

"HIGH PRECISION CNC MOTION CONTROL"

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

GÖKÇE MEHMET AY

IN PARTIAL FULFILLMENT FOR THE REQUIREMENTS OF
THE DEGREE OF MASTER OF SCIENCE

IN

MECHANICAL ENGINEERING

AUGUST 2004

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Kemal İDER
Head of Department

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

Asst. Prof. Dr. Melik DÖLEN
Supervisor

Examining Committee Members

Prof. Dr. Y. Samim ÜNLÜSOY (METU ME) _____

Asst. Prof. Dr. Melik DÖLEN (METU ME) _____

Prof. Dr. M. Kemal ÖZGÖREN (METU ME) _____

Asst. Prof. Dr. A. Buğra KOKU (METU ME) _____

Asst. Prof. Dr. Ahmet M. HAVA (METU EE) _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name :

Signature :

ABSTRACT

HIGH PRECISION CNC MOTION CONTROL

AY, Gökçe Mehmet

M.S., Department of Mechanical Engineering

Supervisor : Asst. Prof. Dr. Melik DÖLEN

August 2004, 181 pages

This thesis focuses on the design of an electrical drive system for the purpose of high precision motion control. A modern electrical drive is usually equipped with a current regulated voltage source along with powerful motion controller system utilizing one or more micro-controllers and/or digital signal processors (DSPs). That is, the motor drive control is mostly performed by a dedicated digital-motion controller system.

Such a motor drive mostly interfaces with its host processor via various serial communication protocols such as Profibus, CAN+, RS-485 etc. for the purpose of receiving commands and sending out important status/control signals. Considering that the motor drives lie at the heart of every (multi-axis) motion control system, the aim of this thesis is to explore the design and implementation of a conventional DC motor drive system suitable for most industrial applications that require precision and accuracy. To achieve this goal, various underlying control concepts and important implementation details are rigorously investigated in this study.

A low power DC motor drive system with a power module, a current regulator and a motion controller is built and tested. Several design revisions on these subsystems are made so as to improve the overall performance of the drive system itself. Consequently, important “know-how” required for building high performance (and high power) DC motor drives is gained in this research.

Keywords: DC motor motion control, DC voltage regulator, current regulator, DC motor control, DC motor drives.

ÖZ

YÜKSEK HASASİYETLİ CNC HAREKET KONTROLÜ

Ay, Gökçe Mehmet

Yüksek Lisans, Makine Mühendisliği Bölümü

Tez Yöneticisi : Yrd. Doç. Dr. Melik Dölen

Ağustos 2004, 181 sayfa

Bu tezde yüksek doğruluklu CNC hareket kontrolü için bir elektrik sürücü sistemi tasarlanması amaçlanmıştır. Modern elektrik motor sürücülerinin üzerinde yapılan araştırmalar; motor sürücülerinin akım kontrollü voltaj kaynağı ve bir veya birden fazla mikroişlemci/mikrodenetleyici ya da sayısal sinyal işlemciler (DSPs) sahip güçlü hareket denetleyicileriyle donanmış olduklarını göstermiştir. Motorun sürücü kontrolü çoğunlukla bu iş için ayrılmış bir sayısal hareket denetleyici ile sağlanır.

Sözü edilen motor sürücüleri komut almak ve de önemli durum ve kontrol sinyallerini bilgisayara iletmek için, Profibus, CAN+, RS-485 vb çeşitli seri iletişim protokolleri kullanmaktadırlar. Motor sürücülerinin hareket kontrolünün merkezinde olduğu düşünülerek, bu tezde yüksek doğruluklu endüstriyel uygulamalarda kullanılabilecek bir DC motor sürücüsü tasarımının ve uygulanmasının incelenmesi amaçlanmıştır.

Bu amaca ulaşmak için çeşitli kontrol kavramları ve önemli uygulama detayları araştırılmıştır. Güç birimi, akım denetleyicisi ve hareket denetleyicisi içeren düşük güçlü bir DC motor sürücüsü üretilip test edilmiştir. Sürücünün performansını artırmak için alt birimleri üzerinde bir çok tasarım değişikliği yapılmıştır.

Sonu olarak yksek gl bir srcs yapabilmek iin; g elektroniki ve kontrol uygulamaları zerine tecrbe ile bilgi edinilmiřtir.

Anahtar Kelimeler: DC motor hareket denetimi, DC motor voltaj dzenleyicisi, DC motor akım denetleyicisi, DC motor srcs

To My Family

ACKNOWLEDGMENTS

I am deeply grateful to my supervisor advisor, Asst. Prof. Dr. Melik Dölen, for his guidance, advice, criticism, encouragement and insight throughout the research.

I wish to thank my examining committee members, Prof. Dr. Y. Samim Ünlüsoy, Prof. Dr. M. Kemal Özgören, Asst. Prof. Dr. A. Buğra Koku and Asst. Prof. Dr. Ahmet M. Hava for their suggestions and criticism.

I am grateful to Prof. Dr. Mehmet Çalışkan for large lab workspace he provided and to Prof. Dr. Tuna Balkan and Asst. Prof. Dr. Buğra Koku for providing lab equipment necessary for this work.

I would like to thank TUBITAK for its partial financial support to this work (MISAG 257).

Special thanks to my brother Avşar Polat Ay for his help on electronic hardware and most valuable support, to my friend Etkin Özen for his assistance on experiments and late night coffees he brought and to E. Doruk Özdemir for his support.

I want to thank Ertan Murat for his technical help on early stages of the study.

Finally I would like to thank my parents Nuran Ay and Süleyman Ay for their support and valuable advices through this study, their wisdom guided my steps.

TABLE OF CONTENTS

PILAGARISM.....	iii
ABSTRACT	iv
ÖZ.....	vi
ACKNOWLEDGEMENTS	ix
TABLE OF CONTENTS	x
LIST OF FIGURES	xii
LIST OF TABLES	xv
LIST OF SYMBOLS	xvi
1. INTRODUCTION.....	1
2. LITRATURE SURVEY	8
2.1. Introduction.....	8
2.2. Power Electronics	8
2.3. Motor Drive Control	9
2.4. Machine Tool Axis Control	17
3. VOLTAGE REGULATOR FOR DC MOTOR.....	21
3.1. Introduction.....	21
3.2. Full Bridge DC-DC converter.....	21
3.3. DC/DC Converter Designs	25
3.3.1. Converter with Discrete Components.....	26
3.3.1.1. Design of Switching Module	26
3.3.1.2. MOSFET Gate Drivers	29
3.3.1.2.1. Selection of Gate Driver	32
3.3.1.3. Signal Isolation and Dead-time Generation.....	36
3.3.1.4. Overall Design of Converter with Discrete Components	41
3.3.2 Converter with Integrated Components	46
3.4. Voltage Regulations for DC to DC Power Converters	49
3.4.1. Discrete-time Voltage Regulation.....	50
3.4.2. Continuous-time Voltage Regulation	53
3.5. Performance Tests.....	56
3.6. Conclusion	61
4. CURRENT REGULATOR.....	62
4.1. Introduction.....	62
4.2. Current Regulator Design	64
4.2.1. Continuous-time Current Regulator.....	65
4.2.2. Discrete-Time Current Regulator	68
4.3 Hardware Implementation	72
4.3.1. Current Sensors	72
4.3.2. Continuous-time Current Regulator.....	74
4.3.3. Discrete-time Current Regulator	77
4.4 Performance Tests.....	80

4.4.1. Continuous-Time Current Regulator	80
4.4.2. Discrete-Time Current Regulator	85
4.5. Conclusion for	91
5. MOTION CONTROLLER.....	92
5.1. Introduction.....	92
5.2. Elements of Motion Control.....	93
5.2.1. Designing Motion Controller.....	95
5.2.1.1. Incremental Velocity Controller	99
5.2.1.2. Incremental Position Controller	102
5.3 Implementation of Motion Controllers	103
5.3.1. Hardware Considerations.....	104
5.3.2. Firmware	110
5.3.2.1 Initiation of Microcontroller	111
5.3.2.2 Interrupt Routine	112
5.3.2.3 Control Routine.....	113
5.4. Performance Tests.....	116
5.5. Conclusion	124
6 CONCLUSION AND FUTURE WORK.....	125
APPENDICES	131
Appendix A.....	131
Appendix B.....	134
Appendix C.....	144
Appendix D.....	149
Appendix E	152
Appendix F.....	158
Appendix G.....	174
REFERENCES.....	176

LIST OF FIGURES

Figure 1.1 Block diagram of LSC 30/2.....	3
Figure 1.2 Maxon LSC 30/2	4
Figure 1.3 Medel MD:01	5
Figure 1.4 Tunçmatik SmartDrive PRO 4Q	6
Figure 1.5 TUBITAK BILTEN DC motor drive for ISDEMIR.....	6
Figure 2.1 Classical industrial motion controller.....	10
Figure 2.2 Acceleration, Velocity, Position feedback.	12
Figure 2.3 PI+ Controller.....	13
Figure 2.4 Computed Torque Feedforward	13
Figure 2.5 State variable motion controller (incremental format)	14
Figure 2.6 Dynamic Stiffness of the state variable motion controller	15
Figure 2.7 Zero tracking error state variable motion controller.....	16
Figure 3.1 Full-bridge DC-DC power converter for DC motor applications.	23
Figure 3.2 MOSFET H-bridge.....	29
Figure 3.3 Block diagram of a combined signal and power isolation gate-driver..	31
Figure 3.4 Separate signal and power isolation gate driver	31
Figure 3.5 Signal level shifter and bootstrap power gate driver	32
Figure 3.6 Block diagram of IR2113.	34
Figure 3.7 Optical isolation and dead-time generation circuitry.	37
Figure 3.8 Modes of operation for dead-time generator.	38
Figure 3.9 Revised optical isolation and dead-time generation circuit.....	40
Figure 3.10 Gate signals for one inverter leg.....	41
Figure 3.11 First version of converter A.....	42
Figure 3.12 Final version of the converter A.....	44
Figure 3.13 First version of converter A.....	45
Figure 3.14 Enhanced converter A.	45
Figure 3.15 A simplified block diagram of L298.	47
Figure 3.16 Circuitry for converter B.	48
Figure 3.17 Converter topology using L298.....	49
Figure 3.18 PIC program to control the converter.....	52
Figure 3.19 Block diagram of controller for voltage regulator B	54
Figure 3.20 Controller for voltage regulator B	55
Figure 3.21 Voltage regulation on converter A	57
Figure 3.22 Voltage regulation on converter B	57
Figure 3.23 Detailed delay observed on voltage regulator B.....	58
Figure 3.24 Voltage regulation on converter A	59
Figure 3.25 Voltage regulation on converter B	60
Figure 4.1 Continuous-time current regulator.	63
Figure 4.2 Discrete-time current regulator.....	64
Figure 4.3 Root locus of current plant.	65
Figure 4.4 Bode plot of current plant.....	66

Figure 4.5 Root locus of compensated system.....	67
Figure 4.6 Bode plot of compensated system.	68
Figure 4.7 Root locus of discrete-time current plant.	69
Figure 4.8 Bode plot of discrete-time current plant.	70
Figure 4.9 Root locus of compensated discrete-time system.....	71
Figure 4.10 Bode plot of compensated discrete-time system.	71
Figure 4.11 LEM Hall effect sensor.	73
Figure 4.12 LEM output voltage.....	73
Figure 4.13 Current sensor circuitry.	74
Figure 4.14 Continuous-time current regulator circuit.	75
Figure 4.15 Block diagram of continuous-time current regulator.	76
Figure 4.16 Continuous-time current regulator.	76
Figure 4.17 Flow chart of discrete-time current regulator firmware.	79
Figure 4.18 Sine input and output without offset.	83
Figure 4.19 Trapezoidal Input.....	83
Figure 4.20 Triangular Input at 50 Hz.	84
Figure 4.21 Frequency response of current regulator.	85
Figure 4.22 A/D conversion test input 10 Hz- 80 Hz.	87
Figure 4.23 A/D test results at 100 Hz.....	88
Figure 4.24 A/D test results at 150 Hz.....	88
Figure 4.25 A/D test results at 200 Hz.....	89
Figure 4.26 A/D test results at 250 Hz – 350 Hz.	89
Figure 5.1 Standard continuous time velocity cont. implementing tachogenerator.	94
Figure 5.2 Discrete-time Velocity Controller.....	95
Figure 5.3 Root locus of velocity plant.....	98
Figure 5.4 Bode plot of velocity plant.....	98
Figure 5.5 Root locus of incremental P velocity controller compensated system.	100
Figure 5.6 Bode plot of incremental P velocity controller compensated system... ..	100
Figure 5.7 Step response of Closed Loop Incremental P Velocity Controller.....	101
Figure 5.8 Model of Incremental P Velocity Controller.....	101
Figure 5.9 Closed loop root locus of incremental position controller.....	102
Figure 5.10 Closed loop bode of incremental position controller.....	103
Figure 5.11 Control block diagram.....	104
Figure 5.12 Control circuitry.....	104
Figure 5.13 D/AC circuit.....	105
Figure 5.14 RS-232 Interface circuit.....	107
Figure 5.15 Encoder interface circuit.....	108
Figure 5.16 Timing diagram of encoder interface.....	109
Figure 5.17 Logic circuitry implementing 4X interpolation.....	110
Figure 5.18 Program flow chart.....	111
Figure 5.19 Interrupt routine flow chart.....	112
Figure 5.20 Control Routine of incremental velocity controller.....	114
Figure 5.21 Control Routine of incremental position controller.....	115

Figure 5.22 Controller circuitry	116
Figure 5.23 Velocity output of incremental velocity controller	117
Figure 5.24 Position response of incremental velocity controller	117
Figure 5.25 Velocity output of incremental velocity controller	118
Figure 5.26 Position response of incremental velocity controller	118
Figure 5.27 Velocity output of incremental velocity controller	119
Figure 5.28 Position response of incremental velocity controller	119
Figure 5.29 Velocity output of incremental position controller	120
Figure 5.30 Position response of incremental position controller	120
Figure 5.31 Velocity output of incremental position controller	121
Figure 5.32 Position response of incremental velocity controller	121
Figure 5.33 Velocity output of incremental position controller	122
Figure 5.34 Position response of incremental velocity controller	122
Figure A.1 Circuit symbols and sign conventions for various MOSFET devices.	129
Figure A.2 Current-voltage characteristics of n-channel MOSFET	129
Figure B.1 Motor model by superposition.....	132
Figure B.2 Ripple current and voltage.....	133
Figure B.3 Ripple Current Measurements A	134
Figure B.4 Ripple Current Measurements B.....	135
Figure B.5 Armature Inductance Measurements	137
Figure C.1 High side and low side resistive current shunts.....	142
Figure C.2 Shunt Resistive Current Sensor	144
Figure C.3 Current sensor circuitry	145
Figure C.4 Hall Effect Principle	146
Figure E.1 RS-232 electrical levels and data format	150
Figure E.2 DTE device connector.....	152
Figure E.3 DCE device connector	153

LIST OF TABLES

Table 3.1 Ratings of the motor.	27
Table 3.2 Summary of MOSFET driver chips.....	33
Table 3.3 H-Bridge ICs.....	46
Table 4.1 Hall Effect Sensors	72
Table 5.1 Friction experiment data	109
Table 5.2 Truth table of Encoder Interface Circuit.....	112
Table 6.1 Technical attributes of various motor drives	126
Table 6.2 Comparison of microcontrollers	129
Table B.1 Armature inductance experiment results.....	136
Table B.2 Back emf constant experiment results.....	138
Table B.3 Friction experiment data	139
Table C.1 Comparison of Current Sensors	141
Table D.1 Sine experiment results.....	147
Table G.1 Cost analysis of converter A	174
Table G.2 Cost analysis of converter B	174
Table G.3 Cost analysis of current regulator	174
Table G.4 Cost analysis of motion controller	175
Table G.5 Cost analysis of motor drives.....	175

LIST OF SYMBOLS

b	Viscous friction coefficient [Nm/rad/s]
C	Capacitance [F]
$C(k)$	Counter value
C_b	Bootstrap capacitance [F]
d_A	duty cycle
F_{OSC}	Oscillator frequency [Hz]
F_{PWM}	PWM frequency [Hz]
F_{st}	Saw tooth signal frequency [Hz]
$G_p(z)$	Transfer function of plant
$G_{PC}(z)$	Current regulator plant
$G_{PP}(z)$	Position plant discrete time transfer function
$G_{PV}(z)$	Velocity plant discrete time transfer function
i	Current drawn by the motor [A]
i_o	output current [A]
I_{CBL}	Leakage current of the bootstrap capacitor
I_{QBS}	Quiescent current for the high-side driver circuitry
J	Equivalent Mass Moment of Inertia of Motor [kgm ²]
K_e	Back emf constant [V/rad]
K_p	Proportional gain
K_t	Torque constant [Nm/A]
L_a	Armature inductance [H]
$M(z)$	Manipulation
Q_g	Gate charge of MOSFET (IRF 530N) at high side
Q_{LS}	Voltage level-shift-charge (per cycle) of IR2113
R	Resistace [R]
R_a	Armature resistance [R]
t_{on}	switch on period [s]
t_{off}	switch off period [s]
T_{A+}, T_{A-}	switch states of leg A

T_{B+}, T_{B-}	switch states of leg B
T_c	Coulomb (Dry) Friction Coefficient [Nm]
T_L	Load torque [Nm]
T_m	Torque developed by motor [Nm]
T_{OSC}	Oscillation period [s]
T_s	switching period [s]
T_{sm}	Max. switching period [s]
v_o	output voltage [V]
V_a	Terminal voltage [V]
V_{An}	output voltage of leg A [V]
V_{Bn}	output voltage of leg B [V]
V_{cc}	Supply voltage of IR2113 [V]
V_{dc}	voltage source [V]
V_F	Forward bias voltage of bootstrap diode [V]
V_L	Voltage drop across the low side MOSFET [V]
V_o	average voltage output of the converter [V]
Z	Zero order hold
$\theta(z)$	Position [rad]
ω	Motor's angular velocity [rad/s]

CHAPTER 1

INTRODUCTION

Today's industry heavily relies on precision motion control. Newer machinery requires higher accuracy and speed in order to perform efficiently. It can be said that improved efficiency and better quality are this era's motto. Industrial applications are spread to a wide range, covering different applications at both form and function. These applications range from computer hardware such as; CD, DVD players, hard discs to machine tools such as precision machining centers, lathes or milling machines; to industrial automation applications such as textile machinery, paper machinery, packaging machines; to robotics and to aerospace industry. These diverse applications have a common point which is the increasing demand for precision motion control.

Motion control problems in these various fields are of the same in nature. All of these devices have an electric motor, a machine part driven by motor and a process to be controlled. No matter what the type of motor is; in order the control motion motor power should be controlled. Thus power electronics is an issue for motion controllers. Power converters, with high efficiency and accuracy are a necessity for motion control. Computer technology developments has added another component to motion controllers, which is microprocessors. Advent of microprocessors has enabled control engineers to put embedded controllers on applications. Nowadays motor motion controllers measure velocity and current, calculate and apply required manipulation input to the system according to the command taken from a main computer. Additional to these subjects in order to have a precise motion control application, designers should have an extensive knowledge on mechanical systems.

Thus mechanical engineering is another important topic for motion control applications.

Field of motion control as stated above is an interdisciplinary field. Control designer should be knowledgeable on control theory, power electronics, mechanical engineering, computer engineering and control engineering fields in order to design a precise motion controller.

There are various DC and AC motor drive producers in the industry. Some of these drive manufacturers are, Emerson Motor Technologies, ABB group, Maxon Motor and Hitachi. These companies produce high precision DC motor drives with very high power such as 18 MW. Different designs are implemented in the industry some of which will be summarized below.

Emerson motor technologies' DC motor drive Mentor II is a digital variable speed drive with a current rating of 25A to 1850A. Mentor II is programmable DC motor drive with a simple interface for PC connection. The drive has a comprehensive self tuning algorithm which increases current loop performance and a PID digital control is added to the system to further increase its performance. Mentor II has a RS-485 port for enabling communication between other drives for synchronization, PLCs and computers for control. Speed and position command can either be analog or digital thus creating more implementation choices. [1]

ABB DCS500 DC motor drivers (with 25A to 5200A current ratings) have integrated digital speed and torque/current controller. It has also 16 digital I/Os, 4 analog inputs, 3 analog outputs, a tachogenerator and an encoder input. Also, the drive can be optionally furnished with serial communication modules for Profibus, CS31, ModBus, ModBus+, CANopen, ControlNet, and DeviceNet. ABB is also the vendor of a PC program for monitoring and controlling DCS500 enabling users to create their own control routines. [2]

Another example is Maxon motor 4-Q-DC servo control LCS 30/2 drive with a power rating up to 50W. The drive has a current controlled voltage regulator. A PI current and a PI speed control is implemented on this drive with a current limiter between controllers for safe operation as shown in the block diagram of this drive in Figure 1.1. LCS 30/2 has a tachogenerator and encoder inputs for speed regulation. Maxon supplies a serial interface card on demand with a RS-485 interface. [3]

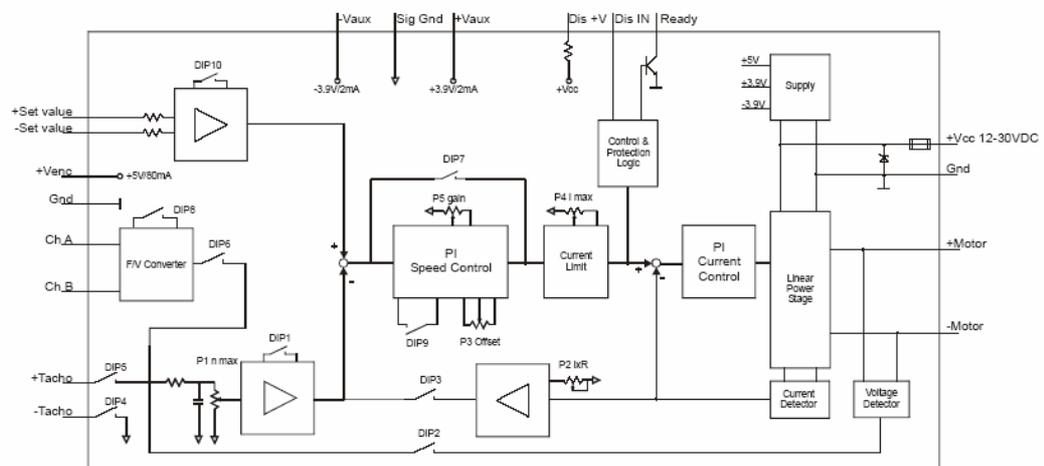


Figure 1.1 Block diagram of LSC 30/2

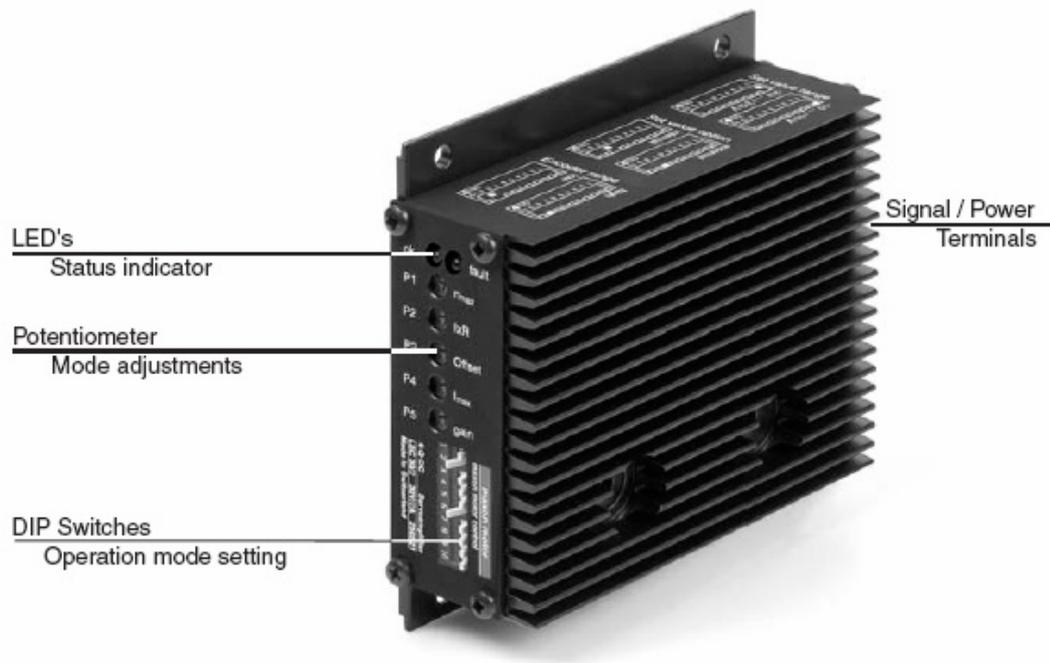


Figure 1.2 Maxon LSC 30/2

Hitachi's L300P series DC motor drives with a power rating up to 132kW are all digital drive implementations. A typical driver has an EMI filter for power and command modules, a 4 port LED display for monitoring. Motor commands can either be generated by manually from operator panel or by a special PC software. Similarly, FL300P series have a built-in RS-485 communications interface that can be alternatively replaced DeviceNet, Profibus, Lonworks and Ethernet interfaces on customer's demand [4].

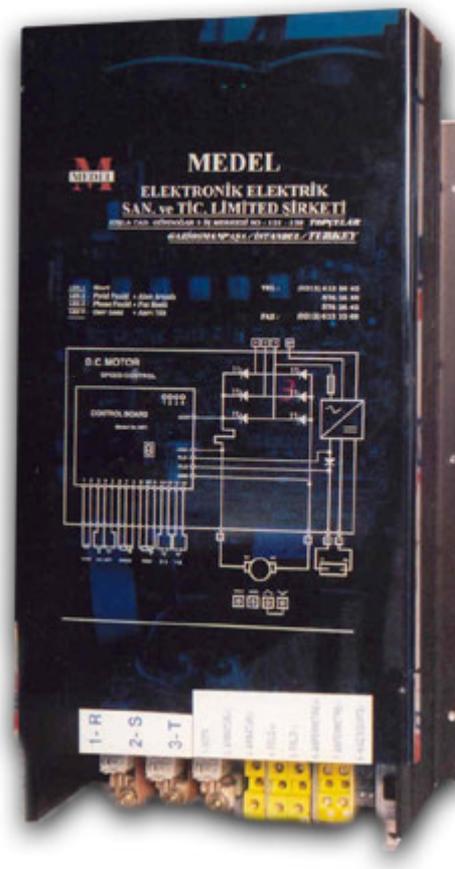


Figure 1.3 Medel MD:01

An example for Turkish motor drive manufacturers is Medel Elektronik Elektrik Sanayi Ve Ticaret Limited Şirketi which produces a DC motor drive MD01 with 1 to 4 Hp (0.75-3kW) with continuous time current and velocity control. The drive has a numeric display to monitor the process and has two modes of operation which are velocity or current control modes. Motor drive is manually controlled. [61]



Figure 1.4 Tunçmatik SmartDrive PRO 4Q

Tunçmatik is another DC motor drive manufacturer in Turkey. Company's DC motor drive SmartDrive PRO 4Q is a 10-1000 Hp (7.46- 745kW) motor drive with a digital motion controller. Motor drive has a 10bit motion controller with a PI motion control with 0.1% error. DC motor drive has RS-232 and RS-485 interfaces and has analog and digital I/O. [62]



Figure 1.5 TUBITAK BILTEN DC Motor drive for ISDEMIR

Also TUBITAK BILTEN has DC motor drive projects. An example of such project is the DC Motor drive system for universal machine boom turning DC motors for

ISDEMIR. The DC motor drive manufactured is a 70kW power with unity power factor.

As summarized above, contemporary electrical drives are usually equipped with a current regulated voltage source inverter and on top of those, a motion controller utilizing one or more micro-processors/micro-controllers and/or digital signal processors (DSPs). Such motor drive interfaces are mostly through RS-485 and motor drive control is mostly performed by a dedicated digital-motion controller. The motor drives lie at the heart of every (multi-axis) motion control system. Hence, the aim of this thesis is to explore the design and implementation of a DC motor drive system suitable for industrial applications requiring high precision motion control. In order to achieve this goal, various underlying control concepts and important implementation details are explored in this study. A small DC motor motion controller with a power module, a current regulator and a motion controller is built and tested to gain important “know-how” necessary for building high performance (and high power) DC motor drives to be used in high-precision motion control.

The organization of this thesis is as follows: Chapter 2 describes the state of the art in the relevant fields in modern motion control. The next chapter focuses on the design and implementation of two different voltage regulators while Chapter 4 discusses the design of two current controllers for the DC motor: continuous time current controller, and discrete-time current controller. One of each voltage and current regulator designed at these chapters are chosen and a motion controller is designed and implemented which is described at Chapter 5. Finally, Chapter 6 draws some conclusions based on the experiments on these regulators. Future improvements to these regulators are also included to that chapter.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

The motion control field is an interdisciplinary field which involves power electronics, control engineering and mechatronics. In order to conduct a study on motion control, first power electronics components involved in motor motion control should be reviewed. At Section 2.2 a brief review on power electronics components is given with this thought in mind. Section 2.3 extensively discusses motion control techniques of DC motor and state of the art control techniques. This chapter concludes with a review of axis motion control techniques at Section 2.4.

2.2 Power Electronics

Since 1950s with the invention thyristors started the modern solid state power electronics era. Invention of thyristors was followed by developments of other type of devices such as gate turn-off thyristors, bipolar power transistor, power MOS field transistors (MOSFET's), insulated gate bipolar transistors (IGBT's), static induction transistors(SIT's), static induction thyristors (SITH's) and MOS-controlled thyristors. Introduction of power MOSFET's was in mid 1970's. Power MOSFET's found a large market acceptance and it is dominant in high frequency low power applications such as switching mode power supplies and brushless DC motor drives. Development of Darlington power transistor modules with built in feedback diodes gradually pushed the voltage fed transistor inverter rating to several hundred kilowatts. In early 1980's, IGBT's were introduced. IGBT's brought a visible change in the trend of power electronics. The IGBT is a hybrid

device that combines the advantages of the MOSFET and the bipolar transistor. Although device is slightly more expensive than the power transistor, the advantages of higher switching frequency, MOS gate drive, the absence of the second breakdown problem, snubberless operation, reduced Miller effect, and the availability of the monolithic gate driver with “smart” capability provided an overall system advantage to IGBT power converters [5, 6, 7]. For high frequency, high power applications, SIT has been introduced in early 1990’s. The reliability, noise and radiation hardness of the SIT are superior to the power MOSFET [5,8]. Recent advances in power electronics components discussed above led to decrease in cost of motor drives and an increase in power of these drives. However, development in power electronics components alone was not the cause of increased capabilities of motor drives. Advances in microcontrollers and control systems were also an important issue in motor drive development.

2.3 Motor Drive Control

High performance drive control systems are widely used in control applications such as machine tools, material conveyors, transportation systems, packaging, printing, web handling, robots, textiles, and food processing. The motion control algorithms are based on the mechatronics assumption of nearly ideal electromagnetic torque control. Feedback devices, chiefly encoders and resolvers, are employed in these systems to sense motor position and calculate sample average motor velocity. Lorenz’s review states that at 1999 the vast majority of motion control algorithms close the motion control loops in one of two ways; an average velocity loop is cascaded with position loop and multiple state variable loops are closed in parallel [9]. Cascaded loops usually employ current, speed and position control loops. Current limiting systems are crucial because of the low internal impedance of motors, most notably DC motors.

There are two different methods for current limiting; these are interventionist system and regulating system [10]. The interventionist scheme normally operates in

speed mode and enters the current limiting mode only when the current exceeds a threshold value. In current-limiting mode, gate signals are blocked, this pulls down the armature current. When the armature current is between acceptable limits, speed controller resumes its action. The advantages of this controller are; its speed, relative simplicity and low cost. Also designer does not design a current controller so that design of a velocity controller is enough [11]. However this system has the disadvantage of allowing the current to overshoot in order for the limiting action to intervene effectively. A speed control loop with a current regulating loop is called a regulating speed control loop. In this cascaded configuration current loop is always active and the output of the speed controller, which is limited to a pre-set value, constitutes the command signal to the current loop. This method allows smooth and continuous transition from speed regulation mode to the current regulating mode, so uncontrolled current overshoots are avoided.

Industrial motor drives apply two different kinds of motion control paradigms. The first one is an average velocity loop cascaded with a position loop. The second type is state variable loops which are closed in parallel. Most used industrial speed controllers are proportional integral (PI) speed controllers. In fact Lorenz, Lipo and Novotny [55] state that, the *de facto* industry standard for motion control is to use a PI velocity loop and a proportional position loop with a velocity command separately fed via what is generally described as a "velocity feedforward" path. A digital implementation of such a controller is shown at Fig. 2.1, most of this review is adapted from above mentioned authors review.

Major advantages of this approach are straight-forward implementation to microcontroller and its ability to maintain zero steady state error to a step change. However the undesirable speed overshoot, the sluggish response due to sudden change in load torque and sensitivity to controller gains are disadvantages of this method [12]. Barry proposes Proportional Integral Derivative (PID) velocity loops which are rarely used because calculating sample average velocity suffers from resolution limitations so severe as to make them impractical. Schmidt and Lorenz propose an acceleration feedback method to overcome this problem [13].

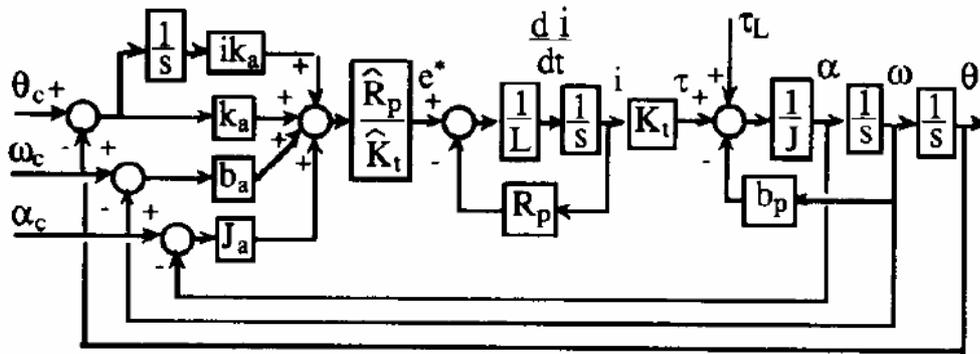


Figure 2.2 Acceleration, velocity, position feedback [13].

A second velocity loop frequently used in cascaded loop systems is Pseudo-Derivative Feedback (PDF). PDF is less responsive to the velocity command than PI. However, the change in structure allows PDF to have higher integral gains while avoiding overshoot to otherwise infeasible (step change) motion commands. This trade off allows better response to low frequency disturbances without a sluggish speed response [9,14]. Kang shows that by applying a torque observer to PDF controller; overall speed system becomes robust against mechanical parameter variations as well as sudden load changes. Lorenz proposes a PI+ controller where a PDF controller is augmented with a feed-forward term.

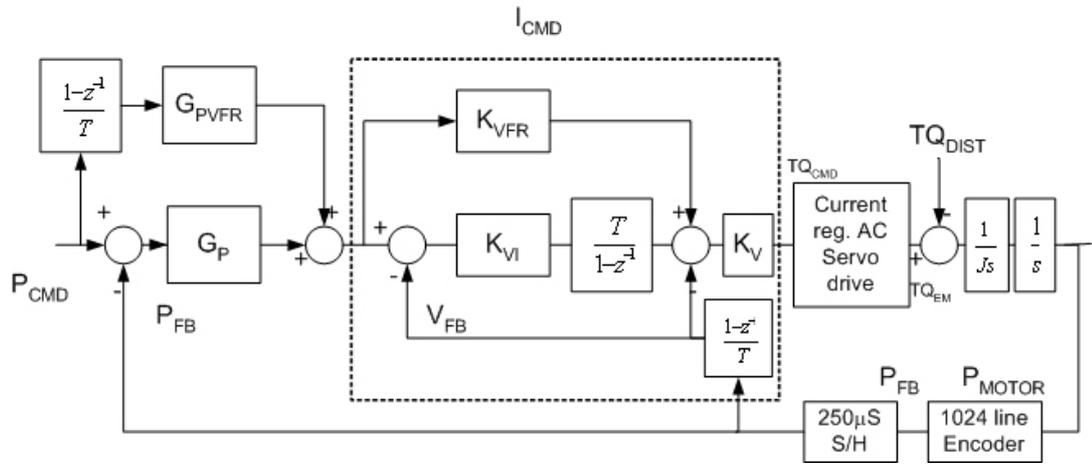


Figure 2.3 PI+ Controller [9].

Feedforward term injects the command ahead of the integral making the system more responsive to commands. Barakat proposes a speed control for a DC motor using Computed Torque Control scheme. [15] The objective of the researchers was to maintain angular velocity of the motor in the presence of changes in the dynamics of the motor and the operating environment. Authors proposed a torque control scheme based on the inverse dynamics of the motor by modifying a conventional PID controller.

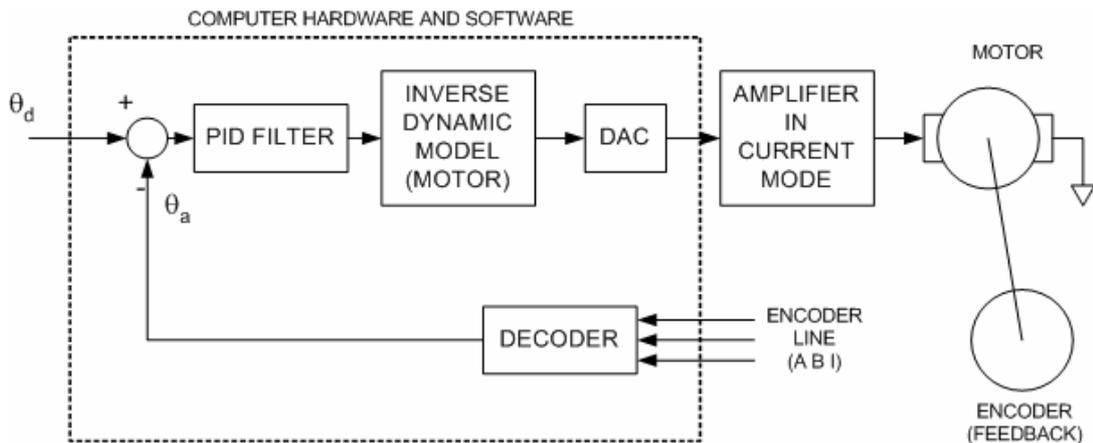


Figure 2.4 Computed Torque Feedforward [15]

Pulse Width Modulation (PWM) signals used to control voltage acting on motor creates ripples in torque. Lee proposed a cascade controller to eliminate these ripples observed in torque. [16] An addition to conventional PI, Mahfouz suggests

an I-P controller where proportional term is moved to feedback path and acts like feedback compensation. [17]

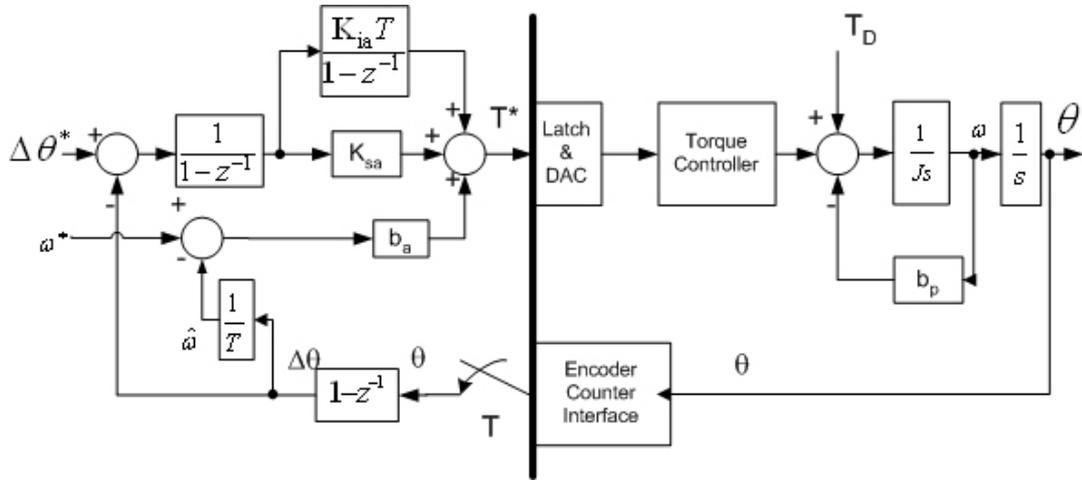


Figure 2.5 State variable motion controller (incremental format) [55].

As Lorenz et. al. [55] showed classical industry standard digital motion controller can be expressed as state variable digital motion controller shown at Fig. 2.6. This topology is mathematically identical to that of Fig. 2.1 with the gain relationships:

$$\begin{aligned}
 b_a (\text{nm} / \text{rad} / \text{s}) &= K_{pv} \\
 K_{sa} (\text{nm} / \text{rad}) &= K_{pp} K_{pv} + K_{iv} \\
 K_{ia} (\text{nm} / \text{rad} \cdot \text{s}) &= K_{pp} K_{iv}
 \end{aligned}
 \tag{2. 2}$$

One direct benefit of this topology is the physical units scaling which relates directly to the disturbance rejection, (dynamic stiffness) of the system as shown in Fig.2.7. From Fig. 2.7 it can be seen that the disturbance rejection dynamics of the state variable motion control has predictable properties, including "infinite static stiffness" as provided by the integral-of-position-error state. This control topology provides a system with extremely high stiffness to disturbance inputs. It should be noted that the state variable system topology of Fig. 2.6 is still error-driven and thus will produce dynamic tracking errors. Figure 2.6 also shows the incremental format typically used so that the position command variables can be implemented with the smallest word size needed to allow for continuous operation at maximum speed.

Note that such feedforward (computed torque) methods provide means for zero error tracking for nonlinear as well as linear systems. Such approaches neither affect the stability nor the disturbance rejection dynamics provided by the state feedback controller. However, such computed torque feedforward methods cause a more intense focus on torque accuracy and torque dynamics. In effect, by enabling zero tracking error to be achieved, these approaches enable the motion control designer to verify if accurate, dynamically capable, low-distortion torque control has been achieved.

There are some other state variable loops; used in industry such as a PID position controller for which velocity and position commands are fed to controller. In order to have command tracking a full command vector (i.e. position, velocity, and acceleration) should be used. However for most industrial applications acceleration is not available. So this controller must be tuned in order to avoid overshoot. There are other state variable speed control methods. Hannamoto proposes a speed controller without a speed sensor [18]. The controller calculates estimated speed and torque by reduced order observer method. Authors also implemented a torque feed-forward control system for dynamic load changes. Chevrel proposes a parallel DC motor speed controller and compares it with cascade controllers designed with dominant poles compensation and pole placement strategies [19].

There are recent studies on advanced controllers for speed control of motor drives. These controllers use adaptive, fuzzy and learning control techniques. Saneifard illustrates a simple fuzzy logic DC motor speed controller [20]. Kucer tested PI, sliding mode, state space controllers and fuzzy logic controller counterparts [21]. Ahmed describes a novel fuzzy logic speed controller that manages a DC motor. Also controller proposed implements a predictive current control system. Ullah illustrates application of National Semiconductors patented Neural Network, Fuzzy Logic control scheme NeuFuz for DC motor drives [22]. Rashidi proposes new

fuzzy based DC motor speed control systems. Authors conclude that a fuzzy PID controller using neural network has a good tracking performance [23]. Kulic proposes that a PI control system with a suitably trained Artificial Neural Network gain scheduler has better DC drive performance than a conventional PI control [24].

2.4 Machine Tool Axis Control

Since 1950's with the development of NC technology motion control systems of machine tools are researched extensively. The control loops of NC systems are designed to perform a specific task; to control the position and velocity of the machine tool axes. Thus they slightly differ from motor drive control systems. In NC systems each axes is separately driven and should follow the command signal produced by the interpolator. These signals are first processed by axis control systems and then suitable data is fed to motor drive.

Motion control systems of machine tools could be divided into two groups: Open Loop Control and Closed Loop Control. Designer selects suitable control technique for the machine tool. This selection is followed by another decision, whether NC machine should be point-to-point control or contour control. Koren [25] summarizes these techniques and selections. It is shown that open loop control is only suitable for point-to-point systems. However as quality requirements increase feedback control became necessary. Closed loop control techniques are useful for point-to-point control and contour control. Early research was on using P or PI control for point-to-point control. However these techniques were not applicable for reducing contour errors. Koren [26] summarized contour error reduction techniques as follows.:

- Applying more sophisticated axial controllers
- Adding feed forward controllers
- Using a cross-coupling controller.

Koren did not consider that optimal control, adaptive control, repetitive control and predictive control as main control strategies. A later work by Renton and Elbestawi [27] consider that main control strategies are:

- Feed forward control
- Cross- coupling control
- Optimal control.

Also authors suggest that Predictive control, Adaptive control and Learning control are parts of optimal control.

Proportional (P) controller is the simplest technique available. It sends correction signals proportional to the difference between the reference position and the actual position. This method is only suitable for systems with low feed rates and small disturbance loads. This method is improved to Proportional Integral Derivative controller. The method uses a correction signal, which is a combination of a proportional, an integral and a derivative of the position error. PID control is not suitable for corners and nonlinear contours. Also it has significant overshoots.

Feedforward controllers could be used in order to decrease tracking errors. These controllers use disturbances that can be measured before they affect the plant and use a model to determine the command signal required to minimize error. Typically this strategy implements a transfer function that is exact inverse of the plant model. However this method loses usefulness when a model could not be generated or plant model could not be inverted. Tomizuka [28] introduced the idea of Zero Phase Error Tracking Control (ZPETC) as a compromise when a plant model could not be inverted. Nevertheless this strategy did not eliminate gain error, or address the saturation problems of feed forward controller. Furthermore this method requires precise knowledge of the dynamic behavior of the axial drive system; otherwise an

additional error is introduced to the system. Weck [29] proposed an “Inverse Compensation Method“, which adds a low pass filter to the feed forward controller, therefore reducing saturation and improving corner tracking performance. Tomizuka proposed a robust motion control system integrating a friction compensator, a disturbance observer for the velocity loop, a position loop feedback controller and a feedforward controller acting on desired input [30]. Yeh and Hsu proposed cascading a polynomial digital prefilter to ZPETC in order to increase tracking accuracy of ZPETC [31].

Koren proposed Cross Coupling Control [32]. This method switched the focus of controllers from maintaining each axis at its target position, to minimizing path error. This is established by building a contour error model in real time, based on the feedback information from all axes as well as the interpolator. To find an optimal compensating law, and then to feed back correction signals to the individual axes. The difficulty with this method is in determining the point on the target path nearest to the plant's position. This is accomplished by using various closed-form solutions, specific to the type of path being followed. This fails where two path segments meet, or for an arbitrary path. Shih, Chen and Lee propose a new cross coupling control strategy with a simpler design process [33]. Another method is Optimal Control. Optimal controllers create a command sequence, which optimizes system performance. Generalized Predictive Control (GPC) can be taken as a special case of optimal control [34]. The basic idea behind is to make plant's predicted output coincide with a set point or desired known trajectory. Basic GPC method assumes a linear plant model. As a result, it cannot address issues such as backlash and asymmetric performance. Neither this method considers command saturation issues in its basic form.

Another control method is Adaptive Control. This method is usually used in adapting the cutting feed and speed to the variation in the cutting process in a typical machine tool. Smith, Annasway and Slocum evaluate adaptive PID and adaptive phase lead compensators for precision machine tool axis control [36].

Singhose [37] showed that input shaping can be used in order to decrease affect of vibrations in machine tools.

CHAPTER 3

DC TO DC CONVERTERS AND VOLTAGE REGULATORS FOR DC MOTOR

3.1 Introduction

DC Motors are used in motion control applications where accurate speed/torque is instrumental. Since the torque developed by a conventional DC Motor is directly proportional to the current drawn by the motor, the desired torque can be generated by regulating the current accordingly. To accomplish that, a current controlled voltage source (regulator), should be designed properly.

This chapter mainly focuses on DC to DC power converters. Even though there exist several DC to DC power converter topologies in power electronics literature; full bridge DC-DC converters, which enable bidirectional power flow between the motor and DC power supply, are far most common topology in DC motor control.

3.2 Full Bridge DC-DC converter

A power converter feeding a DC motor is expected to have the following capabilities [38]:

- The converter should allow both its output voltage (v_o) and current (i_o) to reverse so as to yield a four quadrant operation in the i_o - v_o plane.

