DESIGN AND IMPLEMENTATION OF A DSP BASED CONTROLLER FOR BRUSHLESS DC MOTORS

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND
ELECTRONICS ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Armağan ERGÜN

November 1996

DESIGN AND IMPLEMENTATION OF A DSP BASED CONTROLLER FOR BRUSHLESS DC MOTORS

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

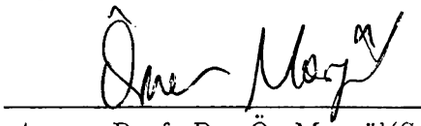
MASTER OF SCIENCE

By

Armağan ERGÜN

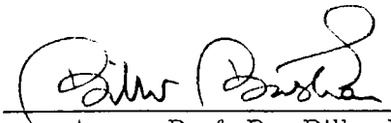
November 1996

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



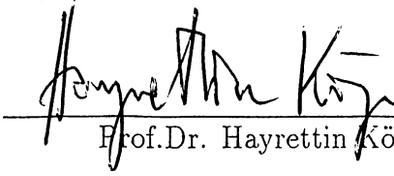
Assoc. Prof. Dr. Ö. Morgül(Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



Assoc. Prof. Dr. Billur Barshan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



Prof.Dr. Hayrettin Köymen

Approved for the Institute of Engineering and Sciences:

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

DESIGN AND IMPLEMENTATION OF A DSP BASED CONTROLLER FOR BRUSHLESS DC MOTORS

Armağan ERGÜN
M.S. in Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. Ö. Morgül
November 1996

This work presents development of a high speed digital signal processor based controller for Brushless DC Motors. A working model of the plant as motor and amplifier is introduced and verified through experiment. Design and implementation of the hardware using wire-wrap board is accomplished and a PID control algorithm is developed. A user interface is designed for easy performance testing of the control system.

Keywords : Motor Control, DSP, High speed control

ÖZET

SAYISAL SİNYAL İŞLEMCİLİ BİR FIRÇASIZ MOTOR DENETLEYİCİ TASARIMI VE UYGULANMASI

Armağan ERGÜN
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Assoc. Prof. Dr. Ö. Morgül
Ekim 1996

Bu tezde, yüksek hızlı bir sayısal sinyal işlemci kullanarak fırçasız bir motor için denetleyici geliştirilmiştir. Motor ve sürücü olarak sistemin bir modeli oluşturulmuş ve deneylerle ispatlanmıştır. İsteklere uygun bir kontrol kartı ve kontrol algoritması tasarlanmıştır. Denetleyicinin performansını kolay bir şekilde değerlendirebilmek icin kullanici ara birimi geliştirilmiştir.

Anahtar Kelimeler : Motor kontrolü, Sayısal sinyal islemcileri, Yüksek hızlı denetleyiciler

ACKNOWLEDGMENTS

I would like to thank to Assoc. Prof. Dr. Ö. Morgül for his supervision, guidance, suggestions and encouragement through the development of this thesis.

I would also like to thank to Assoc. Prof. Dr. Billur Barshan and Prof. Dr. Hayrettin Köymen for reading and commenting on the thesis.

It is a pleasure to express my thanks to all my collegues at TUBITAK-SAGE for their valuable discussions and helps. TUBITAK-SAGE, who supported this work is greatly acknowledged.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Control systems are a necessary part of modern manufacturing, industrial processes, and our daily lives. They range from simple control systems like those on air conditioning to more intricate control systems like those for a missile guidance system. Control mechanism have evolved from mechanical, pneumatic, and electromechanical systems to electronic control systems. Electronic control systems have been implemented with analog components like resistors, capacitors and op-amps. However, with the availability of microprocessors, control systems are being implemented in digital form. The use of microprocessors in digital control systems has created not only some new opportunities due to the powerful processing capabilities of microprocessors, but also a need for a new body of knowledge that utilizes some of these processing capabilities.

A digital controller is a signal processing system that executes algebraic algorithms inherent to the control of feedback systems. Together with the plant (system to be controlled) and signal acquisition circuitry, the digital controller makes up a digital control system such as the one shown in Figure (1-1).

Note that this system requires analog-to-digital (A/D) converters to transform the analog output signals of the plant to digital signals for the input of the controller. The system also requires a digital-to-analog (D/A) converter to

transform the digital output signals of the controller to analog signals for the input of the plant.

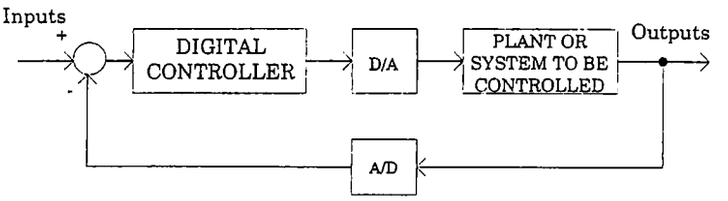


Figure 1-1: Digital Control System

The advantages of the digital control approach over the analog approach are:

1. Ability to implement advanced control algorithms with software rather than special purpose hardware.

2. Ability to change the design without changing the hardware.

3. Reduced size, weight, and power, along with low cost.

4. Greater reliability, maintainability, and testability.

5. Increased noise immunity.

Choosing an appropriate microprocessor is an important factor in efficiently implementing a digital control design. A class of special-purpose (as opposed to general purpose) digital signal-processing microprocessors has been developed to enable fast execution of digital control algorithms. The Texas Instruments TMS320C25 provides several beneficial features for implementing digital control system elements through its architecture, speed, and instruction set.

A prominent feature of the TMS320C25 is the on-chip 16 x 16-bit multiplier that performs two's-complement multiplication and produces a 32-bit product

in a single 100-ns instruction cycle. The TMS320C25 instruction set includes special instructions necessary for fast implementation of sum-of-products computations encountered in digital filtering/compensation and Fourier transform calculations. Most of the instructions critical to signal processing are executed in one instruction cycle.

In this thesis the requirement is to control the position of a brushless DC motor with a bandwidth of 20 Hz and position accuracy of 0.1 degrees in the range of $\pm 6 \cdot \pi$ radians. The overshoot of the control system has to be less than 10 %. The block diagram of the system is shown in Figure(1-2).

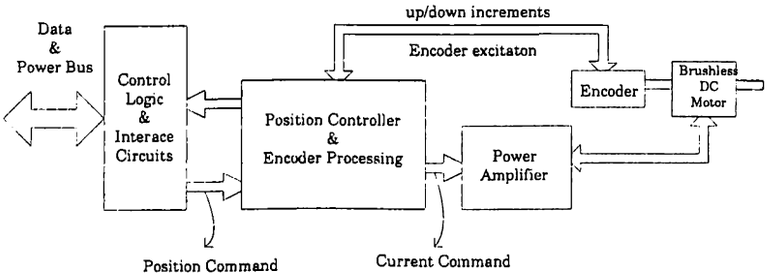


Figure 1-2: System Overview

This system consists of:

- DC-Brushless Motor with incremental Encoder
- Power Amplifier for excitation of DC-Brushless Motor
- Position Control Loops with Encoder Processing implemented with TMS320C25
- Interface to the Data Bus and their Control Circuitry

The primary design goal for the system architecture presented in the following sections is that the controller should provide enough processing power to accommodate advanced control algorithms. A specific statement of that goal is that the system should be able to execute one full loop of position control algorithm within $200\mu s$. This goal was to be achieved while retaining a feasible single processor architecture. The secondary design goal for the system is an easy and flexible motor drive interface.

The goals of the system design forced the following decisions to be made about specifications for the DSP and the input/output hardware. The instruction cycle of the DSP should be 100-ns or lower. The system should have fast memory to execute program code with no wait-states. The decoding of position encoder outputs should be implemented in hardware, as the use of the DSP and interrupts for this task would consume a significant portion of the processing time when the motor drive runs at high rotational speeds. For flexibility, the system should easily interface with both major types of position encoder.

The thesis is organized as follows. Chapter 2 contains a discussion of the brushless motors and their amplifiers with their operations. Chapter 3 introduces a SPICE model for the motor and amplifier as a plant to be controlled. Chapter 4 deals with the design of the control algorithm and simplification of the plant model. Chapter 5 contains a discussion of each major section of the hardware design. Chapter 6 describes and gives examples of software to support all the hardware features of the system. Chapter 7 discusses the experimental verification of the system. Conclusions are presented in Chapter 8.

Chapter 2

BRUSHLESS DC MOTORS AND AMPLIFIERS

2.1 Introduction to Brushless Systems

The key to understanding and using brushless DC motor and amplifier systems is commutation: applying the power supply bus voltage in the proper sequence and at the proper time and controlling the motor current during motor operation.

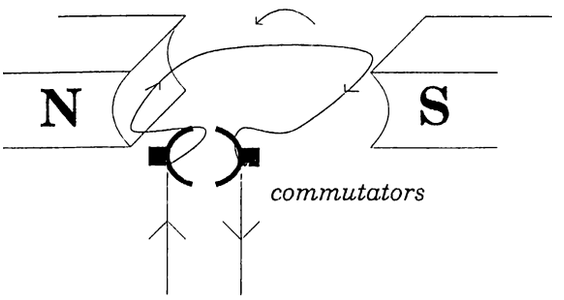


Figure 2-1: Conventional DC Motor

A simple conventional DC brush motor (Figure 2-1) consists of a rotor which can turn within a magnetic field provided by the stator. If the coil connections were to be made through slip rings, this motor would behave like a step motor and

reversing the current in the rotor would cause it to flip through 180°. By including the commutator and brushes, the reversal of current is made automatically and the rotor continues to turn in the same direction.

To turn this motor into a brushless design, we must start by eliminating the windings on the rotor. This can be achieved by turning the motor insideout. In other words we make the permanent magnet as the rotating part of the motor and put the windings on the stator poles. However, we need some means of reversing the current automatically. In a servo application we will use an electronic amplifier or drive, so commutation can be performed by using the low-level signals of an Hall-effect sensor. So unlike the DC brush motor, the brushless version cannot be driven by simply connecting it up to a source of direct current. The current in external circuit must be reversed at defined rotor positions, and the motor is in fact being driven by an alternating current.

On the surface, the concept is simple: replace the brushes and commutator bars of the familiar brush-type DC torque motor with solid-state sensors and switches. Implementation of a reliable system, however, is a greater challenge.

2.2 Advantages of Brushless DC Motors

Brushless motors offer wide advantages in control applications. The absence of brushes provides an increased reliability for brushless DC motors. High speed operation of brush type motors is limited by arcing across commutators. Brushless motors are capable of higher speed operation.

The major advantages of the brushless motors are:

1. Since commutation is performed in circuit elements external to the rotating parts,there are no items which would suffer from mechanical wear. The brushless DC motor will, therefore, have a life expectancy limited only by mechanical bearing wear, and the reliability of the electronic controller.

2. Brushless DC motors provides the absence of brushes which generate Electro Magnetic Interface (EMI) noise and spark. In explosive environments or for equipment sensitive to EMI this is a considerable advantage.

3. Since the heating losses are in the shell, the thermal resistance from the windings to ambient in a brushless motor is less than that of a conventional brush-type DC motor. This means higher currents are possible for the same winding temperature for a brushless motor.

4. The advances of the state-of-the-art magnetical materials have made it possible to design brushless DC motors which have very high torque-to-inertia ratios.

5. The brushless DC motors can be controlled with very efficient amplifier configurations. In cases of severe environmental conditions the controller can be located remotely from the motor. The control system can easily interface with digital and analog inputs, and is therefore well suited for incremental motion.

The brushless DC motor controller has, in general a more complex configuration than the controller for an equivalent conventional DC motor, but may be similar in size and complexity to a closed-loop step motor controller. Step motors are very useful and economical. The designer only has to specify a given number of steps. Step motors can be used efficiently as long as the load moment of inertia and friction levels are within design limits, and the step rate is within the capability of the motor and controller. The brushless DC motor has no inherent capability of moving in discrete steps. However, with the addition of very few parts the brushless DC motor can perform better than step motors in incremental motion systems.

2.3 Operation of Brushless DC Motors

A brushless DC motor has three elements: a fixed, wound member (stator), a rotor with permanent magnets attached to it and a means of sensing rotor position. To understand how these elements work as a motor, consider some elementary magnetics.

When a current-carrying wire is placed in a magnetic field such that the current flow is perpendicular to the direction of the field, a force is exerted between the field producing element (in this case, a permanent magnet) and the wire. This force is the product of the strength of the field, the length of the wire and the current in the wire:

$$F = IlB \tag{2.1}$$

The direction of the force will depend on the orientation of the magnetic field and the direction of the current in the wire. If the wire is disconnected from the current source and the field is moved in a direction perpendicular to the wire (so that the lines of magnetic flux cut the wire), a voltage may be measured across the ends of the wire. The magnitude of the voltage, E, depends on the flux density, B, the length of the wire, l, and the velocity, v, with which the wire is moved through the field, or

$$E = Blv \tag{2.2}$$

The force on the conductor in the magnetic field can be controlled by changes in B, l or I. In motor design, B is affected by the type of permanent-magnet material and magnetic circuit design, l is affected by the length and number of active conductors, and I is the total current. If the current source is removed and the rotor turned, an alternating voltage will appear across the terminals as shown in Figure 2-2. The alternating voltage results from the change in magnetic

polarity as the rotor turns. The amplitude and frequency of the voltage depend on the speed of rotation.

