Mechanical Time Constant (t_m)	ms	1.9
Electrical Time Constant (t_e)	ms	2.6
Static Friction Torque	N-m	0.0294
Thermal Resistance	$^{\circ}\text{C}/\text{W}$	1.3
Thermal Time Constant	minute	20
Maximum Winding Temperature	$^{\circ}\text{C}$	130

2 Termination Functions of RCH20 Encoder

	PIN	FUNCTION
D A T A	1	VCC+
	3	GND
	5	CHA
	7	CHA NOT
	9	CHB
	11	CHAB NOT
	13	INDEX
	15	INDEX NOT
C I O O M N M U T A T	2	S1
	4	S2
	6	S3
	8	N/C
	10	S1 NOT
	12	S2 NOT
	14	S3 NOT
	16	CASE GND

APPENDIX B

SAMPLE OF SOFTWARE

B.1 'C' Program for Motor Control Loop

```
/* This is C program of variable gain motor speed control */

#define TIMPER0      0xD0      /* sampling frequency=10K Hz */
#include"d310biov.h"
#define A 0x501
#define MASK      0x0000FFFFL

main()
{

int      m1,n1,m2,n2,m3,n3,m4,n4,m5,n5,m6,n6;
        int      count1=0,count2=0,count3=0,count4=0,
count5=0,count6=0,count=0;
        int      i1,i2,i3,buffer=0,sum;
        float    kp=0,ki=0,kf=0,error=0,K,sum_error=0;
        int      *ptr_to_A;
        int      InputSpeed=0,OutputSpeed=0,speed=0,

InitDsp();

ptr_to_A = (long int *) A;
*ptr_to_A = (long int) (0.0);

        while(1)
        {
/* Read speed command from keyboard */

                InputSpeed=((double)(*ptr_to_A))/1000;

                /* Read rotor position from encoder */

i1 = ReadAdc(0);
i2 = ReadAdc(1);
i3 = ReadAdc(2);

/* define motor drive table and calculate the converting factor K */
```

```

m1=0;
n1=K*409;
m2=K*355;
n2=K*205;
m3=K*355;
n3=K*(-205);
m4=0;
n4=K*(-409);
m5=K*(-355);
n5=K*(-205);
m6=K*(-355);
n6=K*205;

s1=inputSpeed;
s2=s1*s1;
s3=s2*s1;
s4=s3*s1;

/* define and calculate variable gains */

kf=1.71695e-10*s3-1.54822e-07*s^2+3.8923e-04*s+0.87266;
kp=-8.33957e-13*s4+3.87314e-09*s3-5.59248e-06*s2+3.34716e-03*s1-0.315516;
ki=-6.95027e-15*s4-3.83148e-11*s3+1.93072e-07*s2-1.32687e-04*s1+0.03061;

    /* motor drive */

    if ( i1<500 && i2<500 && i3>500 )
{
        count=count6;
        count1++;
        WriteDACs(m6,n6);
}
    else if ( i1>500 && i2<500 && i3>500 )
{
        count2++;
        WriteDACs(m5,n5);
}
    else if ( i1>500 && i2<500 && i3<500 )
{
        count3++;
        WriteDACs(m4,n4);
}
    else if ( i1>500 && i2>500 && i3<500 )

```

```
{
    count4++;
    WriteDACs(m3,n3);
}
else if ( i1<500 && i2>500 && i3<500 )
{
    count5++;
    WriteDACs(m2,n2);
}
else if ( i1<500 && i2>500 && i3>500 )
{
    count6++;
    WriteDACs(m1,n1);
}

buffer=count1*count2*count3*count4*count5*count;

/* speed determination */

if ( buffer>1 )
{
    sum=count1+count2+count3+count4+count5+count6;
    speed=30*10000/sum;
    count1=0;
    count2=0;
    count3=0;
    count4=0;
    count5=0;
    count6=0;
}

/* motor stop */

if ( InputSpeed==0 ) {
    speed=0;
    K=0;
    WriteDACs(0,0);
}

else {
/* motor rotate clockwise */

    if (InputSpeed>0)
    {
```

```

error=speed-InputSpeed;
sum_error=sum_error+error;
        K=(InputSpeed*kf-sum_error*ki-error*kp)*0.025;
    }

    else
/* motor rotate counterclockwise */
{
InputSpeed=-InputSpeed;
        error=speed-InputSpeed;
sum_error=sum_error+error;
        K=(InputSpeed*kf-sum_error*ki-error*kp)*0.025;
    }

}

```

B.2 Sample of MATLAB M-file

B.2.1 Drive Table

```

/* take encoder signal as function input */

function f=table(u)

if (u(1)>1.5 & u(2)<1.5 & u(3)>1.5)
    f(1)=0;
    f(2)=-0.866;
    f(3)=0.866;
elseif (u(1)>1.5 & u(2)<1.5 & u(3)<1.5)
    f(1)=0.866;
    f(2)=-0.866;
    f(3)=0;
elseif (u(1)>1.5 & u(2)>1.5 & u(3)<1.5)
    f(1)=0.866;
    f(2)=0;
    f(3)=-0.866;
elseif (u(1)<1.5 & u(2)>1.5 & u(3)<1.5)
    f(1)=0;
    f(2)=0.866;

