

To the Graduate Council:

I am submitting herewith a dissertation written by Jih-Sheng Lai entitled "High Frequency Resonant Link Inverter Induction Motor Drives Using Microcomputer Control." I have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Electrical Engineering.

B.K. Bose

Bimal K. Bose, Major Professor

We have read this dissertation
and recommend its acceptance:

L. Symonds
J. M. Bailey
J. S. Fawcett
Edward R. Harris

Accepted for the Council:

C. W. Mink

Vice Provost
and Dean of The Graduate School

HIGH FREQUENCY RESONANT LINK INVERTER
INDUCTION MOTOR DRIVES
USING MICROCOMPUTER CONTROL

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Jih-Sheng Lai

December 1989

Acknowledgments

In the course of this dissertation work, a number of people have generously assisted me. First I wish to thank Professor J. S. Lawler for his introducing me to Power Electronics and his long term advisory support. I want to also thank Professors J. M. Bailey, F. W. Symonds, and E. G. Harris for their assistance and for serving as my supervisory committee.

I wish to give special thanks to Dr. D. W. Hilson for his encouragement and administrative support, R. Taylor for witnessing my inventions, R. L. Lambert for his laboratory support, S. Bays for her proofreading of this manuscript, and the entire staff of the Power Electronics Applications Center for providing working space, funding, and other substantial support. I am also grateful to R. W. Young of Oak Ridge National Laboratory for his technical discussions and partnership.

In particular, I wish to thank my advisor, Dr. B. K. Bose, for his time and consultations and for his initiation of this project from which I have gained invaluable knowledge of Power Electronics.

Finally, I wish to pay tribute to my wife, Ying-Ru, for her moral and physical assistance throughout my degree-seeking years. Without her patience, this project could never have been accomplished.

Abstract

Resonant link inverters show tremendous promise for next generation adjustable speed *ac* machine drives. A bidirectional initial current control concept in the resonant *dc* link circuit which solves the voltage overshoot problem and establishes reliable zero voltage crossing for smooth inverter operation is presented here. The resonant circuit has been systematically analyzed to establish the criteria for initial current selection, and a new circuit has been proposed to establish the bidirectional initial current. The improved inverter permits successful operation at higher than 50KHz resonant frequency and even promises up to 100 KHz. The control of the inverter which involves prediction of the inverter input current has been formulated.

In addition to the improved resonant *dc* link inverter, a quasi-resonant voltage notching scheme has been developed. By a high frequency (100 KHz) resonant circuit, this type of inverter creates zero voltage notches at defined positions so as to permit soft switching of power semiconductor devices. The precise resolution of the voltage notch position in the *dc* link greatly enhances the inverter power capability and reduces the load current ripple in comparison with the resonant link inverter.

Along with the inverter-fed induction motor speed control system, the two proposed schemes have been simulated on a computer using *PC-SIMNON*. For experimental setup, two power inverter circuits at 25 KHz and 60 KHz have been

constructed using the insulated gate bipolar transistor (*IGBT*), and the bidirectional initial current control circuit has been fabricated on wire wrapping boards. The induction motor field-oriented vector control system has been implemented using a TMS320C25 digital signal processor with two-channel *A/D* inputs and two-channel *D/A* outputs. The software development and user interface for the speed control system were supported by a *PC-AT* compatible computer. Both resonant and quasi-resonant link inverters have been tested successfully with series resistance-inductance and induction motor loads.

Table of Contents

1	Introduction	1
2	Literature Review	10
2.1	Traditional Inverter Topologies	11
2.1.1	Square-Wave Inverters	12
2.1.2	<i>PWM</i> Inverters	13
2.2	Resonant Mode <i>DC/DC</i> Converters	18
2.2.1	Operation of Zero Current Switching Resonant Link <i>DC/DC</i> Converter	20
2.2.2	Operation of Zero Voltage Switching Resonant Link <i>DC/DC</i> Converter	20
2.3	Resonant Link Inverters for <i>AC</i> Power Conversions	23
2.3.1	Resonant <i>AC</i> Link Inverters	23
2.3.2	Resonant <i>DC</i> Link Inverters	28
2.3.3	A Quasi-Resonant Notch Link Inverter	32
3	Resonant DC Link Circuit Analysis	37
3.1	Resonant <i>DC</i> Link Circuit Modeling	37
3.2	Mathematical Analysis	39
3.3	Experimental Results of Resonant <i>DC</i> Link Circuit	47

4	Current Initialization Resonant Link Inverter System	51
4.1	A Bidirectional Current Initialization Resonant Link Inverter System	52
4.2	Operation Modes of Current Initialization Circuit	53
4.3	Prediction of the Initial Current	55
4.4	Simulation of the Proposed Resonant Link Inverter System	58
4.5	Experimental Results of the Resonant <i>DC</i> Link Inverter System . . .	64
4.5.1	Experimental Results of a 25 KHz System	64
4.5.2	Experimental Results of a 60 KHz System	67
4.5.3	Experimental Results of the Inverter Input Current Prediction	73
4.5.4	Experimental Results of Efficiency Evaluation	75
5	Quasi-Resonant DC Voltage Notching Inverter System	77
5.1	Operation Modes of Quasi-Resonant Voltage Link Circuit	77
5.2	Quasi-Resonant <i>DC</i> Link Circuit Analysis	80
5.2.1	Modeling of Switching Device Conduction Drop	80
5.2.2	Equivalent Quasi-Resonant Link Circuit Analysis	82
5.2.3	Experimental Results of a Quasi-Resonant Link Circuit Under No-Load Condition	86
5.3	Current Initialization	87
5.4	Optimized Control of Switching Frequency	89
5.5	Simulation of Quasi-Resonant <i>DC</i> Link Inverter System	91
5.6	Experimental Results Under Loaded Condition	95
6	Simulation Program Design	99
6.1	Modeling of a Hysteresis Band Inverter	101
6.2	Modeling of Induction Motor	103
6.3	Modeling of Some Circuit Elements	105
6.3.1	Loop Break	105

6.3.2	Single Shot Generator	106
6.3.3	Vector Transformation	106
7	Hardware Circuit Design	108
7.1	Power Circuits	110
7.2	Gate Drive Circuits	117
7.2.1	Gate Drive Circuit Using Transistors	117
7.2.2	Gate Drive Circuit Using IR2110	119
7.2.3	Gate Drive Circuit Using Pulse Transformer	123
7.2.4	Gate Drive Circuit Using a Single <i>CMOS</i> Driver	125
7.3	Control Circuit	127
7.3.1	Control Circuit for the Quasi-Resonant Switch	127
7.3.2	Lock-out Circuit	129
7.4	<i>F/V</i> Converter Circuit Design	130
7.5	Isolation Amplifier Circuit Design	133
8	Speed Control System and Software Design	135
8.1	Data Input and Output	137
8.1.1	Interrupt Service Routines Design	137
8.1.2	Input/Output Scaling	138
8.1.3	Handshaking Design	139
8.2	Speed Controller Design	141
8.2.1	Digital Controller Design	141
8.2.2	Vector Controller Design	143
8.3	Monitor Program Design	144
8.3.1	<i>DSP-16</i> Board Initialization	144
8.3.2	Sampling Rate Setting	145
8.3.3	Man-Machine Interface	145

TABLE OF CONTENTS

viii

8.4 Experimental Results of Speed Responses	147
9 Conclusion	151
References	157
Appendices	165
A SIMNON Simulation Program Listing	166
B TMS320C25 Assembly Program Listing	184
C Monitor Program Listing	195
Vita	209

List of Tables

3.1	Components for resonant link current initialization test.	47
3.2	<i>IGBT</i> and ultra-fast recovery diode specifications.	49
4.1	Inverter input current for different switching patterns.	56
4.2	Truth table of inverter input current prediction.	57
4.3	Machine parameters for simulation study.	59
4.4	Designed parameters of the resonant link circuit	60
4.5	Specification for 60 KHz resonant circuit components	68
4.6	Efficiency Evaluation	76
5.1	Circuit parameters of a 60KHz quasi-resonant link system.	86
5.2	Comparison of sum of phase current error square.	95
7.1	Switching characteristics of IXGH40N60A	113
7.2	A comparison of test results and IR2110 specification	122
7.3	Designed data of the pulse transformer	123
8.1	Interrupt memory locations and priorities	138

List of Figures

2.1	A voltage-fed square-wave inverter circuit.	13
2.2	Output voltage waveforms of a square-wave inverter.	14
2.3	A voltage fed <i>PWM</i> inverter.	16
2.4	Output voltage waveforms of a sinusoidal <i>PWM</i> inverter.	17
2.5	Comparison of conventional switching mode and resonant mode <i>dc/dc</i> converters.	19
2.6	Operation of zero current switching resonant mode <i>dc/dc</i> converters.	21
2.7	Operation of zero voltage switching resonant mode <i>dc/dc</i> converters.	22
2.8	Topologies of resonant <i>ac</i> link inverters.	24
2.9	Power circuit of a series resonant <i>ac</i> link inverter.	25
2.10	Operation of a series resonant <i>ac</i> link inverter.	26
2.11	A parallel resonant <i>ac</i> link inverter system.	27
2.12	Operation of a parallel resonant <i>ac</i> link inverter.	28
2.13	Circuit topologies of resonant <i>dc</i> link inverters.	29
2.14	Operation of a series resonant <i>dc</i> link inverter.	31
2.15	Operation of a parallel resonant <i>dc</i> link inverter.	33
2.16	A quasi-resonant notch link inverter.	34
2.17	Typical operation of a quasi-resonant notch link inverter.	35

3.1	Resonant dc link inverter for induction motor drive.	38
3.2	Equivalent circuit of the resonant link inverter during resonant interval.	39
3.3	Current and voltage waveforms in a resonant cycle.	42
3.4	Maximum and minimum resonant circuit voltages as functions of initial current I ($I = I$).	44
3.5	Maximum resonant circuit voltage $[V(t)]$ as function of I for different inverter input currents.	45
3.6	Minimum resonant circuit voltage $[V(t)]$ as function of I for different inverter input currents.	46
3.7	Simulation and experimental results of an unloaded resonant link circuit test.	48
3.8	Experimental results of unloaded resonant link circuit using <i>IGBT</i> as the shunt switch.	50
4.1	Proposed resonant link inverter with bidirectional current initialization circuit.	52
4.2	Different operation modes of the resonant link circuit.	54
4.3	Equivalent analog multiplexer for estimation of inverter input current.	57
4.4	Block diagram of resonant link inverter and induction motor speed control system.	59
4.5	Simulation results of inverter input current prediction.	61
4.6	Simulation results of the proposed resonant link inverter induction motor drive system (50KHz).	62
4.7	Simulation results of the proposed resonant link inverter induction motor system (25KHz).	63
4.8	Schematic of the tested single phase resonant link system with <i>RL</i> load.	65

4.9	Experimental voltage and current results of 25 KHz resonant link inverter for single-phase RL load.	66
4.10	Actual and reference current waves of 25 KHz resonant link inverter for a single-phase RL load.	67
4.11	Experimental results of 60 KHz resonant link voltage and current under no-load condition.	69
4.12	Resonant link current and voltage under loading condition.	70
4.13	Parallel loading of resonant link inverter experimental setup.	71
4.14	Experimental results of phase current and voltage for the resonant link inverter.	72
4.15	Comparison of the actual and reference phase currents.	74
4.16	Comparison of the actual and predicted inverter input currents.	75
5.1	Quasi-resonant dc voltage notching circuit operation modes and waveforms.	78
5.2	Experimental results of $IGBT$ and diode conduction drops.	81
5.3	Equivalent quasi-resonant link circuit.	83
5.4	Maximum and minimum quasi-resonant link voltages as functions of initial current I	85
5.5	Quasi-resonant link circuit experimental results under no-load condition.	87
5.6	Quasi-resonant dc voltage notching scheme with current initialization circuit.	88
5.7	Block diagram of quasi-resonant voltage notch link inverter induction motor drive.	90
5.8	Control block diagram of voltage notching and current initialization circuit.	92

5.9	Simulation results of the quasi-resonant <i>dc</i> notch link voltage and current waves.	93
5.10	Simulation result of the quasi-resonant <i>dc</i> notch link inverter current waves.	94
5.11	Experimental quasi-resonant current and voltage under loaded condition.	96
5.12	Experimental phase current of quasi-resonant link inverter for induction motor load.	97
6.1	Block diagram of resonant link inverter drive for <i>SIMNON</i> simulation.	100
6.2	Circuit model of a hysteresis band inverter.	102
6.3	Circuit model of a D-type flip-flop.	102
6.4	Induction machine <i>dq</i> equivalent circuits at synchronous rotating frame.	103
6.5	Circuit Model of a single shot generator.	106
7.1	Photograph of the experiment setup.	109
7.2	Photograph of the measurement setup.	111
7.3	Bidirectional resonant <i>dc</i> link power circuit and sensors.	112
7.4	Experimental results of current sense resistor and amplifier circuits. .	114
7.5	Quasi-resonant <i>dc</i> voltage notch link inverter power circuit.	116
7.6	Gate drive circuit using transistors.	118
7.7	A new gate drive IC IR2110 using bootstrap technique.	120
7.8	Gate drive circuit using IR2110.	122
7.9	Gate drive circuit using pulse transformer.	123
7.10	Gate drive output responses using pulse transformer.	124
7.11	Gate drive circuit using <i>CMOS</i> gate drivers.	125
7.12	Gate-source voltage response of <i>IGBT</i> using <i>CMOS</i> gate driver. . . .	126
7.13	Control circuit	128

7.14	Schematic for quasi-resonant control switch circuit.	129
7.15	Lock-out circuit.	130
7.16	Schematic for F/V converter with isolation circuit.	131
7.17	F/V converter output.	132
7.18	Experimental and approximated step responses of the F/V converter circuit.	133
7.19	Schematic for isolation amplifier circuit.	134
8.1	Computer speed control system for resonant link inverter drive. . . .	136
8.2	Input signal flow diagram.	139
8.3	Motor speed and A/D input signal regression.	140
8.4	Induction motor acceleration profile.	148
8.5	Induction motor acceleration profile with PC communication delay. .	149

Chapter 1

Introduction

Inverter circuit topologies have evolved from early square-wave inverters through *PWM* inverters to present day resonant link inverters. Resonant link inverters for *ac* machine drive applications were extended from resonant-mode *dc/dc* power converters. The evolution of circuit topologies can be attributed to both material supplies and industry demands. From the supply side, power density, voltage and current levels, and switching speed of semiconductor devices have dramatically increased such that new resonant circuit topologies are now possible. From the demand viewpoint, requirements of high efficiency, compact size, good output quality, and quiet operation have driven the development of new high frequency resonant power conversion equipment. The development of new inverter topologies has attracted much attention and some breakthroughs have been made.

The resonant link inverter differs from the conventional inverter by the addition of a resonant link circuit between the supply source and the inverter. The resonant link circuit provides periodic zero voltage or zero current interval to eliminate switching loss. Unlike resonant mode *dc/dc* converters for power supplies, application to *ac* machine drives requires the additional consideration of the current flowing back

from the machine. This feedback current causes major difficulty in resonant link inverter design, especially in regard to the voltage overshoot problem, and therefore a solution to this problem is very desirable.

The resonant link circuit is composed of an inductor and a capacitor in parallel or series. This circuit is also called “resonant tank.” According to the connection of the resonant tank, there are two types of resonant link inverters, i.e., shunt resonance, zero voltage switching-based voltage-fed inverters [13,21,38,39] and series resonance, zero current switching-based current-fed inverters [19,20,28]. The two types of inverters can be easily distinguished by looking at the connection of the load. If the load is connected in series, the resonant tank determines the current applied to the load, and the power source inputting to the inverter bridge is pulsed “current source” [28]. However, if the load is connected in parallel with the resonant tank, the tank determines the voltage applied to the load, and the power source to the inverter bridge is a pulse train of “voltage source.”

When distinguished by the source being applied to the inverter bridge, the resonant link can be classified into two categories: resonant *ac* link [19,20,38,39]. and resonant *dc* link [13,21,28]. The resonant *ac* link converter provides *ac* voltage or current source to the converter bridge, while the resonant *dc* link inverter inputs *dc* voltage or current source to the inverter.

Because of the induced power loss and latching problems, zero current switching has some difficulties in higher frequency operation [32]. On the other hand, zero voltage switching does not have such problems, and is more promising for next generation *ac* machine drive applications. Comparing resonant *ac* and resonant *dc* link systems, the resonant *dc* link is preferred for *ac* power applications because the switch need not to be bidirectional, which makes the cost less.

Zero voltage switching of devices provides a number of advantages over the full voltage switching in a regular *PWM* inverter. The major advantages of zero voltage

switching resonant *dc* link inverter are summarized as below.

1. On the converter side:

- The switching loss being dominant in a *PWM* inverter practically vanishes resulting in improvement of efficiency.
- The device cooling requirement is minimal.
- Converter power density is high.
- The snubber circuit is not required.
- Acoustic noises can be eliminated in high switching frequency.
- The reliability of the inverter is improved because switching stress on devices is relieved.

2. On the output side:

- The capacitive coupling current on machine windings is low because of low inverter output voltage change rate dV/dt .
- Output *EMI* is low.
- Generated torque and acceleration are smooth due to high modulation frequency.
- Mechanical noise disappears under smooth generated torque.

For an *ac* machine drive, soft switching is extremely important because the capacitive coupling current in machine windings may cause serious damage to the motor when inverter output dV/dt is high. Large capacitive current occurs when a regular *ac* machine is controlled by a conventional full voltage switching *PWM* inverter.

In addition to its advantages, the zero voltage switching resonant *dc* link inverter has three fundamental problems:

1. Voltage overshoot – High voltage stress is a generic problem in resonant link inverters. In a zero voltage switched quasi-resonant converter, the link voltage stress is eight times the supply voltage [42]. For a resonant dc link inverter [13], if the inverter operates with no load condition, the peak resonant link voltage swings to $2V_s$, where V_s is the dc supply voltage. With loading of the inverter this voltage swing becomes worse. If the inverter switching is such that the inverter input current changes from maximum positive (active mode) to maximum negative (feedback mode) value, the overshoot voltage will be higher than $3V_s$. A passive voltage clamping with energy recovery [14] has been proposed to limit the voltage to approximately $2.5V_s$. An active clamping technique with energy recovery has been proposed [14] instead of using passive clamping, where clamping level is reduced to $1.3V_s$. Active clamping shows difficulty in operation during transition from active mode to feedback mode. Further, the resonance delay at every cycle reduces the resonant frequency causing performance degradation of the inverter system.
2. Zero crossing failure – Zero crossing of resonant link dc voltage in every cycle is mandatory for successful operation of the inverter. Failure of zero crossing tends to occur due to finite Q of the resonant circuit where the voltage tends to build up in successive resonant cycles. The build-up of the fixed inductor initial current (I_m) was proposed [13] to solve this problem. However, the fluctuating and bidirectional inverter input current (I_d) tends to cause zero crossing failure unless I_m is established on the worst case basis and this worsens the voltage overshoot problem. To date, there has been no quantized solution of optimum initial current requirement.
3. Inverter phase current ripple – In any inverter operation, the phase current should closely track the sinusoidal command current so that the machine har-

monic heating is minimum. One difficulty of the resonant link inverter is that the inverter is constrained to be switched at zero voltage causing integral *PWM* (*IPWM*) to occur naturally. This lack of resolution in pulse width causes higher peak-to-peak ripple in phase current. To alleviate this problem, the resonant frequency should be high so that volt-second integral of a resonant pulse becomes low. To date, the resonant *dc* link inverter has been operated at 65KHz frequency [23]. Although the voltage clamping limits the peak voltage, volt-second area of a pulse becomes larger and aggravates the phase current ripple problem.

To solve these three problems, a current initialization scheme has been proposed [8,21]. The current initialization resonant link circuit limits the link voltage to $2V_s$ and assures zero crossing under all conditions of inverter operation. The initial current is bidirectional in nature and is established by an auxiliary circuit which operates at low frequency. The resonant circuit has been analyzed to establish the criteria of the initial current. The inverter-machine system has been simulated using *PC-SIMNON* [6,36] and performance has been studied at 25KHz and 50KHz resonant frequencies.

The simulation result shows that the current initialization resonant link inverter largely improves the reliability and performance of the ordinary resonant link inverter. However, two penalties still exist.

1. High voltage penalty – The inherent nature of resonant link operation demands that the *dc* link voltage should be double to that of the supply voltage. This additional link voltage does not contribute any additional output power, and therefore imposes voltage penalty on the devices.
2. Harmonic current penalty – In a resonant link inverter, the *dc* link voltage oscillates freely and the inverter switching has to be synchronized to appropriate

zero voltage notches. The lack of resolution of notch position causes higher ripple in machine current wave causing more harmonics heating.

A quasi-resonant voltage notching scheme is possible by adding a control circuit in the current initialization resonant link inverter described in [21]. This control circuit retains the fixed dc link voltage until at a desired point a voltage notch is created depending on the demand of the current controller. The precise resolution of voltage notch position in the dc link greatly enhances the inverter power capability and reduces the load current ripple in comparison with the previously developed resonant link inverter.

Both the peak voltage and current harmonic penalties are solved in the proposed quasi-resonant voltage notched inverter. The inverter dc link voltage is latched to the peak resonant voltage by the control circuit until a resonant voltage notch is commanded at a precise position. The larger volt-second area of extended pulse permits larger fundamental output voltage, or for the same output voltage, the link voltage can be reduced to a lower value. Thus, the power output of the inverter is substantially increased when compared to a resonant link inverter. In addition, the resolution of the notch position improves the output harmonic spectrums. The proposed quasi-resonant link inverter induction motor drive has also been computer simulated and its performance improvement over a resonant link inverter drive has been verified.

After computer simulation verification, hardware experiments were carried out to implement the proposed bidirectional current initialization resonant link and quasi-resonant link inverters.

The first implemented system is a current initialization controlled resonant link inverter drive running at 25KHz resonant frequency. The power circuit consists of a three phase bridge rectifier, two input filter capacitors, a resonant inductor (76

μH), a resonant capacitor ($.5 \mu\text{F}$), and a total of nine switching devices.

The resonant inductor was designed using a Litz wire to eliminate skin effect losses and using an air core to avoid the saturation effect. The resonant capacitor must have low *ESR* over the resonant frequency range; thus, a high voltage and high current rated polypropylene capacitor was selected. Because the switching device must be fast enough to be switched in the short zero voltage crossing interval, the fast switching “insulated gate bipolar transistor” (*IGBT*) was chosen for the experiment setup. If inverter efficiency is the first consideration, however, some newly developed devices like *MOS* controlled thyristors (*MCT*)[43]¹ and *MOS* depletion-mode thyristors (*DMT*) [2] may be better choices because of their lower conduction drop.

The gate drive circuits for the inverter bridge use pulse transformers for isolation. Because the pulse transformer is not suitable for arbitrary duty cycle operation, an optocoupler isolation gate drive has been developed for initial current control switches. Although the pulse transformer type is considerably faster and size economical, it is only good for integral cycle operation. The optocoupler with discrete component gate drive can be used for any duty cycle, but it requires additional power supplies.

The control circuit consists of sensor signal conditioning circuits, a resonant link controller, a three-phase *PWM* controller, a *PC*-based speed controller, and three isolation amplifiers. The voltage sensor is a simple *RC* voltage divider network with a voltage follower. The *ac* current sensor is a high frequency *CT* with a current-to-voltage converter. For *dc* current sensing, a low resistance shunt resistor (0.01Ω) is used with an amplifier circuit. The speed sensing employs a hollow-shaft incremental encoder with 1800 pulse per revolution and 100KHz maximum operation frequency.

¹*MCT* is not commercially available yet, but samples are available for test.

The resonant link controller is a dedicated analog and digital circuit. The main function of the analog circuit is inverter input current prediction and voltage comparison, while the digital circuit is to furnish the control logic. An analog multiplexer is adopted to predict inverter input current. For speed control, a “digital signal processor” *DSP TMS320C25*² with a data acquisition system (*DSP-16*)³ embedded in an *IBM PC-AT*⁴ compatible computer *PEP301*⁵ is employed. Only one input channel of the data acquisition system is used to acquire the motor speed, while two output channels are used to generate reference current signals. The reference currents are fed into the *PWM* control circuit to control the inverter. The *PWM* controller is a simple voltage comparator with a small hysteresis band feedback circuit.

The second implemented system is the quasi-resonant voltage notching system. This system needs one additional switch to adjust switching frequency. Because the resonant link system is a subset of the quasi-resonant system, the second system is constructed in a universal manner so the system can be switched from resonant link to quasi-resonant or vice versa. Although a switch is added, the size of this system is considerably smaller than the first system because of higher resonant frequency (60 KHz). The resonant inductor in the second system is only 21.2 μH , and the resonant capacitor is .33 μF .

The gate drive of the second system is also different from that of the first system. All gate circuits of ten switches are isolated by optocouplers and controlled by *CMOS* drivers. This design isolates the power from control circuits to avoid noise disturbances. For upper legs of the inverter bridge, the gate drive requires additional isolated power supply. To reduce the system size, a compact 1MHz power supply

²Trademark of Texas Instruments Inc.

³Trademark of Ariel Corp.

⁴Trademark of International Business Machines Corp.

⁵Trademark of Tektronix Inc.

was designed on board.

The sensor signal conditioning circuits in the second system are the same as the first one except that the *dc* current is sensed by a Hall sensor IHA100⁶ for isolation purposes. Most control circuitry in the second system is the same as the first system with exception of one additional switch control circuit for quasi-resonant notch control.

The newly developed schemes were first analyzed, and then simulated with satisfactory results. The complete power and control circuits were designed according to these analysis and simulation results. Both systems – the current initialization controlled resonant *dc* link system and the quasi-resonant voltage notch link system – have been successfully constructed and tested.

⁶Trademark of F. W. Bell Inc.

Chapter 2

Literature Review

High frequency resonant link inverters were derived from conventional inverters. The difference between the two is in a resonant tank, either series or parallel LC circuit, which is inserted into a converter between the power source and the inverter to generate periodic zero voltage or zero current. The inverter can then be switched at zero voltage or zero current crossing so that switching loss is eliminated, and the efficiency is increased.

Based on the inverter power circuit, those inverters suitable for *ac* machine drives can be divided into three categories: square-wave inverters, *PWM* inverters, and resonant link inverters. The evolution of the inverter is due in large part to performance requirements and device switching speed.

Ideally, the power device should have zero switching time and zero switching loss, but this is not realistic. Switching loss causes heating problems, lowers efficiency, and increases the size of the inverter. Square-wave inverters, with low switching frequency, are high in efficiency but poor in harmonics and transient responses. Output harmonics will further cause machine torque pulsation. On the other hand, *PWM* inverters reduce the harmonics but introduce switching losses. The high

performance inverter drive requires no torque pulsation and fast transient responses and thus needs higher switching frequency, which in turn causes more losses. The trade-off between performance and efficiency becomes a problem in selecting suitable inverters. High frequency resonant link inverters originally proposed to reduce the size of passive components [4,29,33] now become the solution to the above problem. Because in resonant link inverters the switching loss is near zero, switching frequency can be very high in order to meet the performance requirement and to reduce the size of passive components without sacrificing efficiency.

The application of resonant link inverters to *ac* machine drives requires bidirectional power flow in the link circuit, which is normally not required in *dc/dc* converters. However, the idea of resonant link inverter is extended from resonant mode *dc/dc* converters [13].

Because the concept of resonant link inverters for *ac* power applications is somewhat borrowed from resonant mode *dc/dc* conversions, and the circuit topology is derived from traditional inverter circuits, this chapter will describe traditional inverter topologies, resonant mode *dc/dc* converters, and various resonant link inverter topologies suitable for *ac* machine drives. Typical operation and simulation results of all inverter circuits are attached for comparison.

2.1 Traditional Inverter Topologies

The traditional inverter means the input power source is stiff *dc* voltage or current source and the output is *ac* power. The inverter switches need to be turned on or off under full voltage or full current. Such hard switching produces large amount of energy losses; therefore, the traditional inverters are normally limited in low frequency operations.

According to the chopping control, traditional inverter can be square-wave or

PWM type. Both traditional inverters share the same power circuit topology but differ in chopping control and device switching speed requirement. This section will describe typical traditional inverter topologies and discuss their characteristics.

2.1.1 Square-Wave Inverters

As indicated from the name, square-wave inverters generate square-wave type output voltages or currents. For 3-phase inverters, the output separates in six distinct periods in one cycle. Each 60° interval has a distinct gating pattern [5,11]. The *dc* source to the inverter can be constant voltage supplied by a capacitor filter or constant current supplied by an inductor filter. The inverter is called “voltage-fed” if its input is constant voltage source or “current-fed” if its input is constant current source. The inverter switches can be three devices turned-on or two devices turned-on at any instant.

Because each cycle requires only six switchings, this type of inverter is allowed to employ slow devices such as thyristors or gate-turn-off thyristors *GTOs*. The device turn-off time is typically tenths of a micro second. Figure 2.1 shows a typical voltage-fed inverter. The voltage across the *dc* rail V_d is assumed to be constant. Each thyristor has an anti-parallel diode to allow the current to feed back from the load. The load may be *RLC* circuit, induction motor, or synchronous motor. The *dc* voltage source can be supplied by an *ac* source through a rectifier circuit to provide a fix output amplitude or a controlled rectifier circuit to provide adjustable output amplitudes. For adjustable speed drives, however, the *dc* voltage source must be varied with the output frequency in order to maintain constant torque operation.

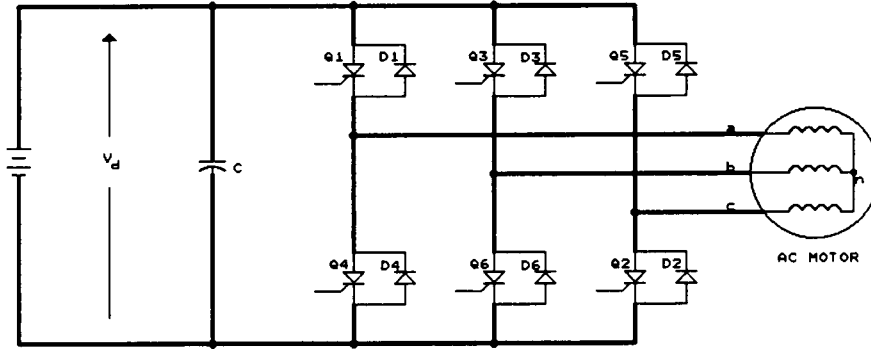
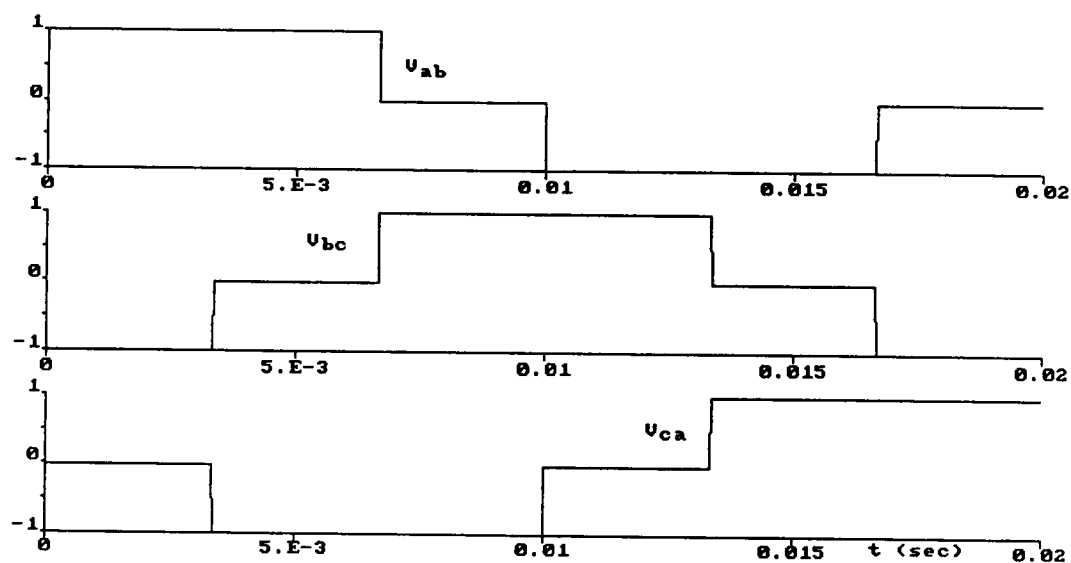


Figure 2.1: A voltage-fed square-wave inverter circuit.

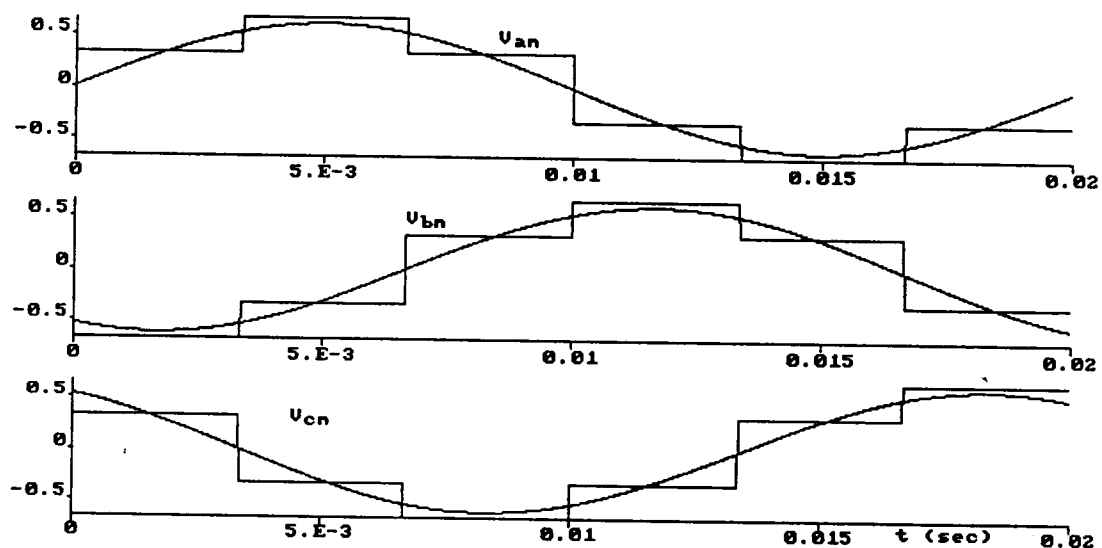
Figure 2.2(a) shows one cycle of output line voltage of the voltage-fed square-wave inverter with three thyristors turned-on at any instant. The switching sequence for this output is $Q_1Q_6Q_5$, $Q_1Q_6Q_2$, $Q_1Q_3Q_2$, $Q_4Q_3Q_2$, $Q_4Q_3Q_5$, $Q_4Q_6Q_5$. Using dc link voltage as the base, the peak line-to-line voltage is 1 pu. Figure 2.2(b) shows one cycle of output phase-to-neutral voltage waveforms. The peak phase voltage is $2/3$ pu. The fundamental sine waves with 50 Hz frequency are also shown for comparison.

2.1.2 PWM Inverters

The principle of the square-wave inverter is relatively simple but the output harmonic contents are excessive, causing machine heating and torque pulsating problems [5]. After the advent of fast self-commutated switching devices like power transistors and power *MOSFETs*, pulse width modulation (*PWM*) techniques were successfully applied to the inverter control. Instead of switching on and off (chopping) 6 times in a fundamental cycle in square-wave inverters, the *PWM* inverter employs more chopping in one cycle to synthesize the output waveform. The output



(a) Line-to-line voltage.



(b) phase-to-neutral voltage.

Figure 2.2: Output voltage waveforms of a square-wave inverter.

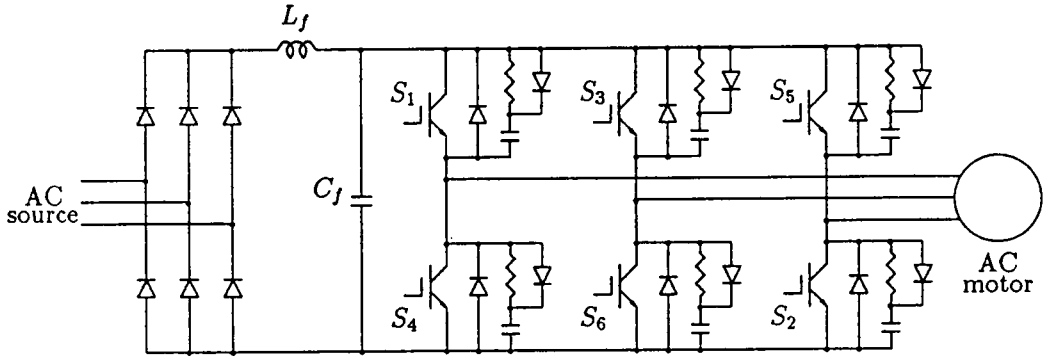
harmonics are significantly reduced, and the performance of the inverter is largely improved.

Figure 2.3(a) shows a voltage-fed *PWM* inverter circuit. The inverter uses the power transistor as the switching device. Again, the anti-parallel diode allows the current to feed back from the load. The device suitable for *PWM* inverters are power transistors (either bipolar or Darlington type), power *MOSFETs*, insulated gate bipolar transistors (*IGBTs*) [1], and *MOSFET*-controlled thyristors (*MCTs*) [43]. According to the modulation technique, *PWM* inverters can be classified into several categories such as sinusoidal *PWM*, selective harmonics elimination, adaptive current control *PWM*, ..., etc [5]. Figure 2.3(b) shows the basic principle of sinusoidal *PWM*, where a high frequency triangular wave serves as the carrier wave, and a low frequency sinusoidal wave periodically modulates the carrier wave to generate on-off signals for the inverter switches. For a 3-phase inverter, the modulating sinusoidal waves separate from each other at 120° .

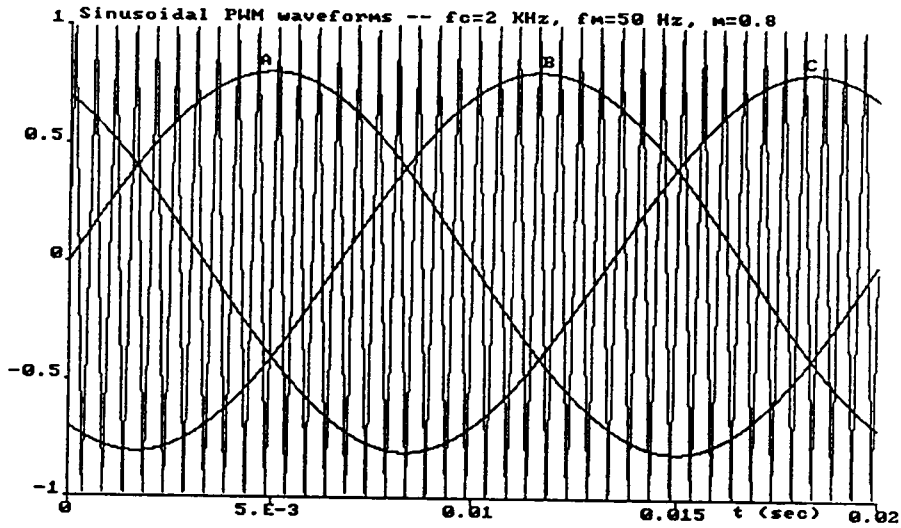
Figure 2.4(a) shows one cycle of output line voltage waveforms of the sinusoidal type *PWM* inverter. Figure 2.4(b) shows one cycle of output phase-to-neutral voltage waveforms. Notice that the chopping frequency is dependent on the carrier frequency and is much higher than that of square-wave inverters.

From the waveforms of line voltages V_{ab} , V_{bc} , and V_{ca} , it can be seen that the output voltage is synthesized by a group of voltage pulses with different pulse widths. The output phase voltages V_{an} , V_{bn} , and V_{cn} also show pulse-synthesized waveforms with less magnitude than line voltages.

Typically, the switching frequency of *PWM* inverters varies from less than 1 KHz to tenths of KHz, depending on the device switching speed and the performance requirement. Higher switching frequency introduces more switching losses but less harmonic contents. The device switching time is the key factor to achieving high frequency switching. The power *MOSFET* is the fastest device but suffers from

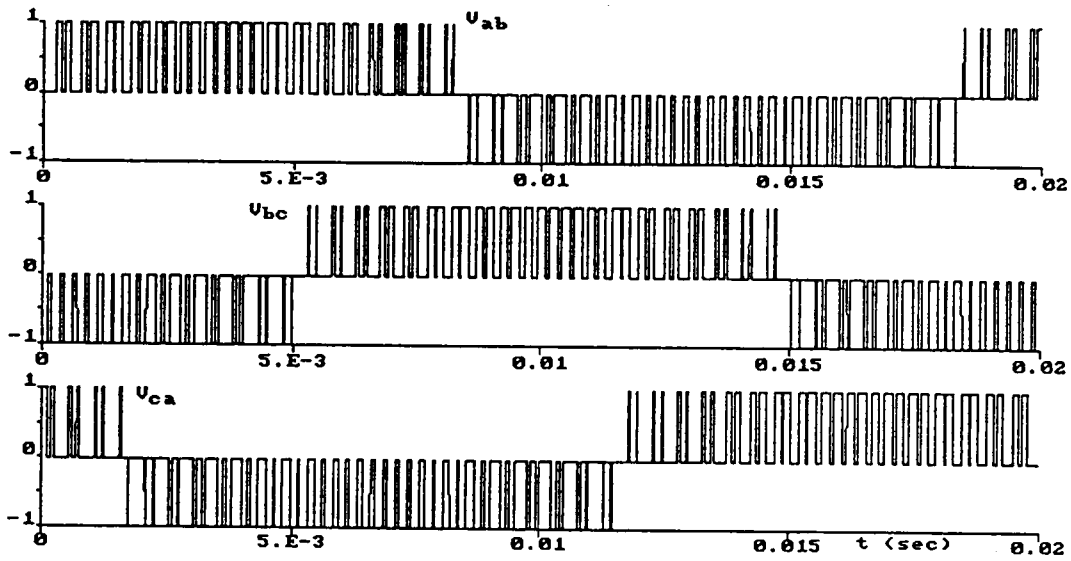


(a) Circuit diagram.

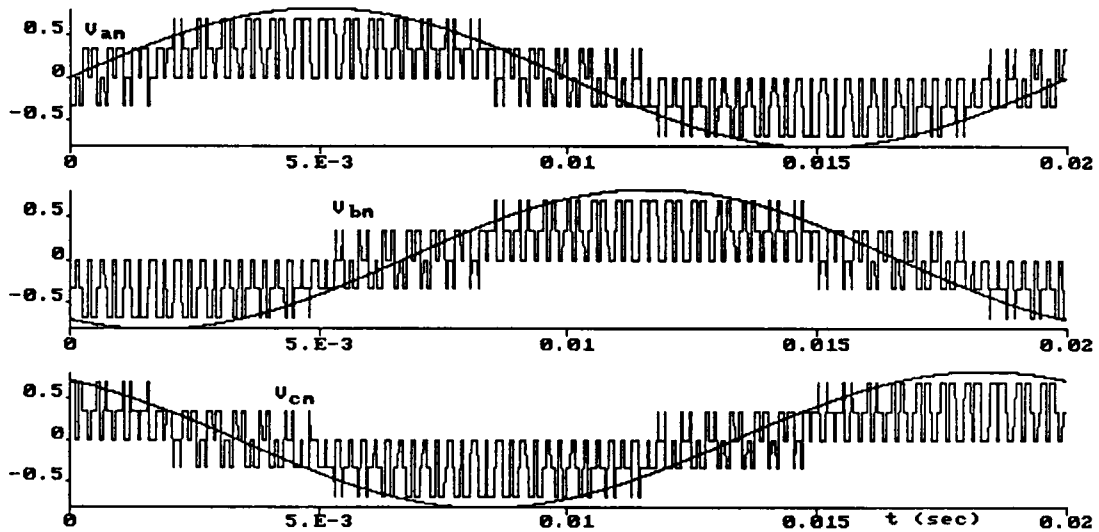


(b) Sinusoidal modulating method.

Figure 2.3: A voltage fed *PWM* inverter.



(a) Line-to-line voltage.



(b) phase-to-neutral voltage.

Figure 2.4: Output voltage waveforms of a sinusoidal *PWM* inverter.

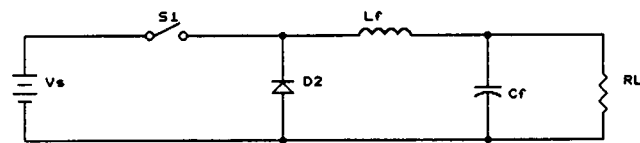
high conduction losses and slow body diode. Bipolar transistors switch at moderate speed but consume large base drive power. Recently developed *IGBTs* and *MCTs* are ideal for high frequency operation (>20 KHz) while maintaining low conduction drops and low gate drive consumption [1,43].

2.2 Resonant Mode *DC/DC* Converters

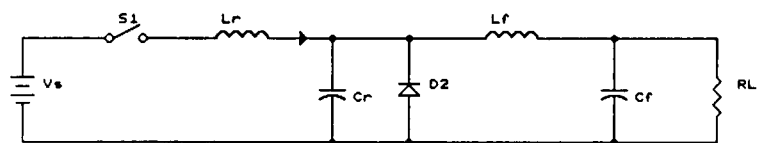
The main reason for employing high frequency *dc/dc* converters was to reduce the power density and to attain fast transient responses. Recent developments emphasized more features on bidirectional power flow [44], zero voltage switching [42], and ultra high switching frequency operation (>1 MHz) [17,18,25,42], etc.

The fundamental principle of resonant mode *dc/dc* converters can be seen from Figure 2.5 [26,42]. Figure 2.5(a) is a conventional switching mode *dc/dc* converter where the link voltage is a stiff *dc* voltage source V_s and the output is modulated by the switch S_1 . In order to provide a clean output voltage, the pulse-width modulated output is normally filtered by large-size passive components, L_f and C_f . Although the size can be reduced by increasing the switching frequency, the induced high frequency switching losses will be excessive as mentioned before. Typical switching frequency in this scheme seldom exceeds 50 KHz.

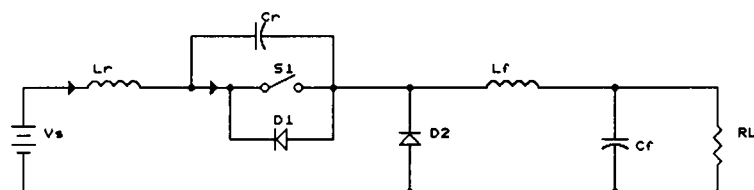
The resonant mode, differing from the switching mode by the insertion of a resonant tank L_r and C_r into the conventional converter, allows zero current switching as shown in Figure 2.5(b) or zero voltage switching as shown in Figure 2.5(c). The resonant frequency in zero current switching converters has been higher than 5 MHz [17,18,24,26]. For zero voltage switching, the frequency has been tested higher than 10 MHz [42]. Under Mega Hertz operation, the size of load side filters L_r and C_r of resonant mode converters is much smaller, and the efficiency is higher than those of traditional switching mode converter.



(a) A conventional switching mode dc/dc converter.



(b) A zero current switching resonant mode dc/dc converter.



(c) A zero voltage switching resonant mode dc/dc converter.

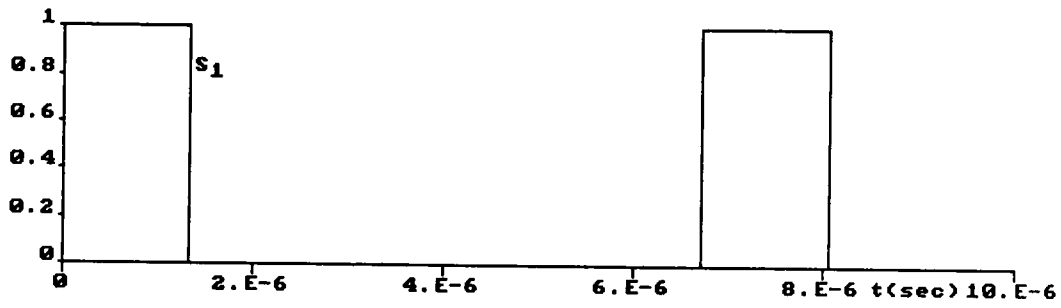
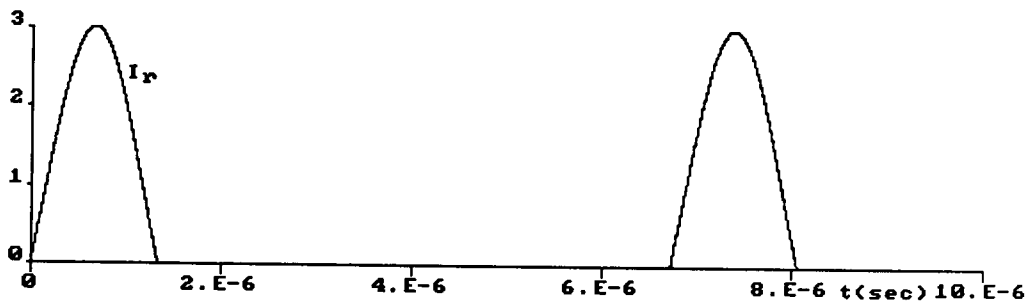
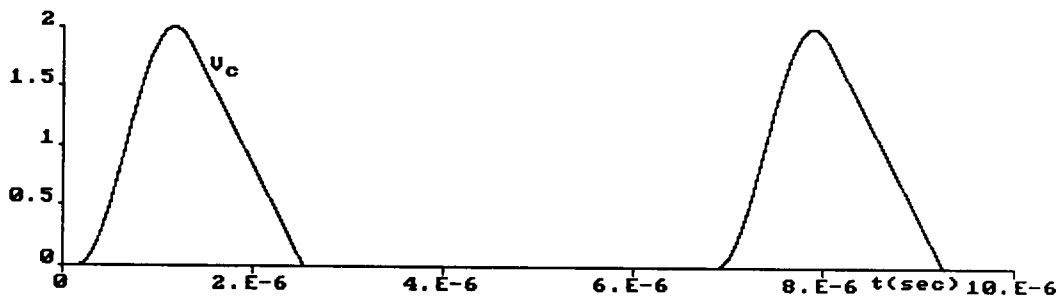
Figure 2.5: Comparison of conventional switching mode and resonant mode dc/dc converters.

