

**Voltage-Source Inverter Output
Waveform Compensation Using Adaptive
Intelligent Control**

by

Lemuel Gregory Barnes III

Dissertation submitted to the Faculty of the Virginia Polytechnic
Institute and State University in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in
Electrical Engineering

APPROVED:

Krishnan Ramu

Krishnan Ramu, Chairman

H.F. VanLandingham
H.F. VanLandingham

Dan Chen

D.Y. Chen

C.E. Nunnally
C.E. Nunnally

Michael P. Deisenroth
M.P. Deisenroth

May, 1994

Blacksburg, Virginia

Voltage-Source Inverter Output Waveform Compensation Using Adaptive Intelligent Control

by

Lemuel Gregory Barnes III

Dr. Krishnan Ramu, Committee Chairman

Department of Electrical Engineering

(ABSTRACT)

A single-layer neural network-based voltage compensation technique which generates minimum-distortion sinusoidal output voltages from a three-phase PWM inverter used for uninterruptible power supplies (UPS) is described. The proposed compensation technique is implemented in a microprocessor-based controller constructed in the stationary d-q frame where the controller sampling rate is twice the inverter switching frequency. The structure of a feed-forward artificial neural network connects network inputs and outputs through multiple linear or nonlinear neuron models, and processes these input/output data associations in a parallel distributed manner. Network inputs in the form of UPS load voltage commands and load current feedback are propagated forward in the network each controller sampling period generating the inverter output voltage commands, the network outputs, which are converted to three-phase inverter switching signals using the space vector PWM waveform generation process. Each controller sampling period, the network weights are modified by the controller learning process with the objective of

minimizing the cost function $\frac{1}{2} \cdot [\epsilon_{dv}^2 + \epsilon_{qv}^2]$ where ϵ_{dv} and ϵ_{qv} are the measured d- and q-axis UPS load voltage errors. Once the cost function is minimized, the neural network input/output mapping approximates the PWM inverter/output filter circuit inverse transfer characteristics.

Three neural network-based controller configurations were studied via computer simulation:

- i.) A controller with one hidden layer and hyperbolic tangent squashing functions (six hidden layer nodes used) and the conventional backpropagation learning algorithm utilized.
- ii.) A controller without hidden layers (single-layer network) and the conventional backpropagation learning algorithm utilized.
- iii.) A controller without hidden layers and a modified backpropagation learning algorithm utilized.

The third controller configuration was incorporated into the design of an experimental controller. Three-phase UPS system experiments and simulations using this control technique were run using balanced passive, unbalanced passive, and nonlinear loads at inverter switching frequencies consistent with the use of GTO technology. Using the same load models and system parameters, three-phase UPS system simulations were run utilizing average load voltage control and the repetitive controller technique described in [3]. Results obtained from these simulations were served as a basis for evaluating the performance of the neural network-based controller.

Improvements associated with the use of the neural network-based controller concept are:

- i.) The core controller design is independent of UPS system parameters such as the inverter modulation and switching frequencies, and output filter design and component values. Characteristic of nonlinear controllers, the neural network-based controller is relatively insensitive to system parameter variations and modeling errors. Unlike conventional nonlinear controllers, the neural network-based controller can operate effectively at relatively low sampling rates or inverter switching frequencies.
- ii.) Load voltage waveform phase delays associated with controller operation (computational delays) and load voltage waveform synthesis are inherently compensated. Though conventional linear controllers can be designed to compensate for such delays, controller design is based on inverter/filter systems parameters which are particular to the UPS application and subject to uncertainties as in the case of output filter component tolerances or accurate prediction of load current feedback.

The application of the neural network-based inverter controller concept is intended for high-power three-phase UPS systems where inverter switching frequencies are reduced and a broad range of UPS loads and output filter designs are encountered.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Krishnan Ramu, for his advice and perspective with regard to this thesis.

I would like to thank my family for their comfort and support. In particular, I am very grateful to my great-aunt, Mrs. Alice Bragg. Without her generosity and understanding, it would not have been possible for me to pursue this activity.

Finally, I wish to acknowledge the upperclassmen in my cadet squadron who during my freshman year at Virginia Tech gave me the intestinal fortitude required to leave the security of employment in the face of hard economic times to pursue this goal.