- The converter should be able to control its current by holding it at its maximum “acceptable” level during fast acceleration and deceleration regimes. This acceptable current level (i.e. limit) is usually several times higher than the *nominal* current rating of the motor.
- The average voltage output of the converter (V_o) should vary linearly with its corresponding reference input, independent of the load on the motor in order to control the position (and/or the velocity) of the motor accurately.
- The converter should generate an output (armature) current with an acceptable form that minimizes the fluctuations in torque and speed of the motor.
- The converter output should respond to its command input as fast as possible such that the converter could be conveniently represented as a constant gain (without a dead-time/delay) in the overall transfer function model of the servo drive.

A linear power amplifier generally satisfies all these requirements. However, due to its lower energy efficiency, this choice is limited to a very low power range. Hence, either the switch mode power converters or the frequency controlled converters are the major alternatives in medium- to high-power DC motor drive applications.

A typical (switch-mode) DC-DC power converter is illustrated in Fig. 3.1. Even though this converter (also called H-bridge or full bridge) is mostly utilized for DC-DC power conversion, it has other distinct applications:

- DC-to-low frequency AC (50/60 Hz, sine wave) power conversion in a single phase uninterruptible AC power supplies (UPS).
- DC-to-high frequency AC conversion in switch mode transformer-isolated DC power supplies.

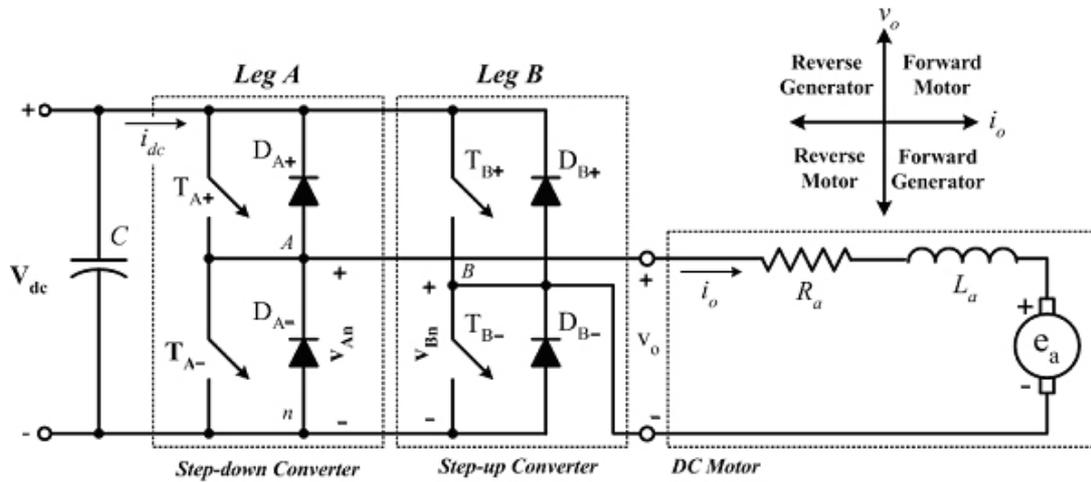


Figure 3.1 Full-bridge DC-DC power converter for DC motor applications.

Note that although the H-bridge topology remains essentially the same in all these applications, the type of (switch) control depends on the application. In the converter shown in Fig. 3.1, the input voltage is a fixed magnitude DC voltage V_{dc} (also called DC bus voltage). The output of the converter is a DC voltage V_o that can be controlled in magnitude as well as polarity. Likewise, the magnitude and the direction of the output current (i_o) can be also regulated. Hence, this H-bridge converter can operate in all 4 quadrants of the i_o - v_o plane, and the power flow through the converter can be in either direction: V_{dc} voltage source \leftrightarrow DC motor/generator.

In this converter topology, four **unidirectional** switches, which are controlled electrically, can conduct the current in only one direction (as indicated by arrow). For sake of discussion, they can be assumed as ideal switches, which can turn on and turn off instantaneously. As can be seen, since the diodes, which are capable of conducting current in the opposite direction, are connected in anti-parallel with these switches, a distinction must be made between the **on-state** versus the **conducting-state** of a switch. Due to these anti-parallel (also called **freewheeling**) diodes; when a switch is turned on, it may or may not carry a current, depending on the direction of the output current i_o . If the switch conducts a current in the

indicated direction, it is said to be in its conducting state. However, no distinction is needed when the switch is turned off.

The H-bridge converter consists of two *legs*, namely A and B. Each leg is made up of two switches along with their anti-parallel diodes. The two switches in each leg are switched in such a way that when one of them is in its off-state, the other one is switched on. Ideally, the two switches are never off simultaneously. Unfortunately, in practice, there exists a very small time interval where both switches might be conducting the current at the same time. This results in the short circuiting of the DC bus during this switching transient. Hence, both of the switches are intentionally switched off a short time interval, known as *blanking time* or *dead time*, to avoid short circuiting of the DC bus in practice.

Note that if the converter switches in each leg are switched in such a way that both switches in a leg are not off simultaneously, then i_o will flow continuously. Thus, the output voltage is solely dictated by the status of the switches. For instance, the output voltage of leg A (v_{An}), with respect to the negative terminal of the DC bus (n), is dictated by the switch states T_{A+} and T_{A-} as follows:

- When T_{A+} is on, i_o will flow through T_{A+} (which is in conduction state) if i_o is positive. When i_o is negative, the current will flow through D_{A+} where T_{A+} is in on-state. In either case, T_{A+} being on ensures that the point A (of Fig 3.1) is at the same potential as the positive terminal of the DC bus. Hence, with T_{A+} is on T_{A-} is off, $v_{An} = V_{dc}$.
- When T_{A-} is on, negative i_o will flow through T_{A-} (in conduction state) since D_{A+} is reverse biased (that is, the potential at the diode's cathode terminal is higher than that of its anode terminal). When i_o is positive, the current will flow through D_{A-} where T_{A-} is in on-state. Thus, with T_{A-} is on T_{A+} is off, $v_{An} = 0$.

From the argument above, it is obvious that v_{An} depends only on the switch status and is independent of the direction of i_o . Therefore, the output voltage of the

converter leg, averaged over one switching period (T_s), depends only on the DC bus voltage (V_{dc}) and switch-on period of T_{A+} (t_{on}):

$$V_{An} = \frac{1}{T_s} \int_{t=0}^{T_s} v_{An}(t) dt = \frac{V_{dc}t_{on} + 0 \cdot t_{off}}{T_s} = \frac{V_{dc}t_{on}}{T_s} = V_{dc} \cdot d_A \quad (3.1)$$

where $d_A \equiv t_{on}/T_s$ is the duty cycle (ratio) of T_{A+} , t_{off} is the off-interval of switch T_{A+} where $t_{on} + t_{off} = T_s$. Similar arguments do apply to the converter leg B, and V_{Bn} becomes

$$V_{Bn} = V_{dc} \cdot d_B \quad (3.2)$$

where d_B refers to the duty cycle of T_{B+} . Consequently, the converter's (average) output $V_o = V_{An} - V_{Bn}$ can be controlled by the duty cycle and is independent of the magnitude and the direction of current.

The converter's average output voltage is controlled by pulse width modulation (PWM) switching strategy. Two PWM switching techniques are utilized in power electronics literature:

- In PWM with bipolar voltage switching, (T_{A+} , T_{B-}) and (T_{A-} , T_{B+}) are treated as two switch pairs; the switches in each pair are turned on and off simultaneously.
- PWM with unipolar voltage switching is also known as double-PWM. The switches in each leg are controlled independently of the other one.

3.3 DC/DC Converter Designs

In this study, two converter designs one with discrete components and one with integrated components (namely A and B) are considered. Converter A uses MOSFET as the power switch and consists of MOSFET gate driver, optocoupler stage, and a dead-time generator while Converter B employs a single H-bridge integrated circuit (IC) that includes all relevant components. The descriptions of these converters follow.

3.3.1 Converter with Discrete Components

In Sec. 3.2, H-bridge, which uses ideal (unidirectional) switching devices, has been presented. In Power Electronics literature, there exist several alternatives for switching devices such as thyristors, Gate Turn Off (GTO) thyristors, variacs, and power transistors. As fully gate-controlled devices, power transistors such as Bipolar Junction Transistor (BJT), Metal Oxide Semiconductor Field Effect Transistors (MOSFET), and Insulated Gate Bipolar Transistor (IGBT) are most common choice for full bridge DC-DC power converter applications. Among all these power transistors, MOSFETs appear to be the best alternative for the converter considered in this study: MOSFETs with appreciable on-state current carrying capability along with off-state blocking voltage capability have been available for power applications since the early 1980s. They have become as widely used as BJTs and are in fact replacing BJTs in many applications where high switching speeds are important [38].

The converter topology, which utilizes MOSFETs as fast power switches in the H-bridge, is very versatile in nature as one can utilize the same system to drive higher-rated DC motors by simply switching to higher-rated MOSFETs (or even IGBTs) and increasing the DC bus voltage accordingly. Description of the design follows.

3.3.1.1 Design of Switching Module

The initial step of the power converter design is to select a proper MOSFET. To accomplish that, one should consider the ratings of the load (motor). The DC motor being used in this project was salvaged from an old photocopy machine and was manufactured by TOSHIBA Corporation (Model: DGM-321-2AA). The DC motor has a built-in gearbox with 1:22 gearing ratio. Table 3.1 summarizes motor ratings.

Table 3.1 Ratings of the motor.

Power	45W
Voltage	18V
Current	2.5A
Nominal Speed	59 rpm
Rated Torque	3 Nm

Based on these given ratings, a DC bus voltage of 24V seems to be adequate for all intensive purposes. This DC bus voltage is to be provided by a 24V-30W (1.3A) regulated power supply, which was currently available in the CNC Machine Tools and Automation Lab. With this information at hand, an N-channel MOSFET (IRF530N) is selected. The important properties of IRF530N are as follows:

- V_{DSS} : 100V (breakdown voltage)
- I_D : 15A (continuous drain current)
- $R_{ds(on)}$: 0.11Ω (equivalent drain-source resistance at turn-on)

Another important factor in selection of this device is its wide availability in the local part suppliers. The H-bridge topology employing this device is illustrated in Fig. 3.2. Notice that the gate resistance has an important impact on the switching dynamics of the MOSFET. Roughly speaking, a small gate resistance yields a faster switching device while reducing the heat generated by the device (i.e. switching losses). For instance, [39] discusses the effect of gate resistance on the switching performance of the MOSFET. Interestingly, a gate resistance of 0.47Ω yields the optimum switching performance on many MOSFET devices [39]. On the other hand, small gate resistances also increase the magnitude of gate currents at the turn on and turn off instances and such current pulses may not be supplied adequately by the gate driver of the device. Therefore, choosing a gate resistance is a compromise between the capabilities of the MOSFET gate driver and the switching performance of the converter. In this application, $R_G = 47\Omega$ appears to yield good switching performance with the selected gate driver. Notice that in Fig. 3.2 fast diodes (1N914

or 1N4148) are used in conjunction with the gate driver resistors. While turning off the MOSFETs, the diodes are forward biased and thus the energy stored in stray capacitances of the MOSFET (see Appendix A for more information) is not dissipated over the gate resistors. This scheme also creates an asymmetry between the switching times (e.g. slow turn on, fast turn off). Effectively, one can create a very small blanking time (5-10 ns) where the two MOSFETs of a leg are evidently turned off during a short period. Notice that a couple of power *Zener* diodes are connected between gate and the source of each device. The basic idea behind these protection devices is that; if the voltage across the gate and source of a MOSFET exceeds certain levels, which is usually 20V-30V, the thin oxide layer at the gate is damaged. Especially at high switching frequencies (50 kHz to 1 MHz) parasitic inductances might create voltage fluctuations at the gate-source of the device [39]. The magnitude of such oscillations may easily go above these tolerable gate voltage levels. Consequently, the Zener diodes clamp the gate voltage (V_{gs}) to $\sim 15.7V$ at turn on and $\sim -5.7V$ at turn off which in turn properly protecting the device.

Note that freewheeling diodes are not included to the design owing to the fact that an enhanced mode MOSFET like IRF530N intrinsically constitutes such an anti-parallel diode (called body diode). However, conservative designers sometimes include fast recovery diodes or snubber capacitors.

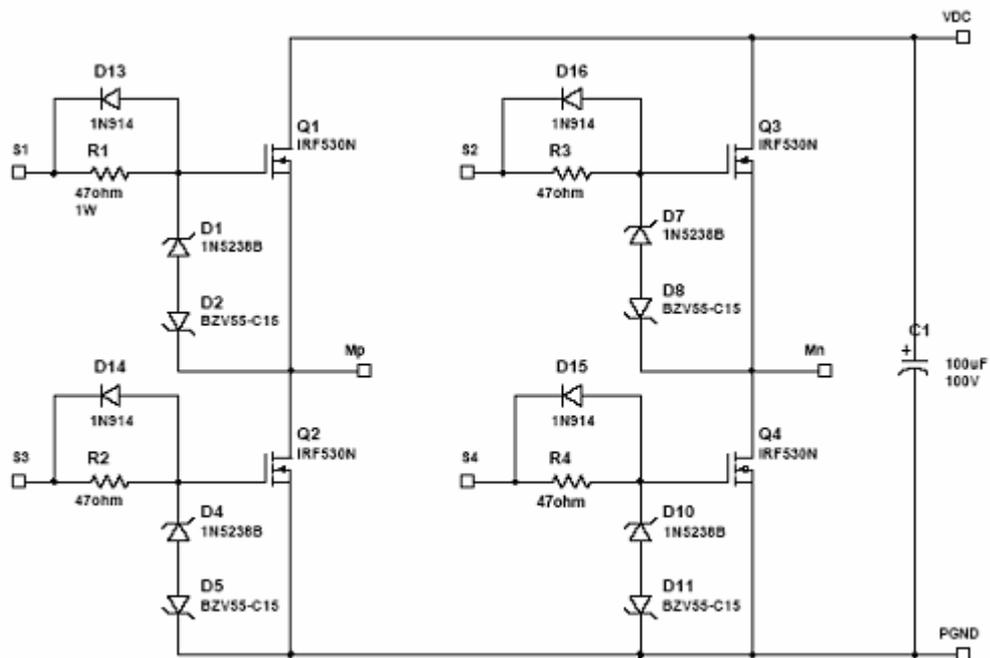


Figure 3.2 MOSFET H-bridge.

3.3.1.2 MOSFET Gate Drivers

The next stage of the design is to select a proper MOSFET gate driver circuitry which supplies the required voltage and current to the gate of the MOSFET so as to control the switching instances of the devices. There are various MOSFET gate-drivers available today from manufacturers, some with more features than others [40,41,42,43]. Most of the gate-drivers can easily switch a MOSFET up to a frequency of 100kHz and some can even go as high as 200kHz but in development now are low to medium kW converters that operate in the MHz region [44,45] for integration, which necessarily includes the gate-driver. This further increases the demands on the design, which now must also be integrate-able in the appropriate technology. Furthermore, recent development work in integrated converters has shown that in the MHz-range, the gate-driver can become an appreciable part of the volume [44].

There are three main types of gate-driver topologies for high voltage (up to 1000V) applications:

1. **Combined signal and power isolation** [46, 47] and shown in Fig. 3.3(a), can produce a compact design but is tricky to design and some have a limited switching frequency for the MOSFET. This is mainly due to modulation design criteria. Not all types of gate-drivers in this category require modulation, as some use pulse transformers [48, 49, Fig. 3.3(b)] to deliver the energy to the gate of the MOSFET. Recently work has been done on the use of piezoelectric transformers to establish isolated power transfer [46].
2. **Separate signal and power isolation** [40, 43]. This type of topology, shown in Fig. 3.4, is popular for higher power converters as it is less likely to form problematic ground loops, easily made by the multiple connections between the gate-driver and the power circuit. The drawback of this method is a double and bulky design. This type of gate-driver approach also provides isolation between the input signal and the MOSFET.
3. **Signal level shifting and power bootstrap** [41,50]. This is the simplest topology, depicted in Fig. 3.5, and is widely used in industry. It is more suited for low cost products; however, the level shifter is prone to error especially from reverse recovery currents generated by the bootstrap diode. It can also be difficult to implement protection features into this type of topology.

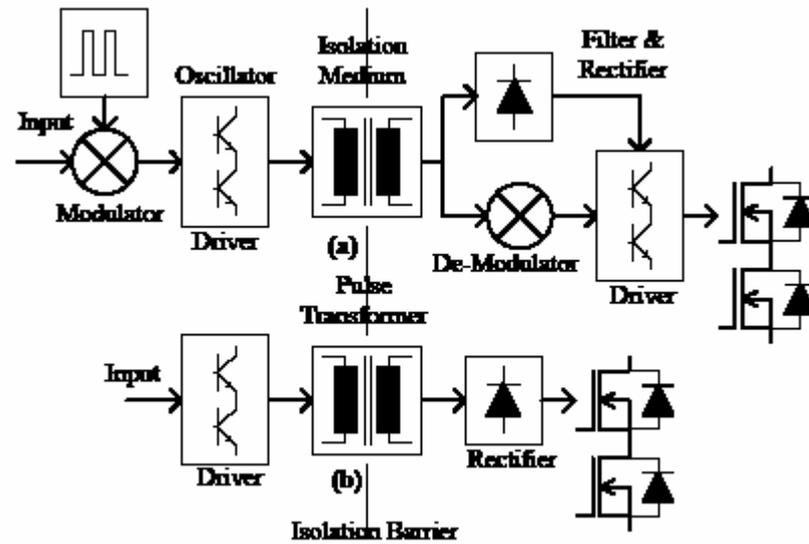


Figure 3.3 Block diagram of a combined signal and power isolation gate-driver: (a) Modulated signal version (b) Pulse transformer version [44].

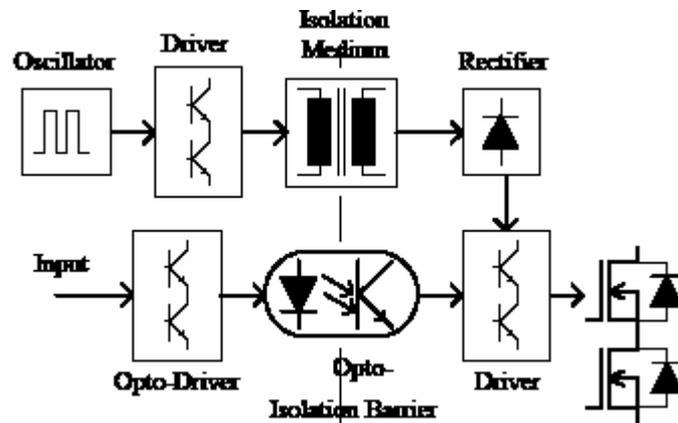


Figure 3.4 Separate signal and power isolation gate driver [44].

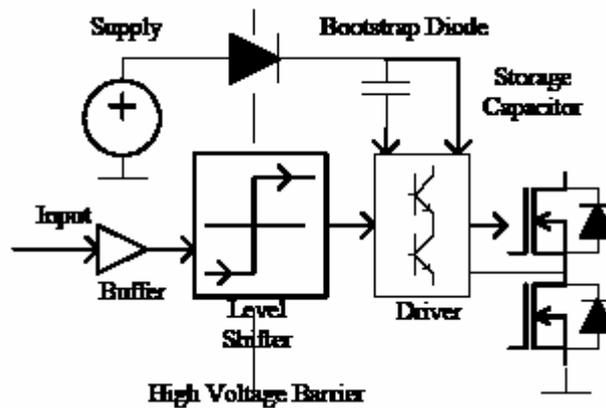


Figure 3.5 Signal level shifter and bootstrap power gate driver [44].

Other gate-driver schemes include capacitive coupling for cascaded devices in high voltage applications ($>1000\text{V}$) [51]. Integrated converters must also be tested during the design phase, and unlike discrete component converters cannot simply be repaired by device replacement.

3.3.1.2.1 Selection of Gate Driver

There exist various gate driver chips available in the market, Table 3.2 summarizes some of these popular chips widely used in industry. Some of these drivers require the MOSFET source terminal to be grounded (for the lower two MOSFETs in a full bridge or just a simple switching circuit). Some like IR2110/2113 can drive a MOSFET with the source at a higher voltage. These chips do have an on-chip charge pump, which means they can generate the 10-15 volts required to turn on the upper MOSFET in a full bridge. Likewise, the TDA340 even controls the switching sequence while some drivers can supply as much as 6 Amps current as a very short pulse to charge up the stray gate capacitance.

Table 3.2 Summary of MOSFET driver chips.

Manufacturer	IC	Features
Maxim and others	ICL 7667	Dual inverting driver
Maxim	MAX622/MAX1614	High side drivers
Maxim	MAX626/MAX627/MAX628	Low side drivers
International Rectifier	IR2110	High and low side driver
Harris / Intersil	HIP4080/4081/4082	Full bridge drivers
SGS Thomson (ST)	TD340	New full bridge driver with analogue or PWM speed demand input
Hewlett Packard	HCPL 316J	Gate driver optocoupler with integrated desaturation detection
Fuji Electric	EXB8**	Hybrid gate driver IC

Note that in the leg of a H-bridge converter, the gate drive requirements for a power MOSFET or IGBT utilized as a high side switch (e.g. T_{A+} and T_{B+} of Fig. 3.1) driven in full enhancement, i.e., lowest voltage drop across its terminals, can be summarized as follows[60]:

- Gate voltage must be 10-15V higher than the drain voltage. Being a high side switch, such gate voltage would have to be higher than the rail voltage, which is frequently the highest voltage available in the system.
- The gate voltage must be controllable from the logic, which is normally referenced to ground. Thus, the control signals have to be level-shifted to the source of the high side power device, which, in most applications, swings between the two rails.
- The power absorbed by the gate drive circuitry should not significantly affect the overall efficiency.

Considering above requirements as well as availability of the above-mentioned gate driver ICs in the local market, the International Rectifier's IR2113 is chosen to drive the MOSFETs. A block diagram of IR2113 is shown in Fig 3.6. To control each MOSFET independently, one needs to allocate one isolated power supply for each high-side MOSFET gate driver. On the other hand, since the sources of low-side MOSFETs (see T_A - and T_B - of Fig. 3.1) are connected to power ground, the gate drivers of these devices can share one isolated power supply. As a result, a full-bridge converter would need three isolated power supplies for all the gate drivers ($2 \times \text{IR2113}$); which in turn lead to a relatively expensive and bulky design.

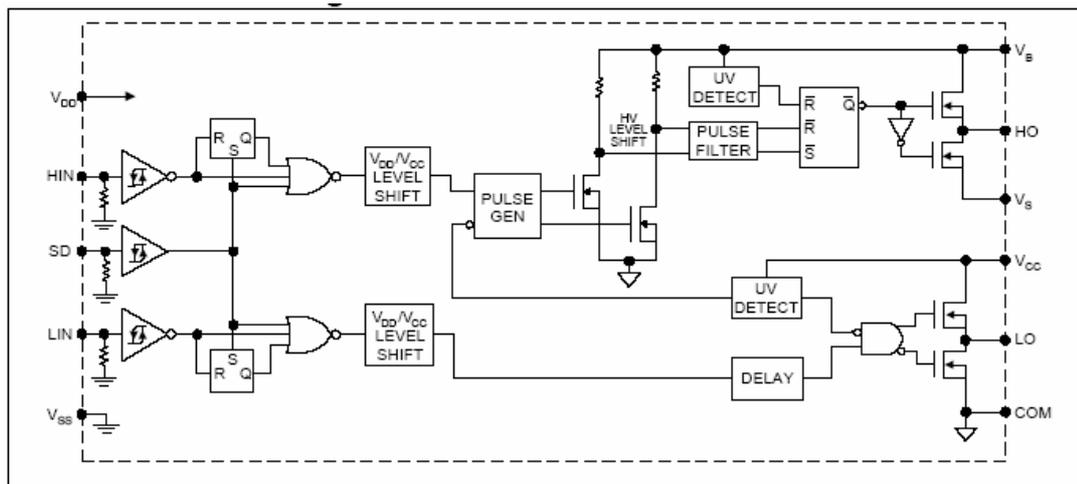


Figure 3.6 Block diagram of IR2113 [41].

To overcome the need for three isolated power supplies for the gate drivers, IR2113 can be used in *bootstrap* mode. As mentioned briefly in the previous section, in bootstrap scheme, a capacitor is charged up while the low-side MOSFET is turned on. When this MOSFET is off, the charged capacitor is electrically isolated. As high side MOSFET is turned on, the charge accumulated at this capacitor is used to drive the high-side MOSFET. Therefore, only one dedicated power-supply is sufficient to drive not only the low-side MOSFETs but also the high-side devices as well. The price to be paid here is that in this scheme, one cannot control each switch

independently. In order to keep the bootstrap capacitors charged, the low side MOSFET(s) has to be turned on for a sufficient period of time. Consequently, the high-side MOSFETs cannot be completely turned on for one PWM period.

The most important design parameter for IR2113 is the minimum value of bootstrap capacitance (C_b):

$$C_b \geq 2 \cdot \frac{[2Q_g + Q_{LS} + (I_{QBS} + I_{CBL}) \cdot T_s]}{V_{cc} - V_F - V_L - V_{\min}} \quad (3.3)$$

where

I_{QBS} \equiv Quiescent current for the high-side driver circuitry: 230×10^{-6} [A],

I_{CBL} \equiv Leakage current of the bootstrap capacitor [A],

Q_g \equiv Gate charge of MOSFET (IRF 530N) at high side: 37×10^{-9} [C],

Q_{LS} \equiv Voltage level-shift-charge (per cycle) of IR2113: 5×10^{-9} [C],

T_s \equiv Max. switching period: 250×10^{-6} [s] (4 kHz),

V_{cc} \equiv Supply voltage of IR2113: 15 [V],

V_F \equiv Forward bias voltage of bootstrap diode: 0.72 [V],

V_L \equiv Voltage drop across the low side MOSFET (IRF 530N) [V].

Some manufacturers give the leakage current for an (ESR-type) electrolytic capacitor as

$$I_{CBL} = \max\{0.002C_B V_B, 0.4\} [\mu\text{A}]$$

where V_B is the nominal (working) DC voltage of the capacitor [V]; C_B is the capacitance in [μF]. Note that the leakage current for non-electrolytic capacitor can be neglected for all practical purposes. Hence, the use non-electrolytic capacitors as bootstrap capacitors are encouraged [60]. Crunching in the number yields $C_{b,\min} > 20$ [nF]. Hence, C_b is selected as 100 nF to be on the safe side.

Finally, the only remaining element to be selected is the bootstrap diode. The manufacturer (International Rectifier) recommends the use of an ultra-fast rectifier

diode with soft-recovery regime like 11DF4 because relatively large reverse recovery currents of the bootstrap diode might damage the voltage level-shifting circuitry of IR2113. Unfortunately, since local part suppliers do not carry the specified diode, another alternative needs to be investigated. Due to low DC bus voltage (24V) used in this study, a general-purpose diode like 1N4148 seems to be adequate for the operation.

3.3.1.3 Signal Isolation and Dead-time Generation

The next design stage is to isolate the power converter from the control circuitry. Furthermore, a dead-time generator is necessary to assure that the MOSFETs at a particular converter leg are not both turned on even for a short period of time. Thus, short circuiting of the DC bus is avoided all together. Two designs are considered. In the first design, CNY-17 (or 4N38) optocouplers are used in “non-inverting” (i.e. “pull-up”) configuration to transmit gate signals to the IR2113 as illustrated in Fig. 3.7. As can be seen, an RC network with a diode is used to delay the input signal in an asymmetric fashion. That is, the RC network has two different time constants depending on input voltage. Fig. 3.8 illustrates two modes of operation for the given circuit. Here, when the photo-transistor is driven into conduction, the diode (D) becomes reverse-biased and the supply voltage is applied to the capacitor via 1k Ω resistor as shown in Fig. 3.8(a). The effective time constant in this case will turn out to be

$$\tau_1 = RC = 1000 \cdot 10 \times 10^{-9} = 10 \mu\text{s}$$

Note that the current through this resistor will flow until the capacitor voltage approximately reaches to the supply voltage. Hence, the TTL (gate) signal being applied to the photodiode will be not only level-shifted to 15 V but also delayed approximately by 10 [μs].

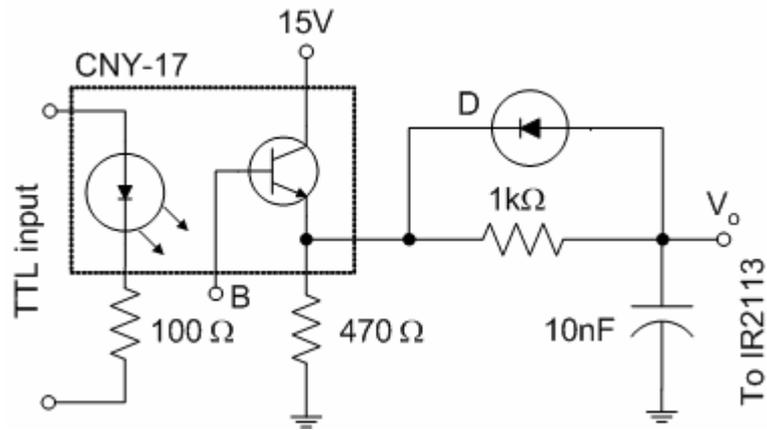


Figure 3.7 Optical isolation and dead-time generation circuitry.

Likewise, when TTL signal goes out to low, the phototransistor does not conduct current any more. As illustrated in Fig. 3.8(b), the capacitor is discharged through the forward-biased diode as well as the pull-down resistor (470Ω). Thus, the effective time constant becomes

$$\tau_2 = RC = 470 \cdot 10 \times 10^{-9} = 4.7 \mu\text{s}$$

Consequently, the gate turn-off signal will be delayed by $4.7 \mu\text{s}$. Notice that the overall design will employ two of these circuits per each converter leg as illustrated in Fig. 3.9. One of the circuits driving IR2113 is to receive the inverted gate signal as its input. That is, if one of the switches at one of the legs (say, low side) is turned off, the other (high side), which receives complementary gate signal, will turn on. Note that due to this asymmetric delay scheme, there will be theoretically a blanking period of $\Delta\tau = \tau_1 - \tau_2 = 5.3 \mu\text{s}$ where both MOSFETs are turned off simultaneously.

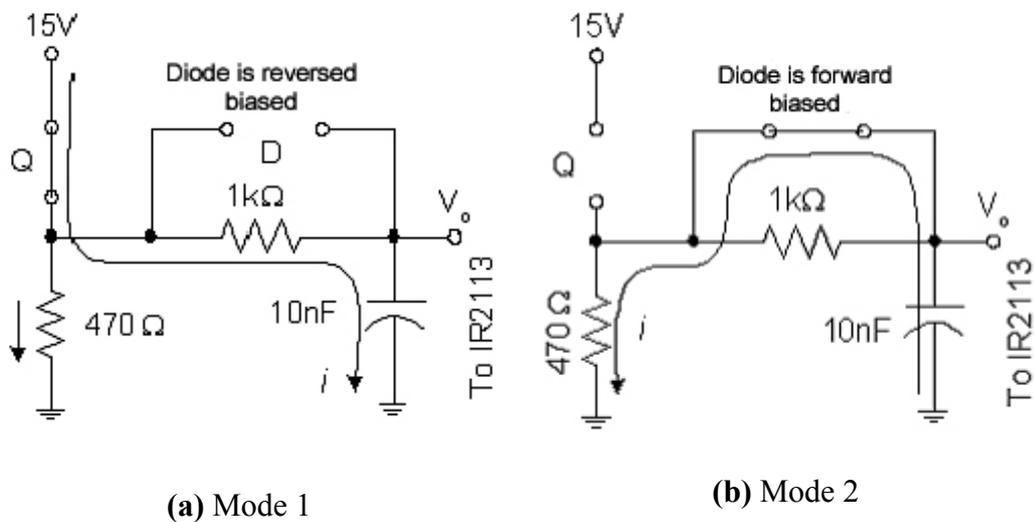


Figure 3.8 Modes of operation for dead-time generator.

It is critical to note that the RC circuit used in the dead-time generator distorts the logic signal being transmitted. However, the Schmidt trigger inside IR2113 (see Fig. 3.6) not only shapes out this distorted signal but also reduces the noise being induced during the operation.

When this circuit is tested, it has been observed that the PIC16F877 providing TTL gate control signals to the optocouplers could not drive them effectively. Contrary to its technical specifications, the micro-controller can barely supply a few mA @ 5V. Thus, an extra amplification stage is deemed necessary to obtain a crisp logic signal at the output of the optocoupler. Furthermore, the tests reveal that the CNY-17 turns out to be a very slow optocoupler with a 25 μ s turn-on delay and a 15 μ s turn-off delay which in turn harshly limits the PWM switching frequency to a few kHz. It is interesting to note that some designers claim that a maximum switching frequency of 40 kHz is attainable with CNY-17 by probably increasing the forward photodiode current beyond 50 mA which brings up the aging problem of the photodiode in the long run. Consequently, the resulting design is severely degraded by the performance of these optocouplers and thus, a major revision on the design is essential.

To overcome this problem, a fast optocoupler 6N136 is employed. The revised circuit is illustrated in Fig. 3.9. In this design, instead of sending out individual gate signals to each MOSFET, a combined signal for each leg is facilitated. For instance, when the gate input is HIGH, only the high-side MOSFET is turned on. Similarly, if it goes LOW, only the low-side MOSFET is switched on. Hence, only one optocoupler per each converter leg is needed in this scheme. To drive each optocoupler properly, a transistor array (in Darlington configuration) like ULN2003A has to be utilized. Since the transistor array behaves like an inverter, the output of the inverting 6N136 is essential non-inverted (gate) signal. Hence, a Schmidt trigger is used both to shape the distorted signal and to invert it for the proceeding operations. Notice that the principle of the dead-time generation in this revised system is very similar to that described previously. The only difference is that an inverting Schmidt trigger is used to shape and buffer the associated signals right before (and after) the RC network.

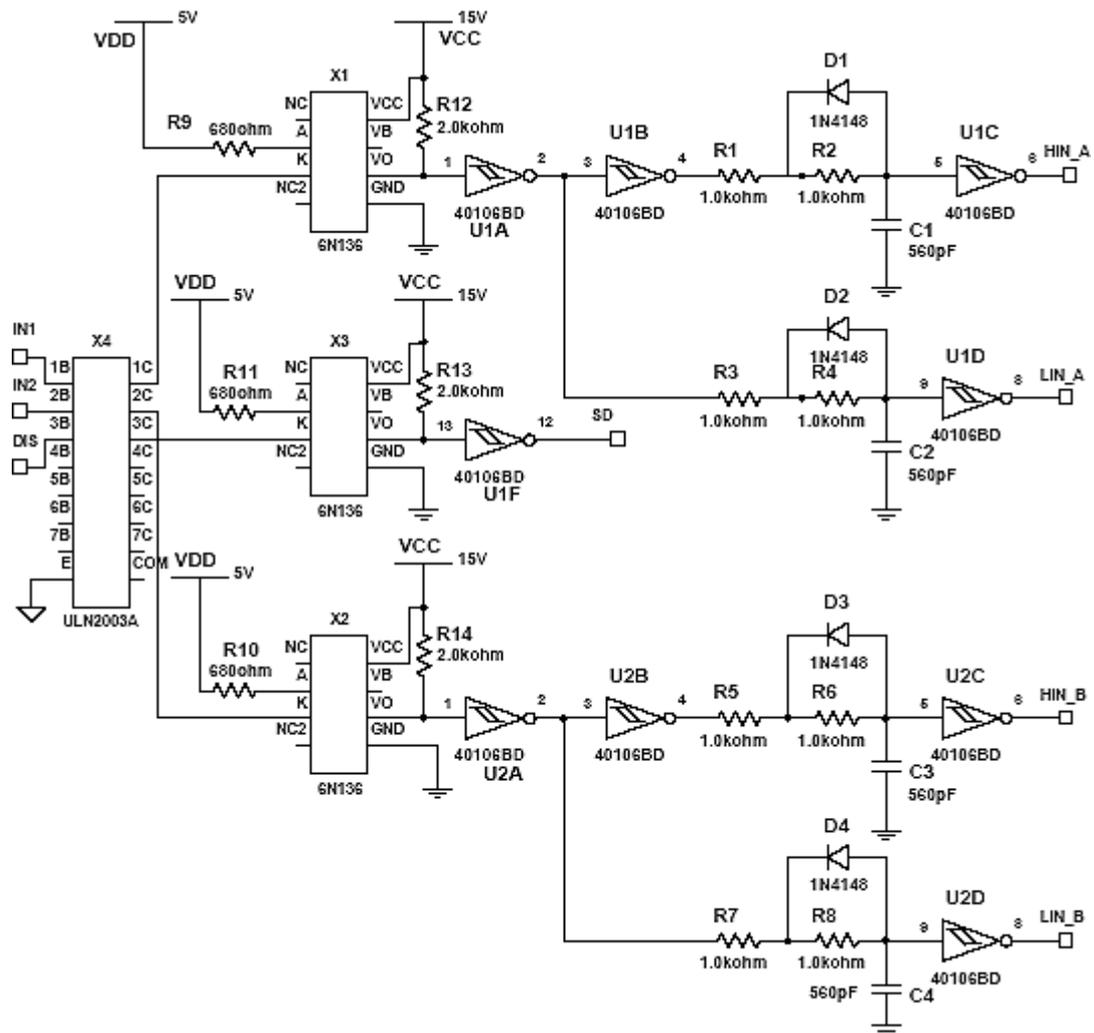


Figure 3.9 Revised optical isolation and dead-time generation circuit.

Following the thought process described earlier, the resulting dead-time of this circuit can be simply written as $\Delta\tau = \tau_1 - \tau_2 = RC = 1000 \times 560 \times 10^{-12} = 0.560 \text{ } [\mu\text{s}]$. Fig. 3.10 shows the gate signals (high and low side of a converter leg) for IR2113. As can be seen, the delay is slightly higher than the expected. This is due to the fact that the propagation delay (i.e. speed) of a typical CMOS IC (like the inverting Schmidt trigger used here) is a function of its capacitive loads as well its fan-outs. Since these factors have not been taken into account in the calculation above, this slight discrepancy in the dead-time is observed.

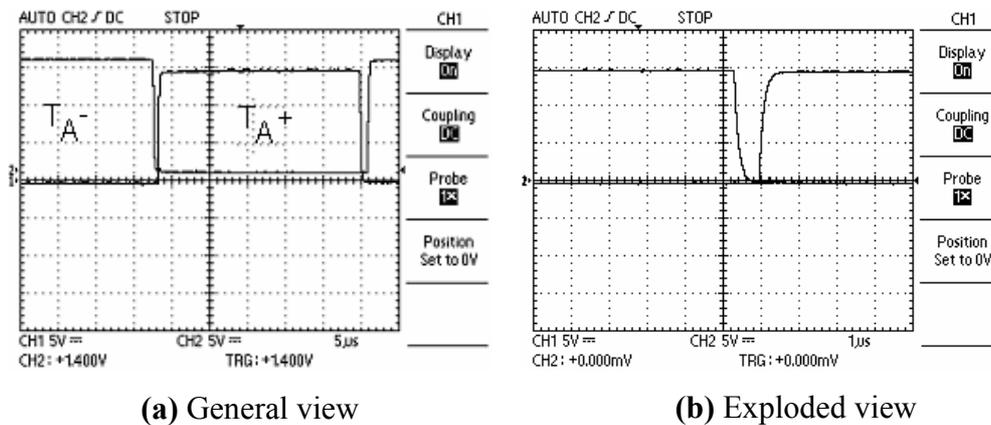


Figure 3.10 Gate signals for one inverter leg.

3.3.1.4 Overall Design of Converter with Discrete Components

In the previous sections, the design of important converter elements is elaborated. Fig. 3.11 illustrates a converter design incorporating these elements. The system, which employs CNY17-4 as its optocouplers, appears to have four independent gate control signals:

- AL is the gate signal for T_{A-}
- AH is the gate signal for T_{A+}
- BL is the gate signal for T_{B-}
- BH is the gate signal for T_{B+}

Notice that one cannot turn on and turn off T_{A-} and T_{A+} (or T_{B-} and T_{B+} for that matter) independently due to the restriction imposed by the bootstrap operation. Furthermore, one should avoid shorting the DC bus by making sure that none of these transistors are switched on at the same time. Thus, the gate control signals of lower-side transistors should be the inverted gate signals of their high-side counterparts.

Unfortunately, the performance tests showed that this converter has significant design problems involving slow optocouplers (CNY-17 or 4N38) being poorly driven by the microcontroller. From time to time, it has been observed that these optocouplers do introduce a small DC offset (which is sometimes higher than 10% of V_{DD}) at their output stage. This bias triggers IR2113 to turn on the high- and low-side MOSFETs simultaneously. Consequently, a permanent damage in the process was inflicted on IR2113, which happens to be the most expensive component of the converter.

The revised version of the converter, which apparently eliminates all these problems, is illustrated in Fig. 3.12. As elaborated in Sec. 3.2.1.3, this topology utilizes an enhanced opto-isolation plus dead-time generator stage. Like its predecessor, it accepts TTL gate control signals. However, the two TTL inputs, namely IN1 and IN2, control the operation of each converter leg:

- If IN1 is HIGH then T_{A+} is ON & T_{A-} is OFF.
- If IN1 is LOW, T_{A+} is OFF & T_{A-} is ON.
- If IN2 is HIGH then T_{B+} is ON & T_{B-} is OFF.
- If IN2 is LOW, T_{B+} is OFF & T_{B-} is ON.

With respect to the implementations of these converters, the first converter is built on a prototyping board as shown in the Fig. 3.13. As the failures on this converter raised their ugly heads, the revised version of the converter is built on a brand new breadboard as illustrated in Fig. 3.14.

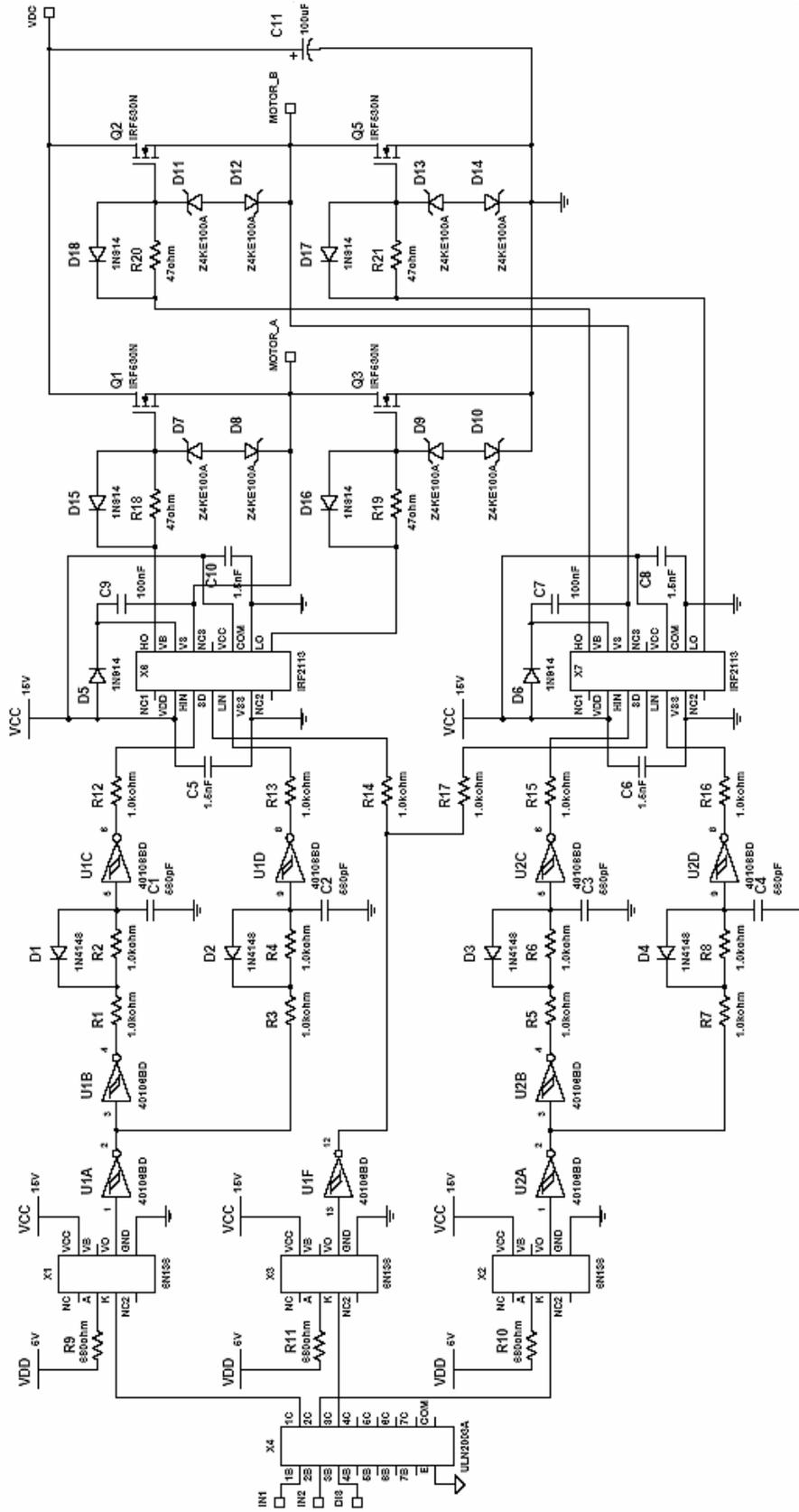


Figure 3.12 Final version of the converter A.

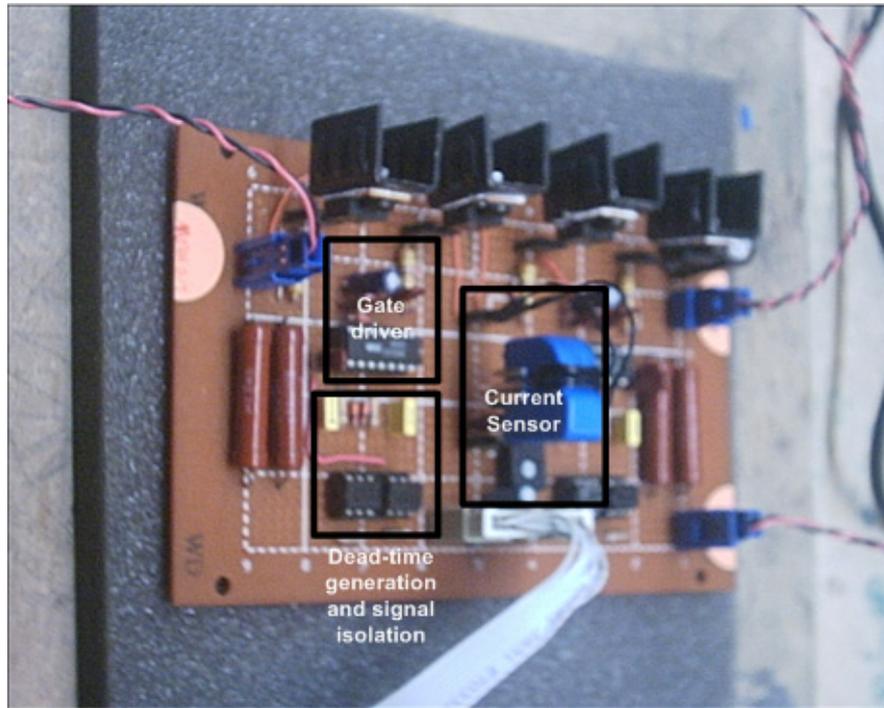


Figure 3.13 First version of converter A.

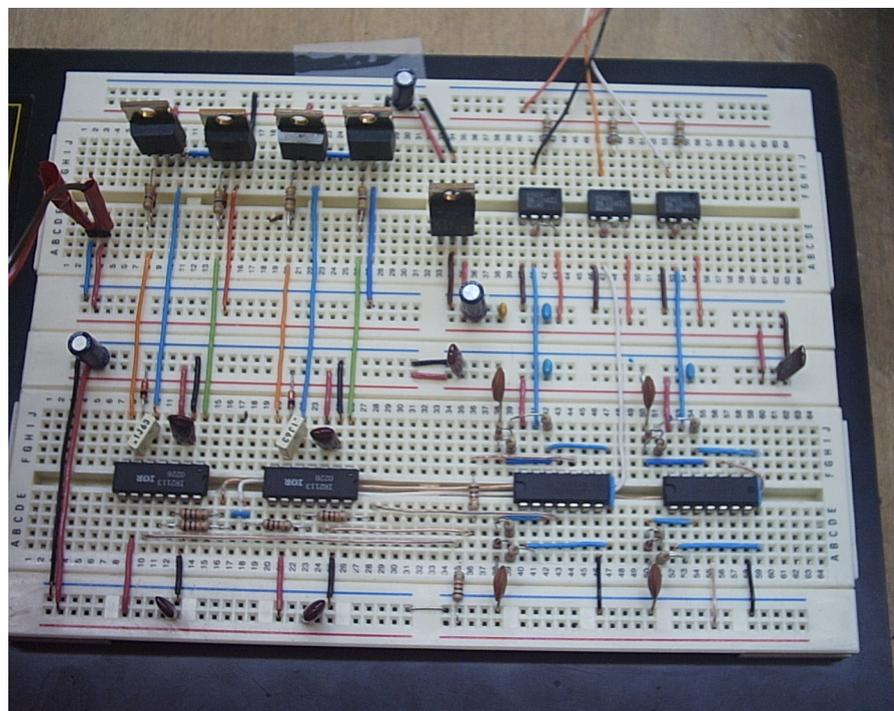


Figure 3.14 Enhanced converter A.