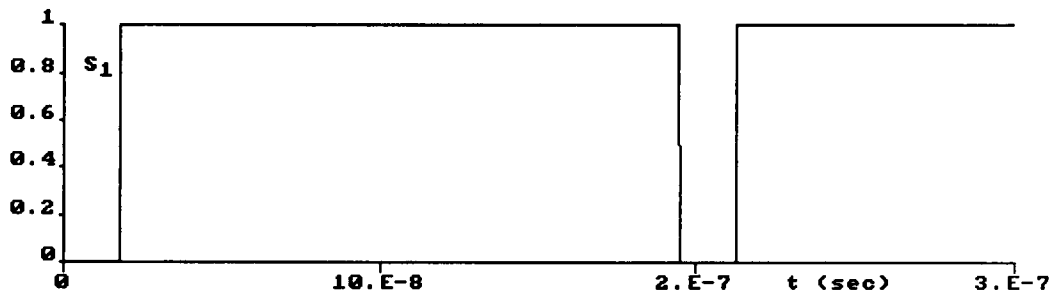
2.2.1 Operation of Zero Current Switching Resonant Link DC/DC Converter

For the circuit shown in Figure 2.5(b), the operation of zero current switching can be half-wave or full-wave. The full-wave operation requires anti-parallel diode to return excessive energy so that the output is less sensitive to load variation. The half-wave operation, on the other hand, does not have circulating current and is suitable for higher frequency operation. Figure 2.6 shows simulation results of the half-wave mode operation. The resonant switch S_1 turns on to start the resonance and off when the resonant current I_r drops to zero. Using the supply voltage V_s and the load current as the base, the resonant link voltage V_c is twice the base value, and the resonant link current is three times the base value. These high resonant link voltage and current are major disadvantages of resonant mode converters.

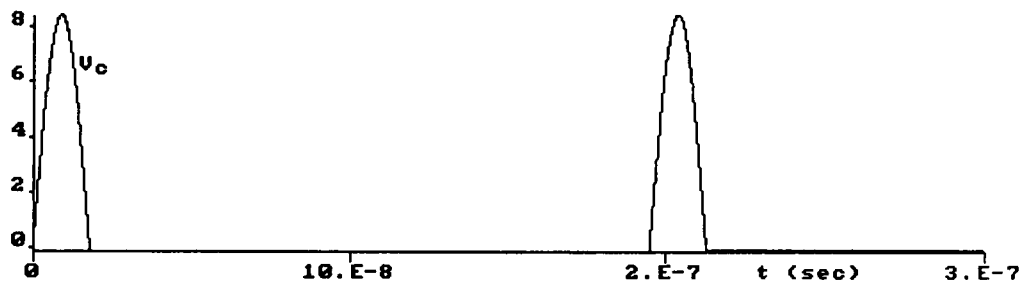
2.2.2 Operation of Zero Voltage Switching Resonant Link DC/DC Converter

For the circuit shown in Figure 2.5(c), the operation of zero voltage switching is shown in Figure 2.7. Contrary to the above zero current switching, the resonant switch S_1 turns on when the resonant voltage drops to zero and turns off to start the resonance. The resonant voltage is excessively high in this scheme. As shown in Figure 2.7, the peak voltage of V_{cr} is eight times the supply voltage. Such peak overshoot voltage and the switching frequency are functions of the load. A large load yields lower frequency and higher peak voltage, and vice versa. Because the switching frequency differs from the resonant frequency, the scheme is called “quasi-resonant link” converter.

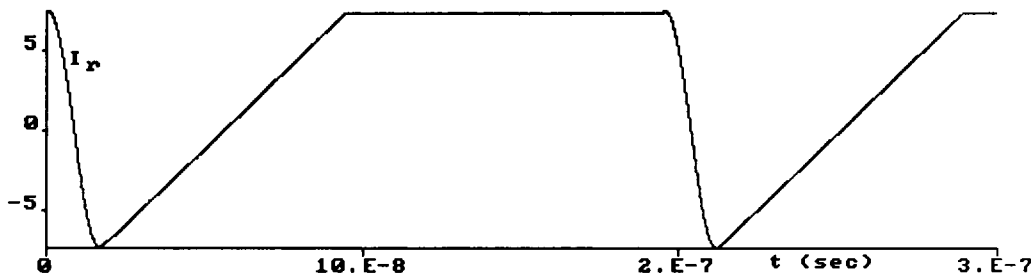
(a) resonant switch signal S_1 (b) resonant current I_r (c) resonant voltage V_c Figure 2.6: Operation of zero current switching resonant mode dc/dc converters.



(a) resonant switch S_1



(b) resonant voltage V_c



(c) resonant current I_r

Figure 2.7: Operation of zero voltage switching resonant mode dc/dc converters.

2.3 Resonant Link Inverters for AC Power Conversions

Resonant link inverters for *ac* power conversions differ from resonant mode *dc/dc* power supplies in output voltage and current levels and the requirement of bidirectional power flow. Because of high power in nature, the extremely high voltage or high current stress in resonant mode *dc/dc* converter is a fatal aspect for high power *ac* applications. Designing a resonant link inverter for bidirectional power flow is thus more challenging.

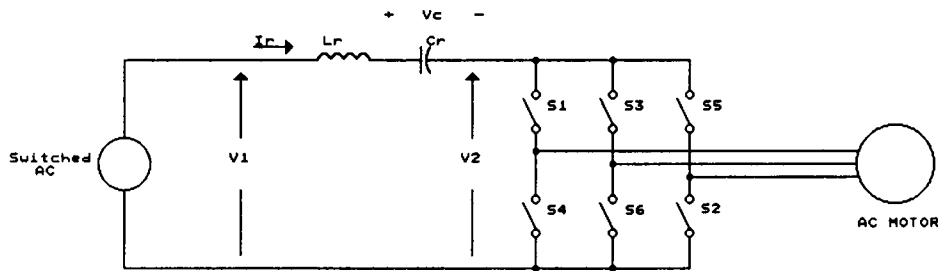
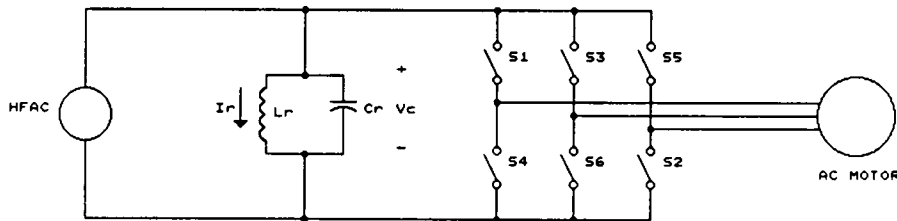
Because of the limitation of the device power level and efficiency considerations, the resonant frequency in *ac* power conversions is significantly less than that of *dc/dc* conversions. However, several resonant link inverters were recently proposed for ultrasonic operation (>20 KHz) [10,13,21,22,38,39,40].

The resonant tank can be connected in series or parallel between the power source and the converter, and the link voltage or current can be *ac* or *dc*. Depending on the link voltage or current source, resonant link inverters can be divided into two categories: resonant *ac* link and resonant *dc* link. This section will present recently developed resonant link circuit topologies and their simulation results.

2.3.1 Resonant AC Link Inverters

Resonant *ac* link converters employ a resonant circuit between the *ac* power source and the inverter bridge. The energy can be transferred directly between two *ac* sources without rectifying into *dc* source. The sizable passive components can thus be eliminated. The resonant tank is normally a small *LC* element, either series or parallel, to the converter circuit.

Figure 2.8 shows two newly developed resonant *ac* link inverters, where $S_1 \sim S_6$

(a) A series resonant *ac* link inverter.(b) A parallel resonant *ac* link inverter.Figure 2.8: Topologies of resonant *ac* link inverters.

are the conventional inverter bridge, and L_r and C_r represent the resonant tank. The two topologies are resonant *ac* link because the link current or voltage between the power source and the inverter is *ac*. Such *ac* voltage link systems require switches with reverse blocking and bidirectional switching capabilities. Two anti-parallel thyristors or two anti-parallel transistor-diode pairs are required.

A Series Resonant AC Link Inverter

Figure 2.8(a) is a simplified circuit of a series resonant *ac* link converter for *ac* machine drives. The complete power circuit for 3-phase power conversion is shown in Figure 2.9 [19,20]. The primary side switch bank $Q_{11} \sim Q_{34}$ and the secondary

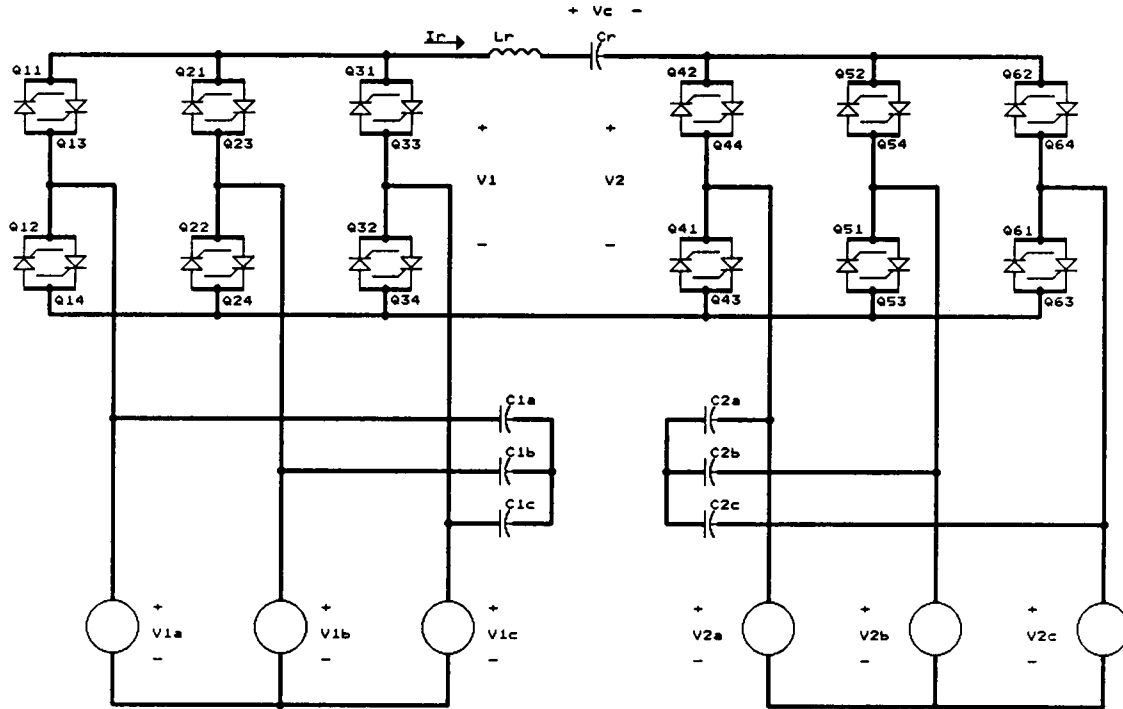
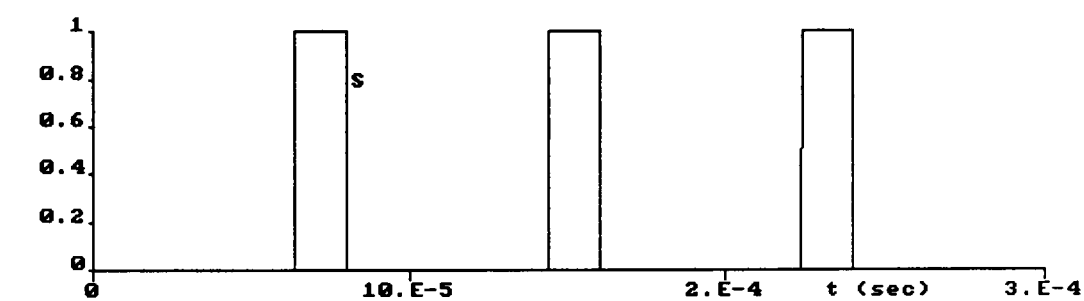


Figure 2.9: Power circuit of a series resonant *ac* link inverter.

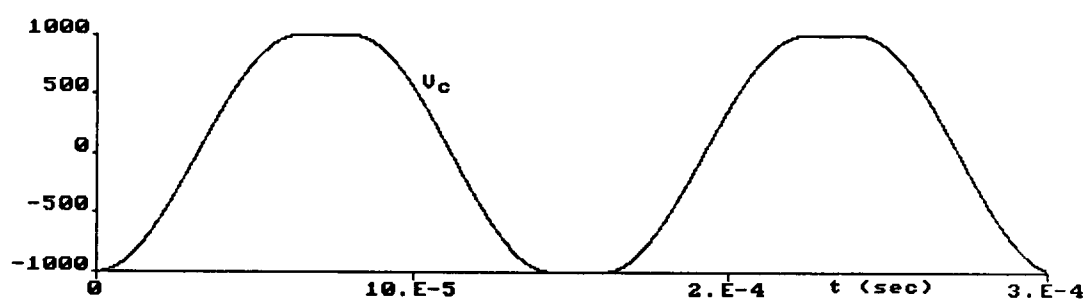
side switch bank $Q_{41} \sim Q_{64}$ provide the switched *ac* voltages V_1 and V_2 by different switching patterns. Two sets of capacitors $C_{1(abc)}$ and $C_{2(abc)}$ filter out high frequency components of $V_{1(abc)}$ and $V_{2(abc)}$. $V_{1(abc)}$ and $V_{2(abc)}$ are low frequency voltages and are assumed to be constant during L_r and C_r resonance. Each switching matrix contains 12 thyristors for 3-phase operation or 8 thyristors for single phase operation.

The resonant circuit L_r and C_r are excited by both source voltages V_1 and V_2 where V_1 results from the primary side switching matrix, and V_2 results from the secondary switching matrix. Figure 2.10 shows a typical operation of the above series resonant link power inverter.

This type of converter allows four-quadrant *dc/dc*, *dc/ac*, and *ac/ac* conversions.



(a) inverter switch

(b) resonant link current I_r (c) resonant link voltage V_c Figure 2.10: Operation of a series resonant ac link inverter.

The main drawback is the excessive amount of switching devices which must be equipped with reverse blocking capability. When applying this scheme to dc/ac conversion, one leg of thyristors at the dc side can be replaced by diodes [19].

A Parallel Resonant AC Power Inverter

Figure 2.8(b) is a simplified circuit of a parallel resonant ac link inverter for ac machine drives. The complete system should have an additional high frequency excitation source or a complete high frequency link system [38,39]. A typical system using bidirectional switches is shown in Figure 2.11 [38]. Each high frequency source is linked by a parallel LC circuit which provides a high impedance to the high frequency current and a low impedance to the system low frequency current. The output voltage is modulated by the inverter switches. Because the switching only occurs at zero voltage crossing, the low frequency voltage synthesis cannot be modulated by traditional PWM methods. The new modulation method is called “pulse density modulation” (PDM) [38,39] or “integral PWM ” ($IPWM$) [13].

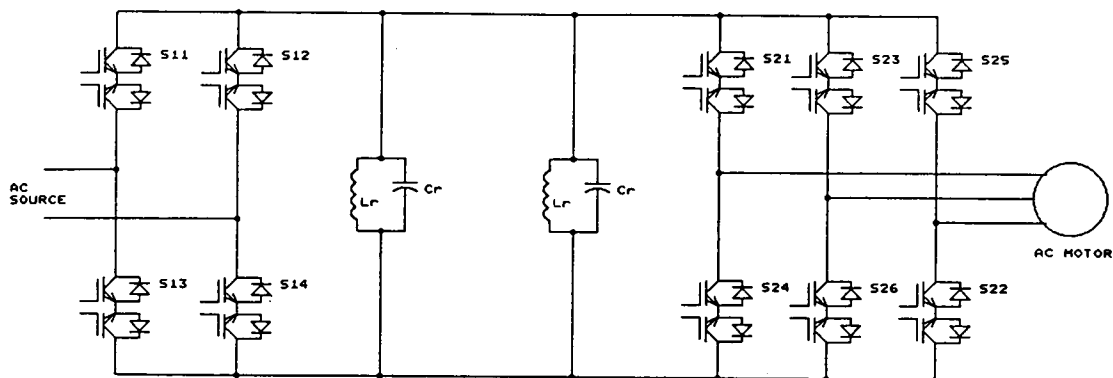


Figure 2.11: A parallel resonant ac link inverter system.

Figure 2.12 shows typical waveforms of the parallel resonant ac link inverter

system. The link voltage V_c is high frequency ac , while the output voltage V_o is pulse modulated low frequency ac . The density of pulses over the low frequency cycle is modulated according to a low frequency reference signal V_o^* .

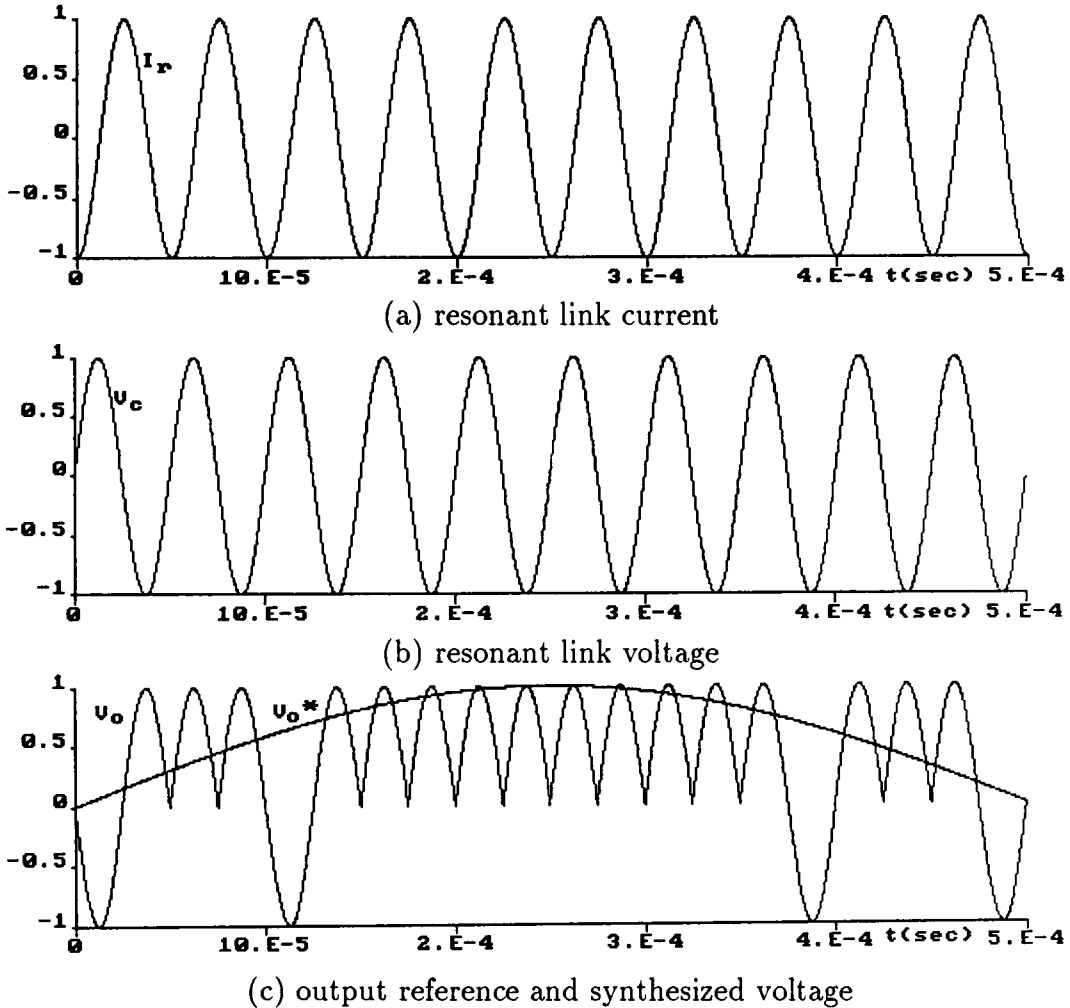
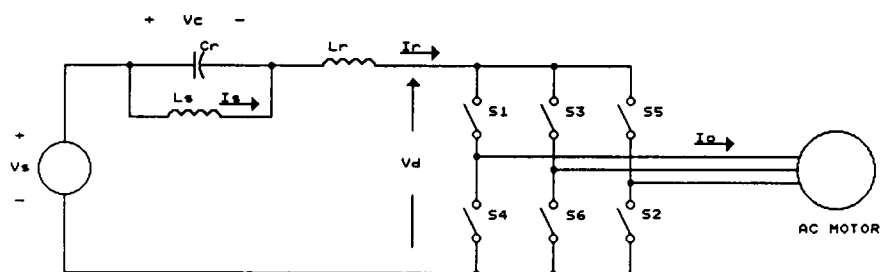


Figure 2.12: Operation of a parallel resonant ac link inverter.

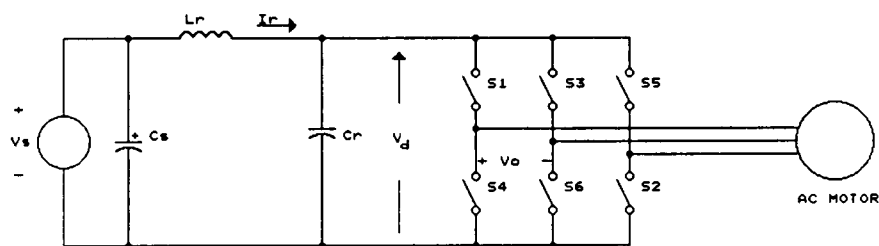
2.3.2 Resonant DC Link Inverters

Similar to the duality in resonant ac link inverters, resonant dc link inverters also present dual circuit topologies: series resonant dc link and parallel resonant dc link

as shown in Figure 2.13 [13,28]. The series resonant *dc* link inverter input is pulse *dc* current while the parallel resonant *dc* link inverter is pulse *dc* voltage.



(a) A series resonant *dc* link inverter.



(b) A parallel resonant *dc* link inverter.

Figure 2.13: Circuit topologies of resonant *dc* link inverters.

Instead of using bidirectional switches for resonant *ac* link inverters, resonant *dc* link inverters need only unidirectional switches because either the link current (series resonant) or the link voltage (parallel resonant) is *dc*. To allow bidirectional power flow, the series resonant *dc* link system provides *ac* voltage across the inverter, and the device should have reverse blocking capability such as thyristors. Similarly, the parallel resonant *dc* link system provides reverse current flow path in inverter switches which means anti-parallel diodes are required.

A Series Resonant *DC* Link Inverter

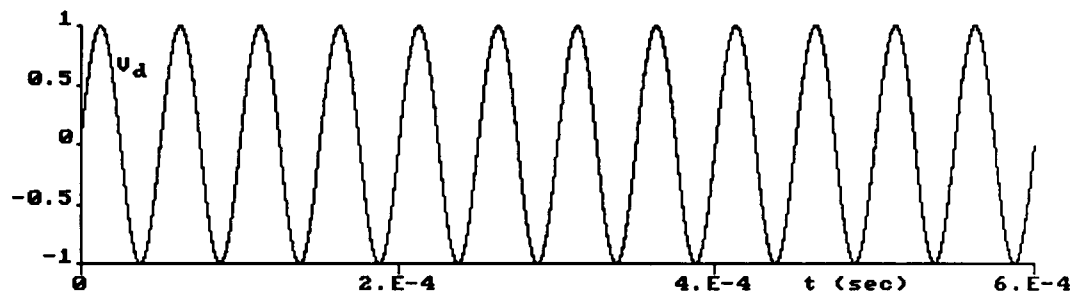
The series resonant *dc* link inverter provides a pulse *dc* current source to the inverter by employing a series *LC* tank across a constant current source. As shown in Figure 2.13(a), a large inductor L_s produces the current source I_s , and small *LC* components L_r and C_r generate *dc* current pulse train I_r . The peak value of the resonant current is twice the source current. Figure 2.14 shows simulation results of the series resonant *dc* link inverter.

Because the voltage across the inverter bridge V_d is *ac*, the switch must be able to block the reverse voltage. The traditional thyristor has reverse blocking capability, but its switching speed is not suitable for high frequency operation. New switching devices such as *IGBTs* and *MCTs* theoretically possess reverse blocking capabilities [3,43], but none of the currently available high frequency devices are capable of blocking full reverse voltage. A possible solution is to series a diode with a fast switching device to replace the slow thyristor, which inevitably sacrifices the efficiency because of the additional voltage drop.

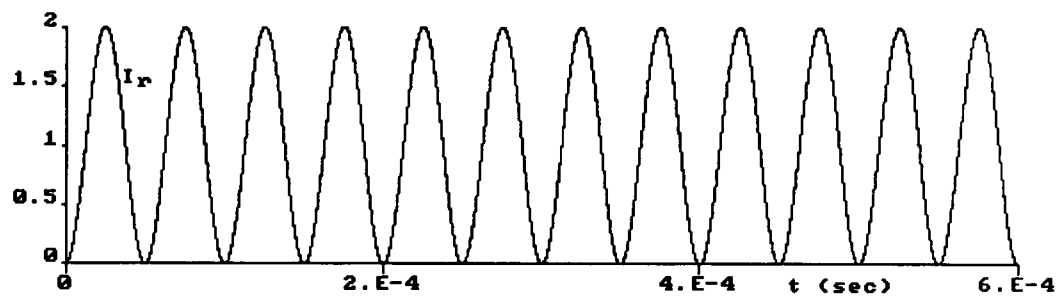
The low frequency sinusoidal output current I_o is synthesized by varying the density of high frequency pulse trains. A reference current I_o^* is provided for modulation. The modulation technique can thus be defined as “current pulse density modulation” (*CPDM*) to distinguish it from “voltage pulse density modulation” (*VPDM*) employed in parallel resonant inverters.

A Parallel Resonant *DC* Link Inverter

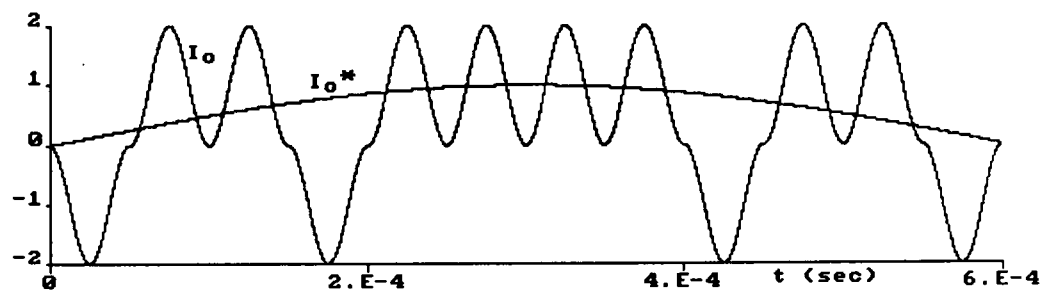
Instead of inputting *dc* current source to the inverter in the series resonant *dc* link inverter, the parallel resonant *dc* link inverter inputs *dc* voltage source to the inverter. Figure 2.13(b) shows a simplified circuit of parallel resonant *dc* link inverter where C_s is a filter capacitor, and L_r and C_r constitute a resonant tank. When compare



(a) resonant link voltage



(b) resonant link current



(c) output reference and synthesized current

Figure 2.14: Operation of a series resonant *dc* link inverter.

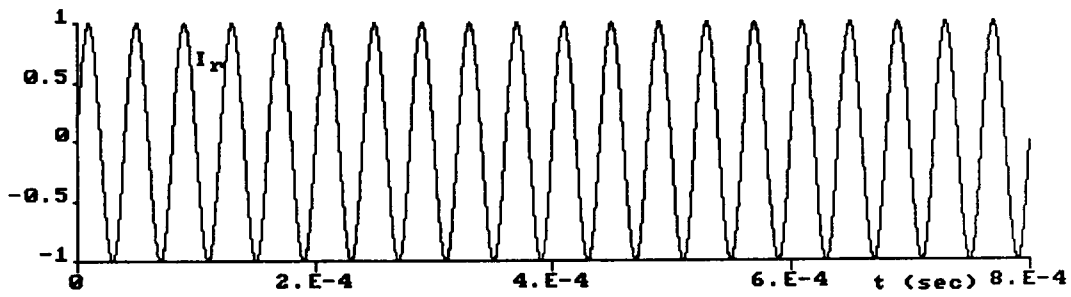
to traditional *dc* link inverters, the difference can be seen in the circuit where the additional small-sized resonant tank is inserted into the location between the stiff *dc* voltage source and the inverter bridge. Figure 2.15 shows typical operation of the resonant *dc* link inverter.

The resonant link current I_r is *ac*, but the inverter input source V_d is *dc* voltage pulse train. The output voltage can be synthesized by different switching patterns. A typical output waveform V_o is shown with a reference voltage V_o^* . The inverter devices are switched at zero-voltage crossing intervals, causing integral pulse modulation. The modulation technique can be defined as voltage pulse density modulation (*VPDM*).

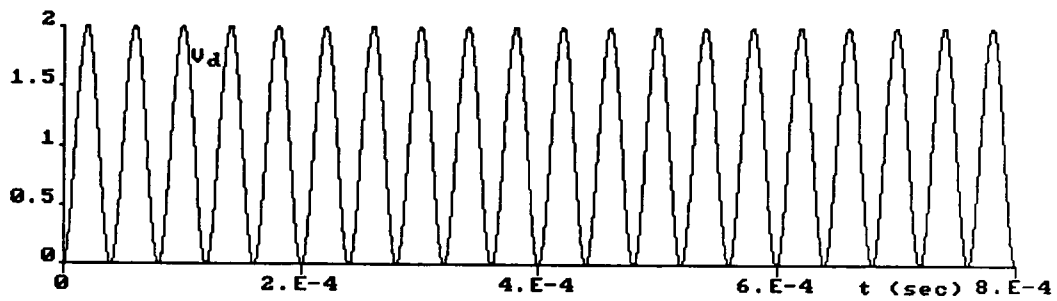
Because the resonant link circuit should allow the load current to flow back from the inverter switches, the inverter switch is normally constituted of a transistor-diode pair. The devices which have reported successful operation with parallel resonant *dc* link inverters are power transistors [13,14] and *IGBTs* [23]. The resonant *dc* link inverter is most feasible to be implemented with currently available switching devices. A resonant *dc* link inverter has been reported at 65 KHz resonant frequency [23].

2.3.3 A Quasi-Resonant Notch Link Inverter

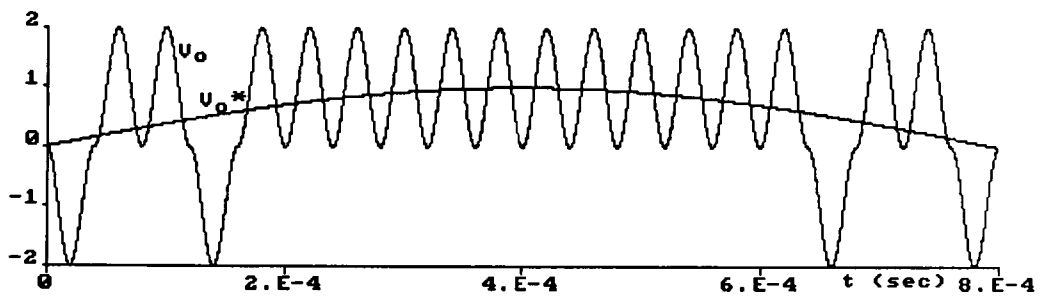
The quasi-resonant link inverter is defined as an inverter with switching frequency differing from resonant frequency [41]. The resonant frequency is produced by the *LC* resonant components, but the inverter switching frequency is regulated by loading conditions or control circuits. In resonant mode *dc/dc* converters, the switching frequency is highly dependent on the load. The *dc/dc* converters shown in Section 2.2 are inherently quasi-resonant link. In *ac* power applications, however, such inefficient load dependent and high voltage and current stress quasi-resonant link converter should not be considered. A new concept of quasi-resonant notch link



(a) resonant link current



(b) resonant link voltage



(c) output reference and synthesized voltage

Figure 2.15: Operation of a parallel resonant *dc* link inverter.

inverter has been proposed [22], and its simplified quasi-resonant notch link inverter circuit is shown in Figure 2.16.

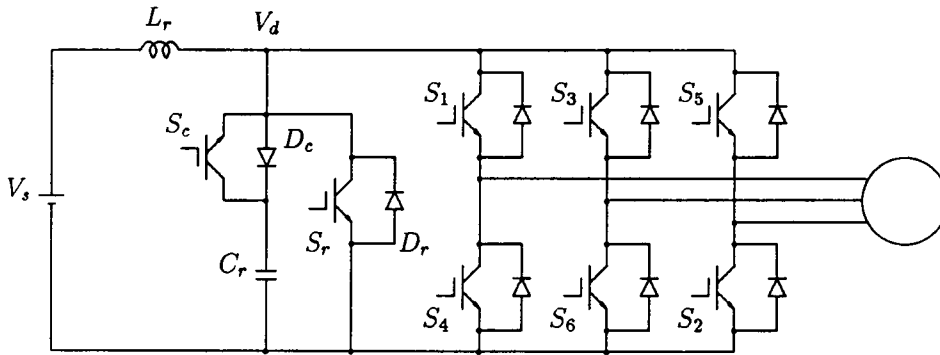
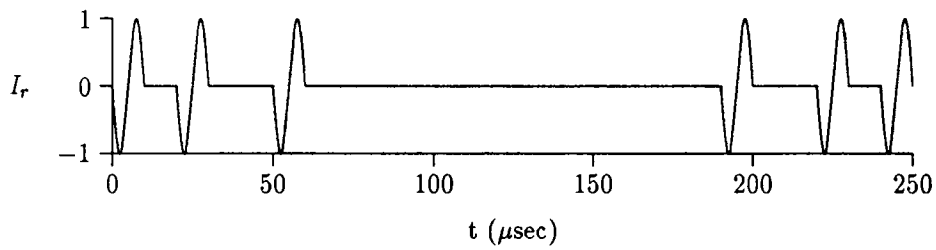


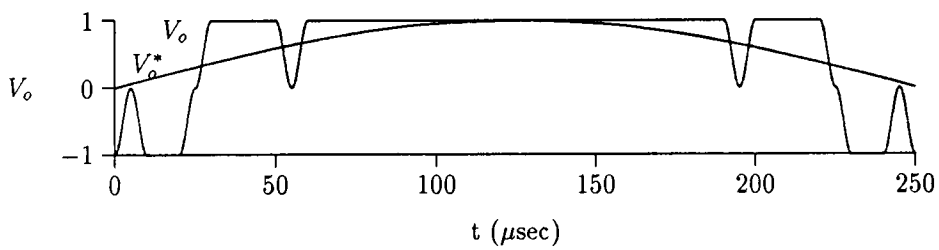
Figure 2.16: A quasi-resonant notch link inverter.

The circuit employs an additional switch pair to regulate the switching frequency, resulting in smooth notches on the dc link voltage. The resolution of notches depends on performance requirement. A typical example is to have the resonant frequency at 100 KHz to ensure the PWM resolution and the resulting switching frequency at 20 KHz to avoid audible noises. Because the inverter switches are not necessary to chop at integral cycle, the resulting modulation can be optimized PWM , and the performance is significantly better than that of resonant link inverters [22]. Figure 2.17 shows typical resonant notch link inverter operation.

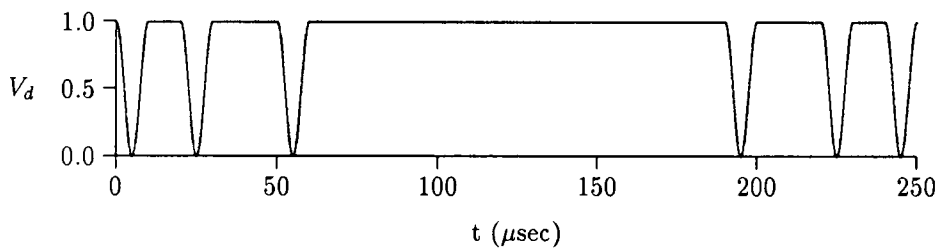
The waveforms show a 100 KHz resonant frequency and notching type PWM operation. The switching frequency can be easily regulated by minimum ripple, least current error square, or other optimum PWM techniques [22]. As shown in Figure 2.17, during diode blocking the resonant link current I_r is zero, and the resonant link voltage V_d maintains constant. The operation is similar to dc link PWM operation except that the resonant notch link circuit creates smooth notches



(a) resonant link current

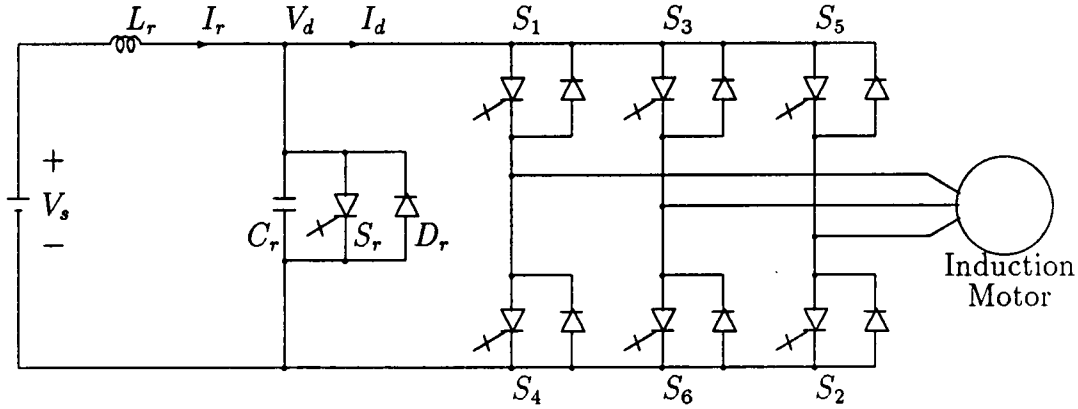


(b) output reference and synthesized voltage



(c) quasi-resonant link voltage

Figure 2.17: Typical operation of a quasi-resonant notch link inverter.

Figure 3.1: Resonant *dc* link inverter for induction motor drive.

The switch S_r , which establishes the positive initial current (I_m) in L_r can be shown as redundant with appropriate control of the inverter. The inverter devices are switched at zero voltage ($V_d = 0$) to fabricate *PWM* voltage waves for the motor.

A fundamental concept is that the resonant capacitor voltage V_d is determined by the capacitor current, the difference between the resonant inductor current I_r and the inverter input current I_d . The current I_d is a function of motor phase currents and the inverter switching pattern. If the current I_d is constant during a resonant cycle, then the maximum and minimum (to assure zero crossing) values of V_d are determined by the initial resonant link current. A detailed parametric study showing how the initial current affects the capacitor voltage is presented.

Figure 3.2 shows the equivalent circuit of the resonant link system during resonant interval. Suppose the inductance of an *ac* machine is large enough, the inverter input current I_d will remain constant during a resonant cycle. To simplify the analysis, the circuit modeling in this chapter assumes I_d is a constant current source. In actual practice, I_d may vary and its polarity may be positive or negative.

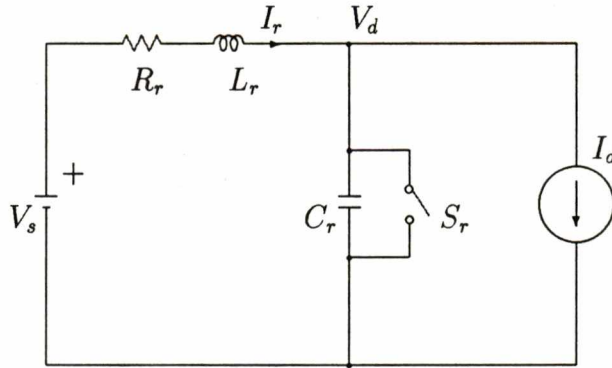


Figure 3.2: Equivalent circuit of the resonant link inverter during resonant interval.

3.2 Mathematical Analysis

According to Figure 3.2, the network equations can be written as

$$V_s(t) = I_r(t)R_r + \frac{dI_r(t)}{dt}L_r + V_d(t) \quad (3.1)$$

$$I_r(t) = I_d(t) + C_r \frac{dV_d(t)}{dt} \quad (3.2)$$

with initial states $V_d(0) = 0$ and $I_r(0) = I_m$. Solving equations (3.1) and 3.2) yields

$$V_d(t) = (V_s - I_d R_r) \left[1 - \frac{1}{\sqrt{1 - \zeta^2}} e^{-\alpha t} \sin(\omega t + \theta) \right] + \frac{(I_m - I_d) Z_r}{\sqrt{1 - \zeta^2}} e^{-\alpha t} \sin \omega t \quad (3.3)$$

$$I_r(t) = I_d + e^{-\alpha t} \left[(I_m - I_d) \cos \omega t + \frac{2V_s - (I_d + I_m) R_r}{2\omega L_r} \sin \omega t \right] \quad (3.4)$$

where

$$\omega_r = 1/\sqrt{L_r C_r} = \text{resonant frequency (zero damping)}$$

$$\alpha = R_r/2L_r = \text{decrement factor}$$

$$\zeta = R_r/2Z_r = \text{damping ratio}$$

$$Z_r = \sqrt{L_r/C_r} = \text{characteristic impedance}$$

$\omega = \omega_r \sqrt{1 - \zeta^2}$ = oscillation frequency

$$\theta = \cos^{-1} \zeta = \tan^{-1} \frac{\sqrt{1 - \zeta^2}}{\zeta}$$

In (3.3) the voltage component $I_m Z_r$ largely affects the resonant voltage $V_d(t)$. The occurrence of voltage overshoot means that the maximum voltage is higher than $2V_s$, and the zero voltage crossing means that the minimum voltage is lower than zero. Thus the study of extreme points of the resonant voltage will provide sufficient information on both voltage overshoot and zero voltage crossing problems. The extreme points can be obtained by solving $dV_d(t)/dt = 0$, i. e.,

$$\frac{dV_d(t)}{dt} = \omega_r e^{-\alpha t} \left[\frac{2(V_s - I_d R_r) + (I_d + I_m) R_r}{2\sqrt{1 - \zeta^2}} \sin \omega t + (I_m - I_d) Z_r \cos \omega t \right] = 0 \quad (3.5)$$

which gives

$$\tan \omega t_n = \frac{2\sqrt{1 - \zeta^2}(I_d - I_m) Z_r}{2V_s - (I_d + I_m) R_r} \quad (3.6)$$

The time t_n can be written in the form

$$t_n = [n\pi + \tan^{-1} \frac{2\sqrt{1 - \zeta^2}(I_d - I_m) Z_r}{2V_s - (I_d + I_m) R_r}] / \omega \quad n = 1 \text{ and } 2 \quad (3.7)$$

where $n = 1$ corresponds to the time where the voltage is at maximum, and $n = 2$ corresponds to the time where the voltage is at minimum. The voltage at extreme points can then be given as

$$V_d(t_1) = (V_s - I_d R_r) \left[1 - \frac{1}{\sqrt{1 - \zeta^2}} e^{-\alpha t_1} \sin(\omega t_1 + \theta) \right] + \frac{(I_m - I_d) Z_r}{\sqrt{1 - \zeta^2}} e^{-\alpha t_1} \sin \omega t_1 \quad (3.8)$$

where

$$t_1 = [\pi + \tan^{-1} \frac{2\sqrt{1 - \zeta^2}(I_d - I_m) Z_r}{2V_s - (I_d + I_m) R_r}] / \omega$$

$$V_d(t_2) = (V_s - I_d R_r) \left[1 - \frac{1}{\sqrt{1 - \zeta^2}} e^{-\alpha t_2} \sin(\omega t_2 + \theta) \right] + \frac{(I_m - I_d) Z_r}{\sqrt{1 - \zeta^2}} e^{-\alpha t_2} \sin \omega t_2 \quad (3.9)$$

where

$$t_2 = [2\pi + \tan^{-1} \frac{2\sqrt{1-\zeta^2}(I_d - I_m)Z_r}{2V_s - (I_d + I_m)R_r}] / \omega$$

By introducing the two parameters, $\mathcal{X}_L = I_d Z_r / V_s$ (load current factor) and $\mathcal{X}_0 = I_m Z_r / V_s$ (initial current factor), a general normalized form of (3.8) and (3.9) can be given as

$$\begin{aligned} \frac{V_d(t_n)}{V_s} = & (1 - 2\mathcal{X}_L \zeta) [1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\frac{\zeta}{\sqrt{1-\zeta^2}} t_n} \sin(\omega t_n + \theta)] \\ & + \frac{1}{\sqrt{1-\zeta^2}} e^{-\frac{\zeta}{\sqrt{1-\zeta^2}} t_n} (\mathcal{X}_0 - \mathcal{X}_L) \sin \omega t_n \end{aligned} \quad (3.10)$$

In (3.10), θ is a function of ζ , ω and ζ are system constants, and $\mathcal{X}_0$ and $\mathcal{X}_L$ are variables related to current. Figure 3.3 shows the typical normalized current and voltage waves in the resonant link where I_r is normalized with the base current V_s / Z_r and V_d is normalized with the *dc* voltage V_s .

Voltage $V_d(t_1)$ is the maximum voltage at time t_1 , and $V_d(t_2)$ is the minimum voltage at time t_2 . The voltage overshoot occurs when $V_d(t_1) > 2V_s$, and the zero voltage crossing is lost when $V_d(t_2) > 0$.

Normally, the inverter input current I_d varies between $-I_L$ and $+I_L$ where I_L is the peak inverter phase current. It can be seen from (3.8) that the value of $I_m - I_d$ is the dominant factor causing voltage overshoot. Equation (3.9) shows that the same factor also affects zero voltage crossing. Thus controlling the value of $I_m - I_d$, i.e., controlling the initial current can solve both the voltage overshoot and zero voltage crossing problems. Evidently, for a certain inverter input current I_d , different initial inductor currents I_m will yield different maximum and minimum values of V_d . Figure 3.4 shows the maximum voltage $V_d(t_1)$ and the minimum voltage $V_d(t_2)$ as functions of I_m and which have been solved in (3.8) and (3.9) respectively under the condition $I_d = -I_L$. The resonant circuit is assumed to operate at 50KHz with $Q = 150$.

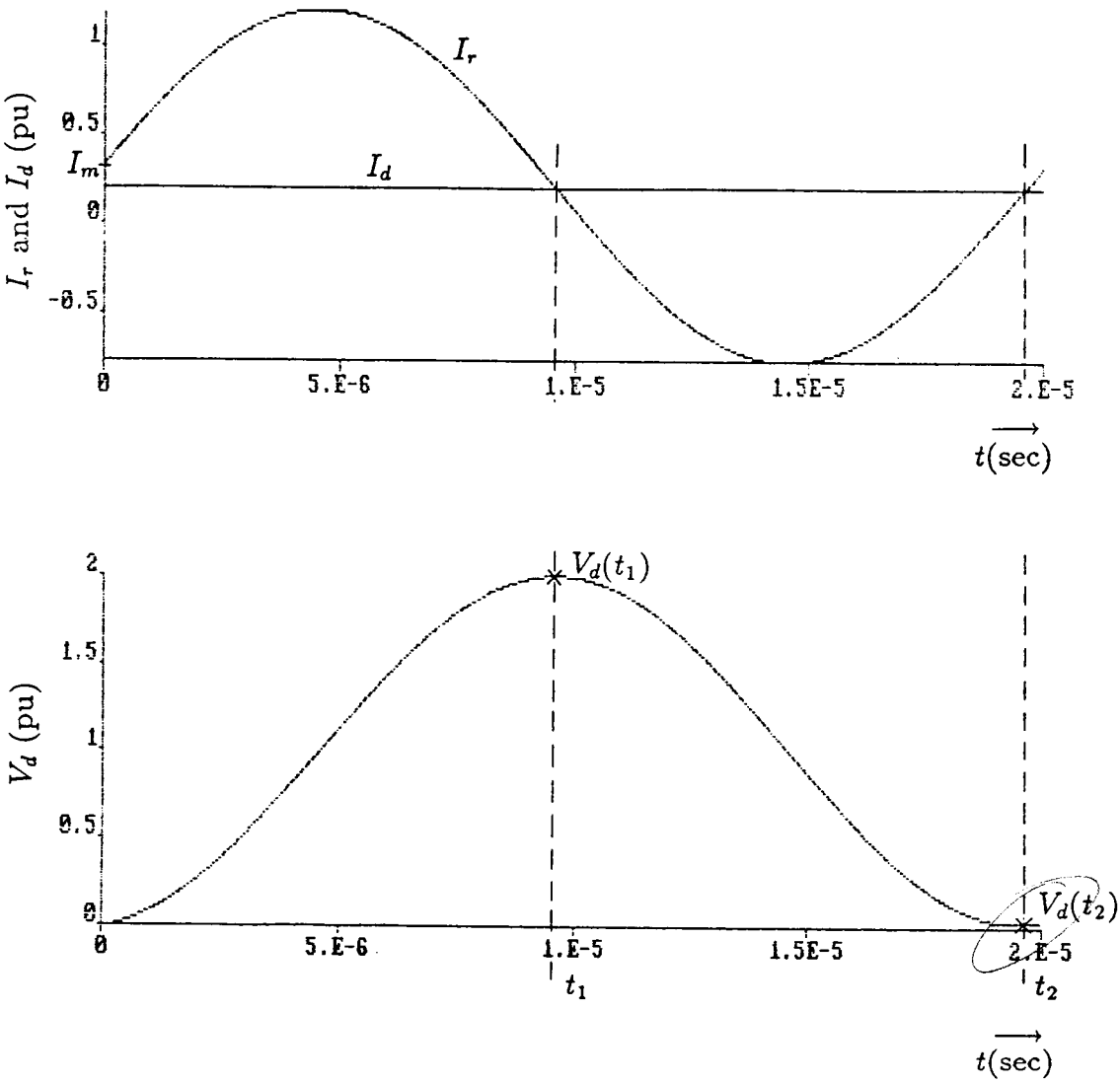


Figure 3.3: Current and voltage waveforms in a resonant cycle.