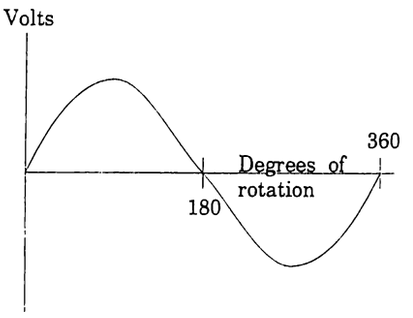


Figure 2-2: Alternating voltage appearing on the terminals

The torque produced by a motor with a given winding and physical geometry is directly related to the voltage it produces when the rotor is externally driven, or when the motor is used as a generator. In fact, the motor torque constant, K_T, and the motor voltage or back EMF constant, K_B, are equal when K_T is expressed in newton-meters per ampere and K_B is expressed in volts per radian per second:

$$K_T = K_B \tag{2.3}$$

Figure 2-3 schematically shows a typical brushless motor wound with three phases and the voltages seen between the phases when the motor is run as a generator at constant speed. Note that the motor is wound to provide overlapping, sinusoidal 3-phase voltages, electrically spaced 120° apart. In this example, since the motor has two poles, the electrical degrees equal to mechanical degrees; that is, the electrical spacing of the phase voltages corresponds to the rotor's physical position.

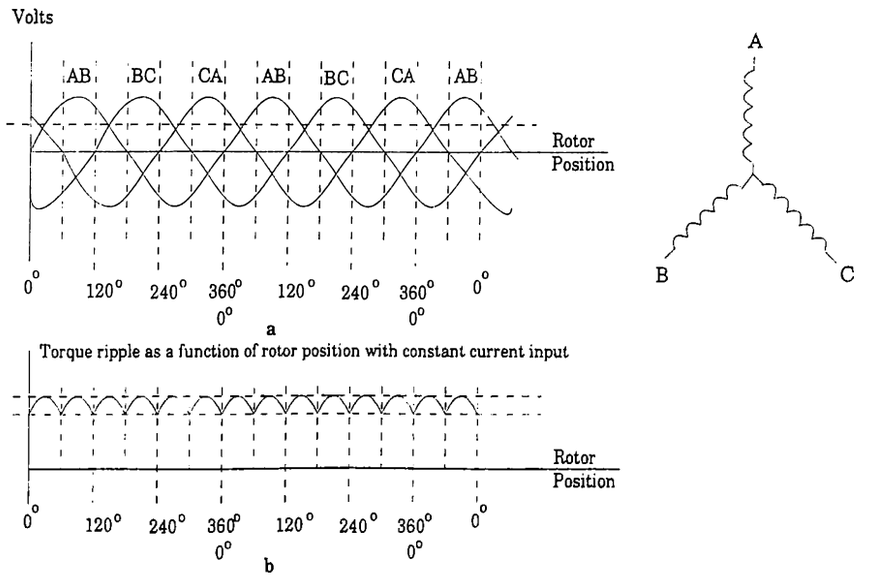


Figure 2-3: Typical Brushless Motor wound with 3 Phases

2.4 Operation of the Amplifier

An amplifier is needed to control the power delivered to the motor and ultimately to control the system. In a brushless amplifier additional circuitry is required to electronically commutate amplifier power to the motor windings so that rotation can occur. Amplifiers coordinate various feedback signals with the command input signals and adjust the power output level to the motor. A brushless amplifier will provide electronic commutation, feedback and command signal comparison, and the power amplification needed for brushless system control.

Commutation circuitry switches power to the different motor windings in a way analogous to a conventional motor's brushes and commutators. Commutation circuitry relies on rotor position information from Hall effect devices or resolvers to switch power between motor windings at the proper time. Com-

mutation sensors are located about the motor axis of rotation, and they are electrically connected to the terminals of the amplifier. A block diagram of a basic pulse width modulation (PWM) amplifier for a brushless motor configured for current loop operation is shown in Figure 2-4.

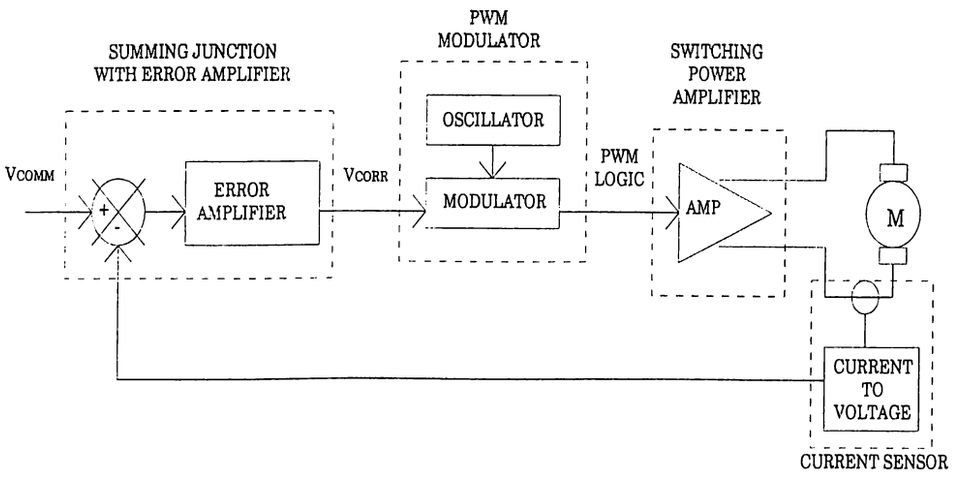


Figure 2-4: Simplified Block Diagram of the PWM Amplifier

Pulse width modulation is a technique of controlling voltage applied to a load so that the amplifier power transistors always operate in a full on or full off state. The advantage of this technique is that power dissipation in the transistors is minimized. A simplified pulse width modulator is shown in Figure 2-5. The figure includes a voltage comparator which receives the analog correctional voltage on its non-inverting input and a constant frequency triangle wave on its inverting input. In operation, the analog voltage is compared to the constant frequency triangle wave. When the analog voltage exceeds the triangle wave voltage, the output of the comparator is high. But, when the analog voltage is less than the triangle wave voltage, the output of the comparator is low. The result is a constant frequency, variable duty cycle pulse train generated at the output of the comparator.

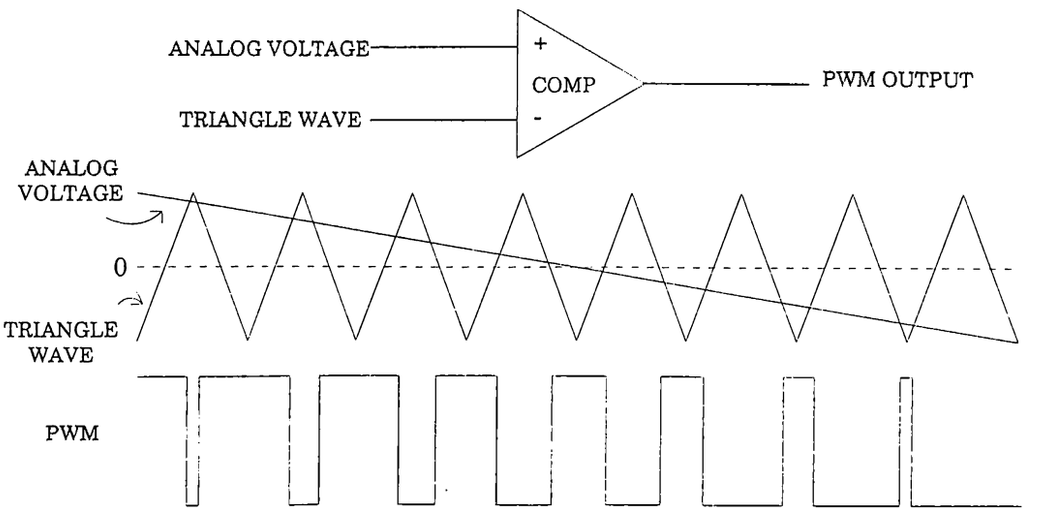


Figure 2-5: Simplified Pulse Width Modulator

The comparator duty cycle is defined as the percentage of time that the output of the comparator is high with respect to the period. So, the duty cycle is proportional to the analog correctional voltage. As the analog voltage becomes more positive, the duty cycle increases, and as the analog voltage becomes more negative, the duty cycle decreases. The resulting pulse modulation is sent to the switching power amplifier.

The switching power amplifier consists of six MOSFET switches arranged as shown in Figure 2-6. This configuration is commonly referred to as an H bridge. When switches 1 and 5 are on while others are off $+V_{supply}$ is applied to phase C. When switches 2 and 4 are on while others are off $-V_{supply}$ is applied to the phase C. These switching states are called state 1 and state 2, respectively. This scheme will be applied to other phases as well. The output of the PWM modulator is used to switch the power amplifier between statel and state 2. Switching amplifier between states 1 and 2 regulates the effective voltage and current output to the

motor.

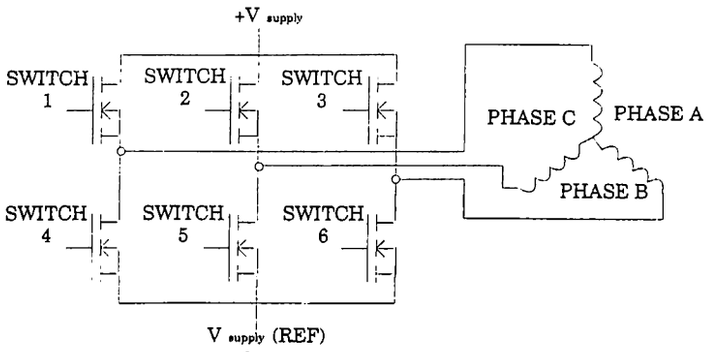


Figure 2-6: Configuration of the Switching Power Amplifier

In general, the best position for commutation is that point at which the back EMF waveform is centered between commutation points. Commutating at the zero crossing of the back EMF waveform is ineffective since there is no resultant torque no matter how much current is applied to the phase. Peak torque per unit current for a running motor is achieved at the peak of the back EMF waveform.

There are two means of commutation: 6-step (or square-wave or trapezoidal) and sinusoidal (or sinewave). Six-step is simpler to implement and is used in this work. Six sequence commutation takes advantage of the three phases as shown in Figure 2-7a. Looking from left to right, a positive or negative peak occurs in one of the phases in every 60 electrical degrees: positive peak in phase A, negative peak in phase C, positive peak in phase B, negative peak in phase A, positive peak in phase C, and negative peak in phase B. These 60 degree ranges then repeat as the motor is rotated in the same direction. Each phase has a positive and negative 60 electrical degree operating range containing a peak. Each of the six ranges represents the optimum rotor position for application of current to that phase to produce torque.

Reversing voltage and current polarity into the three negative operating ranges will produce torque in the same direction. Figure 2-7b illustrates the resulting continuous rotating torque. The correct sequencing of phase current into the six operating ranges to provide continuous rotating torque is called the six sequence commutation method. Current is switched to each phase in this sequence with polarity indicated: A, -C, B, -A, C, -B (repeats).

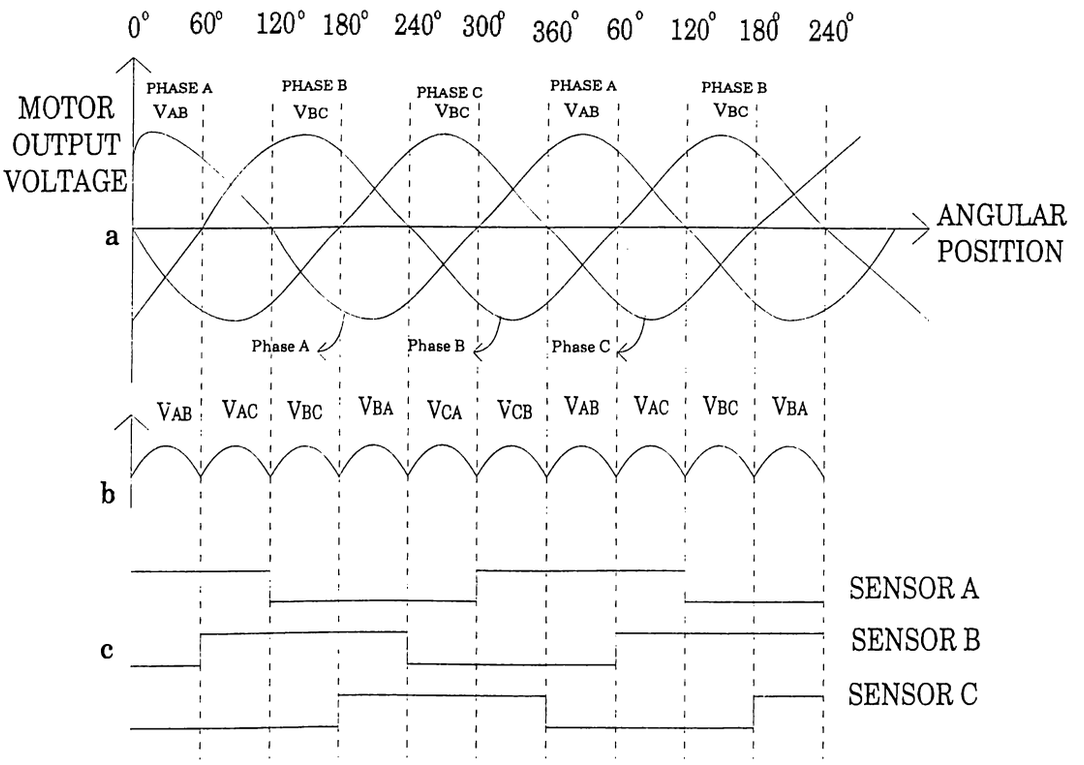


Figure 2-7: Hall effect device outputs

Rotor position feedback from the Hall effect devices indicates the rotor magnet positions relative to the winding phases. The switching amplifier uses this positional information to control the power switched and reversed to the phase next in sequence.