3.3.2 Converter with Integrated Components

The hardware complexity of the converter discussed in the previous section is quite high. It is a well-known fact that the reliability of a circuit is inversely proportional to the number of its components. Thus, to design a very reliable and cost effective DC motor driver, application specific integrated power modules (ASIPMs), which include all of the converter elements discussed previously, are also investigated in this study. There exist various H-bridge IC manufactured by various companies. Table 3.3 tabulates some of these ICs.

Table 3.3 H-Bridge ICs.

Manufacturer	IC	Features
National Semiconductor	LMD18200T	3A, H-Bridge, up to 55V
SGS Thomson (ST)	SN754410	1A, Quadruple half H-bridge, 4.5V to 33V
SGS Thomson (ST)	L9230	6A, SPI controlled H-Bridge, 5V to 28V
SGS Thomson (ST)	L298	2A, Dual H-Bridge, up to 46V
Vishay Siliconix	Si9988	0.65A ,H-Bridge, 13.2V
Zetex Semiconductors	ZXMHC6A07T8	1.8A, MOSFET H-Bridge, up to 60V, no logic
Motorola	MPC17533	0.7A, Dual H-Bridge, 2 to 6.8V
Motorola	MC33886	5.2A, H-Bridge, 5 to 30V

SGS Thomson's L298, which is a popular chip among hobbyists, is chosen because of its availability in market as well its appealing price. The chip, which is originally

intended to drive step motors, does contain two independent H-bridges consisting of BJTs in a single monolithic package as illustrated in Fig 3.15. With 46V power supply and 2A of nominal output current, the chip is quite suitable for this application.

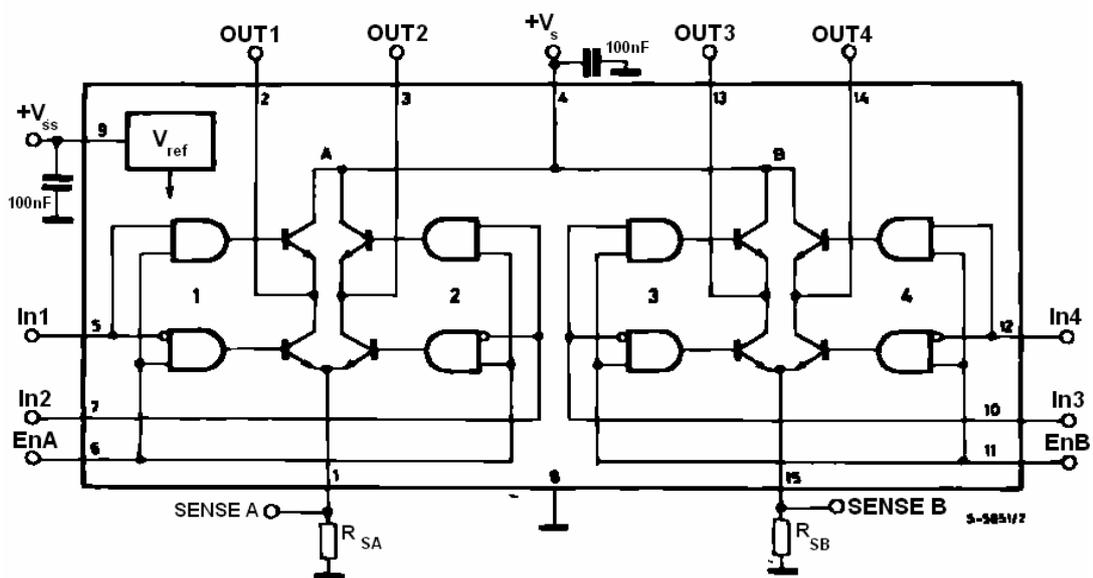


Figure 3.15 A simplified block diagram of L298.

The most important feature of this IC is that, it constitutes not only BJTs but proper gate drivers as well as isolation circuitry. The resulting system simply accepts TTL type gate control signals. As a result, the overall design is extremely simple. Even though L298 is compatible with its discrete counterpart, the switches in L298 can be fully turned on/off.

It is critical to notice that the BJTs in package do not include any anti-parallel diodes. Since DC motor is an inductive load, a current path must be provided when these transistors are not in conduction. The external diode bridge, which consists of fast recovery diodes (like BYW51), must perform this important function as

illustrated in Fig. 3.16. Likewise, the implementation of the overall converter is given in Fig. 3.17.

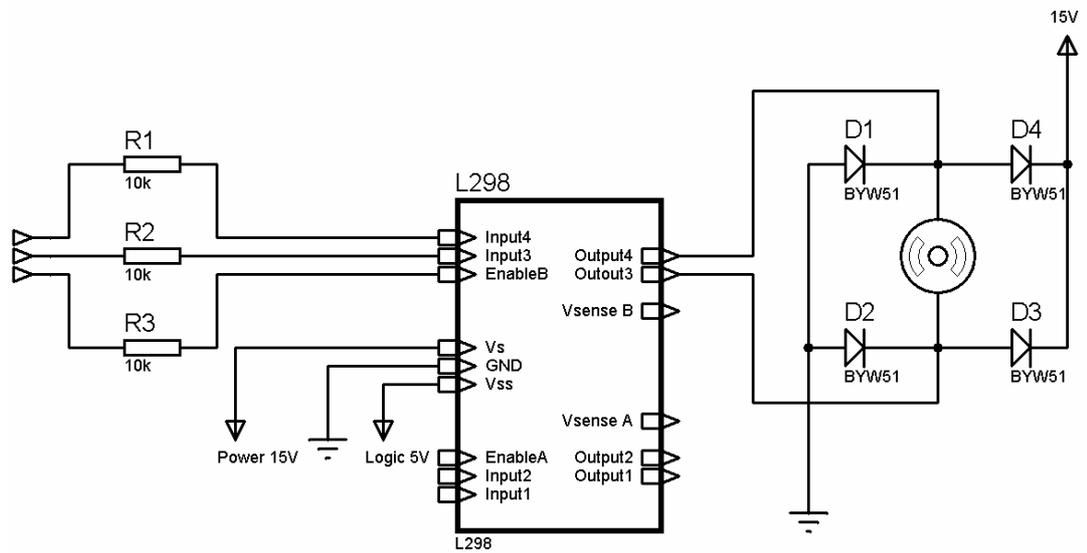


Figure 3.16 Circuitry for converter B.

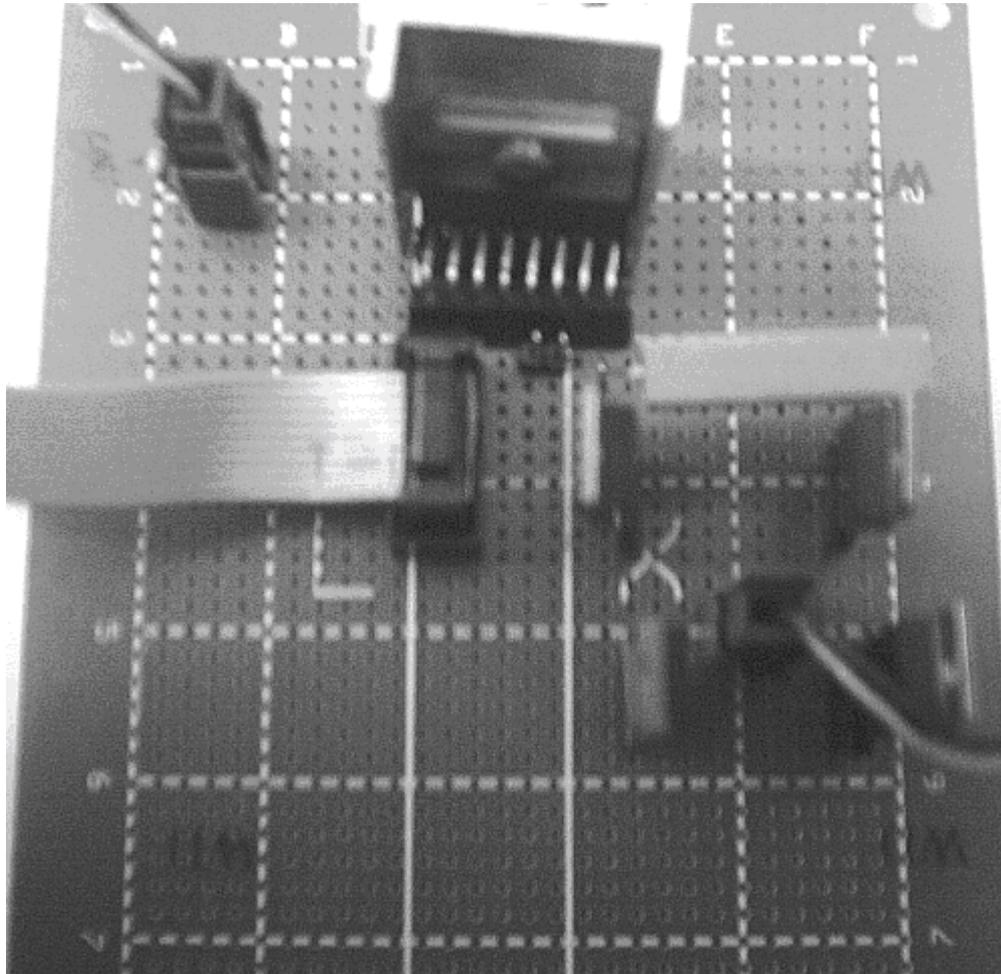


Figure 3.17 Converter topology using L298.

3.4 Voltage Regulations for DC to DC Power Converters

In order to generate the desired voltage across the terminal of the DC motor, the switching operation of a converter must be controlled accurately. Thus, the regulator's task is to generate proper *pulse width modulation* (PWM) signals in order to synthesize this voltage. Two voltage regulator schemes are considered in this research, the first one is discrete time voltage regulation, the next one is continuous time counterpart. The description of these regulator schemes follow.

3.4.1 Discrete-time Voltage Regulation

Most modern microcontrollers have specific features to control various electrical motors; these features include on board A/D converters, PWM generators with dead-time control, counters, digital I/O ports etc. Thus one can utilize such microcontrollers to control a DC to DC converter directly, via generating gate control signals through these controllers.

For this purpose; PIC 16F877, which has an 8-bit Reduced Instruction Set Computer (RISC) CPU with 14 bit instructions, is considered to control the converters designed in Section 3.3. The microcontroller has a 368 byte data memory and 8kx14 words of flash program memory with an assortment of peripherals. Some peripherals of PIC 16F877 are as follows:

- Two 8 bit Timer/Counter modules
- One 16 bit Timer/Counter module
- Two 10-bit Capture/Compare/PWM modules
- Eight 10 bit A/D conversion modules
- A Universal Synchronous Asynchronous Receiver (USART) module
- Synchronous Serial Port (SSP)
- 8 bit wide Parallel Slave Port (PSP)

As indicated above, even though PIC is an 8 bit microcontroller, some of its peripheral units, such as PWM module and Timer/Counter module, are wider than 8 bits which complicates the control of these modules. For instance, the PWM module has two registers per each PWM output. One needs to write the proper value of the duty cycle (10 bits) to the register titled CCPRxL and CCPxCON where “x” denotes the PWM channel 1 or 2 at the output pins 16 and 17 respectively. CCPRxL contains the most significant 8 bits while the remaining (least significant two) bits should be stored in the fourth and fifth bits of CCPxCON. Note that this does not really mean that the resolution of PWM duty cycle is 10 bits. In fact, the PWM resolution is a function of PWM frequency and microcontroller’s clock frequency.

Hence the expression for the maximum PWM resolution that achieved by the desired PWM frequency can be simply expressed as

$$\text{Resolution} = \frac{\log\left(\frac{f_{OSC}}{f_{PWM}}\right)}{\log(2)} \quad (3.4)$$

where

$$\begin{array}{ll} f_{osc} & \text{Clock frequency [Hz],} \\ f_{PWM} & \text{Desired PWM frequency [Hz].} \end{array}$$

PWM frequency is specified by setting the control register T2CON associated with Timer2 peripheral of the microcontroller as well as the period register PR2. The corresponding PWM period can be simply calculated as

$$\text{PWM Period} = [(PR2)+1] \times 4 \times T_{OSC} \times \text{pre-scale} \quad (3.5)$$

where pre-scale (1, 4, 16) written in T2CON is used to divide the clock frequency effectively by that value. Once a PWM frequency is selected with a proper pre-scaler, one can solve for PR2 value (one byte). As an example in this application,

- clock frequency $f_{osc} = 4 \text{ MHz}$
- PWM frequency $f_{PWM} = 4 \text{ kHz}$
- pre-scale is selected as 1

Therefore using equation (3.6), PR2 register value becomes \$FF (255).

Notice that the maximum PWM frequency with 10 bit resolution is found to be 4 kHz from (3.4) [52]. The PWM frequency could be increased by selecting higher clock frequency for PIC. This could be simply accomplished by using a 20 MHz quartz crystal at the oscillator of the microcontroller. Note that the PIC microcontroller, which is initially built on a prototyping board by soldering, is to operate at a low clock frequency of 4MHz. Such a medium is not suitable for high frequency operations due to parasitic effects introduced by that environment. Therefore, 20 MHz clock frequency is avoided due to the difficulties in the implementation.

Firmware for discrete-time voltage regulator simply determines the duty cycle value ranging between 0 and 1023 and sets the duty cycle value into each PWM duty variables. Then, the duty cycle value is written to PWM registers according to the desired direction (i.e. converter leg) where PWM1 output drives one converter leg while PWM2 is to drive the other. Hence, PWM1 and PWM2 signals are to be directly connected to the gate control inputs of the converter IN1 and IN2 respectively. Program flow chart is given at Fig. 3.18.

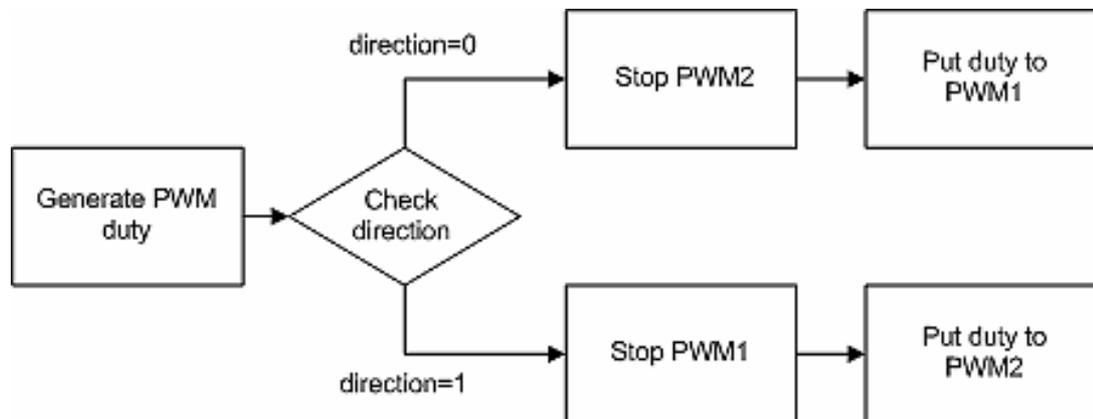


Figure 3.18 PIC program to control the converter.

Unfortunately due to restrictions imposed on the clock frequency of the PIC, maximum PWM frequency is severely limited if 10 bit PWM duty cycle resolution is desired. Continuous-time PWM controllers which can operate at very high PWM frequencies 20 to 300 kHz can overcome these limitations imposed by limited PWM frequency generated by the discrete-time voltage regulator. The description of a high performance continuous-time voltage regulator follows.

3.4.2 Continuous-time Voltage Regulator

In recent years high performance PWM control ICs such as (LM3524, TL494, TL598, SG2525, etc.) which include many desirable features have become widely available in market. Some of these chips have the following desirable PWM control features:

- a saw tooth generator/oscillator,
- integrated op amp and comparator,
- dead time generator,
- reference voltage generator,
- direction steering flip flop for push pull operation, etc.

The continuous time voltage regulator considered here, takes advantage of PWM control chip TL494. The block diagram of a unipolar voltage regulator is illustrated in Fig. 3.19. As shown in the block diagram, the voltage command V^* ranges in between $-2V$ to $2V$. The absolute value of this voltage command has to be calculated to have a compatible signal with respect to the saw tooth signal generated by TL494. In the meantime, the sign of the voltage command has to be determined with the utilization of a simple comparator. The absolute value of the voltage command is then compared to the saw tooth and the PWM output signal is generated using the following thought process:

- If $|V^*| - V_{st} \geq 0$ then PWM output is high
- If $|V^*| - V_{st} < 0$ then PWM output is low

Hence, using this PWM output as well as the direction signal, one can control the operation of any full bridge converter. The gate control signal of the first converter leg (IN1), which is TTL compatible, can be simply obtained by applying logical AND operator to the PWM output signal and the direction signal. Likewise, the gate control signal of the other converter leg (IN2) can be obtained in a similar fashion, except that complimentary (inverted) direction signal is facilitated. Therefore, the direction signal simply selects the converter leg for pulse width modulation, while turning the low side transistor on at the remaining converter leg.

Fig. 3.20 shows the implementation of voltage regulator which follows the idea outlined above.

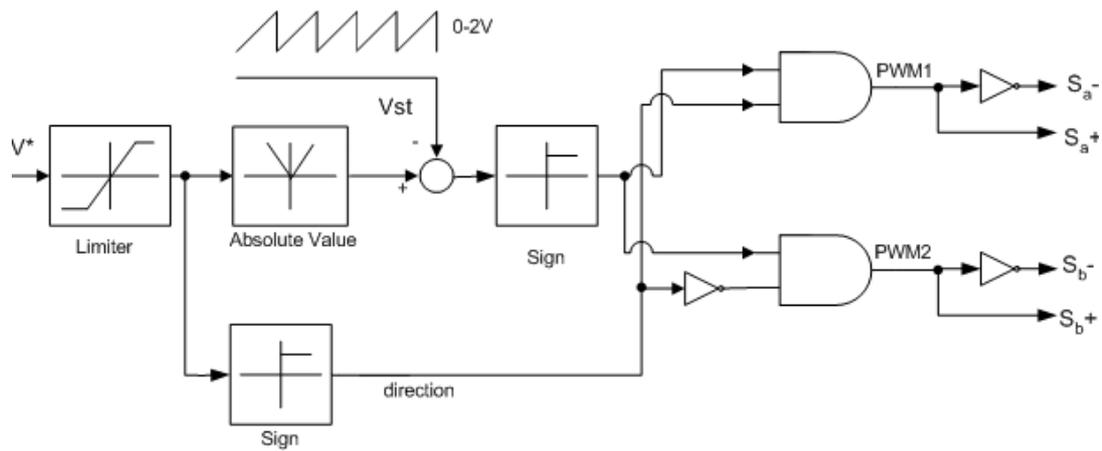


Figure 3.19 Block diagram of controller for voltage regulator B.

In this design, TL 494 is basically used as a saw tooth generator. TL494 has a high precision internal oscillator, where the internal oscillation frequency is determined by a resistance and capacitance value, R_T and C_T :

$$F_{st} = \frac{1}{R_T C_T} \quad (3.6)$$

Hence, the PWM frequency is selected as 50 kHz, by setting above mentioned parameters as $2k\Omega$ and $10nF$. Notice that TL494 constitutes an internal PWM comparator, however this internal comparator has not been utilized in this circuitry shown in Fig. 3.20. This is due to the fact that TL494 incorporates (an intentional) small DC bias on the input signal (error or command) so that the maximum and the minimum PWM duty cycles are set as 3% and 97% respectively. As a result the modulated output signal may have unwanted DC components at unipolar switching scheme. The corresponding implementation is shown in Fig. 4.12 at the next chapter.

Next section discusses the results of performance tests of the voltage regulators on the designed converters.

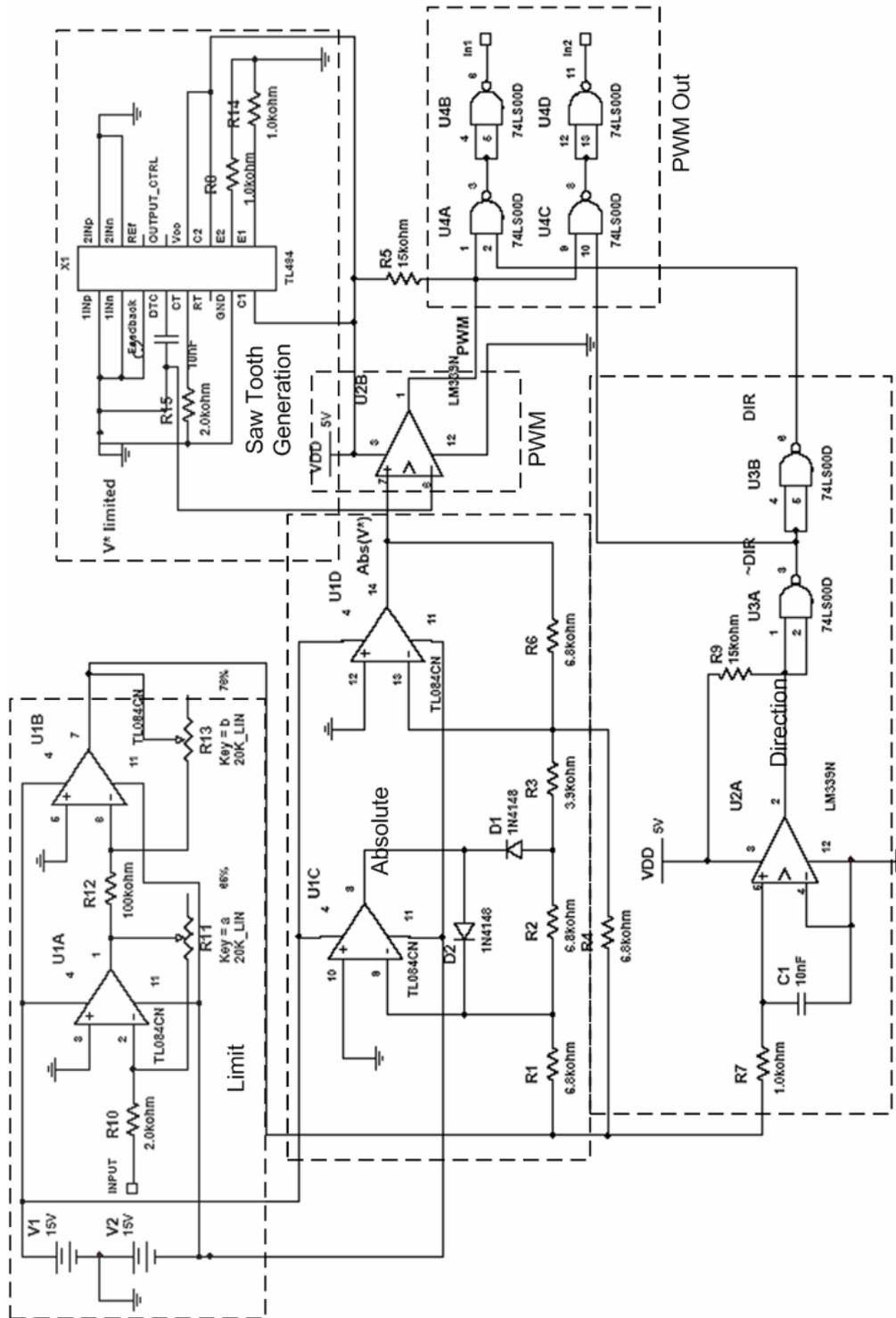


Figure 3.20 Controller for Voltage regulator B.

3.5 Performance Tests

The following section tests the performance of discrete time voltage regulator on both converters designed.

Performance tests are conducted in order to evaluate the performance of voltage regulators and converters. In order to test the performance of voltage converters a series of experiments were conducted by utilizing PIC16F877 that generates PWM signals at 4 kHz and continuous-time voltage controller that generates PWM signals at 50 kHz. Unipolar as well as bipolar switching performance of converter A and converter B are tested.

In the test employing PIC16F877, a sine wave of 50 Hz is generated by the programmable function generator (analog signal), this relatively low frequency signal, which does not effect the performance of the on board A/D converter is used for sampling the voltage command at every 1 ms. Note that it has been later on documented that on board A/D conversion is troublesome at frequencies higher than 100 Hz. The converted digital command signal, which is represented by 10 bits, ranges in between 0-1023 decimal. In bipolar switching scheme this command is directly transferred to one of the PWM duty cycle registers hence only one PWM is generated by the microcontroller. This signal is directly applied to gate control signal of one of the converter leg (IN1) and the inverted PWM signal is connected to the gate control signal (IN2). Hence, from the implementation point of view bipolar switching scheme is the simplest.

With respect to unipolar switching two PWM output signals have to be generated by the microcontroller. To accomplish that the voltage command is compared to 512, if the command is bigger than that value PWM2 output is disabled (i.e. pull down to low) else if the command is less than 512 PWM1 output is disabled. The

duty cycle of the enabled PWM output is simply calculated as $2 \times |\text{command} - 512|$. The results are presented in Figs 3.21 and 3.22.

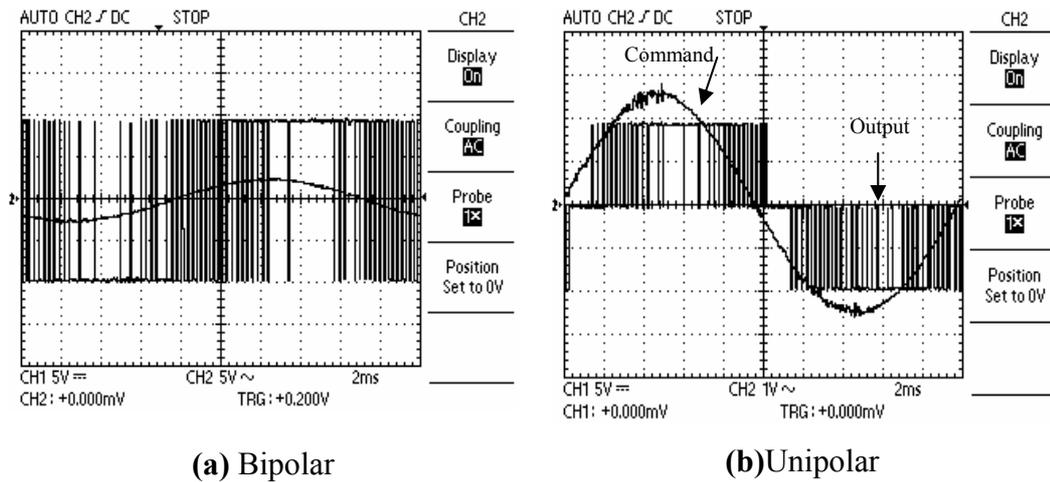


Figure 3.21 Voltage regulation on converter A.

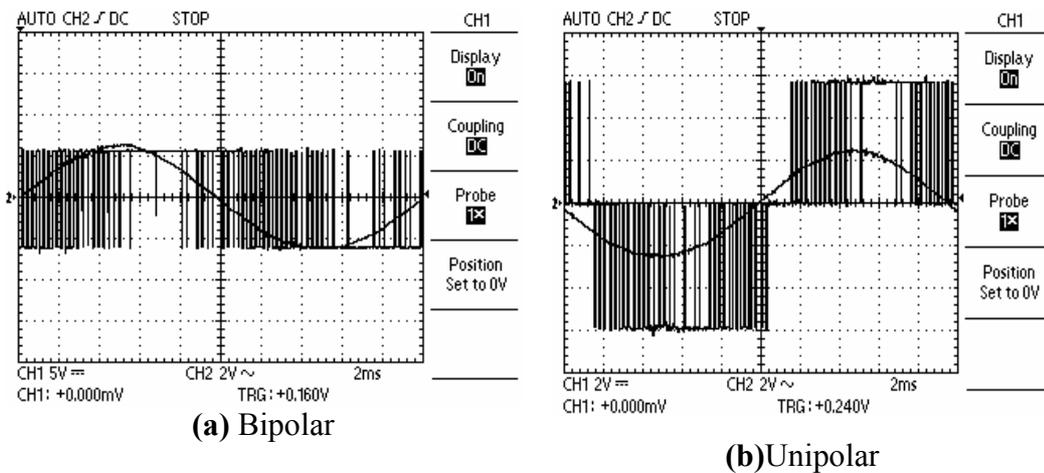


Figure 3.22 Voltage regulation on converter B.

As can be seen, both converters being tested perform extremely well and the desired output voltage is mainly synthesized. However, one can observe a small lag between the commanded signal and the generated output voltage. The main reason of this lag is associated with the firmware which samples the A/D converter at 1 ms thus have a delay of 4 PWM periods, which can be easily seen in all figures. Also a detailed view of delay is given at Fig. 3.23

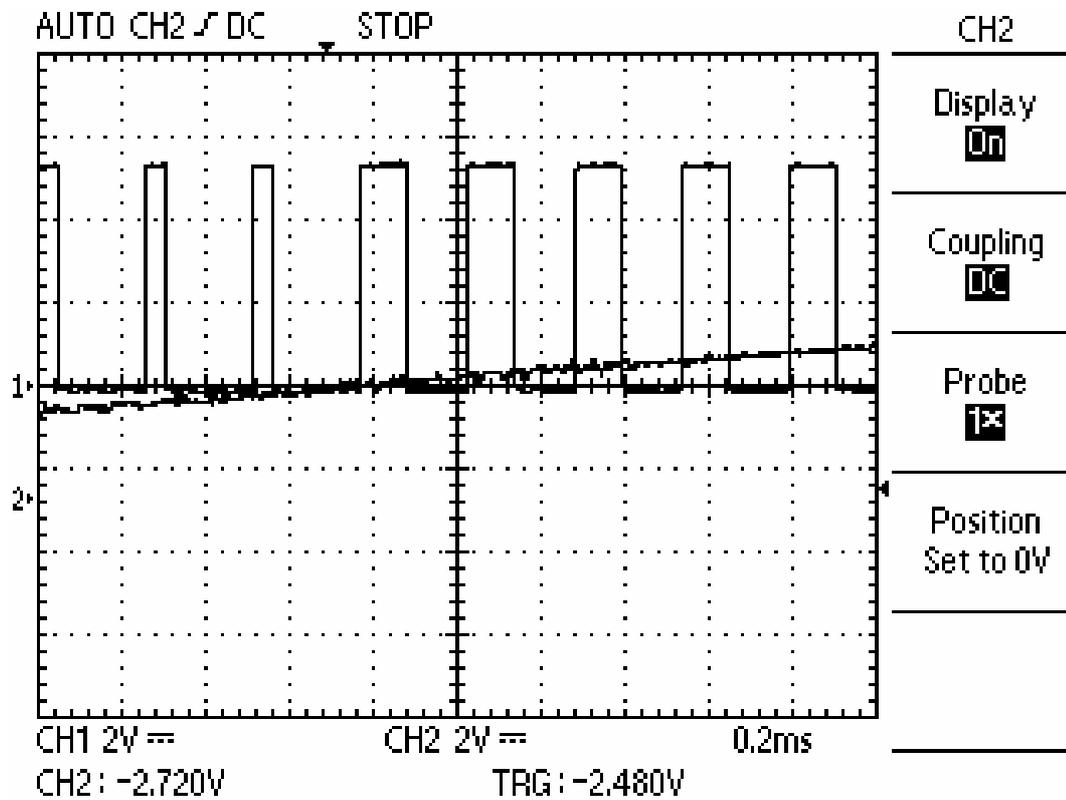


Figure 3.23 Detailed delay observed on voltage regulator B.

The test of converters employing continuous-time voltage regulator is conducted using signal generator also. The signal generator generated 50 Hz sine signal and this signal is directly fed to the voltage controller. The results of these tests are given at Figs. 3.24 and 3.25

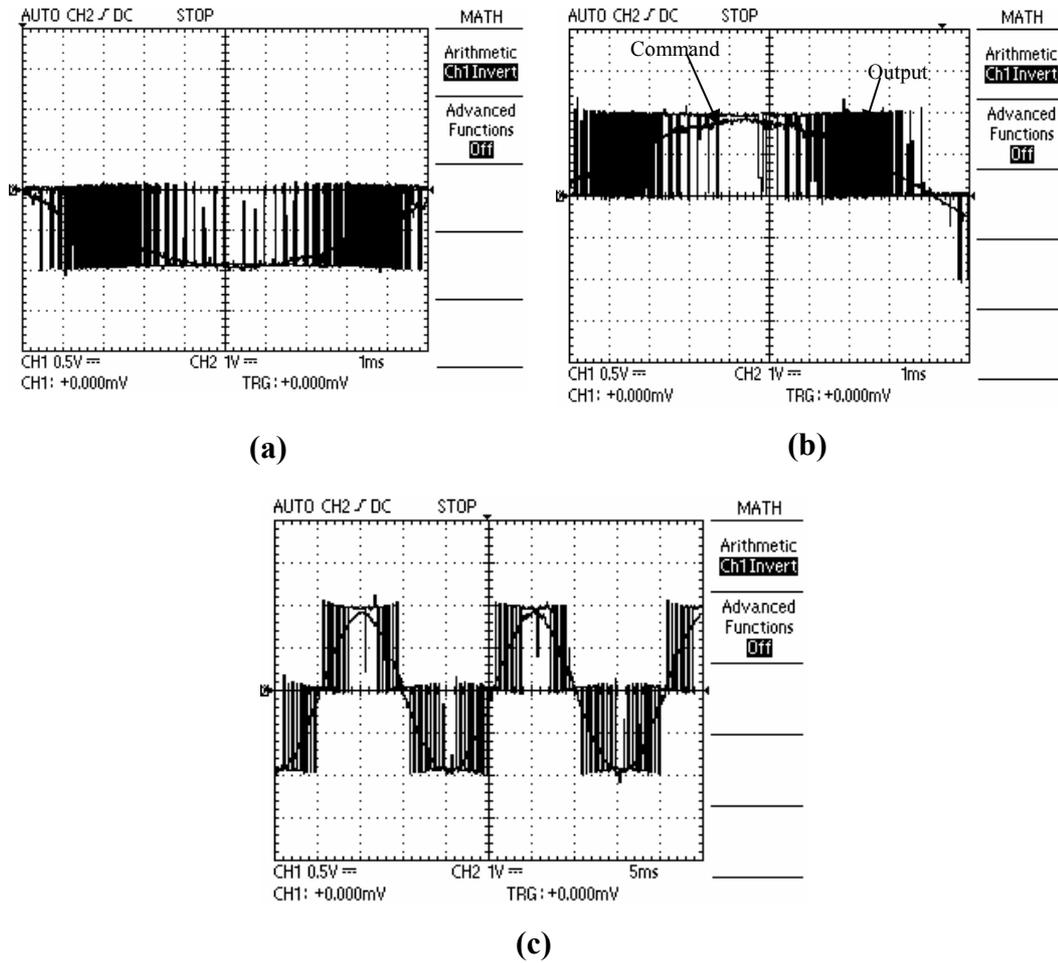


Figure 3.24 Voltage regulation on converter A.

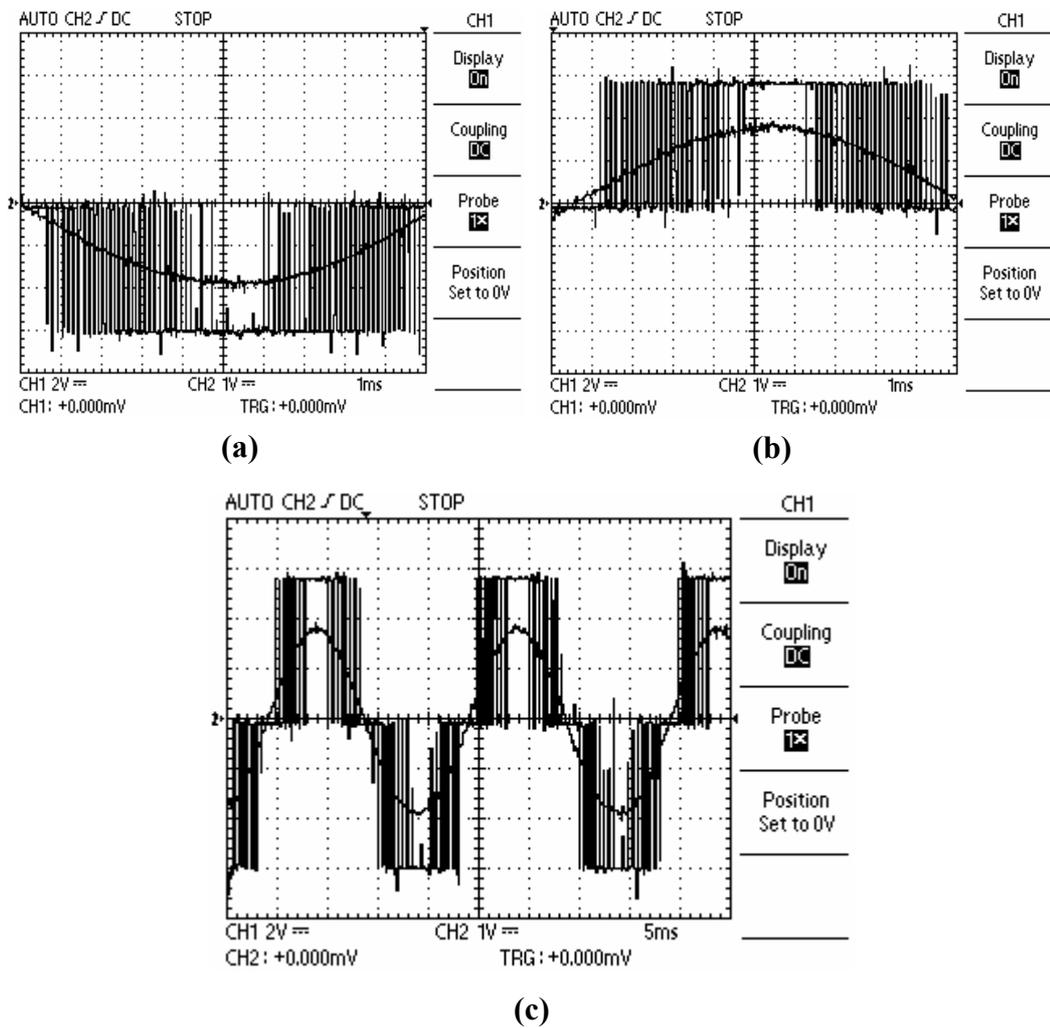


Figure 3.25 Voltage regulation on converter B.

3.6 Conclusion

In this chapter, two DC to DC converters for a DC motor drive have been designed. One of the converters (converter A) is implemented using discrete circuit components. The other one (converter B) employs an integrated H-bridge chip (L298). Similarly two voltage regulation schemes on these converters are designed:

- A discrete-time voltage regulator operating at a PWM frequency of 4 kHz
- A continuous-time regulator operating at a PWM frequency of 50 kHz.

The performance of discrete-time voltage regulator is tested on both converter topologies, despite a small lag between the command and the modulated output voltage, all converters performed well in these performance tests. As a result, these converters are all found adequate for DC motor control.

CHAPTER 4

CURRENT REGULATOR

4.1 Introduction

Current regulation is the first step in the motion control of a DC motor. From the standpoint of control engineering, current control helps not only stabilize the overall motion control system conveniently but also improve disturbance rejection property of the controlled system as discussed in Chapter 2.

In fact, torque developed by a conventional DC motor is directly proportional to the current drawn by the motor. Thus, controlling current flow enables the modulation of the torque. To control the current of a DC motor, one needs to take a look at its electrical model:

$$V_a = L_a \frac{di}{ds} + iR_a + k_e \omega \quad (4.1)$$

$$T_m = K_t i \quad (4.2)$$

where

L_a	:	Armature inductance [H],
R_a	:	Armature resistance [Ω],
T_m	:	Torque developed by motor [Nm],
V_a	:	Terminal voltage [V],
i	:	Current drawn by the motor [A],
K_e	:	Back emf constant [V/rad/s],
K_t	:	Torque constant [Nm/A],
ω	:	Motor's angular velocity [rad/s].

Thus, taking the Laplace transforms of (4.1) and (4.2) yields the desired transfer function of the DC motor:

$$\frac{I(s)}{V(s)} = \frac{1}{L_a s + R} \quad (4.3)$$

Here, $\omega(s)$ (or back EMF voltage) is treated as a disturbance input to the system for the sake of convenience even though the motor speed is measured in this study.

Two current controllers shall be examined in this chapter. The first one is continuous-time current regulator (a.k.a. analog) as shown in Fig. 4.1. Similarly, a discrete-time counterpart is illustrated in Fig. 4.2. Note that both current regulators rely on a high-bandwidth voltage regulator which is capable of delivering the desired voltage at the terminals of the motor such as those discussed in Chapter 3. Since the switching frequency (i.e. PWM frequency) of modern DC/DC converters are usually quite high (20 kHz up to 1MHz), the voltage regulator can be assumed ideal (i.e. infinite bandwidth) and could be represented by a simple gain for all practical purposes.

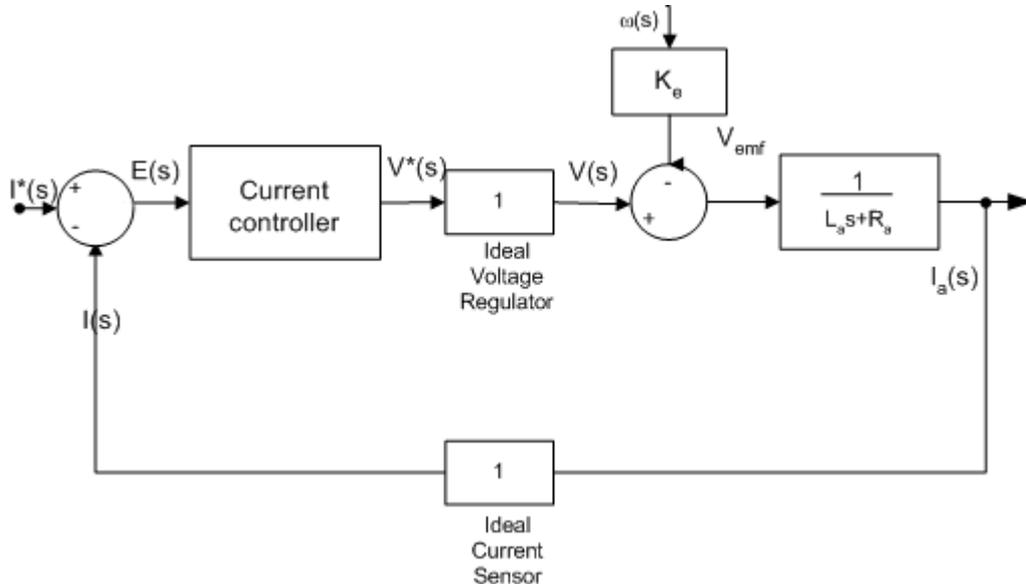


Figure 4.1 Continuous-time current regulator.

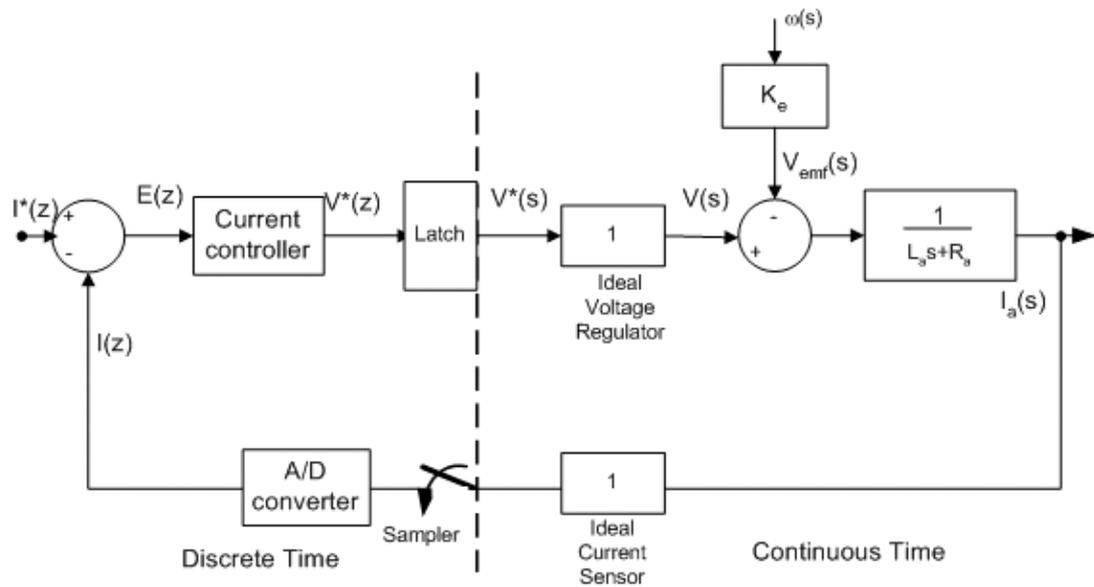


Figure 4.2 Discrete-time current regulator.

First of all, the relevant motor parameters such as L_a , R_a and K_e are required to design these current regulators. A series of experiments were conducted to determine these unknown parameters. Details of experiments and methodology can be found at Appendix A. Parameter values found are:

$$\begin{aligned} L_a & : & 14.7 \text{ mH} \\ R_a & : & 2.5 \text{ } \Omega \\ K_e & : & 0.036 \text{ V/rad} \end{aligned}$$

Armed with this information, the next section concentrates on the design of these regulators.

4.2 Current Regulator Design

At the initial step of the current regulator design, a control engineer needs to determine the current loop's bandwidth which in turn imposes restrictions on the locus of the dominant pole for the controlled system. Current loop bandwidths of modern electrical drive systems are usually in the range of 1 kHz to 10 kHz. Since the drive system to be designed in this study will be at the lower-end of the high

performance electrical drives, an overall current bandwidth of 1 kHz seems to be an attainable objective. With this design constraint in mind, the following sections describe the design of these regulators.

4.2.1 Continuous-time Current Regulator

Substituting found L_a and R values to equation gives current plant as:

$$G_{PC}(s) = \frac{I(s)}{V(s)} = \frac{1}{0.014s + 2.5} \quad (4.4)$$

In order to understand behavior of the system root locus plot and Bode plot of plant is drawn. These plots are given in Figs. 4.3, and 4.3.

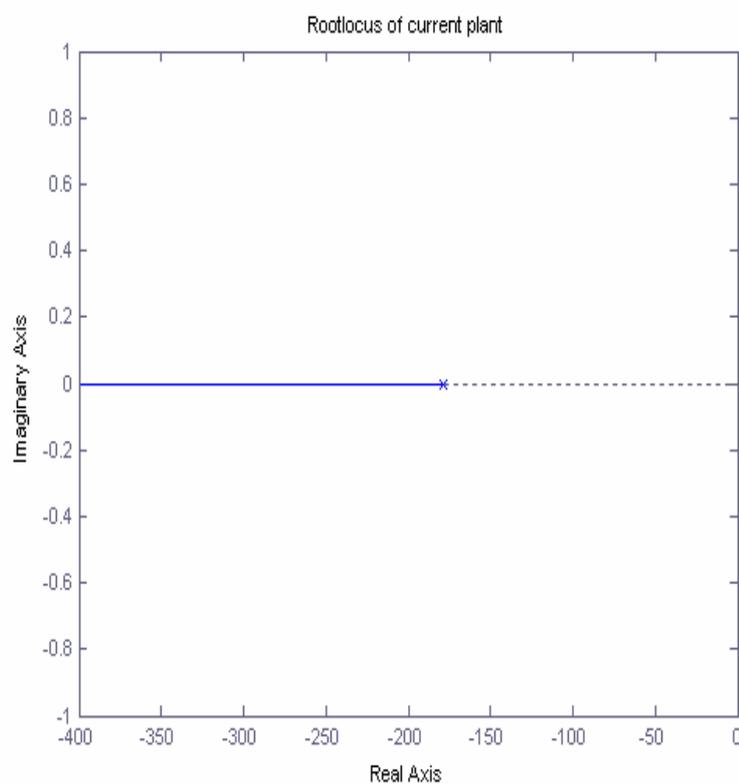


Figure 4.3 Root locus plot of current plant.