The characteristic impedance is chosen as $Z_r = V_s/I_L$ and in Figure 3.4 the current I_m is normalized with the base current $I_L = V_s/Z_r$. A few interesting observations can be made from Figure 3.4. If the initial current I_m is set to be I_L , then the voltage $V_d(t_1)$ swings to $3.2V_s$. Such a high voltage overshoot can be expected because of the large initial current. The magnitude of $V_d(t_1)$ becomes minimum when $I_m = -1\text{pu}$, i.e., $I_m - I_d = 0$. But under such $I_m = I_d$ condition, the zero voltage crossing is lost. Since the control of negative $(I_m - I_d)$ is impractical under nonzero switching time, the initial current is always chosen to be positive. The desired value of $(I_m - I_d)$ can be easily designed from the graphical plots. A typical solution point is $(I_m - I_d) = 0.3I_L$ as indicated in Figure 3.4. This, however, causes an unavoidable voltage overshoot of a few percent. If $I_m - I_d$ is larger, the voltage overshoot increases, but the value of $V_d(t_2)$ decreases and results in an increase of zero crossing interval. It is important to note that I_d normally increases during a resonant cycle because of high V_d magnitude. To allow for this current rise, the initial current should be slightly higher than the selected value (i.e., $0.3I_L$) of Figure 3.4.

Figure 3.5 shows the general plot of $V_d(t_1)$ from (3.8) for different inverter input currents which includes the case of $I_d = -1.0I_L$. Figure 3.6 gives the general plot of $V_d(t_2)$ from (3.9) including the case of $I_d = -1.0I_L$.

It is evident from the curves that the initial current I_m has to be made programmable as a function of the inverter input current I_d in order to satisfy the criteria required for reducing voltage overshoot and insuring zero voltage crossing. Typical zone of I_m variation is shown in Figures 3.5 and 3.6. The zero skew of 0.3 occurs because of positive initial current requirement which is explained in Figure 3.4. For example, if the inverter is unloaded (i.e., $I_d = 0$), the initial current $I_m = 0.3$ will satisfy the initial current requirement.

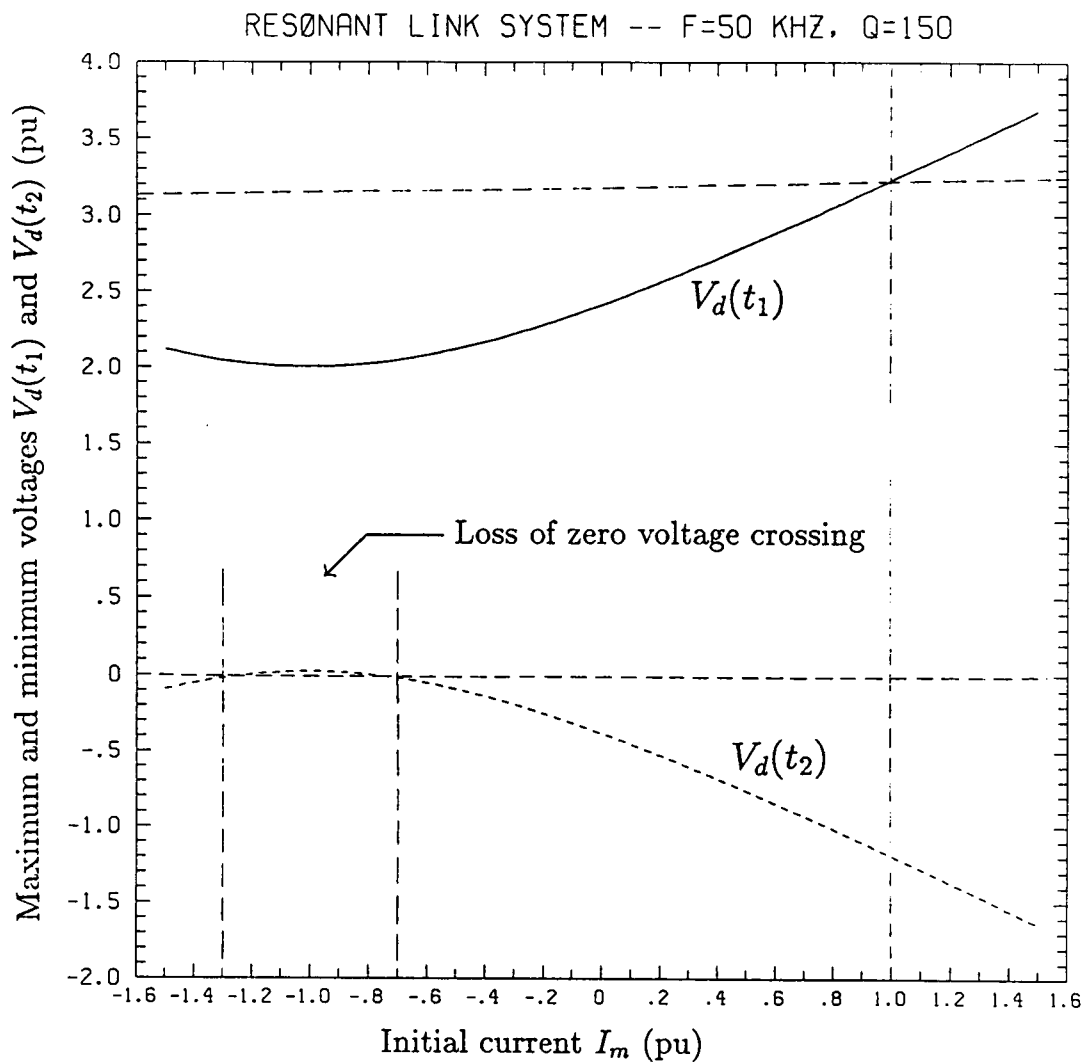


Figure 3.4: Maximum and minimum resonant circuit voltages as functions of initial current I_m ($I_d = -I_L$).

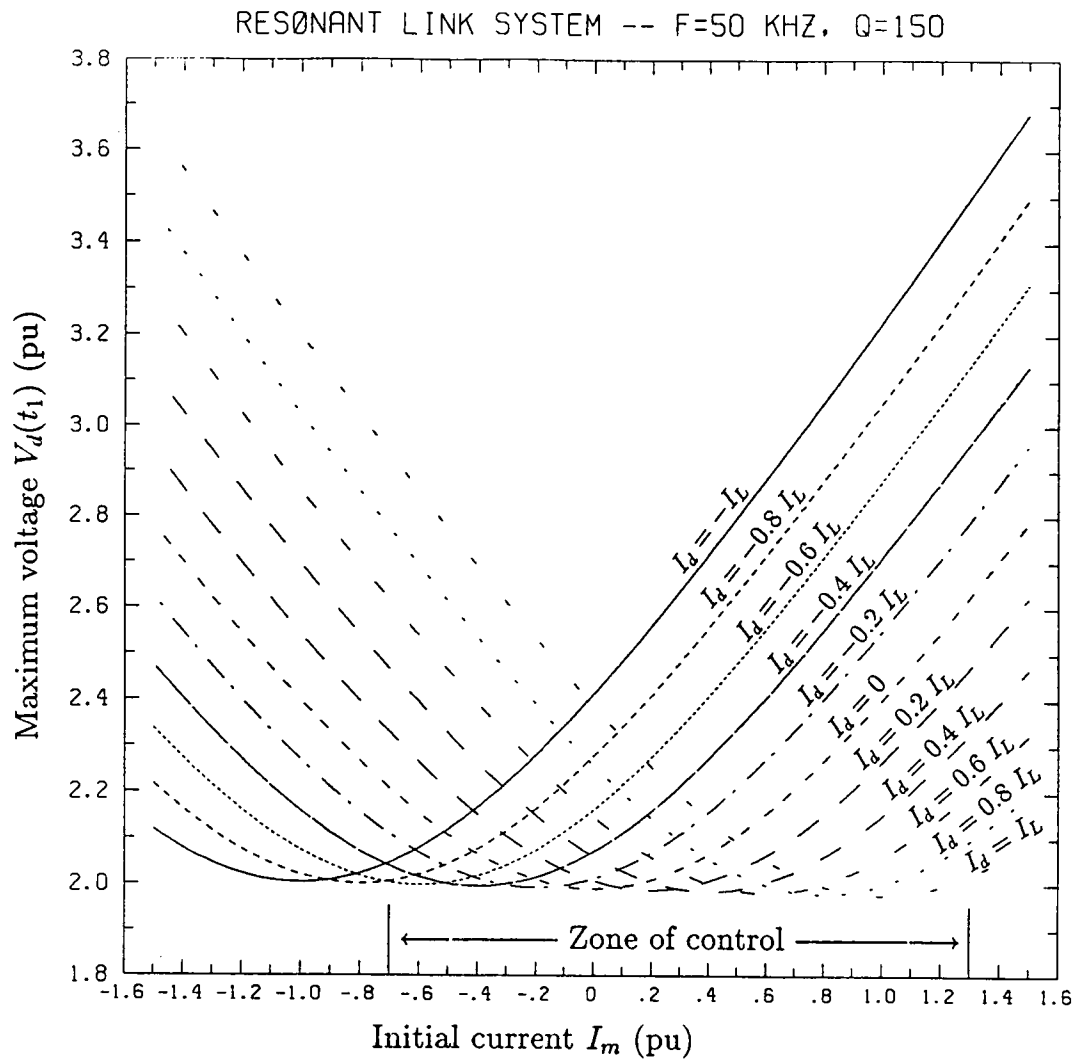


Figure 3.5: Maximum resonant circuit voltage [$V_d(t_1)$] as function of I_m for different inverter input currents.

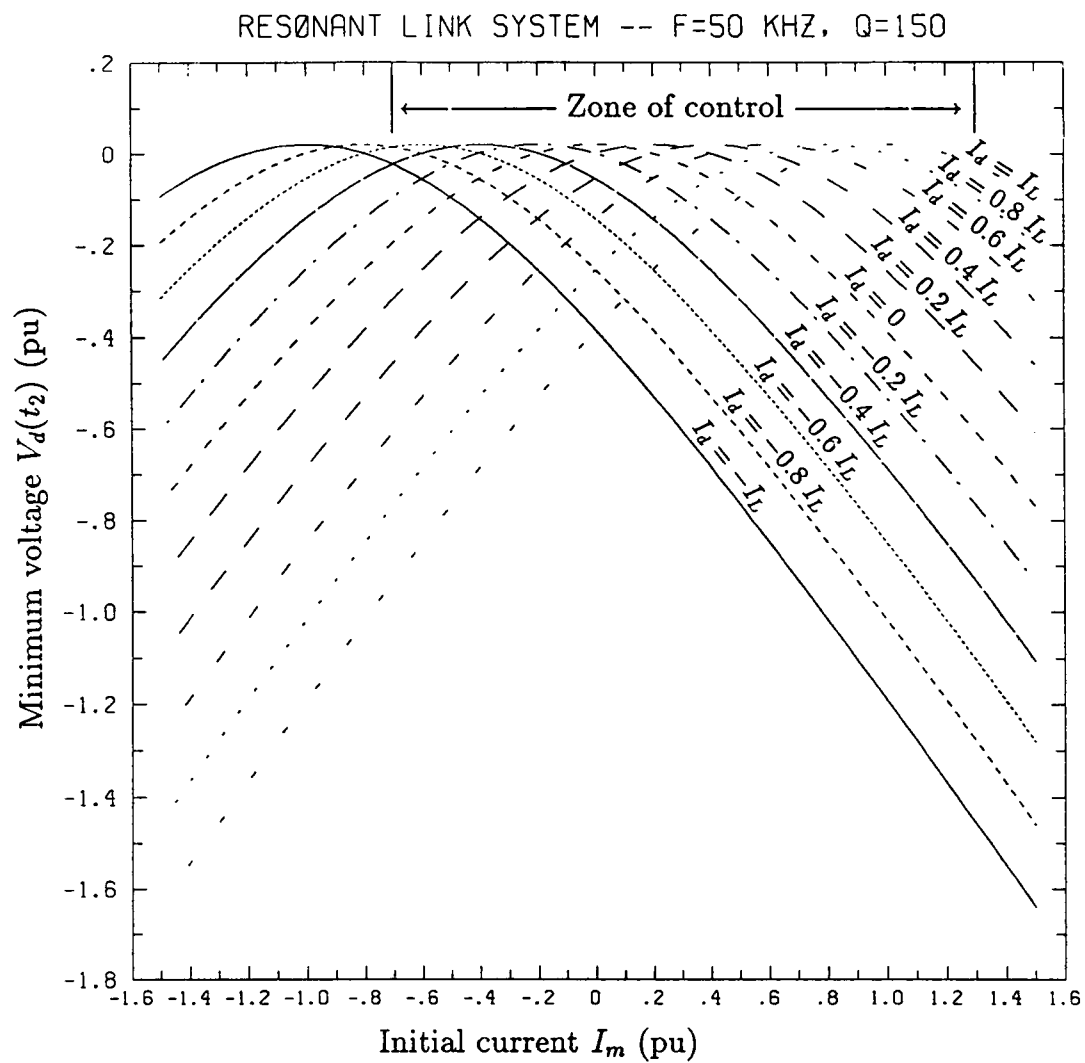


Figure 3.6: Minimum resonant circuit voltage $[V_d(t_2)]$ as function of I_m for different inverter input currents.

If the load current does not remain constant during resonance interval, the initial current can be adjusted to a higher value to prevent zero voltage crossing failure. The load model is not necessary to be precise because the purpose of this modeling and analysis is to predict a setup value and not to decide the final setup value. The actual initial current requirement can be easily adjusted by trimming a potentiometer and should be obtained from the experiment. An experimental setup and its results are presented in the following section.

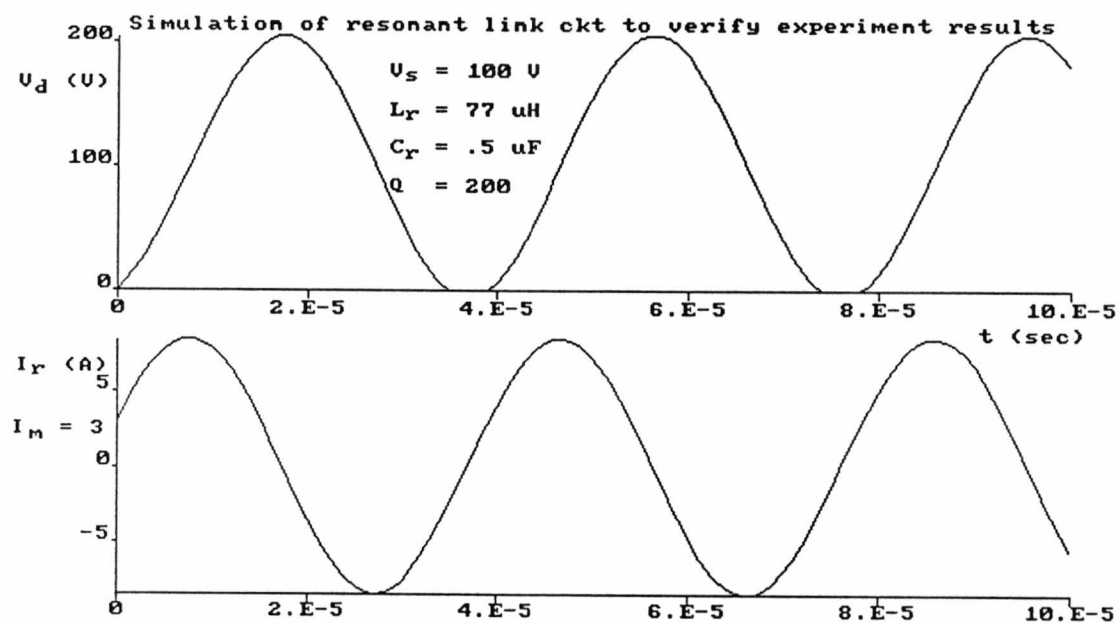
3.3 Experimental Results of Resonant DC Link Circuit

To verify the initial current requirement in the resonant *dc* link circuit, an experiment has been carried. The resonant circuit consists of a $77\ \mu\text{H}$ inductor and a $.5\ \mu\text{F}$ capacitor. The resonant frequency is 25.65 KHz, and the characteristic impedance is $12.4\ \Omega$. The test circuit is the same as Figure 3.1 except the inverter is unloaded. The components are listed in Table 3.1.

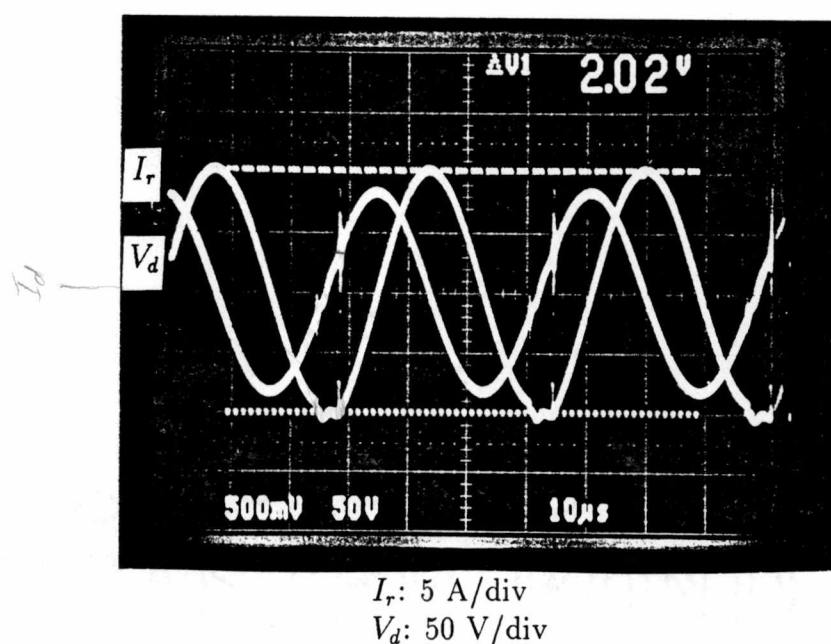
Table 3.1: Components for resonant link current initialization test.

Device	Specification	Note
S_r	500 V, 13 A	a <i>MOSFET</i> MTH13N50 series with a diode MUR1560
D_r	600 V, 15 A	Ultra fast recovery diode MUR1560
L_r	$77\ \mu\text{H}$	19-strand AWG-24 Litz wire on air core
C_r	$0.5\ \mu\text{F}$	2 Polypropylene capacitors 97F8522 in series

The total resonant circuit Q -value was estimated at 200, and the initial inductor current in this experiment was set at 3 A for 100 V supply voltage. The computer simulation and experimental results are shown in Figure 3.7.



(a) Simulation results of resonant link voltage and current.



(b) Experimental results of resonant link voltage and current.

Figure 3.7: Simulation and experimental results of an unloaded resonant link circuit test.

The zero crossing interval (about 4 μsec) is to provide the time for establishment of the required initial current. With supply voltage 100 V, the peak resonant link voltage V_d was measured at 202 V, and the peak resonant current I_r was measured at 8 A.

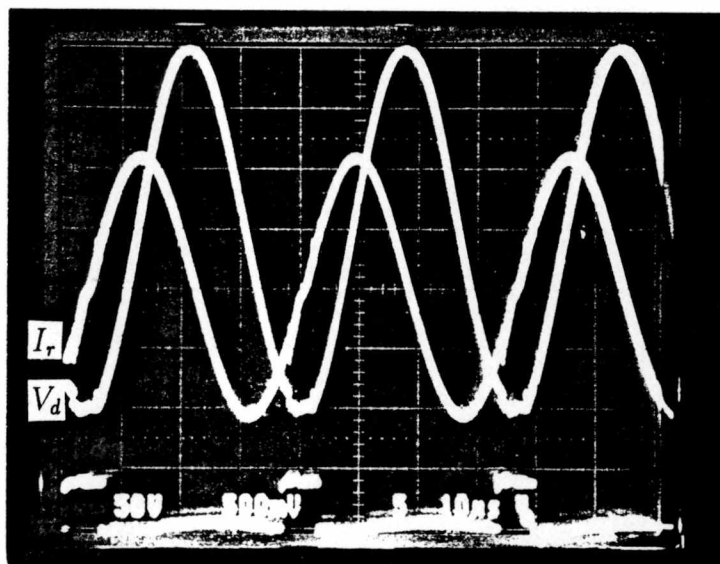
The experimental results agree with the simulation results with the exception that current and voltage spikes occur when the shunt switch S_r turns off. This phenomenon is caused by the current transfer between the shunt switch and the resonant capacitor. The data sheet [54] shows that the power *MOSFET* turn-off fall time is 180 nsec which gives a current transfer rate of 16.7 A/ μsec for 3 A initial current. The non-zero length wire between the resonant capacitor and the shunt switch thus causes the spikes. This phenomenon has been improved by using *IGBT* as the shunt switch because the current transfer rate is lower. Using fast switching type *IGBT* as the shunt switch, the turn-off fall time is 800 nsec, and the current transfer rate is 0.56 A/ μsec for even higher initial current as shown in Figure 3.8. Table 3.2 shows *IGBT* and ultra-fast recovery diode specifications.

Table 3.2: *IGBT* and ultra-fast recovery diode specifications.

Device	Specification	Note
S_r	600 V, 40 A	IXYS <i>IGBT</i> IXGH40N60A
D_r	600 V, 30 A	Motorola ultra-fast recovery diode MUR3060PT

The experimental results using *IGBT* as the shunt switch are shown in Figure 3.8 where the lower trace shows the gate signal. The circuit components remain the same as those in Table 3.1 except the shunt switch-pair is replaced by an *IGBT* IXGH40N60A [52] and an ultra-fast recovery diode MUR3060PT. With the supply voltage $V_s = 150$ V the peak resonant voltage V_d is measured at 300 V. It should be noted that the spike caused by the current transfer vanishes after using *IGBT* as the switching device. If fast current transfer is required for a higher frequency

system, circuit layout will be the key factor in avoiding suspicious spikes.



I_r : 5 A/div
 V_d : 50 V/div

Figure 3.8: Experimental results of unloaded resonant link circuit using *IGBT* as the shunt switch.

Chapter 4

Current Initialization Resonant Link Inverter System

The circuit in Figure 3.1 has been tested successfully for positive initial current establishment. However, when the inverter input current changes from positive to negative, the circuit would not be able to provide negative initial current. In *ac* machine drives, the inverter input current varies between positive and negative, so does the resonant inductor current. Because the original resonant link circuit is not satisfactory for bidirectional initial current control, a new circuit is proposed here to solve such problem.

To verify the new proposed current initialization circuit, a complete system has been simulated using *PC-SIMNON*. With resonant link frequency 50 KHz and 25 KHz, the system has been simulated for a 3-phase, 220 V, 5 HP induction motor. Two 3 KW prototype systems have been fabricated at 25 KHz and 60 KHz respectively. Both of simulation and experiment show satisfactory results.

4.1 A Bidirectional Current Initialization Resonant Link Inverter System

Figure 4.1 shows the resonant inverter with a bidirectional current initialization circuit. The shunt switch S_r is retained for redundancy. Two additional switches S_A and S_B are added to the original resonant link circuit. Switch S_A also serves as start-up function.

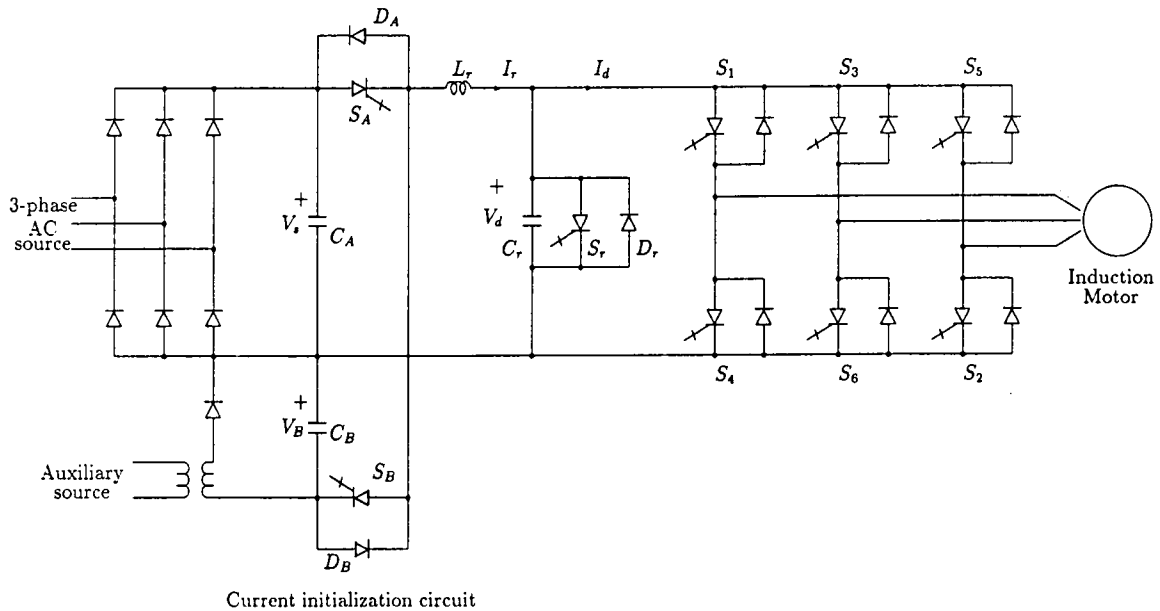


Figure 4.1: Proposed resonant link inverter with bidirectional current initialization circuit.

Normally, switch S_A remains closed for regular operation. For positive initial current establishment, simply turn on switch S_r , and the inductor current will increase with a slope V_s/L_r . For decreasing the inductor current, S_A is turned off and

S_B and S_r are turned on so that the current decrementation slope is given by V_B/L_r , where V_B is the auxiliary *dc* voltage established by a half-wave rectifier. In normal motoring condition of the system, switches S_A and S_B operate at a small fraction of the resonant frequency and contribute some additional loss, but the overall efficiency improvement of the resonant inverter is retained.

4.2 Operation Modes of Current Initialization Circuit

Basic operation of a resonant link circuit is function of *LC* resonance and switch modulation. Figure 4.2 shows the bidirectional current initialization resonant link circuit operation at different modes.

The operation can be summarized into three modes as below.

- Mode 1: This is a regular resonant cycle when S_A is on, but S_r and S_B are off. The equivalent circuit in this mode has been analyzed in Chapter 3. The circuit resonates at the designed frequency and voltage and current magnitudes.
- Mode 2: In this mode, the current I_d increases as a result of inverter switching so that I'_m has to be incremented to match with the incoming I_d . Therefore, S_A and S_r remain on but S_B remains off during the zero crossing interval. The inductor current I'_m at the leading edge increases to I_m at the trailing edge where the current increases with the slope V_s/L_r . With t_a and t_b as the time at the leading edge and the trailing edge respectively, the operation can be expressed as

$$I_r(t_b) - I_r(t_a) \approx \frac{V_s}{L_r}(t_b - t_a) \quad (4.1)$$

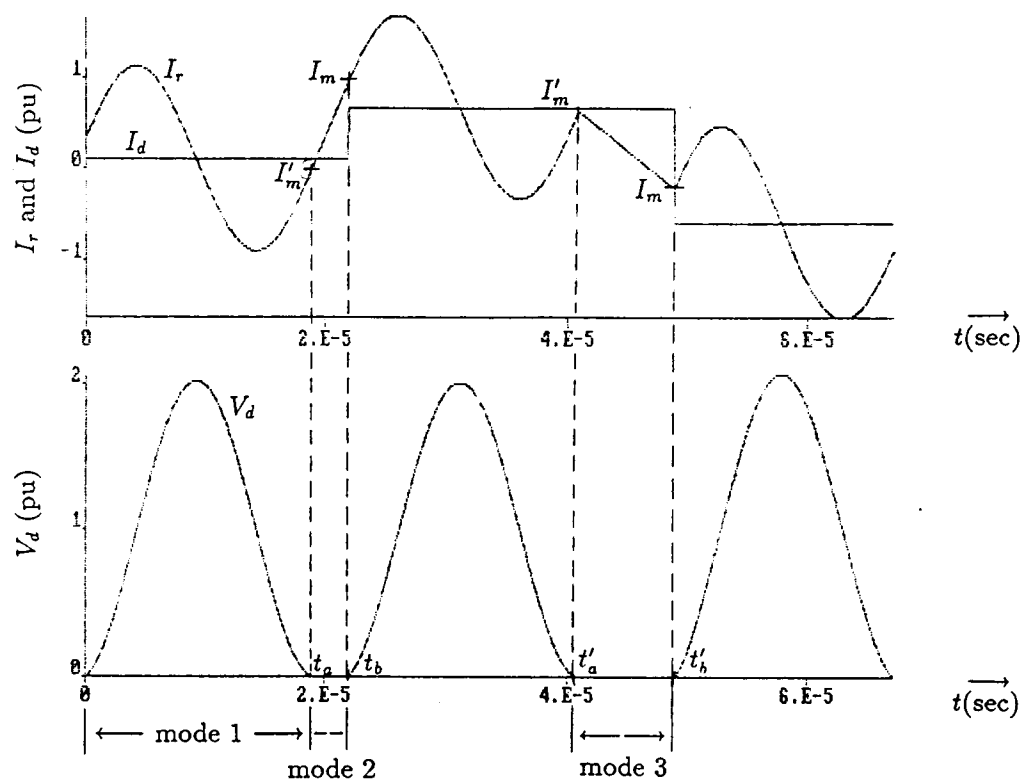


Figure 4.2: Different operation modes of the resonant link circuit.

where $I_r(t_b) = I_m$ and $I_r(t_a) = I'_m$. The control circuit should detect I_r and return to mode 1 when the desired current is reached.

- Mode 3: In this mode, the current I_d transitions to a lower values (which can be positive or negative), i.e., the current I_d is less than I'_m such that I'_m requires decrementation ($I_m < I'_m$). By turning off S_A and turning on S_B and S_r during zero crossing interval, the inductor current will decrease at the slope V_B/L_r . With time t'_a at leading edge and t'_b at the trailing edge, the current decrementation can be given as

$$I_r(t'_b) - I_r(t'_a) = I_m - I'_m \approx \frac{V_B}{L_r}(t'_b - t'_a) \quad (4.2)$$

where V_B can be assumed as constant.

4.3 Prediction of the Initial Current

The initial current (I_m) is to be known at the leading edge of zero crossing interval so that the appropriate switching action of the initialization circuit can be taken. Since the inverter is being switched at the trailing edge as shown in Figure 4.2, the incoming current I_d is to be predicted at the leading edge to determine the required I_m value. Note that it is possible to switch the inverter at the leading edge, but then the function of S_r can be performed by the inverter switches. In fact, the simulation of inverter system which is described later does not use switch S_r . However, adding a redundancy to the circuit would increase fault tolerance and system reliability.

The phase currents of the inverter are controlled by hysteresis based bang-bang type current controller. Since the magnitude of command and phase currents is practically invariant during the small zero crossing interval, the switching logic of the inverter is determined at the leading edge. Implementation of this logic to

switch the inverter is withheld until at the end of zero crossing interval. By knowing the switching logic and the inverter phase currents, the current I_d can be uniquely predicted. Table 4.1 summarizes the relation of I_d with respect to the phase currents for different switching patterns of the inverter.

Table 4.1: Inverter input current for different switching patterns.

switching pattern	on-switches	inverter input current
1	S, S, S	$i_a + i_b + i_c = 0$
2	S, S, S	$i_a + i_b = -i_c$
3	S, S, S	$i_a + i_c = -i_b$
4	S, S, S	i_a
5	S, S, S	$i_b + i_c = -i_a$
6	S, S, S	i_b
7	S, S, S	i_c
8	S, S, S	$-(i_a + i_b + i_c) = 0$

Note that the sum of the phase currents, i.e., $i_a + i_b + i_c$ is zero for isolated neutral of the machine. Define the predicted inverter input current as $\hat{I}_d$. The following simple relation can be derived by Boolean manipulation of Table 4.1.

$$\hat{I}_d = Si_a + Si_b + Si_c \quad (4.3)$$

where S , S , and S are the Boolean variables which correspond to 0 or 1 depending on off or on condition of the respective switches. Table 4.2 shows the (4.3) in truth table form.

It is evident from Table 4.2 that the inverter input current can be predicted by an analog multiplexer with input of three-phase currents and output selected by the

Table 4.2: Truth table of inverter input current prediction.

S_1	S_3	S_5	$\hat{I}_d$
0	0	0	0
0	0	1	i_c
0	1	0	i_b
0	1	1	$-i_a$
1	0	0	i_a
1	0	1	$-i_b$
1	1	0	$-i_c$
1	1	1	0

switching logics. A simple equivalent analog multiplexer implementation is shown in Figure 4.3.

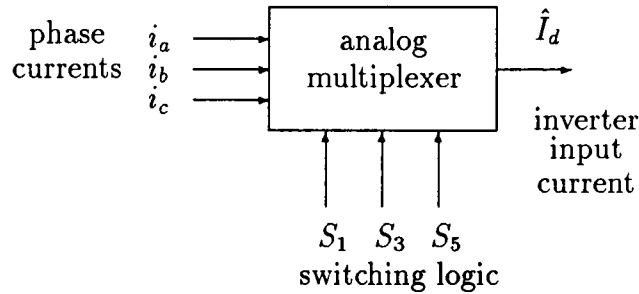


Figure 4.3: Equivalent analog multiplexer for estimation of inverter input current.

The circuit shows the initial current prediction can be simply implemented by an IC chip. Three input analog signals to the multiplexer are phase currents i_a , i_b , and i_c . The predicted inverter input current $\hat{I}_d$ is then function of switching pattern as shown in Table 4.2.

4.4 Simulation of the Proposed Resonant Link Inverter System

A complete inverter-machine speed control system has been modeled and simulated on an *IBM PC-AT* compatible computer *PEP301* using *PC-SIMNON* language. The block diagram of the system is shown in Figure 4.4.

The induction motor control system uses indirect method of vector control [5,7] where the speed control loop generates the torque component of current I_{qs}^* whereas the flux component of current I_{ds}^* is set independently. Both the currents I_{qs}^* and I_{ds}^* are converted into stationary reference frame and then three phase reference currents I_a^* , I_b^* , and I_c^* . The reference currents are compared with the actual phase currents I_a , I_b , and I_c to give the required switching pattern. The hysteresis-band current controller generates the switching logic for the inverter and the current prediction circuits. The predicted current $\hat{I}_d$ is then fed into resonant link control circuit to ensure reliable resonant operation.

Each block of the system (Figure 4.4) is individually modeled in dynamic expression. After verifying all the individual blocks, the whole system is interconnected and simulated for a period of steady state operation.

The simulation program design is discussed in Chapter 6. For complete resonant link inverter system simulation, a 3-phase 5 HP induction motor driven by a resonant link inverter has been studied. Table 4.3 shows parameters of the simulated induction motor, and Table 4.4 shows designed parameters of the resonant link circuit.

The switch in the simulation program is assumed to be ideal which means no switching delay, and no switching losses. Such assumption has been made in the previous no-load condition without misinterpreting the actual system operation ac-

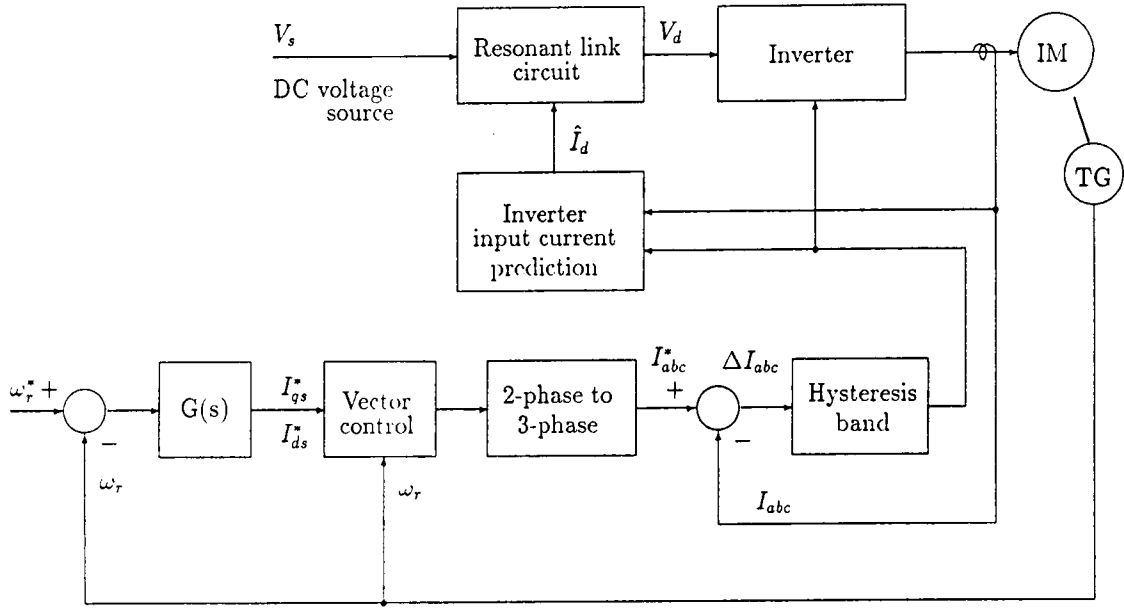


Figure 4.4: Block diagram of resonant link inverter and induction motor speed control system.

Table 4.3: Machine parameters for simulation study.

Induction Machine (3-phase, 208V, 60Hz, 5HP):		
Stator resistance	r_s	0.25Ω
Stator reactance	x_s	0.4Ω
Rotor resistance	r_r	0.817Ω
Rotor reactance	x_r	0.4Ω
Magnetizing reactance	x_m	42.09Ω

Table 4.4: Designed parameters of the resonant link circuit

DC supply voltage	V_s	300V
Maximum inverter phase current	I_L	25A
Characteristic impedance	Z_r	11.89 Ω
Resonant frequency	f_r	50KHz
Resonant inductor	L_r	37.8 μ H
Resonant capacitor	C_r	0.27 μ F
Resonant circuit quality factor	Q	150

According to the results of Chapter 3. For 50KHz resonant frequency, Figure 4.5 shows simulation results of resonant voltage and currents. It can be observed from the resulting waveforms that

- The voltage V_d maintains resonance continuously at a peak value $2V_s$ which indicates that the overshoot and zero crossing problems are solved.
- The initial resonant current I_r was correctly predicted as evident from the plots of I_r , I_d , and $\hat{I}_d$. The only mismatch of the inverter input current I_d and its predicted value $\hat{I}_d$ is at the first resonance cycle. This is understandable because the switching function is unknown initially.
- The inverter input current variation is very minimum and is negligible during the short resonance interval.

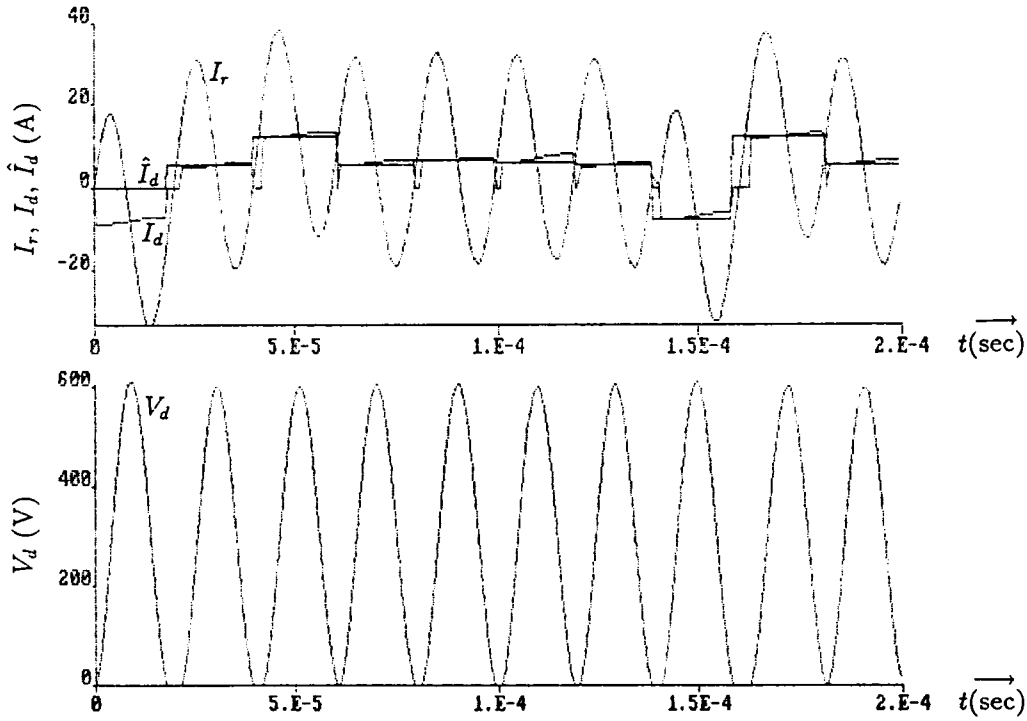


Figure 4.5: Simulation results of inverter input current prediction.

Figure 4.6 shows simulation results of three-phase actual and reference currents. Three phase currents I_a , I_b , and I_c are well tracking with the corresponding reference currents I_a^* , I_b^* , and I_c^* .

The peak-to-peak current ripple for 50 KHz frequency is 2 A. To show superior performance under higher frequency, the same simulation was compared to a 25 KHz frequency inverter. Figure 4.7 shows the simulation results for 25 KHz resonant frequency system where the typical peak-to-peak ripple is about 4 A. Both 50 KHz and 25 KHz resonant link inverter systems are running under the same loading condition, but the 25 KHz system generates twice ripple currents. This tells that the inverter can get better performance with higher resonant frequency.

The algorithm for initial current prediction based on predictive estimation of the

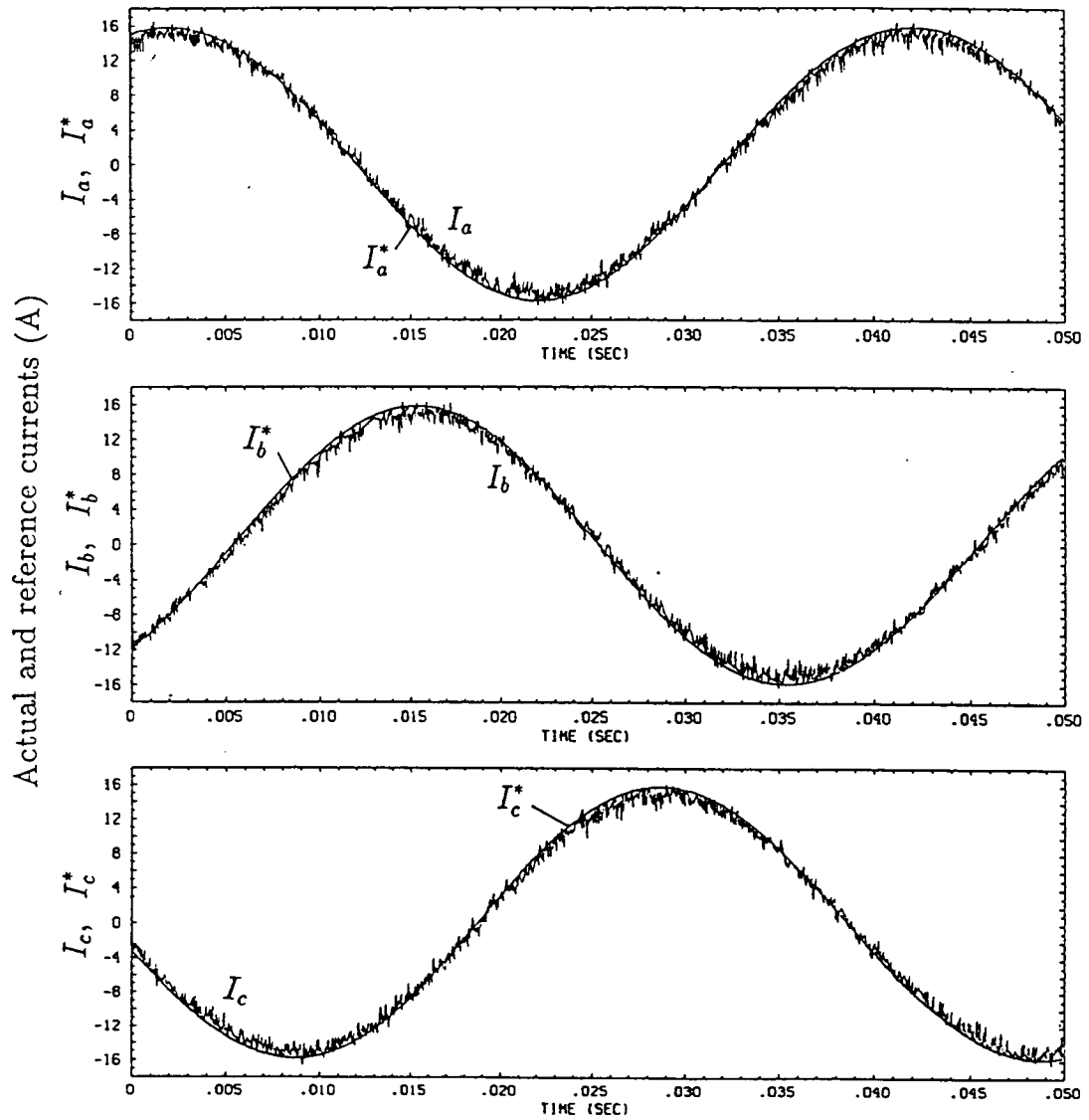


Figure 4.6: Simulation results of the proposed resonant link inverter induction motor drive system (50KHz).

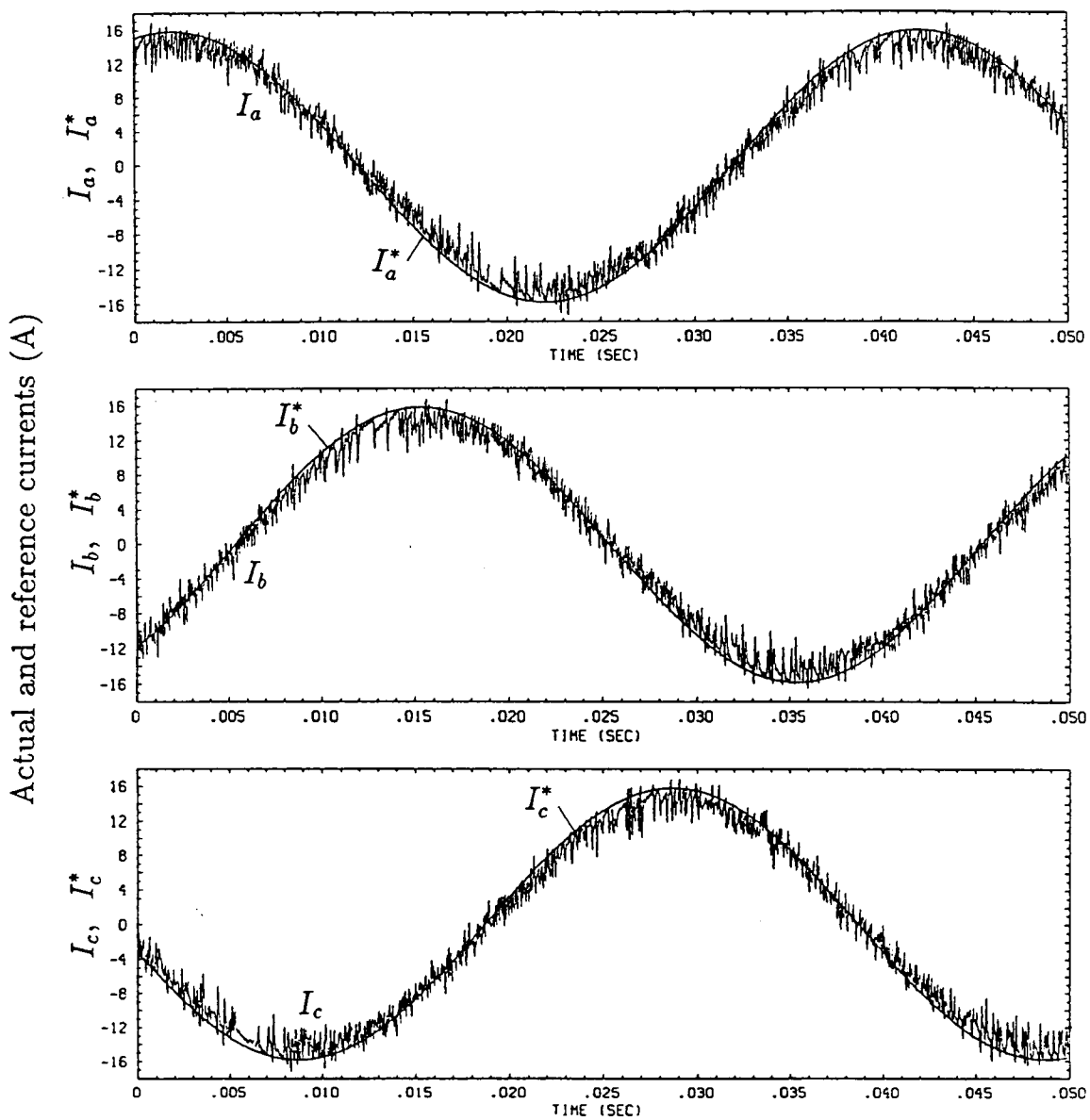


Figure 4.7: Simulation results of the proposed resonant link inverter induction motor system (25KHz).

inverter input current has been shown effective according to the simulation results. The successful operation at both 50 KHz and 25 KHz indicates that the proposed scheme will suit at even higher resonant frequencies provided that high frequency passive and active components are available. This can be proven by the following experimental results.