Figure 2-7c shows the outputs from the three Hall devices labelled sensor A, B, and C. The Hall devices as shown indicate exactly six rotor positions and the

Phase	State 1	State 2
A	1,6	3,4
-C	1,5	2,4
B	3,5	2,6
-A	3,4	1,6
C	2,4	1,5
B	2,6	3,5

Table 2.1: Six Sequence Commutation

optimum switching points. Each Hall device is in phase and centered with the positive and negative operating ranges of one phase.

Six MOSFET switches in the switching amplifier provide the voltage and current reversals for rotation. The Hall device feedback and the commutation circuitry determine the switching sequence of the six MOSFETS. Figure 2-6 and Table 2.1 show the MOSFET arrangement

The commutation points in Figure 2-7a center the peak of the back EMF waveform in the commutation zone, and provide equal sharing of the motor phases in the process of producing torque. However, due to the commutation points the variation in the K_T of the motor is approximately 50% for the back EMF waveshape shown. To improve this variation of torque during commutation. called torque ripple, consider the scheme shown in Figure 2-7b. In this case, the motor is commutated twice as often during one revolution by using the negative half of the back EMF waveform as well as the positive half. The torque now falls approximately 13% below the peak. Further improvements can be achieved by modifying the shape of the back EMF waveform so that it is more trapezoidal, so that the back EMF waveform is flat during the period of commutation. The ripple of this type of motor is much less than a motor with a sinusoidal back EMF.

One of the most common forms of brushless DC motor amplifiers is the Current Mode or Current Loop configuration. This type of amplifier uses the PWM technique to control motor current, thereby controlling motor torque output. If

the PWM duty cycle is a function of the difference between the actual and desired motor current, the loop on motor torque is effectively closed. However, a sensing device to monitor motor current must be introduced. The monitored motor current signal is compared to the desired or commanded current; the error is amplified and compensated, and the result is used to vary the duty cycle of the PWM signal to the power transistors.

Chapter 3

MODELLING OF THE BRUSHLESS MOTOR AND AMPLIFIER USING SPICE

Although SPICE is designed as an electronic circuit simulator, it can also be used to simulate mechanical or electro-mechanical systems. Model of the motor and amplifier will be treated in the sequence.

3.1 Model of the Brushless Motor

The first step in modeling the motor is to develop an electrical equivalent to the mechanical system. The basic equation which describes the mechanical system is:

$$T_{total} = J\frac{d^2\theta}{dt^2} \tag{3.1}$$

where

T_{total} is total torque (including friction) applied to the motor shaft from all sources ($g \cdot cm$)

J is the moment of inertia of the mechanical system ($g \cdot cm \cdot s^2$)

θ is the motor shaft angle (radians)

This equation can also be expressed as the following two equations:

$$T_{total} = 2\pi J \frac{dS}{dt} \tag{3.2}$$

$$S = \frac{1}{2\pi} \frac{d\theta}{dt} \tag{3.3}$$

where

S is the shaft speed (rev/sec).

Note that the circuit equation for a capacitor is:

$$i = C \frac{dv}{dt} \tag{3.4}$$

Hence, we can implement equation (3.2) by modeling torque as a current and the moment of inertia $2\pi J$, as a capacitor. This will give the shaft speed as the voltage across the capacitor. This is convenient because we can model any additional moment of inertia as an additional capacitor in parallel with the first one. Also, we can add various torque and drag forces as parallel current sources. This makes the model easier to use in a system.

We can use equation (3.4) again on equation (3.3) to give the shaft angle as the voltage across a capacitor which has a current equal to the shaft speed applied to it.

To use the model, we apply a current proportional to the shaft torque between nodes SHAFT_SPEED and 0 (1 amp = 1 $g \cdot cm$) shown in Figure (3-2). The voltage on that node will correspond to the shaft speed (1 volt = 1 rev/sec), and the voltage on SHAFT_ANGLE will be shaft angle (1 volt = 1 radian).

Now we need to model the mechanical losses of the motor. The simplest ones are linear losses: damping and eddy current losses. They are described by the equation:

$$T_{damping} = 2\pi B \cdot S \tag{3.5}$$

where

B is the damping and eddy current losses ($g \cdot cm \cdot \sec/rad$) and is a constant

S is the shaft speed (rev/sec)

We translate into our model units and get:

$$I(SHAFTSPEED) = 2\pi B \cdot V(SHAFTSPEED) \tag{3.6}$$

This is just the equation for a resistor attached to node SHAFT_SPEED and ground, with a resistance value of $\frac{1}{2\pi B}$.

Another mechanical loss is the frictional loss. This loss is a fixed torque which opposes the direction of rotation. To model this loss we use a table to specify the shape of the loss curve and a current source to multiply the loss curve by the loss factor $F(g \cdot cm)$

Another torque is the magnetic detent torque which tends to align the rotor magnetic poles with the stator poles. This torque is periodic, and is described by the equation:

$$T_{detent} = D \cdot \sin(N_d \cdot A \cdot \theta) \tag{3.7}$$

where

D is the magnetic detent torque ($g \cdot cm$) and is a constant

A is the number of north poles on the rotor and is a constant

N_d is an integer determined by the number of stator slots and the structure of the rotor.

We can translate this directly into a current source. Now we need to model the electrical properties of the stator windings. The properties which are required for a first order model are winding inductance, winding resistance, winding capacitance, winding mutual inductance, winding back emf, and the torque on the rotor from the winding current. The first four of these are simple electrical properties of the winding which are modeled directly by SPICE. For the last two, following [1], we get the following equations for back emf and torque:

$$V_{bn} = C_b \cdot S \cdot \sin(A \cdot \theta - (N-1)\frac{2\pi}{P}) \tag{3.8}$$

$$T_{dn} = C_t \cdot i_n \cdot \sin(A \cdot \theta - (N-1)\frac{2\pi}{P}) \tag{3.9}$$

where

V_{bn} is the back emf voltage for the phase n winding
C_b is the back emf voltage constant (volts·sec/rev)
T_{dn} is the drive torque from the phase n winding
C_t is the torque constant (g·cm/amp)
i_n is the current in the phase n winding (amp)
S is the shaft speed (rev/sec)
A is the number of north poles on the rotor
N is the phase number (1, 2, 3 in our case)
P is the number of motor phases

Figure (3-1) shows the electrical part of the motor model drawn using the Schematic Tool of SPICE. Figure (3-2) shows the mechanical part of the motor model drawn using the Schematic Tool of SPICE.

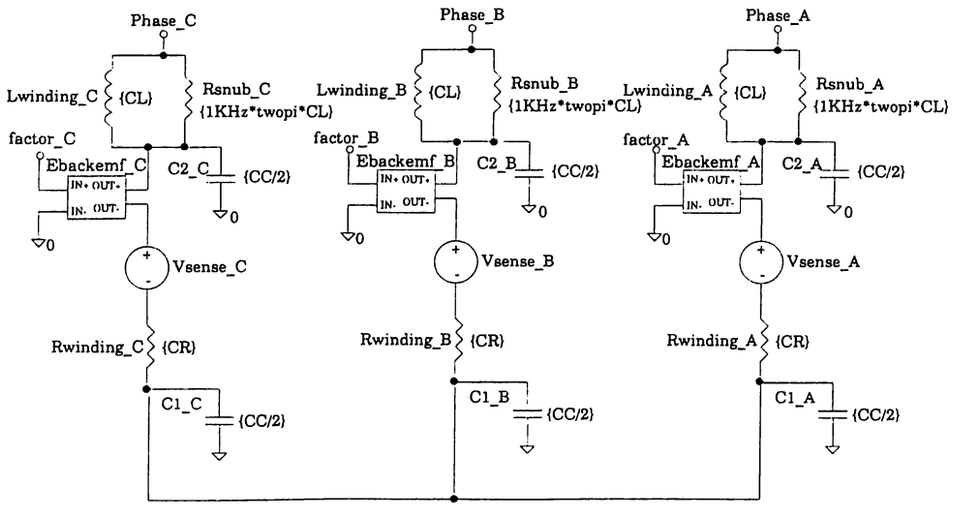


Figure 3-1: Schematic drawing of the electrical part of the motor

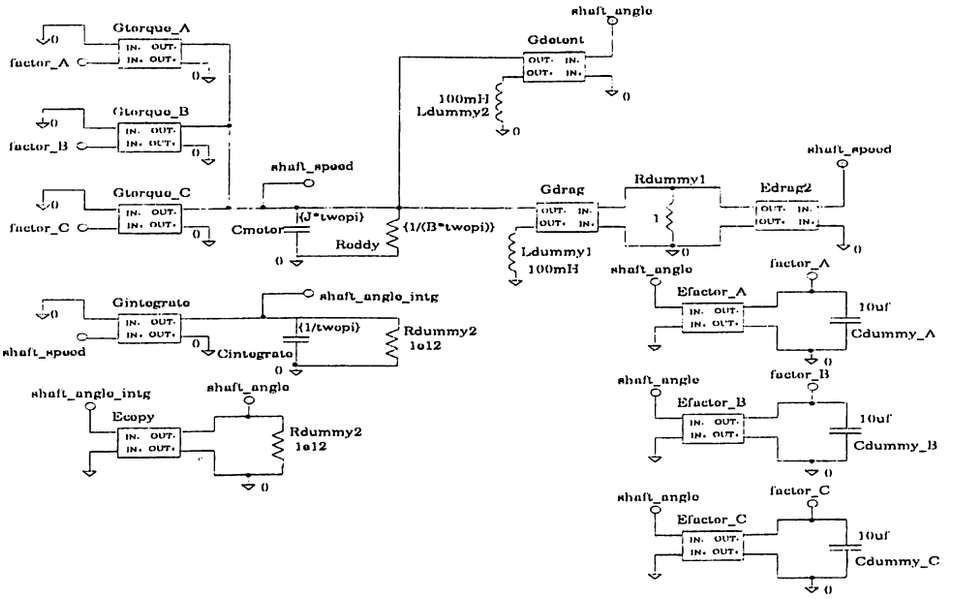


Figure 3-2: Schematic Drawing of the Mechanical Part of the Motor

3.2 Model of the Brushless Amplifier

To model the brushless amplifier, we first need to model the Hall effect device outputs of the motor. Hall effect devices are used to sense the magnetic field of the rotor and produce TTL level outputs. For example, if the backemf voltage waveform of Phase A is positive the sensor output is logic high otherwise low. This can easily be modeled by a voltage controlled voltage source.

In order to model the 6-step commutation strategy we need to design the logic which will implement the desired switching sequence according to the feedback from Hall effect sensors. The PWM input must also be introduced to distinguish between State 1 and State 2, as described in Table2.1. Using logic high to turn on and logic low to turn off the transistors we obtain the following truth table shown in Figure(3-3a)

Hall effect outputs of the corresponding phases				Numbers of the Switching MOSFETs					
P	H_A	H_B	H_C	1	2	3	4	5	6
0	1	0	0	1	0	0	0	0	1
0	1	1	0	1	0	0	0	1	0
0	0	1	0	0	0	1	0	1	0
0	0	1	1	0	0	1	1	0	0
0	0	0	1	0	1	0	1	0	0
0	1	0	1	0	1	0	0	0	1
1	1	0	0	0	0	1	1	0	0
1	1	1	0	0	1	0	1	0	0
1	0	1	0	0	1	0	0	0	1
1	0	1	1	1	0	0	0	0	1
1	0	0	1	1	0	0	0	1	0
1	1	0	1	0	0	1	0	1	0

a

P : PWM output (digital)

Reduced Logic Equations:

$1 = P \cdot H_A \cdot H_C + P \cdot H_A \cdot H_C$

$2 = P \cdot H_B \cdot H_C + P \cdot H_B \cdot H_C$

$3 = P \cdot H_A \cdot \bar{H}_B + \bar{P} \cdot H_A \cdot H_B$

$4 = \bar{P} \cdot \bar{H}_A \cdot H_C + P \cdot H_A \cdot \bar{H}_C$

$5 = P \cdot \bar{H}_B \cdot H_C + \bar{P} \cdot H_B \cdot \bar{H}_C$

$6 = \bar{P} \cdot H_A \cdot \bar{H}_B + P \cdot \bar{H}_A \cdot H_B$

b

Figure 3-3: Truth Table and reduced Logic Equations of Switching Amplifier

Taking advantage of the classical method of Karnough maps the truth table can be expressed by the logic equations shown in Figure(3-3b). The implementation of the equations in SPICE is made by using the standard TTL AND-OR

gates. To control the current delivered to the motor a PWM model has to be introduced. This is easily accomplished by using a comparator OP-AMP as shown in Figure(2-5). The spice model is very similar with an 20kHz triangular-wave oscillator connected to the inverting input and the analog feedback signal connected to the noninverting input of the OP-AMP.

3.3 Simulations

In order to verify the models of the motor and amplifier two sets of simulations are done. First set considers the transient behavior of the system where a step current command of 5 Amps is applied. From the Figure(3-4) and Figure(3-5) it can be seen that the shaft speed of the motor is changing linearly with time, whereas the shaft angle has a parabolic nature. This agrees with our previous discussion. As can be seen from the Figure(3-6) the actual phase currents settle at the commanded current value of 5 Amps. The PWM signal of 20kHz appears as noise and has no effect on the shaft. From the Figure(3-7) it can be seen how the current feedback signal effectively changes the the PWM duty cycle.