Table of Contents

I.	INTRODUCTION	1
1.1	Uninterruptible Power Supplies	1
1.2	Survey of PWM Waveform Generation Techniques	13
1.2.1	Carrier PWM	14
1.2.2	PWM Regular Sampling Methods	15
1.2.2.1	Symmetric Regular Sampling	17
1.2.2.2	Asymmetric Regular Sampling	18
1.2.3	Modulation Waveform Modification Using Added Third-Harmonic Distortion	21
1.2.4	Carrier PWM Inverter Output Waveform Spectral Characteristics	22
1.2.5	Optimized Switching Strategies	23
1.2.6	Space Vector PWM	27
1.2.7	Space Vector PWM Implementation	33
1.2.8	An Evaluation of Sinusoidal PWM Techniques	39
1.2.9	Software Inverter Simulations Using Open-Loop Carrier and Space Vector Sinusoidal PWM Techniques	55
1.3	Inverter Switching Distortion	61
1.3.1	Inverter Leg Switching with Positive-Sense Load Current	63
1.3.2	Inverter Leg Switching with Negative-Sense Load Current	66

Table of Contents

1.3.3 Fundamental Voltage Modification By Inverter
Switching Distortion 69

1.4 Summary of PWM Waveform Generation Techniques 72

1.5 Survey of Closed-Loop Inverter Control Techniques 74

1.5.1 Nonlinear Controllers 85

1.6 Adaptive Intelligent Inverter Control Strategy 87

1.7 Scope 101

II. Adaptive Intelligent Inverter Control Method 103

2.1 Introduction 103

2.2 Neurocontroller Configurations and Learning
Techniques 115

2.3 The Inverter Neurocontroller Technique 126

2.3.1 Multi-Layer Neurocontroller Steady-State
Operation 137

2.3.2 Single-Layer Neurocontrollers 154

2.3.3 Modified Backpropagation Learning Process 161

2.4 Experimental Neurocontroller Design 167

III. SIMULATION AND EXPERIMENTAL RESULTS 175

3.1 Overview of Experimental and Simulation Studies 175

3.2 Initial Evaluation of Neurocontroller Operation 179

3.2.1 Nonlinear Load Simulations 183

3.2.2 Comparisions with Conventional Inverter Control
Methods 189

3.3 Evaluation of An Experimental Neurocontroller 201

Table of Contents

3.3.1	Balanced Resistive Load Experiment Results	203
3.3.2	Unbalanced Resistive Load Experiment	222
3.3.3	Nonlinear Load Experiment Results	233
3.3.4	Dynamic Control Experiments	248
IV.	THESIS SUMMARY	254
4.1	Overview of Neurocontroller Application	254
4.2	Contributions and Conclusions	260
4.3	Suggestions for Future Study	263
Appendix 1: GTO and Diode Models for Inverter Leg Circuit		
	Simulation	265
A1.1	Overview of GTO Operation	265
A1.2	GTO Simulation	267
A1.2.1	GTO Off-State Model	269
A1.2.2	GTO Turn-On Model	269
A1.2.3	GTO On-State Model	270
A1.2.4	GTO Turn-Off Model	270
A1.2.5	GTO Reverse Blocking State	274
A1.2.6	GTO Simulation Software	274
A1.3	Overview of Power Diode Operation	275
A1.3.1	Diode Model Capacitance	279
A1.3.2	Diode Model Resistance	281
A1.3.3	Power Diode Simulation Software	281
Appendix 2: The Inverter Leg Model		283
Appendix 3: Three-Phase Load Models		293

Table of Contents

A3.1	Passive Inductive Load Model	293
A3.2	Nonlinear Load Model	299
A3.3	Load Model Function	307
Appendix 4: Simulation Software Implementation of PWM		
	Techniques	308
A4.1	Carrier PWM Methods	308
A4.2	Three-Phase Carrier PWM Gating Generation	310
A4.3	Space Vector Modulation	311
A4.4	Summary	317
Appendix 5: Inverter/Load Simulation Software		318
A5.1	Simulation Software Input	318
A5.2	GTO Model Data Structure	319
A5.3	Diode Model Data Structure	322
A5.4	Inverter Model Data Structure	323
A5.5	Load Model Data Structure	324
A5.6	Inverter/Load Simulation Data Structure	325
A5.7	Simulation Start-up	326
A5.8	Inverter/Load Simulation Process	328
A5.9	The Program Control Routine	332
Appendix 6: The Modified Backpropagation Learning Algorithm .		334
A6.1	Preliminaries	334
A6.2	The Conventional Backpropagation Algorithm	339
A6.3	The Modified Backpropagation Algorithm	341
Appendix 7: Output Filter Circuit Analysis		345