4.5 Experimental Results of the Resonant *DC* Link Inverter System

To implement the proposed resonant *dc* link inverter system, the component selection and availability are very critical. The optimized designed parameters may not be feasible because of the device limitation. One of the key components is the resonant link capacitor. The experiment in the 25 KHz system used two capacitors in series which inherently increased the resistance in the resonant path and the size of the whole system. The second implemented system thus adopted smaller capacitors in parallel so that the size can be reduced and the switching frequency can be increased to 60 KHz. The experimental setup and test results of two systems at 25 KHz and 60 KHz will be presented in this section.

4.5.1 Experimental Results of a 25 KHz System

The first tested system is a single-phase inverter with *RL* load. This test is to verify the operation of the loading condition and the proposed current initialization technique. Figure 4.8 shows the schematic and parameter values of the tested system. All devices are 600 V, 40 A *IGBT* IXGH40N60A [52].

Using a reference current with 500 Hz, 1 A_{peak} sinusoidal waveform produced by a function generator, the test results are shown in Figure 4.9. The picture shows

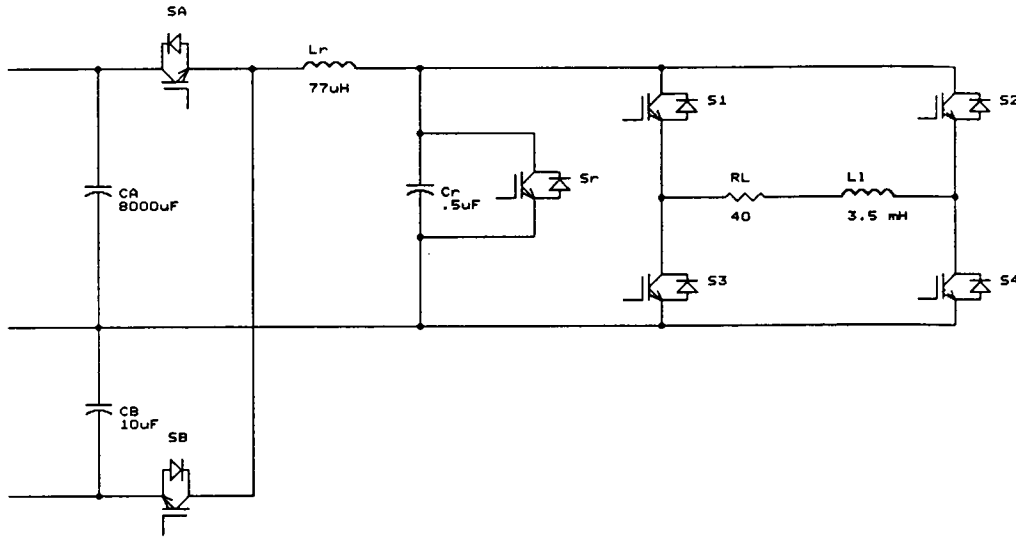
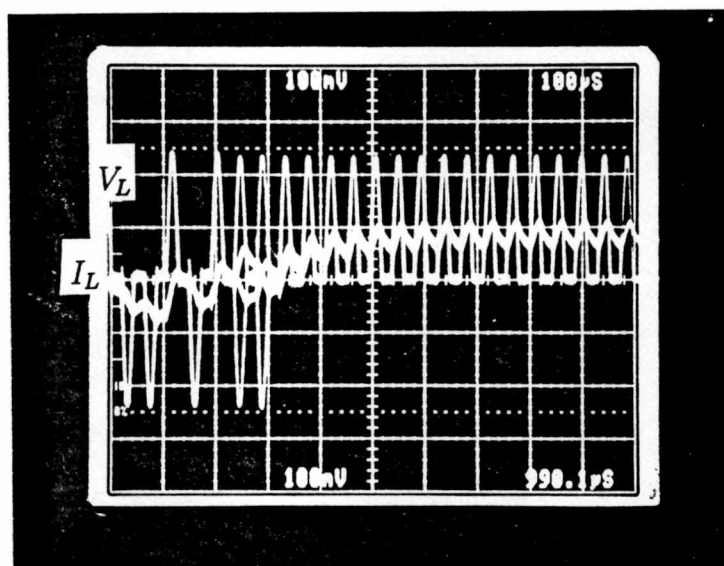


Figure 4.8: Schematic of the tested single phase resonant link system with RL load.

a half-cycle operation, where the peak voltage maintains constant as desired, and the typical current ripple is $.2 A_{peak-to-peak}$. This current ripple can be reduced by increasing the load inductance. In other words, for AC machines with higher leakage inductance, the current ripple will be significantly reduced.

Contrary to traditional dc link inverters which synthesize the load voltage by a square pulse train, the resonant link inverter synthesizes the output by a resonant pulse train with positive and negative alternation. In stead of having the near infinite dV/dt supply voltage in regular dc link PWM inverters, the resonant link inverter output has limited dV/dt . This indicates that the capacitor coupling current on machine windings will be largely reduced, and the machine operation will be more efficient and more reliable.

With $3 A_{peak}$ reference current, the actual and reference currents are shown in Figure 4.10. The reference current is covered inside of actual current envelope, and the typical ripple is $0.6 A_{peak-to-peak}$ as expected. The fundamental output



Load voltage V_L

Load current I_L

Resonant frequency – 25 KHz

Source voltage – $V_s = 60V$

I_L : 1 A/div

V_L : 50 V/div

Figure 4.9: Experimental voltage and current results of 25 KHz resonant link inverter for single-phase RL load.

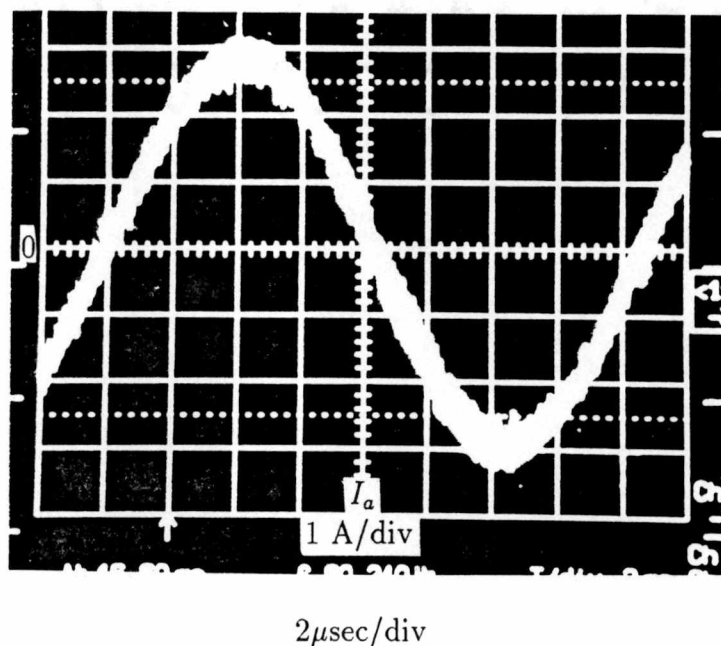


Figure 4.10: Actual and reference current waves of 25 KHz resonant link inverter for a single-phase RL load.

frequency is 60 Hz.

It is evident that the high frequency resonant link inverter can easily synthesize output at the kilo Hertz range. In an experimental practice, the reference input frequency was adjusted from low frequency to 1 KHz without any difficulty. With higher resonant link frequency, the output fundamental frequency can be further increased.

4.5.2 Experimental Results of a 60 KHz System

Because the resonant link system operated successfully at 25 KHz, a higher frequency at 60 KHz was then explored. A $21.2 \mu\text{H}$ resonant inductor was made by Litz wire

on an air core. The resonant capacitor consists of four Orange Drop¹ SBE716P600V capacitors in parallel. Each capacitor is $0.082 \mu\text{F}$ and is rated at 600 V, 8 A_{rms}. Table 4.5 shows specifications for 60KHz resonant circuit components.

Table 4.5: Specification for 60 KHz resonant circuit components

Component	Specification	Note
S_r	600 V, 40 A	IXYS IGBT IXGH40N60A
D_r	600 V, 30 A	Motorola ultra-fast recovery diode MUR3060PT
L_r	21.2 μH	210 AWG-35 stranded Litz wire on air core
C_r	0.328 μF	Four 0.082 μF SBE716P600V in parallel

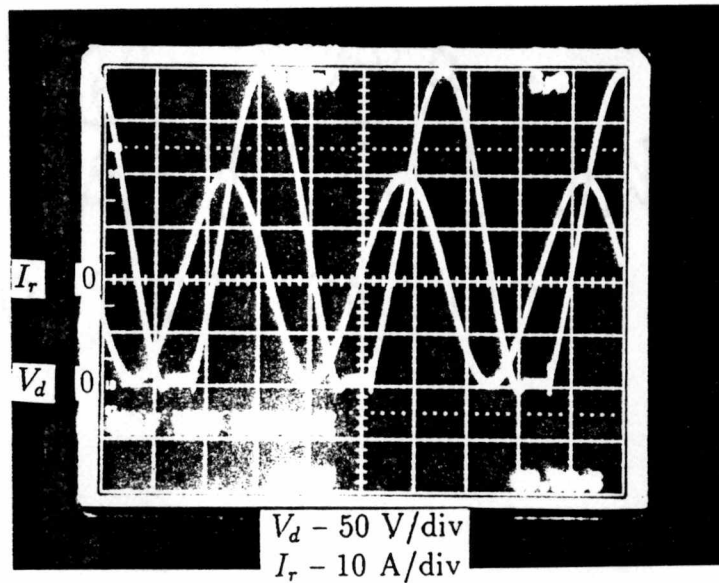
The resonant components are now in much smaller size and are PC board mountable. Under no-load condition, the resonant link voltage and current waveforms are shown in Figure 4.11.

With supply voltage 150 V, the peak resonant voltage is 300 V, and the peak current is 20 A. The actual switching frequency is less than the resonant frequency because of the initial current establishment. It can be seen that under constant initial current control the peak voltage maintains constant as expected.

After successful test under no-load operation, the resonant link circuit was then hooked up to a 3-phase 2-HP induction motor. Under loading condition, the resonant link current and voltage are shown in Figure 4.12. With supply voltage 85 V, peak phase current 4 A, the resonant link voltage V_d maintains constant at 175 V, and the resonant link current I_r varies between negative and positive. It can be seen that the negative peak current drops from -11 A to -15 A occasionally depending on the inverter input current.

Mostly the inverter input current appears positive but drops to negative when

¹Trademark of SB Electronics.



Resonant Link voltage V_d

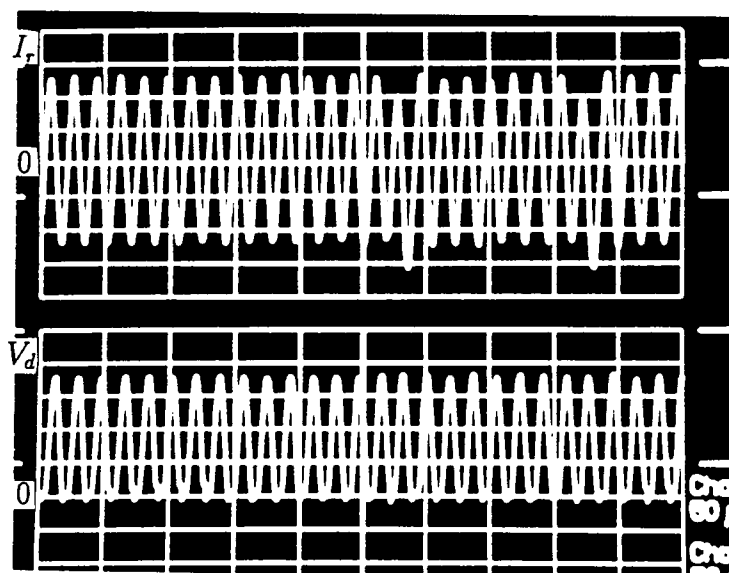
Resonant Link current I_r

Resonant frequency - 60 KHz

Source voltage - $V_s = 150\text{V}$

t: 5 $\mu\text{sec/div}$

Figure 4.11: Experimental results of 60 KHz resonant link voltage and current under no-load condition.



Resonant frequency - 60 KHz

Source voltage - $V_s = 85V$

Upper trace - Resonant Link current I_r (5 A/div)

Lower trace - Resonant Link voltage V_d (50 V/div)

t : 50 μ sec/div

Figure 4.12: Resonant link current and voltage under loading condition.

drawing regenerative current. Although the resonant link current I_r varies with the inverter input current I_d , the peak resonant voltage V_d remains constant. This proves the current initialization concept. However, the zero voltage crossing interval varies with different inverter input current.

In order to increase the loading, the 2-HP induction motor was paralleled with a R - L load. Each R - L branch consists of four parallel $40\ \Omega$ resistors and a $3\ \mu\text{H}$ inductor. The resistive load is adjustable from $10\ \Omega$ to $40\ \Omega$ for each phase. Figure 4.13 shows the experimental setup. For such parallel loading, the induction motor speed can only be controlled by inverter output fundamental frequency, and the resonant link inverter serves as a variable voltage variable frequency (VVVF) power supply.

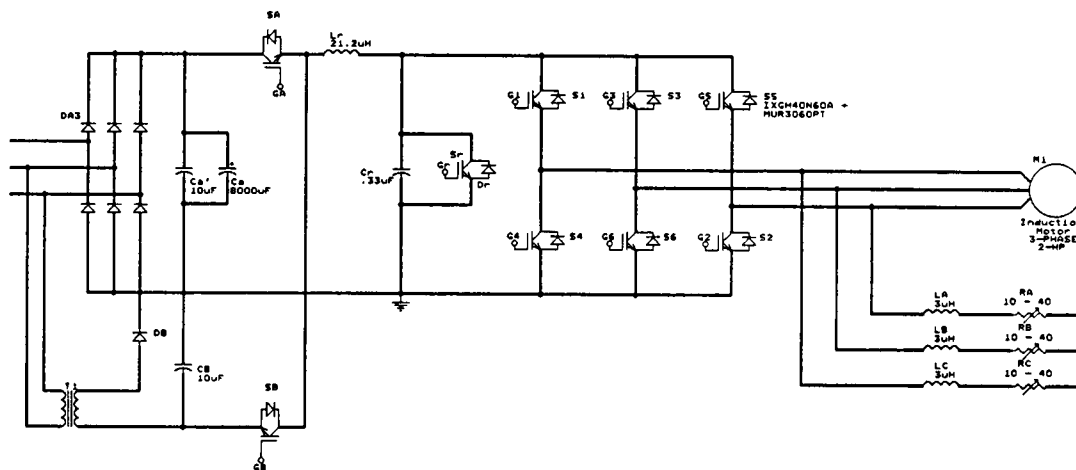
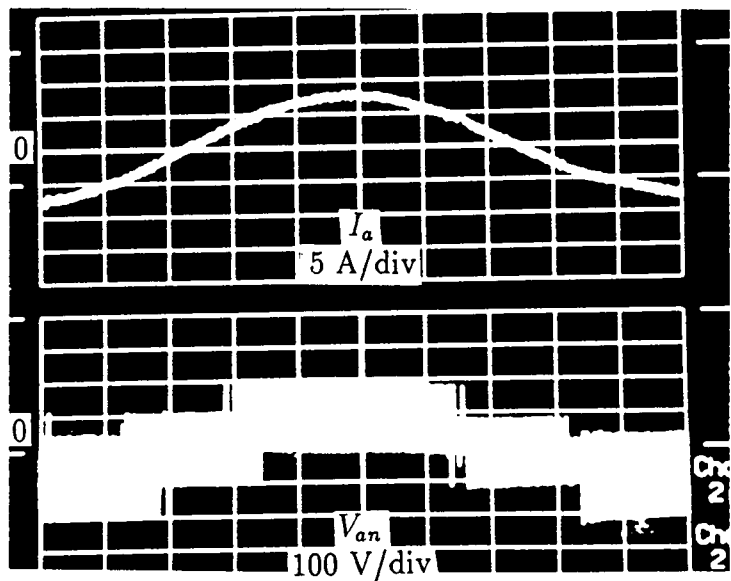
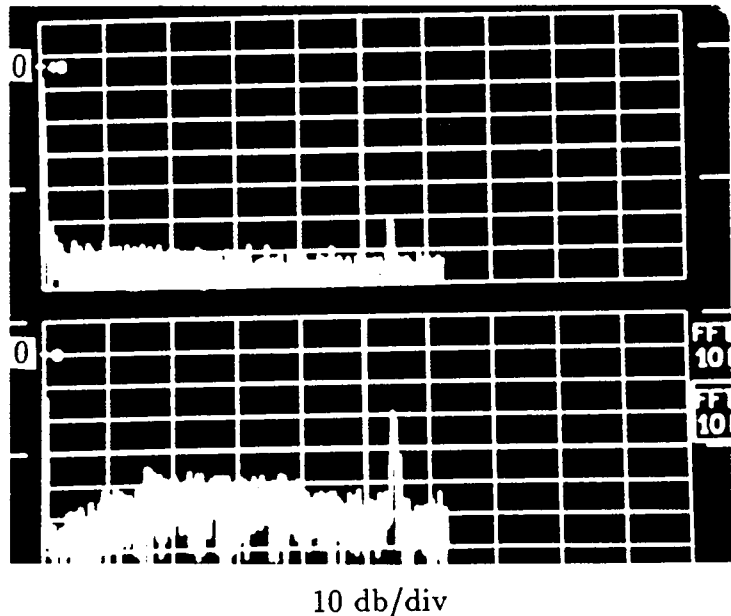


Figure 4.13: Parallel loading of resonant link inverter experimental setup.

With 100 V dc link supply voltage and 50 Hz fundamental reference input frequency, the resulting output phase current and voltage are shown in Figure 4.14(a). The upper trace shows the phase A current which is a clean sinusoidal wave. Figure 4.14(b) shows frequency spectra of the output phase voltage and current. Most harmonic contents are less than $-50\ \text{db}$. Only few components including 55 KHz



(a) phase current and voltage (2 ms/div)



(b) corresponding frequency spectra (10 KHz/div)

Figure 4.14: Experimental results of phase current and voltage for the resonant link inverter.

switching frequency are high at about -40 db. The lower trace shows the phase A output voltage which is synthesized by the resonant pulse train as seen from Figure 4.9. The most significant harmonic content is -20 db at 55 KHz switching frequency. However most objectionable harmonic contents are at -40 db range. It can be seen that the high frequency resonant link inverter requires only small size filter if the output is to serve as a power supply.

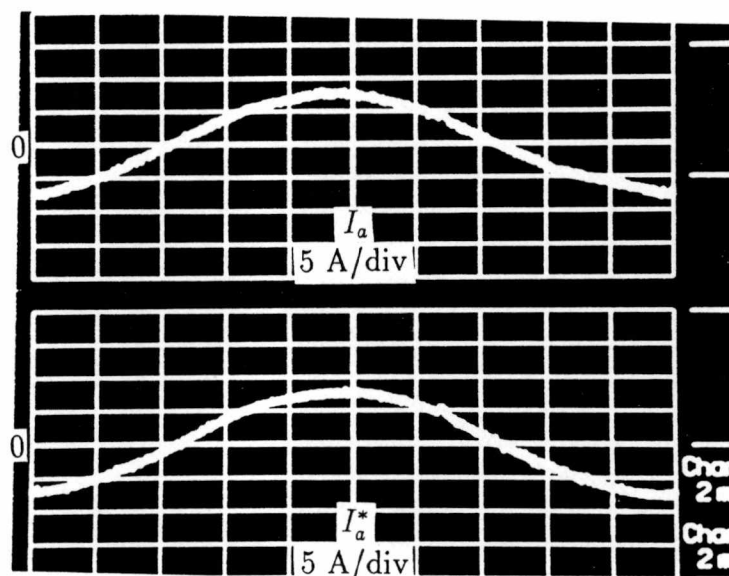
Figure 4.15 compares the actual and reference currents. Although the reference current is somewhat corrupted by circuit noises, the output phase current remains a clean sinusoidal wave. With higher than 50 KHz switching, the output current ripple is smoothed out. The frequency spectrum of the actual current shows superior to that of the reference current.

All of the frequency spectra were analyzed by the built-in *FFT* function of a digital storage oscilloscope Lecroy 9400. Each wave was evaluated on 2500 points base.

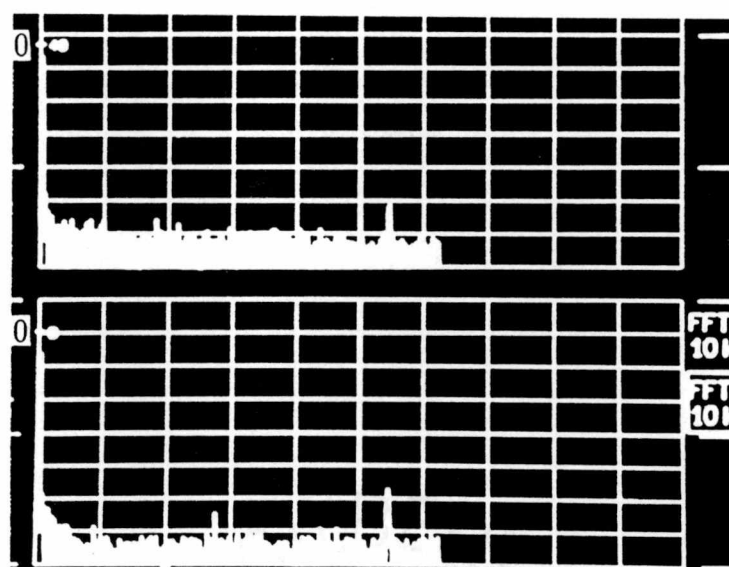
4.5.3 Experimental Results of the Inverter Input Current Prediction

With an analog multiplexer, the inverter input current can be easily predicted as described in Section 4.3. For induction motor load, the experimental results of the actual inverter input current and the predicted current are shown in Figure 4.16. The experiment employed a high speed *CMOS* chip 74HC4051 [61] to predict the initial current and a Tektronix A6302 current probe to measure the actual current.

The upper trace shows the actual inverter input current. The current drops and tends to oscillate during zero voltage crossing. This is because the switching device tends to turn off when the drain-source voltage drops to zero. However, the predicted current (as shown in lower trace) only reflects load side currents and



(a) actual and reference currents (2 ms/div)



10 dB/div

(b) corresponding frequency spectra (10 KHz/div)

Figure 4.15: Comparison of the actual and reference phase currents.

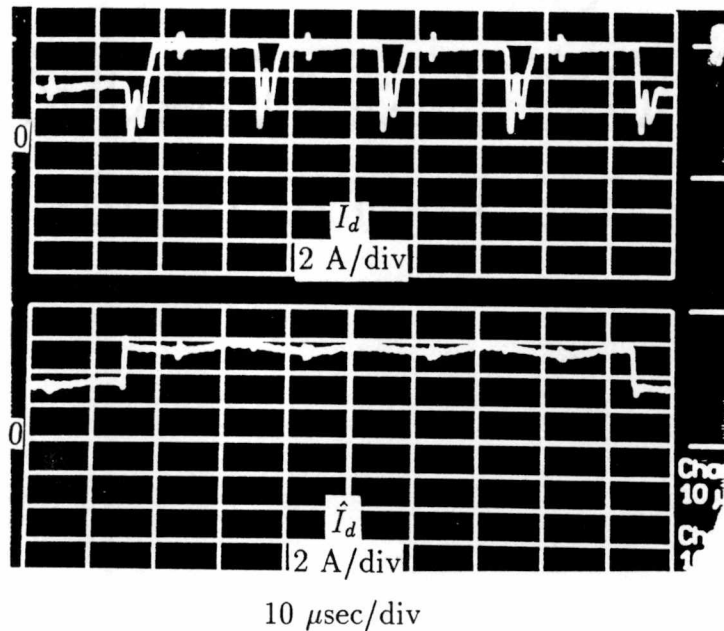


Figure 4.16: Comparison of the actual and predicted inverter input currents.

correctly predicts the inverter input current at the next cycle without any oscillation.

The experiments for single-phase and three-phase inverter systems have shown success of the proposed current initialization algorithm and the new circuit topology. A limitation of raising resonant link frequency is mainly due to circuit components but not the developed algorithm.

4.5.4 Experimental Results of Efficiency Evaluation

The efficiency of resonant link inverter system has been evaluated to show the superiority of the proposed system. Table 4.6 shows power losses and efficiency evaluation results.

The measurement for no-load condition is to see the power losses on the resonant link circuit including the resonant inductor L_r , capacitors C_a and C_r , the switching

Table 4.6: Efficiency Evaluation

dc link V_s	measurement	V_{rms} (V)	I_{rms} (A)	power (W)	efficiency
85	no-load	33.7	0.70	34.4	0.869
	output	36.1	3.88	408.7	
	input	34.6	5.22	470.2	
100	no-load	39.2	0.78	46.6	0.882
	output	42.5	4.51	519.1	
	input	41.2	5.61	588.6	
132	no-load	50.8	0.93	61.8	0.942
	output	56.9	5.95	964.9	
	input	53.9	7.07	1024.4	

of the shunt switch pair S_r - D_r , and the conduction of the diode bridge D_a and the control switch pair S_a - D_a . After loading, the additional power losses mainly come from the conduction of inverter switches $S_1 \sim S_6$.

All of the analytical results are based on the acquired digitized waveforms. The digitizer is a Lecroy 9400 oscilloscope, and the controller is a Tektronix PEP-301 computer. Although the accuracy may be sacrificed by 8-bit resolution, the results agree with the actual loading condition very well. The actual loading condition can be easily calculated by output *RMS* current and the load impedance.

Chapter 5

Quasi-Resonant DC Voltage Notching Inverter System

As described in Chapter 2, the proposed new quasi-resonant *dc* voltage notching inverter is ideal for next generation *ac* machine drives. This scheme can be seen as an extension of the parallel type resonant *dc* link inverter by adding one control switch pair S_c and D_c as shown in Figure 2.16. The basic operation principle has been described in Figure 2.17. This chapter will present the circuit modeling, analysis, simulation, and experimental results of the quasi-resonant *dc* voltage notching inverter.

5.1 Operation Modes of Quasi-Resonant Voltage Link Circuit

Consider the quasi-resonant circuit as shown in Figure 5.1(a). The circuit contains resonant link *LC* components L_r and $\mathcal{L}_r^{\mathcal{C}}$, quasi-resonant control switch pair S_c and D_c , and initial current control switch S_r . Under no load condition, a resonant cycle

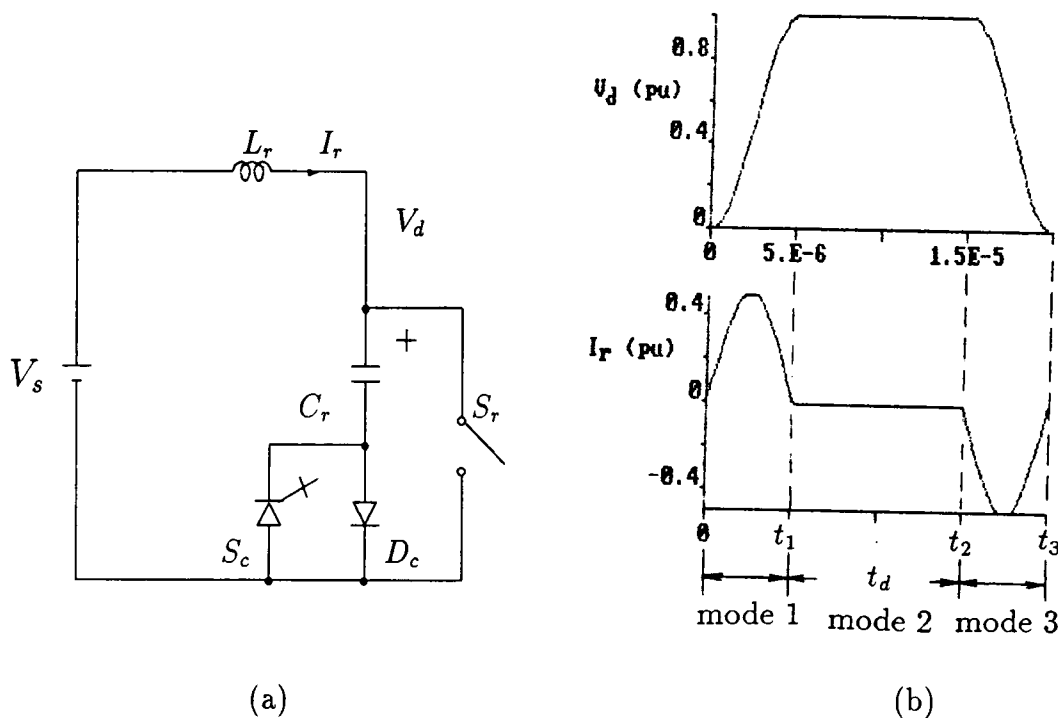


Figure 5.1: Quasi-resonant dc voltage notching circuit operation modes and waveforms.

can be divided by three operation modes as shown in Figure 5.1(b). In comparison with ordinary resonant link circuit operation, this circuit adds a delay mode to maintain constant link voltage.

Assume the switching devices are ideal. The analysis can be simplified as below.

1. Mode 1: The capacitor voltage V_c swings from zero to $2V_s$ in half of the resonant cycle due to LC resonance. Neglecting resistivity of passive components, $V_c(t)$ can be approximated by

$$V_c(t) = V_s(1 - \cos \omega_r t) \quad (0 < t < t_1) \quad (5.1)$$

The resonant frequency can be calculated by

$$\omega_r = \sqrt{\frac{1}{L_r C_r}} \text{ rad/sec} \quad (5.2)$$

The time interval of mode 1 can then be solved by

$$t_1 = \frac{\pi}{\omega_r} \text{ sec} \quad (5.3)$$

This gives $V_c(t_1) = 2V_s$.

2. Mode 2: The diode D_c blocks the resonance so that the voltage V_c remains constant, i.e.

$$V_c(t) = 2V_s \quad (t_1 < t < t_2) \quad (5.4)$$

The time interval $t_d = t_2 - t_1$ can be controlled by a time delay circuit. Typical time delay interval of mode 2 is tenths of a microsecond; therefore, the capacitor voltage variation is negligible during such a short interval.

3. Mode 3: The switch S_c turns on, and the voltage V_c swings back to zero at t_3 . The voltage equation is the same as that in mode 1, i.e.

$$V_c(t) = V_s(1 - \cos \omega t) \quad (t_2 < t < t_3) \quad (5.5)$$

The complete quasi-resonant cycle t_3 consists of an LC resonant cycle and a time delay t_d , i. e.

$$t_3 = \frac{2\pi}{\omega_r} + t_d = 2\pi\sqrt{LC} + t_d \quad (5.6)$$

The modulated quasi-resonant frequency can then be calculated by

$$f_r = \frac{1}{2\pi\sqrt{LC} + t_d} \quad (5.7)$$

At the end of the quasi-resonant cycle, $t = t_3$, $V_c(t_3)$ returns to 0.

5.2 Quasi-Resonant DC Link Circuit Analysis

In quasi-resonant *dc* link circuit, the switching devices S_c and D_c are inserted into the resonant path. Thus, the circuit model will be similar to that of the resonant *dc* link system if the control switch pair is ideal. However, the experimental result shows that the switching devices behave somewhat differently from theoretical aspects and the specifications provided by the manufacturer. This section will include an empirical model of the switching device in the quasi-resonant *dc* link circuit analysis.

5.2.1 Modeling of Switching Device Conduction Drop

In order to model the new circuit, the device conduction voltage drop has to be considered. Traditionally, according to the manufacturer's data sheet, the device can be modeled as a constant voltage source in series with a resistor when conducting. This model is effective in most cases, but a true model obtained from experiments shows a significant inductive behavior in switching devices.

Let V_{ds} represent the voltage drop between drain and source of the *IGBT* or between cathode and anode of the diode and I_d represent the drain current. Figure 5.2 shows the experimental results of the conducting voltage drop of the devices. When the current I_d is positive, *IGBT* conducts. On the other hand, the diode conducts when I_d is negative.

It can be seen that the device conduction drop is a function of current, i.e.,

$$V_{ds} = f(I_d). \quad (5.8)$$

This function could be nonlinear. The traditional linear diode model

$$V_{ds} = 0.7 + I_d r_d$$

does not fit the experimental result because the voltage drop is different for the same current magnitude but with different slope. The tendency is that when the

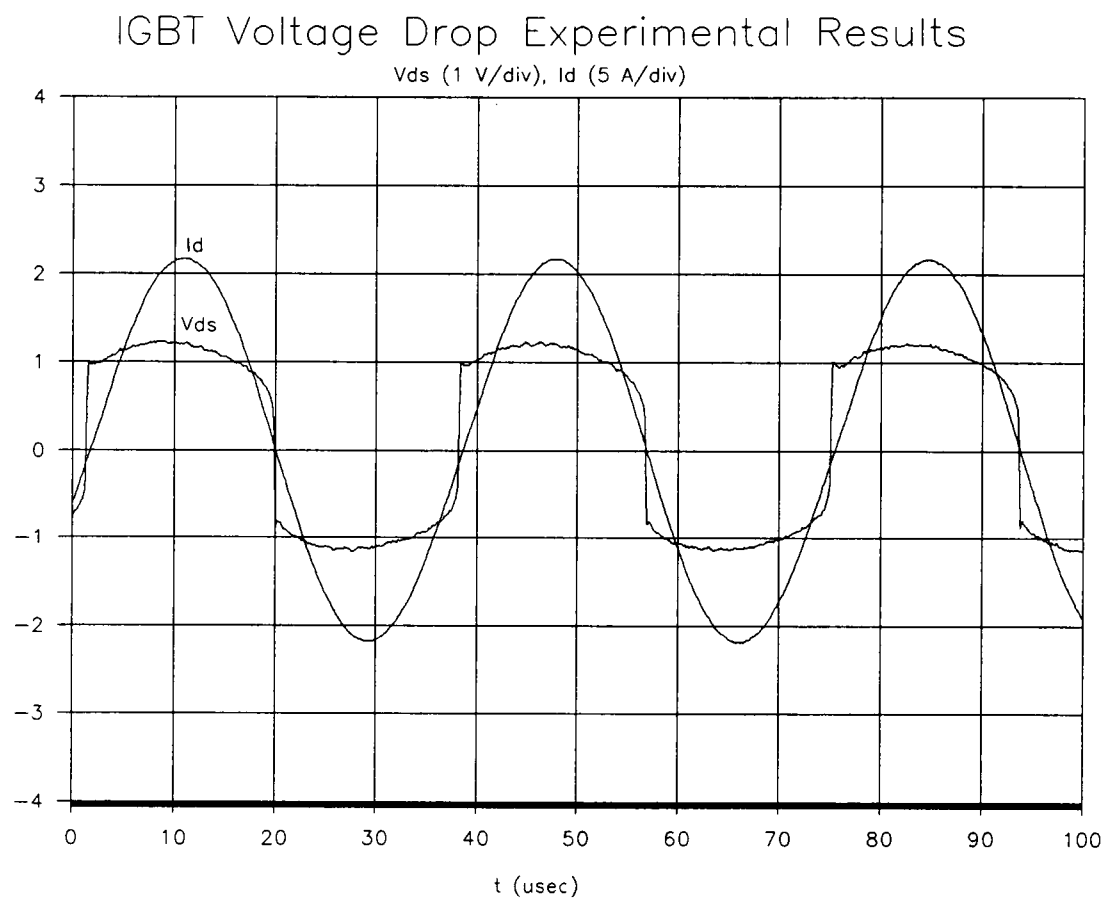


Figure 5.2: Experimental results of *IGBT* and diode conduction drops.

current I_d is increasing, the voltage drop is higher. This implies that the circuit model should include a derivative term of the current I_d . A simple mathematical model is proposed as (5.9).

$$V_{ds} = k_0 + k_1 I_d + k_2 \frac{d I_d}{d t} \quad (5.9)$$

It is not difficult to convert constants k_0 , k_1 , and k_2 into electrical circuit model by relating both sides of (5.9). Similar to the traditional diode model, k_0 is a voltage source, and k_1 is a resistive element. The additional constant k_2 is obtained as inductive element by equating voltage quantities.

5.2.2 Equivalent Quasi-Resonant Link Circuit Analysis

Assume the inverter input current is a constant current source during the resonance interval. Including the above device model, the complete quasi-resonant *dc* link circuit can be modeled as Figure 5.3. The circuit is similar to that of Figure 3.2 except additional components k_0 , k_1 , and k_2 representing a switching device.

The network equations of this circuit can be derived as

$$V_s(t) = I_r(t)(R_r + k_1) + \frac{d I_r(t)}{d t}(L_r + k_2) + V_c(t) + k_0 \quad (5.10)$$

$$I_r(t) = I_d + C_r \frac{d V_c(t)}{d t} \quad (5.11)$$

with initial states $V_c(0) = 0$ and $I_r(0) = I_m$.

Solving the above equations yields

$$V_c(t) = (V_s - I_d R_r' - k_0) \left[1 - \frac{1}{\sqrt{1 - \zeta^2}} e^{-\alpha t} \sin(\omega t + \theta) \right] + \frac{(I_m - I_d) Z_r}{\sqrt{1 - \zeta^2}} e^{-\alpha t} \sin \omega t \quad (5.12)$$

$$I_r(t) = I_d + e^{-\alpha t} \left[(I_m - I_d) \cos \omega t + \frac{2V_s - (I_d + I_m) R_r}{2\omega L_r'} \sin \omega t \right] \quad (5.13)$$

where

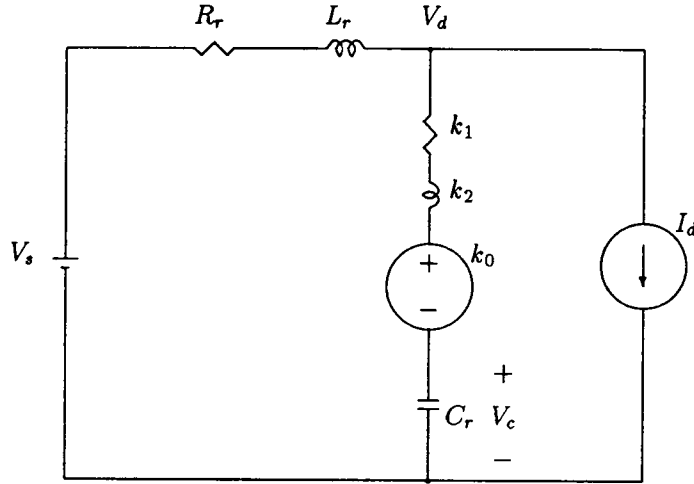


Figure 5.3: Equivalent quasi-resonant link circuit.

$$R'_r = R_r + k_1$$

$$L'_r = L_r + k_2$$

$$\alpha = R'_r / 2L'_r$$

$$\zeta = R'_r / 2Z_r$$

$$\omega_r = 1 / \sqrt{L'_r C_r} = \text{characteristic impedance}$$

$$Z_r = \sqrt{L'_r / C_r} = \text{characteristic impedance}$$

$$\omega = \omega_r \sqrt{1 - \zeta^2} = \text{oscillation frequency}$$

$$\theta = \cos^{-1} \zeta = \tan^{-1} \frac{\sqrt{1 - \zeta^2}}{\zeta}$$

Similar to the analysis in resonant *dc* link circuit, the maximum and minimum capacitor voltages can be solved by equating the derivative of the voltage equation to

zero. Referring to Figure 5.1, the voltage at time t_1 is maximum, and the minimum is at t_3 . The solution of maximum and minimum capacitor voltages are shown in (5.14) and (5.15).

$$V_c(t_1) = (V_s - I_d R_r - k_0) \left[1 - \frac{1}{\sqrt{1 - \zeta^2}} e^{-\alpha t_1} \sin(\omega t_1 + \theta) \right] + \frac{(I_m - I_d) Z_r}{\sqrt{1 - \zeta^2}} e^{-\alpha t_1} \sin \omega t_1 \quad (5.14)$$

where

$$t_1 = \left[\pi + \tan^{-1} \frac{2\sqrt{1 - \zeta^2}(I_d - I_m)Z_r}{2V_s - (I_d + I_m)R'_r} \right] / \omega$$

$$V_c(t_3) = (V_s - I_d R'_r - k_0) \left[1 - \frac{1}{\sqrt{1 - \zeta^2}} e^{-\alpha t_3} \sin(\omega t_3 + \theta) \right] + \frac{(I_m - I_d) Z_r}{\sqrt{1 - \zeta^2}} e^{-\alpha t_3} \sin \omega t_3 \quad (5.15)$$

where

$$t_3 = t_d + \left[2\pi + \tan^{-1} \frac{2\sqrt{1 - \zeta^2}(I_d - I_m)Z_r}{2V_s - (I_d + I_m)R'_r} \right] / \omega$$

where t_d is modulated by the control circuit.

A numerical example is presented here with both analytical and experimental results. Table 5.1 shows circuit parameters of a 60KHz quasi-resonant link system where switch constants k_0 , k_1 , and k_2 are obtained from the experimental results (Figure 5.2).

Using the extreme case that $I_d = -1pu$, the maximum and minimum voltages, $V_c(t_1)$ and $V_d(t_3)$, can be plotted as functions of the initial current as shown in Figure 5.4.

As explained in Chapter 3, in order to ensure zero voltage crossing, the voltage $V_c(t_3)$ should be less than zero, and the initial current should be controlled in a similar manner. For a negative inverter current ($I_d = -1pu$), the initial current $I_m = -0.65pu$ will satisfy the initial current requirement. Similarly, under no-load condition, the initial current $I_m = 0.35pu$ will satisfy the requirement.

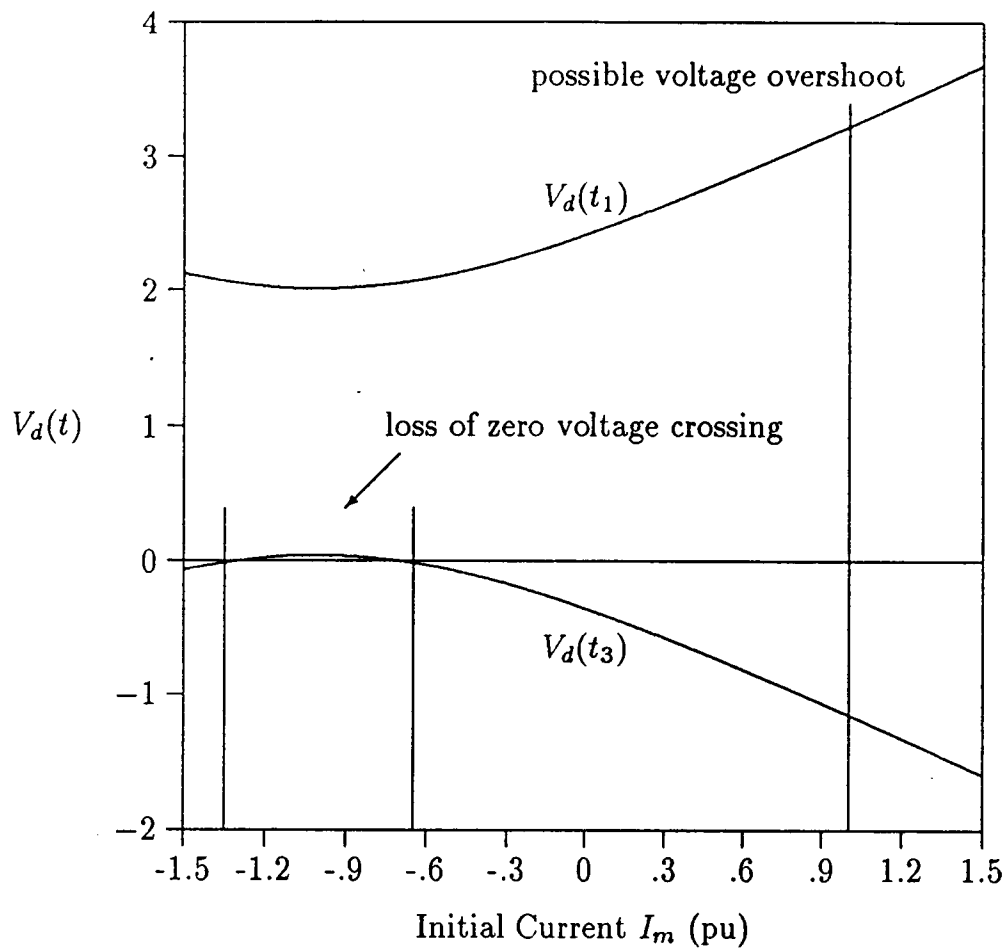


Figure 5.4: Maximum and minimum quasi-resonant link voltages as functions of initial current I_m .

Table 5.1: Circuit parameters of a 60KHz quasi-resonant link system.

Resonant inductor L_r	21.2 μH
Resonant capacitor C_r	0.328 μF
Resonant path resistance R_r	0.020 Ω
Switch equivalent voltage drop k_0	0.8 V
Switch equivalent resistance k_1	0.04 Ω
Switch equivalent inductance k_2	0.8 μH

In practical circuit design, the initial current can be adjusted to a higher value to prevent zero voltage crossing failure. An experimental setup and its results are presented in the following section.

5.2.3 Experimental Results of a Quasi-Resonant Link Circuit Under No-Load Condition

The experimental setup is similar to that of the previous resonant *dc* link circuit. The resonant link circuit parameters are shown in Table 5.1, and two additional devices are

S_c : IXGH40N60A (IXYS).

D_c : MUR3060PT (Motorola).

Figure 5.5 shows quasi-resonant link voltage and current produced by a 60 KHz resonant tank with modulated frequency at 20 KHz. When the diode D_c blocks the resonance, the current stays zero and the link voltage remains constant. The overall operation agrees well with the prediction of the circuit analysis and simulation.

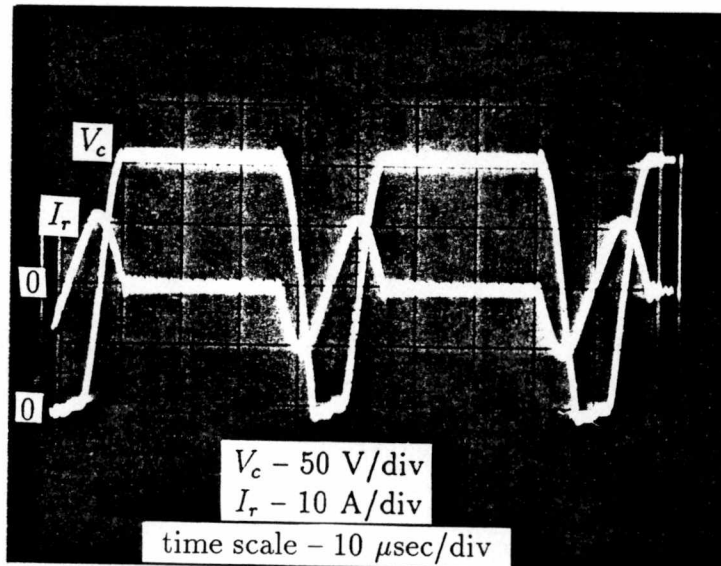


Figure 5.5: Quasi-resonant link circuit experimental results under no-load condition.

The link voltage V_c remains twice the source voltage V_s when the diode is blocked and smoothly drops to zero when switch S_c turns on. This allows the inverter switching under zero voltage crossing. Because the switching frequency is controllable by modulating S_c , the *PWM* resolution can be high, and the performance can be optimized as described in following sections.

5.3 Current Initialization

To provide bidirectional initial current control, the quasi-resonant link inverter should include a current initialization circuit as does the resonant *dc* link inverter. Figure 5.6 shows the proposed *PWM* inverter for an *ac* motor drive using the quasi-resonant *dc* voltage notching scheme with current initialization circuit.

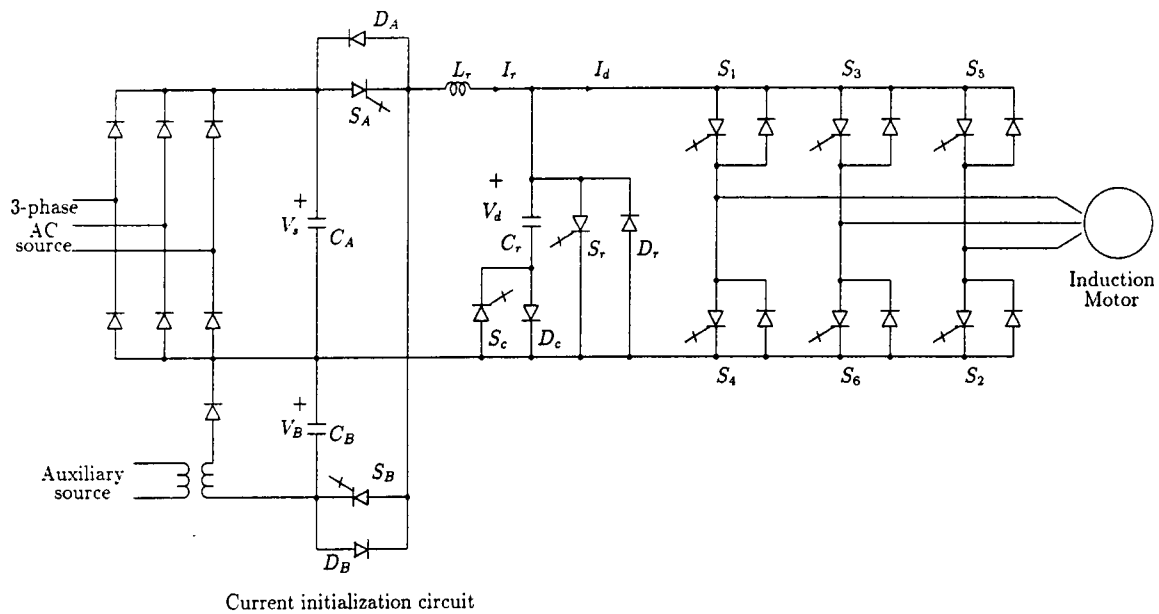


Figure 5.6: Quasi-resonant *dc* voltage notching scheme with current initialization circuit.