As the simulation time is increased the nonlinear behavior of the system can be observed. The shaft speed saturates because of the increased backemf term and the shaft angle changes linearly with time as illustrated in Figure(3-8) and Figure(3-9).

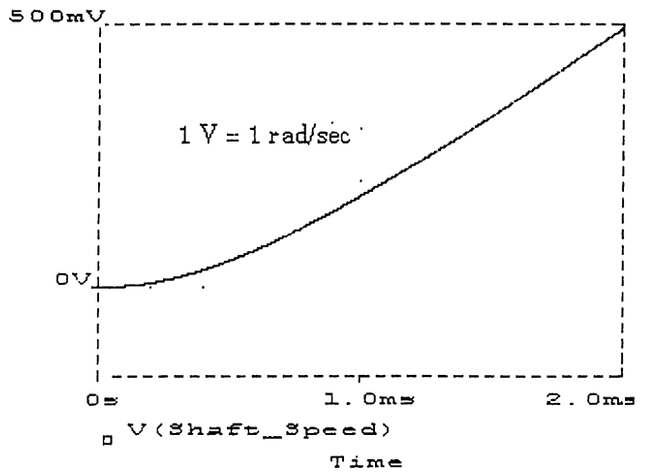


Figure 3-4: Shaft Speed Graph for Short Term Simulations

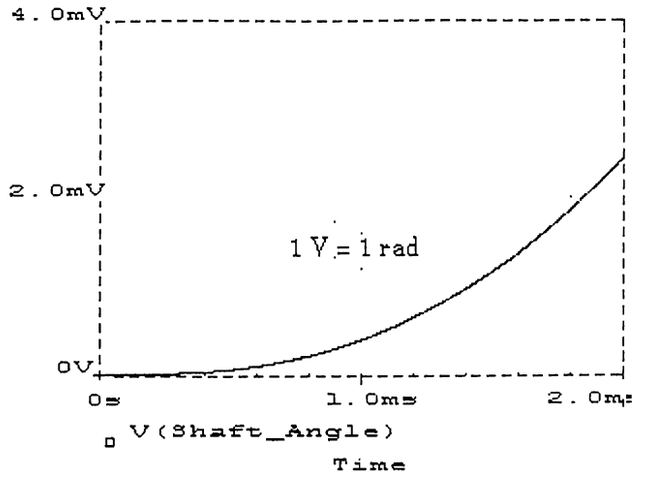


Figure 3-5: Shaft Angle Graph for Short Term Simulation

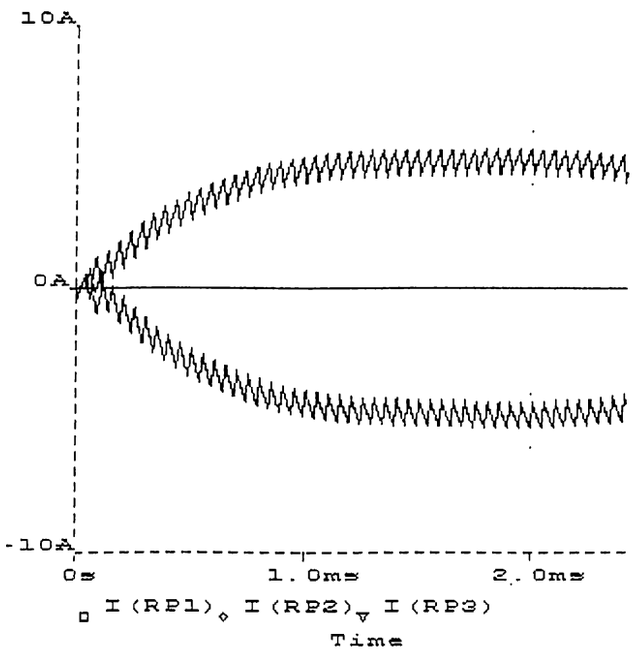


Figure 3-6: Phase Currents for Short Term Simulation

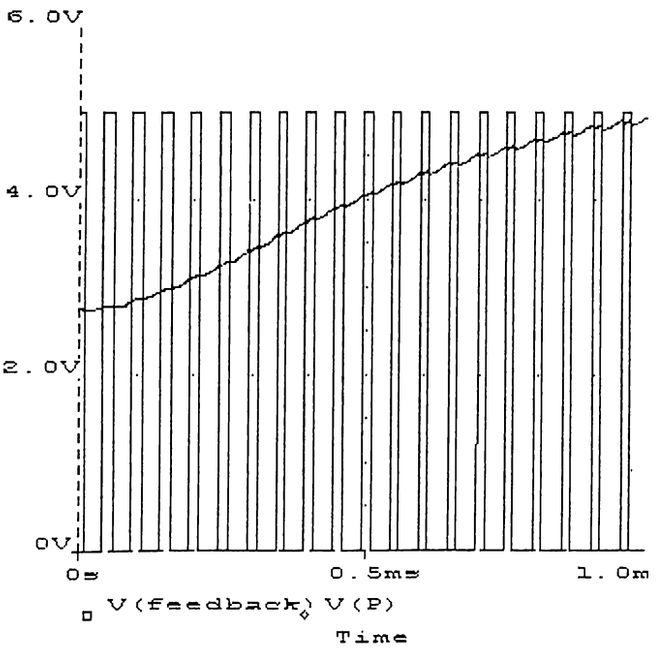


Figure 3-7: PWM and Feedback Signals for Short Term Simulation

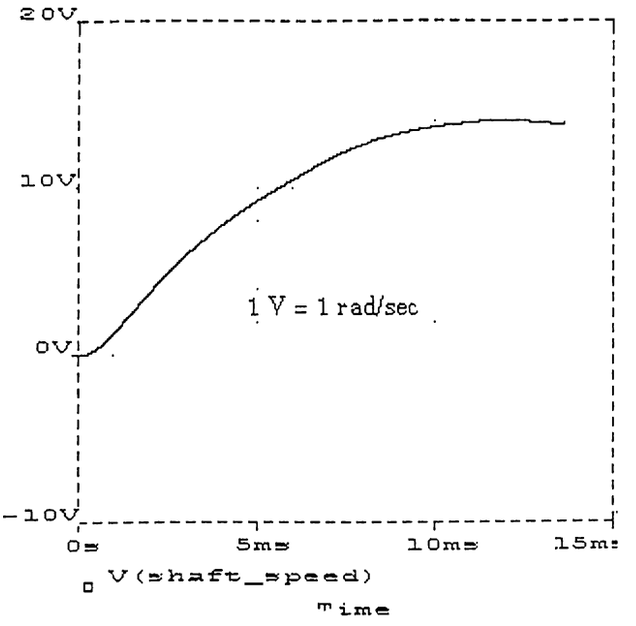


Figure 3-8: Shaft Speed Graph for Long Term Simulation

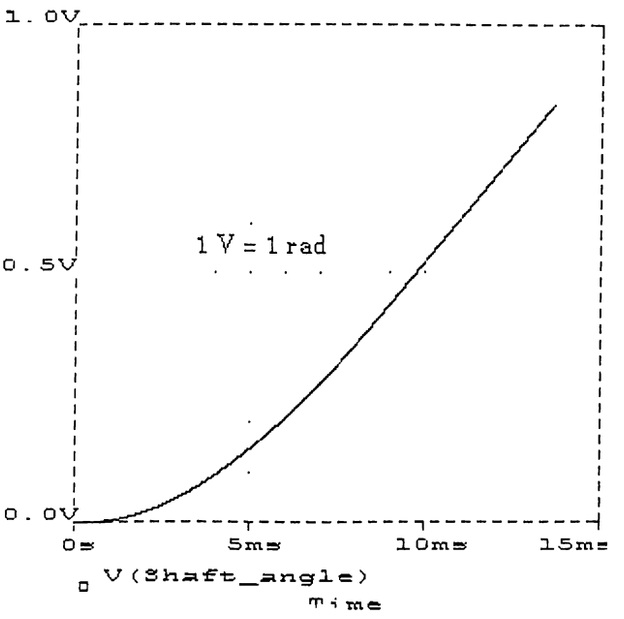


Figure 3-9: Shaft Angle Graph for Long Term Simulations

Chapter 4

DESIGN OF THE POSITION CONTROLLER

4.1 Simplification of the motor model

In order to design the controller we have to simplify the model described in the previous chapter. Once the brushless motor is properly commutated the normal brush-type motor models can be used for closed loop analysis.

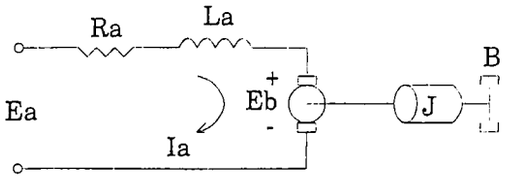


Figure 4-1: Brush type motor model

As can be seen from the Figure(4-1) the electrical and mechanical equations which describe the behavior of the system are:

$$Ea = Eb + Ra \cdot Ia + La\frac{dIa}{dt} \tag{4.1}$$

$$J\frac{d^2\theta}{dt^2} + B\frac{d\theta}{dt} = Kt \cdot Ia \tag{4.2}$$

where

Ea : applied voltage (armature voltage)

Eb : backemf voltage

Ra : winding resistance

La : winding inductance

Kt : torque constant

J : total inertia (motor+load)

B : friction

Taking Laplace transform of equations (4.1) and (4.2), we obtain:

$$Ea(s) = Eb(s) + Ia(s) \cdot [Ra + sLa] \tag{4.3}$$

$$[Js^2 + Bs] \cdot \theta(s) = Ia(s) \cdot Kt \tag{4.4}$$

In our case the amplifier has a current loop and can be treated as a current source. Hence, the above equations reduce to:

$$\frac{\theta(s)}{Ia(s)} = \frac{Kt}{[Js^2 + Bs]} \tag{4.5}$$

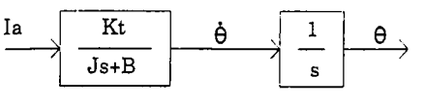


Figure 4-2: Simplified motor model

Figure(4-2) shows the equation(4.5) in block diagram.

4.2 Simplification of the amplifier model

In order to simplify the current loop, consider the circuit that performs the PWM. Such a circuit is shown in Figure(4-3).

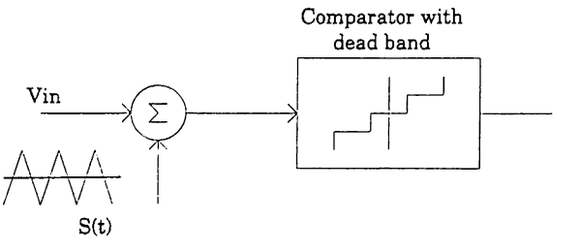


Figure 4-3: Phase Generating Circuit

The circuit consists of a comparator with a deadband and an input element that sums Vin with a triangular waveform S(t). When the combined input is above a threshold, state 1 is chosen. When the combined input is below a negative threshold, state 2 is chosen.

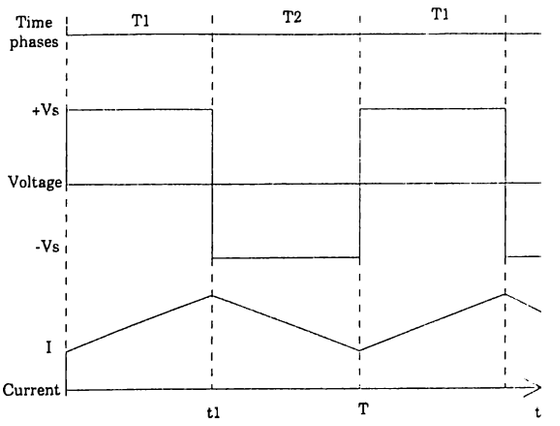


Figure 4-4: Continuous current mode

The switching period, T, is divided into two states: state 1 and state 2. These

states are shown in Figure(4-4).

Consider again the H-bridge configuration for one phase in Figure(4-5)

The states are defined as follows:

state 1 :

switch 1, switch 4 ON

switch 2, switch 3 OFF

state 2 :

switch 2, switch 3 ON

switch 1, switch 4 OFF

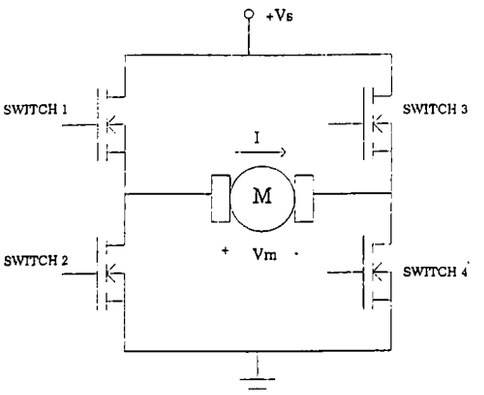


Figure 4-5: H-type PWM amplifier

Accordingly, the output voltage during state 1 is:

$$Vm = Vs \tag{4.6}$$

and during state 2 is:

$$Vm = -Vs \tag{4.7}$$

From the Figure(4-3) it can concluded that the times T1 and T2 are given by:

$$T1 = \frac{T}{2} + K \cdot Vin \tag{4.8}$$

$$T2 = \frac{T}{2} - K \cdot Vin \tag{4.9}$$

Note that the constant K has a unit of time over voltage. K is given by [3]:

$$K = \frac{T}{\triangle S} \tag{4.10}$$

where $\triangle S$ is the peak-to-peak magnitude of S(t). Accordingly, the average value of the output voltage equals to:

$$Vm = \frac{Vs \cdot T1}{T} - \frac{Vs \cdot (T - T1)}{T} \tag{4.11}$$

$$= \frac{2 \cdot Vs \cdot T1}{T} - Vs$$

substituting (4.8) we get the following:

$$Vm = \frac{2K \cdot Vs}{T} \cdot Vin \tag{4.12}$$

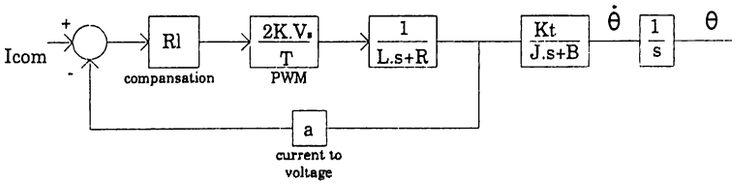


Figure 4-6: Block diagram of the simplified amplifier and motor

For our case Vs=24 Volts, T=50μsec, K=$4.16 \cdot 10^{-6}$ so we end up with a gain

constant of 4 for the PWM part of the brushless amplifier.