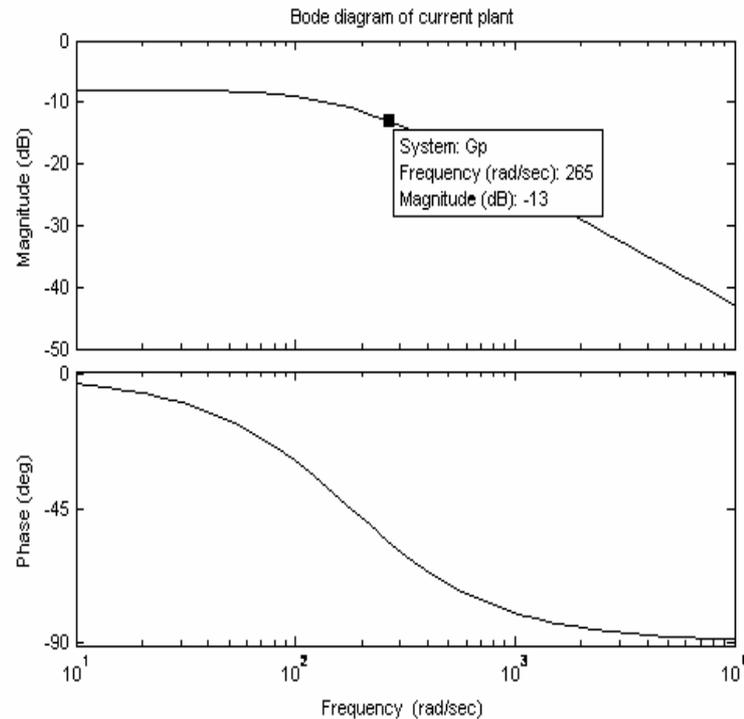


Figure 4.4 Bode plot of current plant.

By examining these plots, one can see that an open loop pole is located at -178 [rad/s] which shows that open-loop system is stable as expected. However, the cut-off frequency of the system is about 265 rad/s which is way below the specification.

In order to fulfill above mentioned requirement, a proportional (P) controller with a gain 50 could be employed. As can be seen from root locus plot, a much higher gain could have been chosen. However, such gains usually tend to aggravate the switching noise in the current which in turn deteriorate the overall control performance of the regulator. Compensated system's root locus plot along with Bode diagram is given in Figs. 4.5 and 4.6. As can be seen, the closed loop system has a dominant pole at -3750 [rad/s] and new bandwidth frequency is 500 Hz which is better than the design specification.

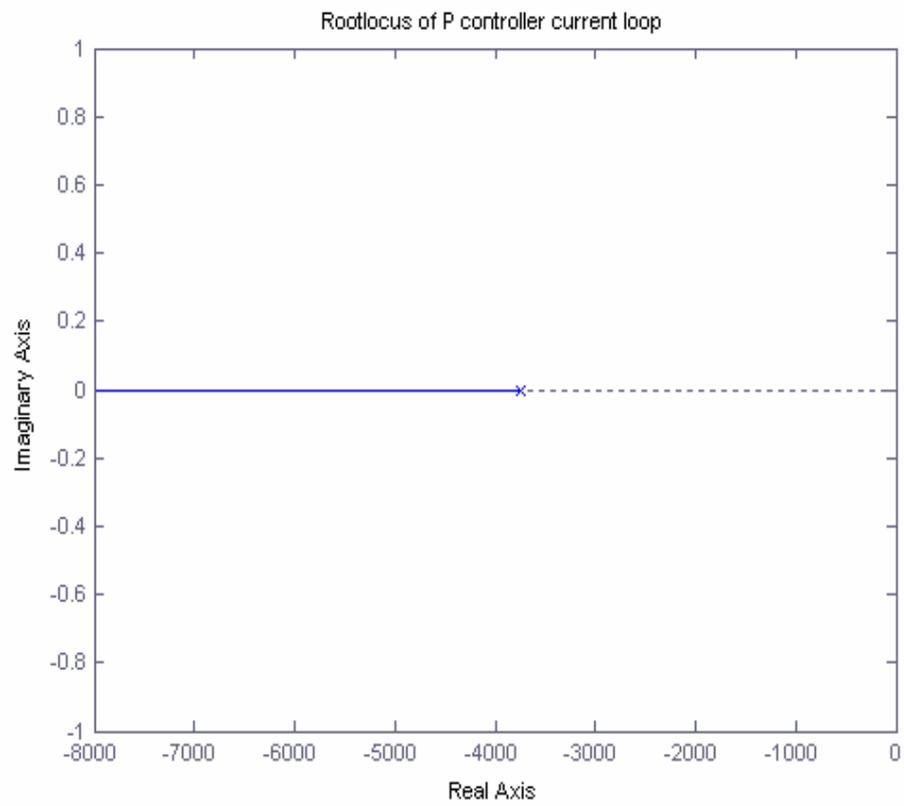


Figure 4.5 Root locus plot of compensated system.

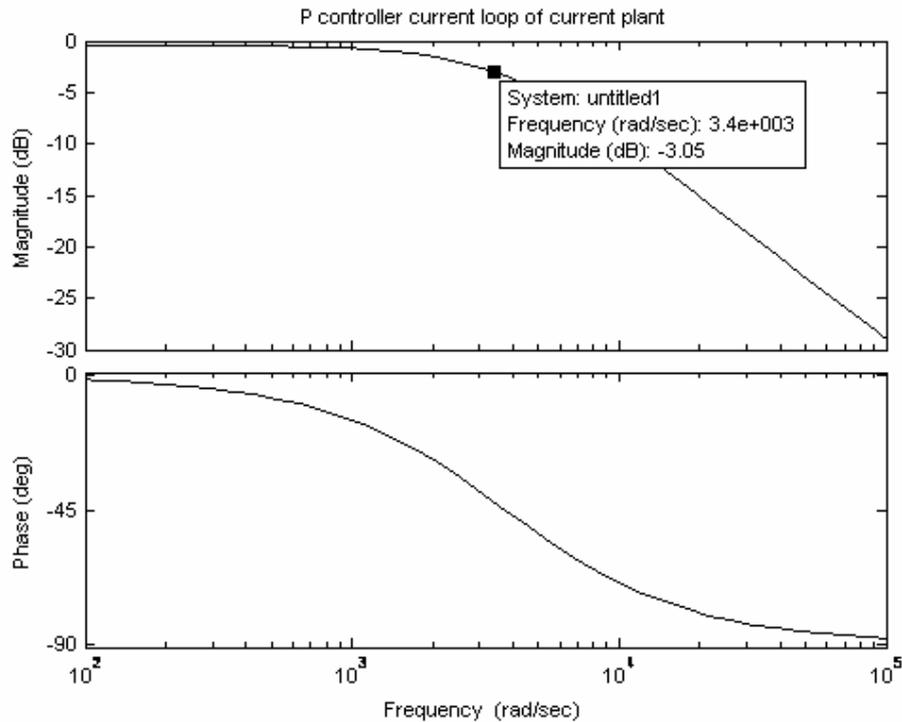


Figure 4.6 Bode plot of compensated system.

4.2.2 Discrete-time Current Regulator

In order to design a discrete-time current regulator, the plant transfer function in continuous-time should be converted to a discrete one. Since the switching frequency of the voltage regulator (A) is 4 kHz, the sampling period of the current loop should be much lower than this frequency. Thus, the sampling frequency is chosen as 1 kHz which is smaller than the PWM frequency by 4 folds. The resulting discrete time model with zero order hold (ZOH) becomes

$$G_{PC}(z) = \frac{I(z)}{V(z)} = \frac{0.06541}{z - 0.8365} \quad (4.5)$$

The discrete system root locus is illustrated in Fig. 4.7 while Bode diagram of the system is plotted in Fig. 4.8.

Examining root locus of the system shows that the open loop pole is at 0.8365. Just like its continuous-time counterpart, the system is stable and has a break frequency at 226 rad/s. As stated previously at Section 4.2, the required bandwidth of system is 3000 rad/s (470 Hz) so a compensator is needed to satisfy the bandwidth requirement.

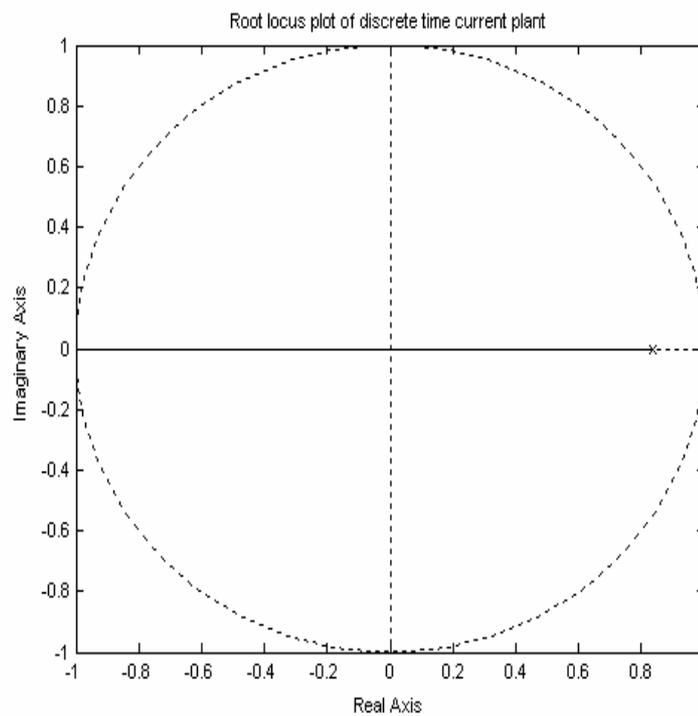


Figure 4.7 Root locus plot of discrete-time current plant.

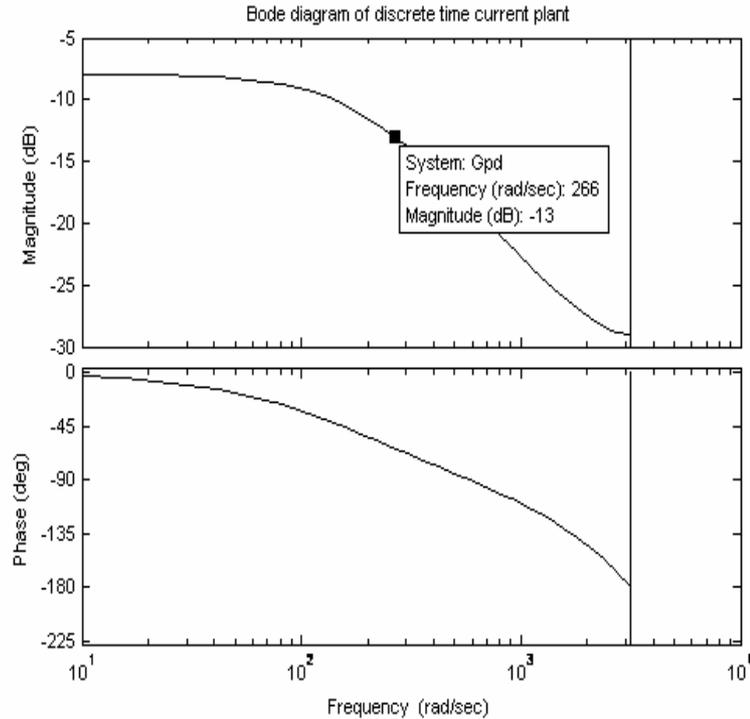


Figure 4.8 Bode plot of discrete time current plant.

Hence, a discrete-time P controller with a gain of 12 is considered. It should be noted that a gain value of 12.8 gives a closed loop pole right at the origin of the unit circle. However, if the closed loop pole enters the left-hand side of the unit circle due to parameter variation in plant, the system will go into the forced oscillations at half the sampling frequency (500 Hz). As a precaution, the proportional gain is chosen as 12 which gives a bandwidth frequency of 3000 rad/s and places closed loop pole at 0.0515 as shown in Fig. 4.9 and 4.10 in order to have a controller similar to a dead-beat controller. Next section describes the implementation of all these current regulators.

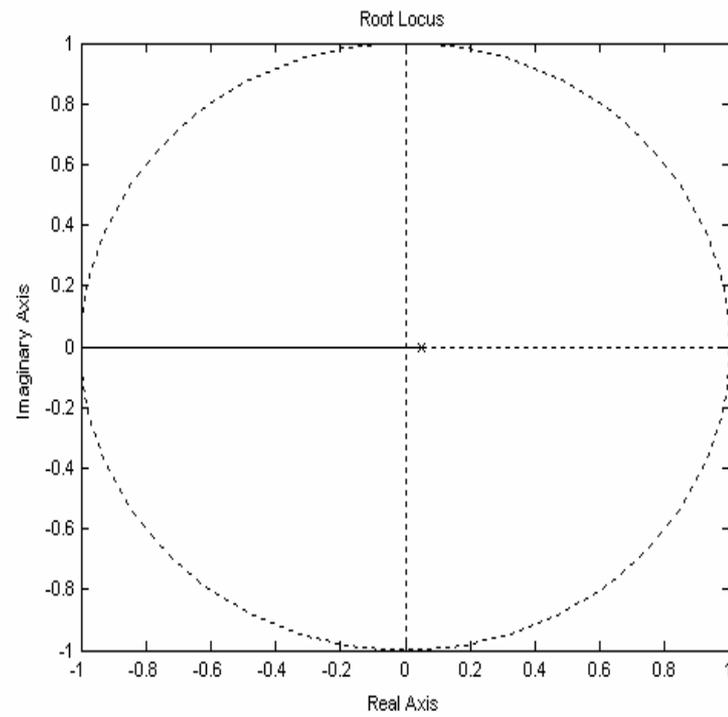


Figure 4.9 Root locus plot of compensated discrete-time system.

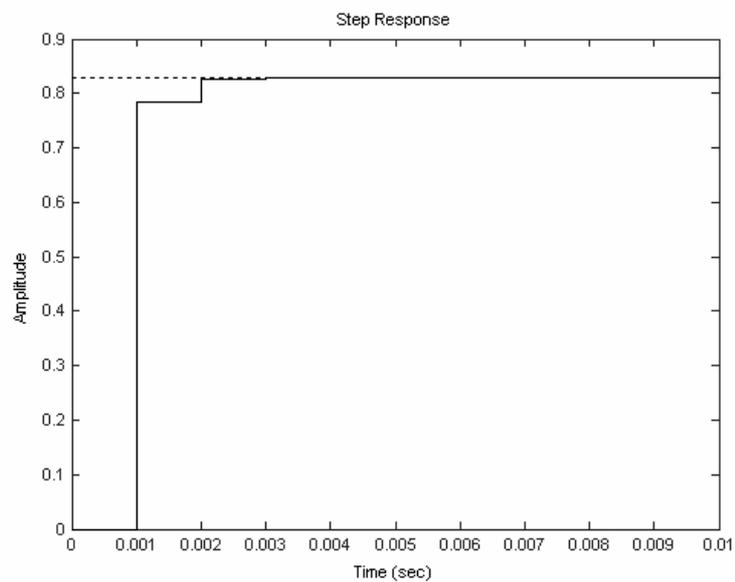


Figure 4.10 Step response of compensated discrete-time system.

4.3 Hardware Implementation

In this section, implementation of current regulators designed at Section 4.2 will be discussed. Hardware limitations concerning controller implementations are to be investigated in detail. The following section concentrates on the current sensing issues.

4.3.1 Current Sensors

Three most popular current sensors in motor control applications are:

- Shunt resistors
- Hall effect sensors
- Current transformers

Shunt resistors are popular current sensors because they provide an accurate measurement at a low cost. Hall effect current sensors are widely used because they provide a non-intrusive measurement and are available in a small IC package that combines the sensor and signal-conditioning circuit. Current-sensing transformers are also a popular sensor technology, especially in high-current or AC line-monitoring applications. A summary of the advantages and disadvantages of each of the current sensors is provided at Appendix C. A Hall effect sensor will be used in order to have isolation on current sensor.

There are various Hall effect sensor manufacturers in the industry. At table 4.1 a list of manufacturers and sensors is given.

Table 4.1 Popular Hall effect sensors.

Manufacturer	Sensor
Allegro Micro systems inc	A1321/2/3
Micronas Semiconductor Holding AG	HAL556, HAL560, HAL566
LEM Inc.	LTS25-NP/SP10

LEM LTS25-NP/SP10 is chosen as current sensor. LEM is a closed-loop compensated multi-range current transducer. Current can be measured in two

different ways, by using current sense pins (especially for PCB applications) and by passing a wire through current sensing hole. It has a range of -100A to 100A and an output of 0-5V. However recommended nominal current (I_{PN}) is -25A to 25A between these values output linearity is guaranteed with an error of 0.1%, and output ranges between $2.5V \pm 0.625V$. Fig 4.11 shows LEM and Fig. 4.12 shows current versus output voltage characteristics.

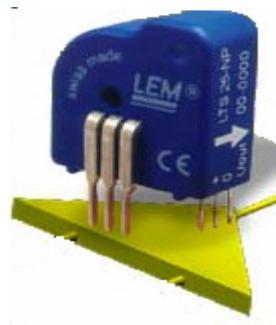


Figure 4.11 LEM Hall effect sensor.

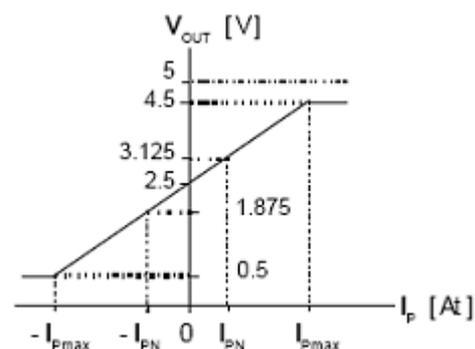


Figure 4.12 LEM output voltage.

In this work, the current drawn by the motor is less than 3 A; the corresponding output voltage of the device is way too low to be useful. Hence, some amplification is needed. There are two options to decrease nominal current; the first is connecting sensor pins in different configurations. However this method delivers only a nominal current of 8A. The second method is to use the sense hole on LEM. Multiple (primary) windings increase ampere-turns of the device and thus increasing the resolution of the device. So eight primary turns was provided to have a measurement range of $\pm 3.125A$ where the output is $2.5 \pm 0.625V$. This range was still not very suitable so an amplifier circuit is designed, to amplify LEM output to a more suitable range. Circuit shown at Fig. 4.13 subtracts 2.5V from LEM output voltage and performs amplification. Then, a bias of 2.5V is added to the result. An amplification gain of 4 simply gives an output voltage range of 0-5V for -3.125A and 3.125A respectively. Sensor gain K_s is found to be $0.8 [V/A]$. Also, a filter at 750Hz is incorporated in order to filter switching noise (at PWM frequency) in the

current measurement. Four op amps are required to implement this circuit. A quad JFET op-amp package TL084 is chosen so that only one IC will be sufficient to implement all these functions.

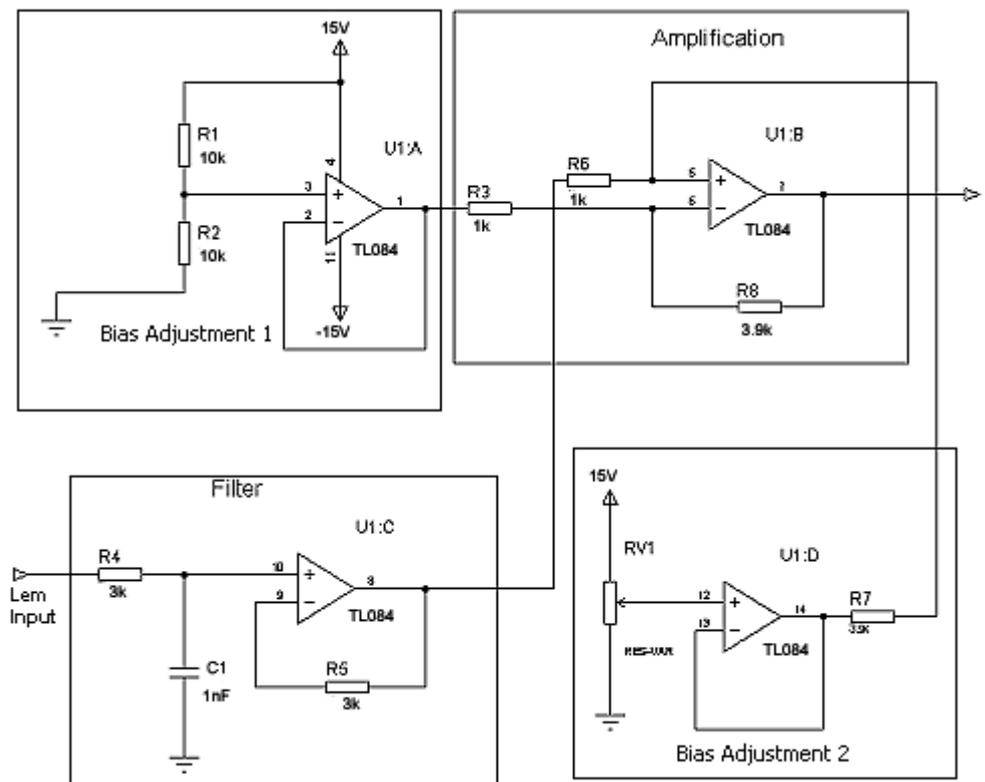


Figure 4.13 Current sensor circuitry.

This current sensor will be used for both continuous-time current regulator and discrete-time current regulator. Next sections describe current regulator implementations.

4.3.2 Continuous-time Current Regulator

Continuous time regulators are implemented using analog circuitry, mainly op amps. Continuous time current regulator designed at Section 4.2 is a proportional controller and can be implemented using only op amps and comparators. Two op amps will be used to apply controller designed previously. Fig. 4.14 illustrates the corresponding implementation. Here, the first op-amp is used to invert current

command so that the error could be calculated by subtracting the command right from the current feedback. The second (inverting) op-amp circuit is then utilized not only to add the inverted command and the current feedback but also to amplify the error by the specified proportional gain, $K_p = 50$. Dual JFET op amp package TL082 includes two op-amps, this choice gives the opportunity to implement the controller with only one IC. Note that a $100\text{k}\Omega$ pot is used to implement the proportional gain:

$$K_p = \frac{R_3}{R_{V1}} \quad (4.6)$$

In this circuit R1, R2 and R3, R4 are matched to each other with a precision of $20\ \Omega$, in order to avoid errors in controller. If resistor pairs were not matched transfer function of the op amp circuit will change and undesired results might occur. The reason of using variable resistor instead of a non-variable $100\text{k}\ \Omega$ is to have some tuning option on controller.

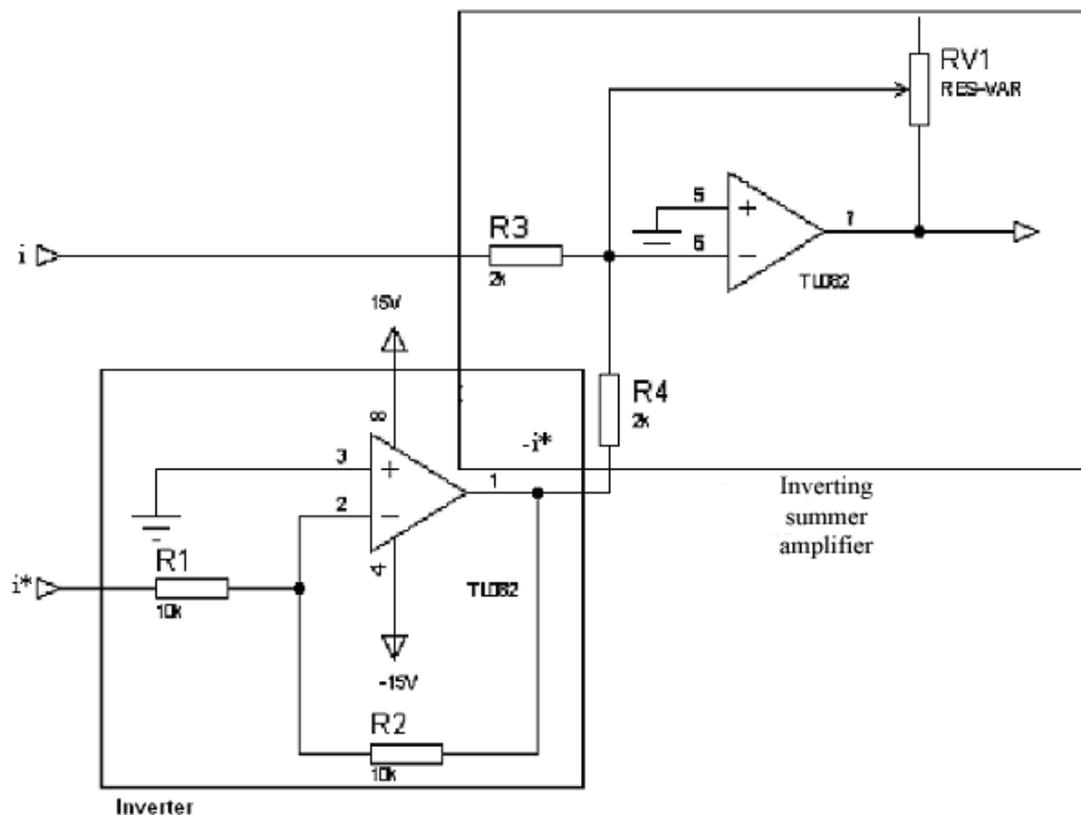


Figure 4.14 Continuous-time current regulator circuit.

Note that scaling is an important issue in the implementation of the controllers. In the controller design phase, all feedback gains are assumed unity. However, in practice, that is really not the case. Considering that feedback has a gain of 0.8 from current sensor (i.e. 3.125A gives 2.5V), current command should also be scaled accordingly. New block diagram is obtained as shown in Fig. 4.15.

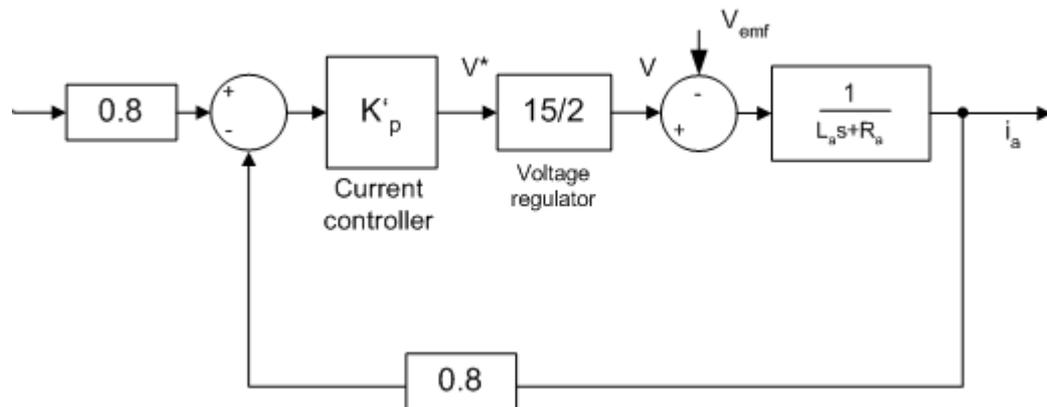


Figure 4.15 Block diagram of continuous-time current regulator.

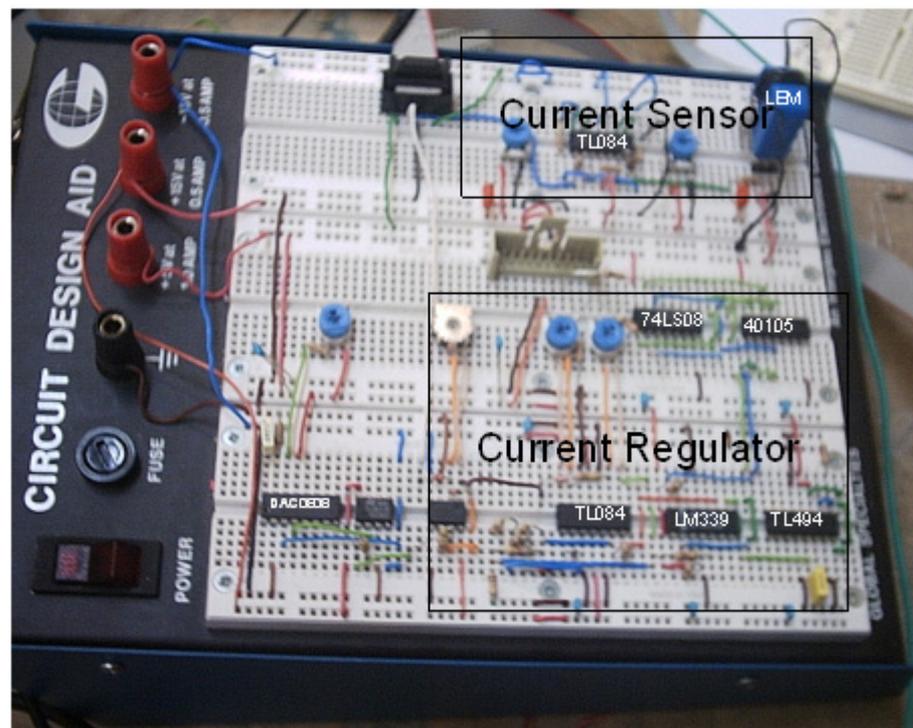


Figure 4.16 Continuous-time current regulator.

4.3.3 Discrete-time Current Regulator

Discrete-time controllers are implemented using microcontrollers or microprocessors. In this study, the regulator is implemented using a PIC16F877 microcontroller. PIC16F877 was chosen because of seven A/D converters and two PWM output channels are integral parts of this microcontroller. Another important reason for choosing PIC16F877 is its availability in local part suppliers. PIC firmware for current regulator can be divided into two important parts. As can be seen in Fig. 4.17, analog current feedback ranging between 0-5V is sampled and converted to digital data via PIC's A/D converter. PIC has a 10 bit A/D converter which can be set to operate between different reference voltages. The output of current sensing circuitry is connected to this input. A/D conversion module of firmware gets current feedback and compares this value to the current command. Error is multiplied by gain in order to generate manipulation signal. Note that as mentioned in the previous section, due to the current feedback gain, the theoretical gain should be reduced accordingly. Although the same situation was encountered at continuous time current regulator design, higher gain bears no problems in the continuous time domain. However, a higher gain in discrete time current regulator results in forced oscillations as stated at section 4.2. So the effective gain used in the micro-controller should be $12 \cdot 0.8 / 7.5 = 1.28$. This multiplication with a rational number is problematic since 16F877 doesn't have a hardware multiplication module and floating point support. So above gain should be achieved by using multiplications with powers of 2, which operation PIC hardware supports.

$$(2^0 + 2^{-2} + 2^{-5}) \cdot E = 1.28125 \cdot E \quad (4.7)$$

Applying (4.7) above gives the desired gain, with an error of %0.098. Note that in order to multiply a binary number with 2^N (N is a positive integer), one needs to shift all the bits of that number to the left by N bits while stuffing the "evacuated" bit positions by 0. Similarly, a division by 2^N (or multiplication by 2^{-N}) can be carried out by shifting all the bits to the right by N bits. Note that many microprocessors do have specific instructions to perform bit shifting operations directly.

For instance, 16F877's instruction set includes RRF (rotate right through carry) and RLF (rotate left through carry) to carry out these essential multiplication and division operations.

Note that sampling time is set by a timer generated interrupt on PIC so that every 1 ms an interrupt is generated. This interrupt sets a flag to activate control routine. This routine reads current feedback, calculates error, generates manipulation signal and send it to voltage regulator by PWM. Flow chart of PIC control firmware is given at Fig. 4.17.

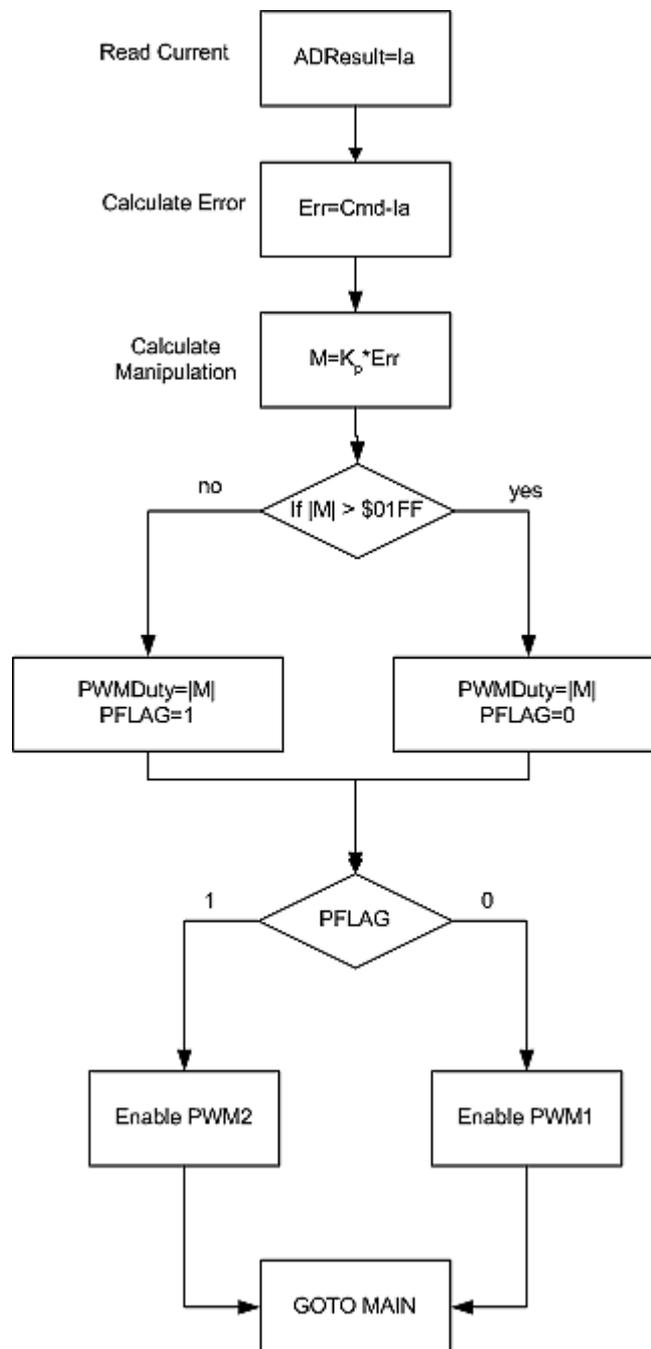


Figure 4.17 Flow chart of discrete-time current regulator firmware.

4.4 Performance Tests

In these tests, converter B (integrated H-bridge IC) is used in conjunction with continuous-time current controlled voltage regulator, owing to the fact that this chip drives reliably the DC motor considered in this study. During the tests it has been observed that L298 is overheating if run at extended periods of time (a few minutes) at 24V.

The implemented current regulators are to be tested using standard inputs:

- sine
- trapezoidal,
- triangle inputs.

The test results are presented in the following sections.

4.4.1 Continuous Time Current Regulator

In this test the current command is generated via a programmable function generator. Notice that the current command has to be in the range of 0-5V to maintain compatibility with the output voltage of the Hall effect current sensor. The results of scope outputs captured in the performance tests are presented in Figs 4.18, 4.19 and 4.20.

As can be seen from Fig. 4.18 the current outputs track the current command pretty well. In order to determine the bandwidth of the current loop, its Bode plot considered. To create the plot, one needs to simply plot out the ratio of the magnitudes for the input and the output. The result of such an effort is illustrated at Fig. 4.21. As can be seen the cut off frequency of the current loop is around 700 Hz as expected. Even though, theoretical bandwidth is higher than 1 kHz, the low pass filter with cut off frequency of 730 Hz is included in the current amplification stage, in order to reduce switching noise at 50 kHz, dominates the dynamics of the current loop. The data used in preparation of Bode plot was given at Appendix C.

Sine inputs with 0.4V magnitude is applied to the regulator with frequency ranging from 30 Hz to 750 Hz. These results are used to generate bode plot of the system.

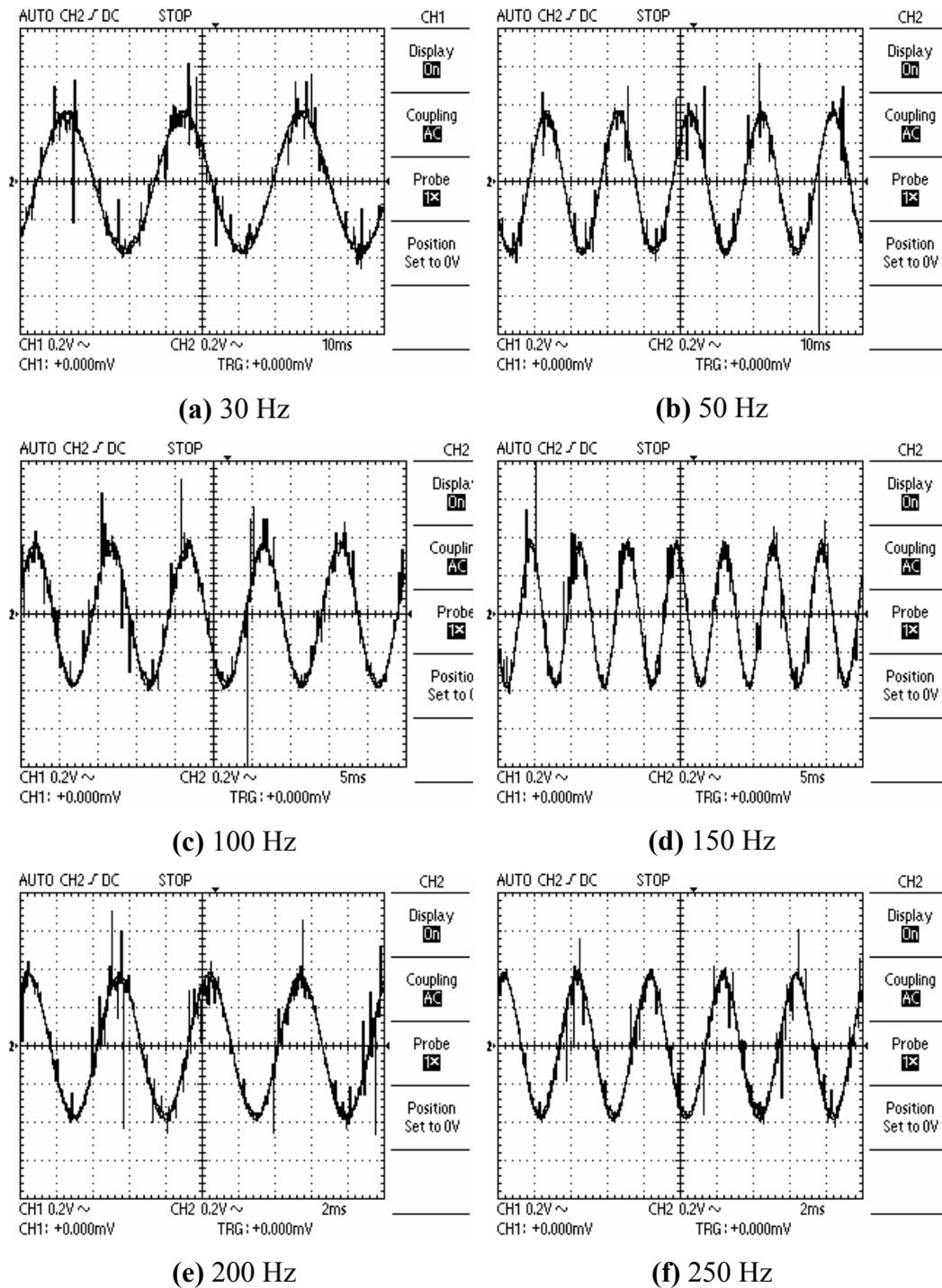


Figure 4.18 (Continued)

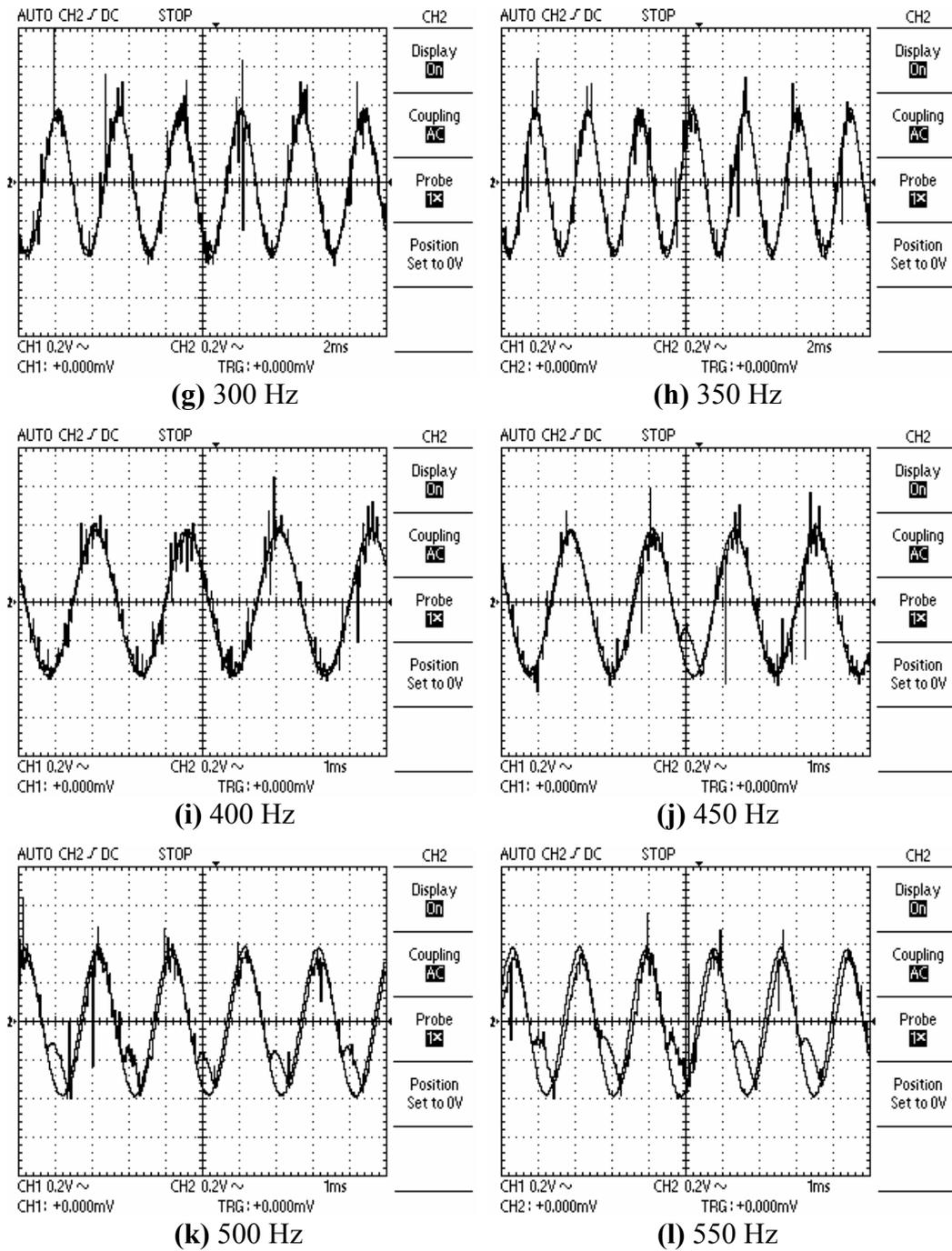


Figure 4.18 (Continued)

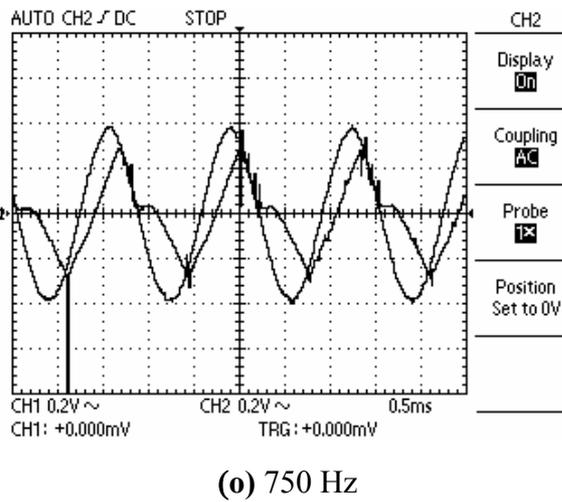
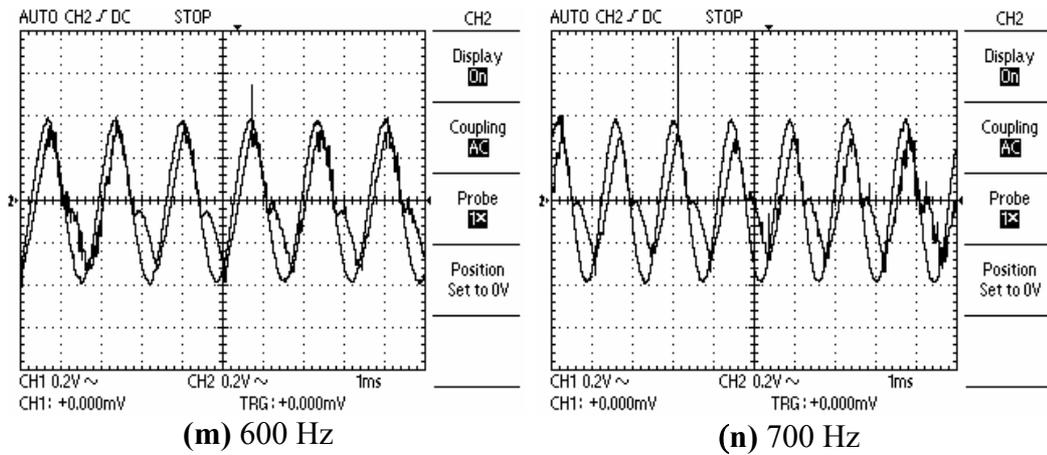


Figure 4.18 Sine input and output without offset.

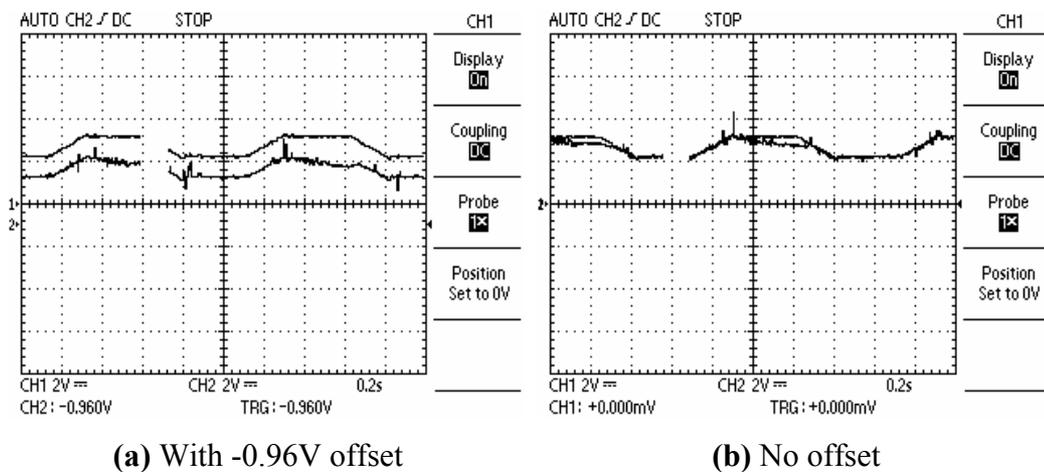


Figure 4.19 Trapezoidal Input.

With respect the response of current regulator to trapezoidal inputs, all results shown in Fig. 4.19 , the regulator follows the trapezoidal command. However after an overshoot, a steady state error is observed. This error is the effect of back emf voltage acting on motor. As speed increases back emf voltage also increases and the disturbance caused by back emf voltage creates that error. To overcome this problem a back emf decoupling could be designed. However, this error can't be decoupled because the power supply used to drive the motor is not suitable for this task, which is a 15V 1.5A power supply. Also note that some noise pickup from cable as well as poor ground connections made through crocodile connectors.

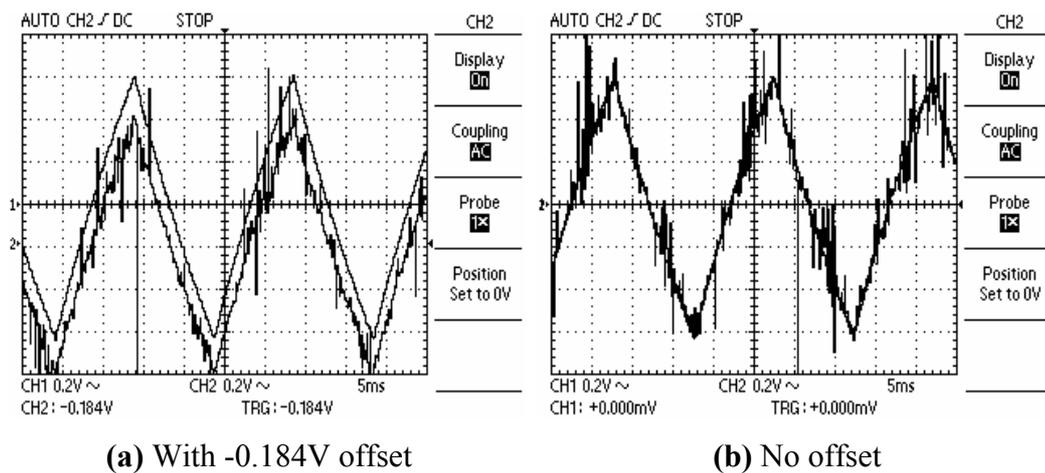


Figure 4.20 Triangular input at 50 Hz.