The *MCT* devices are shown as preferred switching elements although *IGBTs* or *BJTs* can be used. The front-end diode bridge rectifier generates the DC supply voltage V_s . This voltage is applied to the current initialization circuit, L_r - C_r resonant circuit, and the voltage notching control circuit before application to the input of the three-phase PWM inverter. The resonant circuit, in conjunction with the voltage notching control circuit, provides the quasi-resonant voltage notching scheme.

Consider the proposed circuit as shown in Figure 5.6. The inverter input current

I_d can be positive (active mode) or negative (regenerative mode). When the transition occurs from fully active mode to fully regenerative mode, the resonant voltage V_c can be as high as three times the supply voltage V_s as shown in Figure 5.4. In order to avoid this voltage overshoot, the inductor initial current has to track the inverter input current I_d in both positive and negative directions [21,8].

The proposed bidirectional current initialization circuit is controlled by switches S_r , S_A , and S_B as shown in Figure 5.6. If the positive initial current is required, S_r and S_A on, the inductor current I_r increases at positive rate V_s/L_r . However, if the negative initial current is required, S_r and S_B on, I_r decreases at negative rate $-V_B/L_r$. The criterion of establishing the required initial current is based on the previous analytical results shown in Figures 5.4.

In order to minimize the voltage overshoot and to ensure zero voltage crossing, as indicated in Figures 3.5 and 3.6, the typical zone of I_m variation shows that the initial current I_m is always greater than the inverter input current I_d by 0.35 p.u.

5.4 Optimized Control of Switching Frequency

Because the time interval of mode 2 is controllable, the switching frequency can be adjustable. If a fixed switching frequency is required, the time interval can be simply controlled by employing a fix delay circuit.

In an *ac* motor drive, this switching frequency can be further controlled by minimizing the phase current ripple. Consider the block diagram of the inverter-fed induction motor drive control system shown in Figure 5.7. The complete system looks similar to Figure 4.4 except the resonant link inverter is replaced by a quasi-resonant notch link inverter. The control of the additional quasi-resonant control switch is not shown in the block diagram. With this control switch, however, the *PWM* optimization can be possible. One of feasible optimized *PWM* methods is

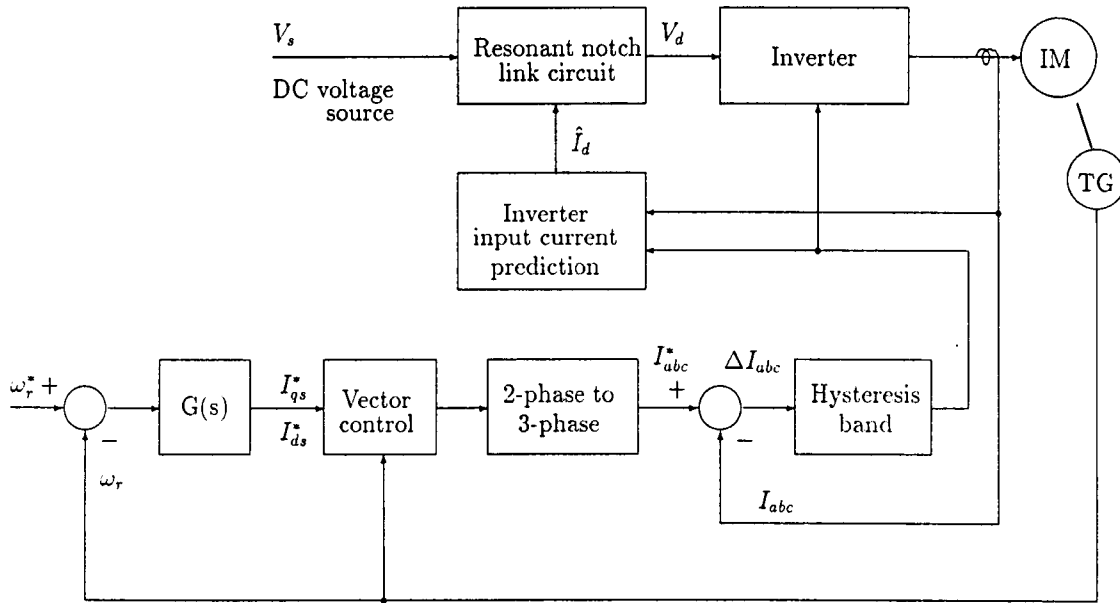


Figure 5.7: Block diagram of quasi-resonant voltage notch link inverter induction motor drive.

described below.

Minimum Phase Current Error Algorithm The current error ΔI_{abc} containing three components ΔI_a , ΔI_b , and ΔI_c indicates phase current ripple. The current ripple, causing additional machine heating must be minimized. To minimize phase current ripple, an optimized control algorithm of switching frequency has been developed by regulating the time interval t_d . The algorithm is described below.

Define the performance index as the sum of current error square σ .

$$\sigma = \Sigma[(I_a - I_a^*)^2 + (I_b - I_b^*)^2 + (I_c - I_c^*)^2] \quad (5.16)$$

Selecting an error criterion σ^* , the control of voltage notch is based on the comparison result of σ^* and actual σ . The diode D_c blocks the resonance, and the switch S_c resumes the resonance whenever $\sigma > \sigma^*$.

Figure 5.8 shows the control block diagram of voltage notching and current initialization circuits. The resonant frequency is designed at 100 KHz, but the inverter switching frequency is regulated by the error criterion σ^* . The complete inverter-fed induction motor drive control system has been formulated and simulated on a computer using *SIMNON*.

5.5 Simulation of Quasi-Resonant DC Link Inverter System

Figure 5.9 shows simulation results of the quasi-resonant notch link voltage and currents where the error criterion is selected as $\sigma^* = 16$. This implies that the maximum absolute phase current error is less than 4 A, or the maximum ripple current is 4 A. The *dc* link voltage V_c periodically creates a notch, and the notch frequency is regulated by controlling phase current error criterion.

The overall switching frequency in this case is about 50 KHz. To show the superiority of this scheme, the simulation result of the current ripple has been compared to that of the ordinary 50KHz resonant link inverter system (Figure 4.6). Figure 5.10 shows simulation results of three-phase actual and reference currents obtained from the proposed resonant-notch link system.

To quantize the performance of the proposed system, the sum of current error square over a fundamental cycle with 10 μ sec sampling time is compared to that

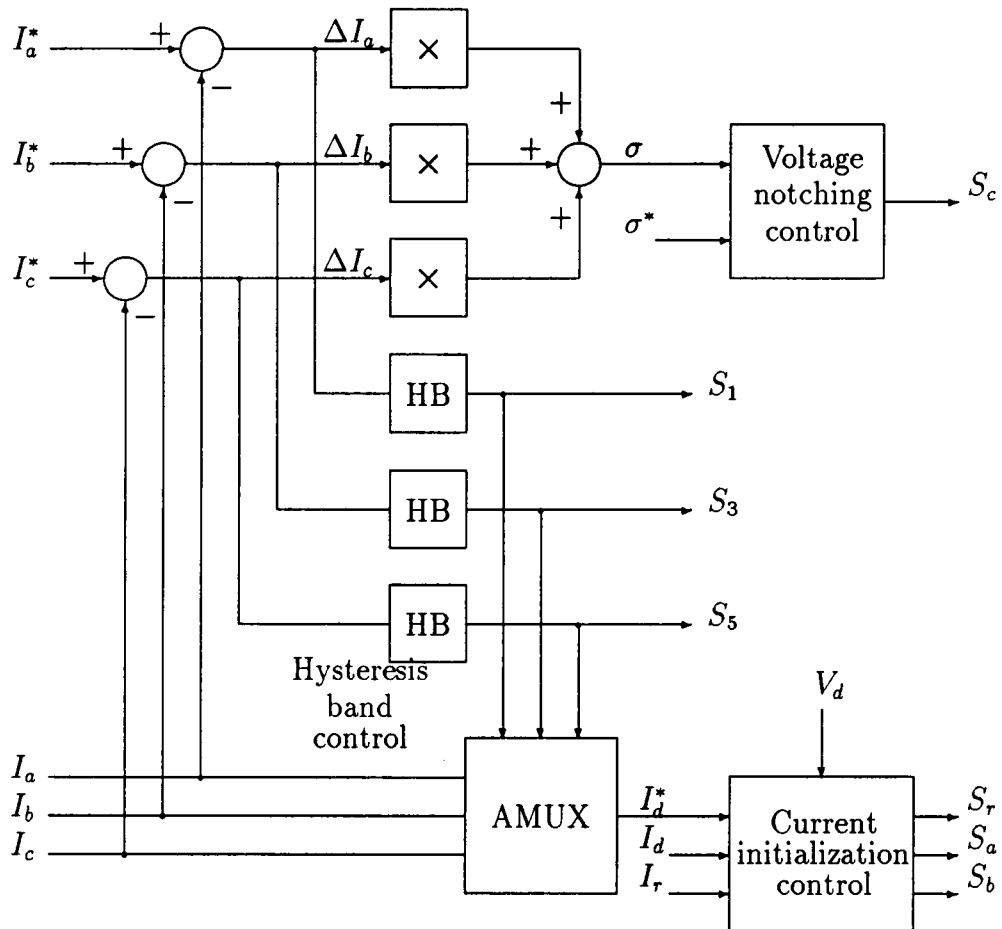


Figure 5.8: Control block diagram of voltage notching and current initialization circuit.

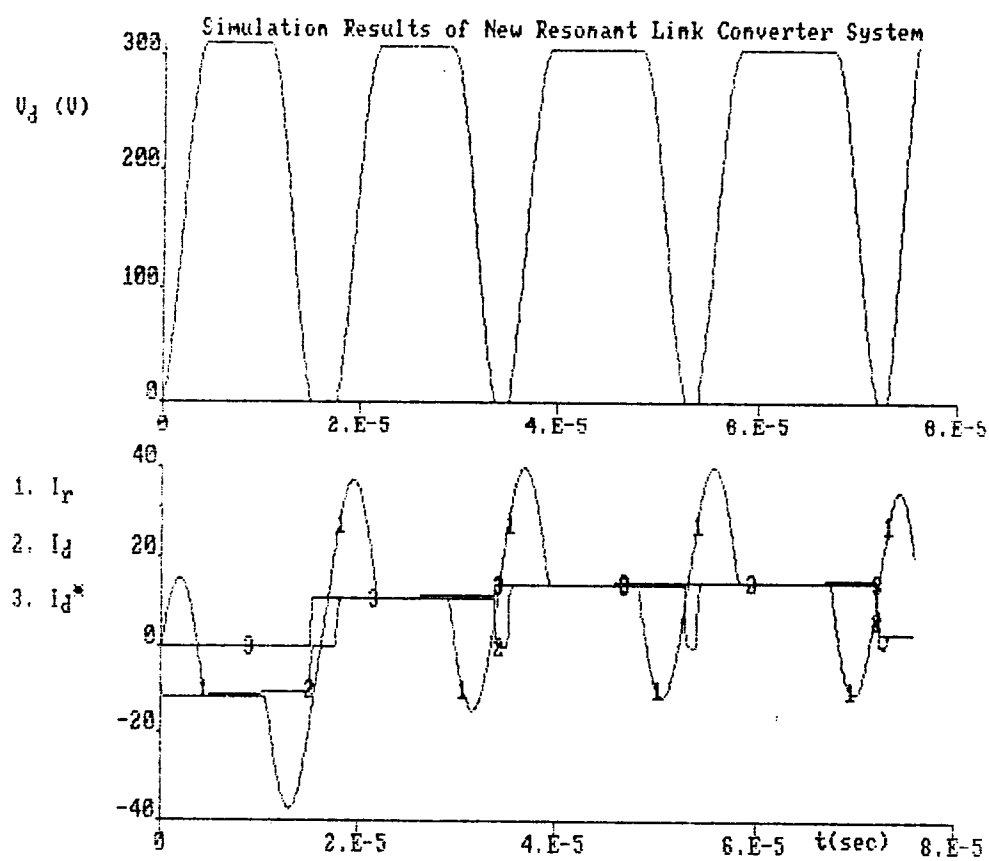


Figure 5.9: Simulation results of the quasi-resonant *dc* notch link voltage and current waves.

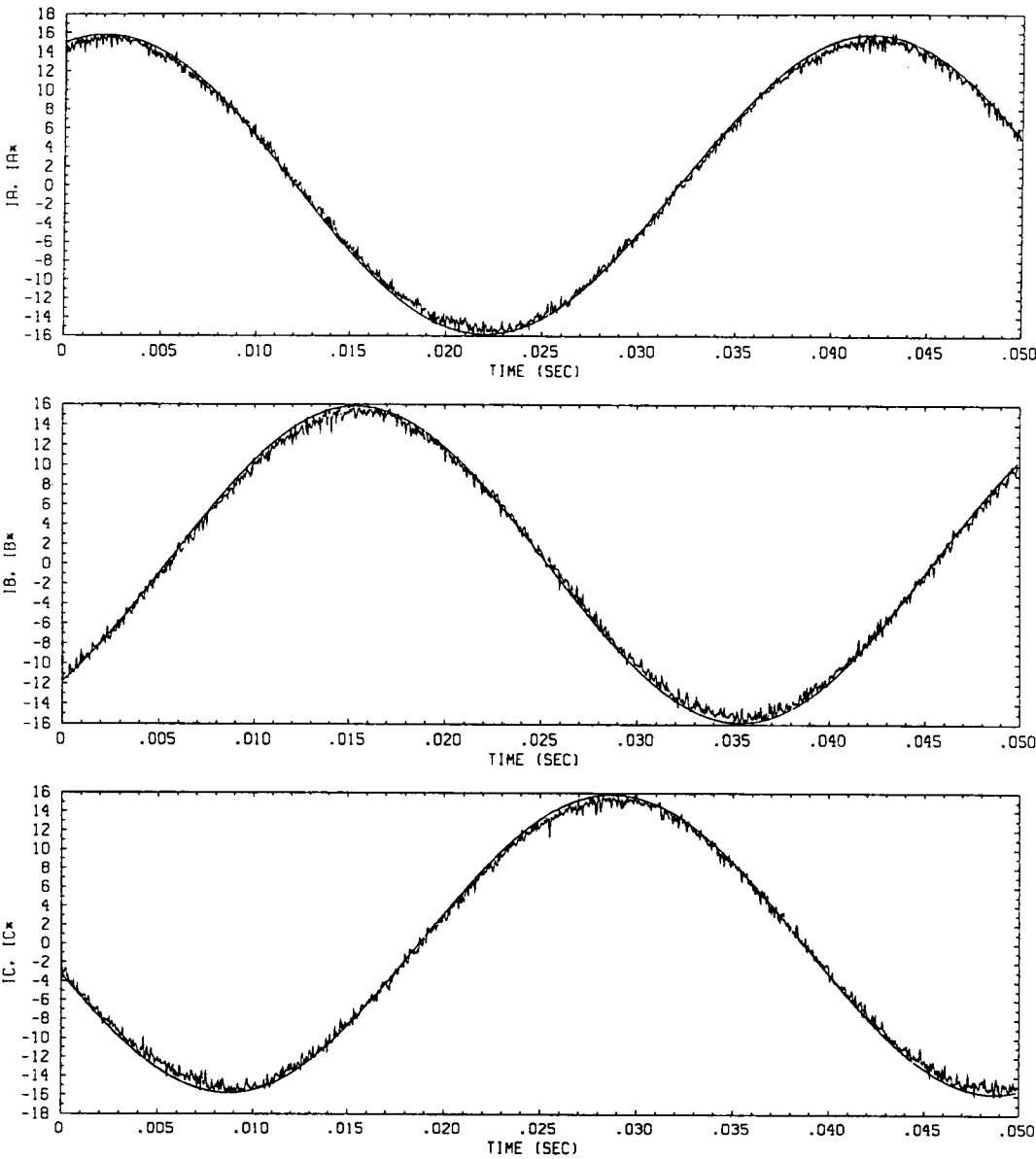


Figure 5.10: Simulation result of the quasi-resonant *dc* notch link inverter current waves.

of the ordinary resonant link system as shown in Table 5.2. The sample data are obtained from simulation results of both resonant link and quasi-resonant link systems.

With higher resonant frequency but the same level of switching frequency, the proposed quasi-resonant notch link converter system significantly reduces phase current ripples.

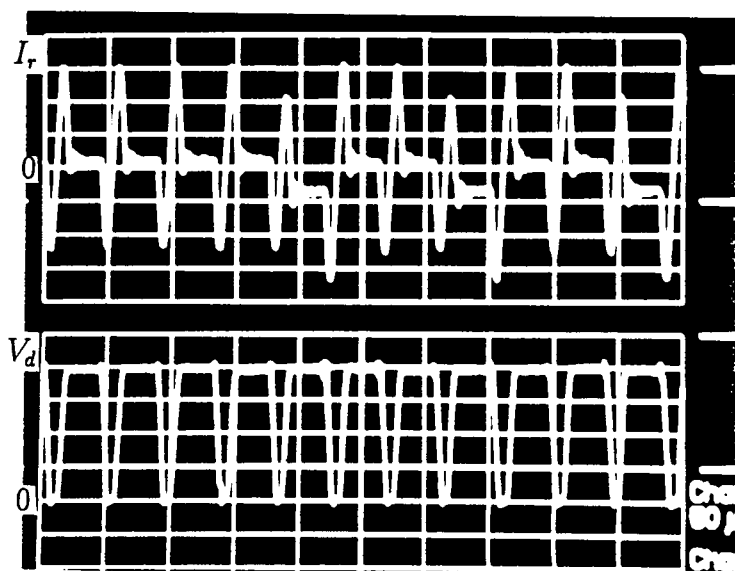
Table 5.2: Comparison of sum of phase current error square.

Phase current error square σ	phase A	phase B	phase C
Proposed resonant notch link system	1575.4	1649.9	1592.4
Ordinary resonant link system	3313.6	3333.3	3402.3

5.6 Experimental Results Under Loaded Condition

Using a simple delay control, the preliminary experiment of the quasi-resonant link inverter under loading condition has shown successful operation at low power output. Figure 5.11 shows experimental results of quasi-resonant link current and voltage under loading condition.

The inverter input current alternatively changes between positive and negative, but the link voltage remains constant. As mentioned in Section 5.4, the pulse width can be optimized by least error square method. With controllable pulse width, the potentiality of the quasi-resonant link system for practical applications will be much greater than that of the resonant link system. Figure 5.12 shows $1.8A_{peak}$ load current at 20Hz fundamental frequency.



Upper trace - I_r (2 A/div)

Lower trace - V_d (20 V/div)

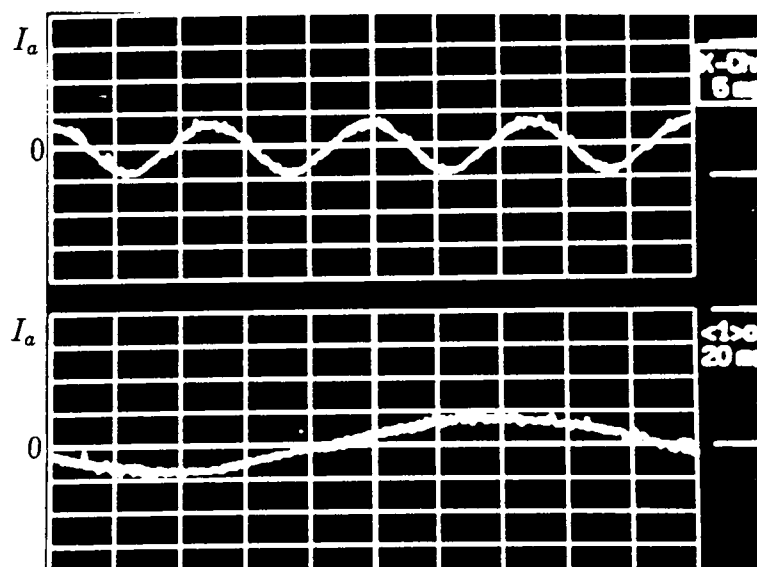
Resonant frequency - 60 KHz

Switching frequency - 22 KHz

Source voltage $V_s = 19V$

2 HP induction motor load

Figure 5.11: Experimental quasi-resonant current and voltage under loaded condition.



Upper trace – phase current at 20 msec/div

Lower trace – phase current at 5 msec/div

Resonant frequency – 60 KHz

Switching frequency – 22 KHz

Source voltage $V_s = 19$ V

2 HP induction motor load at $1.8 A_{peak}$ phase current

Figure 5.12: Experimental phase current of quasi-resonant link inverter for induction motor load.

The upper trace shows five cycles of a phase current, while the lower trace is expanded from one of the five cycles. To simplify the experiment, the quasi-resonant switch is controlled by a fixed frequency delay circuit so that the 60 KHz resonant frequency is regulated to 22 KHz. When comparing to experimental results of the 25KHz resonant link inverter (Figure 4.9), the current ripple is significantly less in the 22KHz quasi-resonant link inverter. With less switching, smaller passive components, and lower current ripple, the quasi-resonant *dc* voltage notching inverter presents superior performance and great potential for industry applications.

Chapter 6

Simulation Program Design

The complete resonant/quasi-resonant link inverter-machine speed control system has been modeled and simulated on an *IBM PC-AT* compatible computer *PEP301* using *PC-SIMNON*. *SIMNON* is a simulation package for solving differential and difference equations. The package has well-defined commands for dynamic system description, user-interface, and graphics.

The dynamic system can be described in differential equations for continuous systems and in difference equations for discrete systems. In *SIMNON*, two different systems can be modeled individually and linked together for complete system simulation. This feature is very convenient for power electronics system simulations because a power electronics system normally combines with several continuous and discrete subsystems.

For the resonant link inverter system, the *SIMNON* simulation diagram can be represented as shown in Figure 6.1. The continuous subsystem is represented by (C), and the discrete subsystem is represented by (D). Each subsystem contains input and output signals.

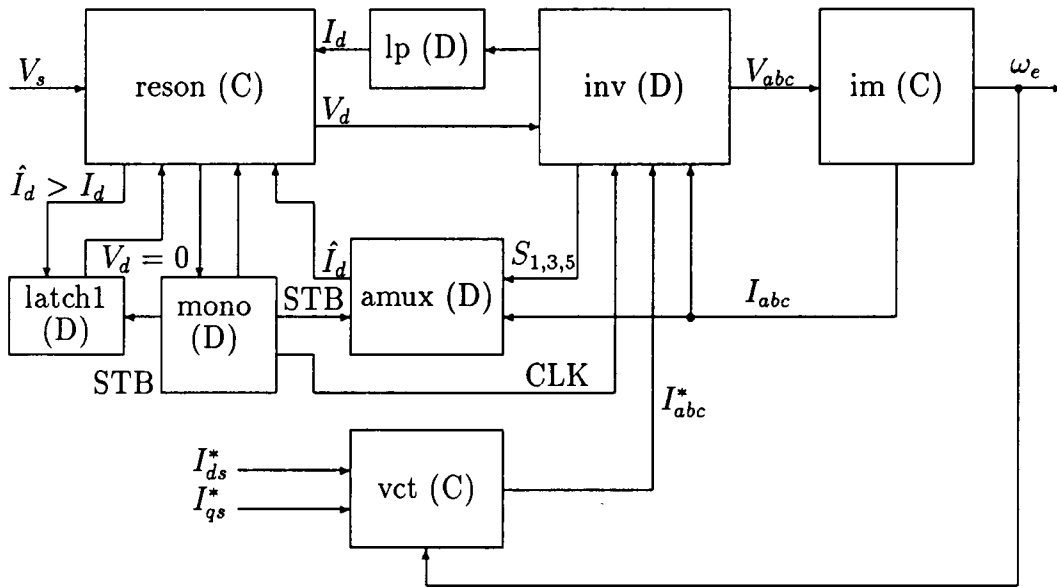


Figure 6.1: Block diagram of resonant link inverter drive for *SIMNON* simulation.

The function of each subsystem is listed below.

1. reson – the resonant link circuit with current initialization control. This block will be replaced by “modu” in the quasi-resonant voltage notching link inverter.
2. inv – a hysteresis band inverter controlled by zero-crossing pulses.
3. im – a 3-phase 5-HP induction motor modeled in dq equivalent circuits.
4. amux – a multiplexer circuit performing initial current prediction.
5. lp – a loop break to prevent *SIMNON* connection looping.
6. mono – a single shot generator to produce a short pulse at the beginning of the zero voltage crossing.

7. latch1 – a latch circuit to hold the condition that $\hat{I}_d > I_d$.
8. vct – a vector transformation to transform dq -axis currents into three-phase reference currents.

The complete simulation program code listing is shown in Appendix A. Before writing the *SIMNON* program code, each subsystem has to be modeled. The form of modeling may be in mathematical equations or in circuit diagrams. For Figure 6.1, the mathematical model of the blocks “reson” and “amux” have been described in previous chapters. This chapter will discuss how to model the hysteresis band inverter, induction motor, and other circuit elements.

6.1 Modeling of a Hysteresis Band Inverter

The inverter consists of six switches for a three-phase bridge circuit. Each switch contains an *IGBT* and an anti-parallel diode. Although the switching device has finite turn-on and turn-off time, it is not necessary to model device switching behavior for circuit topology study. An ideal switching device with zero switching time should be sufficient for the system-wide study. The device conduction voltage can also be neglected in inverter simulation. The control of the hysteresis band inverter can be simply modeled as a comparator with hysteresis feedback. Figure 6.2 shows one leg of inverter control circuit diagram. Comparator input signals are phase *A* reference and actual currents I_a^* and I_a , and the output (*A*) is fed into a D-type flip flop for latching. Using a synchronous signal (*CLK*) generated from zero voltage crossing circuit as the clock of the D-type flip flop, the inverter control becomes integral cycle *PWM* (*IPWM*).

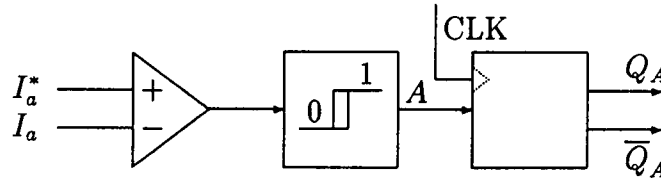


Figure 6.2: Circuit model of a hysteresis band inverter.

The output of the comparator (A) is a *TTL* signal. In *SIMNON* simulation, the *TTL* signal can be simply represented by either 0 or 1. Output signals of the D-type flip flop (Q_A and $\overline{Q}_A$) directly feed into gate drive circuits, where Q_A controls the upper leg, and $\overline{Q}_A$ controls the lower leg. Figure 6.3 shows the circuit diagram of the D-type flip-flop.

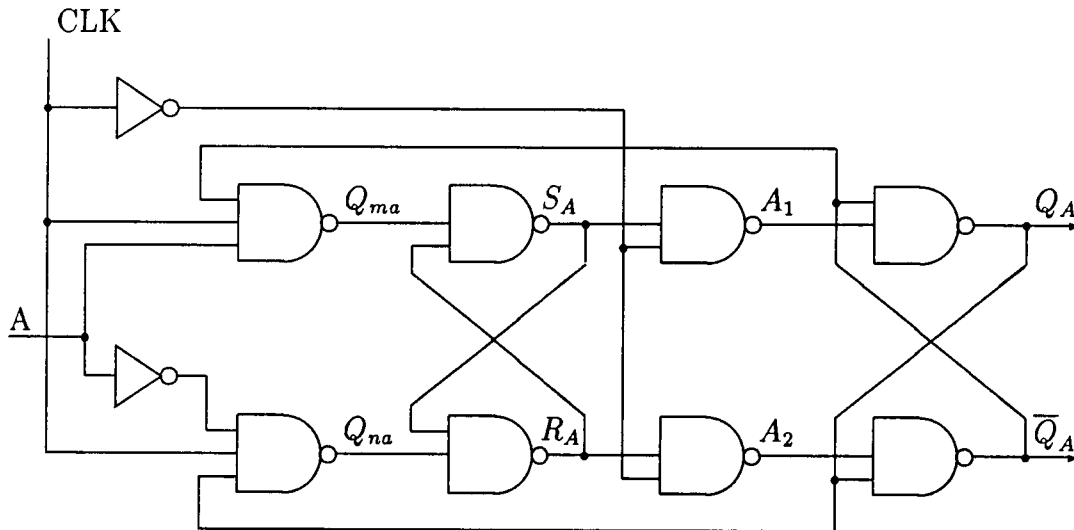
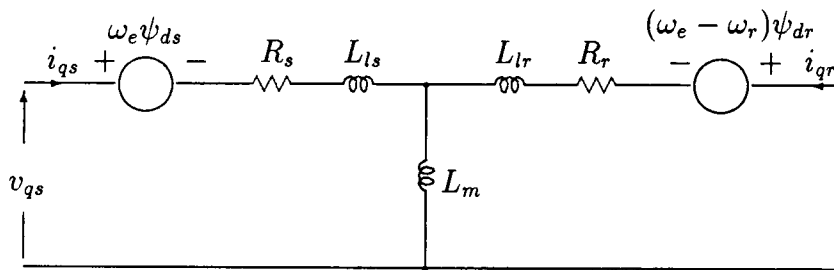


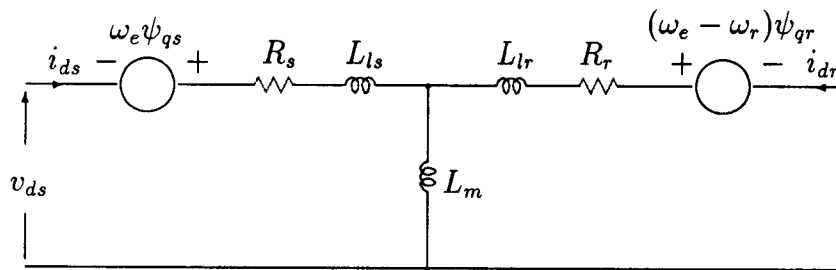
Figure 6.3: Circuit model of a D-type flip-flop.

6.2 Modeling of Induction Motor

The dynamic model of the induction machine is nonlinear and is in high order. In order to correctly describe the induction machine, a dq equivalent model at synchronous rotating frame [6] has been studied and simulated to verify the modeling effectiveness. Figure 6.4 shows dq equivalent circuits of the induction machine at synchronous rotating frame.



(a) q-axis circuit



(b) d-axis circuit

Figure 6.4: Induction machine dq equivalent circuits at synchronous rotating frame.

The notations are defined as

$$\psi_{qs} = L_{ls}i_{qs} + L_m(i_{qs} + i_{qr}),$$

$$\psi_{qr} = L_{lr}i_{qr} + L_m(i_{qs} + i_{qr}),$$

$$\psi_{ds} = L_{ls}i_{ds} + L_m(i_{ds} + i_{dr}),$$

$$\psi_{dr} = L_{lr}i_{dr} + L_m(i_{ds} + i_{dr}),$$

ω_e = supply angular frequency,

ω_r = rotor angular frequency,

and R_s , L_{ls} , R_r , L_{lr} , and L_m are machine parameters.

The state equation is derived from the above dq equivalent circuits for *SIMNON* simulation.

$$\begin{aligned} \frac{di_{qs}}{dt} &= \frac{1}{L_k} [L_r V_{qs} - L_r R_s i_{qs} - (\omega_e L_r L_s - \omega_{sl} L_m^2) i_{ds} \\ &\quad + \omega_e R_r i_{qr} - \omega_r L_r L_m i_{dr}] \\ \frac{di_{ds}}{dt} &= \frac{1}{L_k} [L_r V_{ds} + (\omega_e L_r L_s - \omega_{sl} L_m^2) i_{qs} - R_s i_{ds} \\ &\quad + \omega_r L_r L_m i_{qr} + L_m R_r i_{dr}] \\ \frac{di_{qr}}{dt} &= \frac{1}{L_k} [-L_m V_{qs} + L_m R_s i_{qs} + \omega_r L_m L_s i_{ds} \\ &\quad - L_s R_r i_{qr} + (\omega_e L_m^2 - \omega_{sl} L_s L_r) i_{dr}] \\ \frac{di_{dr}}{dt} &= \frac{1}{L_k} [-L_m V_{ds} - \omega_r L_m L_s i_{qs} - L_m R_s i_{ds} \\ &\quad - (\omega_e L_m^2 - \omega_{sl} L_r L_s) i_{qr} - L_s L_r i_{dr}] \\ \frac{d\omega_r}{dt} &= (T_e - T_L)/J \\ T_e &= \frac{3P}{4} [L_m (i_{dr} i_{qs} - i_{qr} i_{ds})] \end{aligned} \tag{6.1}$$

where

$$L_k = \frac{1}{L_s L_r - L_m^2}$$

$$\omega_{sl} = \omega_e - \omega_r$$

T_e = generated torque

T_L = load torque

J = inertia

P = number of poles

6.3 Modeling of Some Circuit Elements

In Figure 6.1, the blocks “reson”, “inv”, and “im” contain main circuit elements and most of the system dynamics. There are five state variables in “im” and three in “reson.” The inverter model is discrete which takes less simulation time although it contains more state variables. Some small circuit elements are included in for control purposes. The function of “latch1” is the same as a D-type flip-flop and will not be discussed again. Modeling of other circuit elements will be presented in this section.

6.3.1 Loop Break

In connecting all subsystems, *SIMNON* complained of a loop caused by *dc* link current connection. This can be solved by adding a simple filter circuit to break the loop. The difference equation of the filter is expressed in (6.2).

$$\begin{aligned} y(k+1) &= 0.9y(k) + 0.1I'_d \\ I_d(k) &= y(k) \end{aligned} \tag{6.2}$$

I'_d means the inverter input current from the inverter side, and I_d represents the inverter input current at the resonant link circuit side. $y(k)$ is the state variable at time k , and $y(k+1)$ is at time $k+1$. With sampling time 1×10^{-8} , the small loop break solves the *SIMNON* connection problem.

6.3.2 Single Shot Generator

When the resonant voltage drops to near zero, a pulse needs to be generated for the inverter clock and other latching purposes. This can be implemented by a single shot generator circuit. The simulation diagram of the single shot generator is shown in Figure 6.5(a).

The clock signal CLK is always zero with the exception of a short pulse which occurs at the beginning of zero voltage crossing. This pulse width is adjustable by changing the sampling time of the program. Figure 6.5(b) shows typical signal waveforms of the single shot generator.

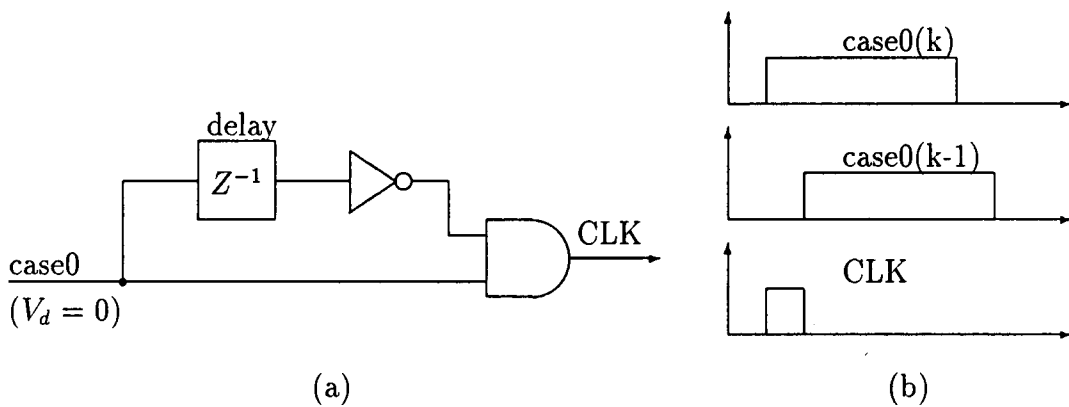


Figure 6.5: Circuit Model of a single shot generator.

6.3.3 Vector Transformation

Normally the reference dq currents are obtained from a vector control loop as shown in Figure 4.5. For only one fundamental cycle study of the resonant link inverter system, this loop can be excluded. The single cycle dq reference currents I_d^* and I_q^* can be obtained from a steady-state study and fed into the *SIMNON* program.

The vector transformation requires the speed signal to generate sinusoidal refer-

ence phase currents. There are two stages of transformation: (a) from synchronously rotating frame to stationary reference frame, (b) from stationary reference frame to three-phase [5].

The relation between synchronous rotating frame and stationary reference frame can be expressed as

$$\begin{aligned} i_{dss}^* &= i_{ds}^* \cos \omega t - i_{qs}^* \sin \omega t \\ i_{qss}^* &= i_{ds}^* \sin \omega t + i_{qs}^* \cos \omega t \end{aligned} \quad (6.3)$$

where i_{dss}^* and i_{qss}^* represent stationary reference frame currents.

For three-phase currents, the conversion becomes

$$\begin{aligned} i_a^* &= i_{qss}^* \\ i_b^* &= -\frac{1}{2}i_{qss}^* - \frac{\sqrt{3}}{2}i_{dss}^* \\ i_c^* &= -\frac{1}{2}i_{qss}^* + \frac{\sqrt{3}}{2}i_{dss}^* \end{aligned} \quad (6.4)$$

Chapter 7

Hardware Circuit Design

The hardware circuit design for the proposed resonant link inverter with bidirectional initial current control is based on the algorithm described in Chapters 4 and 5 and the *SIMNON* simulation program. The actual design value may be different from the simulation study because of the limitation of the component availability.

The whole system is separated into several levels. The bottom level of the system is a heat sink with power devices and their connections. Above the power devices are the gate drive circuit and gate drive power supplies, mounted on a wire-wrapping board. The current sensors are embedded between the power circuit connections and emerge from the wire-wrapping board. The control circuit sits on top of the gate drive circuit along with the connections for gate control signals and sensor signals. The top level of the system consists of a F/V converter circuit, some isolation circuits, and two voltage sensing circuits. The motor speed feedback is connected to the F/V converter circuit through an isolation amplifier to the computer data acquisition system. Two computer output signals are connected to the control circuit through isolation circuits.

Figure 7.1 shows the photograph of the complete hardware experimental setup. The inner part is a heat sink with power devices and gate drive circuitry on the top,



Figure 7.1: Photograph of the experiment setup.

and the outer part is a control circuit with sensor and isolation circuit on the top. Shielded wires connect two sections:

1. Between the gate drive circuitry and the control circuit.
2. Between the isolation circuit and the computer data acquisition system.

The computer is not shown in the picture.

The right hand side of the experimental setup is the measurement system. Figure 7.2 shows the photograph of the resonant link inverter induction motor drive experiment measurement. The upper oscilloscope (TEK 2267) shows resonant link current and voltage, and the lower oscilloscope (Lecroy 9400) shows motor current and line-to-line voltage. Below the two oscilloscopes is a TM5006 box with three plug-in current probe amplifiers AM503.

7.1 Power Circuits

Using *IGBTs* as power switching devices, two power circuits have been constructed for resonant *dc* link and quasi resonant *dc* voltage notching inverters. By shorting the additional control switch, the second system can be easily switched back to the resonant *dc* link system. Figure 7.3 shows the complete power circuit along with voltage and current sensors for the resonant *dc* link inverter.

The *ac* source, obtained from a three-phase autotransformer, supplies a bridge rectifier (D_A) to yield a *dc* power which is filtered by capacitors (C_A and C'_A). The larger capacitor C_A filters low frequency ripples while the high- Q capacitor C'_A filters high frequency ripples. Diode D_B and capacitor C_b constitute an auxiliary source to provide negative initial current control.

S_a and S_b are bidirectional initial current control switches. When a negative initial current is required, S_b is on and S_a is off. Because the on-time of S_b is

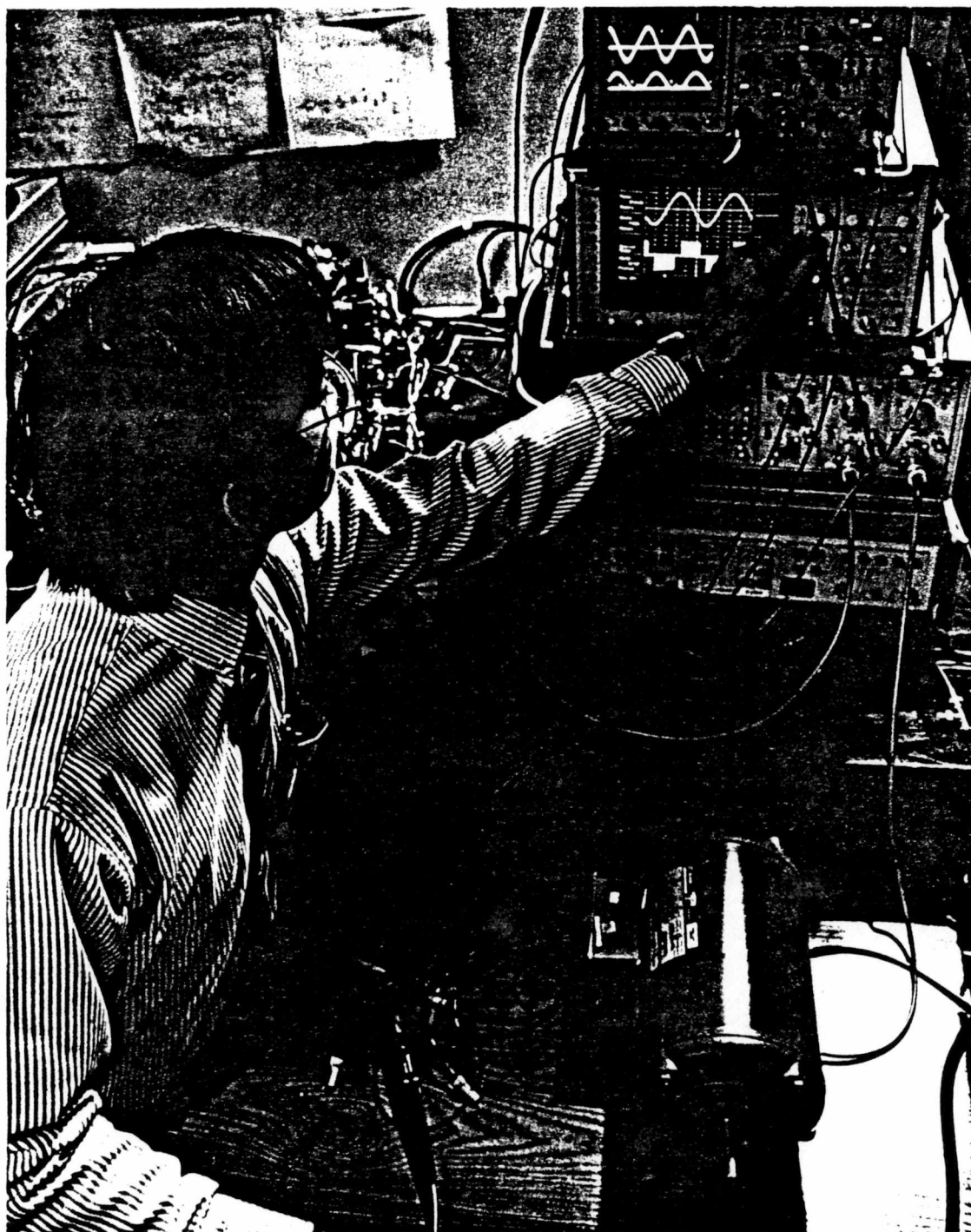


Figure 7.2: Photograph of the measurement setup.

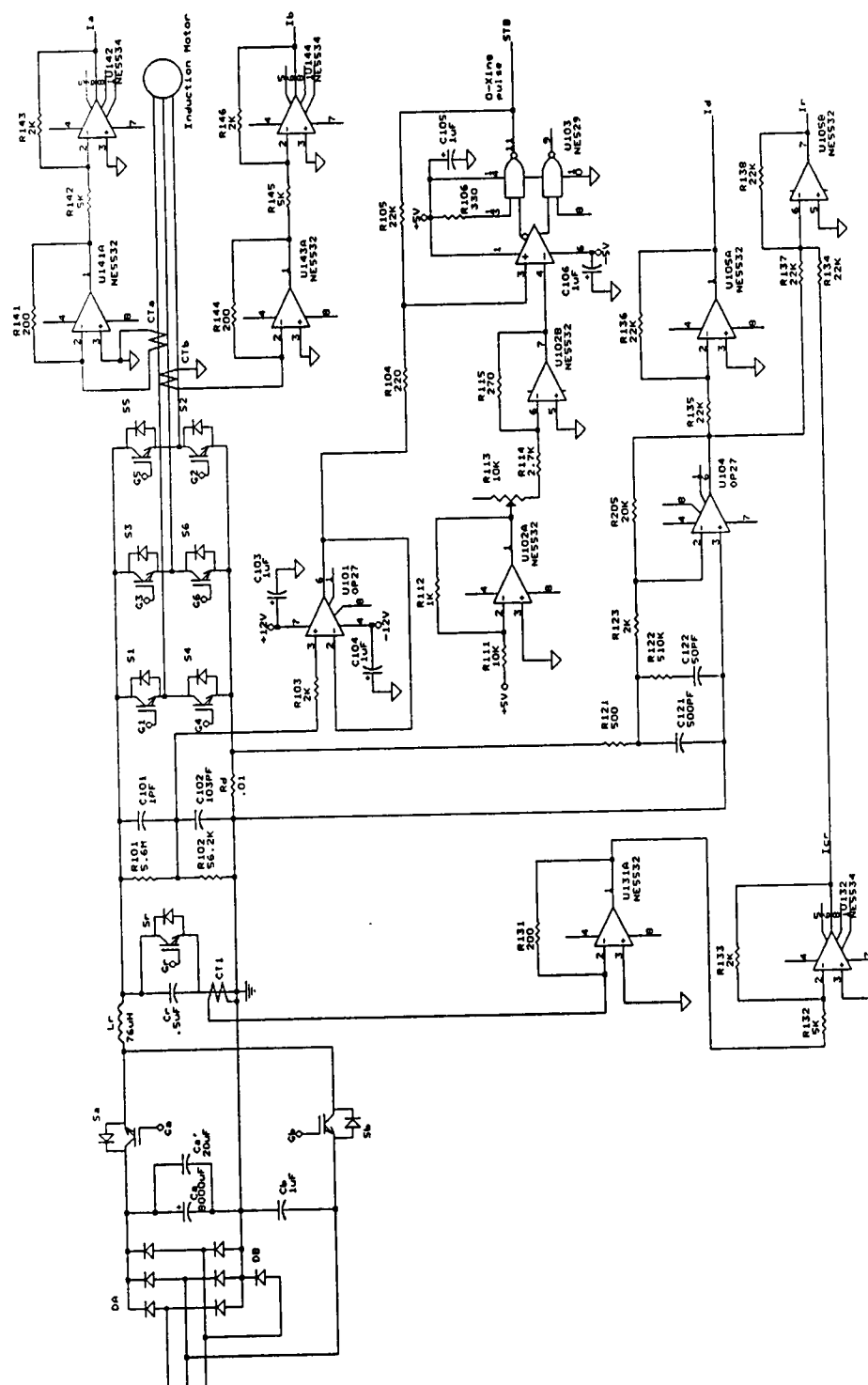


Figure 7.3: Bidirectional resonant dc link power circuit and sensors.

very short (typically $2 \sim 5 \mu\text{sec}$), the gate drive must be designed for short pulse operation. However, S_a operates for a long duty cycle and is in series with the resonant link circuit which requires a higher gate drive voltage to reduce voltage drop of *IGBT*. Both S_a and S_b require isolated gate-to-source voltage.

The resonant inductor L_r and the resonant capacitor C_r constitute the resonant link circuit. The inductor L_r is made of a Litz wire to avoid resistance increase in high frequency and is on an air core to avoid saturation. The capacitor C_r (polypropylene) is near ideal with .02% dissipation factor. Detailed design data of L_r and C_r are shown in the laboratory notebook [23].

The switching device is *MOSIGT* IXGH40N60A rated 600 V and 40 A [52]. Maximum specifications of the switching characteristics and experimental results are listed in Table 7.1.

Table 7.1: Switching characteristics of IXGH40N60A

Switching time		max. spec.	actual test
$t_{d(on)}$	turn-on delay time	100 nsec	70 nsec
t_r	rise time	200 nsec	100 nsec
$t_{d(off)}$	turn-off delay time	1 μsec	500 nsec
t_f	fall time		
	resistive	0.5 μsec	0.3 μsec
	inductive	0.8 μsec	— ¹

¹ The measurement of this value is affected by load inductance, diode reverse recovery time, and snubber circuit.

The total switching time should include gate drive circuit delay. For IR2110 gate drive, turn-on delay is 130 nsec, and turn-off delay is 100 nsec.

The voltage divider uses high resistance and low capacitance network R101, R102, C101, and C102 with 100:1 ratio. The amplitude and phase responses had

been tested with satisfactory results. A voltage follower U101 is employed to reduce output impedance. The zero crossing is detected using an ultra high speed comparator NE529. Because the device has a conduction voltage drop, the zero crossing does not need to be exactly zero voltage. In Figure 7.3, the zero crossing voltage is generated by an amplifier IC U102 and is adjustable by trimming R113 where the zero crossing reference voltage is generated by another voltage divider IC U102.

For *dc* current detection, a $0.01\ \Omega$ current sense resistor R_d is inserted into the inverter input bus. A resistor-capacitor network and amplifier circuits U104 and U105 are used to filter high frequency noise and obtain a transform ratio of $1\text{A}/0.1\text{V}$. Figure 7.4 shows test results for current sense resistor and amplifier circuits. The lower trace is the reference input current, and the upper trace is the output response.