Table of Contents

Appendix 8: Experimental Neurocontroller Design 352

A8.1 Design Philosophy 352

A8.2 PWM Inverter Switching Signal Generation 354

A8.3 Space Vector PWM Implementation 362

A8.4 SVP Software Timer Interrupts 369

A8.5 Analog-to-Digital Conversion Circuitry 369

A8.6 Analog Interface Circuitry 373

A8.7 ANCP Function 376

A8.9 NCP Function 381

A8.10 Experimental Neurocontroller Design Summary 383

A8.11 A General Procedure for Single-Layer Neurocontroller
 Implementations 385

References 388

Vita 393

List of Figures

1.1: Uninterruptible Power Supply	2
1.2: Output Filter Circuit	4
1.3: Ideal Inverter Output Pulse Waveform	16
1.4: Inverter Leg Circuit	29
1.5: Inverter Leg Circuit Switching Logic Table	30
1.6: Inverter Switching States	31
1.7a: Inverter Output Line-to-Line Voltage Waveform Spectral Information, Asymmetric Regular Sampled Sinusoidal PWM Technique, Magnitudes Normalized with Respect to the Inverter DC Power Supply Voltage	41
1.7b: Inverter Output Line-to-Line Voltage Waveform Spectral Information, Asymmetric Regular Sampled Sinusoidal PWM with 25% Added Third-Harmonic Distortion, Magnitudes Normalized with Respect to the Inverter DC Power Supply Voltage	42
1.7c: Inverter Output Line-to-Line Voltage Waveform Spectral Information, Space Vector Sinusoidal PWM Technique, Magnitudes Normalized with Respect to the Inverter DC Power Supply Voltage	43
1.7d: Inverter Output Line-to-Line Voltage Waveform Spectral Information, Optimized Fixed-Pattern Sinusoidal PWM Technique, Magnitudes Normalized with Respect to the Inverter DC Power Supply Voltage	44
1.8a: Plot of $\sqrt{\sum_{k=1}^{\infty} v_{in}^2(k \cdot f_m)/k}$ (x10 Scale) Verses the Modulation Index Command Given to the Asymmetric Regular Sampled Sinusoidal PWM Waveform Generation Process	46
1.8b: Plot of $\sqrt{\sum_{k=1}^{\infty} v_{in}^2(k \cdot f_m)/k}$ (x10 Scale) Verses the Modulation Index Command Given to the Asymmetric Regular Sampled Sinusoidal PWM Waveform Generation Process with 25% Added Third-Harmonic Distortion	47
1.8c: Plot of $\sqrt{\sum_{k=1}^{\infty} v_{in}^2(k \cdot f_m)/k}$ (x10 Scale) Verses the Modulation Index Command Given to the Space Vector Sinusoidal PWM Waveform Generation Process	48
1.9: Positive-Sense Current Distortion	65

List of Figures

1.10: Negative-Sense Current Distortion	68
1.11: Three-Phase Linear Controller	83
1.12: Repetitive Controller Application	84
1.13: VSS Inverter Output Current Controller	88
1.14a: Two-Layer ANN	95
1.14b: One-Layer ANN	96
2.1: Neuron Unit	104
2.2: Hopfield Neural Network	106
2.3: Plant Emulator	108
2.4: General Neurocontroller Configuration	109
2.5: Direct Adaptive Control Configuration	117
2.6: Hybrid Control Scheme	118
2.7: Direct Inverse Control Configuration	119
2.8: Indirect Learning Architecture	121
2.9: Indirect Adaptive Control Configuration	122
2.10: Initial Neurocontroller Configuration	129
2.11: Phase A Line-to-Line Load and Corresponding Sinusoidal Reference Voltages Obtained from the Balanced Passive Load Simulation where $f_s = 1260$ Hz., $f_m = 60$ Hz, and $M = 0.75$	138
2.12: Phase A Line-to-Line Load Voltage Spectral Information, Balanced Passive Load Simulation, $f_s = 1260$ Hz., $f_m =$ 60 Hz., and $M = 0.75$	139
2.13: Nonlinear Load Simulation Phase A Line-to-Line Load and Reference Voltages, $f_s = 540$ Hz., $f_m = 60$ Hz., and $M =$ 0.75	141
2.14: Nonlinear Load Simulation Phase A Line-to-Line Load Voltage Spectral Analysis $f_s = 540$ Hz., $f_m = 60$ Hz., and $M = 0.75$	142
2.15: Nonlinear Load Simulation Phase A Line-to-Line Load and Reference Voltages $f_s = 900$ Hz., $f_m = 60$ Hz., and $M =$ 0.75	143
2.16: Nonlinear Load Simulation Phase A Line-to-Line Load Voltage Spectral Analysis $f_s = 900$ Hz., $f_m = 60$ Hz., and $M = 0.75$	144
2.17: Nonlinear Load Simulation with Average Load Voltage Controller, Phase A Line-to-Line Load and Reference Voltages, $f_s = 540$ Hz, $f_m = 60$ Hz., and $M = 0.75$	146
2.18: Nonlinear Load Simulation with an Average Load Voltage Controller, Phase A Line-to-Line Load Voltage Spectral Information, $f_s = 540$ Hz., $f_m = 60$ Hz., and $M = 0.75$..	147