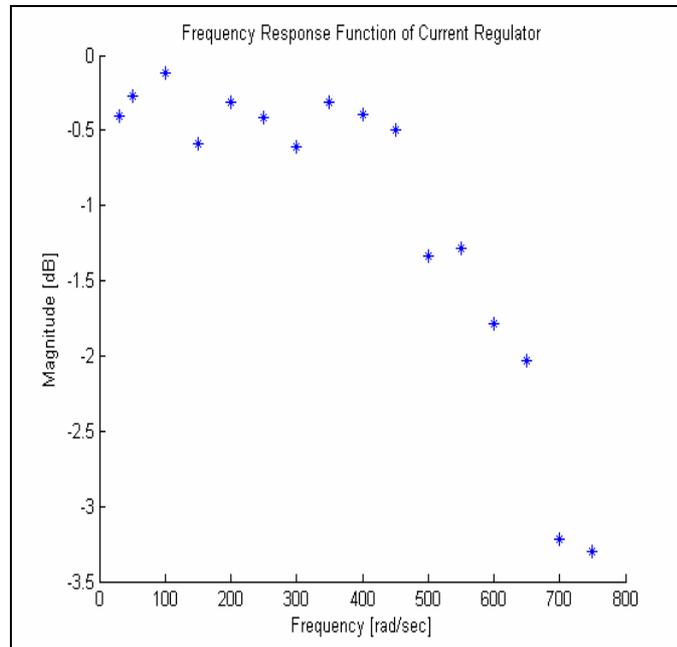


Figure 4.21 Frequency response of current regulator.

4.4.2 Discrete-time Current Regulator

As mentioned earlier, the use of PIC16F877 microcontroller as current regulator has many attractive features such as the simplicity, reduced cost and ease of hardware implementation. During the design phase of the discrete-time controller, it has been observed that the regulator exhibits erratic behavior. In order to find out the source of the fault in the discrete-time regulator, a great deal of time was spent. The following items were checked to get down to the bottom of this problem:

- Power supplies,
- Converters,
- RS-232 interface
- Current sensors and interfaces
- Measurement devices
- Firmware
- Cables and connections

When investigated rigorously it was found out that the A/D converter of the microcontroller is not working properly beyond certain input frequencies (right above 100 Hz). It was quite surprising to see that this limitation of the A/D converter was documented neither in its datasheet nor the application notes/bulletins nor the technical documents published by the manufacturer; Microchip. Furthermore none of the user groups of PIC microcontrollers were aware of this problem.

To document this problem the following tests were conducted. Sinusoidal analog voltages at various frequencies were applied to one of the A/D converters of the microcontroller. Then, the result was sampled at sampling frequency of 1 kHz. The result was written to the PWM registers to generate PWM output. The output was then filtered out with a low pass filter with a cut off frequency of 484 Hz. The experimental results are presented in Figs 4.22 through 4.26.

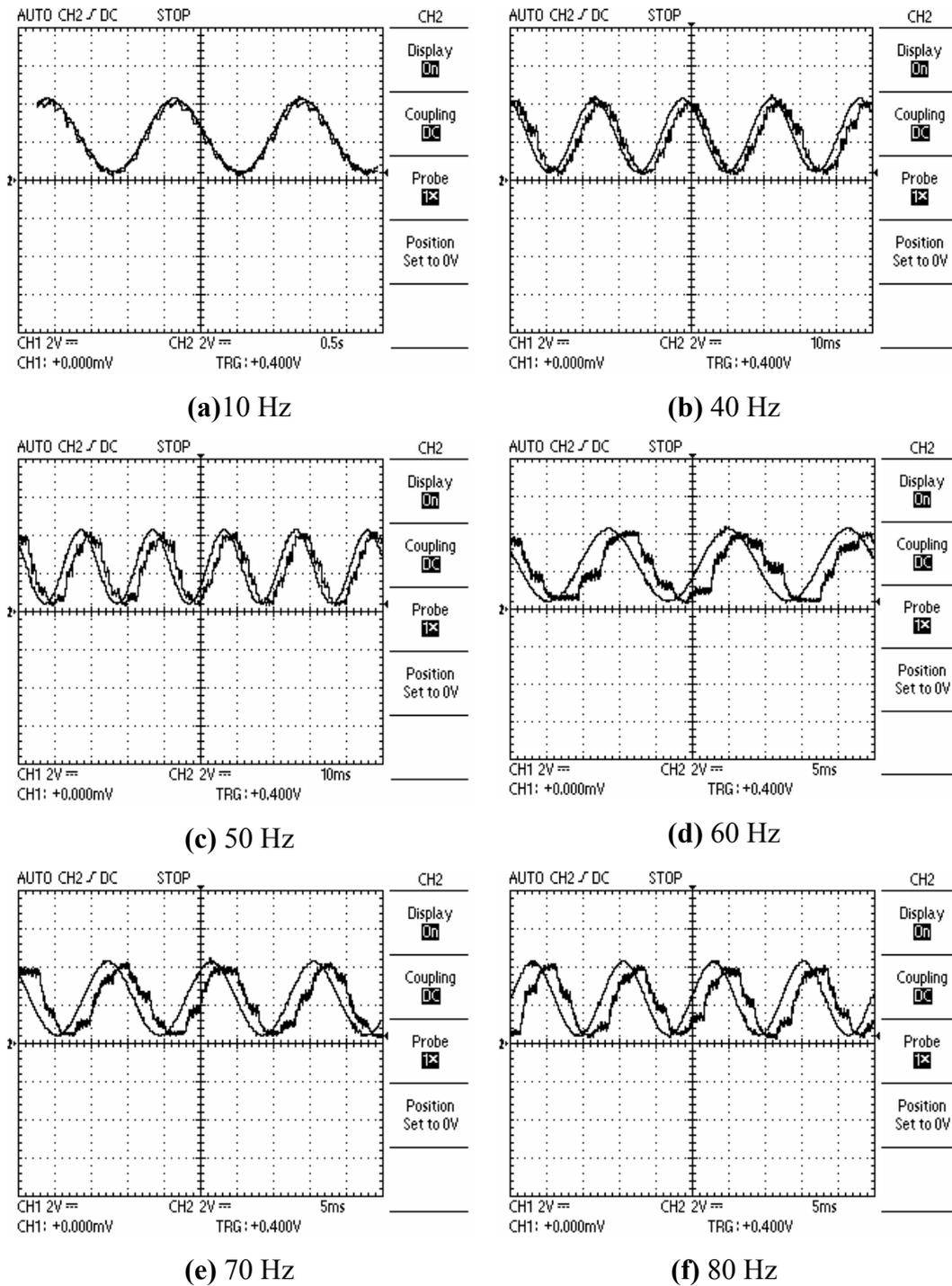
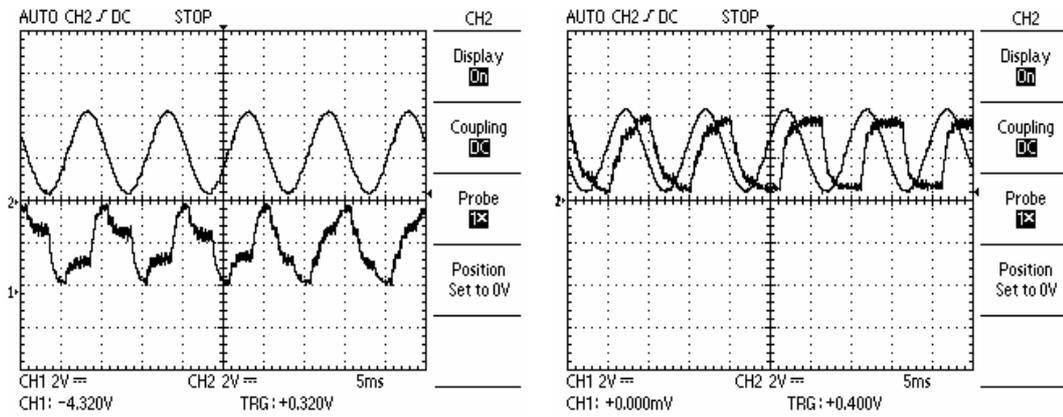


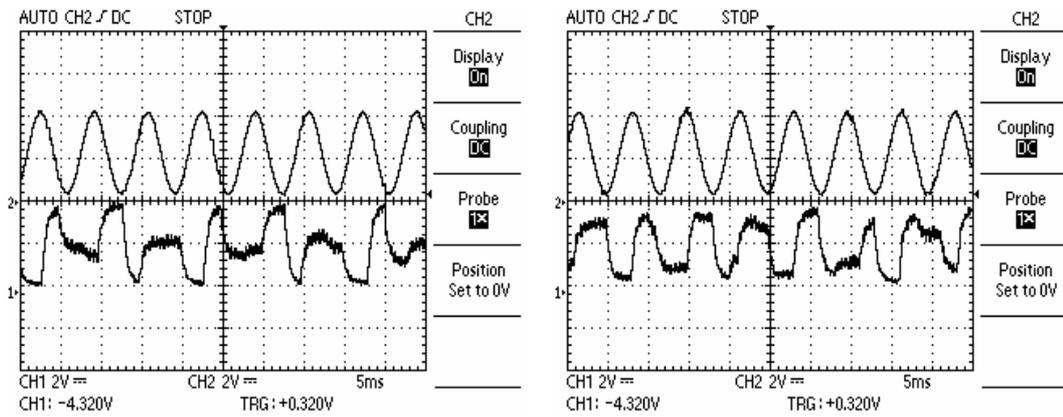
Figure 4.22 A/D conversion test input 10 Hz- 80 Hz.



(a) Output with an offset

(b) No offset

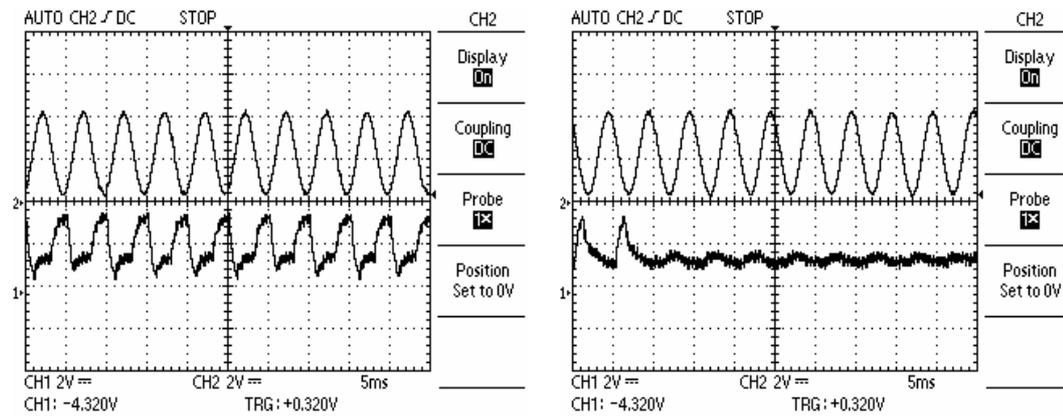
Figure 4.23 A/D test results at 100 Hz.



(a) Output with an offset

(b) Output with an offset

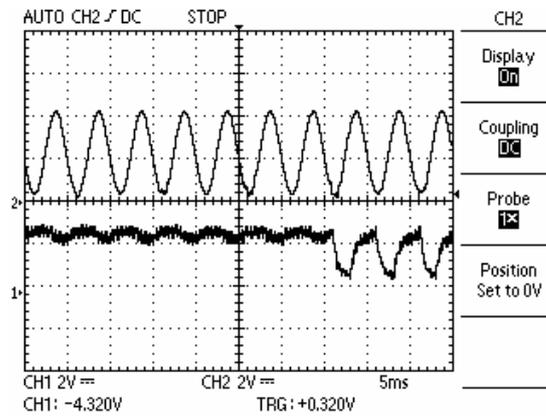
Figure 4.24 A/D test results at 150 Hz.



(a) Output with an offset

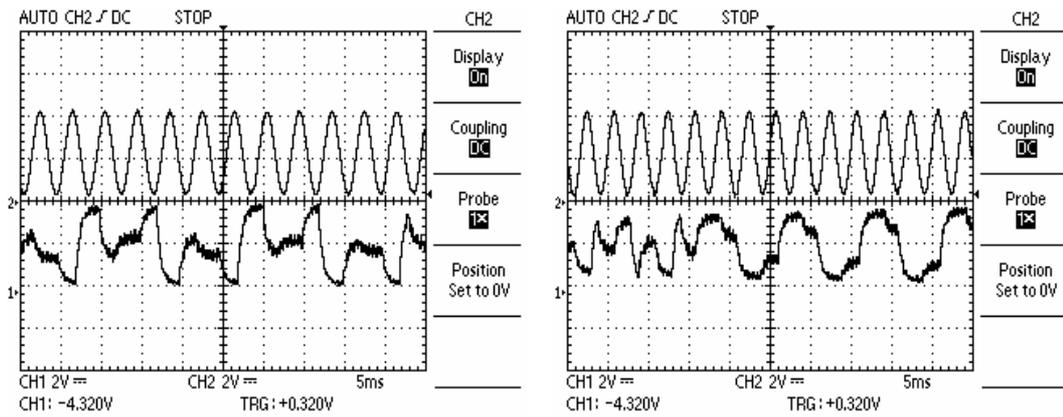
(b) Output with an offset

Figure 4.25. (Continued)



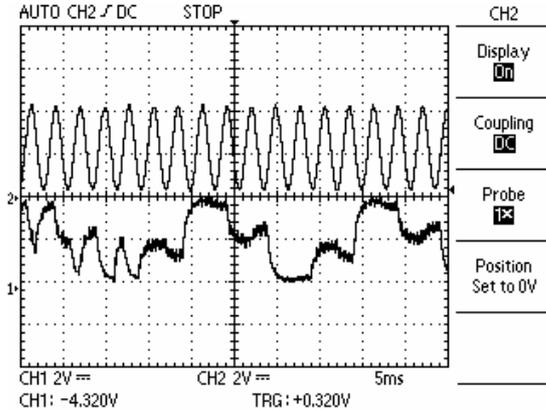
(c) Output with an offset

Figure 4.25 A/D test results at 200 Hz.



(a) Output with an offset, Input at 250 Hz

(b) Output with an offset, Input at 300 Hz



(c) Output with an offset, Input at 350 Hz

Figure 4.26 A/D test results at 250 Hz – 350 Hz.

As can be seen in Fig. 4.22, at frequencies below 80Hz, A/D conversion results were as expected. However, it can be seen from Fig. 4.23 that starting at 100 Hz, A/D converter exhibits a strange dynamic response. Figs 4.24-4.26 show higher frequency responses, it is clear that there is a dramatic error on PIC 16F877's A/D conversion. Two different PIC 16F877s showed the same problem so this effect is thought to be a result of microcontrollers Sample and Hold circuitry. To solve this problem a 560 pF capacitor is connected between the A/D pin and ground. However, even this capacitor doesn't eliminate the erratic behavior problem. In light of this development, the discrete-time current regulator design efforts have to be abandoned.

4.5 Conclusion

Two current regulators are designed and tested in this chapter. The discrete time controller is found to be problematic because of the A/D conversion module. The continuous-time controller operates as expected and good results are obtained for various inputs. However, these tests showed that controller can't compensate for back emf after certain speeds because the power supply could not generate enough power.

Experiments and implementation show that Hall effect current sensor is an easy to implement solution. As in this case, if design is made considering the recommended current limits for linear operation, there exists no problem. Although for this research DC motor current is very low considering LEM's capabilities. If higher current sensing capabilities are desired, LEM amplification circuit can be configured to higher currents easily. Therefore, the same circuitry can be used for other applications as well.

CHAPTER 5

MOTION CONTROLLER

5.1 Introduction

High-precision control of electrical motors plays a critical role in many industrial applications, such as CNC machine tools, automation, robotics, flexible manufacturing systems, aerospace/aviation, automotive engineering, etc. Among many electrical motors, brush-type (conventional) DC motor is still widely used in such applications, owing to the fact that they are inexpensive, rugged, and easy-to-control. However, it has a basic drawback in its very design: the electrical power has to be transferred to the rotating windings via armature brushes. Hence, mechanical contact between armature brushes and the rotating contactors is required. Evidently, the armature mechanical assembly is susceptible to wear and tear. In the earlier days, the brushes, which were usually made from carbon, has to be regularly maintained. Thanks to powder metallurgy, the armature brushes are made from improved composite materials so that their life span and as well as the overall performance of motor have been greatly enhanced. Even though AC servomotors, induction motors and switched reluctance machines are getting popular everyday; the conventional DC motors are still excellent alternatives for many motion control applications, where the cost and ease of implementation are of critical importance.

The objective of this chapter is to design high-performance discrete-time motion controller for a conventional DC motors. To achieve desired performance

objectives such as command tracking and disturbance rejection, all of the controllers to be discussed here require a high-bandwidth (and accurate) torque modulator (i.e. current controllers) such as the one discussed in Chapter 3. Note that once a proper torque modulator for a particular electric motor is designed, the motion controller topology is independent of the type of motor (e.g. brushless DC motor: permanent magnet synchronous motor) to be controlled. Consequently, it is relatively easy to adapt these motion control principles to other motor systems as well.

5.2 Elements of Motion Control

In order to design a high precision motion controller for a DC motor one needs to develop the model of such a system. For instance, the equation governing the dynamics of a typical brush-type DC motor can be written as

$$J\dot{\omega} + b\omega + T_c \operatorname{sgn}(\omega) + T_L = T_m \quad (5.1)$$

where

- J: Equivalent Mass Moment of Inertia of Motor [kgm^2],
- b: Viscous friction coefficient [Nm/rad/s],
- T_c : Coulomb (Dry) Friction Coefficient [Nm],
- T_L : Load torque [Nm],
- T_m : Torque developed by motor [Nm],
- ω : Angular speed of motor [rad/s].

Inspecting (5.1) shows that like many other motors, the load acting as the disturbance has a direct influence on the (controlled) motor's speed. That is, the motor will change its speed under the action of the load even if motor torque is held constant. In order to implement a motion controller, the speed of the motor should be detected. A common method of detecting motor speed is with the use of a tachogenerator. A tachogenerator is a small, permanent, magnet generator

connected to the motor shaft. The output voltage of the generator is proportional to its speed. Older continuous time motion controllers rely on this detection method. Fig. 5.1 below shows an analog velocity controller.

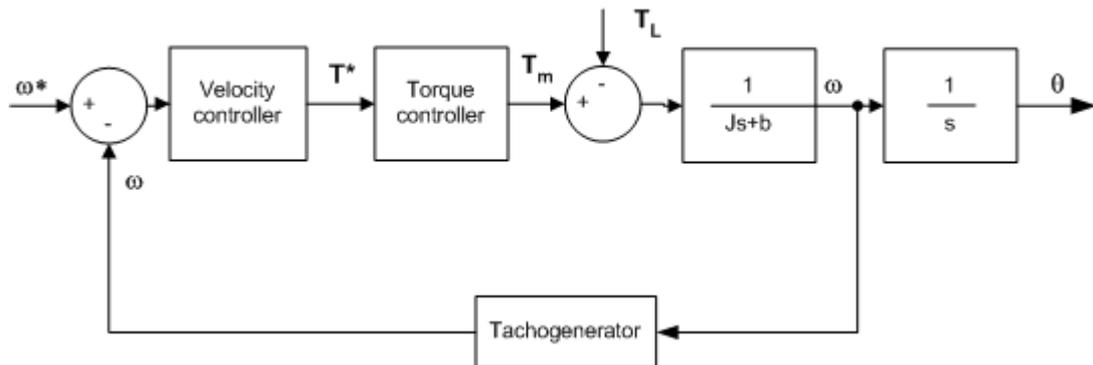


Figure 5.1 Standard continuous-time velocity controller implementing tachogenerator.

Recent motion control techniques use a different approach to motor motion detection. Contemporary motors are equipped with an optical rotary encoder, which produces a relevant signal according to motion of motor.

Incremental optical encoder is suitable for measuring position difference rather than velocity, so the velocity controller shown at Fig. 5.1 should be revised.

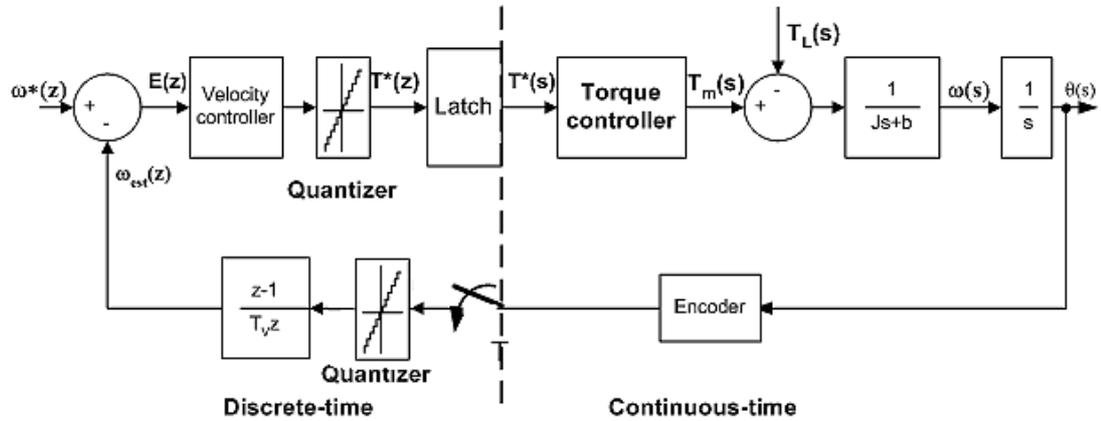


Figure 5.2 Discrete-time Velocity Controller.

As seen at Fig. 5.2 motion loop is closed on estimated velocity. This topology changes velocity plant. Detailed information on controller design can be found at Section 5.2.1.

5.2.1 Designing Motion Controller

As stated previously at Section 5.2 motion control plant transfer function should be revised for incremental optical encoder feedback. Since a discrete time controller will be designed, discrete time transfer function is found. Sampling time (T) is 0.004s.

$$\frac{\omega(z)}{M(z)} = G_{PV}(z) \quad (5.2)$$

$$G_{PV}(z) = \mathbb{Z} \left[G_{ZOH}(z) \frac{1}{Js+b} \right] \quad (5.3)$$

where $M(z)$ is manipulated input. Here, ZOH, which is also referred to as Zero Order Hold, represents latch plus an ideal D/A converter in practice. It's transfer function can be given as

$$G_{ZOH}(z) = \frac{(1-z^{-1})}{s} \quad (5.4)$$

$$G_{PV}(z) = (1-z^{-1})\mathbb{Z}\left[\frac{1}{Js+b} \frac{1}{s}\right] \quad (5.5)$$

$$G_{PV}(z) = \frac{(1-z^{-1})}{J}\mathbb{Z}\left[\frac{\frac{b}{J}}{s(s+\frac{b}{J})}\right] \frac{J}{b} \quad (5.6)$$

$$G_{PV}(z) = \frac{(1-z^{-1})}{b} \frac{1-e^{-\frac{b}{J}T}}{(1-z^{-1})(z-e^{-\frac{b}{J}T})} \quad (5.7)$$

Assuming that $e^{-\frac{b}{J}T}$ term is close to 1 which implies that damping is approaching to zero, (5.7) can be rewritten as

$$G_{PV}(z) = \frac{1-e^{-\frac{b}{J}T}}{b} \frac{1}{(z-1)} \quad (5.8)$$

(5.8) gives standard discrete velocity plant. In order to find velocity loop with estimated velocity feedback discrete time transfer function of position should also be found.

$$\frac{\theta(z)}{M(z)} = G_{PP}(z) \quad (5.9)$$

$$G_{PP}(z) = \mathbb{Z}\left[G_{ZOH}(z) \frac{1}{Js^2 + bs}\right] \quad (5.10)$$

$$G_{PP}(z) = (1-z^{-1})\mathbb{Z}\left[\frac{1}{Js^2 + bs} \frac{1}{s}\right] \quad (5.11)$$

$$G_{PP}(z) = \frac{(1-z^{-1})}{J}\mathbb{Z}\left[\frac{\frac{b}{J}}{s^2\left(s+\frac{b}{J}\right)}\right] \frac{J}{b} \quad (5.12)$$

$$G_{PP}(z) = \frac{(1-z^{-1})}{b} \left\{ \frac{z \left[\left(\frac{b}{J} T - 1 + e^{-\frac{b}{J}T} \right) z + \left(1 - e^{-\frac{b}{J}T} - \frac{b}{J} T e^{-\frac{b}{J}T} \right) \right]}{\frac{b}{J} (z-1)^2 \left(z - e^{-\frac{b}{J}T} \right)} \right\} \quad (5.13)$$

Assuming that $e^{-\frac{b}{J}T}$ term is close to 1, and making relevant simplifications (5.13) boils down to the following equation:

$$G_{PP}(z) = \frac{1 - e^{-\frac{b}{J}T}}{b} \frac{T}{2} \frac{z+1}{(z-1)^2} \quad (5.14)$$

$$G_{PP}(z) = G_{PV}(z) \frac{T}{z-1} \frac{z+1}{2} \quad (5.15)$$

Therefore, an estimated velocity transfer function can be defined as

$$\frac{\hat{\omega}(z)}{M(z)} = \frac{1 - e^{-\frac{b}{J}T}}{b} \frac{1}{z-1} \frac{1+z^{-1}}{2} \quad (5.16)$$

Note that the important system parameters are given as follows

- $b=8.5 \times 10^{-5}$ [Nm/rad/s]
- $J=0.0011$ [kgm²]
- $T=0.004$ [s] ($f_s=250$ Hz)

The reader is encouraged to refer to Appendix B on how to determine DC motor parameters. Substituting given viscous friction, inertia and sampling time values yield:

$$\frac{\hat{\omega}(z)}{M(z)} = 1.818 \frac{z+1}{z^2 - z} \quad (5.17)$$

Characteristics of plant is given at Figs. 5.3 and 5.4 below. It should be noted that K_t found at Chapter 4 is taken into account.

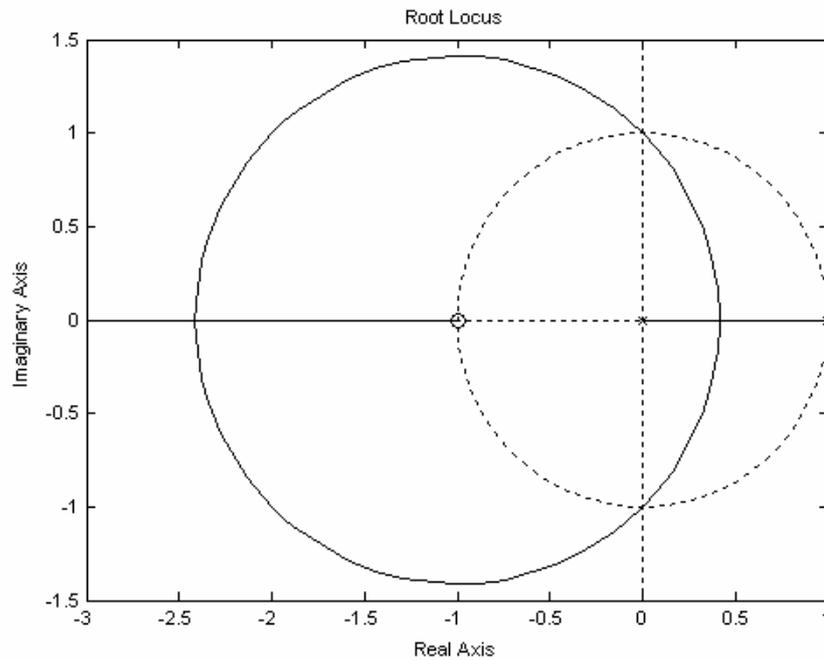


Figure 5.3 Root locus of velocity plant.

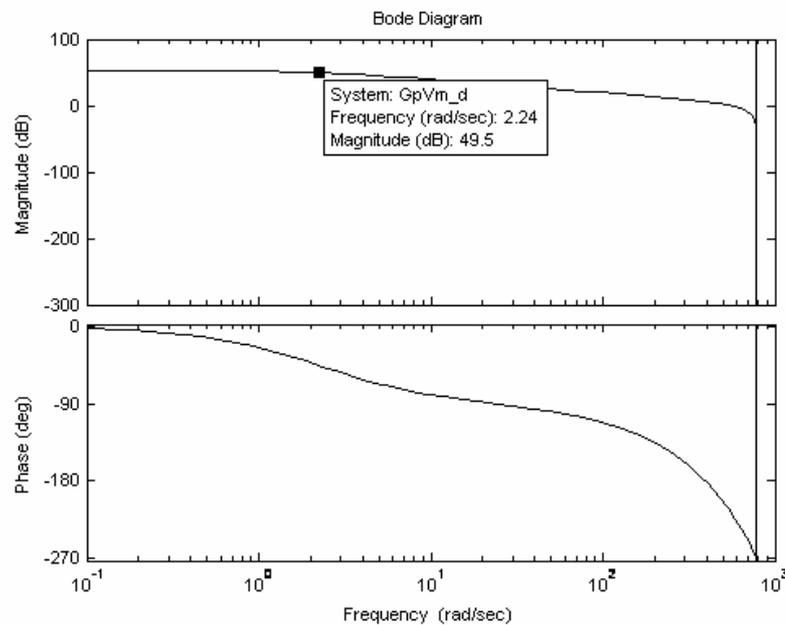


Figure 5.4 Bode plot of velocity Plant.

Inspecting Fig. 5.3 shows that, there are open-loop poles at 0, 0.9997 and a zero at -1. Root locus of the plant shows that, open loop system is stable however its response is pretty slow. Bode diagram at Fig. 5.4 confirms this fact and the

bandwidth frequency of the system is found to be 2.24 rad/s. Considering the required bandwidth of 100 rad/s this bandwidth frequency is not acceptable, so a controller should be designed to solve this problem.

5.2.1.1 Incremental Velocity Controller

As explained in Chapter 2, there exist several motion controller topologies in literature. If designed properly, the incremental velocity controller exhibits excellent disturbance rejection as well as remarkable command tracking properties. The controller to be discussed here is essentially the one presented by Lorenz et. al. [55]. As discussed in the previous section, the plant models had to be developed. First inspecting the root locus of the system given in Fig. 5.3 shows that, a gain is sufficient to obtain a stable system with desired controlled system characteristics. Considering this, an incremental Velocity controller with a proportional gain of $K_p = 2.3$ is designed. Root Locus of compensated system is given in Fig. 5.5. Examining root locus plot of system shows that closed loop system is stable where closed loop poles located at 0.452 and 0.377. Hence the system exhibits the response of an under damped system. In fact, compensated systems bandwidth frequency is 132 rad/s, the resulting bode plot is shown in Fig. 5.6 and response of system to step response is given at Fig. 5.7.

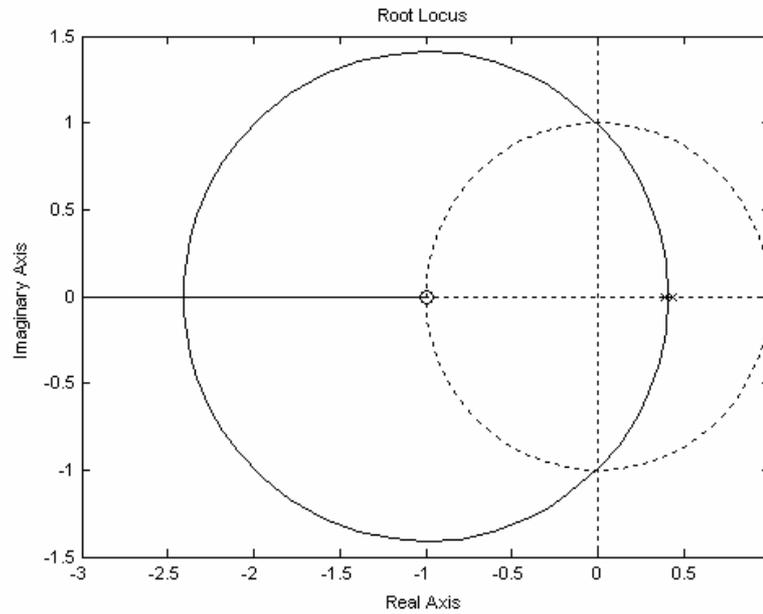


Figure 5.5 Root locus plot of incremental P velocity controller compensated system.

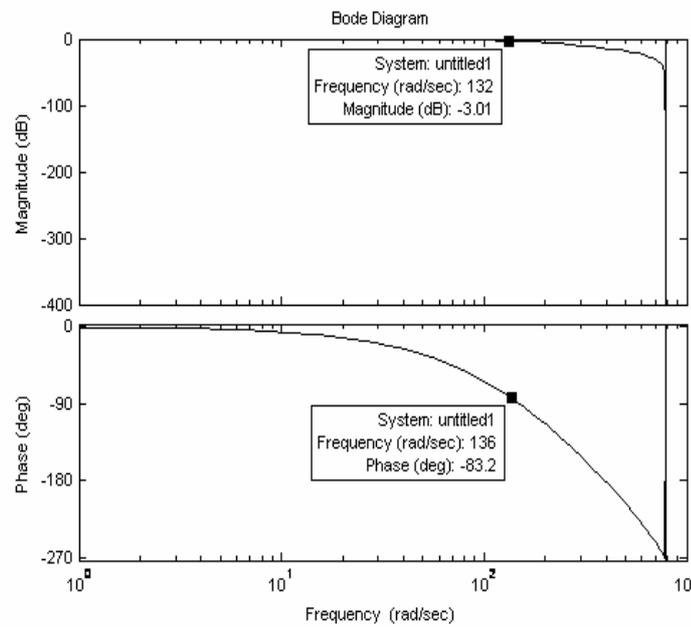


Figure 5.6 Bode plot of incremental P velocity controller compensated system characteristics.

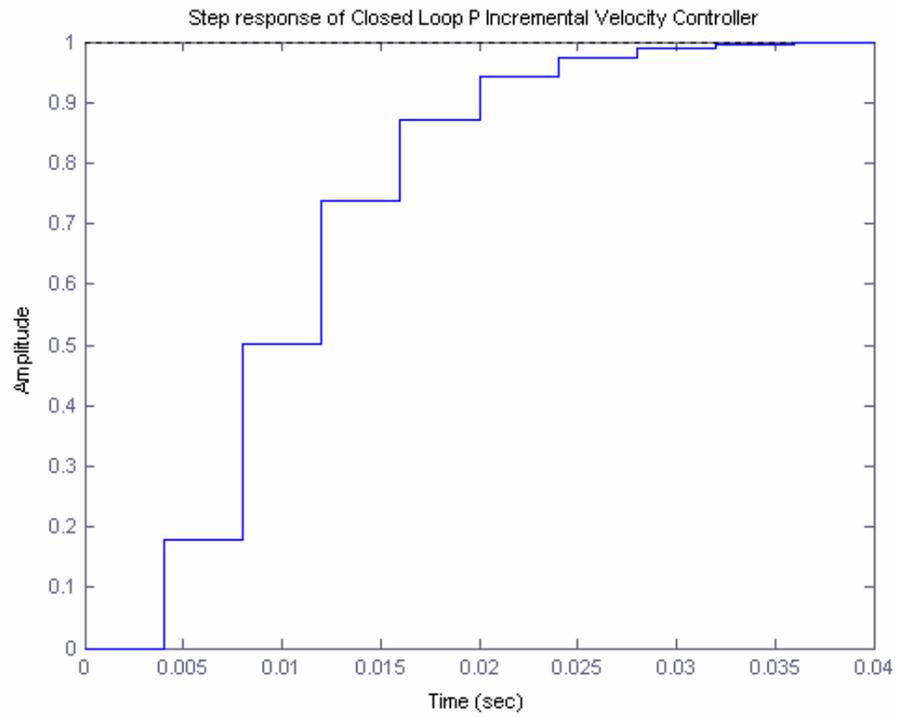


Figure 5.7 Step response of Closed-loop Incremental P Velocity Controller.

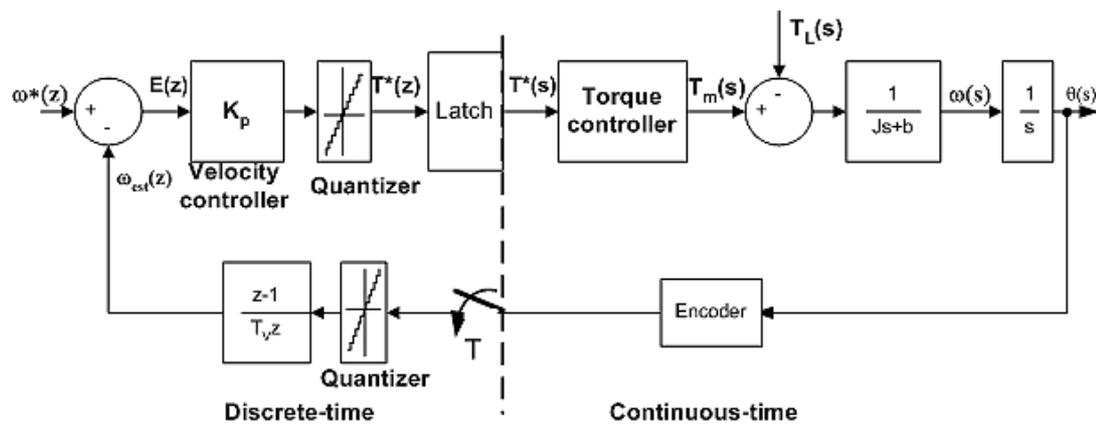


Figure 5.8 Model of incremental P velocity controller.

5.2.1.2 Incremental Position Controller

Lorenz et. al. [55] shows that an incremental state-space position controller is equivalent to incremental PI velocity controller. Hence, incremental PI velocity controller is in a way, position regulator.

In order to have a critically or under damped system, K_p and K_i gains are set to; 2.4035 and 0.1265, respectively, setting a zero for controller at 0.95 as can be seen from the closed loop root locus of the compensated system given at Fig. 5.9. As shown in Fig. 5.10 compensated systems bandwidth is 165 rad/s.

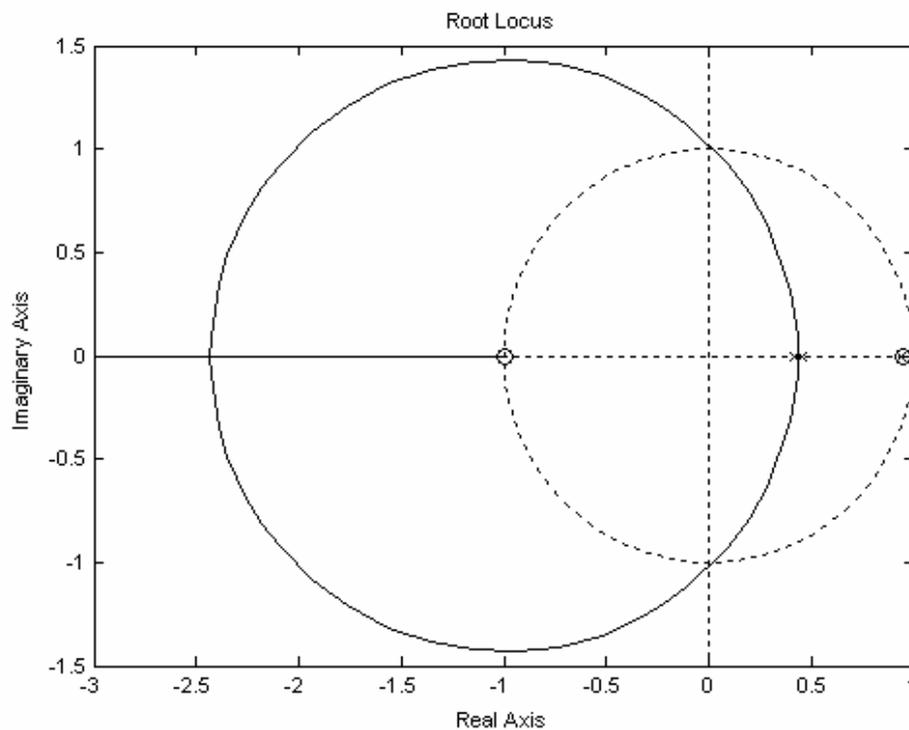


Figure 5.9 Closed loop root locus of incremental position controller.

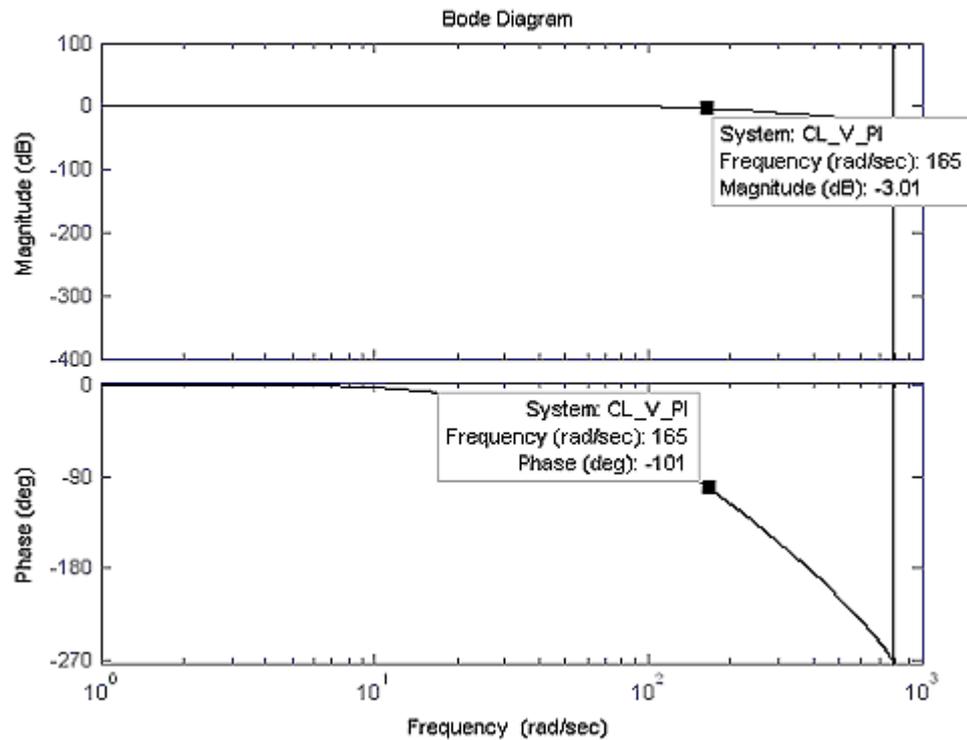


Figure 5.10 Closed loop bode of incremental position controller.

Note that all of these designed motion controllers do require a high bandwidth torque modulator (current regulator) such that the desired torque from the motor can be generated almost instantaneously. This kind of assumption actually is quite valid for modern electrical drives including the one designed in this study. Next section discusses the implementation issues of the designed motion controllers.

5.3 Implementation of Motion Controllers

Motion controllers proposed at Section 5.2 are implemented employing a PIC16F877 microcontroller. PIC16F877 is responsible for running control firmware and generating current command. Encoder interface circuitry sends incremental position feedback to the PIC. A simple block diagram is given at Fig. 5.11.

Circuitry used for implementing the motion controller is shown at Fig. 5.12. Detailed information on hardware and firmware is at sections 5.3.1 and 5.3.2 respectively.

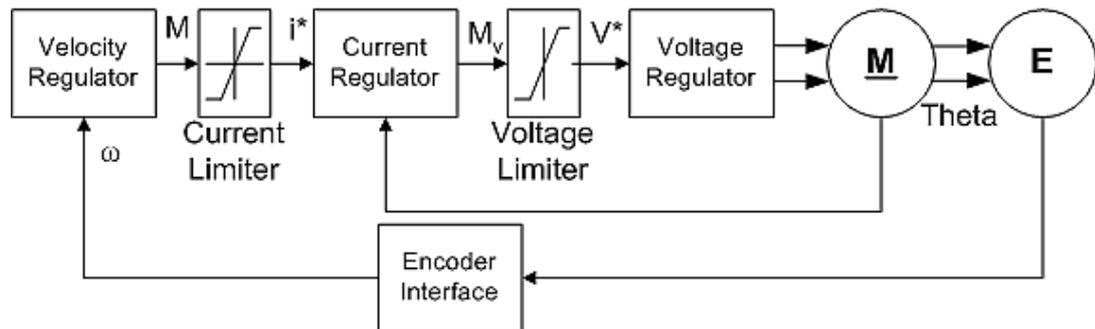


Figure 5.11 Control block diagram.

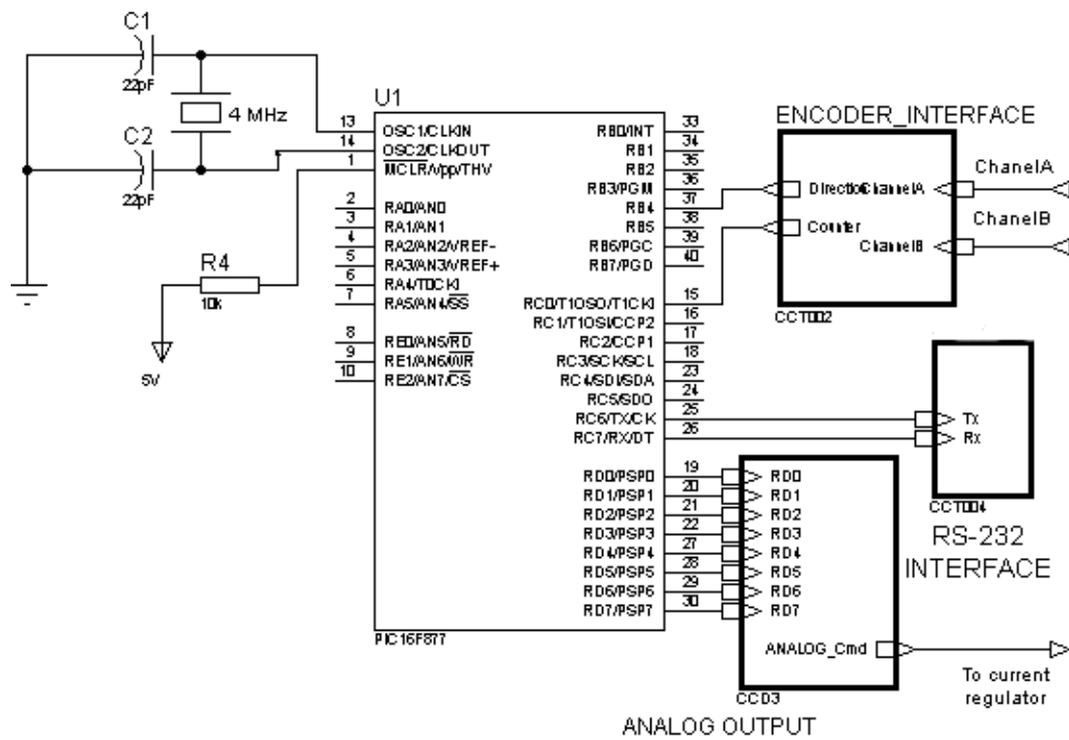


Figure 5.12 Control circuitry.

5.3.1 Hardware Considerations

Peripheral units of the controller are to carry out three critical tasks:

- Generating command for the current controller

- Sending measurements to the PC via serial port
- Encoder interfacing (a.k.a. position decoding)

Command Generation: As elaborated in Chapter 3, a PIC16F877 micro-controller is equipped with proper hardware units (A/D converters, PWM generators) to design a purely discrete-time current controller in theory. Unfortunately, due to its unacceptable A/D conversion performance at high frequencies, such a digital implementation has become implausible in practice. Therefore, a D/A converter is needed to convert the digital current commands of the microcontroller to the reference voltage for the continuous-time current regulator. Note that the current regulator requires a command voltage ranging between 0 and 5V where 2.5V corresponds to zero current. In order to solve the conversion problem, National Semiconductor's DAC0808 8-bit D/A converter has been selected. DAC0808 interfaces with TTL directly and has an overall conversion error of %0.19. Since the PIC16F877 have a large number of digital I/O pins available, DAC0808 can be directly coupled to the microcontroller as illustrated in Fig. 5.13.