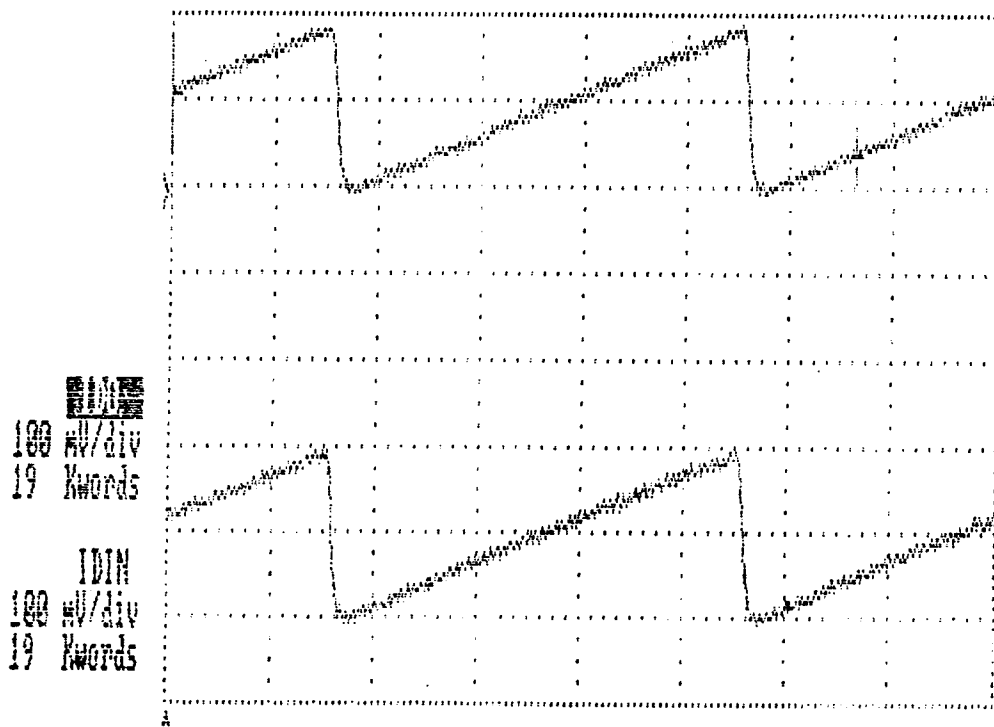


Figure 7.4: Experimental results of current sense resistor and amplifier circuits.

For *ac* current detection, high frequency ferrite *CTs* are used with current-voltage transform circuits. Three *CT* circuits detect

1. resonant capacitor current – $CT1+U131A+U132$,
2. phase A current – $CTa+U141A+U142$, and
3. phase B current – $CTb+U143A+U144$.

All current/voltage transform ratios are 1A/0.1V. The inductor current I_r is the sum of the resonant capacitor current I_{cr} and the inverter input current I_d and is obtained from the summing circuit U105B.

Figure 7.5 shows the complete power circuit and sensor conditioning circuits of the quasi-resonant *dc* voltage notch link inverter. The circuit structure is the same as Figure 7.3 with the exception of one additional switch pair $S_c - D_c$. In order to avoid power circuit noises, this circuit uses a Hall sensor for *dc* current sensing. The comparator output is also isolated from the power circuit. The offset voltage of the Hall sensor is very sensitive to temperature variation and an additional conditioning circuit must be added to compensate for such variation.

By shorting S_c , this circuit becomes a resonant *dc* link inverter. To avoid the skin effect, the inductor L_r is made up of a Litz wire consisting of 260 strands of gage #35 wire. Two inductors at 18.2 μH and 21.2 μH values have been tested. When using 18.2 μH , the resonant frequency is 65 KHz while the 21.2 μH gives 60 KHz.

The resonant capacitor C_r consists of four high voltage polypropylene Orange Drop capacitors SBE716P600V [58]. Each capacitor is rated at 600 V, 8 A_{rms} , and 0.082 μF with less than .02% dissipation factor.

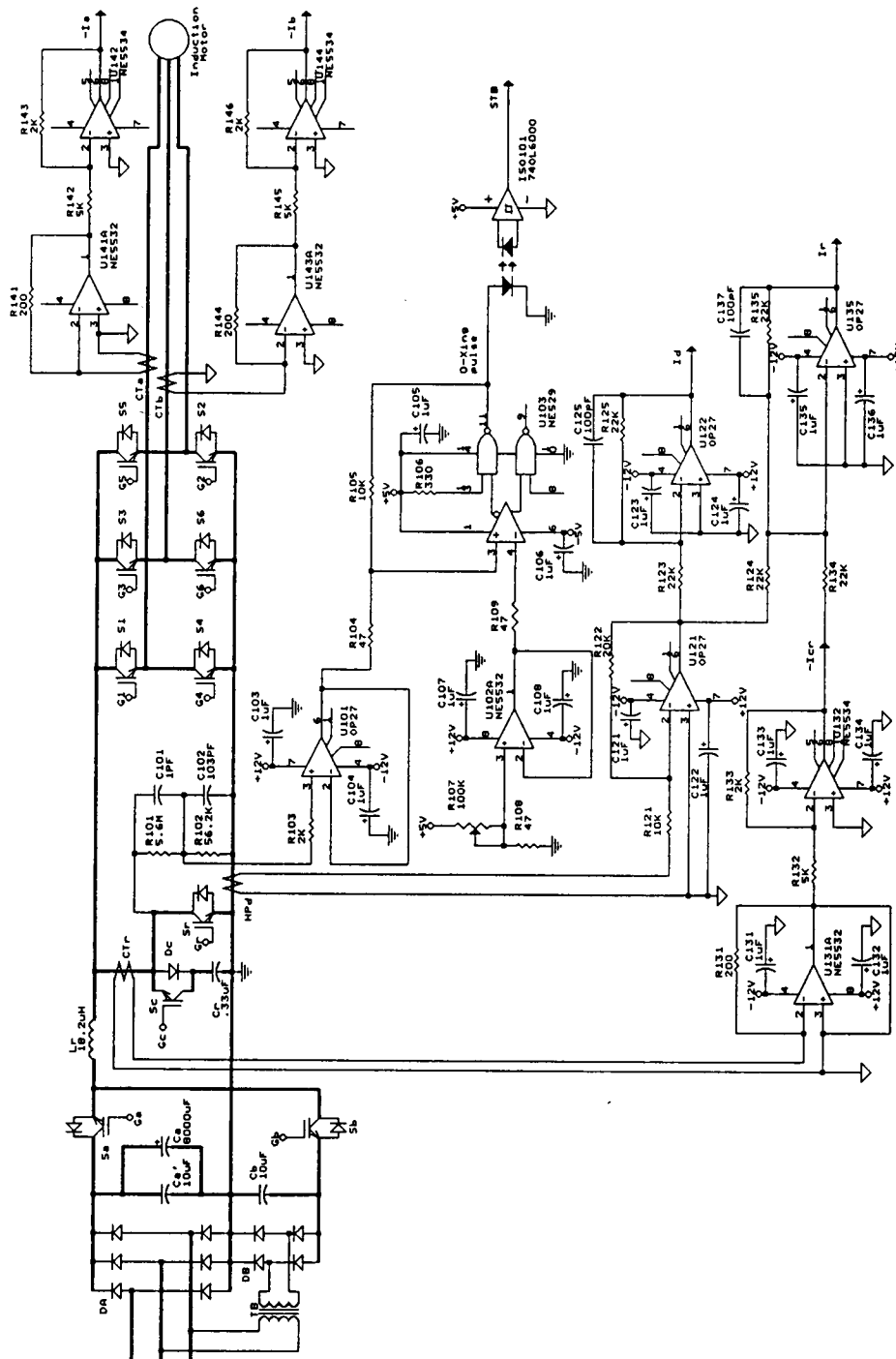


Figure 7.5: Quasi-resonant *dc* voltage notch link inverter power circuit.

7.2 Gate Drive Circuits

As mentioned in Section 7.1, all switches operate under different conditions. S_a operates in a long duty cycle, S_b and S_r operate in short pulse, and $S_1 \sim S_6$ operate in integral cycles, 40 μsec for 25 KHz and 16.6 μsec for 60 KHz. Those switches requiring an isolation gate drive voltage are S_a , S_b , S_c , S_1 , S_3 , and S_5 . Traditional transistor gate drive circuits are simple and fast but lack isolation. Transformer isolation is good for integral pulse operation but not for long duty cycle because the transformer will not hold dc voltage. A recently released new gate drive IC IR2110 using a bootstrap technique to solve the isolation problem has been tested with satisfactory results. An essential condition of all gate drive circuits is the *TTL* interface. Several gate drive circuits which have been built and tested are described below.

7.2.1 Gate Drive Circuit Using Transistors

Figure 7.6 shows a complementary transistor gate drive circuit capable of sinking or sourcing approximately 0.8 A. The choice of transistor is based on switching speed and current capability. The transistor pair 2N2222A and 2N2907A are commonly used. The transistor pair 2N3725 and 2N3467 can be alternative choices.

A simplified calculation for transistor current in the switching operation is shown as the following numeric example [49].

Calculation of Transistor Gate Drive Current

For a typical *IGBT* the specifications are $C_{GS} = 5000$ pF, $C_{DG} = 70$ pF, $V_{GS} = 15\text{V}$, and $V_{DS} = 300$ V. Assume the gate-to-source voltage is linearly charged in 200

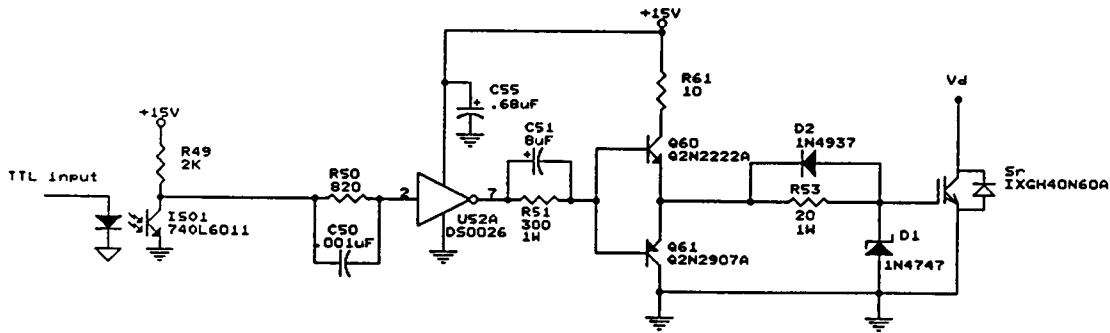


Figure 7.6: Gate drive circuit using transistors.

nsec. The average of current pulse can be calculated as

$$\begin{aligned}
 I_{G1} &= \frac{C_{GS} V_{GS}}{t_r} \\
 &= \frac{5000 \times 10^{-12} \times 15}{200 \times 10^{-9}} \\
 &= 0.375 \text{ A}
 \end{aligned}$$

Further assume that the gate-to-drain capacitance is discharged simultaneously. This gives

$$\begin{aligned}
 I_{G2} &= \frac{C_{DG} V_{DS}}{t_r} \\
 &= \frac{70 \times 10^{-12} \times 300}{200 \times 10^{-9}} \\
 &= 0.105 \text{ A}
 \end{aligned}$$

The two currents sum up to 0.48 A in this example. Transistor 2N2222A with collector current 0.8 A is sufficient to handle the gate current sourcing. In fact, for zero voltage switching the second current can be neglected. Similar calculations can be done for current sinking.

In a driving transistor base, a *TTL* inverter 74AS00 (or 74AS04) and a *CMOS* driver DS0026 provide an interface to the control circuit. This can be simplified using a single driver device or an operational amplifier.

7.2.2 Gate Drive Circuit Using IR2110

The advent of IR2110 largely reduces an engineer's effort in designing a gate drive circuit for *MOS* gate devices including all power *MOSFET* families and *IGBTs*. The peak output current can be as high as 2 A. A bootstrap technique has been used to solve isolation problems on the high side of half bridge power *MOSFETs* or *IGBTs*. Figure 7.7(a) shows the internal structure of IR2110 [50]. The high side input signal H_{IN} passes through a Schmitt-triggered *CMOS* logic and a pulse generator to a level-shifted output. The voltage V_{BS} is bootstrapped by an external diode D_{BOOT} and a capacitor C_{BOOT} as shown in Figure 7.7(b) [51]. A small resistor R_{BOOT} is normally in series with the diode to limit the rate of voltage rising on C_{BOOT} and to prevent possible spurious high side output.

The design of external components depends on the gate charge current, switching frequency, switching device on-time, and quiescent current of IR2110. A simple procedure of designing external D_{BOOT} , R_{BOOT} , and C_{BOOT} for *IGBT* application is shown as below.

Design Procedure for IR2110 Gate Drive Circuit

Consider a typical *IGBT*.

Gate-to-source capacitance $C_{GS} = 5000$ pF,

Gate-to-source voltage $V_{GS} = 15$ V,

Gate-to-source threshold voltage $V_{GS(th)} = 5$ V,

Switching frequency $f = 25$ KHz,

Maximum on-time $t_{on} = 400$ μ sec.

The quiescent current I_{QBS} of IR2110 is given as 500 μ A.

The average current carried by D_{BOOT} equals sum of the gate charge current of $IGBT$ and the quiescent current of IR2110, that is,

$$I_{BS} = I_{QBS} + I_{GS} \quad (7.1)$$

or

$$\begin{aligned} I_{BS} &= I_{QBS} + Q_G f \\ &= I_{QBS} + C_{GS} V_{GS} f \\ &= 0.5 \times 10^{-3} + 5000 \times 10^{-12} \times 15 \times 25 \times 10^3 \\ &= 2.3 \text{ mA} \end{aligned}$$

This current determines the rating of D_{BOOT} and R_{BOOT} . A typical 10Ω , $1/4$ W resistor and a 1 A ultra fast recovery diode such as Motorola MUR160 will be sufficient for most applications.

The design of C_{BOOT} depends on the gate-to-source threshold voltage. Assume the initial charging voltage is 15 V. The capacitor should prevent voltage drop to where V_{GS} would be higher than $V_{GS(TH)}$. Since the voltage drop across the IR2110 output buffer is 1 V, the bootstrap voltage V_{BS} must be kept above $V_{GS(TH)} + 1 = 6$ V. This means the voltage drop of C_{BOOT} during switching and conduction should be less than $15 - 6 = 9$ V. The voltage drop caused by turning on $IGBT$ can be calculated by

$$\Delta V_{sw} = \frac{Q_G}{C_{BOOT}} \quad (7.2)$$

The voltage droop during steady-state conduction drawn from C_{BOOT} is

$$\Delta V_{on} = \frac{I_{QBS} t_{on}}{C_{BOOT}} \quad (7.3)$$

Thus the total voltage drop of C_{BOOT} is

$$\Delta V = \frac{Q_G + I_{QBS} t_{on}}{C_{BOOT}} \quad (7.4)$$

For 400 μsec maximum on time of IGBT, the bootstrap capacitor can be calculated by

$$\begin{aligned}
 C_{BOOT} &= \frac{Q_G + I_{QBS} t_{on}}{\Delta V} \\
 &= \frac{5000 \times 10^{-12} \times 15 + 0.5 \times 10^{-3} \times 400 \times 10^{-6}}{9} \\
 &= .031 \mu\text{F}
 \end{aligned}$$

Figure 7.8 shows the gate drive circuit using IR2110 with the above calculated results. In this circuit, the TTL circuit using 74AS00 is to serve as a buffer, and C_1 and C_2 are bypass capacitors. The circuit has been built and tested successfully up to 250 KHz operation.

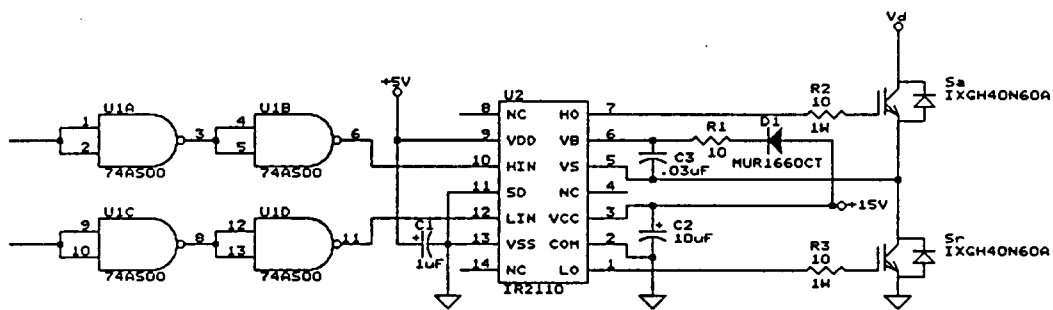


Figure 7.8: Gate drive circuit using IR2110.

Table 7.2 shows a comparison of test results and specifications for low side channel. The test results are worse than those of maximum specification because of higher loading.

Table 7.2: A comparison of test results and IR2110 specification

		maximum specification	test results
$t_{d(on)}$	turn-on delay	110 nsec	130 nsec
$t_{d(off)}$	turn-off delay	90 nsec	100 nsec

7.2.3 Gate Drive Circuit Using Pulse Transformer

A pulse transformer directly isolates the gate-to-source voltage. A positive pulse charges up gate-to-source capacitance C_{GS} to turn on *IGBT* while a negative pulse quickly discharges C_{GS} to turn off *IGBT*. This approach is simple, but the transformer design largely limits the duty cycle and switching frequency. Figure 7.9 shows a gate drive circuit using pulse transformer T1. The designed data of the pulse transformer is shown in Table 7.3.

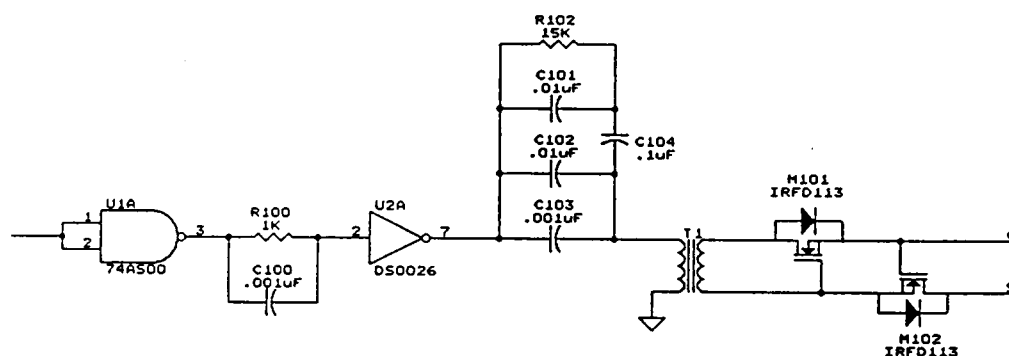
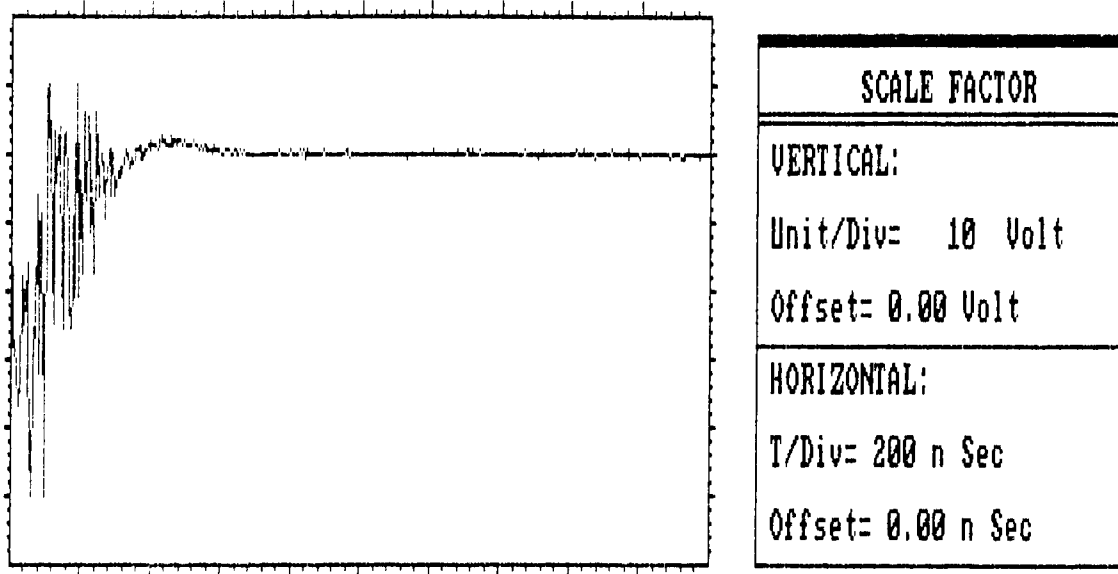


Figure 7.9: Gate drive circuit using pulse transformer.

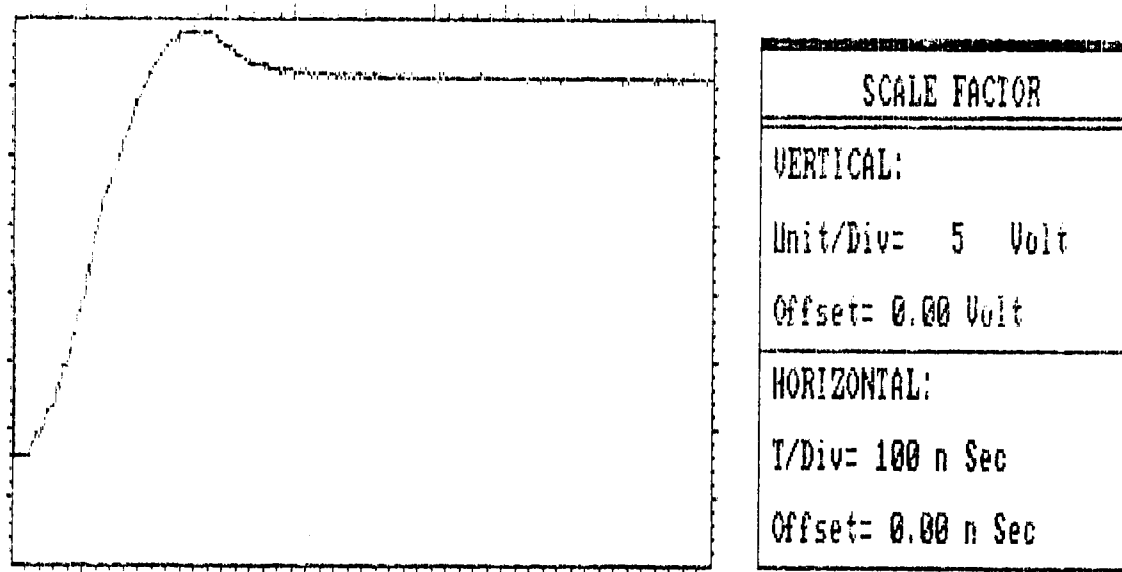
Table 7.3: Designed data of the pulse transformer

Core:	Ferroxcube 3E2A 266CT125 [47].
Winding:	30 AWG TEFZEL insulation bifilar, 12 turns, 1:1 ratio.

Because the turn-on speed of the pulse transformer drive is very fast, the lead length from gate drive to the *IGBT* must be as short as possible. Figure 7.10(a) shows an oscillation of gate-to-source voltage when the gate drive outputs connect through a 1.5 inch lead to *IGBT*. The peak voltage of approximately 30 V will easily damage *IGBT*. After the lead is cut to less than 0.5 inch, the ringing disappears as



(a) Gate drive output response when the gate lead is longer than 1.5 in.



(b) Gate drive output response when the gate lead is shorter than 0.5 in.

Figure 7.10: Gate drive output responses using pulse transformer.

The single chip IXLD429 provides a 6 A peak drive current and 25 nsec rise/fall time with a 2500 pF load. An optocoupler 74OL6011 [48] provides isolation between the gate drive and the control circuit. Total turn-on or turn-off delay is about 100 nsec. An RC circuit associated with a Schmitt trigger inverter 74HC14 [61] generates 1 MHz oscillating frequency for upper leg isolated dc -to- dc power supply. When switching at 100 KHz, the gate drive consumes less than 100 mA.

Figure 7.12 shows the experimental result of gate-to-source voltage (V_{gs}) response of $IGBT$ (IXGH60N60A) using $CMOS$ gate driver (IXLD429). The upper trace shows the gate-to-source voltage, while the lower trace shows the TTL gate input signal. Although the turn-on rise time is fast (about 100 nsec), the turn-on delay is long (about 200 nsec). This turn-on delay includes both the optoisolator and the $CMOS$ gate driver. The turn-off fall time is about 1 μ sec with the same 200 nsec delay.

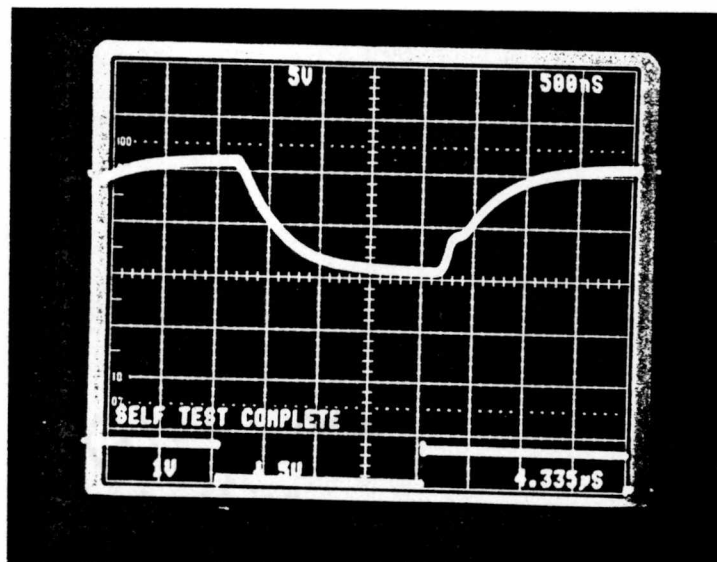


Figure 7.12: Gate-source voltage response of $IGBT$ using $CMOS$ gate driver.

7.3 Control Circuit

Figure 7.13 shows the complete control circuit of the bidirectional initial current controlled circuit with hysteresis band *PWM* circuit.

The zero voltage crossing signal *STB* triggers a 0.1 μsec *CLK* signal through IC U212A. U212B generates a 1 μsec minimum on-time for the shunt switch S_r . The initial resonant capacitor current is determined by a reference voltage divider (R208 and R209) and a voltage follower U203A. The initial current prediction is implemented by a *CMOS* analog multiplexer U208. The comparator U210B determines positive or negative initial current, and the comparator U210A determines switching condition.

When a positive initial current is requested by U217A (pin 5), S_r turns on until $I_r > I_d^*$ with a 1 μsec minimum on-time controlled by U212B.

When a negative initial current is requested by U217A (pin 6), a 4 μsec minimum on-time pulse generated by U216A ensures S_b gate drive will not be latched-up. U216B generates a 8 μsec maximum on-time pulse to prevent unusual switching. The signal S'_b is fed into the lock-out circuit described in Section 7.3.2.

An overcurrent protection circuit is added to prevent fault condition. When there is a short circuit or fault condition causing current to be over 50 A, the protection circuit will turn off all gate signals to protect power devices.

7.3.1 Control Circuit for the Quasi-Resonant Switch

The only difference between the resonant link circuit and the quasi-resonant link circuit control is the additional switch for the quasi-resonant control switch – S_c of Figure 7.5. The optimal control algorithm for the quasi-resonant link inverter can be implemented by three multipliers, a summing circuit, and a comparator circuit. However, a simple delay circuit can quickly prove the concept of the quasi-resonant

link inverter. Figure 7.14 shows a simple delay circuit for quasi-resonant switch control.

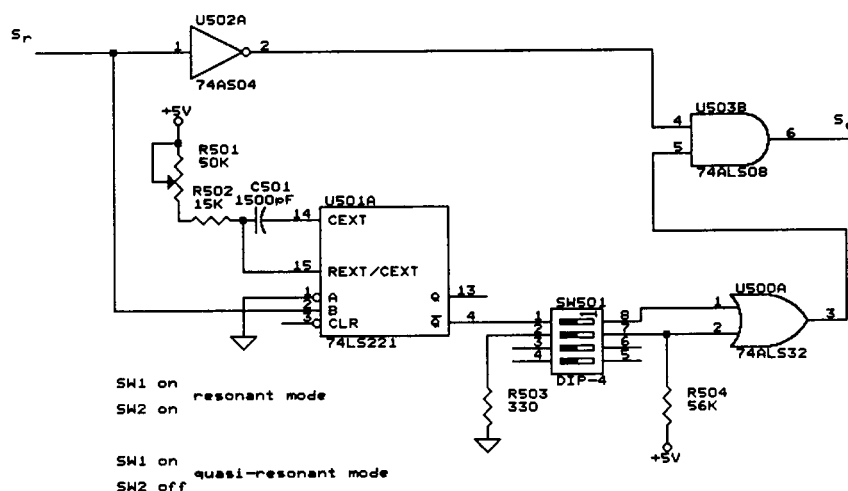


Figure 7.14: Schematic for quasi-resonant control switch circuit.

The circuit employs a 74LS221 single shot circuit to generate the delay, and a dual-in-parallel (*DIP*) switch to change the operation mode. When the switch S_2 of the *DIP* is on, switch S_c is under the delay circuit control while turning off S_2 will return the circuit back to resonant link inverter.

7.3.2 Lock-out Circuit

Switches S_a and S_b cannot be switched simultaneously, and a lock-out circuit is required to prevent shoot-through. Figure 7.15 shows a lock-out circuit for switches S_a and S_b .

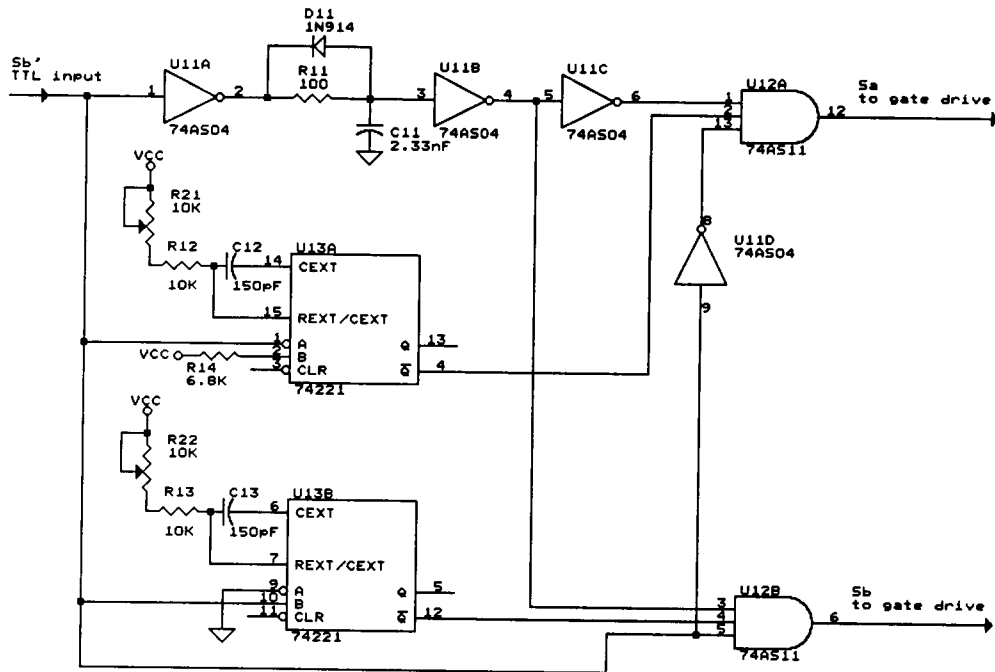
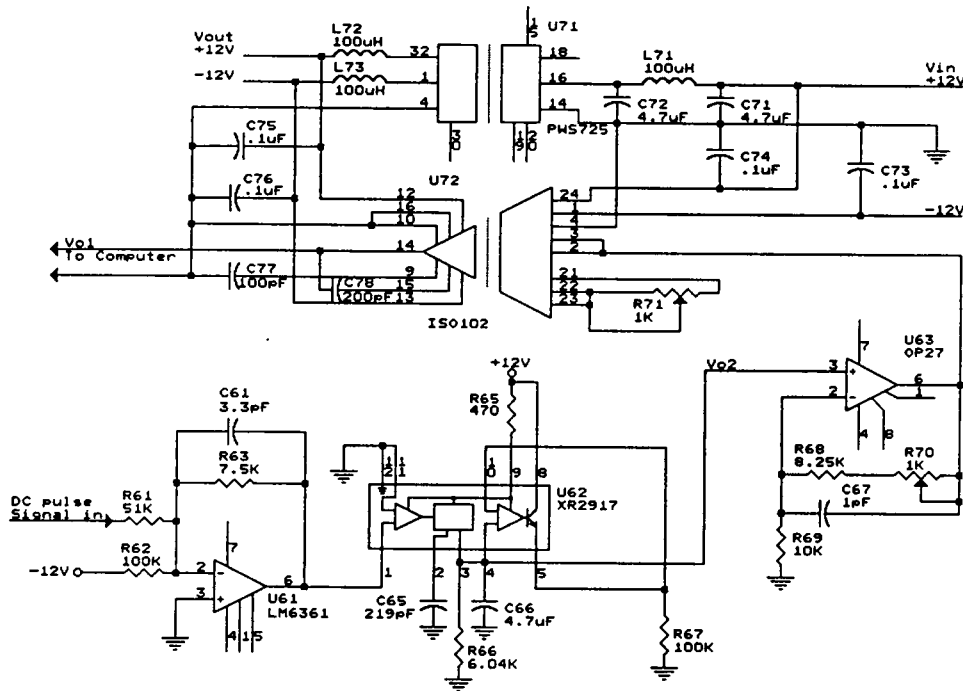


Figure 7.15: Lock-out circuit.

The pulse width of single shot generators U13A and U13B can be adjusted by trimming potentiometers R_{21} and R_{22} . The lock-out time is then controlled by single shot outputs. The network consisting of U11A-R11-C11-D11-U11B is a deglitch circuit.

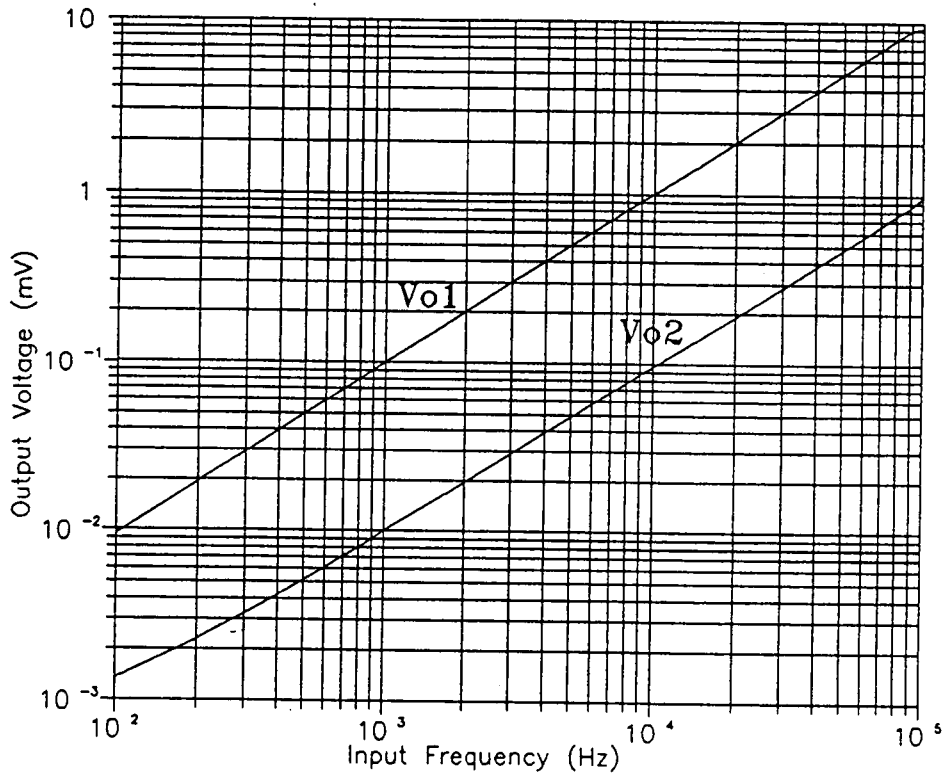
7.4 F/V Converter Circuit Design

The speed feedback through the encoder is high frequency pulse signals. Because the sampling rate of the computer data acquisition is not fast enough to count high frequency pulses, a frequency-to-voltage (F/V) converter is used to convert high frequency pulses to linearized voltage. The F/V converter selected here is XR2917 [46]. The complete F/V converter circuit, including amplification and isolation, is shown in Figure 7.16.

Figure 7.16: Schematic for F/V converter with isolation circuit.

The maximum input frequency is designed at 100 KHz. Because the input of the F/V converter must be *ac*, a simple level shifting stage has been added to the circuit by using high frequency operational amplifier LM6361 [57]. To reduce the noise level of the data acquisition system, the output of the F/V converter V_{o2} is magnified by a non-inverting amplifier OP-27 [56].

The output of the amplification stage V_{o1} goes into an isolation circuit which separates the motor speed signal and the computer signal. The gain can be adjusted by R70, and the offset can be trimmed by R71. After careful trimming, the highly linear output presents up to 90 KHz as shown in Figure 7.17. The upper trace shows the output of the isolation amplifier U72, and the lower trace shows the output of the F/V converter U62.

Figure 7.17: F/V converter output.

In order to see the delay of the F/V converter, the step responses were measured as shown in Figure 7.18. The experimental results were fed into an identification program to find the transfer function of the converter circuit. Using the first-order approximation, the transfer function was solved as shown in (7.5).

$$G(s) = \frac{1 \times 10^4 \text{ (V)}}{1 + 0.135s \text{ (Hz)}} \quad (7.5)$$

The approximation results are also shown in Figure 7.18. It can be seen that the model is precisely estimated by the first order transfer function.

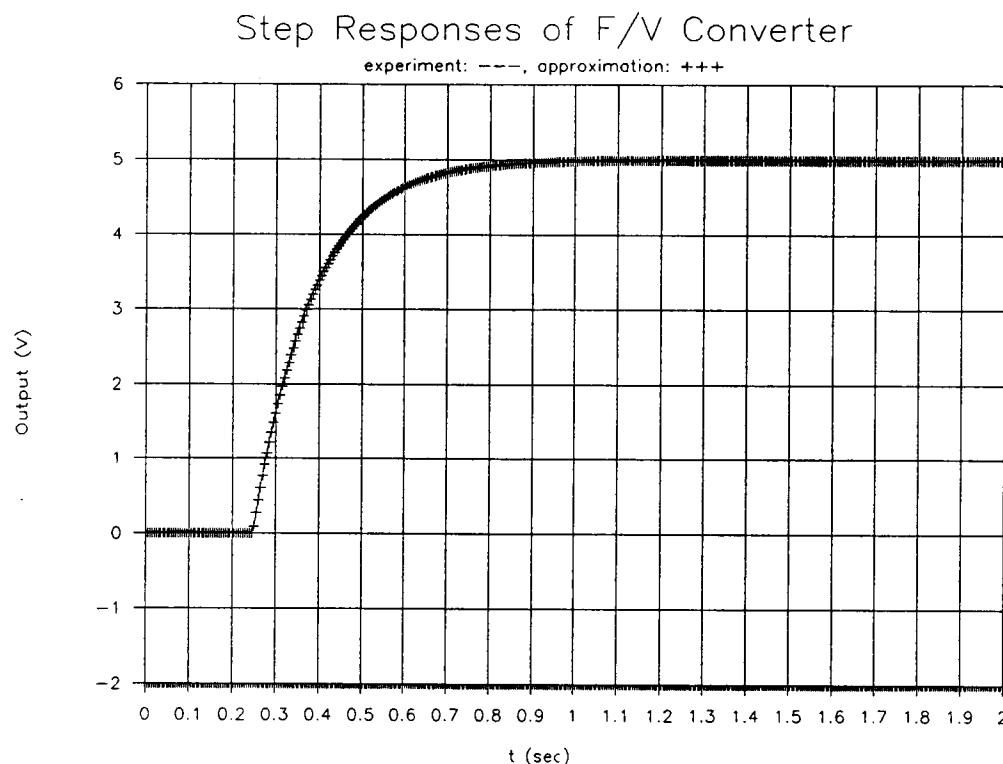


Figure 7.18: Experimental and approximated step responses of the F/V converter circuit.

7.5 Isolation Amplifier Circuit Design

The F/V converter has isolation between the F/V converter circuit and the computer A/D input. Similarly, an isolation is added between the computer D/A outputs and the control circuit as shown in Figure 7.19. Two input signals from computer D/A ports IN1 and IN2 are fed into isolation amplifiers U42 and U43. With 1:1 amplifying ratio, output signals OUT1 and OUT2 are connected to the control circuit.

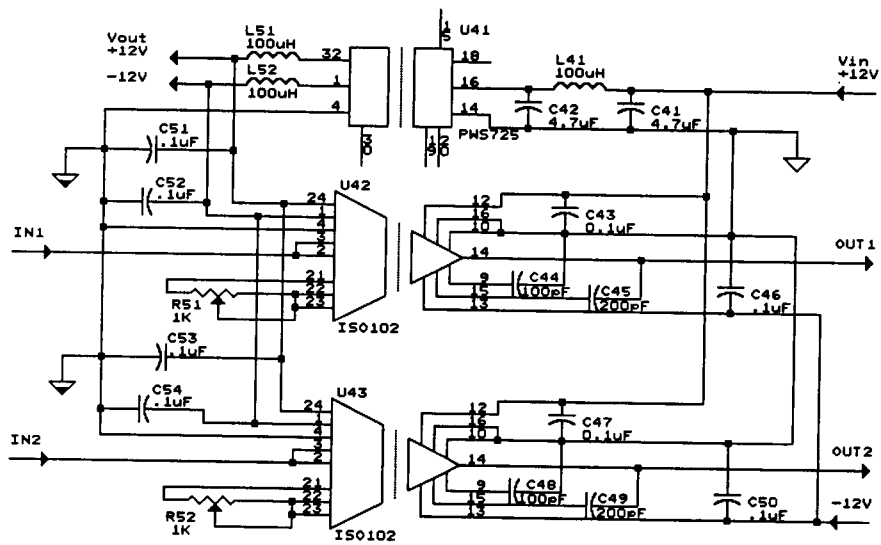


Figure 7.19: Schematic for isolation amplifier circuit.

Chapter 8

Speed Control System and Software Design

The induction motor speed is controlled by a microcomputer system. This system comprises a 16 MHz clock 80386 *CPU*¹ *PC-AT* compatible personal computer – *PEP-301*² and a 40MHz clock TMS320C25 *CPU*³ data acquisition system – *DSP-16*⁴. The *PEP-301* serves for user interactive interface and program development while the *DSP-16* does most computations and data *I/O*.

The complete computer speed control system block diagram for resonant *dc* link inverter or quasi-resonant *dc* link inverter drive is shown in Figure 8.1.

The data acquisition system includes two channel analog-to-digital (*A/D*) inputs and two channel digital-to-analog (*D/A*) outputs. Both input and output signals are filtered by 20KHz anti-aliasing low pass filters. Only one input channel is used

¹Trademark Intel Corp.

²Trademark of Tektronix Inc.

³Trademark of Texas Instrument Inc.

⁴Trademark of Ariel Corp.

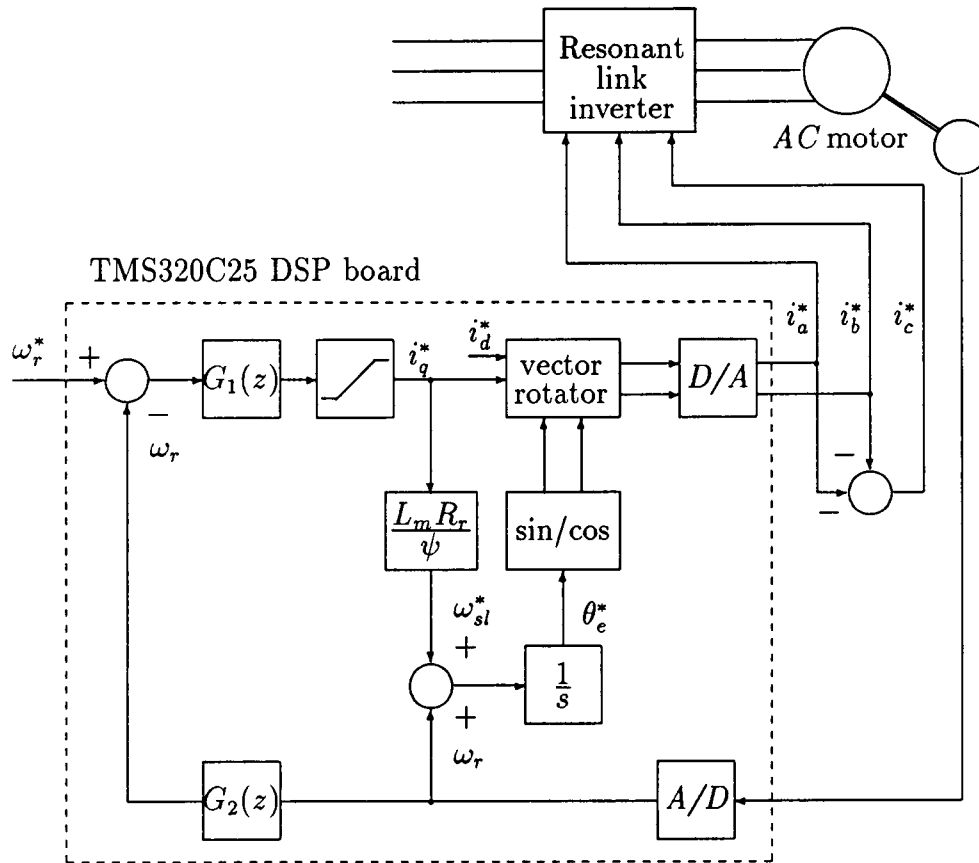


Figure 8.1: Computer speed control system for resonant link inverter drive.

for speed feedback while two output channels are feeding reference current signals into the resonant link inverter.

The sampling rate is programmable. While the highest sampling rate is 50 KHz per *I/O* channel, this program sets both input and output sampling rate at 15 KHz to allow more computation and communication time.

Since the TMS320C25 does not support floating point computation, the program design requires some parameter scaling. With special pipeline CPU structure, however, the digital filters can be implemented in very efficient codes. The complete TMS320C25 assembly program is listed in Appendix B.

8.1 Data Input and Output

There are 16 *I/O* ports in TMS320C25, numbered from 0 to 15. The *DSP-16* uses port 0 as an analog/digital (*A/D*) input and port 13 as a *D/A* output. The handshaking can be performed by port 10 – reading data from *PC*; port 15 – writing data to *PC*; and port 14 – host *I/O* port status. Both *A/D* and *D/A* are interrupt driven. This section will discuss interrupt service routine design and *I/O* scaling.

8.1.1 Interrupt Service Routines Design

The TMS320C25 has three external maskable user interrupts ($\overline{INT0} \sim \overline{INT2}$). By changing the value of the Interrupt Mask Register (*IMR*), the desired interrupt can be obtained. Whenever the interrupt is enabled, the interrupt service routines will be executed following certain priorities. The interrupt memory locations and priorities are shown in Table 8.1.

The program starts from memory location 0 to initialize operation modes and to load system parameters. Interrupt 0 is not used here. The *DSP-16* board employs

Table 8.1: Interrupt memory locations and priorities

Interrupt name	Memory location	Priority	IMR bit	Function
$\overline{RS}$	0	1 (highest)	–	external reset signal
$\overline{INT0}$	2	2	0	external user interrupt # 0
$\overline{INT1}$	4	3	1	external user interrupt # 1
$\overline{INT2}$	6	4	2	external user interrupt # 2
$TINT$	24	5	3	interrupt timer interrupt
$RINT$	26	6	4	serial port receive interrupt
$XINT$	28	7	5	serial port transmit interrupt

$RINT$ as the input interrupt and $\overline{INT1}$ and $\overline{INT2}$ as the output interrupt. With IMR bit locations 4, 2, and 1 corresponding to $RINT$, $\overline{INT2}$, and $\overline{INT1}$, by loading 22 ($2^4 + 2^2 + 2^1$) into IMR , the program will execute the instructions at memory locations 4, 6, and 26 whenever the interrupt is enabled.

Using the same input and output sampling frequency, the interrupt service routines will be executed at every sampling period without conflict. The interrupt is only disabled during initialization and PC bus requesting communication.

8.1.2 Input/Output Scaling

The analog signal in $DSP-16$ ranges from -10 V to $+10$ V, and the digital representation is from -32768 to 32768 . The input signal flow diagram is shown in Figure 8.2. Before scaling the input speed signal, an experiment must be done to observe signals at each stage.

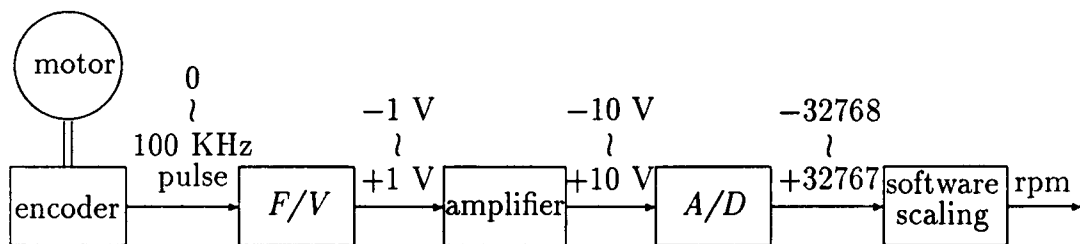


Figure 8.2: Input signal flow diagram.