List of Figures

2.19: Nonlinear Load Simulation Using a Repetitive Load Voltage Controller, Phase A Line-to-Line Load and Reference Voltages, $f_s = 540$ Hz., $f_m = 60$ Hz., and $M = 0.75$	148
2.20: Nonlinear Load Simulation Using a Repetitive Load Voltage Controller, Phase A Line-to-Line Load Voltage Spectral Information, $f_s = 540$ Hz., $f_m = 60$ Hz., and $M = 0.75$	149
2.21: Phase A Line-to-Line Load Voltage Spectral Information Obtained from the Balanced Resistive Load Simulation, Conventional Backpropagation Learning Process Enabled.	159
2.22: Q-Axis Load Voltage Spectral Information, Balanced Resistive Load Simulation, Modified Backpropagation Process Enabled	168
2.23: Final Network Version	169
2.24: Neurocontroller Circuit Block Diagram	170
3.1: Phase A Line-to-Line Load and Corresponding Sinusoidal Reference Voltages Obtained from the Balanced Passive Load Simulation where $f_s = 1260$ Hz., $f_m = 60$ Hz., and $M = 0.75$	180
3.2: Phase A Line-to-Line Load Voltage Spectral Information, Balanced Passive Load Simulation, $f_s = 1260$ Hz., $f_m = 60$ Hz., and $M = 0.75$	181
3.3: Nonlinear Load Current Generated by An Uncontrolled Three-Phase Rectifier	184
3.4: Nonlinear Load Simulation Phase A Line-to-Line Load and Reference Voltages, $f_s = 540$ Hz., $f_m = 60$ Hz., and $M = 0.75$	185
3.5: Nonlinear Load Simulation Phase A Line-to-Line Load Voltage Spectral Analysis $f_s = 540$ Hz., $f_m = 60$ Hz., and $M = 0.75$	186
3.6: Nonlinear Load Simulation Phase A Line-to-Line Load and Reference Voltages $f_s = 900$ Hz., $f_m = 60$ Hz., and $M = 0.75$	187
3.7: Nonlinear Load Simulation Phase A Line-to-Line Load Voltage Spectral Analysis $f_s = 900$ Hz., $f_m = 60$ Hz., and $M = 0.75$	188

List of Figures

3.8: Nonlinear Load Simulation with Average Load Voltage Controller, Phase A Line-to-Line Load and Reference Voltages, $f_s = 540$ Hz, $f_m = 60$ Hz., and $M = 0.75$	190
3.9: Nonlinear Load Simulation with an Average Load Voltage Controller, Phase A Line-to-Line Load Voltage Spectral Information, $f_s = 540$ Hz., $f_m = 60$ Hz., and $M = 0.75$..	191
3.10: Nonlinear Load Simulation Using a Repetitive Load Voltage Controller, Phase A Line-to-Line Load and Reference Voltages, $f_s = 540$ Hz., $f_m = 60$ Hz., and $M = 0.75$	192
3.11: Nonlinear Load Simulation Using a Repetitive Load Voltage Controller, Phase A Line-to-Line Load Voltage Spectral Information, $f_s = 540$ Hz., $f_m = 60$ Hz., and $M = 0.75$	193
3.12: D-Axis Load Voltage (Top Trace) Verses D-Axis Load Voltage Reference (Bottom Trace), Balanced Resistive Load, Controller Operating in an Open-Loop Mode	204
3.13: Q-Axis Load Voltage (Top Trace) Verses Q-Axis Load Voltage Reference (Bottom Trace), Balanced Resistive Load, Controller Operating in an Open-Loop Mode	205
3.14: D-Axis Load Voltage Error (Top Trace) Verses Q-Axis Load Voltage Error (Bottom Trace), Balanced Resistive Load, Controller Operating in an Open-Loop Mode	206
3.15: D-Axis Load Voltage (Top Trace) Verses D-Axis Load Voltage Reference (Bottom Trace), Balanced Resistive Load, Controller Operating in a Closed-Loop Mode Using the Conventional Backpropagation Algorithm	207
3.16: Q-Axis Load Voltage (Top Trace) Verses Q-Axis Load Voltage Reference (Bottom Trace), Balanced Resistive Load, Controller Operating in a Closed-Loop Mode Using the Conventional Backpropagation Algorithm	208
3.17: D-Axis Load Voltage Error (Top Trace) Verses Q-Axis Load Voltage Error (Bottom Trace), Balanced Resistive Load, Controller Operating in a Closed-Loop Mode Using the Conventional Backpropagation Algorithm	209
3.18: Phase A Line-to-Line Load Voltage Spectral Information Obtained from the Balanced Resistive Load Simulation, Conventional Backpropagation Learning Process Enabled .	214