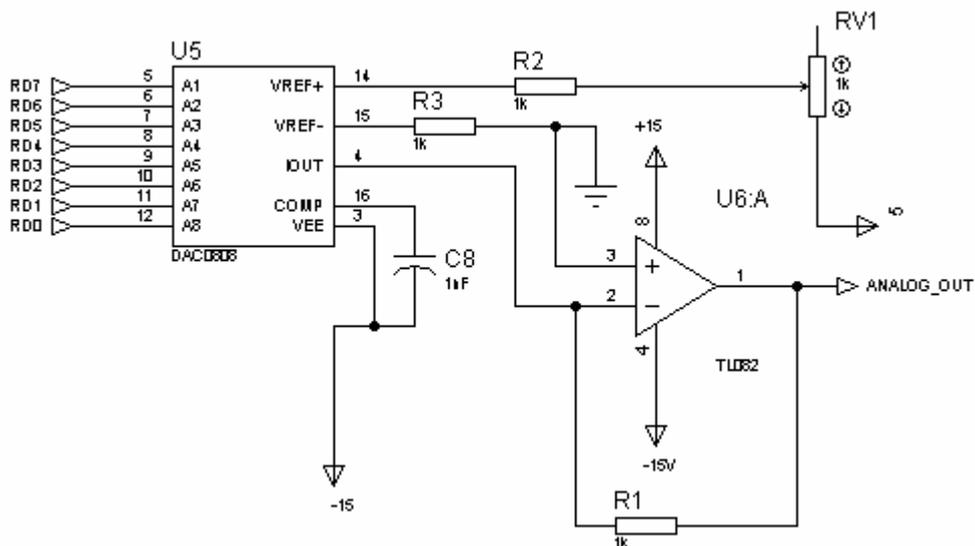


Figure 5.13 D/A circuit.

8 output pins on PORT-D of the micro-controller are allocated to send-out a byte representing the current command. Note that these digital output pins are all buffered and latched. That is, the digital state of a pin remains unchanged until a new bit is sent out to that specific pin. Hence, an 8-bit register is not needed in this particular application. As can be seen from Fig. 5.13, a variable resistor (potentiometer) is further employed to adjust the range of the converted voltage.

Serial Communication: Modern electrical drive systems often need to communicate with its host so as to receive relevant (position/velocity/torque) commands and to send back critical status signals such as errors/faults/alarms or simply measurements. In this application, serial communication between the PC and the microcontroller has to be established via RS-232C (a.k.a. “serial port”) to transmit mainly the motor position measurements for the purpose of evaluating the performance of the designed controller. Furthermore, same communication channel might be also used to receive complex commands from the host so as to accomplish multi-axis motion control. Notice that the PIC16F877 has a built-in *Universal Synchronous Asynchronous Receiver Transmitter* (USART) which can be used to transmit data to the PC. USART sends and receives TTL (i.e. 5 volt logic) signals while RS-232 employs 12-volt logic levels. An interface circuit should be constructed for all intensive purposes. Intersil’s ICL232 chip with dual RS-232 transmitter/receiver is particularly suited for this application [53]. USART output is sent to ICL232 and ICL232 transmits the corresponding signal compatible with ELA RS-232C / V.28 specifications to the serial port of the computer where a terminal program, RS232 Hex Com Tool v4.0 is running. Through this shareware program microcontroller output is written to a text file, which in turn is processed by MATLAB. Also, detailed information on RS-232 asynchronous (single drop) serial communications protocol is further given in the Appendix E.

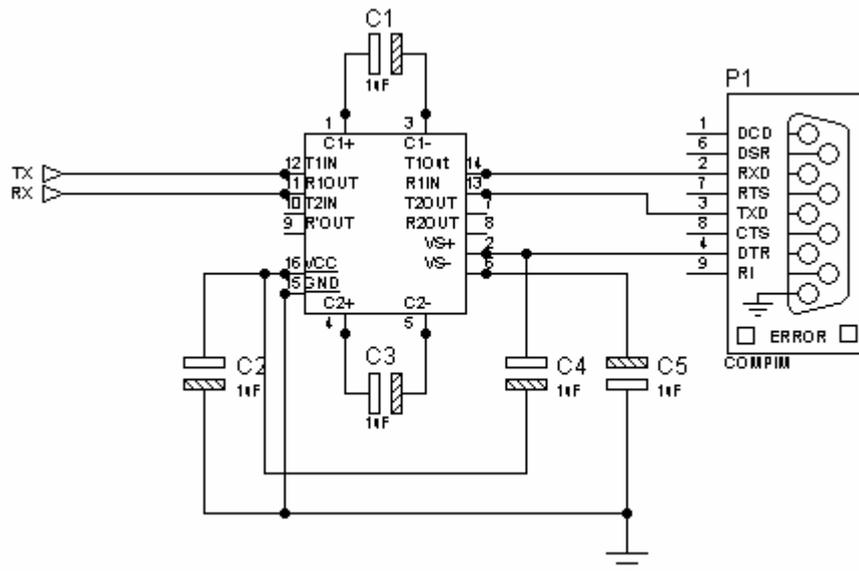


Figure 5.14 RS-232 Interface circuit.

Optical Position Encoder Interface: In order to control motion of motor, one needs to decode the position information supplied by the optical incremental position encoder. There exists different encoder interface approaches in literature, ranging from hardwired solution to dedicated ICs (ASICs). The method preferred in this thesis is a hardwired interface constructed using logic ICs. In this project, encoder directly coupled to the output shaft of the DC motor is built by Autonics Corporation (Model: ENB 2000 3-1). This incremental encoder has two digital channels (A and B) generating 2000 pulses per revolution at each channel. Each pulse stream is spatially 90° apart. The direction of revolution of the encoder shaft is detected by checking out the phase shift between the two pulse streams: If pulse stream at channel A is leading that of the other channel (by 90°), the shaft is rotating, say, clockwise direction. Otherwise, (i.e. the channel A is lagging) the direction is just the opposite. Fig. 5.16 shows the typical output waveforms. The encoder interface, which is actually inspired by the mouse decoder circuit used in a typical PC, is shown in Fig. 5.15. In this solution, a positive-edge triggered D flip-flop is employed to detect the direction of rotation. This flip-flop “samples” its

input (D) right at the instant when the clock input goes out from low to high logic level (0 to 5V). The device holds the sampled data (Q) till the next trigger comes along. Here, the channel B serves as the clock while the other channel is used as the data to be “sampled.” Similarly, to obtain the position of the encoder, one can count the pulses of either channel. The actual position (θ) becomes

$$\theta(k) = \frac{2\pi}{2000} \cdot C(k) \cdot s(k) \quad (5.18)$$

where $C(k)$ is the counter value $\in \{0, 1, \dots, 2000\}$ while $s(k)$ is the sign of direction (e.g. 1 is clockwise; -1 is counterclockwise). This decoding approach, which is referred to as 1X interpolation, disregards the information coming from the other channel. If the exclusive OR (XOR) logical operation is performed on both channels ($y = A \oplus B$), the resolution of the measurement can be literally doubled (i.e. 2X interpolation). Fig. 5.15 shows the implementation of such a function constructed out of NAND logic gates while Table 5.1 demonstrates the corresponding truth table.

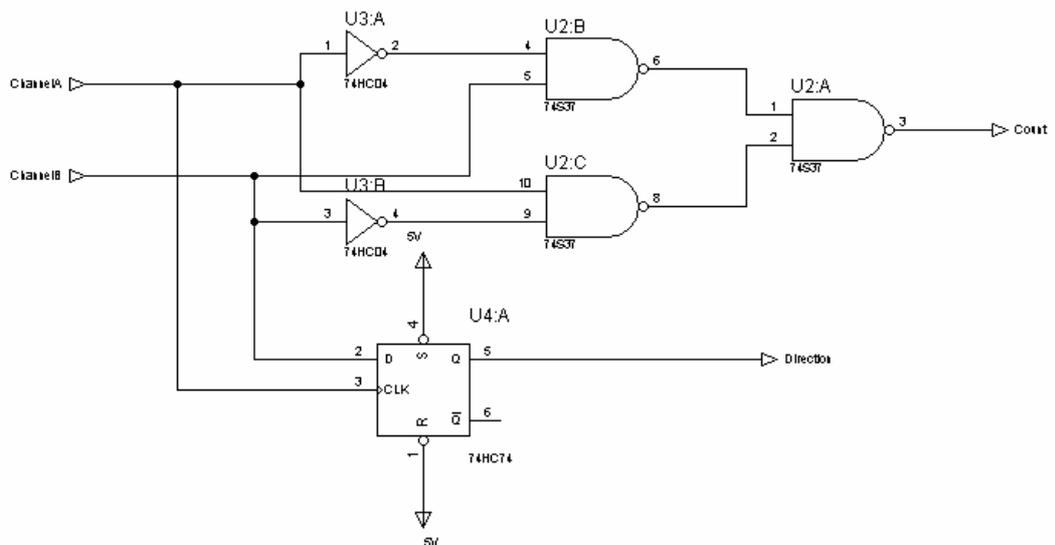
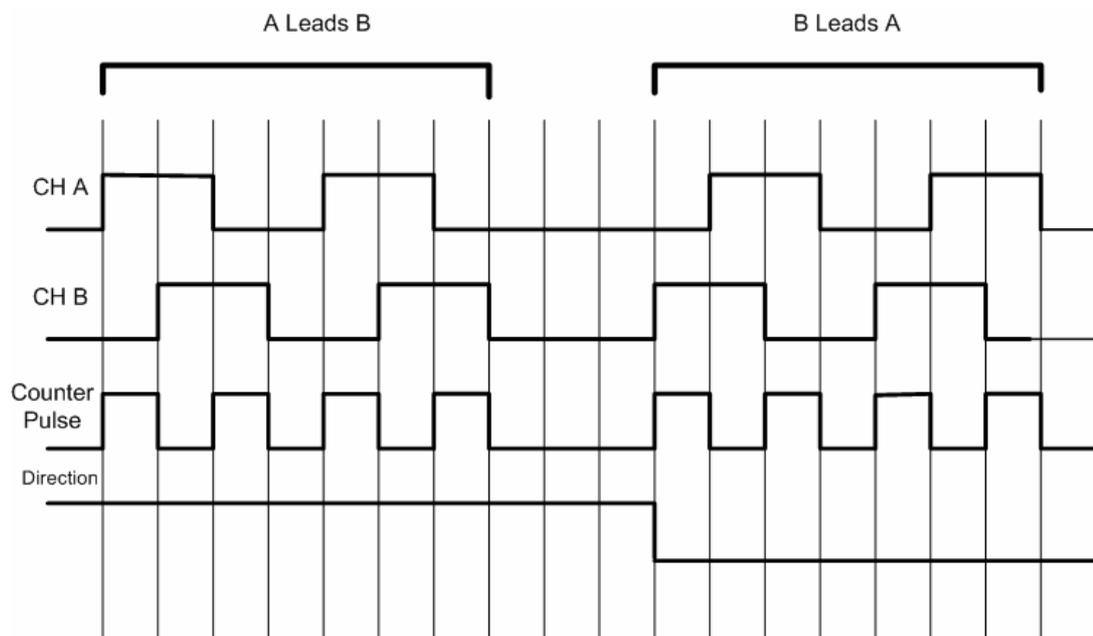


Figure 5.15 Encoder interface circuit.

Table 5.1 Truth table of Encoder Interface Circuit.

A	\bar{A}	B	\bar{B}	A NAND \bar{B}	\bar{A} NAND B	Counter
1	0	0	1	0	1	1
1	0	1	0	1	1	0
0	1	1	0	1	0	1
0	1	0	1	1	1	0

**Figure 5.16** Timing diagram of encoder interface.

It is interesting to note that one can increase the resolution of the decoder further. If one takes a close look at the bit sequences in both channel at the same time in a typical period, four unique states (e.g. when A lead B, the repeating sequence like $\{10_2, 11_2, 01_2, 00_2\}$) can be identified. Consequently, a spatial period ($2\pi/2000$) could be further divided into 4. That is, each region can be easily identified employing combinational logic blocks and every transition from one region to another might be counted out (i.e. 4X interpolation). Fig. 5.17 illustrates the

hardware implementation of such an interpolation scheme. In this paradigm, an extra XOR gate is used to double the frequency of 2X interpolator. Here, an RC network delays one of the inputs to the XOR gate by $\tau \cong RC$. Thus, the XOR generates four pulses (with a width of τ) per one period. The pulse width (i.e. RC parameter) is determined by taking into account the maximum frequency of the encoder along with the other hardware constraints.

Note that the pulses generated by the encoder interface are counted by the 16 bit counter of the microcontroller. One of the I/O pins of the micro-controller is used as the input pin for the direction bit. The implementation details are further discussed in the next section.

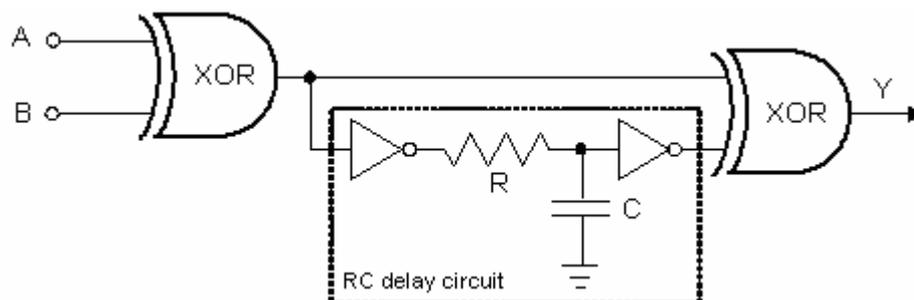


Figure 5.17 Logic circuitry implementing 4X interpolation.

5.3.2 Firmware

Roughly speaking, PIC16F877's firmware is the code controlling all the critical tasks of motion controller such as counting encoder pulses, generating current regulator's command (manipulated input) and sending position data to the computer via RS-232. The developed firmware has five parts:

- Initialization
- Main program
- Control subroutine
- RS-232 subroutine
- Interrupt service routine (ISR).

Initialization part simply sets the initial conditions and activates certain peripherals of the micro-controller. A soft interrupt, which invokes interrupt service routine, is generated at every 4 ms. In ISR, a control flag (CFLAG) is basically set. Other control routine is invoked. In this control routine, the controller sends manipulated input to D/A converter and sets SFLAG which enables RS-232 routine. RS-232 sends incremental position data to PC. Program flow chart is given at Fig. 5.18.

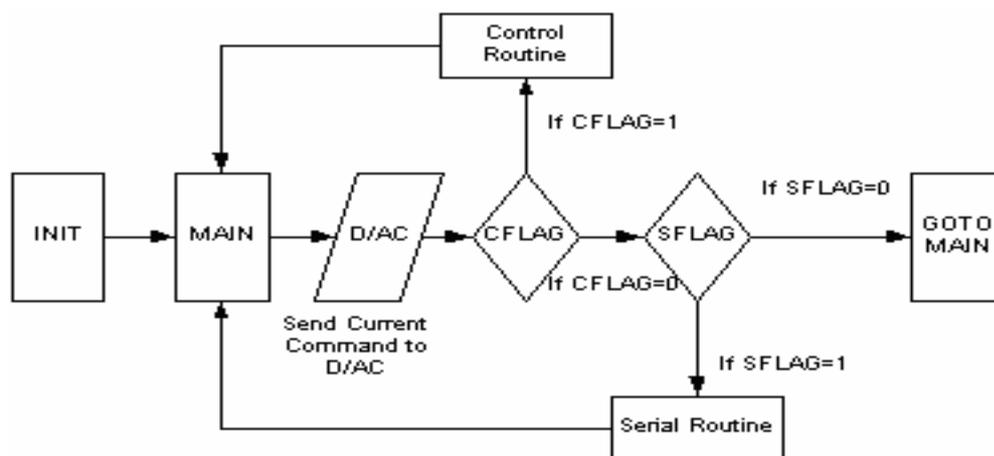


Figure 5.18 Program flow chart.

5.3.2.1 Initiation of Microcontroller

PIC16F877 microcontroller has many peripherals. However, only USART and timer modules are necessary for this operation. At initiation stage these peripherals are enabled. Timer module starts counting and USART module is ready to transmit when initiation routine ends.

The Universal Synchronous Asynchronous Receiver Transmitter (USART) module of PIC16F877 is set to asynchronous mode. In this mode, the USART uses standard non-return-to-zero (NRZ) format (one start bit, eight or nine data bits and one stop bit). The most common data format is 8-bits. An on-chip dedicated 8-bit baud rate generator can be used to derive standard baud rate frequencies from the oscillator.

[54] USART baud rate is set 19200, considering transfer error percentage shown at table 5.2. Inspecting error values shows that although a faster transfer rate could have been chosen increase in error does not justify the fast transfer rate. The optimal solution with maximum baud rate and minimum error is 19.2 K.

Table 5. 2 USART error.

BAUD RATE (K)	F _{osc} 4 MHz	
	KBAUD	% Error
0.3	-	-
1.2	2.404	0.17
9.6	2.404	0.17
19.2	9.612	0.16
28.8	27.798	3.55
33.6	35.714	6.29
57.6	62.500	8.51
HIGH	0.977	-
LOW	250.000	-

Initiation code clears and enables Timer0 module as a timer with interrupt on overflow, so that 4ms timing is accurately generated. Timer1 is cleared and initiated as a counter as an asynchronous counter for counting encoder pulse signal at this stage.

5.3.2.2 Interrupt Routine

Interrupt is generated every 4ms by Timer0 overflow. Interrupt routine is a fast subroutine which only sets a flag, CFLAG that enables control routine. There exists

another approach to timing using interrupt routine. Control program can be executed in interrupt mode. But this method has several drawbacks; this method emphasizes the importance of timing. Slow routines such as serial interface program should not be run in this mode. If control program runs as an ISR and program execution exceeds sampling time, a timing error will occur because of the collision of interrupts. Also running control program as an interrupt means that there can be no other interrupts enabled, in order to avoid interrupt collision. There is one advantage of this method; ease of programming. However, proposed method, using interrupts only for setting flags, is far better. There will be no interrupt collisions, because interrupt routine takes only a few clock cycles to complete. As an other added benefit, fast interrupt means that there will be no timing problems in non interrupt routines. Flow chart of interrupt routine is given in Fig. 5.19

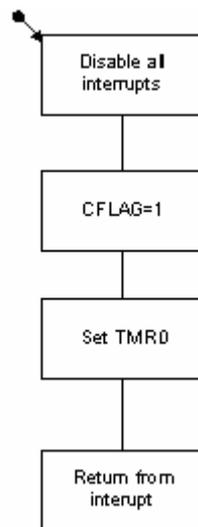


Figure 5.19 Interrupt routine flow chart.

5.3.2.3 Control Routine

Control routine fetches 16-bit Timer1 result as incremental position info and 16-bit signed command from command generator in PIC and then calculates 16-bit signed error. Then via the control equation a 16 bit manipulation signal is generated. This

signal is limited in magnitude and converted to current regulator reference, which is 8 bit signed data. Then this data is send to current regulator through D/AC. Figs 5.20 and 5.21 show controller routine flow chart. Note that PIC is an 8 bit microcontroller that does not have support for floating point nor hardware multiplication. Thus, calculated gains generated by using mathematical procedures to 0.5% numerical precision, by using powers of 2 as stated at Chapter 4. For more information, refer to firmware source code at Appendix F.

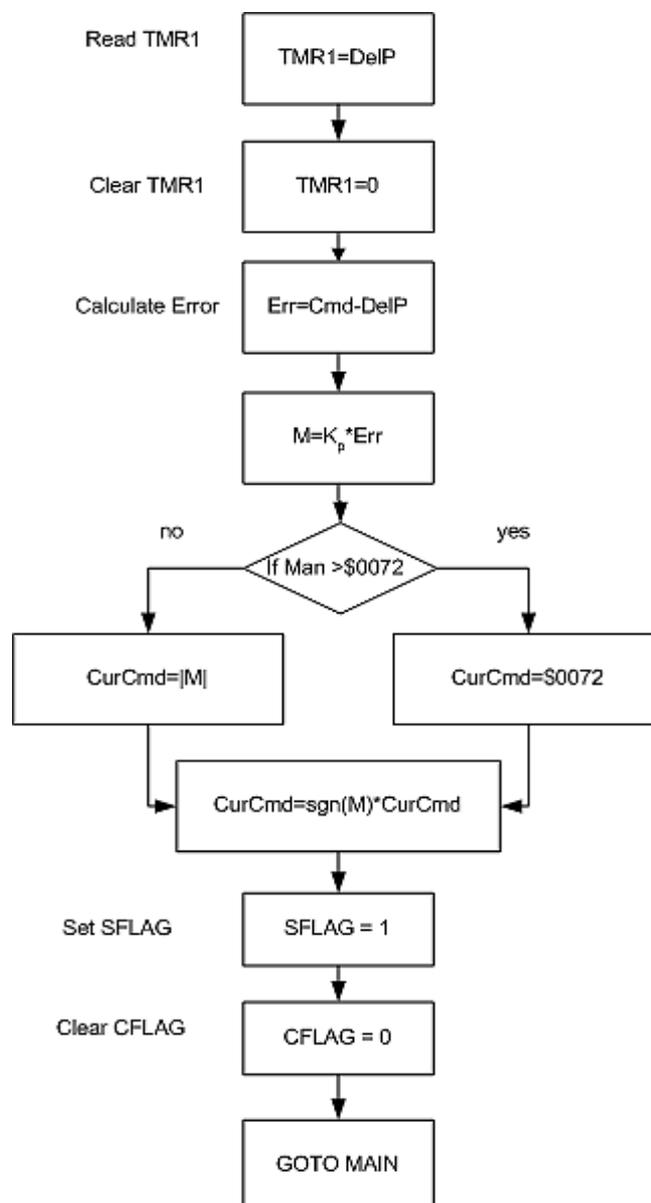


Figure 5.20 Control Routine of incremental velocity controller.

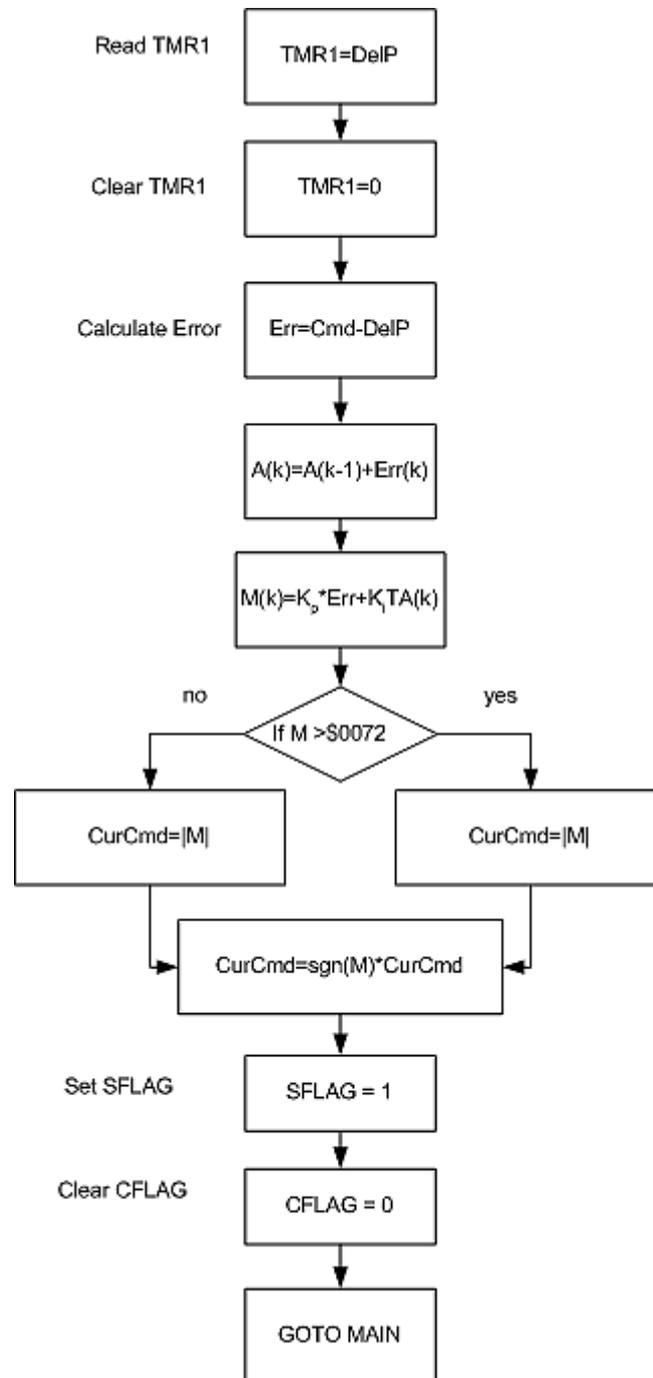


Figure 5.21 Control Routine of incremental position controller.

5.4 Performance Tests

Proposed motion controller is shown in Fig. 5.22 where important components are all marked up on the figure. To test this system, three different reference inputs (commands) are considered:

- Trapezoidal waveform
- Bipolar square wave with dead zone
- Bipolar square wave with no dead zone (e.g. alternating step inputs)

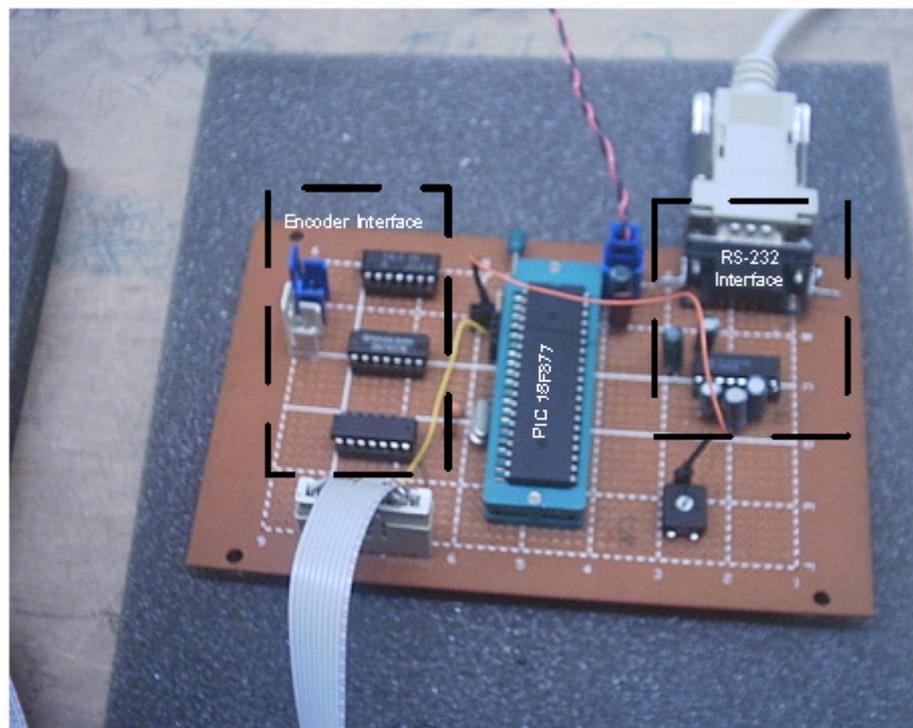


Figure 5.22 Controller circuitry.

Incremental position of the motor is sent through RS-232 to the PC. The (unsigned) integer position data are then converted to decimal and plotted in MATLAB

computing environment. All results are presented in Figs 5.23 through 5.34. For convenience, the velocity command is plotted out as the light colored waveform.

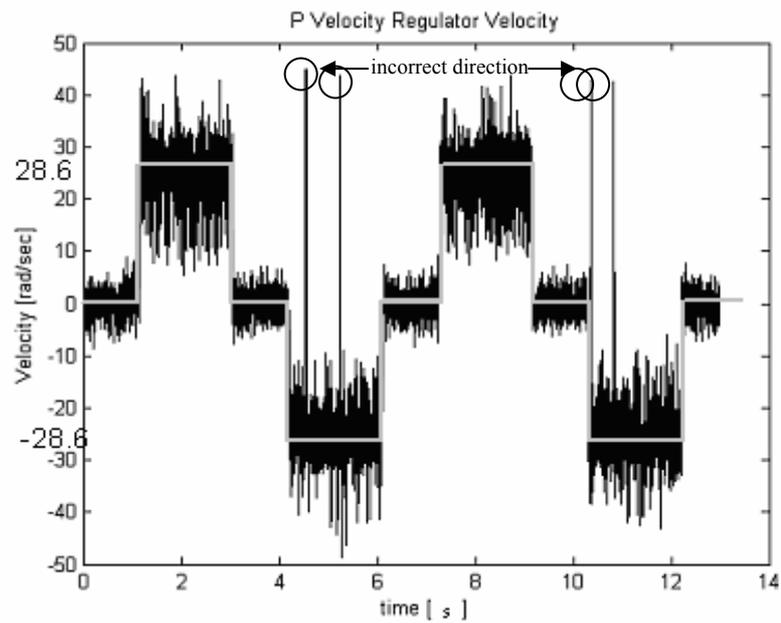


Figure 5.23 Velocity output of incremental velocity controller.

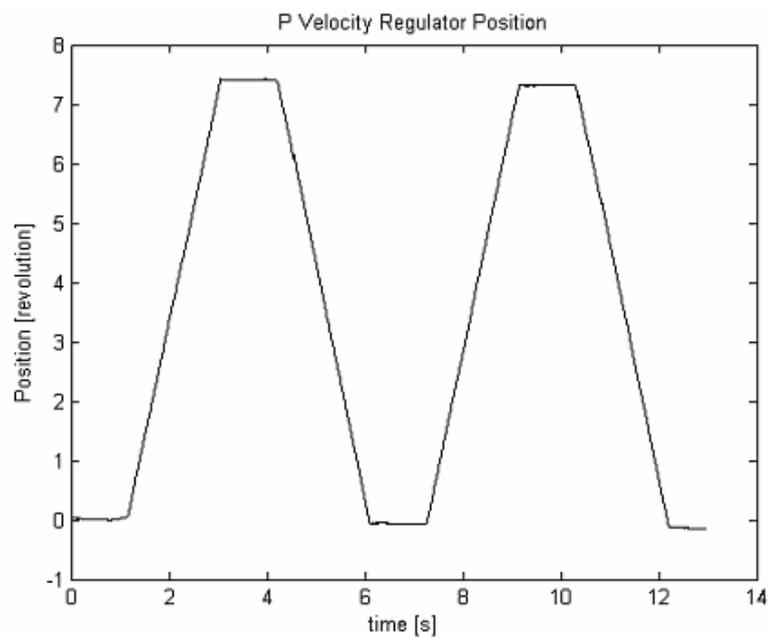


Figure 5.24 Position response of incremental velocity controller.

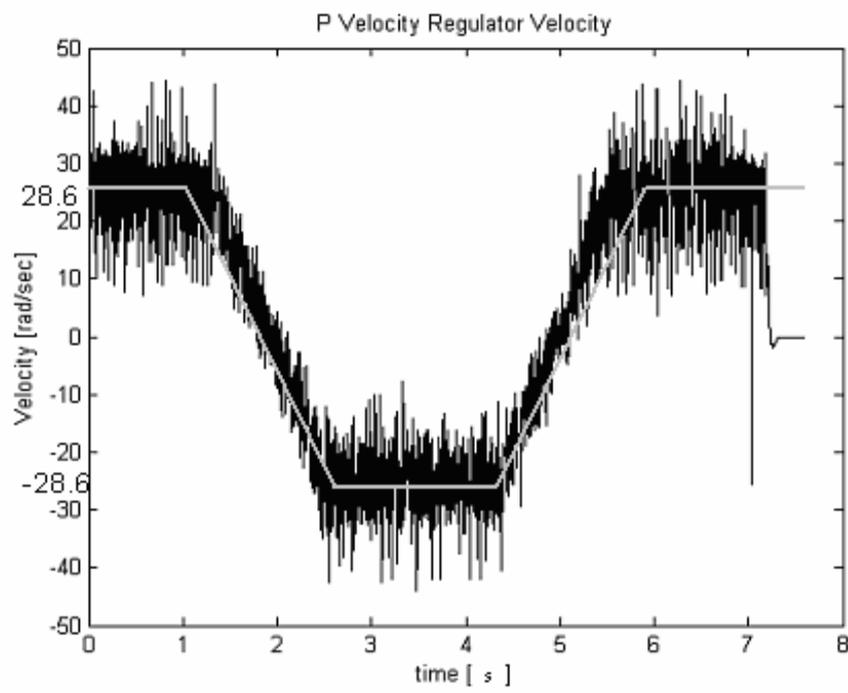


Figure 5.25 Velocity output of incremental velocity controller.

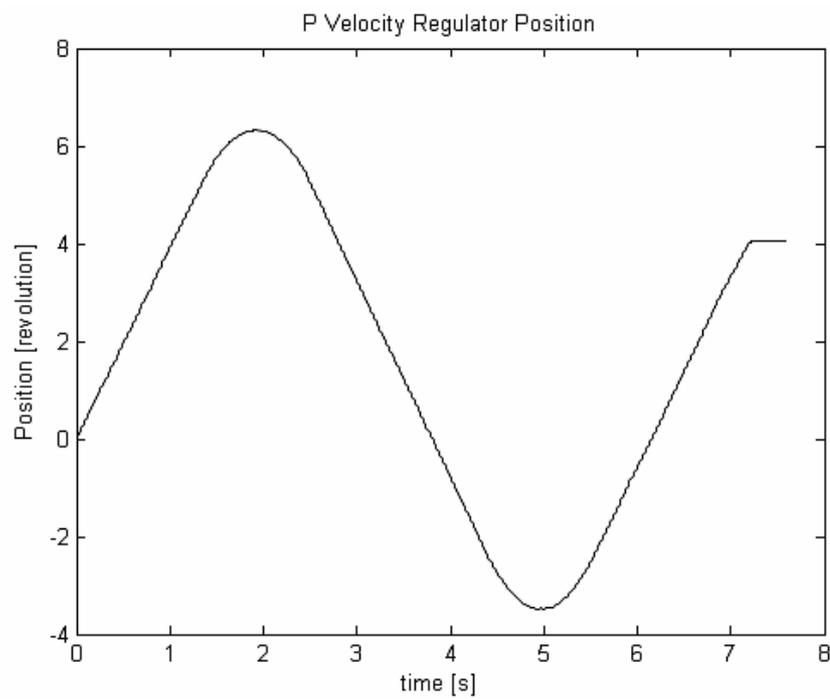


Figure 5.26 Position response of incremental velocity controller.

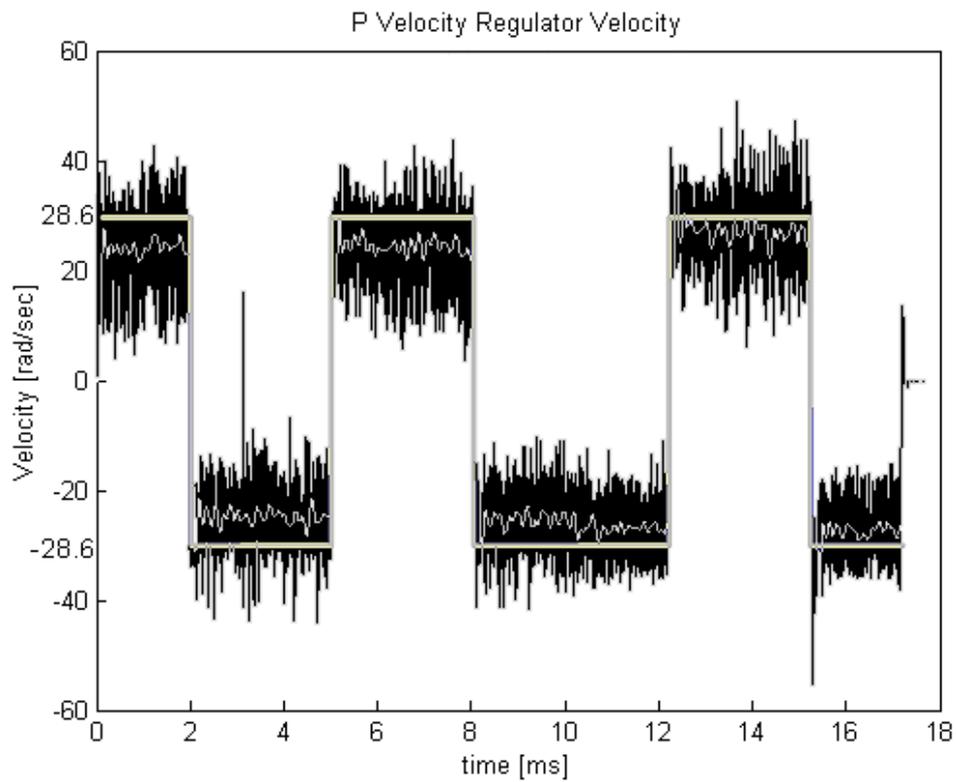


Figure 5.27 Velocity output of incremental velocity controller.

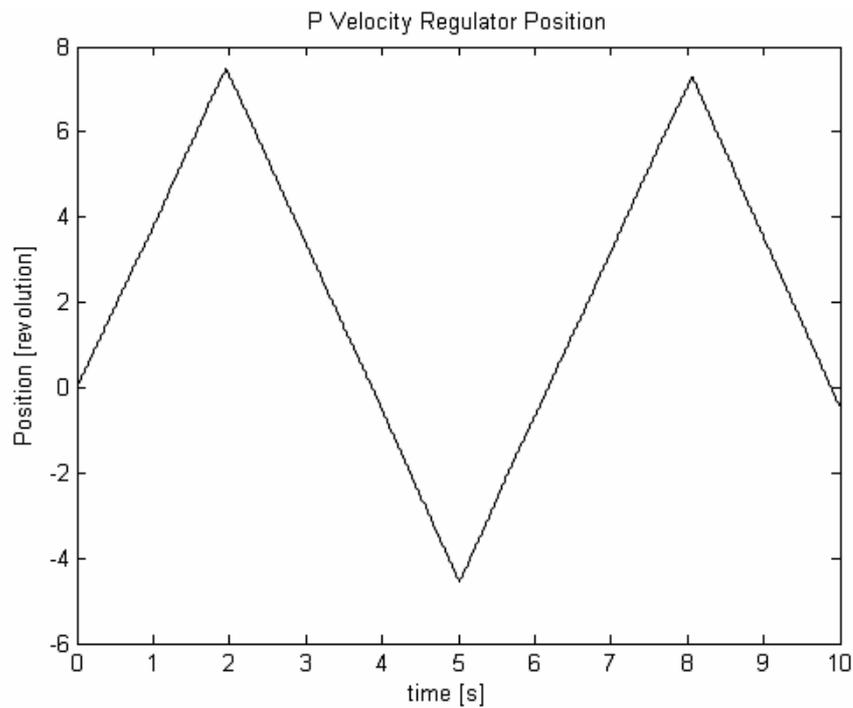


Figure 5.28 Position response of incremental velocity controller.

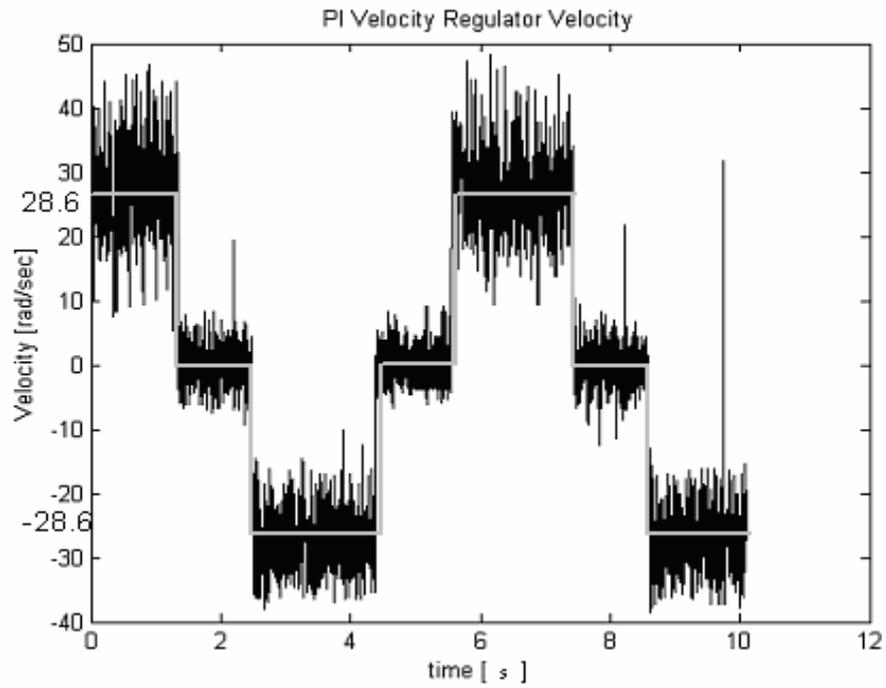


Figure 5.29 Velocity output of incremental position controller.

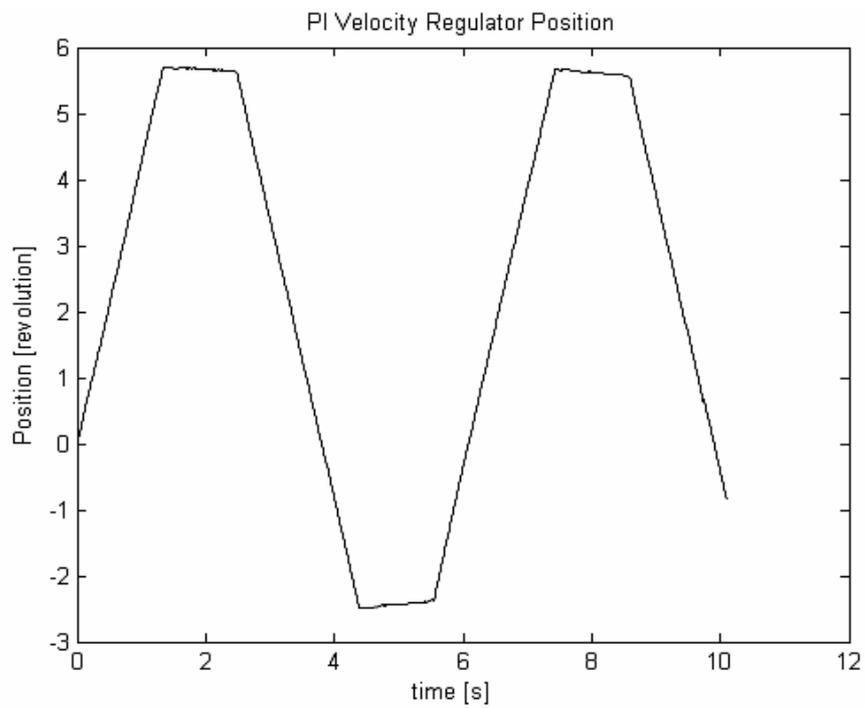


Figure 5.30 Position response of incremental position controller.

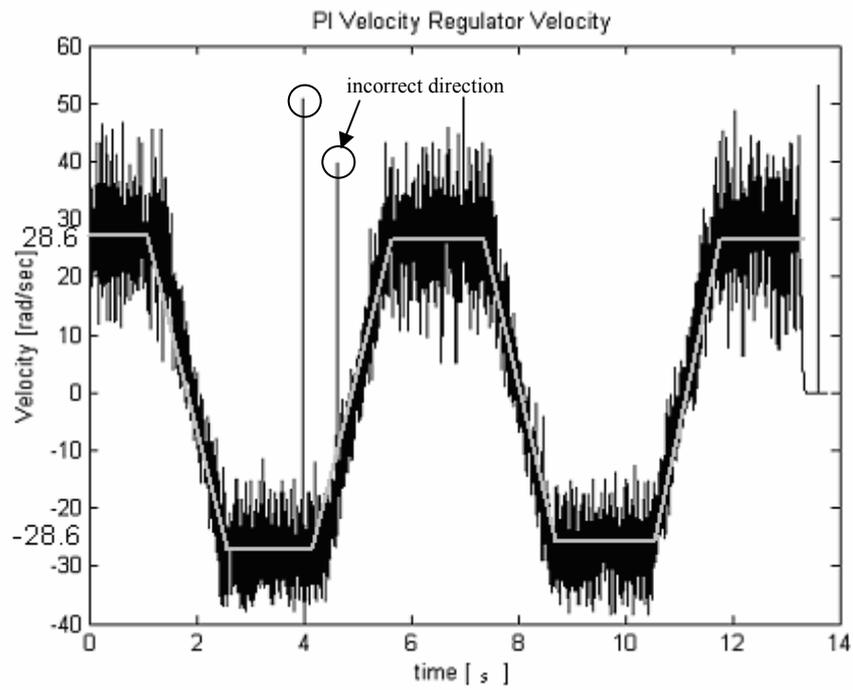


Figure 5.31 Velocity output of incremental position controller.

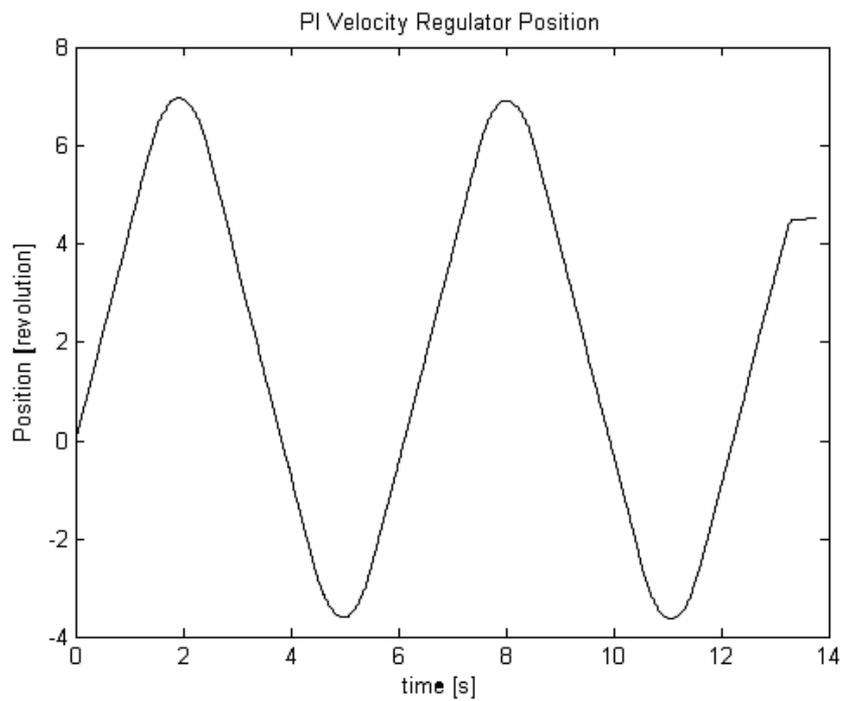


Figure 5.32 Position response of incremental velocity controller.