After combining the current feedback device and compensation factor for the current loop we get the complete model of the brushless system shown in Figure(4-6).

In the simplified amplifier model the compansator for the current loop is represented by Rl and the current sensor by the feedback gain a.

4.3 Controller Design

The next step in designing a digital control system is to design the controller. Before designing the controller, an appropriate structure for the controller must be selected. This will be influenced by the performance requirements of the system and the processing capability of the processor. The controller may be designed in the continuous domain and then converted into discrete form. Alternatively, the entire design may be carried out in the discrete domain. In our case the design is carried out in continuous domain and then converted to discrete domain.

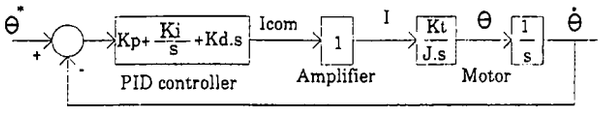


Figure 4-7: Simplified position control system

The structure of the controller is chosen to be Proportional, Integral and Derivative (PID). The PID controller is by far the most commonly used control algorithm. Although it is of limited complexity it can be used to solve a large number of industrial control problems. The closed loop transfer function of the control system given in Figure(4-6) is as follows:

$$\frac{\theta(s)}{Icom(s)} = \frac{2 \cdot Rl \cdot K \cdot V_s}{(L \cdot s + R) \cdot T + 2 \cdot a \cdot Rl \cdot K \cdot V_s} \cdot \frac{Kt}{s \cdot (J \cdot s + B)} \tag{4.13}$$

In order to determine the control gains, further simplification is made on the closed loop transfer function. For our case L=0.54mH, R=0.933Ω and T=50μ sec, so the first sum term in the denominator can be neglegted when compared to the second. Therefore, for a=1 the amplifier gain can be treated as unity.

The desired bandwidth of the position control system is 20Hz, whereas the bandwidth of the amplifier is greater than 2500Hz. The brushless amplifier in current loop mode has therefore a very high bandwidth and can be considered as unity gain. Additionally, friction(B) can be neglected when compared with intertia(J) thus the block diagram shown in Figure(4-7) is the simplified system which will be considered in the control system design.

It can be seen that the closed loop system is of third order. For design purposes, it is important to sort out the poles that have a dominant effect on the transient response. In design, we can use the dominant poles to control the dynamic performance of the system, whereas the insignificiant poles are used for the purpose of ensuring that the controller transfer function can be realized by physical components.

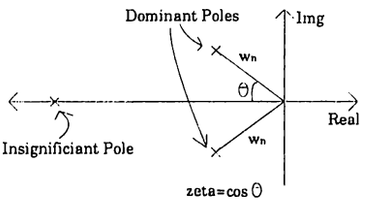


Figure 4-8: Poles of the Closed Loop System

The characteristic function of the closed loop system is in the form of:

$$(s^2 + 2\zeta w_n s + w_n^2) \cdot (s + p)$$

The dominant poles are specified by the w_n (natural frequency) and ζ (damping ratio) parameters. For $w_n = 150$ rad/sec and $\zeta = 0.707$ we get

$$p_1 = -106 + 106j$$

$$p_2 = -106 - 106j$$

as the complex conjugate dominant poles. From Reference [11] it has been recognized that if the magnitude of the real pole is at least 5 to 10 times that of a pair of complex dominant poles, then the pole may be regarded as insignificant insofar as the transient response is concerned. Selecting the real pole at

$$p_3 = -700$$

we obtain a third order equation, which is equated to the closed loop characteristic equation of our system. After some fine tuning the following gains are used:

Kp=9
Kd=0.05
Ki=170

Figure(4-9) shows the closed and open loop system step responses generated using SIMULINK where the plant was the system shown in Figure(4-6). Note that the controller gains (i.e. PID parameters) are found by using the simplified model given in Figure(4-7), while the simulations are performed by using the model given in Figure(4-6). The results are found to have an overshoot of 7.73 % with 9.5 msec rise time and 59 msec 2% settling time.

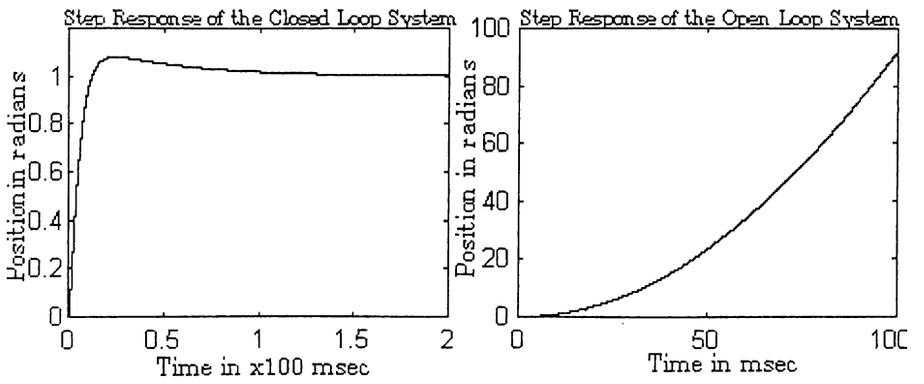


Figure 4-9: Step Responses of the Open and Closed Loop System

4.4 Implementation of PID controller on DSP

The algorithm can be implemented in a straightforward way in a DSP with floating point hardware. Implementation using an ordinary DSP does, however, require special considerations, because all calculations have to be made in integer arithmetic.

4.4.1 Modification and Discretization

The PID algorithm has several drawbacks. Significant modifications of linear and nonlinear behavior are necessary in order to obtain a practically useful algorithm. To obtain equations that can be implemented using computer control it is also necessary to replace continuous time operations like derivation and integration by discrete time operations. The PID algorithm consists of Proportional (P(t)), Integral (I(t)) and Derivative (D(t)) terms which are summed to get the control signal v(t) as:

$$v(t) = P(t) + I(t) + D(t) \tag{4.14}$$

where

$$P(t) = K_p \cdot e(t) \tag{4.15}$$

$$I(t) = \frac{K_p}{T_i} \cdot \int e(t) \tag{4.16}$$

$$D(t) = K_p \cdot T_d \cdot \frac{de(t)}{dt} \tag{4.17}$$

with the error signal as $e(t) = r(t) - y(t)$ and $r(t)$ as reference.

Proportional Term

The proportional term $P(t)$ is implemented simply by replacing the continuous time variables with their sampled equivalences. The proportional term then becomes

$$P(t_k) = K_p(r(t_k) - y(t_k)) \tag{4.18}$$

where $\{t_k\}$ denotes the sampling instants.

Integral Term

To obtain an algorithm that can be implemented on a computer, the integral term I(t) is differentiated

$$\frac{dI(t)}{dt} = \frac{K_p}{T_i} \cdot e(t) \tag{4.19}$$

Approximating the derivative by a forward difference gives

$$\frac{I(t_{k+1}) - I(t_k)}{h} = \frac{K_p}{T_i} \cdot e(t_k) \tag{4.20}$$

where h is the sampling period. Finally, by rearranging terms, we get the following equation to compute the integral term

$$I(t_{k+1}) = I(t_k) + \frac{K_p}{T_i} \cdot h \cdot e(t_k) \tag{4.21}$$

Derivative Term

A pure derivative should not be implemented, because the controller gain becomes very large at high frequency. This leads to amplification of high frequency noise. The derivative term is therefore approximated in s-domain by

$$\frac{D(s)}{E(s)} = s \cdot K_p \cdot T_d \approx \frac{s \cdot K_p \cdot T_d}{1 + s \cdot T_d/N} \tag{4.22}$$

Parameter N is therefore called maximum derivative gain. In analog controllers N is given a fixed value, from Reference [4] it is typically in the range of 5-20. Notice that the approximation is good for signals whose frequency contents are significantly below N/T_d. Also notice that the approximating transfer function has a maximum gain of N at high frequencies.

Note that e(t)=r(t)-y(t), and r(t) can be considered constant. Hence, for the derivative term, we may replace e(t) by -y(t). It is also advantageous not to let the derivative act on the set point signal r(t). The set point is constant for most of the time and its derivative is therefore zero. A step change in the set point may, however, cause an undesirable jump in the control variable if the derivative acts on the set point. With these modifications 4.22 can be rewritten as

$$D(s) = \frac{-s \cdot K_p \cdot T_d \cdot Y(s)}{1 + s \cdot T_d/N} \tag{4.23}$$

which is in time domain

$$D(t) + \frac{T_d}{N}\frac{dD(t)}{dt} = -K_p \cdot T_d \cdot \frac{dy}{dt} \tag{4.24}$$

There are several methods to approximate the derivative. Common methods are the forward difference approximation, the backward difference approximation, Tustin's approximation and ramp equivalence. These approximations all have the same form

$$D(t_k) = a \cdot D(t_{k-1}) - b \cdot (y(t_k) - y(t_{k-1})) \tag{4.25}$$

and are stable only if $|a| < 1$. The backward difference approximation gives good results for all values of T_d. The parameter a goes to zero as T_d goes to zero. Here the backward difference approximation is chosen.

The following is obtained when Equation(4.24) is approximated by a backward difference:

$$D(t_k) + \frac{T_d}{N} \cdot \frac{D(t_k) - D(t_{k-1})}{h} = -K_p T_d \cdot \frac{y(t_k) - y(t_{k-1})}{h} \tag{4.26}$$

Rearranging terms, gives (4.25) with

$$a = \frac{T_d}{T_d + N \cdot h} \tag{4.27}$$

$$b = \frac{K_p \cdot T_d \cdot N}{T_d + N \cdot h} \tag{4.28}$$

which is the formula that will be used to compute the derivative term.

The PID Algorithm

Summarizing we find that a practical version of the PID algorithm can be described by the following equations:

$$P(t_k) = K_p(r(t_k) - y(t_k))$$

$$D(t_k) = a_d \cdot D(t_{k-1}) + b_d(y(t_{k-1}) - y(t_k))$$
$$v(t_k) = P(t_k) + I(t_k) + D(t_k)$$
$$u(t_k) = f(v(t_k))$$
$$I(t_{k+1}) = I(t_k) + b_i(r(t_k) - y(t_k))$$

The function f describes the nonlinear characteristic of the actuator.

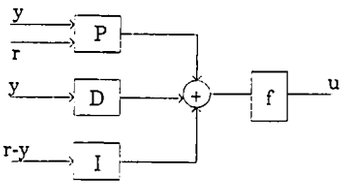


Figure 4-10: Structure of the PID Controller

The parameters a_d, b_d, b_i are related to the primary parameters K_p, T_i, T_d and N for the PID controller as follows:

$$a_d = \frac{T_d}{T_d + N \cdot h} \tag{4.29}$$

$$b_d = \frac{K_p N T_d}{T_d + N \cdot h} \tag{4.30}$$

$$bi = K_p h / T_i \tag{4.31}$$

Since the above equations have to be updated only when the controller parameters are changed, they can be computed off line. Notice that the algorithm is in parallel form.

4.4.2 Implementation Issues

Implementation of a PID-controller using a DSP using fixed point arithmetic will now be discussed. To perform fixed point calculations it is necessary to know orders of magnitude of all variables. Because of the parallel form, the P, I and D terms can be scaled and computed separately and then unified.

Selection of Sampling period

There are several rules of thumb for choosing the sampling period for digital controllers. The sampling period should be short enough so that the pole $s = -N/T_d$, introduced to limit the high frequency gain of the derivative, can be approximated appropriately. This leads to the following rule of thumb (see [5]):

$$\frac{h \cdot N}{T_d} \approx 0.1 - 0.3 \tag{4.32}$$

Coefficient Scaling

Because of the wide number range of the parameters, some restrictions must be imposed on the magnitude of coefficients. From simulations in MATLAB it follows that b_d is the largest parameter. A limit should therefore be set on the gain K_p, and the high frequency derivative gain N. If K_p and N are limited to 16, we have $b_d < K_pN=256$ and $K_p \leq 16$. These parameters must therefore be divided by 256 and 16 respectively before they are stored. To restore the magnitude of the signal, the derivative term must be shifted to the left by 8 bits and the proportional term shifted to the left by 4 bits.

The other parameters, a_d and b_i are within the number range and do not need scaling.

Signal Scaling and Saturation Arithmetic

It must be insured that overflow does not occur when computing the states of the controller .With the structure of the PID controller the states are $D(t_k)$ and $I(t_{k+1})$. Care must be also taken so that overflow does not occur when the P, I and D terms are added to obtain v.