List of Figures

3.19: Q-Axis Load Voltage Reference (Top Trace) Verses the Q-Axis Load Voltage in the Balanced Resistive Load Experiment, Modified Backpropagation Learning Process Enabled	215
3.20: Q-Axis Load Voltage (Top Trace) Verses the Q-Axis Load Voltage Reference (Bottom Trace) in the Balanced Resistive Load Simulation, Modified Backpropagation Learning Process Enabled	216
3.21A: Open-Loop Normalized Q-Axis Load Voltage Error (Top Trace) Verses the Closed-Loop Normalized Q-Axis Load Voltage Error (Bottom Trace), Balanced Resistive Load Experiment	217
3.21B: Q-Axis Load Voltage Error without a Learning Process Delay (Top Trace) Verses the Q-Axis Load Voltage Error with One Controller Sampling Period Learning Process Delay (Bottom Trace), Balanced Resistive Load Simulation	219
3.22: Q-Axis Load Voltage Spectral Information, Balanced Resistive Load Simulation, Modified Backpropagation Learning Process Enabled	220
3.23: Q-Axis Load Voltage Spectral Information, Balanced Resistive Load Experiment, Modified Backpropagation Learning Process Enabled	221
3.24: D-Axis Load Voltage (Top Trace) Verses D-Axis Load Voltage Reference Waveform (Bottom Trace): Unbalanced Resistive Load, Controller Learning Process Disabled ..	224
3.25: Q-Axis Load Voltage (Top Trace) Verses Q-Axis Load Voltage Reference Waveform (Bottom Trace): Unbalanced Resistive Load, Controller Learning Process Disabled .	225
3.26: D-Axis Load Voltage Error (Top Trace) Verses Q-Axis Load Voltage Error (Bottom Trace), Unbalanced Resistive Load, Controller Operating in an Open-Loop Mode	226
3.27: Q-Axis Load Voltage Reference (Top Trace) Verses the Q-Axis Load Voltage in the Unbalanced Resistive Load Experiment, Modified Backpropagation Learning Process Enabled	227

List of Figures

3.28: Q-Axis Load Voltage (Top Trace) Verses the Q-Axis Load Voltage Reference (Bottom Trace) in the Unbalanced Resistive Load Simulation, Modified Backpropagation Learning Process Enabled	228
3.29: Open-Loop Normalized Q-Axis Load Voltage Error (Top Trace) Verses the Closed-Loop Normalized Q-Axis Load Voltage Error (Bottom Trace), Unbalanced Resistive Load Experiment	229
3.30: Q-Axis Load Voltage Spectral Information, Unbalanced Resistive Load Simulation, Modified Backpropagation Learning Process Enabled	230
3.31: Q-Axis Load Voltage Spectral Information, Unbalanced Resistive Load Experiment, Modified Backpropagation Learning Process Enabled	231
3.32: D-Axis Load Voltage (Top Trace) Verses D-Axis Load Voltage Reference Waveform (Bottom Trace): Nonlinear Load, Controller Learning Process Disabled	234
3.33: Q-Axis Load Voltage (Top Trace) Verses Q-Axis Load Voltage Reference Waveform (Bottom Trace): Nonlinear Load, Controller Learning Process Disabled	235
3.34: D-Axis Load Voltage Error (Top Trace) Verses Q-Axis Load Voltage Error (Bottom Trace), Nonlinear Load, Controller Operating in an Open-Loop Mode	236
3.35: Q-Axis Load Voltage Spectral Information, Nonlinear Load Experiment, Controller Operating in an Open-Loop Mode	237
3.36: Q-Axis Load Voltage Reference (Top Trace) Verses the Q-Axis Load Voltage in the Nonlinear Load Experiment, Controller Learning Process Enabled and the Modified Backpropagation Learning Strategy Employed	238
3.37: Q-Axis Load Voltage (Top Trace) Verses the Q-Axis Load Voltage Reference (Bottom Trace) in the Nonlinear Load Simulation, Modified Backpropagation Learning Process Enabled	239
3.38: Open-Loop Normalized Q-Axis Load Voltage Error (Top Trace) Verses the Closed-Loop Normalized Q-Axis Load Voltage Error (Bottom Trace), Nonlinear Load Experiment	243