```

```

    f(3)=-0.866;
elseif (u(1)<1.5 & u(2)>1.5 & u(3)>1.5)
    f(1)=-0.866;
    f(2)=0.866;
    f(3)=0;
else %(u(1)<0.5 & u(2)<0.5 & u(3)>0.5)
    f(1)=-0.866;
    f(2)=0;
    f(3)=0.866;
end

```

B.2.2 Speed Determination and Calculation of SOES at 200 RPM

```

/* take encoder signal as function input */

function f=e200(u)

global count1 count2 count3 count4 count5 count6
global sum speed sum_err error count

/* 6-step counter for recording the time of encoder signal cycle */

if (u(1)>1.5 & u(2)<1.5 & u(3)>1.5)
    count=count6;
    count1=count1+1;
elseif (u(1)>1.5 & u(2)<1.5 & u(3)<1.5)
    count2=count2+1;
elseif (u(1)>1.5 & u(2)>1.5 & u(3)<1.5)
    count3=count3+1;
elseif (u(1)<1.5 & u(2)>1.5 & u(3)<1.5)
    count4=count4+1;
elseif (u(1)<1.5 & u(2)>1.5 & u(3)>1.5)
    count5=count5+1;
elseif (u(1)<0.5 & u(2)<0.5 & u(3)>0.5)
    count6=count6+1;
end

if (count1*count2*count3*count4*count5*count>10)

    sum=count1+count2+count3+count4+count5+count6;

/* convert encoder signal cycle time into motor speed */

```



```
    speed=30*10000/sum;
    count1=0;
    count2=0;
    count3=0;
    count4=0;
    count5=0;
    count6=0;
    count=0;
end

/* calculate the SOES */

if (speed>0)
    error=(speed-200)^2;
    sum_err=sum_err+error;
    error=0;
end

f(1)=speed;
f(2)=sum_err;
f(3)=error;
```

REFERENCES

1. T. Kenjo and S. Nagamori, *Permanent-magnet and Brushless DC Motors*, Oxford University Press, NY, 1985.
2. J. R. Hendershot, T.J.E. Miller, *Design of Brushless Permanent-Magnet Motors*, Oxford University Press, NY, 1995.
3. D. J. Bak, "Brushless DC Motor Use Back EMF for Commutation," *Design News*, pp.78-79, April 1990.
4. K.Iizuka, H. Uzuhashi, M. Kano, T. Endo, K. Mohri, "Microcomputer Control for Sensorless Brushless Motor," *IEEE Transactions on Industry Applications*, vol.21, no.3, pp.595-601, 1985.
5. Y. Kim, J. Ahn, W. You, K. Cho, "Speed Sensorless Vector Control for Brushless DC Motor Using Binary Observer," *proceedings of the IEEE 22nd IECON*, vol.3, part 3, pp.1746-1751, August 1996.
6. T. Senjyu, S. Ashimine, H. Kamifurutono, K. Uezato, "Robust Speed Control Method for DC Servomotor using Adaptive Gain Low," *Electrical Engineering in Japan*, vol.126, no.3, pp.30-40, February 1999.
7. T. Takeshita, N. Matsui, "Sensorless Control and Initial Position Estimation of Salient-Pole Brushless DC Motor," *proceedings of 4th International Workshop on AMC*, vol.1, part 1, pp. 18-23, March 1996.
8. N. Matsui, "Sensorless Operation of Brushless DC Motor Drives," *proceedings of the 19th IECON*, vol.2, pp. 739-744, November 1993.
9. G. L. Blank, J.J. Wrobel, "Commutation Angle Control of a Brushless DC Motor," *proceeding of 14th Annual Symposium on Incremental Motion Control Systems and Devices*, pp.159-163, 1985.
10. U. Huh, J. Lee, T. Lee, "A Torque Control Strategy of Brushless DC Motor with Low Resolution Encoder," *proceedings of the 1995 International Conference on Power Electronics and Drive Systems*, vol.1, pp. 496-501, 1995.
11. K. M. Kamal, *Analytical and Experimental Study of a Brushless dc Motor Speed Control System*, Master's Thesis, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, 1991.
12. N. Hemati, G.G. Reddy, "Simulation of a Current Controlled Brushless DC Motor in the Stationary Frame," *proceedings of the 23rd Annual Symposium on Incremental Motion Control Systems and Devices*, pp.129-136, June 1994.

13. C. S. Berendsen, G. Chapmenois, J. Davoine, "Commutation Strategies for Brushless DC motor: Influence on Instant Torque," *proceedings of IEEE APEC*, pp.394- 400, March 1990.
14. R. C. Becerra, M. Ehsani, "High-Speed Torque Control of Brushless Permanent Magnet Motors," *IEEE Transactions on Industry Electronics*, vol.35, No.3, pp.402-406, August 1988.
15. E. Aha, "Brushless DC Motor-A Tutorial Study," *Motion*, pp.20-26, March/April, 1987.
16. N. Hemati, M. C. Leu, "A Complete Model Characterization of Brushless dc Motor," *IEEE Transactions on Industry Applications*, vol.28, No.1, pp.172-180, January/February 1992.
17. Dalanco Spry, *Model 310 Data Acquisition and Signal Processing Board for the IBM PC and Compatibles*, 1993.
18. Renco Encoders. <http://renco.com/106030.htm>, 8/8/1999.
19. National Semiconductors. <http://www.national.com/pf/LM/LM12CL.html>, 4/8/1999.