The proportional term will always be within the number range, since the multiplication of a fraction with a fraction gives a fraction. Overflow can occur if K_p is larger than 1 when the magnitude of the signal is restored. It is therefore necessary to use saturation arithmetic when computing the proportional term.

Since the derivative depends only on the process output, it is difficult to use analytic scaling methods effectively. A good engineering approach is therefore to simulate the closed loop system and store the output of the derivative for a few representative examples. Since b_d can take large values, saturation arithmetic should be used before storing the derivative. A number of simulations were made in order to obtain typical orders of magnitude of the proportional. integral and derivative term. It turns out, that under normal operation conditions, the variables are within the number range. Since we are allowing a gain larger than one, it is very likely that an overflow will occur under some operating condition, for example during start-up. Saturation arithmetic is therefore used on both states and on the control signal v.

Chapter 5

HARDWARE DESIGN

A block diagram of the system is shown in Figure(5-1). In the following we discuss each major section of the hardware design.

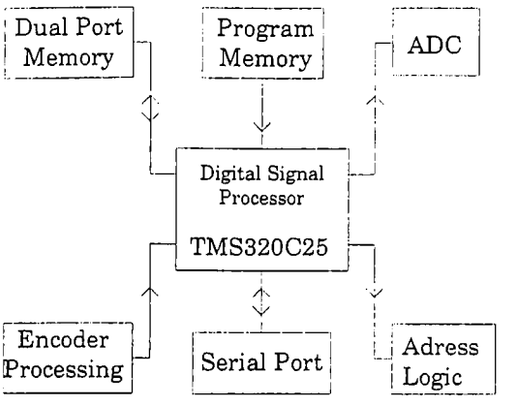


Figure 5-1: System Block Diagram

5.1 Digital Signal Processor

The DSP chip is the central processing unit for the system. The DSP used is the 40 MHz version of the TMS320C25 by Texas Instruments. A detailed

Output Port	Function	Definition
0	not used	not used
1	DAC data	A-PAL
2	not used	not used
3	not used	not used
4	not used	not used
5	Transmit Reg. of ACIA	H-PAL
6	Control Reg. of ACIA	H-PAL
7	Enable Amplifier	H-PAL
8-15	not used	not used

Table 5.1: Output Port Map

discussion of the architecture and programming of the DSP is given in [2]. The DSP is clocked at exactly 40 MHz and has an instruction cycle time of 100 ns. It has a 16-bit word size with a 32-bit ALU and accumulator. It uses a 16 by 16-bit parallel multiplier with a 32-bit result to perform multiplications in one instruction cycle. It also provides an internal 16-bit timer.

A 40 MHz clock oscillator with a CMOS level output is used to drive the external clock input of the DSP. To cause a reset on power-up, a power monitor chip from Dallas Semiconductor is used. The DS1231 Power Monitor Chip uses a precise temperature-compensated reference circuit which provides an orderly shutdown and an automatic restart of a processor-based system. On power-up, the chip asserts reset in approximately 500 ms, allowing the clock oscillator to stabilize and the DSP to reset.

The DSP has three interrupt inputs which support either level-triggered or edge-triggered interrupts. The DSP has an I/O space of 16 words. These are divided into 16 input and 16 output ports. Input and output port maps are shown in Table(5.1) and Table(5.2)

Input Port	Function	Definition
0	not used	not used
1	Read Encoder	A-PAL
2	not used	not used
3	not used	not used
4	not used	not used
5	Receive Reg. of ACIA	H-PAL
6	Status Reg. of ACIA	H-PAL
7	Status of Amplifier	H-PAL
8-15	not used	not used

Table 5.2: Input Port Map

5.2 Memory

The DSP has two independent memory spaces, a program space and a data space, each addressing up to 64 K-words of memory, where the word size is 16 bits. The memory hardware provides two external memory units, a 8 K-word EPROM and 1 K-byte of static Dual Port Ram. A block diagram of the memory hardware is shown in Figure(5-2).

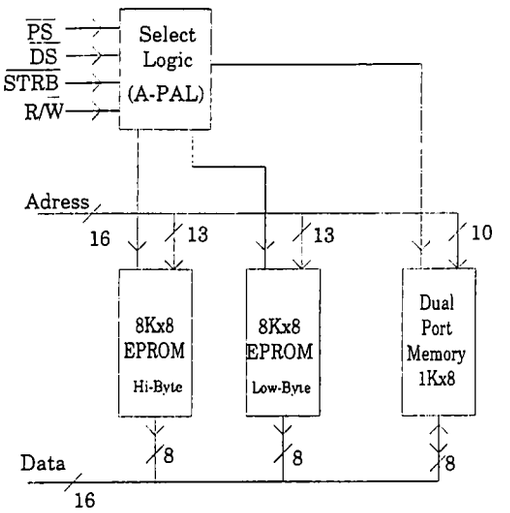


Figure 5-2: Memory Block Diagram

The following buses, port, and control signals provide system interface to the

TMS320C25 processor:

- 16-bit address bus (A15-A0)
- 16-bit data bus (D15 - D0)
- $\overline{PS}, \overline{DS}, \overline{IS}$ (program, data, I/O space select)
- R/$\overline{W}$ (read/write) and $\overline{STRB}$ (strobe)

Throughout this chapter, Q is used to indicate the duration of a quarter phase of the output clock (CLKOUT1 or CLKOUT2). Since speed and maximum throughput is desired, the TMS320C25 is run with no wait states. The TMS320C25 expects data to be valid no later than 2Q-23 ns after $\overline{STRB}$ goes low. (This takes 27 ns for a TMS320C25 operating at 40 MHz, Q=25ns). The access times of the CY7C261-35 from Cypress Semiconductor EPROMs are 35 ns maximum from address and 20 ns maximum from chip enable. On the TMS320C25, address becomes valid a minimum of Q-12 ns=13 ns before $\overline{STRB}$ goes low. Therefore, the data appears on the data bus within 27 ns after $\overline{STRB}$ goes low, as required by the TMS320C25.

The Dual-Port RAM interface is the IDT7130SA35P from Integrated Device Technology. The IDT7130SA39P is high speed 1Kx8-bit Dual-Port Static RAM. The RAM has a 35-ns access time from address and a 25-ns access time from chip enable. Note that these access times are fast enough so that a wait-state generator is not required for this interface.

The DSP also has three blocks of on-chip memory (B0, B1, B2) with a total of 544 words. B1 and B2 are always mapped into the data space. B0, which is 256 words in length, can be software configured into either data space or program space. B0 is mapped into data space upon power-up. The DSP also contains memory mapped registers and reserved locations which are mapped into data space. The memory maps given in Table(5.3) show program and data memory locations.

Type	Address	Description
Program	0000h-001Fh	EPROM, Interrupt Vectors
	0020h-2000h	EPROM
Data	0000h-0005h	Internal, Registers
	0006h-005Fh	Internal, Reserved
	0060h-007Fh	Internal, Block B2
	0080h-01FFh	Internal, Reserved
	0200h-02FFh	Internal, Block B0
	0300h-03FFh	Internal, Block B1
	0400h-0800h	External, Dual Port RAM

Table 5.3: Memory Map: B0 as Data Memory

The Address Logic is implemented by using Programmable Array Logic (PAL). For this purpose Cypress Semiconductor PAL22V10D-15PC which has a maximum pin to pin delay time of 15 ns is used. The PAL22V10 is a second-generation programmable logic array device. It is implemented with the familiar sum-of-products (AND-OR) logic and a programmable macrocell, which is a new product. The programmable macrocell provides the capability of defining the architecture of each output individually. Each of the 10 potential outputs may be specified as registered or combinational. Polarity of each output may also be individually selected, allowing complete flexibility of output configuration. The logic equations are written in VHDL (**V**ery-high-speed-IC **H**ardware **D**escription **L**anguage).

The A-PAL is also responsible for the address decoding of E-PAL (Encoder Processing) and D/A converter which will be described in the following sections.

5.3 Digital to Analog Conversion

The Digital to Analog converter provides the necessary analog interface to the Brushless Power Amplifier. Functional Block Diagram of the DAC section is shown in Figure(5-3).

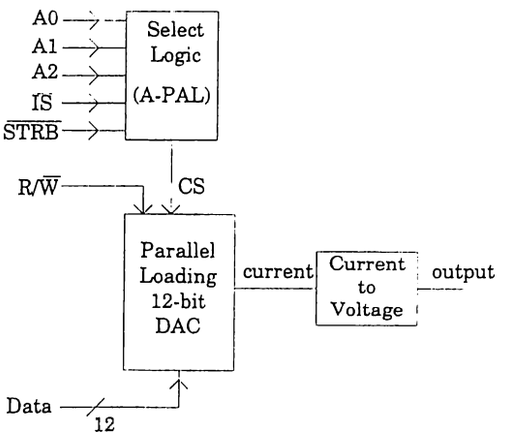


Figure 5-3: DAC Block Diagram

The Digital to Analog interface is implemented using the AD7547 from Analog Devices. The AD7547 is 12-bit current output DAC with level shifters, data registers and control logic for easy microprocessor interfacing. Two control signals $\overline{CS}, \overline{WR}$ control DAC selection and loading. Data is latched into the DAC registers on the rising edge of $\overline{WR}$. The device is speed compatible with TMS320C25 and accepts TTL level inputs.

The primary requirement for the current-output DAC is an amplifier with low input bias currents and low input offset voltage. The current to voltage conversion is made around the LF444CN from National Semiconductor. The LF444 is a low power, JFET input operational amplifier with high gain bandwidth. It has low input bias current of 50pA(max). The DAC is configured in bipolar mode and has the code table shown in Figure(5-4).

Binary Number in DAC Register MSB LSB	Analog Output
1111 1111 1111	$+Vin(\frac{2047}{2048})$
1000 0000 0001	$+Vin(\frac{1}{2048})$
1000 0000 0000	0V
0111 1111 1111	$-Vin(\frac{1}{2048})$
0000 0000 0000	$-Vin(\frac{2048}{2048})$

Vin = 10 V

Figure 5-4: Bipolar Code Table

5.4 Encoder Processing

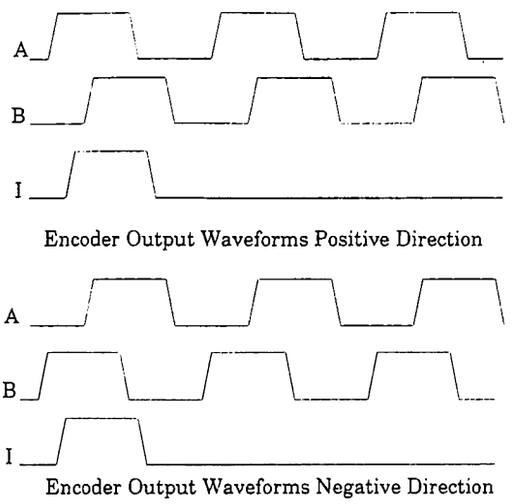


Figure 5-5: Encoder Output Waveforms

The position feedback hardware was designed to decode the three line quadrature output of a position encoder and maintain position information which the

DSP can read at any time. The three line quadrature output consists of two signal lines, A and B, which have pulse-trains 90 degrees out of phase, and an index line I. An illustration of these signals is shown in Figure(5-5). The A and B lines contain a fixed number of pulses per revolution, the number depending upon the encoder. When the direction of rotation is reversed, the phase relationship between the A and B lines reverses. The index line I gives a pulse once per revolution when the angular position is at its starting point.

A block diagram of the position feedback hardware is shown in Figure(5-6) and its schematic is shown Appendix. The A,B and I inputs are passed through comparators to obtain TTL level signals.

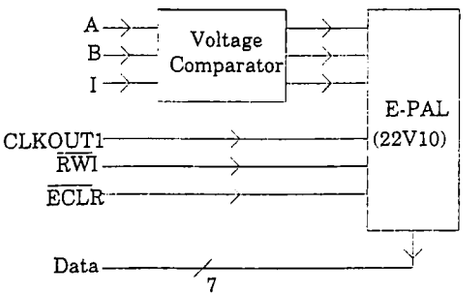


Figure 5-6: Encoder Processing Block Diagram

The comparator used is the LTC1045 from Linear Technology. It consists of six high speed comparators with output latches and three-state capability. It can efficiently translate voltage levels with 1.2 μs response time. A 22V10 PAL is used to determine direction and to count the pulses send by the encoder. The Encoder Processing PAL (E-PAL) implements the state diagram shown in Figure(5-7). The diagram is used to determine the direction of rotation and the necessary count pulses needed for the counter.

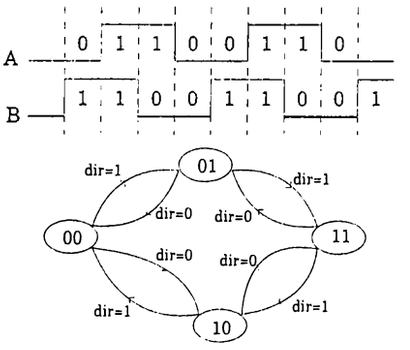


Figure 5-7: State Diagram of E-PAL

At every change of state the 5 bit counter in E-PAL is incremented or decremented depending on the direction signal. The encoder supplies 1024 pulses per revolution and since there are 4 states at each period the counter is clocked 4 times faster. As a result, the resolution of position information is increased by 4 to 4096 counts per revolution. The E-PAL has an enable signal $\overline{RWI}$ which is asserted by an input instruction to port 1 by A-PAL. The CLKOUT1 signal is the master clock output signal (CLKIN frequency/4) of the TMS320C25. Additionally there is a counter clear signal $\overline{ECLR}$ which is asserted by H-PAL after a read operation of the DSP. The code of the E-PAL is written in VHDL language.