List of Figures

3.39: Q-Axis Load Voltage Spectral Information, Nonlinear Load Simulation, Modified Backpropagation Learning Process Enabled	244
3.40: Q-Axis Load Voltage Spectral Information, Nonlinear Load Experiment, Modified Backpropagation Learning Process Enabled	245
3.41: Closed-Loop Normalized Load Voltage Errors, Repetitive Controller (Top Trace) and Neurocontroller (Bottom Trace), Nonlinear Load Simulation	246
3.42: Q-Axis Load Voltage Spectral Information, Nonlinear Load Simulation, Repetitive Controller Utilized	247
3.43: Load Voltage Instantaneous Error (Top Trace) Verses the DC Link Voltage (Bottom Trace): DC Power Supply Voltage Reduction from 200 VDC to 150 VDC, Controller Learning Process is Enabled	252
3.44: Load Voltage Instantaneous Error (Top Trace) Verses the Load Current (Bottom Trace): 100% Load Increase, Controller Learning Process Enabled	253
A1.1: GTO Model	268
A1.2: Diode Reverse Recovery Current and Voltage Waveforms ..	277
A1.3: Diode Model	280
A2.1: Inverter Leg Circuit	284
A2.2: Inverter Leg Model	285
A3.1: Passive Inductive Load Model	294
A3.2: Nonlinear Load Model	300
A4.1: Inverter Switch Configurations	313
A5.1: Simulation Software Structure	329
A7.1: Output Filter Circuit	346
A8.1: Processor Address/Data Bus Circuitry	355
A8.2: Low-Voltage Power Supply, Reset, Clock Circuitry	356
A8.3: Processor Memory Address Decode Circuitry	357
A8.4: Inverter Switch Control Signal Interface	359
A8.5A: Dead-Time Insertion Circuitry	360
A8.5B: Dead-Time Insertion Circuitry	361
A8.6: Space Vector PM Data ROM Circuitry	366
A8.7: Analog-to-Digital Conversion Control Circuitry	370
A8.8: Analog-to-Digital Converters and Interface	371
A8.9: ADC0831 Timing	372
A8.10: Sinusoidal Reference Generator Circuitry	375
A8.11A: Analog Interface Circuitry	377
A8.11B: Analog Interface Circuitry	378
A8.12: DAC and Configuration Switch Port	380

List of Tables

1.1:	Asymmetric Regular Sampled Sinusoidal PWM Simulation Results	58
1.2:	Asymmetric Regular Sampled Sinusoidal PWM With 25% Added Third-Harmonic Distortion Simulation	59
1.3:	Space Vector Sinusoidal PWM Simulation	60
2.1:	Comparison of Controller Techniques, Nonlinear Load. $f_s = 540$ Hz, $f_m = 60$ Hz. Load Voltage Harmonic Components Normalized with Respect to the Fundamental Frequency Component	150
2.2:	Comparison of Controller Techniques, Nonlinear Load. $f_s = 540$ Hz, $f_m = 60$ Hz. Inverter Output Current Harmonic Components Normalized with Respect to the Fundamental Frequency Component	151
2.3:	Comparison of Controller Techniques, Nonlinear Load. $f_s = 900$ Hz, $f_m = 60$ Hz. Load Voltage Harmonic Components Normalized with Respect to the Fundamental Frequency Component	152
2.4:	Comparison of Controller Techniques, Nonlinear Load. $f_s = 900$ Hz, $f_m = 60$ Hz. Inverter Output Current Harmonic Components Normalized with Respect to the Fundamental Frequency Component	153
3.1:	Comparison of Controller Techniques, Nonlinear Load. $f_s = 540$ Hz, $f_m = 60$ Hz. Load Voltage Harmonic Components Normalized with Respect to the Fundamental Frequency Component	195
3.2:	Comparison of Controller Techniques, Nonlinear Load. $f_s = 540$ Hz, $f_m = 60$ Hz. Inverter Output Current Harmonic Components Normalized with Respect to the Fundamental Frequency Component	196
3.3:	Comparison of Controller Techniques, Nonlinear Load. $f_s = 900$ Hz, $f_m = 60$ Hz. Load Voltage Harmonic Components Normalized with Respect to the Fundamental Frequency Component	197
3.4:	Comparison of Controller Techniques, Nonlinear Load. $f_s = 900$ Hz, $f_m = 60$ Hz. Inverter Output Current Harmonic Components Normalized with Respect to the Fundamental Frequency Component	198

List of Tables

3.5: Comparision of the Open- and Closed-Loop Load
Voltage Waveform Spectral Characteristics, Nonlinear
Load Experiment. Load Voltage Harmonic Components
Normalized with Respect to the Reference Waveform Peak
Amplitude 241