The speed signal of F/V converter output is from -1 V to $+1\text{ V}$ which is amplified from -10 V to $+10\text{ V}$ to reduce noise corruption in the A/D converter. By recording several motor speeds and their corresponding A/D input voltages as shown in Figure 8.3, a linear regression result was found to be $10\text{ V} \equiv 1412\text{ rpm}$ or $32767 \equiv 1412\text{ rpm}$.

According to the above relationship, each input data divided by 23.2 will be the motor rpm speed. For a 4-pole machine, the rpm speed equals 30 times the electrical frequency. Thus, the scaling factor for frequency versus speed becomes $1\text{ Hz}/696$.

The output signal scaling is easier because the current and voltage conversion ratio has been well-defined in hardware circuit. The current/voltage ratio is $1\text{ A}/0.1\text{ V}$, which corresponds to $1\text{ A}/327.67$ for 10 V full scale.

8.1.3 Handshaking Design

The communication between PC and $DSP-16$ requires handshaking. This means both sides should be available for data transmission. The I/O port 14 on $DSP-16$ is designed for the status of the PC bus. Bit 0 of port 14 tells the status that PC is going to send data while bit 1 tells that PC is ready to receive data. Thus, the handshaking can be simply implemented by checking this port and waiting for the availability of the PC .

The procedure of reading data from the PC can be summarized as

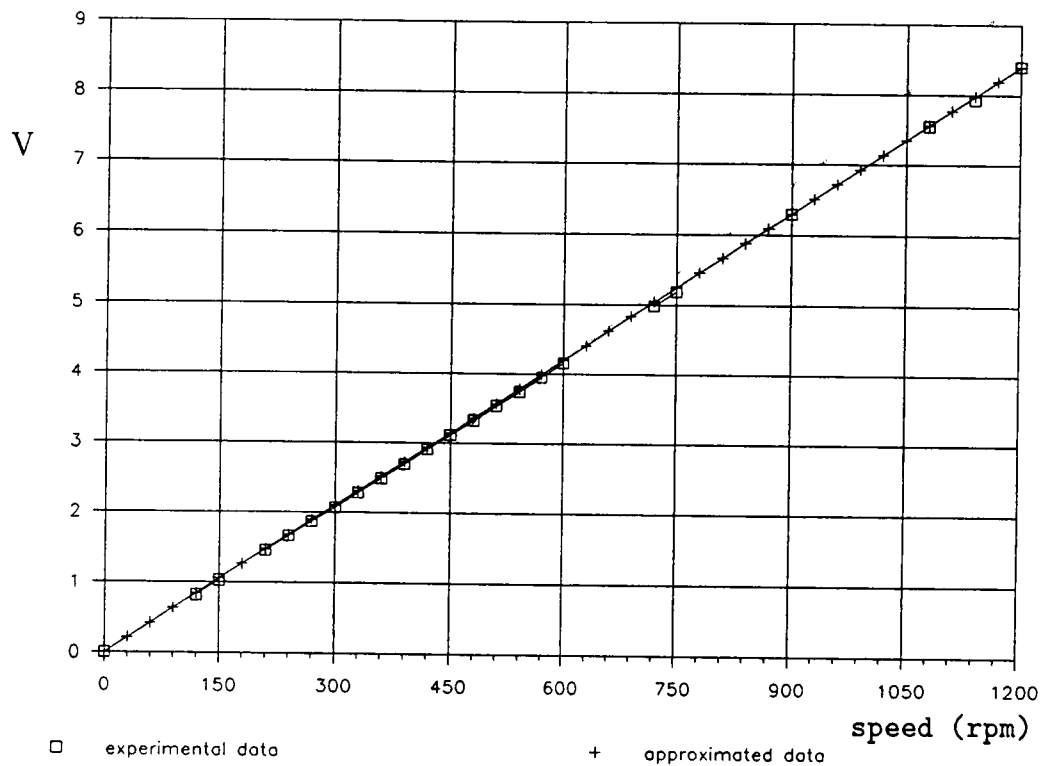


Figure 8.3: Motor speed and *A/D* input signal regression.

1. Check bit 0 of port 14.
2. If it is 0, go back to check.
3. If it is 1, get data from the *PC* by reading port 10.

Similarly, the procedure of sending data to the *PC* can be summarized as

1. Check bit 1 of port 14.
2. If it is not 0, go back to check.
3. If it is 1, send data to the *PC* by output to port 15.

8.2 Speed Controller Design

There are two portions in the induction motor speed control system. The first portion is a speed error detector and a *PI* controller, and the second portion is a vector controller to generate 3-phase sinusoidal currents for inverter reference. Because the TMS320C25 is not efficient in doing floating point computation, the controller is designed in such a way that all numbers are scaled to be suitable for integer manipulation.

8.2.1 Digital Controller Design

As shown in Figure 8.1, there are two digital controllers in the control system. The first is a *PI* controller $G_1(z)$, and the second controller is a low pass filter $G_2(z)$ for feedback speed. The design procedure is to obtain the transfer function in s-domain first, and then transform it into z-domain using bilinear transformation to ensure stability.

The purpose of putting a low pass filter for the feedback speed is to alleviate the noise of the *DSP-16* board ⁵. For a low pass filter at $1/\tau$ corner frequency, the transfer function in s-domain can be represented by

$$G(s) = \frac{1}{1 + \tau s}. \quad (8.1)$$

Assume a sampling period T . Using bilinear transformation

$$s = \frac{2}{T} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right), \quad (8.2)$$

the transfer function in z-domain becomes

$$G(z) = \frac{1 + z^{-1}}{(1 + 2\tau/T) + (1 - 2\tau/T)z^{-1}}. \quad (8.3)$$

⁵The Ariel Corp. has a new product called "*DSP-16 plus*", which is lower noise version of the original *DSP-16*

Assume X is the input of the filter, and Y is the output of the filter. The filter can then be represented by

$$G(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1}}{a_0 + a_1 z^{-1}} \quad (8.4)$$

where

$$a_0 = 1 + 2\tau/T$$

$$a_1 = 1 - 2\tau/T$$

$$b_0 = 1$$

$$b_1 = 1$$

The filter can be easily implemented with the following discrete equation.

$$Y(k) = \frac{X(k) + X(k-1) - a_1 Y(k-1)}{a_0} \quad (8.5)$$

With the corner frequency at 210 KHz and a 15 KHz sampling rate, the coefficients were solved as $a_0 = 8/7$ and $a_1 = 6/7$. The assembly code is economical in using such coefficients because the division can be replaced by shifting. For TMS320C25, total execution time, including data transfer is 1.3 μsec .

The design procedure of the PI controller is similar to the above low-pass filter. In s-domain, the controller is represented by

$$G(s) = K_p + \frac{K_i}{s}. \quad (8.6)$$

The z-domain representation becomes

$$G(z) = \frac{(K_p + K_i T/2) + (-K_p + K_i T/2)z^{-1}}{1 - \tau z^{-1}}. \quad (8.7)$$

The value of K_p and K_i should be varied according to different machines and speed response requirements. Total assembly codes for the PI controller take only 12 steps (or 1.2 μsec).

8.2.2 Vector Controller Design

Because the precise vector control requires machine parameters, this program design simplifies the vector controller by assuming the conditions that 3-phase reference currents I_{abc} are balanced and the angle θ_e^* is increased smoothly. Based on the above assumption, complicated computations can be avoided, and the transformation results can be obtained by a simple square-root routine and a sine wave look-up table.

The d -axis current i_d^* is solved by a function generator to produce flux weakening in high speed regions. The q -axis current i_q^* is directly obtained from the PI controller. The peak amplitude of the 3-phase reference currents I_{abc} can be solved by taking the square root of the sum of two current squares.

The square-root subroutine is designed using Newton's method [9]. The iteration algorithm is given by

$$x(1) = \frac{1}{2}(x(0) + Q/x(0)), \quad (8.8)$$

where Q is the number to be solved, $x(0)$ is the initial guess, and $x(1)$ is the first iteration result. Because each integer division takes 16 steps, the iteration of looping in convergence should be avoided. In this program design, only one iteration is allowed in each subroutine calling. The computation result does not deviate much because the initial guess is very close to the solution.

To search data memory address from the sine wave look-up table in TMS320C25 is very convenient by adding the contents of auxiliary register $AR0$ to the other auxiliary registers. Phase A employs $AR1$ for computational search, while phase B employs $AR5$. The value stored in these two auxiliary registers differentiate 120 degrees. In TMS320C25, the search for each phase takes only two instruction cycles (or 200 $nsec$).

8.3 Monitor Program Design

The monitor program in *PC* serves as *DSP-16* board initialization, sampling rate setting, handshaking with *DSP-16*, speed reference setting, and speed monitoring. The speed reference command is obtained by keyboard input, and the speed monitoring needs graphic interface. Because the whole program regards man-machine interface, it is convenient to implement by C-programming language. The compiler used in this program development is Microsoft C ⁶ version 4.0. Appendix C lists the complete monitor program.

8.3.1 *DSP-16* Board Initialization

The hardware initialization of the *DSP-16* board is different from the previous software initialization. The following tasks are to be done in hardware initialization.

1. Configure what kind of CPU chip, TMS32020 or TMS320C25.
2. Specify *I/O* address.
3. Specify memory page.
4. Specify size of dynamic RAMs.
5. Specify size of static RAMs.
6. Specify the initial sampling rate.
7. Tell the *DSP-16* to load the machine codes of the speed control program.
8. Detect any error conditions.

⁶Trademark of Microsoft Corp.

8.3.2 Sampling Rate Setting

The sampling frequency is specified by the following equation.

$$f_s = \frac{12500}{r} \quad (8.9)$$

The parameter r must be an integer. Selecting 15 KHz sampling, r is 83, and the total assembly code should be executed within 66.7 μ sec or 667 steps per sampling.

8.3.3 Man-Machine Interface

For easy operation, the speed reference is controlled by cursors. Cursors operate up/down for parameter changing, and left/right for speed decreasing and increasing.

However, the cursor operation for speed increment is a ramp input. An additional keyboard operation is designed to get step input of the speed reference. When selecting speed adjustment, the user can directly key-in the required speed in rpm. Both reference and actual speeds are shown on the screen.

The closed-loop control program is named "spccl." Because the program is designed in C-language, the machine pole number and the initial speed can be set by command line input. For example, the command line input

```
spccl 4 300
```

means the machine is 4-pole, and the initial reference speed is 300 rpm.

Under screen interactive condition, the messages on the screen look like

UTK/PEAC INDUCTION MOTOR MOTOR CONTROL SYSTEM
USING ARIEL DSP-16 WITH TMS320C25

Jih-Sheng Lai
UTK/PEAC
7/1/1989

1. No. of poles: 4
2. Desired speed (rpm): 300
3. Actual speed (rpm): 298.5
4. Current (A): 5
5. Quit:

Press a number or cursor to change parameters...

By selecting number 1, the screen will ask the user to key in the number of poles so the program can do the necessary scaling. Selecting number 2 can set up any speed within the limit. If the encoder frequency does not exceed the F/V converter limit, the program allows constant power operation up to 1.5 times of the rated speed. For example, a 4-pole machine can set up the maximum speed at 2700 rpm. When the step increase is requested, the speed control loop will automatically adjust fundamental frequency and dq -axes currents.

The actual speed signal may be corrupted by noises although a low pass filter has been added. However, the variation does not affect the controller performance. The current amplitude means the peak value of reference phase currents. When pressing number 5, the screen will ask the user to confirm again to avoid defective

operation.

As mention before, moving cursors can also change the parameters. For example, when the cursor is at number 1, the color of number 1 will be brighter than regular text, and moving the left-right cursors will decrease or increase the pole number by 2. When moving down the cursor to 2, the color of number 2 will be brighter, and the reference speed can be adjusted by moving cursor left and right. To quit the speed control program, simply move down the cursor to 5 and press the return key.

8.4 Experimental Results of Speed Responses

The speed response of the induction motor depends on gain values of the *PI* controller. As discussed before, the scaling effort can be significantly reduced with careful selection of gain values. With $K_p = 5.655$ and $K_i = 4350$, the acceleration profile is shown in Figure 8.4.

The upper trace shows speed responses accelerating from 300 rpm to 750 rpm. The total rise time is less than 0.5 second. The lower trace shows the induction motor phase current. When running at steady-state, the current magnitude and the fundamental frequency remain constant. During acceleration, the frequency linearly increases 2.5 times, and the q -axis current is varied with the speed difference. The speed response can be easily controlled by changing controller gains or software parameters.

PC Communication Effect Although digitized vector control can be easily implemented by high speed digital signal processors, an acceleration delay appears when *PC* communication disables interrupt. Figure 8.5 shows a communication delay effect during acceleration.

Because the *PC* serves as command input and monitoring functions, a commu-

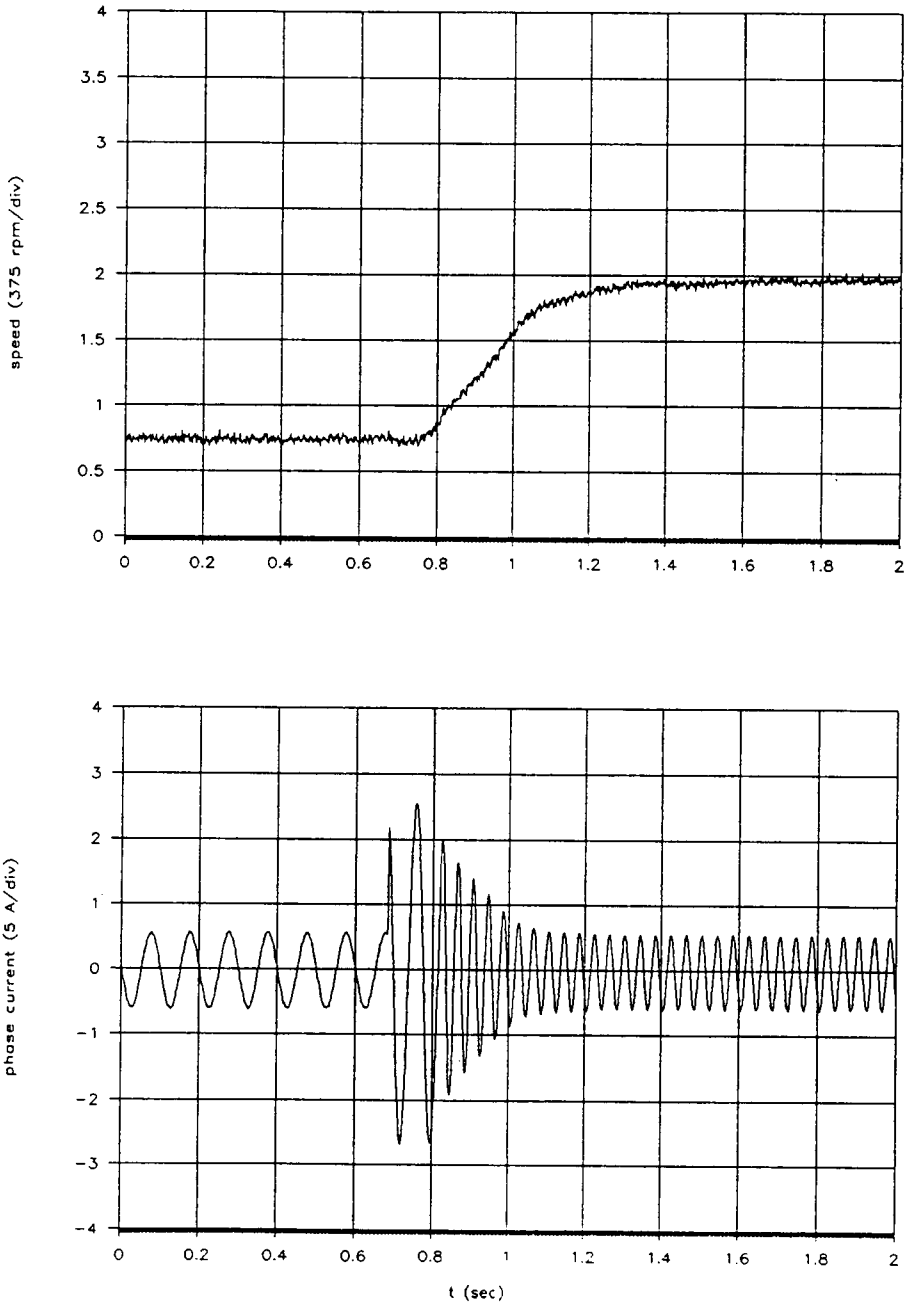


Figure 8.4: Induction motor acceleration profile.

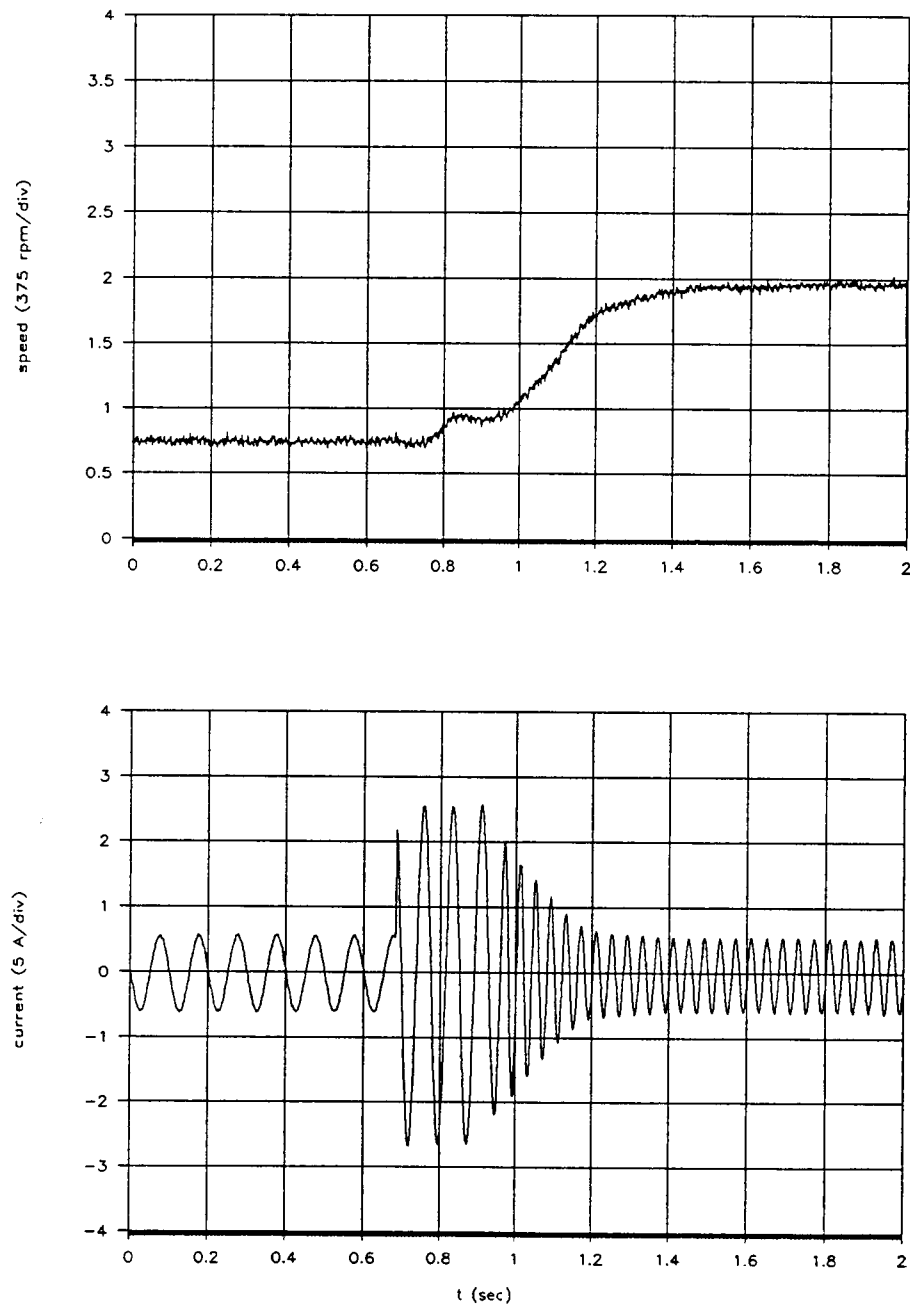


Figure 8.5: Induction motor acceleration profile with *PC* communication delay.

nication between the *PC* and the *DSP* board takes place regularly. If this communication occurs when the motor is accelerating, the program will execute bus data loading but not the controller function. According to Figure 8.5, the fundamental frequency immediately increases after command input but stays the same for two more cycles, then increases again until speed difference reaches zero. A total of 60 msec acceleration delay happens in this case. The situation could be worse when the data bus communicates more frequently.

Such communication delay is not desirable for high performance drives. However, the communication should not exist in a dedicated hardware system. The computer speed control system provides more flexibility in programming and less sensitivity to component variation.

Chapter 9

Conclusion

In this dissertation, a new class of next generation resonant link inverters has been proposed. The voltage overshoot and zero crossing problems have been solved by the proposed current initialization scheme. A new quasi-resonant *dc* voltage notch link inverter has been invented to improve *PWM* switching resolution problems in the resonant link inverter.

Technical accomplishments can be summarized as follow.

1. A quantitative analysis approach of initial current effect to the resonant voltage has been formulated.
2. The voltage overshoot and zero crossing are solved simultaneously.
3. The proposed bidirectional initial current control circuit effectively establishes programmable initial current.
4. A fast algorithm using an analog multiplexer predicts inverter input current for the oncoming cycle.

5. The complete current initialization resonant link circuit, along with the hysteresis band *PWM* inverter induction motor drive, has been simulated using *PC-SIMNON*.
6. A quasi-resonant voltage notching scheme has been developed to eliminate high voltage and high ripple penalties.
7. A minimizing current ripple algorithm has been proposed to create desired voltage notching in the quasi-resonant link circuit.
8. The complete quasi-resonant voltage notching circuit, along with hysteresis-band *PWM* inverter induction motor drive, has been simulated.
9. A 1-HP *IGBT* based resonant inverter has been constructed.
10. A bidirectional current initialization resonant link power circuit has been designed and built.
11. All current and voltage sensors have been designed and successfully tested.
12. An incremental speed sensing circuit has been calibrated.
13. Four different gate drive circuits have been designed and successfully tested.
14. A 3-HP resonant/quasi-resonant inverter has been constructed.
15. A complete control circuit for bidirectional resonant current control has been built in wire-wrapping boards and tested.
16. With zero initial current, the resonant link circuit has been tested and shows a high *Q*-value.
17. A *PC*-based computer control system has been developed.

18. The handshaking between the *PC* and the *TMS320C25* coprocessor has been developed.
19. An interactive speed control program has been written in Microsoft *C* language.
20. A vector control *DSP*-based program has been written in *TMS30C25* assembly language.

When driving the induction motor, the inverter does not introduce any objectionable harmonics or torque pulsation to the motor, and thus the operation condition is extremely quiet and smooth. The only audible sound is from the motor bearing and cooling fan. The resonant link inverter has shown success for high efficiency and high performance drives.

The newly developed quasi-resonant link inverter also shows high efficiency, high performance, and high quality for *ac* machine drives without any significant penalties.

According to the experimental setup and test results, resonant/quasi-resonant link inverter induction motor drives demonstrate the following features.

1. High efficiency – Because there is no switching loss, the efficiency is extremely high. At 1 KW output level, the efficiency has been measured at 94 %.
2. No snubber circuit for inverter switches – The snubber circuit is not necessary for zero voltage switching. This contributes to size savings and efficiency improvement.
3. Minimum cooling requirement – The experiment shows no significant temperature changes on the heat sink because the only heat source is from conduction losses. The cooling fan is absolutely unnecessary for resonant link inverters.

4. No switching noise from power circuits – The inverter switching frequency is far from audible noise, and the power circuit does not generate any switching noise.
5. No audible noises from the motor – The motor runs smoothly without audible frequency harmonics or torque pulsations. The mechanical resonance will not exist in high frequency link inverter drives.
6. Smooth motor acceleration – The high frequency link inverter associated with vector control technique provides very smooth acceleration because of the smooth generating torque.
7. Well-distributed frequency spectrum – Although the output voltage is synthesized by a pulse train, the frequency spectrum is distributed over a high frequency range. If the sinusoidal output voltage is required, the additional passive filter will be of minimum size; however, the output current is pure sine wave with small ripples at the switching frequency.

In actual implementation, some difficulties have been confronted especially with the component sources. Because most high power systems do not use high frequency links, manufacturers do not make high frequency passive components for high voltage high current applications. Similarly, the voltage and current sensing components do not have enough frequency responses for control purposes. The following list shows some of the main difficulties.

1. Resonant inductor – The best material for high frequency inductor is Litz wire, but the Litz wire available on the market is normally small in size. The size equivalent to #12 AWG gage is a typical limit and could only be used for less than 20A applications. Custom design of the Litz wire is possible but too costly.

2. Resonant capacitor – Polypropylene capacitors are best suited for high frequency, high power applications. The problem is the current capability. The Orange drop capacitor from *SB Electronics* provides for only 8 A_{rms} each. Parallel connection is required for this type of capacitor. GE 97F series capacitors provide enough voltage and current ratings but are not available in lower capacitance. Series connection is required for higher resonant frequency and this obviously decreases the Q -value of the resonant link circuit and is not preferred. These additional connections will also cause an increase in size and circuit layout problems.
3. DC current sensor – The high frequency shunt resistor is good for *dc* current sensing but lacks isolation. The Hall probe provides isolation but is very sensitive to temperature variations, possibly causing control failure. The current transformer *CT* also provides isolation but is only good for *ac* current sensing.
4. Circuit noises – The zero voltage crossing generates a series of *TTL* pulses which controls power switches, feeding back significant noises to the analog circuit and causing inaccurate control signals. Circuit noises may affect zero voltage crossing detecting, speed regulation, and reference current corruption, etc. The solution requires elegant circuit layout and decoupling between each circuit stage.

Future work on developing high frequency link inverters can be directed to control circuit simplification, circuit layout, component design optimization, and packaging for commercial applications. Possible candidates of applying resonant link inverters are most industrial drives, automobiles, air crafts, compressors, heat pumps, air conditioning units, uninterruptible power supplies (*UPS*), traction controls, servo systems, and some home appliances.

The application of resonant link inverters not only increases the efficiency of power conversions, but also improves the quality of equipment controls. The problems faced by actual implementation are not difficult to overcome especially with volume production. It can be expected that resonant link inverters will become the main stream of the new generation of power conversions.

REFERENCES

References

- [1] B. J. Baliga, M. Chang, P. Shafer, and M. W. Smith, "The Insulated Gate Transistor (IGT) – A New Power Switching Device," in *Conf. Rec. IEEE IE Annual Meeting*, pp. 794, 1983.
- [2] B. J. Baliga and H. R. Chang, "The MOS Depletion-Mode Thyristor: A New MOS Controlled Bipolar Power Device," *IEEE Electron Device Letters*, Vol. 9, No. 8, pp. 411–413 Aug. 1977.
- [3] B. J. Baliga, *Modern Power Semiconductor Devices*, Wiley-Interscience, New York, N. Y., 1987.
- [4] P. M. Espelage and B. K. Bose, "High Frequency Link Power Conversion," *IEEE Trans. on Ind. Appl.*, Vol. IA-13, No. 5, pp. 388–394, Sept./Oct. 1977.
- [5] B. K. Bose, *Power Electronics and AC Drives*, Prentice-Hall, Englewood Cliffs, N. J., 1986.
- [6] B. K. Bose and P. M. Szczesny, "A microcomputer-Based Control and Simulation of an Advanced IPM Synchronous Machine Drive System for Electric Vehicle Propulsion," in *Conf. Rec. IEEE IE Annual Meeting*, pp. 454–463, 1987.
- [7] B. K. Bose (Ed.), *Microcomputer Control of Power Electronics and Drives*, IEEE Press, New York, N. Y., 1987.

- [8] B. K. Bose and Jih-Sheng Lai, "A Novel Programmable Current Initialization Scheme in Resonant DC link Converter for AC Motor Drives," *Invention Disclosure Letter*, April 19, 1988.
- [9] R. L. Burden and J. D. Faires, *Numerical Analysis*, Prindle, Weber & Schmidt, 3thed., Boston, MA, 1985.
- [10] J. G. Cho, D. Y. Hu, and G. H. Cho, "Three Phase Sine Wave Voltage Source Inverter Using the Soft Switched Resonant Poles," to be presented in 1989 IAS Annual Meeting.
- [11] S. B. Dewan and A. Straughen, *Power Semiconductor Circuits*, John Wiley & Sons, New York, N. Y., 1975.
- [12] D. M. Divan, "Design Considerations for Very High Frequency Resonant Mode DC/DC Converters," *IEEE Trans. on Power Electronics*, Vol. 2, No. 1, pp. 45-54, January 1987.
- [13] D. M. Divan, "The Resonant DC Link Converter - A New Concept in Static Power Conversion," in *Conf. Rec. IEEE IAS Annual Meeting*, pp. 648-656, 1986.
- [14] D. M. Divan and G. Skibinski, "Zero Switching Loss Inverters for High Power Applications," in *Conf. Rec. IEEE IAS Annual Meeting*, pp. 627-634, 1987.
- [15] D. M. Divan, G. Venkataramanan, and R. W. DeDoncker, "Design Methodologies for Soft Switched Inverters," in *Conf. Rec. IEEE IAS Annual Meeting*, pp. 758-766, 1988.
- [16] T. G. Habetler and D. M. Divan, "Performance Characterization of a New Discrete Pulse Modulated Current Regulator," in *Conf. Rec. IEEE IAS Annual Meeting*, pp. 395-405, 1988.

- [17] D. C. Hopkins, M. M. Jovanovic, F. C. Y. Lee, F. W. Stephenson, "Hybridized Off-Line 2-MHz Zero-Current-Switched Quasi-Resonant Converter," *IEEE Trans. on Power Electronics*, Vol. 4, No. 1, pp. 147-154, January 1989.
- [18] M. M. Jovanovic, D. C. Hopkins, and F. C. Y. Lee, "Evaluation and Design of Megahertz-Frequency Off-Line Zero-Current-Switched Quasi-Resonant Converters," *IEEE Trans. on Power Electronics*, Vol. 4, No. 1, pp. 136-146, January 1989.
- [19] J. B. Klaassens and E. J. F. M. Smits, "Series-Resonant AC-Power Interface with an Optimal Power Factor and Enhanced Conversion Ratio," *IEEE Trans. on Power Electronics*, Vol. 3, No. 3, pp. 335-343, July 1988.
- [20] J. B. Klaassens, "DC to AC Series-Resonant Converter System with High Internal Frequency Generating Synthesized Waveforms for Multikilowatt Power Levels," in *Conf. Rec. Power Electronics Specialist*, Gaithersburg, MD, pp. 99-110, 1984.
- [21] Jih-Sheng Lai and B. K. Bose, "An Improved Resonant DC Link Inverter For Induction Motor Drives," in *Conf. Rec. IEEE IAS Annual Meeting*, pp. 742-748, 1988.
- [22] Jih-Sheng Lai and B. K. Bose, "A High Frequency Quasi-Resonant Voltage Notching of a PWM Inverter for AC Machine Drives," *Invention Disclosure Letter*, November 1988.
- [23] Jih-Sheng Lai, *Laboratory report*, July 1989.
- [24] K. H. Liu, R. Oruganti, and F. Lee, "Resonant Switched - Topologies and Characteristics," in *Conf. Rec. Power Electronics Specialist*, pp. 106-116, 1985.

- [25] K. H. Liu and F. Lee, "Zero-Voltage Switching Technique in DC-DC Converters," in *Conf. Rec. Power Electronics Specialist*, pp. 58-70, 1986.
- [26] K. H. Liu, R. Oruganti, and F. Lee, "Quasi-Resonant Converters - Topologies and Characteristics," *IEEE Trans. on Power Electronics*, Vol. 2, No. 1, pp. 62-71, January 1987.
- [27] R. D. Lorenz and D. M. Divan, "Dynamic Analysis & Experimental Evaluation of Delta Modulators for Field Oriented AC Machine Current Regulators," in *Conf. Rec. IEEE IAS Annual Meeting*, pp. 196-201, 1987.
- [28] Y. Murai and T. A. Lipo, "High Frequency Series Resonant DC Link Power Conversion," in *Conf. Rec. IEEE IAS Annual Meeting*, pp. 772-779, 1988.
- [29] R. Oruganti and F. C. Lee, "Resonant Power Processors: Part I - State Plane Analysis," in *Conf. Rec. IEEE IAS Annual Meeting*, pp. 860-867, 1984.
- [30] R. Oruganti and F. C. Lee, "State-Plane Analysis of Parallel Resonant Converter," in *Conf. Rec. Power Electronics Specialists*, pp. 56-73, 1985.
- [31] K. S. Rajashekara, H. Le-Huy, R. Mahfoudhi, and L. A. Dessaint, "Resonant DC Link Inverter-Fed AC Machines Control," in *Conf. Rec. Power Electronics Specialists*, pp. 491-496, 1987.
- [32] R. Rangan, D. Y. Chen, J. Yang, and J. Lee, "Applications of Insulated Gate Bipolar Transistor to Zero-Current Switching Converters," *IEEE Trans. on Power Electronics*, Vol. 4, No. 1, pp. 2-7, January 1989.
- [33] V. T. Ranganathan, P. D. Ziogas, and V. R. Stefanovic, "A Regulated DC-DC Voltage Source Converter Using a High Frequency Link," *IEEE Trans. on Ind. Appl.*, Vol. IA-18, No. 3, pp. 279-287, May/June 1982.

- [34] V. T. Ranganathan, P. D. Ziogas, and V. R. Stefanovic, "Performance Characteristics of High Frequency Links Under Forward and Regenerative Power Flow Conditions," *Conf. Rec. IEEE IAS Annual Meeting*, pp. 831–839, 1983.
- [35] L. Rinehart and A. Plunkett, "The Use of MOSIGTs and MOSFETs," *Proceedings of Motor-Con*, September, 1987.
- [36] H. Elmqvist, K. J. Åström, and T. Schönthal, *SIMNON – User's Guide for MS-DOS Computers*, Version 1.0, 1986.
- [37] G. L. Skibinski and D. M. Divan, "Characterization of Power Transistors Under Zero Voltage Switching," *Conf. Rec. IEEE IAS Annual Meeting*, pp. 493–503, 1987.
- [38] P. K. Sood, T. A. Lipo, and Irving G. Hansen, "A Versatile Power Converter for High-Frequency Link Systems," *IEEE Trans. on Power Electronics*, Vol. 3, No. 4, pp. 383–390, Oct. 1988.
- [39] P. K. Sood and T. A. Lipo, "Power Conversion Distribution System Using a High Frequency AC Link," *IEEE Trans. on Ind. Appl.*, Vol. 24, No. 2, pp. 288–300, March/April 1988.
- [40] Seung K. Sul and T. A. Lipo, "Field Oriented Control of An Induction Machine in a High Frequency Link Power System," in *Conf. Rec. Power Electronics Specialist*, pp. 1084–1090, 1988.
- [41] F. E. Sykes, "Resonant-Mode Power Supplies," *IEEE Spectrum*, Vol. 26, No. 5, pp. 36–39, May 1989.
- [42] W. A. Tabisz, P. M. Gradzki, and F. C. Y. Lee, "Zero-Voltage-Switched Quasi-Resonant Buck and Flyback Converters – Experimental Results at 10 MHz," *IEEE Trans. on Power Electronics*, Vol. 4, No. 2, pp. 194–204, April 1989.

- [43] V. A. K. Temple, "The MCT, a New Class of Power Devices," *IEEE Trans. on Electron Devices*, ED-33, pp. 1609-1615, 1986.
- [44] F. S. Tsai, R. Oruganti, F. C. Lee, "A Novel Control for Bidirectional Power Flow of a Parallel Resonant Converter," *IEEE Trans. on Ind. Appl.*, Vol. IA-23, No. 5, pp. 937-943, Sept./Oct. 1987.
- [45] Ariel, *Operating Mannual for the DSP-16 Data Acquisition Processor*, Ariel Corp., August 1987.
- [46] Exar, *EXAR Databook*, Exar Corp., 1987.
- [47] Ferrocube, *Linear Ferrite Materials and Components*, 7th ed. , Ferrocube, 1984.
- [48] General Instrument, *General Instrument Optoelectronic Products*, General Instrument, 1986/1987.
- [49] International Rectifier, *HEXFET Power MOSFET Designer's Manual*, International Rectifier, 1987.
- [50] International Rectifier, "High-Voltage Bridge Driver - IR2110," *International Rectifier Preliminary Data Sheet No. PD-6.011*, June, 1988.
- [51] Sean Young, "High-Speed, High-Voltage IC Driver for HEXFET or IGBT Bridge Circuits," *International Rectifier Application Note*, 1988.
- [52] IXYS Corporation, *MegaMOS IGTs Preliminary Information*, IXYS Corporation, June, 1987.
- [53] IXYS Corporation, *Latch Resistant CMOS Power MOSFET Drivers Data Sheet*, IXYS Corporation, June, 1988.
- [54] Motorola, *MTH13N50 Designer's Data Sheet*, Motorola Inc., 1984.

- [55] Motorola, *Motorola High-Speed CMOS Logi Data*, Motorola Inc., 1988.
- [56] Motorola, *Motorola Linear Data Book*, Motorola Inc., 1987.
- [57] National Semiconductor, *LM6161/LM6261/LM6361 High Speed Operational Amplifier Preliminary Data Sheet*, National Semiconductor Corp., 1987.
- [58] SB Electronics, *Orange Drop Film Capacitor Product Guide*, S. B. Electronics, 1989.
- [59] Texas Instruments, *TMS320C2x Software Development System User's Guide*, Texas Instruments Inc., 1988.
- [60] Texas Instruments, *ALS/AS Logic Data Book*, Texas Instruments Inc., 1986.
- [61] Texas Instruments, *High Speed CMOS Logic*, Texas Instruments Inc., 1988.
- [62] Texas Instruments, *TTL Logic Data Book*, Texas Instruments Inc., 1988.
- [63] Teledyne, *High Speed Single CMOS Power MOSFET Drivers Data Sheet*, Teledyne, 1988.
- [64] Texas Instruments, *Second-Generation TMS320 User's Guide*, Texas Instruments Inc., 1987.

APPENDICES

Appendix A

SIMNON Simulation Program Listing

```
MACRO remac
SYST reson lp latch1 monol mono2 inv im vct amux recon
STORE ias[im] ibs[im] ics[im] -add
Store iac[inv] ibc[inv] icc[inv] -add
SIMU 0 50e-3 .25e-7 /remacl 2e-6
split 3 1
ashow ias iac -mark
text 'for phase A, 1: actual current -- Ia, 2: reference current -- Ia*'
ashow ibs ibc -mark
text 'for phase B, 1: actual current -- Ib, 2: reference current -- Ib*'
ashow ics icc -mark
text 'for phase C, 1: actual current -- Ic, 2: reference current -- Ic*'
hcopy
END
```

```

CONTINUOUS SYSTEM reson
" -----
"   Resonant DC Link Circuit Simulation
"
"   *   Initial current control scheme *
"
"                                           Jih-Sheng Lai
"                                           Oct. 29, 1987
" -----
"   state variables:
"
"       Vd  -- DC resonant voltage
"       Vcb -- capacitor voltage
"       Ir  -- resonant current
"
"   operation modes:
"
"       mode0 -- regular mode
"       model -- transition from active to reactive
"       mode2 -- transition from reactive to active
" -----

INPUT Id Idref casell stb stb2
OUTPUT case0 casel td Sr Sa Sb
STATE Vd Ir Vcb
DER dVd dIr dVb
TIME t
" -----
"   dVd/dt
" -----
dVd1 = (Ir - Id)/Cr
dVd2 = if (Vd>0.7) then dVd3 else 0
dVd3 = (Ir - Id - (Vd+0.7)/x)/Cr
dVd = if (Sr or Stb) then 0 else dVd1
x = if Vd<0.7 then rl else rs
rs = 5e+6
rl = 0.1
" x1 = if vd>0.7 then 1 else -1
" -----
"   dIr/dt
" -----
dIr1 = (Vs - Vd - Ir*Rr)/Lr
dIr2 = (- Vcb - Ir*Rr)/Lr
dIr = if Sb then dIr2 else dIr1
" -----
"   dVb/dt
" -----
dVb1 = if Vcb<200 then Ir/Cb else (Ir-Vcb/1000)/Cb
dVb2 = if Vcb<200 then 0 else -Vcb/500/Cb
dVb = if Sb then dVb1 else dVb2
" -----
"   control circuit
" -----
case0 = if Vd < 0.7 then 1 else 0
casel = if Id < (Idref+ 10) then 1 else 0
case2 = if Ir < (Idref+ 10) then 1 else 0
casea = casell and case2
caseb = (not casell) and case2
casec = (not casell) and (not case2)
td = ir - idref - 10
casecc = casec and not delay(casec,4e-6)
cased = casea or caseb or casecc
Sr = case0 and (stb or stb2 or casea or caseb or casecc)
Sb = (Sr or stb2) and casecc

```

```
Sa = not Sb
" -----
"   System parameters
" -----
pi = atan(1.0)*4
wr = 2*pi*fr
Zo = Vs/IL
Lr = Zo/wr
Rr = Zo/Q
Cr = 1/(wr*Zo)
Cb = Cr
" -----
"   initial states
" -----
Vcb:150
Vd:1
Ir:0
" -----
"   Constants
" -----
IL:25
Q:75
fr:50e+3
Vs:300
Csn:10e-9
Rsn:100
END
```

```
DISCRETE SYSTEM latch1
"
"  Latch Id and Id* comparator's output
"
INPUT d stb
OUTPUT nql
STATE qd qd1 rd sd
NEW nqd nqd1 nrd nsd
TIME t
TSAMP ts
ts = t+dt
qmd = not(STB and d and qd1)
qnd = not(STB and (not d) and qd)
nsd = not(qmd and rd)
nrd = not(qnd and sd)
d1 = not(nsd and (not STB))
d2 = not(nrd and (not STB))
nqd = not(d1 and qd1)
nqd1 = not(d2 and qd)
nql = nqd
dt = 50e-9
"
"  Initial states for inverter switches
"
rd:0
sd:1
qd:1
qd1:0
END
```

```
DISCRETE SYSTEM monol
INPUT case0
OUTPUT STB
STATE x1
NEW nx1
TIME t
TSAMP ts
ts = t + dt
nx1 = case0
l12 = x1
l13 = not l12
STB = case0 and l13
dt = .25e-6
x1:1
END
```

```
DISCRETE SYSTEM mono2
INPUT case0
OUTPUT STB
STATE x1
NEW nx1
TIME t
TSAMP ts
ts = t + dt
nx1 = case0
l12 = x1
l13 = not l12
STB = case0 and l13
dt = .5e-6
x1:1
END
```

```

DISCRETE SYSTEM inv
"
"   Current controlled inverter
"
INPUT Ia Ib Ic Vd Iac Ibc Icc clk Sr
OUTPUT nal nbl ncl Va Vb Vc IL Iae Ibe Ice
STATE a b c qa qb qc qal qbl qcl ra sa rb sb rc sc
NEW na nb nc nqa nqb nqc nqal nqbl nqcl nra nsa nrb nsb nrc nsc
TIME t
TSAMP ts
ts = t+dt
Iae = Iac-Ia
Ibe = Ibc-Ib
Ice = Icc-Ic
na = if Iae>h then 1 else if Iae<-h then 0 else a
nb = if Ibe>h then 1 else if Ibe<-h then 0 else b
nc = if Ice>h then 1 else if Ice<-h then 0 else c
nal = na
nbl = nb
ncl = nc
qma = not(clk and na and qal)
qna = not(clk and (not na) and qa)
nsa = not(qma and ra)
nra = not(qna and sa)
a1 = not(nsa and (not clk))
a2 = not(nra and (not clk))
nqa = not(a1 and qal) or Sr
nqal = not(a2 and qa) or Sr
qmb = not(clk and nb and qbl)
qnb = not(clk and (not nb) and qb)
nsb = not(qmb and rb)
nrb = not(qnb and sb)
b1 = not(nsb and (not clk))
b2 = not(nrb and (not clk))
nqb = not(b1 and qbl) or Sr
nqbl = not(b2 and qb) or Sr
qmc = not(clk and nc and qcl)
qnc = not(clk and (not nc) and qc)
nrc = not(qmc and rc)
nrc = not(qnc and sc)
c1 = not(nrc and (not clk))
c2 = not(nrc and (not clk))
nqc = not(c1 and qcl) or Sr
nqcl = not(c2 and qc) or Sr
Vqss = Vd*(nqa*2-nqb-nqc)/3
Vdss = Vd*(nqc-nqb)/1.732051
Va = vqss
Vb = -0.5*Vqss-0.8660254*Vdss
Vc = -0.5*Vqss+0.8660254*Vdss
IL = qa*Ia + qb*Ib + qc*Ic
"IL = (Va*Ia + Vb*Ib + Vc*Ic)/Vd
dt = 50e-9
"
"   Initial states for inverter switches
"
ra:1
sa:0
rb:0
sb:1
rc:1
sc:0
qa:0
qal:1
qb:1
qbl:0
qc:0

```

```
qcl:1
a:0
b:1
c:0
"
" Band limit
"
h:.01
END
```

```

CONTINUOUS SYSTEM im
"
" 5-HP Induction Machine model in synchronous reference frame
"
INPUT va vb vc
OUTPUT ias ibs ics wr x3 x4 spd te
STATE iqs ids iqr idr w
DER diqs dids diqr didr dw
TIME t
dw = (Te-TL-B*w)*pole/2/j
spd = 4.77465 * w
diqs = (lr*(Vqs-rs*Iqs)-fse*Ids+lm*rr*Iqr-lrm*w*Idr)/kl
dids = (fse*Iqs+lr*(Vds-rs*Ids)+lrm*w*Iqr+lm*rr*Idr)/kl
diqr = (lm*(rs*Iqs-Vqs)+w*lsm*Ids-Iqr*rr*ls+fre*Idr)/kl
didr = (-w*lsm*Iqs+lm*(Ids*rs-Vds)-fre*Iqr-ls*rr*Idr)/kl
fse = lsr*we - delw*lm
fre = we*lm - delw*lsr
Vqss = (2*Va-Vb-Vc)/3
Vdss = (Vc-Vb)/1.732051
x3 = cos(we*t)
x4 = sin(we*t)
Vqs = Vqss*x3 - Vdss*x4
Vds = Vqss*x4 + Vdss*x3
" Te=(3/2)(P/2)(Lambda_d*iq-Lambda_q*id)
Te = 1.5*pole/2*lm*(idr*iqs-iqr*ids)
iqss = iqs*x3+ids*x4
idss = -iqs*x4+ids*x3
ias = iqss
ibs = -idss*0.8660255-iqss*0.5
ics = -ias - ibs
iqr = iqr*x3+idr*x4
idrs = -iqr*x4+idr*x3
iar = iqr
ibr = -idrs*0.8660255-iqrs*0.5
icr = -iar - ibr
ls = (xm+xs)/wb
lr = (xm+xr)/wb
lm = xm/wb
kl = lsr - lmm
wr = w
lrm = lr*lm
lsm = ls*lm
lsr = ls*lr
lmm = lm*lm
delw = we - w
"
" initial states
"
Iqs:15.1 "peak value
Ids:4.95
Iqr:-14.05 "-14.28
Idr:0.4 "4.085
w:132 "speed = 629.715
"
" constants
"
we:157.07963 "At base speed of 750 rpm"
wb:377
xs:0.4
rs:0.25
rr:0.817
xr:0.4
xm:42.09
J:0.15
B:0.01
TL:20
pole:4
END