5.5 Serial Port

The serial port provides full duplex serial communication at baud rates up to 1.0 Mbps. Data words can be 7 or 8 bits in width. Even, odd or no parity can be used. A stop bit length of 1 or 2 bits can be used. The serial port supports both interrupt-driven and polled operations. A block diagram of the serial port hardware is shown in Figure(5-8).

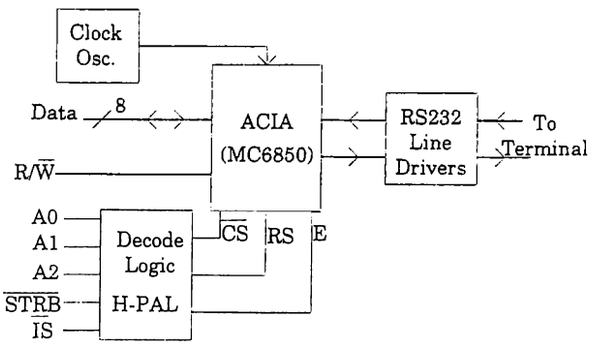


Figure 5-8: Serial Port Block Diagram

The serial port consists of three main parts, the Asynchronous Communication Interface Adapter (ACIA) chip, the logic to interface the ACIA to DSP, and the RS-232 line drivers. The ACIA chip used is the MC68B50P from Motorola Semiconductors. The MC6850 Asynchronous Communications Interface Adapter provides the data formatting and control to interface serial asynchronous data communications information to bus organized systems. The bus interface of the MC6850 includes select, enable, read/write, interrupt and bus interface logic to allow data transfer over an 8-bit bidirectional data bus. The parallel data of the bus system is serially transmitted and received by the asynchronous data interface, with proper formatting and error checking. The functional configuration of the ACIA is programmed via the data bus during system initialization. A programmable Control Register provides variable word lenghts, clock division ratios, transmit control, receive control, and interrupt control.

At the bus interface, the ACIA appears as two addressable I/O locations. Internally, there are four registers: two read-only and two write-only registers. The read-only registers are Status and Receive Data; the write-only registers are Control and Transmit Data. The serial interface consists of serial input and output lines with independent clocks. In this work the clocks are the same and

can be calculated using the following formula:

$$TxClock = RxClock = divisionratio \cdot baudrate \tag{5.1}$$

The divisionratio is programmed using the Control Register of ACIA. The serial interface must be converted between logic levels and RS-232 output level voltages. The integrated circuit used for this purpose is the TC232CPE from Teledyne Semiconductors. This device has an on-chip charge pump voltage converter which generates the necessary RS-232 output levels from the 5 V supply. Using this chip, the RXD signal from the external interface is converted to logic level, and the TXD signal from the ACIA is converted to RS-232 output levels.

The ACIA interface to the DSP is implemented by using H-PAL. It has the necessary logic to address the ACIA by I/O operations from the DSP. H-PAL is a 22V10 device and its code is written using VHDL language.

5.6 Dual Port Memory

The Dual Port Memory is built around the IDT7130SA35P. The ID7130 provides two ports with separate control, address, and I/O pins that permit independent access for reads or writes to any locations in memory. When a port is enabled, access to the entire memory array is permitted. Each port has its own Output Enable control. A block diagram of the Dual Port Memory is shown in Figure(5-9).

The Dual Port Memory is useful in multi-processor system applications, when fast and reliable communication is needed. The position commands are written to the Dual Port Memory locations and an interrupt is generated for the DSP.

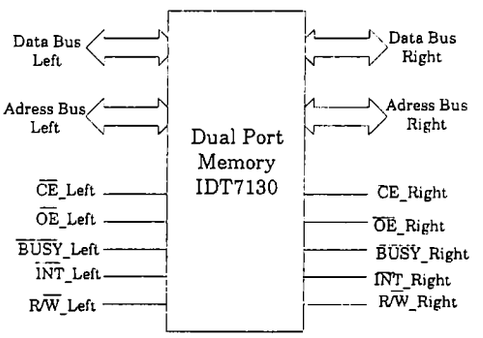


Figure 5-9: Dual Port Memory Block Diagram

5.7 Prototype

Once the hardware design and schematics were complete, a prototype of the system was implemented. Parts were ordered and wire-wrap board layout was created. Test software was written to exercise all the features of the prototype, and each section of the hardware was tested and verified.

Chapter 6

Software Design

The software design of the Control system can be divided into two parts. The first part deals with the development of the Assembler code for the DSP. The second part is the development of a windows based visual application for the verification and performance testing of the system. These two parts will be discussed in the next sections.

6.1 Assembly Code

The program structure is straightforward: An interrupt service routine (ISR) processes the PID control calculations, and a foreground loop is used to implement the user interface and serial communication. The ISR has a simple structure. In order to effect position control we need to read the encoder, calculate the new PID values, and set the output of DAC, all at a constant, predefined rate. The ISR is initiated by a hardware timer (TINT) on the TMS320C25. The TMS320C25 16-bit on-chip timer and its associated interrupt perform various functions at regular time intervals. On the TMS320C25, the timer is a down counter that is continuously clocked by CLKOUT1 and counts (PRD+1) cycles of CLKOUT1. By programming the period (PRD) register from 1 to 65535 (0FFFFh), a timer

interrupt (TINT) can be generated every 2 to 65536 cycles. For the control algorithm the period register effectively determines the sample rate of the digital control system. Figure (6-1) shows the timing of the Software functions.

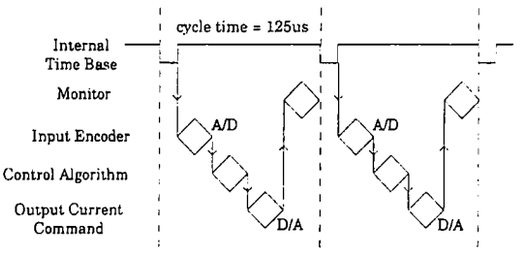


Figure 6-1: Timing of Software Functions

6.1.1 The Monitor Program

The primary function of the Monitor Program is to make an interface for the user. With the help of the Monitor program the user can modify and read memory locations, registers and send special commands for the control of the peripherals. The Monitor program relies on the ACIA interface and communicates with the PC via RS232 Standard. Figure(6-2) illustrates the main flow diagram of the program.

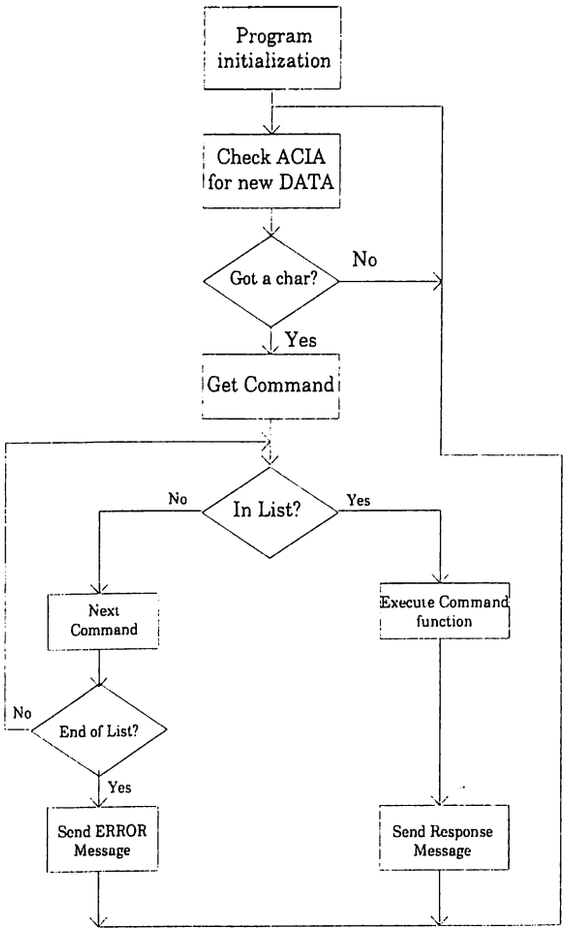


Figure 6-2: Flowchart for Monitor Program

The following commands are implemented and recognized by the user interface in the foreground loop.

Display (argument): **d**

where argument is one of the following

s0 : display status register 0 contents

s1 : display status register 1 contents

a0 : display auxiliary register 0 contents

a1 : display auxiliary register 1 contents

a2 : display auxiliary register 2 contents

a3 : display auxiliary register 3 contents

a4 : display auxiliary register 4 contents

a5 : display auxiliary register 5 contents

a6 : display auxiliary register 6 contents

a7 : display auxiliary register 7 contents

m : display memory (ROM)

mc : display memory continious

r : display RAM

rc : display RAM continious

rs : display RAM stored

The display command is used to display the contents of auxiliary registers, ROM and RAM memory. It is useful for real-time debugging and for monitoring of various variables. The special command **d rs** is used for performance evaluation purposes when the output variable of the control has to be monitored.

Modify (argument): **m**

where argument is one of the following

s0 : modify status register 0 contents

s1 : modify status register 1 contents

a0 : modify auxiliary register 0 contents

a1 : modify auxiliary register 1 contents

a2 : modify auxiliary register 2 contents

a3 : modify auxiliary register 3 contents

a4 : modify auxiliary register 4 contents

a5 : modify auxiliary register 5 contents

a6 : modify auxiliary register 6 contents

a7 : modify auxiliary register 7 contents

r : modify RAM

The modify command is useful for changing register and RAM locations during the program flow.

Enable Amplifier: **e1**

Enable amplifier causes the Brushless DC Motor Amplifier to be enabled.

Disable Amplifier: **e0**

Disables the Brushless DC Motor Amplifier.

Open Control: **o**

Enables the Timer Interrupt and the Control Algorithm starts.

Close Control: **c**

Disables the Timer Interrupt and the Control Algorithm stops.

Start Store: **s**

Special command used in conjunction with the **d rs**. Main purpose of this command is to obtain test results (e.g. step response) of the control system. At every start of the Control Algorithm the position information is saved in memory and can be displayed using **d rs** command when enabled.

To be able to have a structured program design, five general purpose routines are written.

ascii4hex: This routine is used to convert the ASCII characters received from the terminal to equivalent hexadecimal values. It converts the ASCII data contained in data memory addresses ASCHEX and ASCHEX1 to hexadecimal and saves the result again in data memory address ASCHEX

input4ascii: This routine inputs four characters and saves them in data memory addresses INPUT and INPUT1

hex4ascii: This routine converts the data contained in data memory address location HEXASC to ASCII and saves the result in HEXASC and HEXASC1

sendtext: This routine sends ASCII text contained in ROM specified by AR6 starting address and continues to send the characters until an EOT (end of text) marker is encountered.

sendram: This routine sends ASCII text contained in RAM specified by AR6 starting address and continues to send the characters until an EOT marker is encountered.

6.1.2 The Control Algorithm

To implement the algorithm described in Section 4.4 following organization is introduced:

SAVE CONTEXT

load y(n-1) and ysp

PID

input y(n)
compute derivative (D)
compute proportional part (P)
add D, P and I
round off, check for overflow and store in v(n)
compute u(n) from saturation function
output u(n)
compute I

RESTORE CONTEXT

DEVICE	cycles	KHz
TMS320C25	89	112

Table 6.1: Cycle count and maximum sampling frequency for PID controller

The control algorithm is the Interrupt Service Routine (ISR) for the Timer Interrupt. Unless the interrupt service routines are simple I/O handlers, the processing in each ISR generally assure that the processor context is preserved during execution. The context must be saved before the routine executes and must be restored when the routine is finished. Then the states of the controller are cleared, the set point value (ysp) and the process output (y(n-1)) are read from data memory. The program then computes the control signal.

The algorithm consists of additions and multiplications which the DSP is specialized for. Because of the wide number range of the parameters, some restrictions must be imposed on the magnitude of coefficients which is done by scaling.

By using the timer on the TMS320C25 it was possible to count the cycles required for one execution of the PID loop. Table (6.1) shows the number of cycles for the controller and the maximum sampling frequency which can be used.

6.2 Visual Basic User Interface

The visual basic user interface provides a means for easy data exchange between the PC and the DSP Controller Board. It uses the basic building blocks of Windows to accomplish complex tasks such as plotting of data. The user interface can be treated as an extension to the monitor program described in the previous section. It provides fast and reliable communication via the RS232 Serial port. The user can read and modify P, I and D constants give step input commands, enable and disable amplifier and controller, and plot the position step response

in real time.

Once the PID loop is successfully implemented, the next challenge is to tune it. This was made through the extensive use of the visual basic user interface. By altering P, I and D, and plotting the results, the system can be tuned finely.

Chapter 7

Experimental Verification

To verify the performance of the system in a position control situation, the prototype was used to control a 3-phase Brushless DC Motor from Kollmorgen Motion Technologies Group. The motor has an incremental position encoder attached to its shaft which can generate 1024 pulses at every rotation. A signal generator is used to supply the necessary clock signal for the ACIA and 24 Volt is applied to the amplifier. The serial connection between the PC and DSP controller had the parameters of 9600 baud, 8-bit with no parity.