List of Symbols

α	=	Fourier Transform coefficient
β	=	Inverter output voltage waveform switching angle
$\epsilon_{dv}, \epsilon_{qv}$	=	d-q frame UPS load voltage errors
ϕ	=	Flux
τ	=	Inverter output voltage waveform level transition time
C_f	=	Output filter circuit phase-to-phase capacitance
c_k	=	k^{th} training interval learning-rate parameter
f_c	=	Lowest inverter output voltage harmonic frequency
f_f	=	Output filter circuit response pole frequency
f_m	=	Inverter output voltage modulation frequency
f_s	=	Inverter switching frequency
i_{ai}, i_{bi}, i_{ci}	=	Three-phase inverter output currents
i_{abi}	=	$i_{ai} - i_{bi}$
i_{bci}	=	$i_{bi} - i_{ci}$
i_{cai}	=	$i_{ci} - i_{ai}$
i_{di}, i_{qi}	=	d-q frame inverter output currents
i_{al}, i_{bl}, i_{cl}	=	Three-phase UPS load currents
i_{abl}	=	$i_{al} - i_{bl}$
i_{bcl}	=	$i_{bl} - i_{cl}$
i_{cal}	=	$i_{cl} - i_{al}$
i_{abl}'	=	di_{abl}/dt
i_{bcl}'	=	di_{bcl}/dt
i_{cal}'	=	di_{cal}/dt
i_{dl}, i_{ql}	=	d-q frame UPS load currents
$i_{dl}^{(1)}, i_{ql}^{(1)}$	=	$i_{abl}, i_{bcl}, i_{cal}$ expressed in the d-q frame
j	=	$\sqrt{-1}$
L_f	=	Output filter circuit series inductance
M	=	Inverter output voltage modulation index
M_3	=	Third-harmonic distortion index
t_s	=	$1/f_s$
T	=	Inverter controller sampling interval
v_{dc}	=	Inverter DC link voltage
v_{ai}, v_{bi}, v_{ci}	=	Three-phase inverter output phase-to-ground voltages
$v_{abi}, v_{bci}, v_{cai}$	=	Three-phase inverter output phase-to-phase voltages
$v_{abi}^*, v_{bci}^*, v_{cai}^*$	=	Three-phase inverter output phase-to-phase voltage commands
$v_{abin}, v_{bcin}, v_{cain}$	=	three-phase inverter output phase-to-phase voltages normalized with respect to v_{dc}

List of Symbols

v_{di}, v_{qi}	=	d-q frame inverter output phase-to-ground voltages
v_{al}, v_{bl}, v_{cl}	=	Three-phase UPS load phase-to-ground voltages
$v_{abl}, v_{bcl}, v_{cal}$	=	Three-phase UPS load phase-to-phase voltages
$v_{abl}^*, v_{bcl}^*, v_{cal}^*$	=	Three-phase UPS load phase-to-phase voltage commands
$v_{abln}, v_{bcln}, v_{caln}$	=	three-phase UPS load phase-to-phase voltages normalized with respect to v_{dc}
v_{dl}, v_{ql}	=	d-q frame UPS load phase-to-ground voltages
v_{dl}^*, v_{ql}^*	=	d-q frame UPS load phase-to-ground voltage commands
$v_{ds}(k), v_{qs}(k)$	=	d-q inverter output phase-to-phase space vector voltages corresponding to the k^{th} inverter switching state
v_{avg}	=	mean inverter output waveform
$v_{err}(avg)$	=	mean inverter output error waveform
v_{rv}	=	inverter DC power supply ripple voltage
w	=	Neurocontroller connection weight

I . INTRODUCTION

1.1 Uninterruptible Power Supplies

A solid-state uninterruptible power supply (UPS), utilizes a controlled-rectifier, battery, inverter, and output filter circuit to generate sinusoidal voltages for critical loads during commercial mains power disturbances. A diagram of a three-phase UPS is depicted in figure 1.1. Critical loads such as data processing, communication, and medical electronic systems are sensitive to the condition of the voltages supplied to these loads. As a result, UPS operational requirements include specifications limiting the load voltage amplitude variations and waveform harmonic distortion. Additionally, the load voltage phase-displacements are critical. Due to the occasional necessity of transferring load power flow between the UPS output and a system bypass circuit fed by the mains supply, the phase displacements between the UPS output voltages and the corresponding system bypass voltages are kept as small as possible. Load power flow transfer is required when the UPS system is maintained, the UPS system experiences a failure, or a severe UPS output overload condition occurs. Severe overload events include motor starting, energizing load circuit transformers, or clearing load short-circuits through circuit breakers or fuses. Load power transfer between the system bypass and UPS output must take place without producing a significant disturbance in the voltages supplied to the critical load. UPS inverters utilize either transformer-coupled harmonic