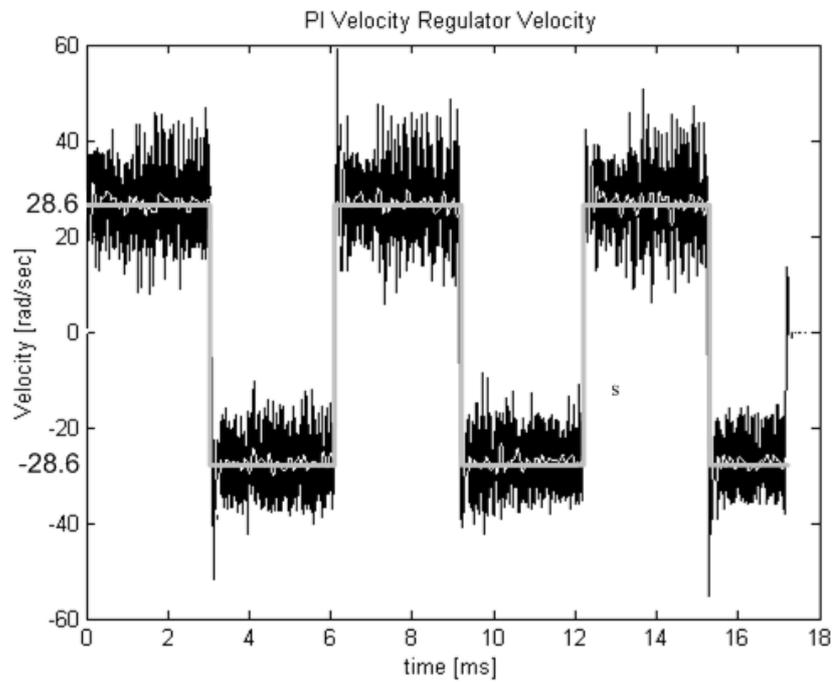


Figure 5.33 Velocity output of incremental position controller

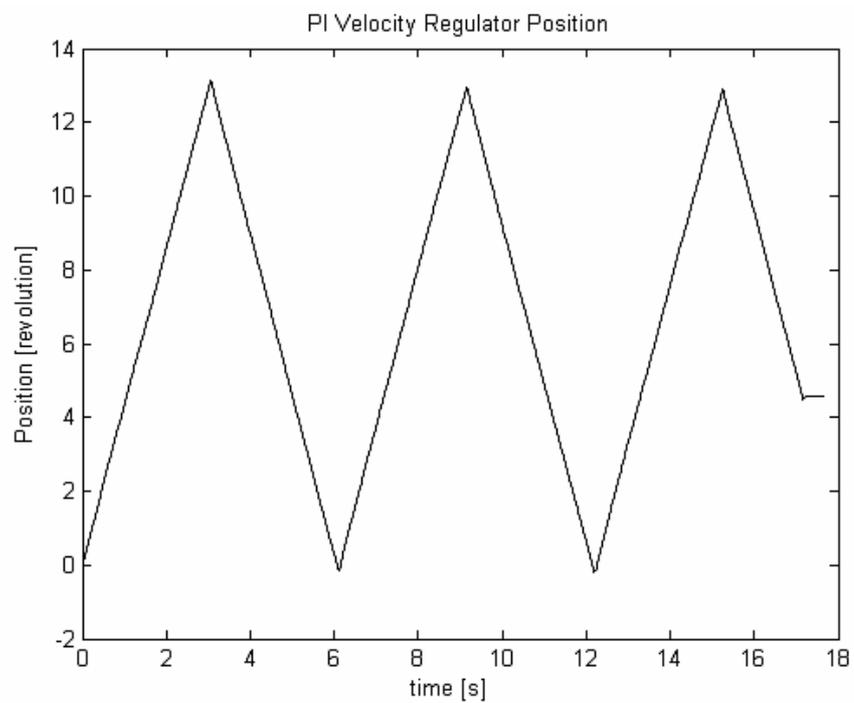


Figure 5.34 Position response of incremental velocity controller

A large amount of data are collected through these tests, the initial points of the plots presented here are selected arbitrarily right from the collected data. The velocity output is integrated to find the position with zero initial condition, however because of arbitrary selection of initialization point (i.e. $\theta = 0$ point), there is a position offset in some of these position plots. Velocity outputs do not have such characteristics but it can be seen that there is an oscillation on velocity. There are three apparent reasons for that. The first one is quantization error which is generated at counter interface. The second one is badly worn out (built in) gearbox of the motor which generates disturbance. The third and the last reason is stick slip friction, whose effect can be observed at Figs. 5.23 and 5.29 where there exists an oscillation around zero velocity. Its effect can even be heard through motor as a result of acoustic emission at 250 Hz. Because of this effect, in Figs 5.24 and 5.30 there is a drift at position as a result of this oscillation in velocity. Another error source is the transmission errors through RS-232. Also, there are sign errors on velocity output, as observed at Figs. 5.23, 5.25, 5.29 and 5.31 where there are visible sign errors on direction. This is due to the unshielded encoder cable connected to the encoder interface; picking up noise and causing coincidental sign changes from time to time.

As can be seen from the figures that incremental velocity controller tracks its reference quite well. Not surprisingly, as shown in Fig. 5.27, where the filtered output is also shown, P velocity controller has a steady state error of approximately 5 rad/s. As expected inspecting Fig. 5.33 shows that incremental position controller (incremental PI velocity controller) has no steady state error.

5.5 Conclusion

In this chapter two simple but effective industrial de facto standard motion controllers are designed and implemented by using PIC16F877 and various peripheral systems.

Both controllers output has oscillations at steady state, which is mostly due to the quantization errors. Also dynamic performance of all controllers discussed are quite satisfactory from the stand point of DC motor control for industrial applications. It's a well known fact that to improve the performance of the controller an accurate as well as high bandwidth position sensor is required. This is unfortunately not the case in this study.

Experiments are conducted on a DC motor with no load condition. Motor driver should be tested at varying load and different operating conditions to assess its performance. However theoretical studies showed that (as indicated in Chapter 2) these controllers has good disturbance rejection properties if designed accordingly.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

The aim of this thesis was to investigate motor motion control techniques and to gain technical know-how (expertise) for building an industry-standard motor motion controller. In order to accomplish this goal, a DC motor driver, which constitutes a current controlled voltage source along with a simple velocity and position controller, were designed and implemented. This motion controller was rigorously tested on a 50W DC motor to assess the effectiveness of designed system.

In this study, two current regulator topologies were investigated; a discrete-time current regulator and a continuous time current regulator. It has been observed that a Hall effect sensor is more suitable for current sensing because it provides galvanic isolation from the power circuit. Of the two regulators designed, discrete time current regulator is found to be unsuitable due to erratic behavior of the microcontroller's A/D converters. Consequently, a sample and hold amplifier is needed for a proper A/D conversion because the corresponding circuitry on PIC16F877 does not work at excitation frequencies higher than 100 Hz. As a matter of fact, the continuous-time current regulator, which can generate PWM frequencies in the range of 3kHz to 300 kHz, exhibited excellent control performance on all tests and it is found suitable for industrial motion control applications.

In this research, various discrete-time motion controllers were also designed and tested. These designs included a position and a velocity controller. The accompanying incremental motion controllers were tested on different (standard) motion commands. Experimental results showed that all of the motion controllers worked satisfactorily in almost every case. In these experiments, standard motion commands were directly generated by microcontroller and results were sent to PC via a RS-232 communications port for further analysis. Table 6.1 summarizes the attributes of two motor drives considered in this study. To outline the basic features of the drives designed, motor drives from Turkish manufacturers were also shown in that table. The motor drive A employs converter A and similarly motor drive B uses converter B. Notice that the given cost also includes 18% of value added tax. A bill of materials is given at Appendix G.

Table 6.1 Technical attributes of various motor drives.

Motor Drive	Motor Drive A	Motor Drive B	Medel MD:01	SmartDrive PRO 4Q
Max Power	1.5kW	92W	1.5kW	7.5kW
Max. Current	15A	2A	10A	25A
Max. Voltage	100V	46V	220V	380V
Control Strategy	Discrete-time	Discrete-time	Continuous-time	Discrete-time
Cost	\$62,06	\$51,47	\$253,94	\$1.829,00

The motor drives A and B were found astonishingly cheaper than the other. While the motor drive B was suitable for low power applications, such as small robot and autonomous vehicle projects; the motor drive A was better suited for industrial applications such as CNC machinery, low power conveyor belt drives, and industrial automation projects.

Notice that one can easily increase the output power of motor drive A by simply increasing the DC bus voltage from 15V to 100V. Furthermore, the rating of the motor drive A can be enhanced dramatically by replacing the existing power

transistors (IRF530N) of the converter with higher rated switches like MOSFETs (e.g. IRF630) or IGBTs since the gate drivers as well as optical isolation circuitry of motor driver A was designed as a general purpose motor drive. However, note that, the motor drives were tested only in laboratory in ideal conditions (no vibrations, no dust, no external heat sources etc.); for industrial applications motor drives should be further tested to assure reliability.

Motor drives in this study were built on prototyping boards and breadboards. This approach caused significant problems during the design and testing of the system due to unforeseeable short circuits right beneath the breadboard and various parasitic effects. Hence, one can recommend the use of the printed circuit board right after the verification of the design on breadboard.

6.2 Future work

A great deal of design improvement opportunities were spotted during the course of this study. The first improvement opportunity is to increase 16F877 which is operating at 4 MHz. In this study, it has been observed that this operation frequency greatly reduced the PWM resolution and along with the PWM frequency, USART data transmission rate and sampling time. Therefore, PIC16F877A operating at 20 MHz can significantly overcome these bottlenecks. For instance one can literally double the transmission speed 28.8 KBAUD to 56.6 KBAUD with the same transmission error (3.55%).

Another improvement which could be implemented relatively easily over the existing hardware; is to use quadrature-encoder interface chips, such as US Digital's LS7083 or LS7084 with 4× interpolation capability. Thus, one can take fully advantage of a higher resolution position measurement without any glitch.

Originally, the proposed motor drive card was to have an RS-485 port for serial communications. The design strategy is to have an RS-485 connection between different motion controller cards for synchronization and a USB interface for data transmission to PC for command generation. In the proposed plan Microchip's PIC16C745/65, which has USB 1.1 peripheral controller, were to be used as USB interface IC and master controller. Note that according to this scheme all the control loops will be implemented on the microcontrollers of each drive system, where the commands received via USB will be processed and distributed out to the different drives by this master microcontroller. Unfortunately, that can not be implemented due to time restrictions imposed on this project, however the use of those chips as well as enhanced USB controllers (Cypress' EZUSB like...) is suggested for future research.

In order to have an industrial motor drive, some design changes should be made. The most important design change is the use of an industrial-grade microcontroller. There are different microcontrollers or microprocessors even digital signal processors in the market, from manufacturers such as Renesas (a joint venture of Mitsubishi, Hitachi...), Microchip and Motorola. Two microcontrollers Renesas' M16C/6N4 and Microchips PIC18F2332 are found to be suitable for future development of motor drive. Important properties of these microcontrollers are tabulated in Table 6.2.

Table 6.2 Comparison of microcontrollers

	M16C/6N4	PIC18F2331
Operating frequency	20 MHz	40 MHz
I/O	87	24
A/D converter	10 bits	10 bits
D/A converter	8 bits	-
USART	Yes	Yes
CAN	Yes	No
Interrupt	31 ext. 9 int. 4 software, 7 levels	22 7 level
Price	\$20-\$30	\$7

Considering the available laboratory equipment; the available programmers and knowledge on PIC microcontrollers; another Microchip product PIC18F2331 is recommended. The important aspects of this microcontroller can be highlighted as follows:

- 6 individual 14-bit PWM channels with programmable dead-time
- quadrature encoder interface for motion control applications
- a *high speed* 10-bit AD converter.

Although there are much powerful microcontrollers which feature higher operating frequencies, more peripherals, 32-bit instructions... etc; the 16-bit microcontroller PIC18F2331 with 8×8 hardware multiplication and with a modest price of \$7 is quite adequate for DC motor drive. On the other hand M16C/6N4 can also be used for a motor drive, which has a wide range of peripherals. Thus, a motor drive design using only M16C/6N4 and a power module could be implemented. However, that requires considerable investment as the Machine Tool and Automation Lab doesn't currently have the necessary equipment for programming M16C/6N4.

Another improvement is area is the firmware development. In this study microcontroller is programmed using Assembly language and it is found that this

low level language greatly increases the development time and efforts. Thus, it is suggested that microcontroller should be programmed using a high level language such as C, C++. Additionally the microcontroller should be in circuit programmable to decrease development time. Thus a much complex firmware with fault detection and programmable control parameters could be developed quickly and using an in circuit debugger the firmware could easily be tested.

This project, (MISAG 257) is a TUBITAK sponsored project and this work is considered as a preliminary design for a high-precision motion control system for a coordinate measurement machine. The work shall progress according to these suggestions.

APPENDIX A: MOSFET AS A POWER SWITCH

MOSFETs with appreciable on-state current carrying capability along with off-state blocking voltage capability have been available for power applications since the early 1980s. They have become as widely used as power bipolar junction transistors (BJTs) and are in fact replacing BJTs in many applications where high switching speeds are important [38].

As shown in Fig. A.1, the MOSFET, like BJT, is a three-terminal device where the input (gate) controls the flow of current between the output terminals (drain and source). A generic MOSFET constitutes the fourth terminal called Bulk or Body (B) to adjust the (gate) threshold of the device (V_T). However, for power MOSFETs, the manufacturers directly connect this terminal to the source. Hence, the active manipulation of V_T is not possible in such applications. The output characteristics – drain current i_D as a function drain-source voltage (v_{DS}) with (steady-state) gate-source voltage (V_{GS}) as a parameter – are illustrated in Fig. A.2.a for an n-channel MOSFET. The output characteristics of a p-channel device are essentially the same when the sign conventions in Fig. A.1.b are followed.

In power electronic applications, the MOSFET is used as a switch to control the flow of power to the load. In such applications, the MOSFET spans the i_D - v_{DS} curves from cut-off through the active region to the ohmic region as the devices

turns on and back again when it turns off. The MOSFET is in cut-off when gate-source voltage is less than some threshold voltage (V_T) which is typically a few volts for most MOSFETs. The device is an open circuit and must hold off the power supply voltage (e.g. V_{dc}) applied to the circuit. As shown in Fig. A.2.a, the drain-source breakdown voltage (BV_{DSS}) must be larger than the applied drain-source voltage to avoid (avalanche) breakdown and the resulting high power dissipation.

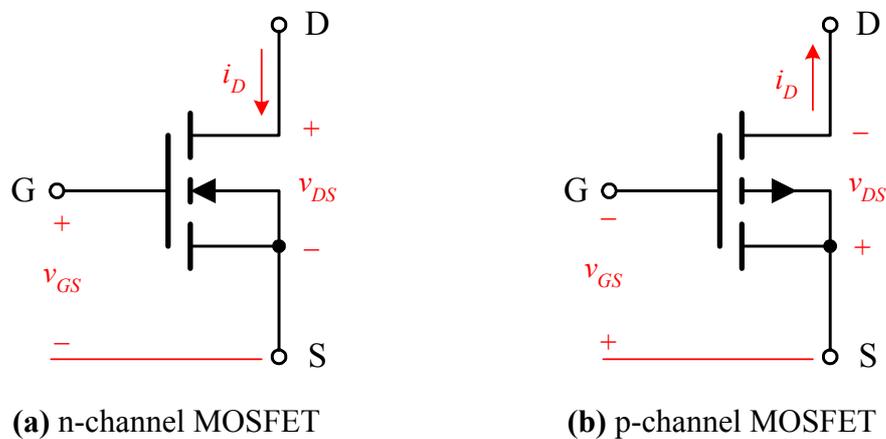


Figure A.1 Circuit symbols and sign conventions for various MOSFET devices.

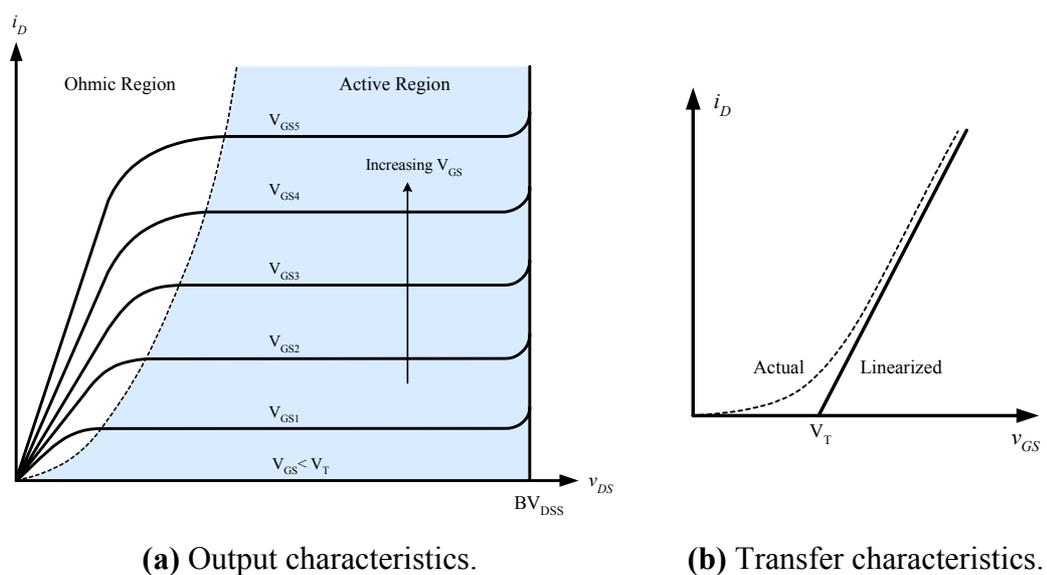


Figure A.2 Current-voltage characteristics of n-channel MOSFET.

When the device is driven by a large gate-source voltage, it goes into the ohmic region where drain-source voltage ($V_{DS,on}$) is small. In this region, the power dissipation can be kept within reasonable bounds by minimizing $V_{DS,on}$ even if the drain current is fairly large. The MOSFET is said to be in ohmic region when

$$v_{GS} - V_T > v_{DS} > 0 \quad (\text{A.1})$$

In the *active* region, the drain current is independent of the drain-source voltage and depends only on the gate-source voltage. The current is sometimes said to have saturated, and consequently the region is referred to as saturation region. When the device is in the active region, the relationship between the drain current and the gate-source voltage (commonly termed to as transfer curve) is quite linear as illustrated in Fig. A.2(b). For the aforementioned reasons, the MOSFET in power electronics applications is never operated in the active region.

It is critical to notice that if adequate reverse voltage is applied to a MOSFET, the device acts like an anti-parallel diode and allows negative current to follow between its terminals. This behavior is simply a consequence of its solid-state physics. Despite the fact that this anti-parallel diode (often times called body diode) is an intrinsic feature, engineers sometimes add extra (fast/ultra-fast recovery) diodes or snubbers (e.g. a capacitor between the drain and the source) for conservative designs.

APPENDIX B: DETERMINING MOTOR PROPERTIES

B.1 Motor Electrical Properties

There are two important motor electrical properties relevant to motor current control:

- Winding resistance (R)
- Armature inductance (L_a)

These parameters affect the dynamics of current loop. These important motor properties are determined by conducting a series of experiments.

B.1.1 Determining Motor Resistance (R)

Examining (4.1) given at Chapter 4 shows that when constant voltage applied to the motor, current flowing through motor is proportional to motor resistance, considering zero velocity condition. A simple experiment where the current on motor is measured at zero velocity gives the motor resistance. The current flowing through motor is measured using a 0.5Ω shunt resistor. The voltage on shunt resistor is read by a multimeter. Motor resistance (R) is found to be 2.5Ω .

Notice that motor resistance is measured by a multimeter while the windings are supposed to be hot, as it is a well known fact that the resistance of the rotor windings do change with increasing winding temperature.

B.1.2 Determining Motor Armature Inductance (L_a)

Motor armature inductance can be found by obtaining frequency response of $I(s)/E(s)$. However in order to obtain this a spectrum analyzer is needed. So another approach is applied.

Employing superposition principle motor can be modeled as shown in Fig. B.1. Thus ripple current can be separated.

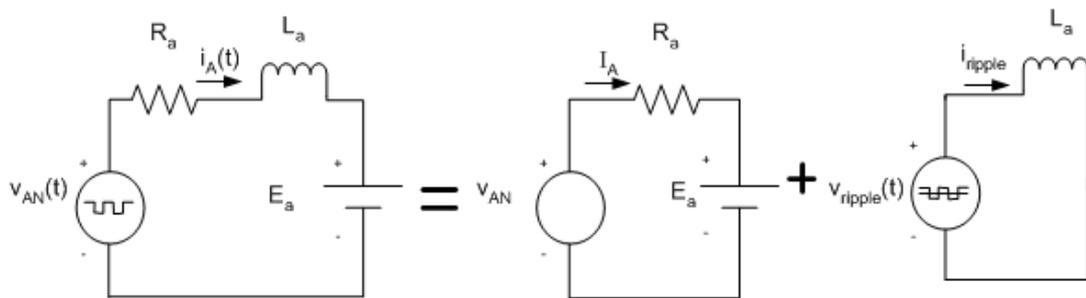


Figure B. 1 Motor model by superposition.

$$i_A(t) = I_A + i_{ripple}(t) \quad (\text{B. 1})$$

$$V_{AN}(t) = V_{AN} + v_{ripple}(t) \quad (\text{B. 2})$$

where

i_{ripple} : saw tooth with zero DC offset

Fig. B.1 shows plot of current and ripple current on motor and voltage values associated with them.

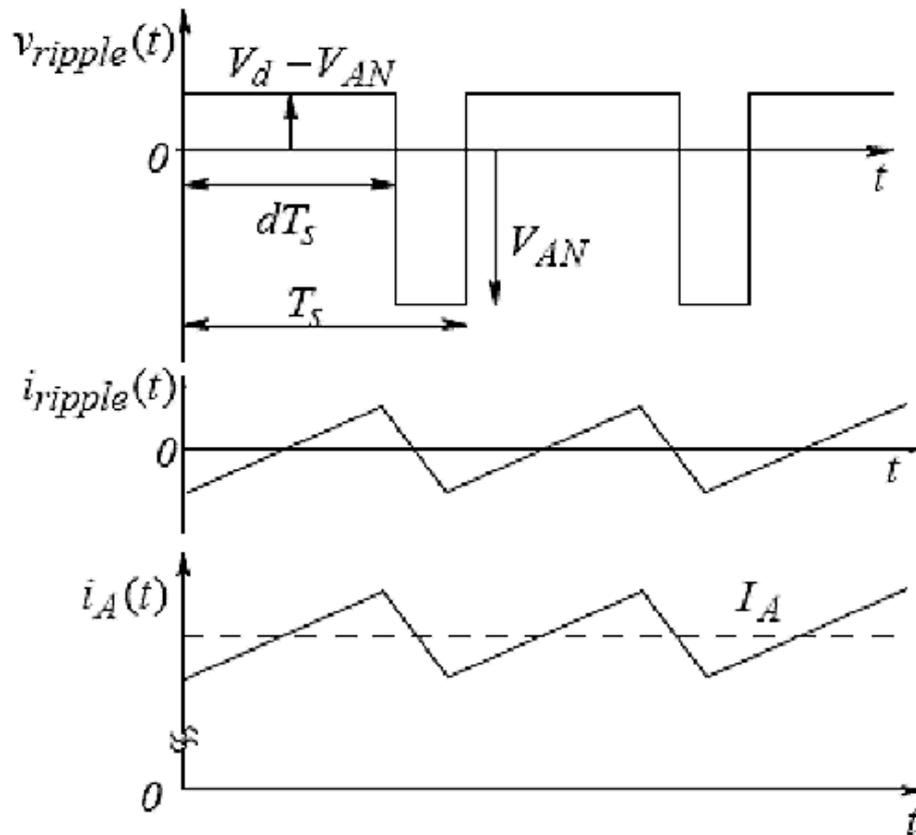


Figure B. 2 Ripple current and voltage [39].

Examining Fig. B.2 and equations given above one can show that

$$\Delta i_A = \frac{\text{volt-seconds}}{L_A} = \frac{(V_d - V_{AN})}{L_A}, \text{ or } \frac{V_{AN}(1-d)T_s}{L_A} \quad (\text{B. 3})$$

(B.3) shows the relation between L_a and current.

In order to find motor current experiment setup described at Section B.1.1 is employed. Motor current is measured through a shunt resistor via an oscilloscope and results are plotted. Then equation (B.3) is applied and armature inductance is found. Next section gives the experimental results.

B.1.2.1 Motor Ripple Current Measurements

A PWM signal is applied to motor leads, to measure motor armature inductance. At Fig.B.3 results for counter clockwise operation and at Fig.B.4 results of clockwise

operation are given. At these figures channel 1 is current with a gain of 2.5V/3A, and channel 2 is input PWM signal.

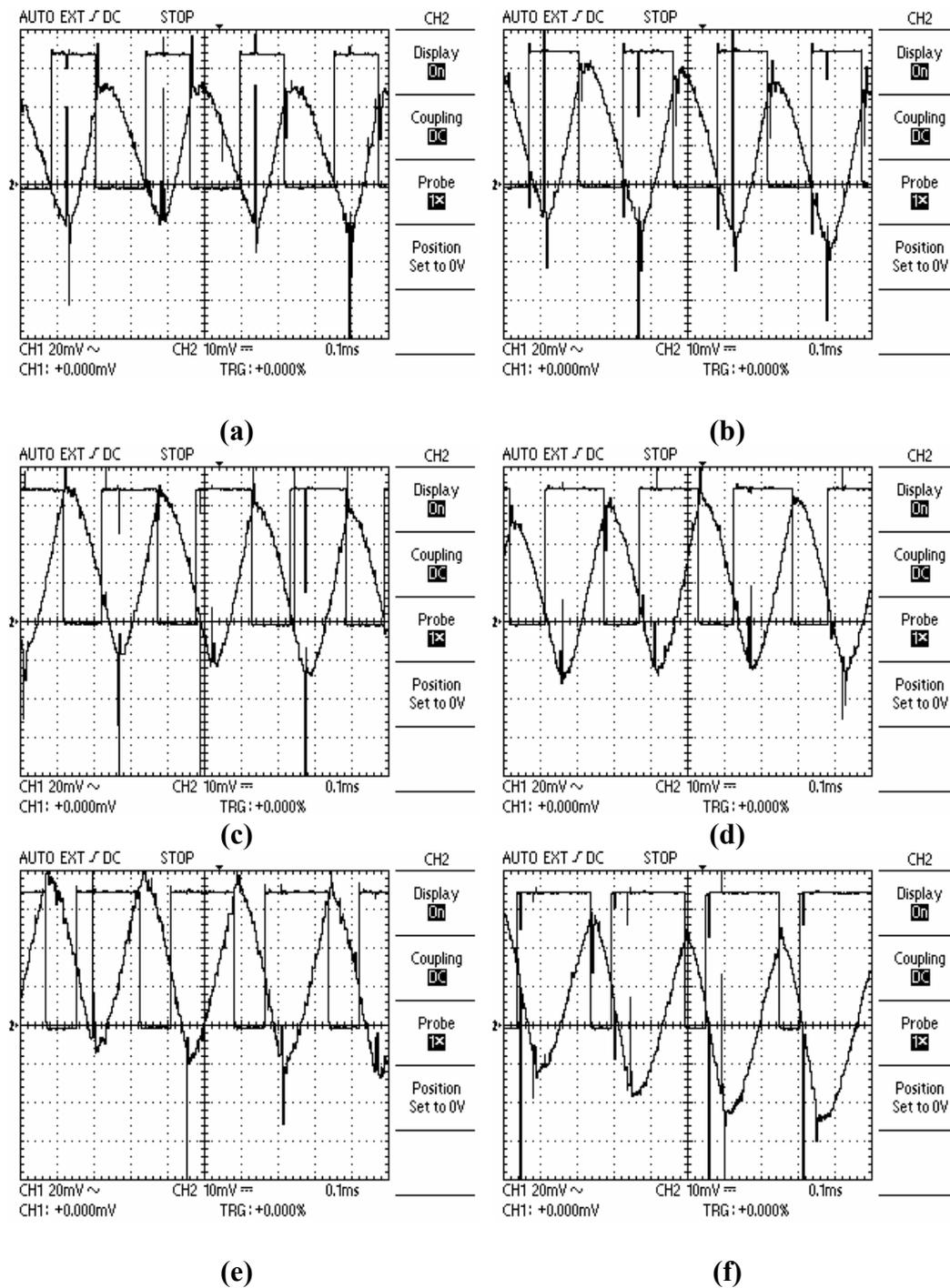
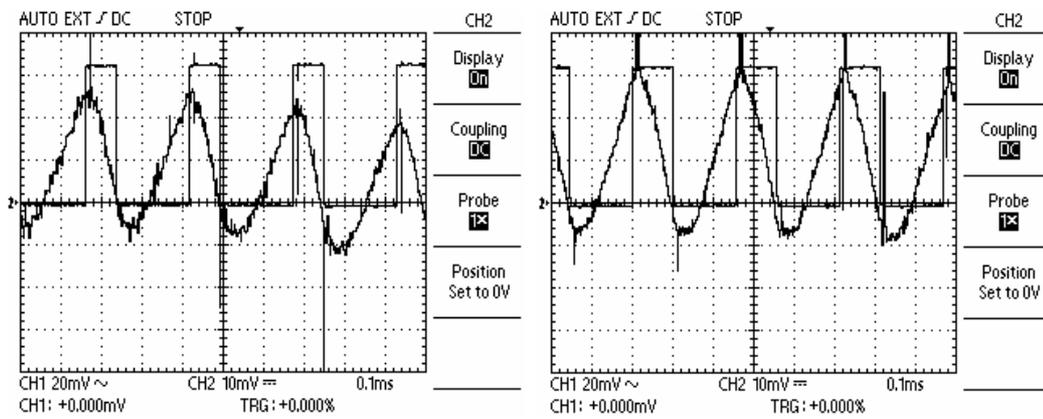
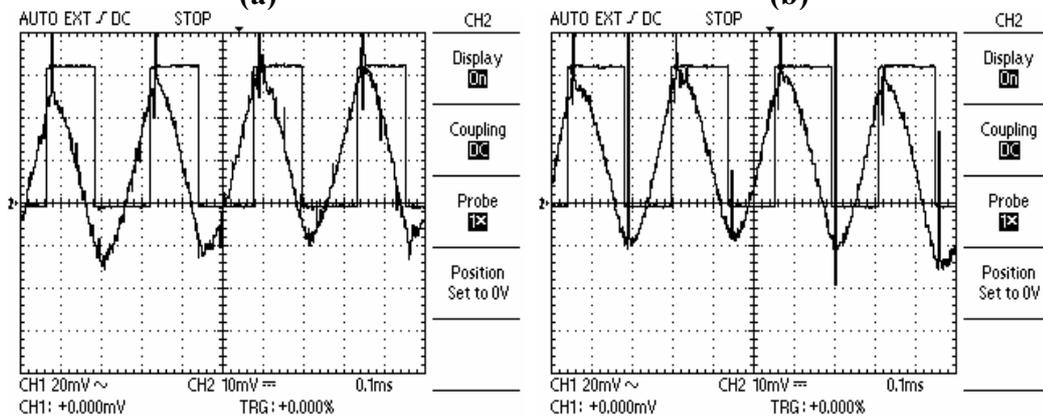


Figure B. 3 Ripple current measurements A.



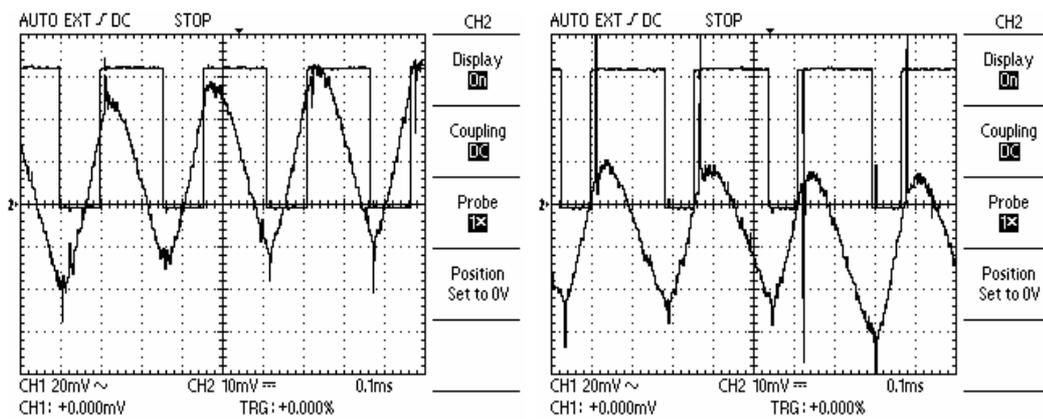
(a)

(b)



(c)

(d)



(e)

(f)

Figure B. 4 Ripple current measurements B.

DC Motor's armature inductance can be calculated via the ripple current using the following expression:

$$L_a = \frac{V_{dc} d(1-d)T_s}{\Delta i_a} = \frac{V_{dc} d(1-d)T_s}{1.2\Delta v_r} \quad (\text{B. 4})$$

where V_{dc} is the DC bus voltage (24.5 [V]); d is the duty cycle; T_s is the PWM period (0.255 [s]); Δv_r is the magnitude of the voltage ripple generated by the LEM module. The following table yields the results:

Table B. 1 Armature inductance experiment results.

Test	d	Δv_r [V]	L_a [mH]
1-CCW	0.47	0.0690	18.8
2-CCW	0.53	0.0840	15.4
3-CCW	0.59	0.0870	14.5
4-CCW	0.62	0.0920	13.3
5-CCW	0.68	0.0960	11.8
6-CCW	0.71	0.0980	10.9
1-CW	0.29	0.0640	16.7
2-CW	0.41	0.0780	16.1
3-CW	0.48	0.0810	16.0
4-CW	0.56	0.0820	15.6
5-CW	0.59	0.0840	15.0
6-CW	0.71	0.0870	12.3
MEAN			14.7

The Fig.B.5 below illustrates the dispersion of estimated inductance values.

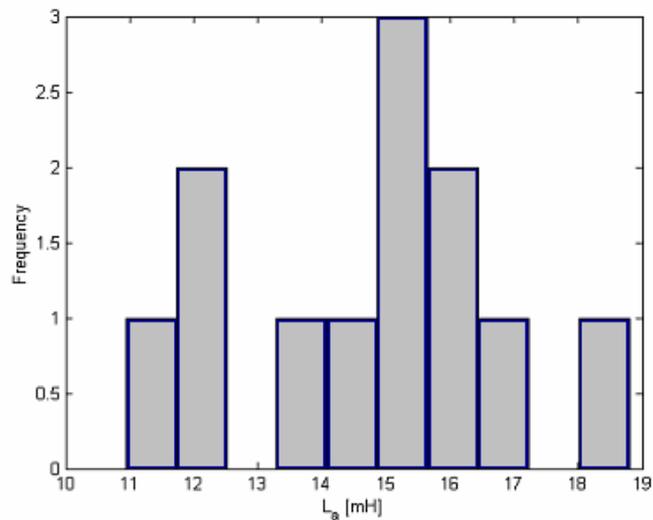


Figure B. 5 Armature inductance measurements distribution.

It should be noted that the L_a value of 14 mH is very high for this motor size. The reason behind this is long cabling employed for motor which is more than two meters.

B.1.2.2 Determining Back EMF Constant (K_e)

Examining (4.1) shows that at constant velocity back emf constant can be determined, if current, velocity and voltage values are known. Thus an experiment setup is constructed. Constant voltage is applied to motor leads and current on motor is measured using a shunt resistor. Motor velocity is measured by an encoder and encoder interface send measured values to PC via RS-232.

Table B. 2 Back emf constant experiment results

DC Voltage (V)	$\omega(\text{rad} / \text{s})$	I (A)
6	120	0.84
9	190	0.98
12	254	1.08
15	325	1.22
-6	114	0.76
-9	178	0.92
-12	235	1.15
-15	299	1.23

At constant velocity (4.1) can be expressed as

$$e - Ri = K_e \omega \quad (\text{B. 5})$$

Rearranging (B.5) gives

$$A = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \dots \\ \omega_i \end{bmatrix}, B = \begin{bmatrix} (|e_1| - Ri_1) \\ (|e_2| - Ri_2) \\ \dots \\ (|e_n| - Ri_n) \end{bmatrix}, x = K_e$$

where $\underline{A} \in \mathbb{R}^{8 \times 2}$, $\underline{B} \in \mathbb{R}^{8 \times 2}$

$$A^T Ax = A^T B \quad (\text{B. 6})$$

$$\underbrace{(A^T A)^{-1} (A^T A)}_I x = (A^T A)^{-1} A^T B \quad (\text{B. 7})$$

Solving (B.7) for data at Table B.1 gives $K_e = 0.039$ [Volts/rad].

B.1.2.3 Determining Torque Constant (K_t)

Torque constant is assumed to be equal to back emf constant if consistent SI units are used in the computation of relevant constants.

$$K_t = 0.039 \text{ [Nm/A]}$$

B.2 Determining Motor Mechanical Parameters

The equation governing the dynamics of a brush type DC motor was given at (B.1). In this equation viscous friction coefficient, coulomb friction and inertia values are unknown and have to be determined using experimental test data of motor.

As stated at Chapter 4 motor torque is a function of voltage acting on motor's leads. In order to estimate viscous friction DC motor is fed by a constant DC voltage source. The current drawn by DC motor is measured by a multimeter. A shunt resistance of 0.47Ω is used. At steady state viscous friction can be expressed as

$$b\omega + T_c \text{sign}(\omega) = K_t i \quad (\text{B. 8})$$

Experiment measurements are at Table (B.3) below:

Table B.3 Friction experiment data

DC Voltage (V)	$\omega(\text{rad} / \text{s})$	I (A)
6	120	0.84
9	190	0.98
12	254	1.08
15	325	1.22
-6	114	0.76
-9	178	0.92
-12	235	1.15
-15	299	1.23

(B.7) can be expressed as:

$$\begin{bmatrix} b \\ T_c \end{bmatrix}_{2 \times 1} \begin{bmatrix} \omega_1 & 1 \\ \omega_2 & 1 \\ \dots & 1 \\ \omega_i & 1 \end{bmatrix}_{4 \times 2} = K_t \begin{bmatrix} i_1 \\ i_2 \\ \dots \\ i_i \end{bmatrix}_{4 \times 1} \quad (\text{B. 9})$$

$$A = \begin{bmatrix} \omega_1 & 1 \\ \omega_2 & 1 \\ \dots & 1 \\ \omega_i & 1 \end{bmatrix}, B = K_t \begin{bmatrix} i_1 \\ i_2 \\ \dots \\ i_i \end{bmatrix}, [b \ T_c] = x \quad (\text{B. 10})$$

where $\underline{A} \in \mathbb{R}^{8 \times 2}$, $\underline{B} \in \mathbb{R}^{8 \times 2}$

$$A^T A x = A^T B \quad (\text{B. 11})$$

$$\underbrace{(A^T A)^{-1}}_I (A^T A) x = (A^T A)^{-1} A^T B \quad (\text{B. 12})$$

Solving (B.11) for data at Table (B.3) gives viscous friction and Coulomb friction.

The result are $b=8.5 \times 10^{-5}$ [Nm/rad/s]; $T_c=0.218$ [Nm/rad/s].

Motor inertia is calculated by assuming motor is a rotor with 70mm diameter and 40mm length.

$$J = m \frac{3d^2 + 4l^2}{24} \quad (\text{B. 13})$$

Solving (B.12) gives $J = 0.0011$ [kg.m²].

APPENDIX C: CURRENT SENSORS

Current sensors are an important part of motion control systems. There are various current sensing methods in the industry. However these methods can be summarized to three main methods which are Shunt Resistor Sensing, Hall Effect Sensors and Current Sensing Transformers. Advantages and disadvantages of these methods are given below at Table C.1.

Table C. 1 Comparison of Current Sensors [56]

Current Sensing Method	Shunt Resistor	Hall Effect	Current Sensing Transformer
Accuracy	Good	Good	Medium
Accuracy vs. Temperature	Good	Poor	Good
Cost	Low	High	Medium
Isolation	No	Yes	Yes
High Current Measuring Capability	Poor	Good	Good
DC Offset Problem	Yes	No	No
Saturation/Hysteresis Problem	No	Yes	Yes
Power Consumption	High	Low	Low
Intrusive Measurement	Yes	No	No
AC/DC Measurements	Both	Both	Only AC

An in depth review of above mentioned sensing methods; Shunt Resistive Current Measurement and Hall Effect sensors, will be given at the following sections.

C.1 Shunt Resistor as Current Sensor

Shunt resistors are popular current-sensing sensors because of their low cost and good accuracy. The voltage drop across a known low value resistor is monitored in order to determine the current flowing through the load. If the resistor is small in magnitude, the voltage drop will be small and the measurement will not have a major effect on the motor circuit. The power dissipation of the resistance makes current shunts impractical for measurements of more than approximately 20 amperes. [56]

The major factors which affect the precision of a shunt are: the temperature coefficient of transfer impedance, power dissipation, low output signal, and in many applications, the need for isolation from the current being measured.[57] There are two methods of shunt resistive current measurement, low side and high side measurements. Also, shunt resistors can provide high and low side measurements as shown in Fig. C.1.

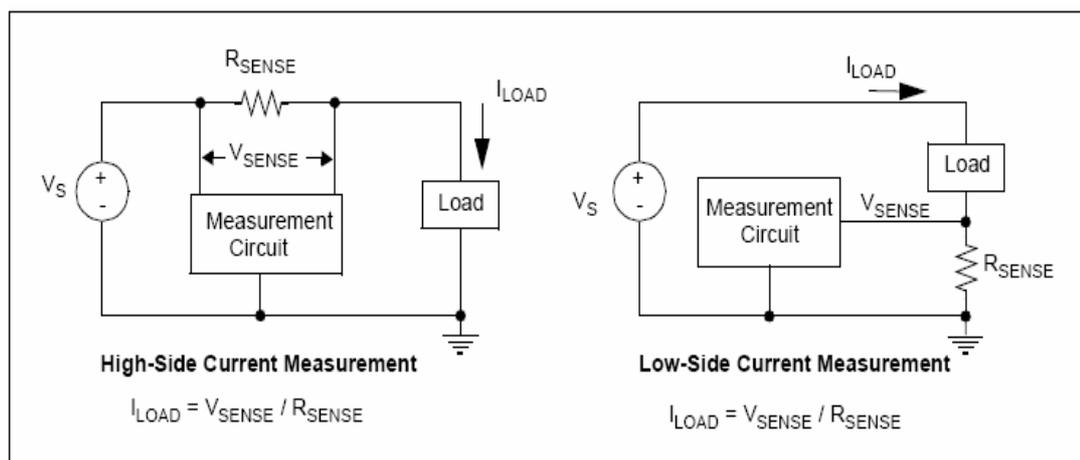


Figure C. 1 High side and low side resistive current shunts[56]

High-side current measurements are the preferred method from a system-integration standpoint because they are less intrusive than low-side measurements. The trade-off with the high-side measurement is that the circuitry is more complex than the

low-side method. High-side resistive shunt measurements will not have a significant impact on the system if the sensing resistor is small and the resulting voltage drop across the shunt is small compared to the supply voltage. In contrast, low-side monitoring disrupts the ground path of the load, which can cause noise and EMI problems in the system. Low-side current measurements are often chosen because low voltage op amps can be used to sense the voltage across the shunt resistor. Note that low-side monitoring is not possible in some applications because the ground connection is made via the mechanical mounting of the motor on the chassis or metal frame. For systems powered via a single wire connection, it may not be practical to insert a shunt resistor between the device and the chassis that functions as the ground wire [56]

For implementation purposes a low side current shunt was chosen. The first reason was that DC motor will be driven by relatively low voltages (15V-24V) thus finding a suitable shunt resistor will be a problem. Also low side measurement was chosen because low voltage op amps can be used to sense voltage across resistor. Also in order to conserve space and not to have two different shunt resistive current sensors for each leg, a rectifier amplifier circuit is designed. This design enabled to have the sensor at series with the motor, thus current sensing at two directions is possible. Fig.s C.2 and C.3 shows the current sensor.

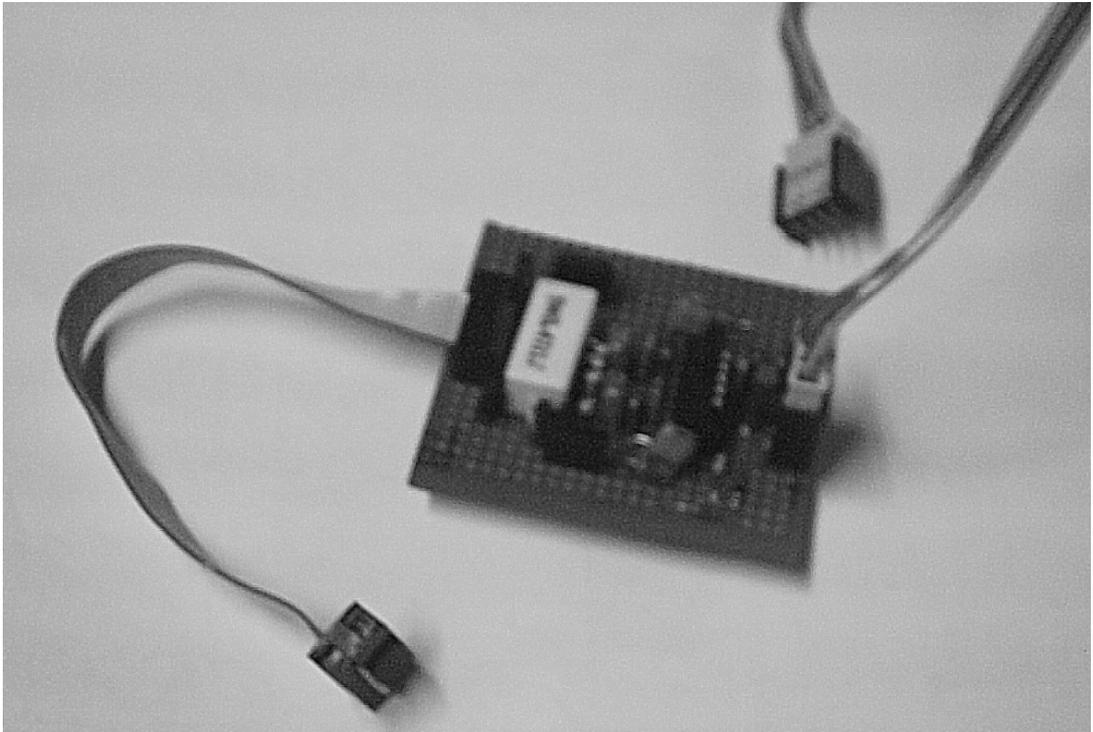


Figure C. 2 Shunt Resistive Current Sensor

C.2 Hall Effect Sensor

When a magnetic field is applied to a conducting or semiconducting material in which a current is flowing, a voltage will be developed across the sides of the material. This is known as the Hall effect.[DC Power] The Hall Effect is based on the principle that a voltage (V_H) is created when current (I_C) flows in a direction perpendicular to a magnetic field (B), as shown in Fig. B.4. Hall Effect current sensors are available in either an open-loop or closed-loop implementation. The closed-loop Hall Effect sensors offer the advantage that their output linearity is better than an open-loop sensor over a wider current measurement range. There are devices available commercially which can be used to measure currents up to 4,000 [A].

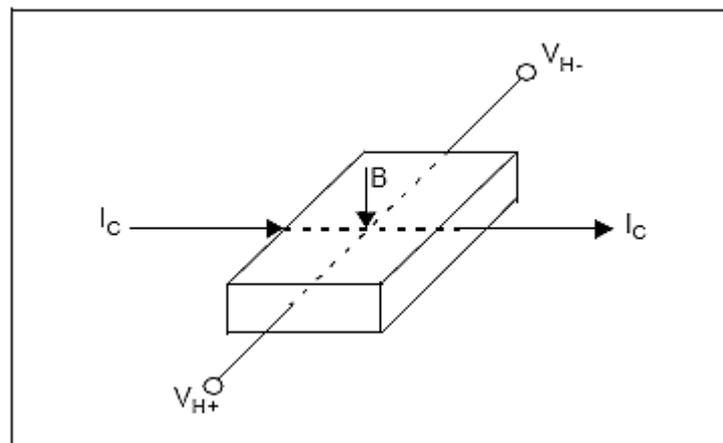


Figure C. 4 Hall Effect Principle

A Hall Effect current sensor consists of a magnetic core with one or more gaps in which Hall effect devices are situated. The current to be measured passes through the center of the core, creating magnetic fields in the gaps. If a constant current is supplied to the Hall effect device, then a voltage is produced that is proportional to the magnetic field in the core. Three major factors effect precision of Hall Effect sensors which are: temperature sensitivity, linearity, leakage flux sensitivity, and sensitivity to control current. There are some compensation methods with electronic

circuitry for Hall Effect sensors; however coefficients are nonlinear and change from device to device.

APPENDIX D: SINE EXPERIMENT RESULTS

Below are the results of sine experiments in order to find bode plot of current compensator.

Table D.1 Sine experiment results.