```

```
DISCRETE SYSTEM lp
"Small time delay to break loop
INPUT Id1
OUTPUT Id
STATE y1
NEW nyl
TIME t
TSAMP ts
ts = t + dt
nyl = Id1*dt/tau+y1*(1-dt/tau)
Id = y1
dt:1.0e-8
tau:1e-7
END
```

```
CONTINUOUS SYSTEM vct
"Forward vector transformation
INPUT iqs ids x3 x4
OUTPUT iac ibc icc
idssc = ids*x3 - iqs*x4
iqssc = ids*x4 + iqs*x3
iac = iqssc
ibc = -(idssc*1.732051 + iqssc)/2
icc = (idssc*1.732051 - iqssc)/2
END
```

```
Sa = not Sb
" -----
"   System parameters
" -----
pi = atan(1.0)*4
wr = 2*pi*fr
Zo = Vs/IL
Lr = Zo/wr
Rr = Zo/Q
Cr = 1/(wr*Zo)
Cb = Cr
" -----
"   initial states
" -----
Vcb:150
Vd:1
Ir:0
" -----
"   Constants
" -----
IL:25
Q:75
fr:50e+3
Vs:300
Csn:10e-9
Rsn:100
END
```

```
DISCRETE SYSTEM latch1
"
"  Latch Id and Id* comparator's output
"
INPUT d stb
OUTPUT nql
STATE qd qd1 rd sd
NEW nqd nqd1 nrd nsd
TIME t
TSAMP ts
ts = t+dt
qmd = not(STB and d and qd1)
qnd = not(STB and (not d) and qd)
nsd = not(qmd and rd)
nrd = not(qnd and sd)
d1 = not(nsd and (not STB))
d2 = not(nrd and (not STB))
nqd = not(d1 and qd1)
nqd1 = not(d2 and qd)
nql = nqd
dt = 50e-9
"
"  Initial states for inverter switches
"
rd:0
sd:1
qd:1
qd1:0
END
```

```
DISCRETE SYSTEM monol
INPUT case0
OUTPUT STB
STATE x1
NEW nx1
TIME t
TSAMP ts
ts = t + dt
nx1 = case0
l12 = x1
l13 = not l12
STB = case0 and l13
dt = .25e-6
x1:1
END
```

```
DISCRETE SYSTEM mono2
INPUT case0
OUTPUT STB
STATE x1
NEW nx1
TIME t
TSAMP ts
ts = t + dt
nx1 = case0
l12 = x1
l13 = not l12
STB = case0 and l13
dt = .5e-6
x1:1
END
```

```

DISCRETE SYSTEM inv
"
"   Current controlled inverter
"
INPUT Ia Ib Ic Vd Iac Ibc Icc clk Sr
OUTPUT nal nbl ncl Va Vb Vc IL Iae Ibe Ice
STATE a b c qa qb qc qal qbl qcl ra sa rb sb rc sc
NEW na nb nc nga nqb nqc nqal nqbl nqcl nra nsa nrb nsb nrc nsc
TIME t
TSAMP ts
ts = t+dt
Iae = Iac-Ia
Ibe = Ibc-Ib
Ice = Icc-Ic
na = if Iae>h then 1 else if Iae<-h then 0 else a
nb = if Ibe>h then 1 else if Ibe<-h then 0 else b
nc = if Ice>h then 1 else if Ice<-h then 0 else c
nal = na
nbl = nb
ncl = nc
qma = not(clk and na and qal)
qna = not(clk and (not na) and qa)
nsa = not(qma and ra)
nra = not(qna and sa)
a1 = not(nsa and (not clk))
a2 = not(nra and (not clk))
nga = not(a1 and qal) or Sr
nqal = not(a2 and qa) or Sr
qmb = not(clk and nb and qbl)
qnb = not(clk and (not nb) and qb)
nsb = not(qmb and rb)
nrb = not(qnb and sb)
b1 = not(nsb and (not clk))
b2 = not(nrb and (not clk))
nqb = not(b1 and qbl) or Sr
nqbl = not(b2 and qb) or Sr
qmc = not(clk and nc and qcl)
qnc = not(clk and (not nc) and qc)
nsc = not(qmc and rc)
nrc = not(qnc and sc)
c1 = not(nsc and (not clk))
c2 = not(nrc and (not clk))
nqc = not(c1 and qcl) or Sr
nqcl = not(c2 and qc) or Sr
Vqss = Vd*(nga*2-nqb-nqc)/3
Vdss = Vd*(nqc-nqb)/1.732051
Va = vqss
Vb = -0.5*Vqss-0.8660254*Vdss
Vc = -0.5*Vqss+0.8660254*Vdss
IL = qa*Ia + qb*Ib + qc*Ic
"IL = (Va*Ia + Vb*Ib + Vc*Ic)/Vd
dt = 50e-9
"
"   Initial states for inverter switches
"
ra:1
sa:0
rb:0
sb:1
rc:1
sc:0
qa:0
qal:1
qb:1
qbl:0
qc:0

```

```
qcl:1
a:0
b:1
c:0
"
"  Band limit
"
h:.01
END
```

```

CONTINUOUS SYSTEM im
"
" 5-HP Induction Machine model in synchronous reference frame
"
INPUT va vb vc
OUTPUT ias ibs ics wr x3 x4 spd te
STATE iqs ids iqr idr w
DER diqs dids diqr didr dw
TIME t
dw = (Te-TL-B*w)*pole/2/j
spd = 4.77465 * w
diqs = (lr*(Vqs-rs*Iqs)-fse*Ids+lm*rr*Iqr-lrm*w*Idr)/kl
dids = (fse*Iqs+lr*(Vds-rs*Ids)+lrm*w*Iqr+lm*rr*Idr)/kl
diqr = (lm*(rs*Iqs-Vqs)+w*lsm*Ids-Iqr*rr*ls+fre*Idr)/kl
didr = (-w*lsm*Iqs+lm*(Ids*rs-Vds)-fre*Iqr-ls*rr*Idr)/kl
fse = lsr*we - delw*lm
fre = we*lm - delw*lsr
Vqss = (2*Va-Vb-Vc)/3
Vdss = (Vc-Vb)/1.732051
x3 = cos(we*t)
x4 = sin(we*t)
Vqs = Vqss*x3 - Vdss*x4
Vds = Vqss*x4 + Vdss*x3
" Te=(3/2)*(P/2)*(Lambda_d*iq-Lambda_q*id)
Te = 1.5*pole/2*lm*(idr*iqs-iqr*ids)
iqss = iqs*x3+ids*x4
idss = -iqs*x4+ids*x3
ias = iqss
ibs = -idss*0.8660255-iqss*0.5
ics = -ias - ibs
iqr = iqr*x3+idr*x4
ids = -iqr*x4+idr*x3
iar = iqr
ibr = -ids*0.8660255-iqr*0.5
icr = -iar - ibr
ls = (xm+xs)/wb
lr = (xm+xr)/wb
lm = xm/wb
kl = lsr - lmm
wr = w
lrm = lr*lm
lsm = ls*lm
lsr = ls*lr
lmm = lm*lm
delw = we - w
"
" initial states
"
Iqs:15.1 "peak value
Ids:4.95
Iqr:-14.05 "-14.28
Idr:0.4 "4.085
w:132 "speed = 629.715
"
" constants
"
we:157.07963 "At base speed of 750 rpm"
wb:377
xs:0.4
rs:0.25
rr:0.817
xr:0.4
xm:42.09
J:0.15
B:0.01
TL:20
pole:4
END

```

```
DISCRETE SYSTEM lp
"Small time delay to break loop
INPUT Id1
OUTPUT Id
STATE y1
NEW nyl
TIME t
TSAMP ts
ts = t + dt
ny1 = Id1*dt/tau+y1*(1-dt/tau)
Id = y1
dt:1.0e-8
tau:1e-7
END
```

```
CONTINUOUS SYSTEM vct
"Forward vector transformation
INPUT iqs ids x3 x4
OUTPUT iac ibc icc
idssc = ids*x3 - iqs*x4
iqssc = ids*x4 + iqs*x3
iac = iqssc
ibc = -(idssc*1.732051 + iqssc)/2
icc = (idssc*1.732051 - iqssc)/2
END
```

```
DISCRETE SYSTEM amux
"
" Analog multiplexer simulation
"
INPUT a b c ia ib ic STB
OUTPUT Idref
STATE Id
NEW NId
TIME t
TSAMP ts
ts = t+dt
aa = not a
bb = not b
cc = not c
d0 = STB and aa and bb and cc
d1 = STB and aa and bb and c
d2 = STB and aa and b and cc
d3 = STB and aa and b and c
d4 = STB and a and bb and cc
d5 = STB and a and bb and c
d6 = STB and a and b and cc
d7 = STB and a and b and c
NId = if d0 then 0 else if d1 then ic else if d2 then ib else IL2
IL2 = if d3 then -ia else if d4 then ia else if d5 then -ib else IL3
IL3 = if d6 then -ic else if d7 then 0 else Id
Idref = NId
dt = 50e-9
END
```

```

CONNECTING SYSTEM RECON
"Connecting system for Induction Machine drive
TIME t
vd[inv] = vd[reson]
ia[inv] = ias[im]
ib[inv] = ibs[im]
ic[inv] = ics[im]
STB[amux] = STB[monol]
clk[inv] = stb[monol]
stb[reson] = stb[monol]
stb2[reson] = stb[mono2]
casel1[reson] = nql[latch1]
stb[latch1] = stb[monol]
d[latch1] = casel[reson]
case0[monol]=case0[reson]
case0[mono2]=case0[reson]
Sr[inv] = Sr[reson]
Va[im] = Va[inv]
Vb[im] = Vb[inv]
Vc[im] = Vc[inv]
Id[reson] = Id[lp]
Idl[lp] = IL[inv]
Idref[reson] = Idref[amux]
Ia[amux] = Ias[im]
Ib[amux] = Ibs[im]
Ic[amux] = Ics[im]
a[amux] = nal[inv]
b[amux] = nbl[inv]
c[amux] = ncl[inv]
Iqs[vct] = 15      "15.658
Ids[vct] = 5
x3[vct] = x3[im]
x4[vct] = x4[im]
Iac[inv] = Iac[vct]
Ibc[inv] = Ibc[vct]
Icc[inv] = Icc[vct]
END

```

```
MACRO momac
SYST modu lp latch1 monol mono2 inv im vct amux mocon
STORE ias[im] ibs[im] ics[im] -add
Store iac[inv] ibc[inv] icc[inv] -add
SIMU 0 50e-3 .25e-7 /momac1 2e-6
split 3 1
ashow Vd
text '50KHz resonant link system simulation -- resonant DC voltage Vd'
ashow Ir Id Idref -mark
text 'Currents 1: resonant -- Ir 2: inverter -- Id 3: predicted -- Id*'
ashow Vcb
text 'auxiliary capacitor voltage -- Vcb'
hcopy
END
```

```

CONTINUOUS SYSTEM modu
" -----
"   Resonant DC Link Circuit Simulation
"
"   * Initial current control scheme *
"
"                                     Jih-Sheng Lai
"                                     Oct. 29, 1987
" -----
"   state variables:
"
"       Vd -- DC resonant voltage
"       Vcb -- capacitor voltage
"       Ir -- resonant current
"
"   operation modes:
"
"       mode0 -- regular mode
"       model -- transition from active to reactive
"       mode2 -- transition from reactive to active
" -----

INPUT Id Idref casell stb stb2 iae ibe ice
OUTPUT case0 casel td Sr Sa Sb
STATE Vd Ir Vcb
DER dvd dir dvb
TIME t
" -----
"       dvd/dt
" -----
dvd1 = Ic/Cr
dvd = if (Sr or Stb) or model then 0 else dvd1
Ic = Ir - Id
" xl = if vd>0.7 then 1 else -1
" -----
"       dir/dt
" -----
dIr1 = (Vs - Vd - Ir*Rr)/Lr
dIr2 = (- Vcb - Ir*Rr)/Lr
dIr = if model then Id else if Sb then dIr2 else dIr1
" -----
"       dvb/dt
" -----
dVb1 = if Vcb<200 then Ir/Cb else (Ir-Vcb/1000)/Cb
dVb2 = if Vcb<200 then 0 else -Vcb/500/Cb
dVb = if Sb then dVb1 else dVb2
" -----
"   mode definition
" -----
model = if Sr then 0 else if ((Ic>-.1 and Ic<0.1) and Sc) then 1 else 0
dIabc = Iae*Iae + Ibe*Ibe + Ice*Ice
Sc = if dIabc < 15 then 1 else 0
" -----
"   control circuit
" -----
case0 = if Vd < 0.7 then 1 else 0
casel = if Id < (Idref+ 10) then 1 else 0
case2 = if Ir < (Idref+ 10) then 1 else 0
casea = casell and case2
caseb = (not casell) and case2
casec = (not casell) and (not case2)
td = ir - idref - 10
casecc = casec and not delay(casec,3.5e-6)
cased = casea or caseb or casecc

```

```
Sr = case0 and (stb or stb2 or casea or caseb or casecc)
Sb = (Sr or stb2) and casecc
Sa = not Sb
" -----
"   System parameters
" -----
pi = atan(1.0)*4
wr = 2*pi*fr
Zo = Vs/IL
Lr = Zo/wr
Rr = Zo/Q
Cr = 1/(wr*Zo)
Cb = Cr
" -----
"   initial states
" -----
Vcb:100
Vd:1
Ir:0
" -----
"   Constants
" -----
IL:25
Q:75
fr:50e+3
Vs:300
Csn:10e-9
Rsn:100
END
```

```

CONNECTING SYSTEM MOCON
"Connecting system for Induction Machine drive
TIME t
vd[inv] = vd[modu]
ia[inv] = ias[im]
ib[inv] = ibs[im]
ic[inv] = ics[im]
STB[amux] = STB[monol]
clk[inv] = stb[monol]
stb[modu] = stb[monol]
stb2[modu] = stb[mono2]
case1[modu] = nql[latch1]
stb[latch1] = stb[monol]
d[latch1] = case1[modu]
case0[monol] = case0[modu]
case0[mono2] = case0[modu]
Sr[inv] = Sr[modu]
Va[im] = Va[inv]
Vb[im] = Vb[inv]
Vc[im] = Vc[inv]
Id[modu] = Id[lp]
Idl[lp] = IL[inv]
Idref[modu] = Idref[amux]
Ia[amux] = Ias[im]
Ib[amux] = Ibs[im]
Ic[amux] = Ics[im]
a[amux] = nal[inv]
b[amux] = nbl[inv]
c[amux] = ncl[inv]
Iqs[vct] = 15 "15.658
Ids[vct] = 5
x3[vct] = x3[im]
x4[vct] = x4[im]
Iac[inv] = Iac[vct]
Ibc[inv] = Ibc[vct]
Icc[inv] = Icc[vct]
Iae[modu] = Iae[inv]
Ibe[modu] = Ibe[inv]
Ice[modu] = Ice[inv]
END

```

Appendix B

TMS320C25 Assembly Program Listing

```

;*****
;
;   SPCCL.ASM  --  UTK/PEAC Induction Motor Speed Control System
;                  Using Ariel DSP-16 board as I/O, filtering,
;                  and signal processing.
;
;
;                  Jih-Sheng Lai
;                  UTK/PEAC
;                  7/1/89
;
;*****
;The program runs in real-time at the sample rate of 15 KHz.
;
;OVERVIEW:
; 1) The input to the ADC is the motor speed signal
; 2) The reference speed is provided by PC keyboard input
; 3) A simple PI controller yields torque and speed commands
; 4) d-axis current is generated by a function generator
; 5) The sine wave is obtained by a look-up table
; 6) Data buffer output written to DAC
; 7) The input speed from ADC is filtered by a low pass filter
;
;***** I/O port addresses *****
;
DRAMDATA:    EQU    8        ;DRAM data buffer read/write port
ADDRESS:     EQU    9        ;DRAM addresses (Row, then Column)
RDHOST:      EQU    10       ;read data sent from PC
REFRESH:     EQU    11       ;Enable buffer refresh
INTHOST:     EQU    12       ;interrupt the PC
WRDAC:       EQU    13       ;write data to the DAC
RDSTAT:      EQU    14       ;read from the status port
WRHOST:      EQU    15       ;write data to the PC

;
;***** Dedicated ON-CHIP REGISTERS *****
;
; not necessary to claim again
;
;DRR:        EQU    0        ;A/D Input register (series port)
;DXR:        EQU    1        ;Data transmit register
;TIM:        EQU    2        ;Timer
;PRD:        EQU    3        ;Period register
;IMR:        EQU    4        ;Interrupt mask register
;GREG:       EQU    5        ;Global memory allocation register

```

```

;
;***** Constants declaration *****
;
CNSTQU:      EQU      25      ;Constant Torque speed range
                        ; (750 rpm)
CURADD:      EQU      1638    ;d-axis current addition constant
                        ; 5 A, (327.7 == 1 A)
CURMUL:      EQU      20      ;d-axis current multiply constant
GAINC1:      EQU      25      ;Current-loop GAIN 1
GAINC2:      EQU      -24     ;Current-loop GAIN 2
GAINSP:      EQU      3       ;SPeed-loop GAIN
MINIDS:      EQU      327     ;Minimum Ids (= 1 A_peak)
SPADD:       EQU      210     ;Speed addition constant
                        ; (offset of A/D)
SPCNST:      EQU      180     ;Speed constant
                        ;(noise level of A/D)
SPLOW:       EQU      5       ;Lower speed increment limit
                        ; (150 rpm)
SPMUL:       EQU      696     ;Speed multiplication constant
SPUP:        EQU      45      ;Upper speed increment limit
                        ;(1350 rpm)

;
;***** Data RAM ON-CHIP Block B2 (60 -- 7F) *****
;
                ORG      060H  ;Start of on-board data RAM
AITEMP:      DS      1       ;Channel A Input interrupt constant
AOTEMP:      DS      1       ;Channel A Output interrupt constant
BITEMP:      DS      1       ;Channel B Input interrupt constant
BOTEMP:      DS      1       ;Channel B Output interrupt constant
BUFBOT:      DS      1       ;pointer to BOTtom of waveform BUFFer
BUFTOP:      DS      1       ;pointer to TOP of waveform BUFFer
DFPC:        DS      1       ;Data From PC
DIVR:        DS      1       ;Divisor constant
DSPD:        DS      1       ;Difference SPeed
DUMMY:       DS      1       ;DUMMY register
IDS:         DS      1       ;d-axis current (field component)
EK:          DS      1       ;Error speed (k)
EK1:         DS      1       ;Error speed (k-1); delay of EK
IQS:         DS      1       ;q-axis current (torque component)
IQSUP:       DS      1       ;* Upeer limit of Iqs
OUTAMP:      DS      1       ;OUTput signal peak AMPlitude
OUTBFA:      DS      1       ;Channel A OUTput BuFFer
OUTBFB:      DS      1       ;Channel B OUTput BuFFer
OUTTMP:      DS      1       ;OUTput signal TeMPorary amplitude
PCST:        DS      1       ;PC bus SStatus
RPM:         DS      1       ;Actual speed from A/D converter
ROOT:        DS      1       ;Squart root location
SPINC:       DS      1       ;INCrement of reference SPeed
                        ; for look-up table

```

```

SPINCO:      DS      1      ;INCrement of command SPeed from PC
SPREF:      DS      1      ;ReFerence SPeed
TEMP:       DS      1      ;TEMPorary constant
TMPSGN:     DS      1      ;SiGN constant
XN:         DS      1      ;Square root iteration parameter
XK:         DS      1      ;Filter input
XK1:        DS      1      ;Delay of XK
YK:         DS      1      ;Output of the low-pass filter

;
;***** Reset & interrupt vectors *****
;
      ORG      0
RESET:  B      INIT      ;Reset vector

      ORG      4
INTA:   B      OUTA      ;Channel A interrupt service routine

      ORG      6
INTB:   B      OUTB      ;Channel B interrupt service routine

      ORG      26
RINT:   B      ININT     ;ADC interrupt service routine

;
;***** Initialize 320C25 *****
;
      ORG      32
INIT:   DINT
      ROVM
      LDPK      0      ;Data memory pointer
      SSXM      ;Set sign extension mode
      SPM       0      ;Product register output no shift
      FORT      0      ;Configure for 16 bit serial
                        ; data word
      SXF       ;Set output channel flag
      LACK      080H    ;Data memory from 8000H
                        ; (32768-40886)
      SACL      GREG    ;Write to global memory register
;
;Load constants
;
      ZAC      ;Load 0
      SACL     XK      ;zero speed initially
      SACL     XK1     ;zero speed initially
      SACL     YK      ;
      LACK     80      ;Load 80
      SACL     DIVR    ; to the divisor
      LALK     1638    ;Load 32767/20
      SACL     OUTAMP   ;Output peak amplitude is 5 A

```

```

        LALK    2640                ;8.06 A
        SACL    IQSUP              ;Upper limit of Iqs is 12 A
;
        ZAC
        SACL    TEMP                ;Initialize output to zero
        OUT     TEMP,WRDAC          ;
        RXF
        OUT     TEMP,WRDAC          ;
        LALK    08180H+03AD4H      ;Setup pointers to 8180H+15060-1
        SACL    BUFTOP             ;
        LALK    08180H             ;
        SACL    BUFBOT             ;
        LACK    10
        SACL    SPINC              ;Set initial speed at 300 rpm
        LAR     ARO,SPINC
        LRLK    ARI,08180H         ;set up initial address for
                                   ; Channel A output
        LRLK    AR5,08180H+0139CH ;set up initial address for
                                   ; Channel B output
        B       PROG
;
; Some subroutines for PC communication
;
; -- WAIT4IT wait for data from PC
;
WAIT4IT:IN     PCST, RDSTAT        ;Get content of the status port
        BIT     PCST, 15          ;Check bit 0
        BBZ     WAIT4IT          ;if not, wait
        IN      DFPC, RDHOST      ;get data
        ZALS    DFPC             ;put it in ACC
        RET
;
;.....Calls from table VEC_TAB.....
;
; -- S_SPD Set up speed reference
;
S_SPD:  CALL    WAIT4IT
        SACL    SPINC0
        LT      SPINC0
        MPYK    SPMUL
        PAC
        SACL    SPREF
        RET
;
; -- S_AMP Set current Amplitude
;
S_AMP:  CALL    WAIT4IT
        SACL    OUTAMP
        RET

```

```

;
; -- SEND data to host
;
SEND:  IN      PCST, RDSTAT      ;Is host ready for data?
      BIT      PCST, 15-1      ;Check bit 1 for host status
      BBNZ     SEND            ;Loop 'till ready
      OUT      YK, WRHOST      ;Output speed to host
      RET      ;Return

;
; -- Send current AMplitude to host
;
SD_AMP: IN      PCST, RDSTAT      ;Is host ready for data?
      BIT      PCST, 15-1      ;Check bit 1 for host status
      BBNZ     SEND            ;Loop 'till ready
      OUT      OUTAMP, WRHOST    ;send current amplitude to host
      RET      ;Return

;
; -- DO_CLR Clear output
;
DO_CLR: LRLK    AR3, 08180H
      LRLK    AR4, 15060
      ZAC
      LARP    AR3
CLR:   SACL    **+, 0, AR4
      BANZ    CLR, **-, AR3
      RET

VEC_TAB: DW PROG1, S_SPD, S_AMP, DO_CLR, SEND, SD_AMP

;
;***** Foreground loop *****
;
PROG:  EQU     $

PROG0: CALL    WAIT4IT
      ZALS    DFPC
      ADLK    VEC_TAB, 0
      TBLR    TEMP
      ZALS    TEMP
      CALA
      B       PROG0

;
;feedback control loops
;
PROG1: LAC     RPM              ;Input of A/D
      ABS                    ;Get rid of negative noise

```

```

;
;Filter for speed input at 1.07 KHz
;
    SACL    XK                ;Lower address for X(k)
    ZAC
    LT      XK1              ;T = X(k-1)
    MPYK    7                ;P = 7*X(k-1)
    LTD     XK              ;Acc = P, X(k-1) = X(k)
    MPYK    7                ;P = 7*X(k)
    LTA     YK              ;Acc = 7*(X(k) + X(k-1))
    MPYK    6                ;P = 6*Y(k-1)
    SPAC
    SFR
    SFR
    SACL    YK              ;Y(k-1) = Y(k)
;
;Error block
;
; -- current loop error block
;
    ZALS    SPREF           ;Calculation for speed loop
    SUB     YK              ;speed error
    SACL    TMPSGN          ;Store sign of speed error
    ABS
    SACL    DSPD            ;Make speed difference positive
                           ;get absolute value for DSPD
;
; -- speed loop error block
;
    ZALS    SPINC0          ;Check the command speed
    SUB     SPINC           ;command error
    BGZ     INC             ;err > 0
    BZ      DONE            ;err = 0, no change
;
;reference command speed is less than actual speed
;reduce SPINC
;
DEC:    LAC     SPINC        ;yes! Decrease SPINC by one
        SUBK    1           ;Subtract one from ACC
        SACL    SPINC        ;Store ACC to SPINC
;put above lower limit
        SUBK    SPLOW        ;Compare to the lower limit
        BGEZ    DONE        ;Done if greater than lower limit
        LACK    SPLOW        ;Load lower limit to ACC
        SACL    SPINC        ;Store ACC to SPINC
;
;reference command speed is greater than actual speed
;increase SPINC
;

```

```

INC:      LAC      SPINC      ;Yes! Increase SPINC by GAINSP
          ADDK     GAINSP
          SACL     SPINC      ;Store ACC to SPINC
          SUB      SPINC0     ;SPINC < SPINC0?
          BLEZ     ULIMIT     ;Yes! check the limit
          LAC      SPINC0     ;No! SPINC = SPINC0
          SACL     SPINC      ;Store ACC to SPINC
;put under upper limit
ULIMIT:   LAC      SPINC
          SUBK     SPUP       ;Compare to the upper limit
          BLEZ     DONE       ;Done if less than upper limit
          LACK     SPUP       ;Load upper limit to ACC
          SACL     SPINC      ;Store ACC to SPINC

;end calculation
DONE:     LAR      ARO,SPINC

;
;current loops
; 1: flux weakening
;
CURT:     LAC      SPINC      ;Get SPINC
          SUBK     CNSTQU     ;Within constant torque region
          BLEZ     CNSTQ      ;Yes! setup Ids constant
          SACL     TEMP       ;No! store the difference
          LALK     CURADD      ;Add maximum current into ACC
          LT       TEMP       ;Calculation for flux weakening
          MPYK     CURMUL      ;Multiply current constant
          SPAC      ;Subtract P register from ACC
          SACL     IDS        ;Solved Ids
          B        CURT2
CNSTQ:    LALK     CURADD      ;constant d-axis current
          SACL     IDS

;
; 2: torque component
;
CURT2:    ZALS     DSPD       ;Get Speed difference
          RPTK     15
          SUBC     DIVR       ;divided by a constant
          SACL     EK
          LAC      IQS        ;Acc=Iqs
          LT       EK1        ;T=E(k-1)
          MPYK     GAINC2     ;P=b1*E(k-1)
          LTD      EK         ;Acc=Iqs+b1*E(k-1), E(k-1)=E(k)
          MPYK     GAINC2     ;P=b0*E(k)
          APAC
          ABS
          SACL     IQS        ;Store to q-axis current
          SUB      IQSUP      ;Compare upper limit of Iqs
          BLEZ     IQLE       ;Iqs <= Iqs^
          LAC      IQSUP
          SACL     IQS        ;Store to q-axis current

```

```

; 3: sqrt(Ids^2+Iqs^2)
IQLE:  SQRA    IDS        ;P = Ids^2
      ZAC      ;Acc = 0
      SQRA    IQS        ;P = Iqs^2; Acc = Ids^2
      APAC     ;Acc = Ids^2 + Iqs^2
      CALL    SQRT       ;Approximated
      LAC      ROOT      ;
      NEG
      SACL     OUTAMP

;
;Interrupt service
;
      LACK     22        ;Enable RINT, AINT, BINT
      SACL     IMR       ;Set interrupt mask register
      EINT     ;Finishing computation.

;
;   Background check
;
PROG2: IN      PCST,RDSTAT ;All the time check PC status port
      BIT     PCST,15    ;Test bit 0
      BBZ     PROG2      ;Continue testing if nothing found
      DINT
      IN      DFPC,RDHOST
      ZALS    DFPC
      ADLK    VEC_TAB,0
      TBLR    TEMP
      ZALS    TEMP
      CALA
      EINT
      B       PROG2

;
;***** Input data interrupt service routine *****
;
;ADC interrupt service routine
;
ININT: BIOZ    QUIT1      ;Skip if CHANNEL B
      LAC      DRR        ;ADC to accumulator
      ADDK     SPADD      ;Add the offset of A/D converter
      SACL     RPM        ;Get speed
      EINT     ;Enable interrupts
      RET      ;Return
QUIT1: EINT
      RET

```

```

;
;***** Output interrupt service routines *****
;
;output interrupt service routine
;
;      Output to Channel A
;
OUTA:  SXF
      LARP      AR1
      LT        *0+           ;get a sample
      MPY       OUTAMP        ;multiply by attenuation factor
      PAC              ;-->ACC
      SACH      AOTEMP,1      ;save it
      OUT       AOTEMP,WRDAC   ;output the sample to DAC
;
OUT2A: SAR      AR1,AOTEMP     ;get the current output pointer
      ZALS      AOTEMP        ;in the ACC
      SUBS      BUFTOP        ;compare to upper limit of buffer
      BGEZ      PTRB          ;past limit, must correct
      EINT              ;else is OK, so re-enable ints
      RET              ;and return to background code
;
PTRA:  ADD      BUFBOT        ;add pointer to bottom of buffer
      SACL      AOTEMP
      LAR       AR1, AOTEMP    ;reload AR1
      EINT              ;re-enable ints
      RET              ;and return to background code
;
;      Output to channel B
;
OUTB:  RXF
      LARP      AR5
      LT        *0+           ;get a sample
      MPY       OUTAMP        ;multiply by attenuation factor
      PAC              ;-->ACC
      SACH      BOTEMP,1      ;save it
      OUT       BOTEMP,WRDAC   ;output the sample to DAC
;
OUTB2: SAR      AR5, BOTEMP    ;get the current output pointer
      ZALS      BOTEMP        ;in the ACC
      SUBS      BUFTOP        ;compare to upper limit of buffer
      BGEZ      PTRB          ;past limit, must correct
      EINT              ;else is OK, so re-enable ints
      RET              ;and return to background code
;
PTRB:  ADD      BUFBOT        ;add pointer to bottom of buffer
      SACL      BOTEMP
      LAR       AR5, BOTEMP    ;reload AR5
      EINT              ;else re-enable ints
      RET              ;and return to background code

```

```

;
;Square root subroutine
;   input X, output ROOT
;   X(n) = Ids
;
SQRT:  RPTK    15                ; X/X(n)
      SUBC    IDS              ;
      SACL    TEMP            ;Get rid of high bytes' gabbage
      LAC     TEMP            ;
      ADD     IDS              ;ACC = X(n)+X/X(n)
      SFR     IDS              ;Divide by 2
      SACL    ROOT            ;X(n+1) = (X(n) + X/X(n))/2
DONESQ: RET                    ;Return of the square root routine
;
;wave table pointers:
;
W_PTRS: DW      08180H
;
      NCLUD   BIGSIN.TBL      ;include wave tables in assembly
;
;end of program!
;

```

Appendix C

Monitor Program Listing

```

/*****
*
*   SPCCCL -- Speed Control by Closed-Loop Method
*           for induction motor speed control
*           at the sampling rate 15KHz
*
*
*                               Jih-Sheng Lai
*                               UTK/PEAC
*                               August 2, 1989
*
*****/
/*
* Please check following variables before running the motor
*
* -- change "divr", divisor of rpmf, to adopt different
*   encoder resolution. (23.2 for Teledyne Encoder)
* -- change DEFSR to correct sample rate
*   50 for 25 KHz
*   83 for 15 KHz
*   100 for 12.5 KHz
*/

/* To be compiled under Microsoft C version 4.0 Medium Model */

#include <stdio.h>
#include <conio.h>
#include <ctype.h>
#include <stdlib.h>
#include <dos.h>
#include <time.h>
#include <string.h>
#include <all1mcm.h>
#include <dsp16.h>

/* Declare Constants, Defaults and Macros */
#define FALSE 0
#define NO 0
#define TRUE 1
#define YES 1
#define RVRS 1
#define ENTER 0x0D
#define BACKSPACE 8
#define ESCAPE 27
#define UPARROW 72
#define DOWNARROW 80
#define LEFTARROW 75
#define RIGHTARROW 77
/* Define ASCII key values */

```

```

#define HILITE (REDB + BRIGHT) /* Highlight attribute */
#define DEFPAL GRE_BLK /* Default pallet green on black */
#define BLU_BLU 0x11 /* Blue on Blue */
#define GRN_GRN 0x22 /* Green on Green */
#define BRT_RED 0xC4 /* Bright Red on Red */
#define BLK_BLK 0 /* Black on Black (for erasure) */
#define BLUB BLU_BLK
#define CYAB CYA_BLK
#define REDB RED_BLK
#define MAGB MAG_BLK
#define YELB YEL_BLK
#define WHIB WHI_BLK
#define DEFSR 83 /* Sample rate divider for 15 KHz sampling */
#define DEFFILE "SPCCL.HEX"
/* Default TMS32025 program to load */
#define DSPINT 0 /* DSP-16 hardware interrupt */
#define SRAM 16 /* Size of program RAM in kwords */
#define ROWCOL 0x01ff /* # of row/col bits (9 if 256k, 10 if 1M) */
#define ESC 0x1b

#define max(a,b) (((a) > (b)) ? (a) : (b))
#define min(a,b) (((a) < (b)) ? (a) : (b))

struct cfg {
    int portbase; /* the base port address for DSP-16 */
    int mempage; /* memory page */
    int intnum; /* the interrupt used */
    int d_ram; /* how much d-ram on board: 0=256K, 1=1M */
    int s_ram; /* how much s-ram on board: 0 =4K, 1=16K */
    int expanded_io; /* 0 if 32025 ports are 8..15.
                     /* 1 if expansion board installed
    int new_chip; /* 0 if no 32025. 1 if 320C25
} configuration;

/* Definition of commands */

#define BEGIN 0
#define SETFREQ 1
#define SETAMP 2
#define QUIT 3
#define GETRPM 4
#define GETAMP 5

/* Function Declarations */

void akey(), dkey(), lkey(), rkey(), ukey(), qkey(),
    pol(), spd(), hlp(), send_it(), herald(), idle();

int get_dsp_cfg(), get_it(), find_cfg();

double max_delay;

```

```

char hexfile[14] = DEFFILE;

/* misc globals */

FILE *setup_outfile, *setup_infile;

/* Declare Static Variables */

int pole = 4, speed = 300, freq, freq0, spdinc=10;
float ampf, amp, rpmf, divr=23.2;
int exitflg = FALSE;
int al80_row, al80_col;

/*****
main(argc,argv)
int argc;
char *argv[];
{
    unsigned row, col;
    int i = 0, ic;
    time_t tnow;
    struct tm *tmnow;

    alvcn(3);          /* Set video mode ( 25 x 80 ) */
    alvcl();           /* Clear screen */
    alcof();           /* Turn cursor off */
    clib();            /* Clear keyboard input buffer */
    herald();          /* Say hello */

    if(argc >= 3){ /* argc at least == 3, so there is at
                    least 2 parameters. This means user
                    has known how to run the program */
        pole = atof(argv[1]);
        speed = atof(argv[2]);
    }
    if(!find_cfg()) /* Ariel-16 configuration check */
        goto quit_sampler;

    sysinit();        /* Initialize hardware and software */
    send_it(GETRPM);
    rpmf = get_it();
    send_it(SETFREQ);
    send_it(spdinc);
    send_it(GETAMP);
    ampf = get_it();
    amp = -ampf/327.67; /* scaled by 32767/100 == 1 A */
    send_it(BEGIN);
    waitkp();          /* Show init status before continuing */
    efx_screen();      /* Draw screen */

```

```

loop: while ( !exitflg )
(
    /* Keyboard interface */

    if ( kbhit() )
    (
        /* Here if key hit */
        i = toupper(getch());
        /* scan for special keys */
        switch ( i ) {
            case DOWNARROW: /* move down cursor */
                dkey();
                break;
            case UPARROW: /* move up cursor */
                ukey();
                break;
            case LEFTARROW: /* Decrease speed */
                lkey();
                break;
            case RIGHTARROW: /* increase speed */
                rkey();
                break;
            case 'Q':
                qkey(); /* Quit application - return to DOS */
                break;
            case 'A': /* enter output current amplitude*/
                /* call fcn to get the value from the user */
                al80_row = 16;
                akey();
                break;
            case '1':
                al80_row = 10;
                pol();
                break;
            case '2':
                al80_row = 12;
                spd();
                break;
            case '3':
                hlp();
                break;
            case '4':
                al80_row = 16;
                akey(); /* amplitude adjustment */
                break;
            case '5':
                qkey(); /* Quit application - return to DOS */
                break;
            case ENTER:
                if ( al80_row == 18 )
                    qkey();
                else if ( al80_row == 16 )
                    akey();
                else if ( al80_row == 10 )
                    pol();
                else if ( al80_row == 12 )
                    spd();
        }
    )
}

```

```

        break;
    default:
        break;
    }
}
else{
/*
time(&tnow);
tmnow = localtime(&tnow);
if(tmnow->tm_sec == ((tmnow->tm_sec / 2) * 2)){
*/
send_it(GETRPM);
rpmf = get_it();
send_it(BEGIN);
alscr( 14, 32);
printf("%6.1f", rpmf/divr);

send_it(GETAMP);
ampf = get_it();
amp = -ampf/327.67;
alscr( 16, 30);
printf("%6.2f", amp);
/*
for (ic = 0; ic !=2; ic++)
idle();
}
*/
}
}
alvcl(); /* Clear screen */
quit_sampler: alvcn(3); /* Set 80 column video mode */
printf ("\n\nDone.\n");
alcon(); /* Turn on cursor */
exit(0);
}
/*****
void herald() /* Print banner */
{
int i;
printf("\n\n\n\t ");
printf("UTK/PEAC Induction Motor Closed-Loop Speed Control System");
printf("\n\t Using DSP-16 with TMS320C25 \n");
printf("\n\t Fix sampling rate at 15 KHz \n");
printf("\n\t Jih-Sheng Lai ");
printf("\n\t UTK/PEAC ");
printf("\n\t 5/1/1989 \n\n\n\n");
abortkp( 1000 );
}

```

```

sysinit()    /* Initialize DSP-16 hardware */
{
    printf("\n Setting address of DSP-16. \n");
    DN = DSETA;
    DBASEADR = configuration.portbase;
    DMPAGE = configuration.mempage;
    if ( ! dsp16( &dcall ) )
        derror( dcall );
    printf("Initializing DSP-16 hardware.\n");
    DN = DINIT;
    DBUFSIZE = configuration.d_ram ? 256 : 1024;
    DPRAMSIZE = SRAM;
    DFILENAME = hexfile;
    if ( ! dsp16( &dcall ) )
        derror( dcall );
    printf("%d words uploaded.\n", DWORDSUP );
    printf("Setting sample rate to %f kHz.", 1250.0/DEFPSR);

    DN = DSSR;
    DINSR = DEFPSR;
    DOUTSR = DEFPSR;
    DSHWIDTH = 0;
    DSYNC = TRUE;                      /* enable synch */
    if ( ! dsp16( &dcall ) )
        derror( dcall );
    max_delay = configuration.d_ram ? 10485.74 : 2621.44;
    send_it( configuration.d_ram );
}

derror( q )
struct parms q;
{
    int i;
    printf("Error %d in DSP-16 call %d!\n", q.retval, q.fun );
    printf("#1: %d #2: %d #3: %d #4: %d", q.p2, q.p3, q.p4, q.p5 );
    waitkp();
}

/*****Special key service routines*****/

void dkey()    /* DOWN arrow key service routine */
{
    alhil(al80_row, 7, DEFPAL);
    al80_row = al80_row + 2;
    if (al80_row > 18)
        al80_row = 10;
    alhil(al80_row, 7, HILITE);
}

```

```
void ukey()                /* UP arrow key service routine */
{
    alhil(al80_row, 7, DEFPAL);
    al80_row = al80_row - 2;
    if (al80_row < 10)
        al80_row = 18;
    alhil(al80_row, 7, HILITE);
}

void lkey()                /* Auto-decrement rate */
{
    int ic;
    if (al80_row == 12){
        speed = speed - 30;
        spdinc = spdinc - 1;
        if (speed < 30){
            speed = 30;
            spdinc = speed*pole/120;
        }
        else{
            send_it(SETFREQ);
            send_it(spdinc);
            send_it(BEGIN);
        }
        alscr( 12, 30);
        printf("%6d", speed);
        for (ic = 0; ic != 3; ic++)
            idle();
    }
    else if (al80_row == 16){
        amp = amp - .1;
        if (amp < .1)
            amp = .1;
        else{
            send_it(SETAMP);      /* the command */
            send_it(-(int)(32767.0 * amp/100.0)); /* the value */
            send_it(BEGIN);
        }
        alscr( 16, 30);
        printf("%6.2f", amp);
        for (ic = 0; ic != 5; ic++)
            idle();
    }
}
```

```

void rkey()                                /* Auto-increment rate */
{
    int ic;

    if (al80_row == 12){
        speed = speed + 30;
        spdinc = spdinc + 1;
        if (speed > 10800/pole){
            speed = 10800/pole;
            spdinc = speed*pole/120;
        }
        else{
            send_it(SETFREQ);
            send_it(spdinc);
            send_it(BEGIN);
        }
        alscr( 12, 30);
        printf("%6d", speed);
        for (ic = 0; ic != 3; ic++)
            idle();
    }
    else if (al80_row == 16){
        amp = amp + .1;
        if (amp > 50.0)
            amp = 50.0;
        else{
            send_it(SETAMP);          /* the command */
            send_it(-(int)(32767.0 * amp/100.0)); /* the value */
            send_it(BEGIN);
        }
        alscr( 16, 30);
        printf("%6.2f", amp);
        for (ic = 0; ic != 3; ic++)
            idle();
    }
}

void akey()
{
    int row, col, ic;

    albox( 15, 36, BLUB + BRIGHT, 30, 2 );
    alpts( 16, 38, REDB + BRIGHT, "Enter Amplitude (0..50)");
    alscr( 16, 30);
    scanf("%f", &amp);          /* send the new amplitude to the DSP-16 */
    if (amp > 50.0)
        amp = 50.0;
    else{
        send_it(SETAMP);          /* the command */
        send_it(-(int)(32767.0 * amp/100.0)); /* the value */
        send_it(BEGIN);
    }
}

```

```

    for ( row = 15; row != 18; row++ )
        for ( col = 30; col != 70; col++ )
            alptc( row, col, DEFPAL, SPACE );
    alscr( 16, 30);
    printf("%6.2f", amp);
    for (ic = 0; ic != 3; ic++)
        idle();
}

void qkey()                /* QUIT service routine */
{
    unsigned row, col;
    char c;
    albox( 17, 36, BLUB + BRIGHT, 30, 2 );
    alpts( 18, 38, REDB + BRIGHT, "Are you sure? Y or N" );
    c = getch();
    if ( toupper( c ) == 'Y' ){
        send_it(QUIT);
        send_it(BEGIN);
        exitflg = TRUE;
    }
    else
        for ( row = 17; row <= 19; row++){
            for ( col = 36; col != 70; col++ )
                alptc( row, col, DEFPAL, SPACE );
        }
}

void pol()                /* POLE change service routine */
{
    int i;
    unsigned row, col;
    al80_row = 10;

                                /* scan for special keys */

    albox( 9, 36, BLUB + BRIGHT, 32, 2 );
    alpts( 10, 38, REDB + BRIGHT, "Enter Pole value (2,4,6,...)" );
    alscr( 10, 30);
    scanf("%d", &pole);  /* send the new amplitude to the DSP-16 */

    for ( col = 36; col != 70; col++ ){
        for ( row = 9; row <= 11; row++)
            alptc( row, col, DEFPAL, SPACE );
    }

    /* print on screen */
    alscr( 10, 30 );
    printf("%6d", pole);
}

```

```

void spd()                                /* SPEED change service routine */
{
    int i, ic;
    unsigned row, col;

    albox( 11, 36, BLUB + BRIGHT, 32, 2 );
    alpts( 12, 38, REDB + BRIGHT, "Enter reference SPEED (rpm) " );
    alscr( 12, 30);
    for ( col = 30; col != 35; col++ )
        alptc( row, col, DEFPAL, SPACE );
    scanf("%d", &speed);
    spdinc = speed*pole/120;
    if (speed > 10800/pole){
        speed = 10800/pole;
        spdinc = 90; /* 90 Hz maximum */
    }
    else if (speed < 120/pole){
        speed = 120/pole;
        spdinc = 1; /* 1 Hz minimum */
    }
    send_it(SETFREQ); /* the command */
    send_it(spdinc);
    send_it(BEGIN);
    for (ic = 0; ic != 5; ic++)
        idle();
    for ( row = 11; row <= 13; row++){
        for ( col = 36; col != 70; col++ )
            alptc( row, col, DEFPAL, SPACE );
    }
    alscr( 12, 30 );
    printf("%6d", speed);
}

void hlp()                                /* HELP service routine */
{
    int i;
    unsigned row, col;
    for ( col = 36; col != 70; col++ )
        alptc( 14, col, DEFPAL, SPACE );
    albox( 13, 36, BLUB + BRIGHT, 32, 3 );
    alpts( 14, 38, REDB + BRIGHT, "Not adjustable! ");
    alpts( 15, 38, REDB + BRIGHT, "Press any key to continue..." );
    clib();
    while ( TRUE ){
        if ( kbhit() ){
            i=getch();
            break;
        }
    }
    for ( row = 13; row <= 16; row++){
        for ( col = 36; col != 70; col++ )
            alptc( row, col, DEFPAL, SPACE );
    }
}

```

```

void idle()
{
    int i;
    for ( i = 1; i != 1000; i++ )
        alptc( 12, 5, DEFPAL + BLINK, SPACE );
}

/*****
void send_it( word )
    /* handshake with DSP-16 and send a word of data */
int word;
{
    DN = DSEND;          /* set up cmd to write a word of data */
    dcall.p2 = word;     /* data to send is in P2 element */
    if( !dsp16( &dcall ) ) /* call DSEND fcn and check for error */
        derror( dcall ); /* call error reporter if it occurred */
}

int get_it()
{
    DN = DREAD;
    if ( ! dsp16( &dcall ) )
        derror( dcall );
    return ( dcall.p2 );
}
*****/

/*****Keyboard service routines*****/

clib()          /* Clear keyboard input buffer */
{
    int i;
    while ( kbhit() )
        i=getch();
}

waitq()
{
    int i;
    clib();
    while ( !kbhit() ){
        if ( kbhit() ){
            i = getch();
            break;
        }
    }
    clib();
}

```

```

waitkp()                                /* Wait for key press */
{
    int i;
    printf("\n\nPress any key to continue...\n" );
    clib();
    while ( TRUE ){
        if ( kbhit() ){
            i=getch();
            break;
        }
    }
}

abortkp(timeout)                        /* Abort if no keypress */
int timeout;
{
    register i;
    for ( i = 0; i < timeout; i++){
        if ( kbhit() ){
            i = getch();
            break;
        }
    }
}
/*****

/***** Screen drawing routines *****/
efx_screen()
{
    int i, length, height;
    alvcl();                                /* Clear screen */
    aldbx(1, 1,YELB, 72, 22);              /* Draw a big double line box */
    alpts(3,14,DEFPAL,"UTK/PEAC INDUCTION MOTOR SPEED CONTROL SYSTEM");
    alpts(4,20,DEFPAL,"USING ARIEL DSP-16 WITH TMS320C25");
                                           /* Print TITLE */
    alpts(6,45,DEFPAL, "Jih-Sheng Lai" );
    alpts(7,45,DEFPAL, "UTK/PEAC" );
    alpts(8,45,DEFPAL, "4/1/1989" );
    alpts(10,7,DEFPAL, "1. No. of poles:      " );
    alpts(12,7,DEFPAL, "2. Desired speed (rpm): " );
    alpts(14,7,DEFPAL, "3. Actual speed (rpm): " );
    alpts(16,7,DEFPAL, "4. Current (A):       " );
    alpts(18,7,DEFPAL, "5. Quit:              " );
    alpts(20,7,CYAB,"Press a number or cursor to change parameter ...");
    alscr( 10, 30);
    printf("%6d", pole);
    alscr( 12, 30);
    printf("%6d", speed);
    alscr( 14, 32);
    printf("%6.1f", rpmf/divr);
    alscr( 16, 30);
    printf("%6.2f", amp);
    al80_row = 12;
    alhil(al80_row, 7, HILITE);
}

/*****/

```

```

/* read configuration file 'DSP16.CFG' */
int find_cfg() /* SETUP */
(
    char c;

    if ( ! get_dsp_cfg() ) {
        alvcl();
        alpts(4,10, DEFPAL + BRIGHT + BLINK, "ERROR !!!" );
        alpts(8,0, DEFPAL, "DSP16.CFG NOT FOUND IN CURRENT DIRECTORY");
        alpts(10,0, DEFPAL, "DAS cannot operate without this file.");
        alpts(12,0, DEFPAL, "Copy DSP16.CFG to current directory.");
        alpts(14,15, DEFPAL, "OR" );
        alpts(16,0, DEFPAL, "Run DSPBUG setup to create DSP16.CFG.");
        alpts(20,0, DEFPAL, "Press any key to return to DOS." );
        while (!kbhit() )
            ;
        {
            return(0);
        }
    }
)

int get_dsp_cfg()
/* returns zero if file not found or if reading error */
(
    int tst;

    if((setup_infile = fopen("DSP16.CFG","rt"))==NULL)
        return(0);
    tst=fscanf(setup_infile,
        "DSP16_CFG_BEGIN! %4x,%4x,%4x,%4x,%4x,%4x,%4.4X,END",
        &configuration.portbase, &configuration.mempage,
        &configuration.intnum, &configuration.d_ram,
        &configuration.s_ram, &configuration.expanded_io,
        configuration.new_chip);
    fclose(setup_infile);
    tst = (tst==EOF) ? 0 : 1;
    return(tst);
)
/*****

```

Vita

Jih-Sheng Lai was born in Taichung, Taiwan, Republic of China on October 6, 1953. He attended city elementary and middle schools and graduated from the Provincial Taichung Industrial High School in June 1971. In the fall of 1971, he entered National Taiwan Normal University, and in June 1975 he received a Bachelor degree in Industrial Education specialized in Electrical Power Engineering. After graduate, he accepted a teaching position at the Provincial Taichung Industrial High School. He started two-year military duty in the Air Force from the fall of 1976. After military service, he entered the Graduate Institute of National Taiwan Normal University and received a Master degree in Industrial Education in June 1980.

From July 1980 he was employed by Mingchi Institute of Technology and served as an Instructor and the Chairman of Electrical Engineering Department. After three years, he received a two-year fellowship from National Science Council to study abroad. He entered the University of Tennessee, Knoxville in fall 1983 and received a Master of Science degree in Electrical Engineering in June 1985. He worked as a computer programmer, a teaching assistant, a research assistant during his study. In the fall of 1986, he received an one-year instructorship. When the University of Tennessee started the Power Electronics program in the fall of 1987, he joined a research project sponsored by the Power Electronics Applications Center, and worked for his Ph.D. dissertation. He received a Ph.D. degree in Electrical

Engineering in December 1989.

The author received Wattec 1985 award with \$2000 scholarship. He has 2 patents pending and 14 publications including 2 books. He is a member of the Institute of Electrical and Electronics Engineers and two honor societies – Phi Kappa Phi and Eta Kappa Nu. Dr. Lai has accepted a research position with Tennessee Center for Research and Development.