Step responses of the open and closed loop systems are obtained for different step inputs using the user interface program. The program gave the ability to plot step responses, to change the control parameters and step input value during operation of the controller. Figure (7-1) shows the open loop step response of the system. Figure (7-2) and Figure (7-5) show both the experimental and simulation results of the step response of the closed loop system for a step input of 50 degrees and 30 degrees, respectively. As can be seen from the Figure (7-2) the settling time (%2) is about 55 msec, the final position error is 0.17 degrees, overshoot is about 7.4% and the rise time is about 10 msec. The position error for the step input of 50 degrees is given in Figure (7-4). The error plot between the experimental and MATLAB step responses for step input of 50 degrees is

illustated in Figure (7-3).

To observe the behaviour of the system for successive step inputs, which will be the case in the application, a special assembly routine is written. When enabled, the routine generates three different step inputs 50 miliseconds apart. The resulting plot is shown in Figure(7-6).

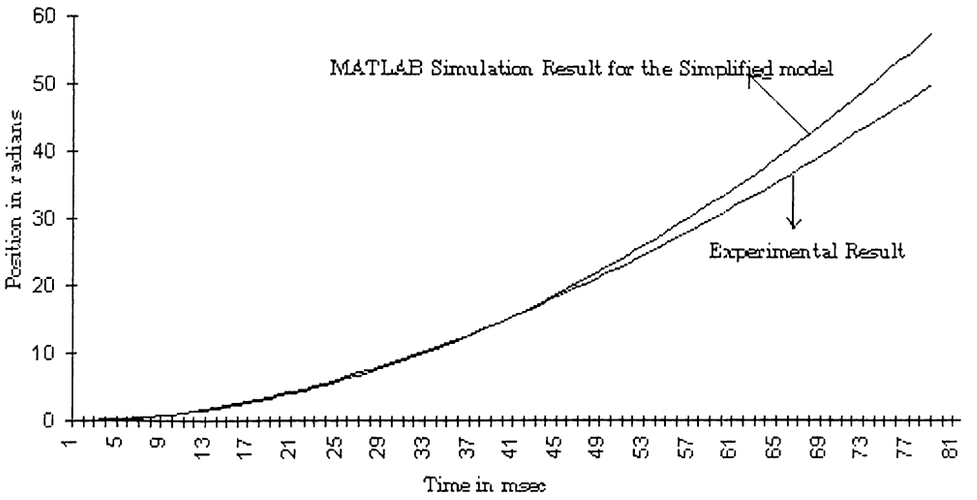


Figure 7-1: Open Loop Step Response of the System (Motor and Amplifier)

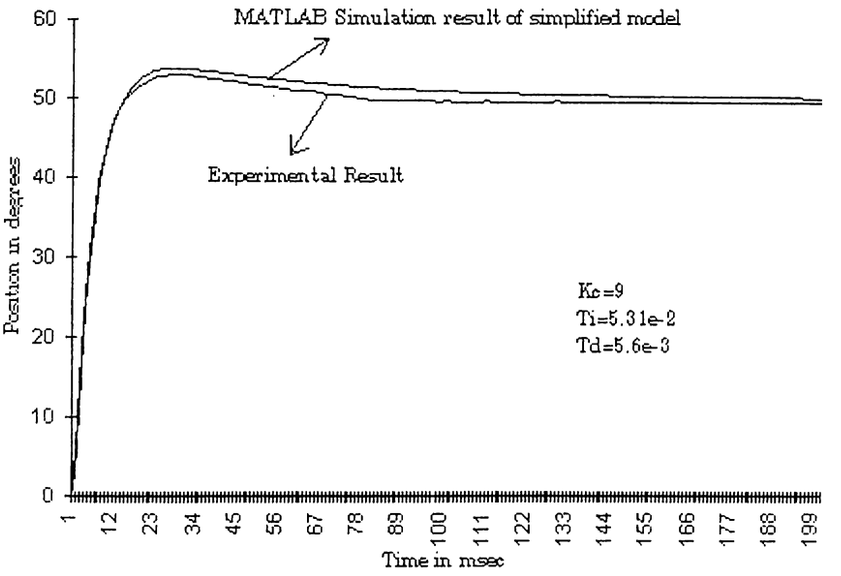


Figure 7-2: Step Response of the Closed Loop System for Step Input of 50 degrees

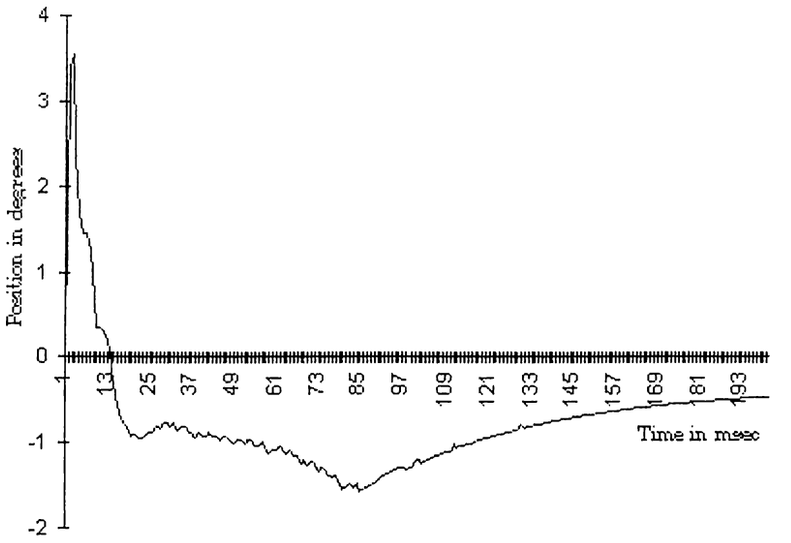


Figure 7-3: Error Plot between the Experimental and MATLAB Step Responses for Step input of 50 degrees

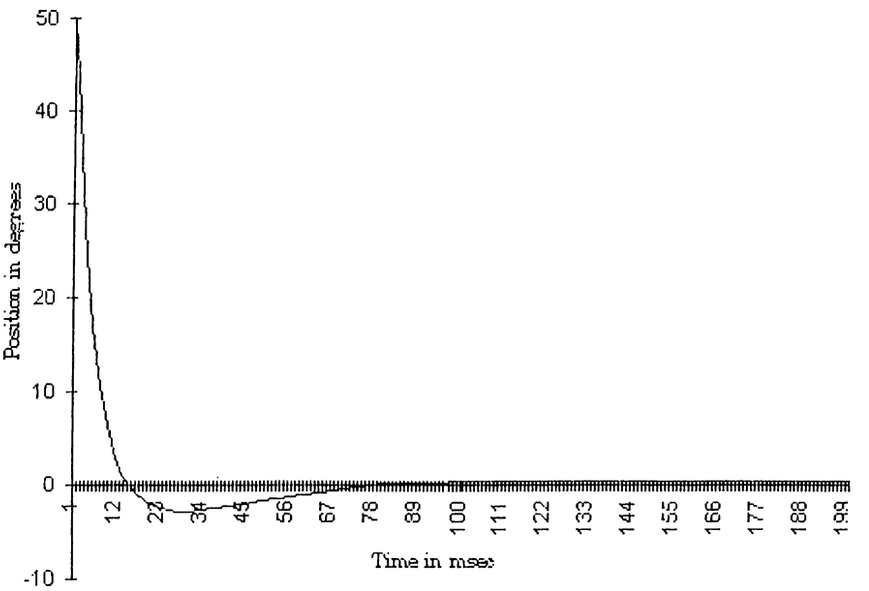


Figure 7-4: Position Error for the Experiment shown in Figure 7.2

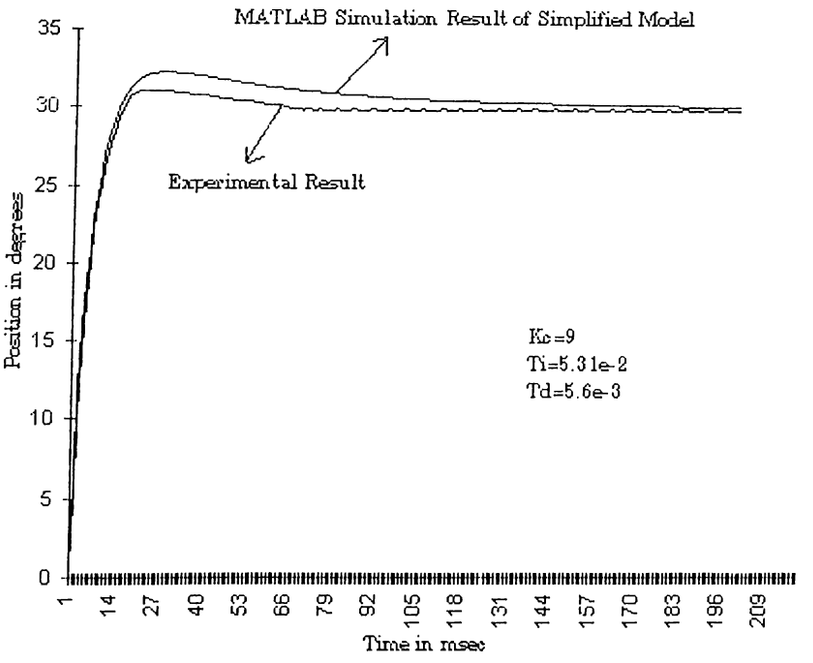


Figure 7-5: Step Response of the Closed Loop System for Step Input of 30 degrees

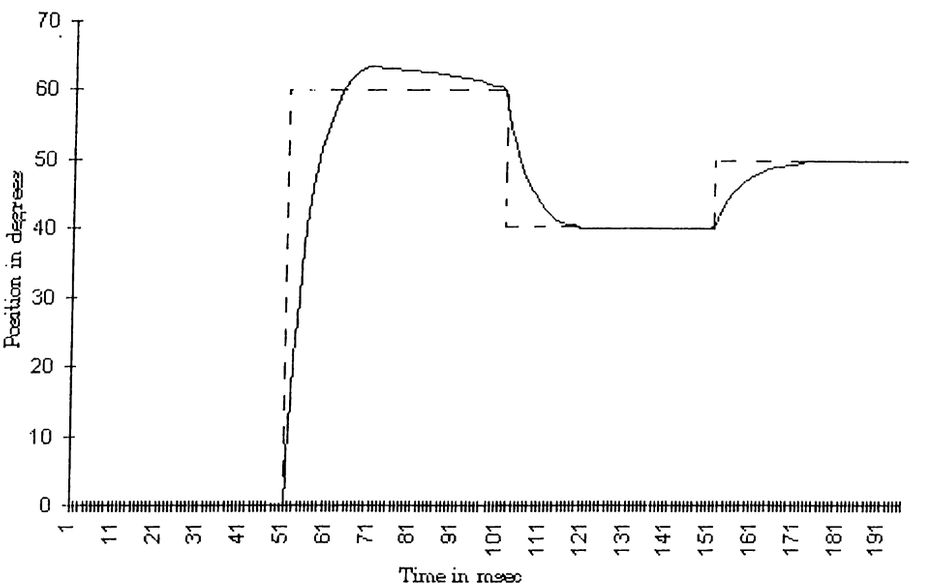


Figure 7-6: Experiment showing the position response of the system for successive step inputs

Chapter 8

Conclusion

The following are the conclusions of this work.

1. Based on the controller requirements of a high performance Brushless DC Motor controller, a DSP based system was designed, tested and experimentally verified.

2. The Encoder Processing hardware provides unique hardware solutions to input/output problems usually performed by software, which saves significant amount of processing time.

3. The system provides enough processing power and memory space for the execution of complicated modern control algorithms and gives the ability to control motor drives at high rotational velocities.

4. A detailed model for the Brushless DC Motor and its Amplifier is developed by using SPICE and is verified through experiments. A PID controller is designed for the simplified model and is tested in the detailed model.

5. The specifications for the system were met or exceeded. Positioning is done by an error of about 0.1^{o} with an overshoot of less than 10%.

6. An expandable monitor program is developed in assembly code.

7. A Windows based user interface program is developed as an extension of the monitor program to enable fast and easy performance testing of the system.

8. The system provides flexibility and easy interface to any of the major types of motor amplifiers with very little or no modifications.

The following recommendations are made for future study.

1. Implementation of the control algorithms in C language, and implementation of the necessary software to interface with the system.

2. A compact packaging of the DSP system for noise-free operation.

3. Expanding the system to control 4 Brushless DC Motors.

Bibliography

[1] C.K. Taft, R.G. Gauthier, T.J. Harned (1995), "Brushless Motor System Design and Analysis", University of New Hampshire

[2] Texas Instruments (1993), "TMS320C2X User's Guide"

[3] Electro-Craft (1995), "DC Motors Speed Controls Servo Systems"

[4] K.J. Astrom, H.Steingrimsson, "Implementation of a PID Controller on a DSP", Research Triangle Park, NC

[5] K.J. Astrom, T.Hagglund 1988, "Automatic Tuning of PID Controllers", Research Triangle Park, NC

[6] Kollmorgen Inland Motor (1996), "Handbook for Brushless Motors and Drive Systems",VA

[7] Kollmorgen Inland Motor 1995, "Brushless DC Motors and Amplifiers"

[8] Texas Instruments (1993), "Digital Signal Processing Applications with the TMS320 Family - Theory, Algorithms, and Implementations", Digital Signal Processing

[9] Texas Instruments (1993), "TMS320 Fixed-Point DSP Assembly Language Tools"

[10] Texas Instruments (1993), "Digital Control Applications with the TMS320 Family"

[11] B.C.Kuo (1991), "Automatic Control Systems", Prentice-Hall