Frequency (Hz)	30,000	50,000	100,000	150,000	200,000	250,000	300,000
	0,184	0,184	0,184	0,184	0,184	0,184	0,184
input	0,360	-0,380	-0,380	0,380	-0,380	-0,400	0,390
	-0,370	0,370	0,380	-0,380	0,380	0,390	-0,390
	0,360	-0,370	-0,380	0,380	-0,380	-0,390	0,390
	-0,370	0,370	0,380	-0,380	0,380	0,390	-0,390
	0,360	-0,370	-0,380	0,380	-0,380	-0,380	0,390
	-0,380	0,370	0,380	-0,380	0,380	0,390	-0,390
Output with offset	0,160	-0,550	-0,560	0,160	-0,540	-0,550	0,170
	-0,530	0,180	0,200	-0,550	0,180	0,180	-0,550
	0,160	-0,560	-0,540	0,160	-0,550	-0,560	0,170
	-0,530	0,160	0,200	-0,540	0,180	0,200	-0,550
	0,160	-0,550	-0,550	0,180	-0,550	-0,550	0,200
	-0,560	0,160	0,200	-0,540	0,200	0,190	-0,540
Output without offset							
	0,344	-0,366	-0,376	0,344	-0,356	-0,366	0,354
	-0,346	0,364	0,384	-0,366	0,364	0,364	-0,366
	0,344	-0,376	-0,356	0,344	-0,366	-0,376	0,354
	-0,346	0,344	0,384	-0,356	0,364	0,384	-0,366
	0,344	-0,366	-0,366	0,364	-0,366	-0,366	0,384
	-0,376	0,344	0,384	-0,356	0,384	0,374	-0,356
Magnitude ratio							
	0,956	0,963	0,989	0,905	0,937	0,915	0,908
	0,935	0,984	1,011	0,963	0,958	0,933	0,938
	0,956	1,016	0,937	0,905	0,963	0,964	0,908
	0,935	0,930	1,011	0,937	0,958	0,985	0,938
	0,956	0,989	0,963	0,958	0,963	0,963	0,985
	0,989	0,930	1,011	0,937	1,011	0,959	0,913
average	0,954	0,969	0,987	0,934	0,965	0,953	0,932

Table D.1 (continued)

Frequency	350,000	400,000	450,000	500,000	550,000	600,000	650,000
	0,184	0,184	0,184	0,184	0,184	0,184	0,184
input	-0,380	0,380	0,380	0,380	0,380	0,380	0,390
	0,380	-0,380	-0,380	-0,390	-0,390	-0,390	-0,380
	-0,380	0,380	0,390	0,380	0,390	0,380	0,390
	0,390	-0,380	-0,380	-0,390	-0,390	-0,390	-0,380
	-0,390	0,380	0,390	0,380	0,390	0,380	0,380
	0,380	-0,380	-0,380	-0,390	-0,390	-0,390	-0,380
Output with offset	-0,540	0,180	0,180	0,140	0,160	0,130	0,120
	0,190	-0,540	-0,550	-0,530	-0,520	-0,500	-0,480
	-0,550	0,190	0,180	0,130	0,150	0,120	0,130
	0,190	-0,540	-0,540	-0,530	-0,520	-0,500	-0,490
	-0,560	0,190	0,180	0,130	0,150	0,140	0,120
	0,190	-0,540	-0,540	-0,520	-0,510	-0,490	-0,480
Output without offset							
	-0,356	0,364	0,364	0,324	0,344	0,314	0,304
	0,374	-0,356	-0,366	-0,346	-0,336	-0,316	-0,296
	-0,366	0,374	0,364	0,314	0,334	0,304	0,314
	0,374	-0,356	-0,356	-0,346	-0,336	-0,316	-0,306
	-0,376	0,374	0,364	0,314	0,334	0,324	0,304
	0,374	-0,356	-0,356	-0,336	-0,326	-0,306	-0,296
Magnitude ratio							
	0,937	0,958	0,958	0,853	0,905	0,826	0,779
	0,984	0,937	0,963	0,887	0,862	0,810	0,779
	0,963	0,984	0,933	0,826	0,856	0,800	0,805
	0,959	0,937	0,937	0,887	0,862	0,810	0,805
	0,964	0,984	0,933	0,826	0,856	0,853	0,800
	0,984	0,937	0,937	0,862	0,836	0,785	0,779
average	0,965	0,956	0,944	0,857	0,863	0,814	0,791

Table D.1 (continued)

Frequency	700,000	750,000
	0,184	0,184
input	0,390	0,380
	-0,380	-0,380
	0,360	0,380
	-0,390	-0,380
	0,380	0,380
	-0,380	-0,380
Output with offset	0,100	0,090
	-0,430	-0,440
	0,100	0,080
	-0,430	-0,430
	0,090	0,090
	-0,420	-0,430
Output without offset		
	0,284	0,274
	-0,246	-0,256
	0,284	0,264
	-0,246	-0,246
	0,274	0,274
	-0,236	-0,246
Magnitude ratio		
	0,728	0,721
	0,647	0,674
	0,789	0,695
	0,631	0,647
	0,721	0,721
	0,621	0,647
average	0,690	0,684

APPENDIX E: RS-232

RS-232 is one of the oldest serial interface standard which was first published in 1969 by Electronic Industries Association (EIA). The aim of RS-232 is to establish an interface between computers and related equipments. In RS-232 context, computer is referred to “Data Terminal Equipment” (DTE) and related instruments are referred to “Data Communication Equipment” (DCE). There are many RS-232 standards developed at different times, the latest one is RS-232E, however most of the applications employ EIA’s RS-232C standard.

In RS-232, characters are sent one by one as a pattern of bits. The most common encoding format is the asynchronous start-stop format which uses a "start bit" followed by seven or eight data bits, possibly a "parity" bit, and one or two "stop bits". Thus 10 bits are used to send a single character, which has the nice side effect that dividing the signaling rate by ten results in the overall transmission speed [64]. Fig. E.1 shows signal diagram of RS232.

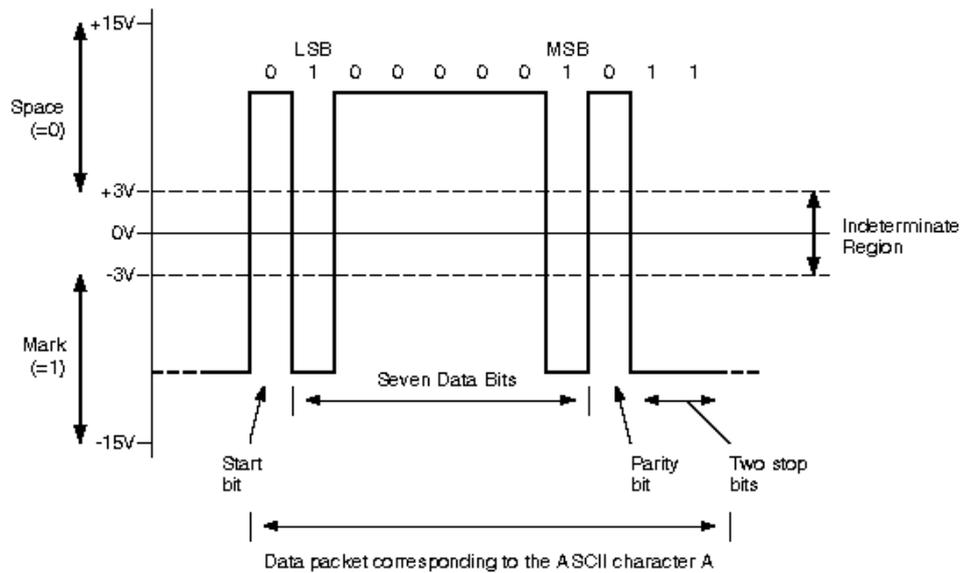


Figure E. 1 RS-232 electrical levels and data format [65].

There are two types of connectors in RS-232 standard, either a 9 pin connection or a 25 pin connection, where a male connector is used for DTE and a female connector is used for a DCE. Figure E.2 shows DTE device connector and Figure E.3 shows a DCE connector [58].

Looking Into the DTE Device Connector

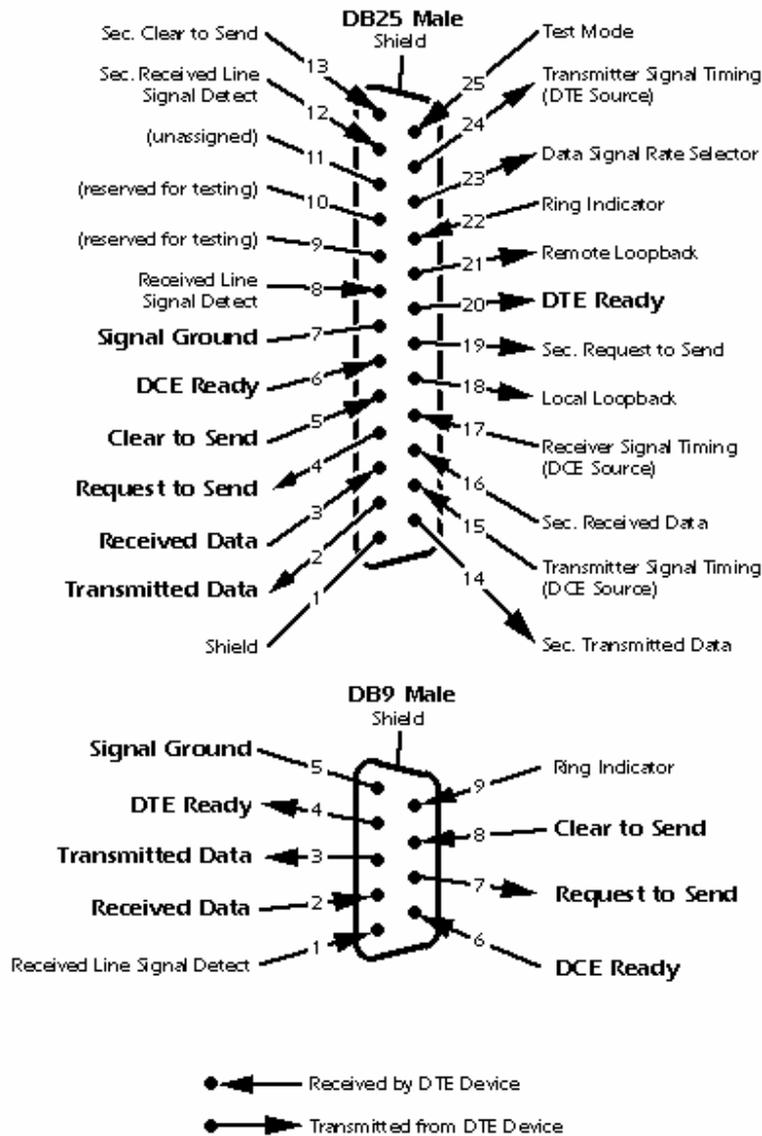


Figure E. 2 DTE device connector

Looking Into the DCE Device Connector

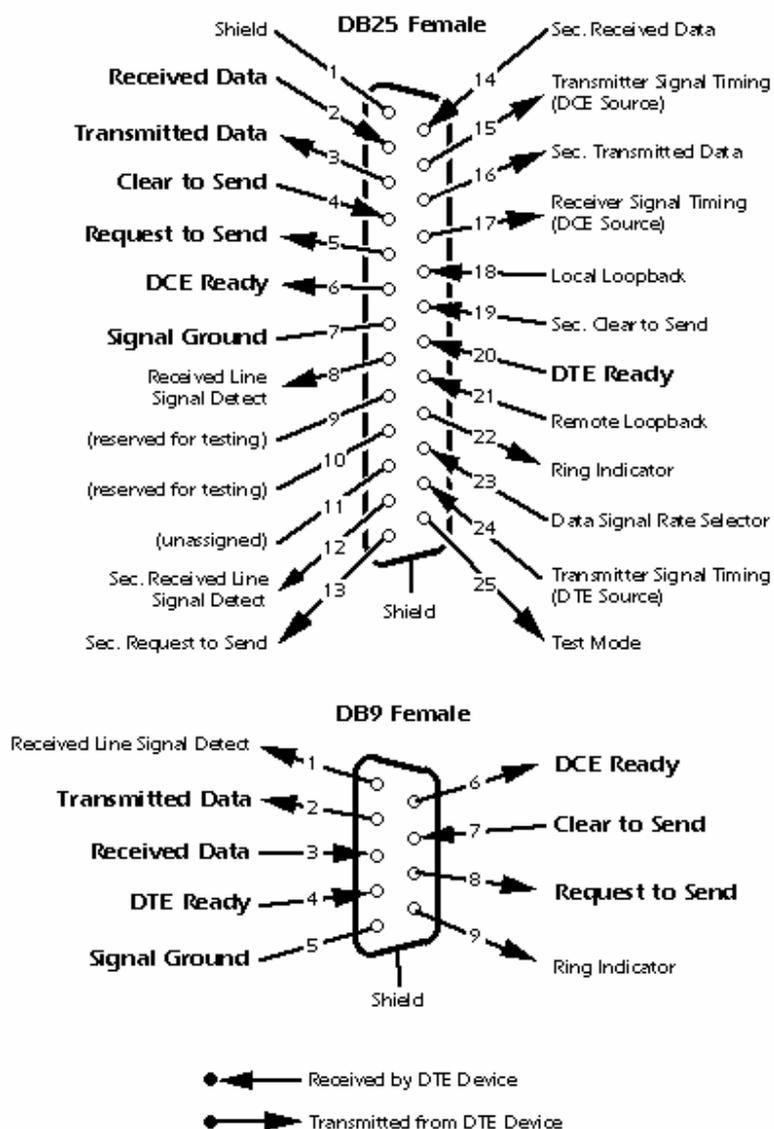


Figure E.3 DCE device connector

However, most of the applications use a 3-wire configuration, which is not specifically mentioned in the standards, where only transmit, receive and DTE ready cables are used.

On the other hand, developers can use a highly complex 25-pins configuration to communicate with appropriate device instead of simple 3-wire.

Another advantage of RS-232 is it has a high noise immunity level. This property is maintained by using “Mark-Space” voltage levels instead of using generic voltage levels specified for digital circuits. In generic digital circuits TRUE is represented between 2.4~5V and FALSE represented between 0~0.4V. In RS-232C, the most widely used RS-232 standard, TRUE corresponds to MARK and represented between 5~15V. Moreover, FALSE corresponds to SPACE and represented between -15~-5V. No signal on communications port means a fault, i.e. the device is shut down, connection is lost etc.

To establish the interface between RS-232 line and generic logic devices, an interface circuit must be used. In RS-232 context, the interface circuit, which transmits signal is called “Line Driver”, which drives the RS-232 line, furthermore, the circuit which receives signal is called “Line Receiver”. Line drivers and receivers only map the input signal from one logic type to another such as from TTL to CMOS or CMOS to TTL etc.[59]

Three important properties should be set by software for RS-232 communications, which are speed, parity, and stop bits. Speed is defined as baud rate which means bits per second such as 19200 or 56000. Parity is a method of verifying accuracy of data, when used by setting parity bit as 1 or 0, can be employed to check transmission errors. There are three parity setups; no parity, even and odd parity. Even and odd parity uses parity bit to set number of 1s in transmission to an even or odd number as the name implies. Stop bits are sent at the end of every byte to synchronize receiving signal hardware.

APPENDIX F: FIRMWARE

The firmware is developed in Assembly language using MPLAB IDE v6.40. The debugging and simulations are conducted on MBLAB IDE and PROTEUS 5.2 Professional.

```

LIST p=16F877
#include <P16F877.inc>
__CONFIG (3D31)
errorlevel -302 ;ignore error when storin to bank1
;Register in operand not in bank 0.
;*** __CONFIG ( _WDT_OFF & _XT_OSC & _PWRTE_ON & _CP_OFF )
;***XT Osc, WDT Off, PWRTE On, CP Off,BOD Off, Low Voltage
Programming Disabled
;***Config will be done from:
;***MPLAP IDE-->Configure/Configuration Bits

;*****variable def*****
variable KRAM=0x20
cblock KRAM

;Program General Inputs
;From Encoder
;CalDelp
;Input: *Hardware* TMR1H/L

;Output:
DelPH ;Delta Position ***Signed***
DelPL

;Variables
T1H
T1L

;***Absolute Position
;CNT
PH2 ;Position High Byte
PH1 ;Position High Byte

PL ;Position Low Byte

;Command Generated by ComGenApp
;Outputs:
CmdH ;Command High ***Signed***

CmdL ;Command Low
;Variables:

```

```

ComCh                                ; For Timing

TCount                                ; For Timing
;
;-----
;Subroutines
;
;-----
;Rs232:
    SHigh
    SLOW
    ;SendData
    RecDataL

;Sign Change Routine
    ChgSignH
    ChgSignL
;
;-----
;Control Algorithms:

; Velocity Loop Variables
;Input: DelPH/L, CmdH/L

;Output:
    IComH
    IComL
    IComLt                                ;temporary reg
;Variables:
    VErrH                                ;Error High Byte

    VErrL                                ;Error Low Byte

    AH                                    ;Accumulator
    AL
    AOldH
    AOldL

    VAD1024H                                ;A divided by 32
    VAD1024L

    VErrD8H                                ;Error divided by 16
    VErrD8L
    VErrD82

    VPErrH                                ;Velocity P controller output
    VPErrL

    VAddH
    VAddL
;Constants:

;End VControl
;
;-----
;Control Var

```

```

VCount                ;Counter for Velocity loop

zCmdH
zCmdL
Temp                  ;Temp, 0-->IntFLAG; Temp, 1-
->Direction, 3 Transmit    ;4 Recieve;
;
_____

wBuffer
CmdBufH
CmdBufL

TEMPK0
TEMPK1
TEMPK2
COUNT                ;

;
_____

endc
variable KRAM=COUNT+1

org 0x00                ;Init. Code
goto init
org 0x04                ;interrupts
goto IntSub

init
movlw    d'6'
movwf    TMR0            ;For 1msec interrupt
bsf     STATUS,RP0      ;Bank1

movlw    b'01000011'    ;datasheet page 50
movwf    OPTION_REG     ;activate Timer0 ; as
                          1:16 timer

movlw    b'10100000'    ;Timer0 overflow
                          ;interrupt
movwf    INTCON         ;interrupt reg.

;PORTA
movlw    0x06            ;All pins are
                          ;digital inputs

movwf    ADCON1
movlw    0xCF
movwf    TRISA          ;PortA input

;PORTB
movlw    b'00010000'
movwf    TRISB         ;PortB input

;PORTC
movlw    b'10000001'
movwf    TRISC         ;PORTC<6:1> Output
PORTC,7,0 input RS232,Counter respectively

```

```

;PORTD
    clrf   TRISD                ;PORTD output

    bcf   STATUS,RP0           ;Bank1

;Counter at PIN PORTC,0
    clrf   TMR1H                ;Clear contents of TMR1H
    clrf   TMR1L                ;Clear contents of TMR1L
    movlw  b'00000111'         ;Set counter F877
datasheet Pg:51
    movwf  T1CON

    movlw  d'6'
    movwf  VCount

;RS232 Init
    movlw  d'12'                ;Baud rate 19.2K
    bsf   STATUS,RP0           ;Bank1
    movwf  SPBRG                ;/
    bsf   TXSTA,2              ;\
    bcf   TXSTA,4              ;/Enable Assyn.

Serial Port
;    bsf   PIE1,4                ;Interrupt Enabled
for Transmit (PIR1,4 will be Interrupt)
    bsf   PIE1,5                ;Interrupt Enabled
for Receive (PIR1,5 will be Interrupt)
    bsf   TXSTA,5              ;Enable transmission
    bcf   STATUS,5
    bsf   RCSTA,7              ;\Enable Assyn Ser.
port
    bsf   RCSTA,4              ;Enable receive

    movlw  0xFF
    movwf  CmdH
    movlw  0xB7
    movwf  CmdL

    clrf   DelPL
    clrf   DelPH
    clrf   PORTD
    clrf   ComCh
    clrf   AH
    clrf   AL
    clrf   AOldH
    clrf   AOldL

goto start

CalDelP                                ;DelP calculation, runs at
every 10ms
;Input:      *Hardware* TMR1H/L
;Output:     DelPH/L
;Variables:  T1H/L

```

```

        clrf   DelPL
        clrf   DelPH

        movf   TMR1H,w
        movwf  T1H
        movf   TMR1L,w
        movwf  T1L

        btfsc  PORTB,4
        goto   DecDelP

        btfss  PORTB,4
        goto   IncDelP

IncDelP
        movf   T1L,w
        addwf  DelPL,f
        btfsc  STATUS,C
        ;
        incf   DelPH,f
        movf   T1H,w
        addwf  DelPH,f

        goto   EndDelPCal

DecDelP
        movf   T1L,w
        subwf  DelPL,f
        btfss  STATUS,C
        ;
        incf   T1H,f
        movf   T1H,w
        subwf  DelPH,f

        goto   EndDelPCal

EndDelPCal
        clrf   TMR1L
        clrf   TMR1H
        return

ComGenApp                                     ;Command Generation Sub.

;Input: ComCh
;Output: CmdH/L

        movf   CmdBufH,w
        movwf  CmdH
        movf   CmdBufL,w
        movwf  CmdL

        movlw  0xFE
        subwf  ComCh,w

```

```

    btfsc STATUS,C
    clrf ComCh

    movlw      0xAF          ;
    subwf     ComCh,0       ;
    btfsc     STATUS,C      ;
    goto      ComP4

    movlw      0x7F          ;
    subwf     ComCh,0       ;
    btfsc     STATUS,C      ;
    goto      ComP3

    movlw      0x30          ;
    subwf     ComCh,0       ;
    btfsc     STATUS,C      ;if ComCH>0xDD
    goto      ComP2
    btfss     STATUS,C
    goto      ComP1

EndComGenApp
    movf     CmdH,w
    movwf   CmdBufH
    movf     CmdL,w
    movwf   CmdBufL

    decfsz   VCount,f      ;increase command
every 3x4ms
    return

    incf     ComCh,f
    movlw   0x06
    movwf   VCount

    return

ComP1                                     ;ComGenPartZero
    btfsc   Temp,5
    goto    ComP1_1
    btfss   Temp,5
    goto    ComP1_2
ComP1_1
    movlw   0x01
    addwf   CmdL,f
    btfsc   STATUS,C
    incf    CmdH,f
    bcf     Temp,5
    goto    EndComGenApp
ComP1_2
    bsf     Temp,5

```

```

        goto EndComGenApp

ComP2

        movlw      0x00                ; 00000011
        movwf     CmdH                ;
        movlw     0x49                ;
        movwf     CmdL                ;
        goto     EndComGenApp        ;

ComP3
        btfsc Temp,5
        goto ComP3_1
        btfss Temp,5
        goto ComP3_2

ComP3_1
        movlw     0x01
        subwf    CmdL, f
        btfss    STATUS, C
        decf    CmdH, f
        bcf     Temp, 5
        goto    EndComGenApp

ComP3_2
        bsf     Temp, 5
        goto    EndComGenApp

ComP4
        movlw 0xFF
        movwf CmdH
        movlw 0xB6
        movwf CmdL
        goto EndComGenApp

; _____

ChgSign
;Sign Change SubRoutine
;Input: ChgSignH/L
;Output:ChgSignH/L

        movf ChgSignL, w
        sublw 0x00
        btfss STATUS, C
        incf ChgSignH, f
        movwf ChgSignL

        movf ChgSignH, w
        sublw 0x00
        movwf ChgSignH

        return

```

```

VControl
;Input: CmdH/L, DelPH/L
;Output: IComH/L [unsigned1024bit, 0A=512; will be ;send to
Analog Current Loop]

        bsf PORTC,4                ;For DEBUGGING

        call ComGenApp              ;CmdH/L    ready
        call CalDelP               ;DelPH/L   ready

;Send Velocity
        movf DelPL,w
        movwf SLow
        movf DelPH,w
        movwf SHigh
;-----

VErrCal

        movf DelPL,W
        subwf CmdL,f
        movf DelPH,W
        btfss STATUS,C
        incf DelPH,W
        subwf CmdH,f

        movf CmdH,w
        movwf VErrH
        movf CmdL,w
        movwf VErrL

VPController
;Input: VErrH/L
;Output: VPErrH/L
;Variables: VErrD32H/L, VErrD16H/L
        movf VErrH,w
        movwf ChgSignH
        movf VErrL,w
        movwf ChgSignL

        btfsc VErrH,7              ;If VErr<0
        call ChgSign

        movf ChgSignH,w
        movwf VErrD8H
        movwf VAD1024H
        movf ChgSignL,w
        movwf VErrD8L
        movwf VAD1024L

;Math calculation code was generated using automated ;program
located at
;www.piclist.com/cgi-bin/constdivmul.exe

```

```

;(1-May-2002 version)

;Integrator
; VAD1024 = VAD1024 * 0.1265
; Temp = TEMPK
; VAD1024 size = 16 bits
; Error = 0.5 %
; Bytes order = little endian
; Round = no

; ALGORITHM:
; Clear accumulator
; Add input / 8 to accumulator
; Add input / 1024 to accumulator
; Move accumulator to result
;
; Approximated constant: 0.125977, Error: 0.413785 %

;   Input: VAD1024L .. VAD1024H, 16 bits
;   Output: VAD1024L .. VAD1024H, 14 bits
; Code size: 23 instructions

;copy accumulator to temporary
    movf  VAD1024H, w
    movwf TEMPK1
    movf  VAD1024L, w
    movwf TEMPK0

;shift accumulator right 7 times
    rlf  VAD1024L, f
    rlf  VAD1024H, w
    movwf VAD1024L
    clrf VAD1024H
    rlf  VAD1024H, f

;add temporary to accumulator
    movf  TEMPK0, w
    addwf VAD1024L, f
    movf  TEMPK1, w
    skpnc
    incfsz      TEMPK1, w
    addwf VAD1024H, f

;shift accumulator right 3 times
    rrf  VAD1024H, f
    rrf  VAD1024L, f
    clrc
    rrf  VAD1024H, f
    rrf  VAD1024L, f
    clrc
    rrf  VAD1024H, f
    rrf  VAD1024L, f

```

```

        btfsc VErrH,7
        goto SubD1024
        btfss VErrH,7
        goto AddD1024

AddD1024
        movf VAD1024L,w
        addwf AOldL,f
        btfsc STATUS,C
        incf AOldH,f

        movf VAD1024H,w
        addwf AOldH,f

        goto EndD1024C

SubD1024
        movf VAD1024L,w
        subwf AOldL,f
        btfss STATUS,C
        incf VAD1024H,f

        movf VAD1024H,w
        subwf AOldH,f
        goto EndD1024C

EndD1024C
        movf AOldH,w
        movwf AH
        movf AOldL,w
        movwf AL

;;Send Absolute Error
;   movf ChgSignH,w
;   movwf SHigh
;   movf ChgSignL,w
;   movwf SLow
;;-----

; VErrD8 = VErrD8 * 2.4035
; Temp = TEMPK
; VErrD8 size = 16 bits
; Error = 0.5 %
; Bytes order = little endian
; Round = no

; ALGORITHM:
; Clear accumulator
; Add input * 2 to accumulator
; Add input / 4 to accumulator
; Add input / 8 to accumulator
; Add input / 32 to accumulator
; Move accumulator to result
;
; Approximated constant: 2.40625, Error: 0.114416 %

```

```

;   Input: VErrD8L .. VErrD8H, 16 bits
;   Output: VErrD8L .. VErrD82, 18 bits
; Code size: 44 instructions

;copy accumulator to temporary
    movf  VErrD8H, w
    movwf TEMPK1
    movf  VErrD8L, w
    movwf TEMPK0

;shift accumulator right 2 times
    clrc
    rrf   VErrD8H, f
    rrf   VErrD8L, f
    clrc
    rrf   VErrD8H, f
    rrf   VErrD8L, f

;add temporary to accumulator
    addwf VErrD8L, f
    movf  TEMPK1, w
    skpnc
    incfsz      TEMPK1, w
    addwf VErrD8H, f

;shift accumulator right 1 times
    rrf   VErrD8H, f
    rrf   VErrD8L, f

;add temporary to accumulator
    movf  TEMPK0, w
    addwf VErrD8L, f
    movf  TEMPK1, w
    skpnc
    incfsz      TEMPK1, w
    addwf VErrD8H, f

;shift accumulator right 2 times
    rrf   VErrD8H, f
    rrf   VErrD8L, f
    clrc
    rrf   VErrD8H, f
    rrf   VErrD8L, f

;shift temporary left 1 times
    clrc
    rlf   TEMPK0, f
    rlf   TEMPK1, f
    clrf  TEMPK2
    rlf   TEMPK2, f

;add temporary to accumulator
    clrf  VErrD82
    movf  TEMPK0, w

```

```

addwf VErrD8L, f
movf  TEMPK1, w
skpnc
incfsz      TEMPK1, w
addwf VErrD8H, f
movf  TEMPK2, w
skpnc
incfsz      TEMPK2, w
addwf VErrD82, f

```

```

btfss VErrH, 7
goto  AddVP
btfsc VErrH, 7
goto  SubVP

```

AddVP

```

movf  AL, w
addwf      VErrD8L, f
btfsc STATUS, C
incf  AH, f
movf  VErrD8L, w
movwf VPErrL

movf  AH, w
addwf V ErrD8H, w
movwf VPErrH

goto  EndAddVP

```

SubVP

```

movf  V ErrD8L, w
subwf AL, f
btfss STATUS, C
incf  V ErrD8H, f
movf  AL, w
movwf VPErrL

movf  V ErrD8H, w
subwf AH, w
movwf VPErrH

goto  EndAddVP

```

EndAddVP

```

movf  VPErrH, w
movwf IComH
movf  VPErrL, w
movwf IComL

```

;Absolute value of ICom

```

    movf  IComH,w
    movwf ChgSignH
    movf  IComL,w
    movwf ChgSignH

    btfsc VPErrH,7
    call  ChgSign

    movf  ChgSignH,w
    movwf IComH
    movf  ChgSignL,w
    movwf IComL
;-----
TestOutputbits
    movf  IComH,f
    btfss STATUS,Z                ;skip if VPErrH=0
    goto  Saturate

    movlw 0x72                    ;%89
    subwf IComL,w                ;
    btfsc STATUS,C                ;if result negative
    goto  Saturate                ;

;VPErr<127 only 7 bits of VPErr is used

    movf  IComL,w
    movwf ChgSignL
    clrf  ChgSignH

    btfsc VPErrH,7                ;if VErrH<0
    call  ChgSign                ;

    movf  ChgSignL,w
    movwf IComLt

;Signed 8 bit IComLt

    movf  IComLt,w
    addlw 0x80
    movwf IComL
;    movwf PORTD

;Signed 8 bit IComL, zero at 0x80
EndVControl
    bcf   Temp,0
    bcf   PORTC,4                ;For DEBUGGING

    goto start

Saturate
    btfsc VPErrH,7

```

```

        goto SatNeg
        btfss VPErrH,7
        goto SatPos

SatNeg
        movlw 0x0E                                ;-%89
        movwf ICComL
        goto EndVControl
SatPos
        movlw 0xF2                                ;+%89
        movwf ICComL
        goto EndVControl

;-----RS232-----
;Subroutine to wait and receive a byte
;Returns character in W
;
getc  ; Subroutine, Input(None) Output=W  Received Data

        bcf     STATUS,RP0        ; Select Bank 0.
getc1
        btfss   PIR1,RCIF        ; Skip if RC int flag set (buffer
empty)
        goto    getc1            ; Try again
        movf    RCREG,W          ; Read the character
        bcf     PIR1,RCIF        ;Clear the interrupt flag
        return

;Subroutine to transmit a byte and wait
;W = Character
;
putc  ; Input(W) Output
Serial
        bcf     STATUS,RP0        ; Select Bank 0.

        movwf   TXREG            ; Write it!
putc1
        bsf     STATUS,RP0        ; Select Bank 1
;        movf    TXSTA,W          ; Peek transmit status

        btfss   TXSTA,1          ; Skip if TXbuffer empty
        goto    putc1            ; Try again
        bcf     STATUS,RP0        ; Select Bank 0.

        return

SendErr
SErr1
;-----
        bsf     PORTC,3

        movf    SHigh,w
        call   putc                ;Send Error High

```

```

movf Slow,w           ;W=AN1L
call putc

bcf Temp,3
bcf PORTC,3
goto start
; _____RS232_____END_____
;-----WAIT-----
WaitCom               ;Wait until rec.
0x22 '"' from computer
    bsf STATUS,5
    bcf INTCN,7       ;enable Global Int
    bcf STATUS,5
    movlw 0x23       ;'#'
    call putc        ;
    call getc        ;Check whether there
exists command
    movwf RecDataL   ;
    movlw 0x22       ;
    subwf RecDataL,W ;
    btfsc STATUS,2   ;Check for Zero if zero
goto next step
    goto EndWait
    movlw 0x21       ; '!'
    call putc        ;
    goto WaitCom
EndWait
    bsf STATUS,5
    bsf INTCN,5       ;clear Tmr0 Int
    bsf INTCN,7       ;enable Global Int
    bcf INTCN,6       ;disable Per. Int.
    bcf PIE1,5        ;disable rec. Int.
    bcf STATUS,5     ;Bank0
    bsf Temp,4        ;Don't enter RecCom
    movlw d'6'
    movwf TMR0
    goto start2
; _____END__WAIT_____

start

start2

    movf IComL,w
    movwf PORTD
    btfsc      Temp,0
    goto VControl
    btfsc Temp,3
    goto SendErr

    goto start

```

IntSub

```
        bcf    STATUS,RP0                ;Bank0
        movwf   wBuffer                ;Store Work

        bsf    Temp,0                  ;Enable VControl
        bsf    Temp,3                  ;Enable SendErr
;Timing
        movlw   d'6'
        addwf   TMR0,1

        bsf    STATUS,RP0
        bcf    INTCON,TOIF             ;clear Tmr0 Int
        bsf    INTCON,GIE             ;enable Global Int
        bcf    STATUS,RP0
        movf   wBuffer,w                ;
        retfie
        end
```

APPENDIX G: COST ANALYSIS

Cost analysis of the modules developed for motor drives are given below in Tables G.1 to G.4. The costs of components are taken from Arrow Electronics. Note that these are the prices for 1-25 units, if more units are bought the price will significantly decrease.

Table G.1 Cost analysis of converter A.

Converter A			
Component	Manufacturer	Quantity	Price
IR2113	International Rectifier	2	\$5,02
CD40106	Fairchild Semiconductor	2	\$0,15
6N136	Toshiba Corporation	3	\$0,58
IRF530N	Philips	4	\$0,36
Total			\$13,52

Table G.2 Cost analysis of converter B.

Converter B			
Component	Manufacturer	Quantity	Price
L298N	STMicroelectronics	1	\$3,23
BYW29E	Philips	4	\$0,33
Total			\$4,55

Table G.3 Cost analysis of current regulator.

Current Regulator			
Component	Manufacturer	Quantity	Price
LM339N	ON Semiconductor	1	\$0,24
TL494N	Fairchild Semiconductor	1	\$0,36
TL084CN	STMicroelectronics	2	\$0,28
TL082CN	STMicroelectronics	2	\$0,24
DAC0808LNC	National Semiconductor	1	\$0,65
74LS08N	Fairchild Semiconductor	1	\$0,29
CD40106	Fairchild Semiconductor	1	\$0,15
LTS25-NP	LEM	1	\$12,00
Total			\$14,73

Table G.4 Cost analysis of motion controller.

Motion Controller			
Component	Manufacturer	Quantity	Price
74HC74AN	STMicroelectronics	1	\$0,13
SN7437N	TI	1	\$0,28
74HC04	Philips	1	\$0,12
PIC16F877	Microchip	1	\$5,92
ICL232IPE	Intersil	1	\$2,89
Total			\$9,34

The two motor drives designed in this study are motor drive A, which employs converter A and motor drive B, which employs converter B. The cost analysis of motor drivers are given at Table G.5.

Table G.5 Cost analysis of motor drives

Motor Drive A		Motor Drive B	
Module	Price	Module	Price
Converter A	\$13,52	Converter B	\$4,55
Current Regulator	\$14,73	Current Regulator	\$14,73
Motion Controller	\$9,34	Motion Controller	\$9,34
PCB	\$10,00	PCB	\$10,00
Misc	\$5,00	Misc	\$5,00
Total	\$52,59	Total	\$43,62

REFERENCES

- [1] Control Techniques; “<http://www.controltechniques.com/index.asp>”, Last Checked: 30.07.2004
- [2] DCS500B / DCS600 - Technical Data - actual; “[http://library.abb.com/GLOBAL/SCOT/SCOT204.nsf/VerityDisplay/4D23AD56F3BB57E7C1256E3600332428/\\$File/3adw000165R0101_dcs4_5_6_technical_data_e_a.pdf](http://library.abb.com/GLOBAL/SCOT/SCOT204.nsf/VerityDisplay/4D23AD56F3BB57E7C1256E3600332428/$File/3adw000165R0101_dcs4_5_6_technical_data_e_a.pdf)”, Last Checked: 30.07.2004
- [3] Maxon Motors Operating instructions: <http://www.maxonmotorusa.com/downloads/items.cfm?product=%2DQ%2DDC%20Servo%20Control%20LSC%2030%2F2>
- [4] Hitachi L300P catalog, <http://www.hitachi-ies.co.jp/english/library/pdf/l300p.pdf>
- [5] Bose, B.K.; “Power electronics and motion control-technology status and recent trends“ Industry Applications, IEEE Transactions on , Volume: 29 , Issue: 5 , Sept.-Oct. 1993, Pages:902 – 909
- [6] Van Wyk, J.D., “Power electronic converters for motion control”, Proceedings of the IEEE, Volume: 82 , Issue: 8, Aug. 1994, Pages 1164 – 1193
- [7] Jahns, T.M.; Blasko, V., “Recent advances in power electronics technology for industrial and traction machine drives”, Proceedings of the IEEE, Volume: 89, Issue: 6, Jun 2001, page(s): 963-975
- [8] Ying-Kun L.; Tong Z.; Song-Fa L.;” Silicon power devices: RF power transistor & electric power device”, Solid-State and Integrated-Circuit Technology, 2001. Proceedings. 6th International Conference on , Volume: 1, 22-25 Oct. 2001, Pages:149 - 154 vol.1
- [9] Ellis, G.; Lorenz, R.D., “Comparison of motion control loops for industrial applications”, Industry Applications Conference, 1999. Thirty-Fourth IAS Annual Meeting. Conference Record of the 1999 IEEE , Volume: 4 , 3-7 Oct. 1999
- [10] Joos, G.; Sicard, P.; Goodman, E.D.;”_A comparison of microcomputer-based implementations of cascaded and parallel speed and current loops in DC motor drives”, Industry Applications, IEEE Transactions on , Volume: 28 , Issue: 1 , Jan.-Feb. 1992, Pages:136 – 143

- [11] El Zawawi, A.M.O.; Ashour, H.A.; "A fast acting current limit for a PC-based DC drive", Electrotechnical Conference, 1996. MELECON '96., 8th Mediterranean , Volume: 1 , 13-16 May 1996
- [12] Ahmed, F.I.; El-Tobshy, A.M.; Mahfouz, A.A.; Ibrahim, M.M.S.; "P-I and I-P controllers in a closed loop for DC motor drives", Power Conversion Conference - Nagaoka 1997., Proceedings of the , Volume: 2 , 3-6 Aug. 1997, Pages:613 - 618 vol.2
- [13] Schmidt, P.B.; Lorenz, R.D.; "Design principles and implementation of acceleration feedback to improve performance of DC drives", Industry Applications, IEEE Transactions on , Volume: 28 , Issue: 3 , May-June 1992, Pages:594 – 599
- [14] Kang, J.K.; Lee, J.T.; Kim, Y.M.; Kwon, B.H.; Choi, K.S.; "Speed controller design for induction motor drives using a PDF control and load disturbance observer", Industrial Electronics, Control and Instrumentation, 1991. Proceedings. IECON '91., 1991 International Conference on , 28 Oct.-1 Nov. 1991, Pages:799 - 803 vol.1
- [15] Barakat, N.; Rajagopalan, R.; "Speed control of a DC motor using a feedforward computed torque control scheme", Intelligent Control, 1996., Proceedings of the 1996 IEEE International Symposium on , 15-18 Sept. 1996, Pages:432 – 437
- [16] Lee, C.K.; Kwok, N.M.; "Torque ripple reduction in brushless DC motor velocity control systems using a cascade modified model reference compensator", Power Electronics Specialists Conference, 1993. PESC '93 Record., 24th Annual IEEE , 20-24 June 1993, Pages:458 – 464
- [17] Ahmed, F.I.; El-Tobshy, A.M.; Mahfouz, A.A.; Ibrahim, M.M.S.; " P-I and I-P controllers in a closed loop for DC motor drives", Power Conversion Conference - Nagaoka 1997., Proceedings of the , Volume: 2 , 3-6 Aug. 1997, Pages:613 - 618 vol.2
- [18] Hanamoto, T.; Tanaka, Y.; Mochizuki, T.; Xu, Z.; Ogawa, T.; "Speed control method of DC motor drive system with speed observer" Industrial Electronics, Control and Instrumentation, 1991. Proceedings. IECON '91., 1991 International Conference on , 28 Oct.-1 Nov. 1991 Pages:613 - 618 vol.1
- [19] Chevrel, P.; Sicot, L.; Siala, S.; "Parallel versus cascade schemes for DC-motor speed and current control" Decision and Control, 1997., Proceedings of the 36th IEEE Conference on , Volume: 1 , 10-12 Dec. 1997, Pages:233 - 238 vol.1
- [20] Saneifard, S.; Prasad, N.R.; Smolleck, H.A.; Wakileh, J.J.; "Fuzzy-logic-based speed control of a shunt DC motor", Education, IEEE Transactions on , Volume: 41 , Issue: 2 , May 1998, Pages:159 – 164

- [21] Kucer, P.; Chrabaszez, I.; Altus, J.; Spanik, P.; Dobrucky, B.; "Comparison of several types of speed controllers with the kind based on fuzzy logic", Industrial Electronics, 1996. ISIE '96., Proceedings of the IEEE International Symposium on , Volume: 1 , 17-20 June 1996, Pages:427 - 432 vol.1
- [22] Ullah, Z.; Neely, W.S.; Khan, E.; "A microcontroller implementation of a NeuFuz control system" Fuzzy Systems, 1995. International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium., Proceedings of 1995 IEEE International Conference on , Volume: 4 , 20-24 March 1995, Pages:1785 - 1792 vol.4
- [23] Rashidi, F.; Rashidi, M.; Hashemi-Hosseini, A.; Speed regulation of DC motors using intelligent controllers, Control Applications, 2003. CCA 2003. Proceedings of 2003 IEEE Conference on , Volume: 1 , 23-25 June 2003, Pages:925 - 930 vol.2
- [24] Kulic, F.; Kukolj, D.; Levi, E.; "Artificial neural network as a gain scheduler for PI speed controller in DC motor drives", Neural Network Applications in Electrical Engineering, 2000. NEUREL 2000. Proceedings of the 5th Seminar on , 25-27 Sept. 2000, Pages:199 – 203
- [25] Koren, Y.; "Computer control of manufacturing systems". New York : McGraw-Hill, 1983.
- [26] Koren Y. and C.C.Lo, "Advanced Controllers for Feed Drives, Annals of the CIRP", vol.41, no.2, pp.689-697, 1992
- [27] Renton D. and Elbestawi M. A., "High speed servo control of multi-axis machine tools", International Journal of Machine Tools and Manufacture, Volume 40, Issue 4, March 2000, Pages 539-559
- [28] Tomizuka M., "Zero Phase Error Tracking for Digital Control", ASME Transactions Journal fo Dynamic Systems, Measurements and Control 109, pages: 65-68, 1987
- [29] Weck M., Sharp G.Y., "Sharp corner tracking using the IKF control strategy", Annals of the CIRP 39 volume 1, pages 437-441, 1990
- [30] Lee H.S.; Tomizuka, M.; "Robust motion controller design for high-accuracy positioning systems", Industrial Electronics, IEEE Transactions on , Volume: 43 , Issue: 1 , Feb. 1996, Pages:48 – 55

- [31] Yeh, S.-S. and Hsu, P.-L.; "An Optimal and Adaptive Design of the Feedforward Motion Controller", IEEE/ASME TRANSACTIONS ON MECHATRONICS, VOL. 4, NO. 4, DECEMBER 1999
- [32] Koren Y., "Cross Coupled biaxial control for manufacturing systems", ASME Transactions, Journal of Dynamic Systems, Measurement and Control 102 (4) 1980, 265-272
- [33] Shih, Y.-T., Chen, C.-S., Lee, A.-C.; "A novel cross-coupling control design for Bi-axis motion", International Journal of Machine Tools & Manufacture 42 (2002) 1539–1548
- [34] Clarke D. W., Mohtadi C. and P. S. Tuffs, "Generalized predictive control—Part I. The basic algorithm", Automatica, Volume 23, Issue 2, March 1987, Pages 137-148
- [35] Ahmed, F.I.; Mahfouz, A.A.; Ibrahim, M.M.; "A novel fuzzy controller for DC motor drives", Electrical Machines and Drives, 1999. Ninth International Conference on (Conf. Publ. No. 468) , 1-3 Sept. 1999, Pages:325 – 328
- [36] Smith, M.H., Annaswamy, A.M., Slocum, A.H., "Adaptive Control Strategies for a Precision Machine Tool Axis", Precision Engineering, Vol. 17, No. 3, 1995, pp. 192-206.
- [37] Singhose W. E., Searing W. P. and Singer N. C.; "Improving repeatability of coordinate measuring machines with shaped command signals" Precision Engineering, Volume 18, Issues 2-3, April-May 1996, Pages 138-146
- [38] Mohan, N.; "First Course on Electric Machines and Drives Using an Integrative Approach"; <http://www.ece.umn.edu/groups/electricdrives.>; Last Checked: 30.07.2004
- [39] de Rooij, M.A.; Strydom, J.T.; van Wyk, J.D.; Beamer, P.;"Development of a 1 MHz MOSFET gate-driver for integrated converters"Industry Applications Conference, 2002. 37th IAS Annual Meeting. Conference Record of the ,Volume: 4 , 13-18 Oct. 2002 Pages:2622 - 2629 vol.4
- [40] "2.0A Gate Drive optocoupler with integrated (VCE) Desaturation Detection and Fault Status Feedback", Hewlett Packard technical datasheet for the HCPL316J integrated circuit.
- [41] "High and Low Side Driver", International Rectifier technical datasheet for the IR2113 integrated circuit.

- [42] “IGBT-Driving Hybrid ICs (EXB8..-Series)”, Fuji Electric application manual.
- [43] “Hybrid Dual MOSFET-Driver “, Semikron technical datasheet for the SKHI 21 A hybrid gate driver circuit.
- [44] Strydom J.T., van Wyk J.D., de Rooij M.A., “Integration of a 1-MHz Converter with Active and Passive Stages”, Proceedings of the IEEE Applied Power Electronics Conference (APEC), Vol. 2, Anaheim, California, U.S.A., March 2001, pp 1045 – 1050
- [45] Canales F., Barbosa P., Aguilar C., Lee F.C, “A Fixed-Frequency Zero- Voltage-Switching Three-Level DC/DC Resonant Converter”, Proceedings CPES Seminar, Virginia Tech, Blacksburg, Virginia U.S.A., April 2001, pp 155 – 160.
- [46] Vasic D., Costa F., Sarraute E., “A new MOSFET & IGBT gate drive insulated by a piezoelectric transformer”, Record of the 32nd IEEE Annual Power Electronics Specialists Conference (PESC), 2001, Vol 3, pp 1479 – 1484.
- [47] Hui S.Y., Tang S.C., Chung H.S., “Optimal Operation of Coreless PCB Transformer-Isolated Gate Drive Circuits with Wide Switching Frequency Range”, IEEE Transactions
- [48] Tang S.C., Hui S.Y.R., Chung H., “A Naturally Soft-switched High- Frequency Gate Drive Circuit for Power MOSFETs/IGBTs”, Proceedings IEEE Conference on Power Electronics and Drive Systems (PEDS), July 1999, Hong Kong, pp 246 – 252.
- [49] Wood P., “Transformer-Isolated HEXFET® Driver provides very large duty cycle ratios”, International Rectifier, Application note 950A, Chapter 9, pp I-75 – I-77.
- [50] Lin R.L., Lee F.C., “Single-power-supply-based transformerless IGBT/MOSFET gate-driver with 100% high-side turn-on duty cycle operation performance using auxiliary bootstrapped charge pumper”, Record of the 28th IEEE Annual Power Electronics Specialists Conference (PESC), 1997, Vol 2, pp 1205 –1209.
- [51] Hess H.L, Baker R. J., “Transformerless Capacitive Coupling of Gate Signals for Series Operation of Power MOS Devices, IEEE Transactions on Power Electronics, Vol. 15, No. 5, September 2000, pp 923 – 930.
- [52] PIC16F877 Datasheet, Microchip, DS30292C
- [53] ICL232 Datasheet, Intersil FileNo: 3020.6, June 2001
- [54] PICmicro™ Mid-Range MCU Family Reference Manual, Microchip, DS33023A

[55] Lorenz R.D, Lipo T. A., Novotny D. W.; “Motion control with induction motors”, Proceedings of the IEEE, Vol. 82, no.8, August 1994

[56] Lepkowski J., “Motor Control Sensor Feedback Circuits”, Microchip Application Note No:AN894

[57] Petter, J.; McCarthy, J.; Pollak, P.; Smith, C.C.; “Survey of DC current measurement techniques for high current precision power supplies”, Nuclear Science Symposium and Medical Imaging Conference, 1991., Conference Record of the 1991 IEEE, 2-9Nov.1991; Pages:961 - 967 vol.2

[58] http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html;
Last Checked: 01.August.2004

[59] Hopkins, T; “Line driver and receiver considerations”, Motorola Application note, AN-708A

[60] “HV Floating MOS-Gate Driver ICs”; International Rectifier Application note AN978-b

[61] Medel Elektronik, <http://www.medelektronik.com/eng/dc.htm> Last Checked: 13.August.2004

[62] Tunçmatik Elektronik, “<http://www.tuncmatik.com/DisplayProduct.aspx?q=14>”
Last Checked: 13.August.2004

[63] Bilten, http://www.bilten.metu.edu.tr/ge/ge_sunum_files/frame.htm#slide0091.htm Last Checked: 13.August.2004

[64] Wikipedia Encyclopedia, “<http://en.wikipedia.org/wiki/RS232>”, Last Checked: 01.August.2004

[65] RS232 information, “<http://www.cpmsspectre.pwp.blueyonder.co.uk/rs232/rs232.htm>” Last Checked: 01.August.2004