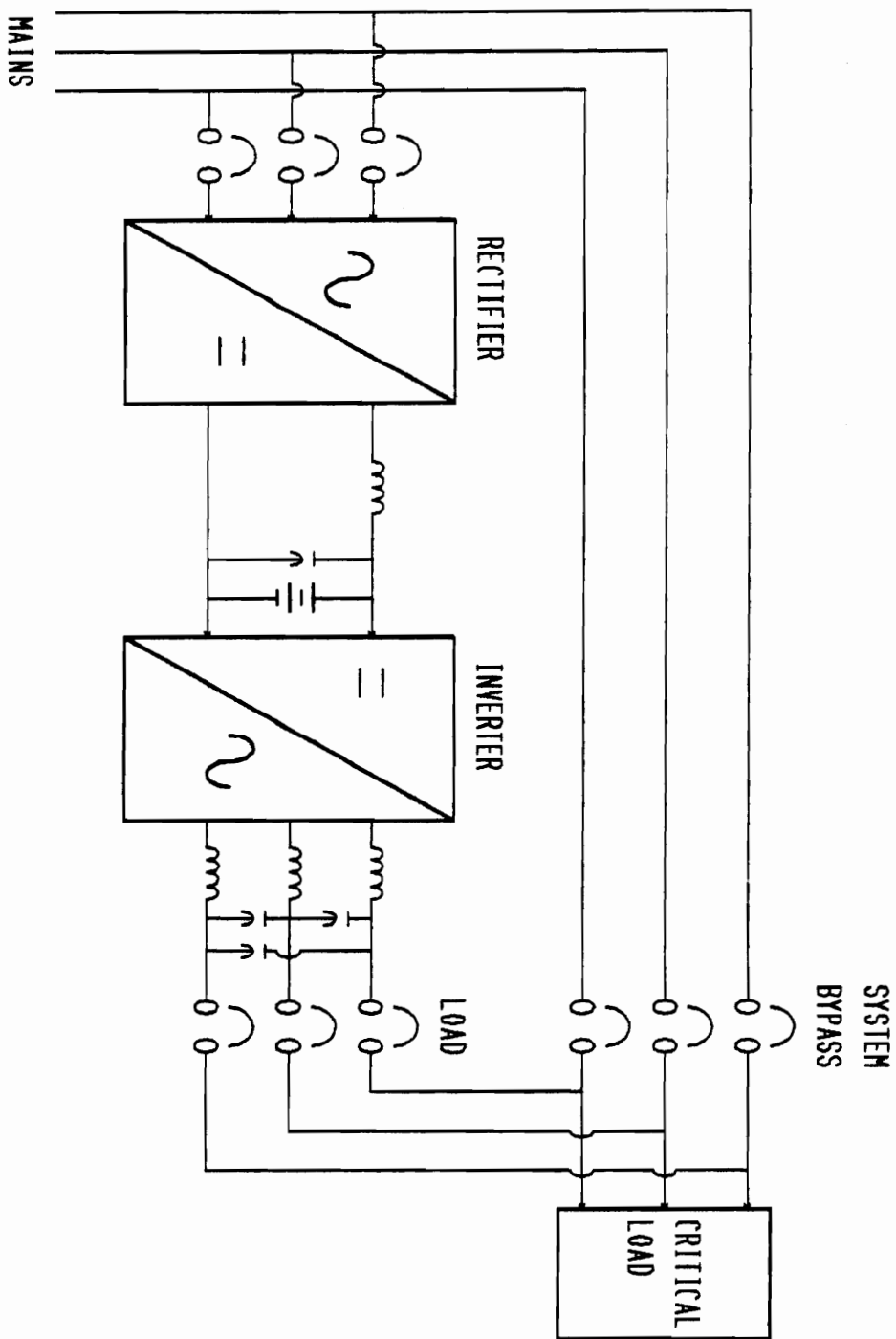


FIGURE 1.1: UNINTERRUPTIBLE POWER SUPPLY

neutralization or PulseWidth Modulation (PWM) inverter output waveform generation techniques. PWM techniques are typically employed due to the ability of these techniques to control the spectral characteristics of the inverter output waveform. Additionally, the use of PWM techniques translates into fewer power circuit components required in the implementation compared to the transformer-coupled harmonic neutralization method.

The output filter circuit is designed to suppress the unwanted harmonic frequency components present in the inverter output voltage waveforms. The filter circuit depicted in figure 1.2 is typical in three-phase UPS designs. Equations describing the output filter circuit operation are

$$L_f \cdot di_{abi}(t)/dt = v_{abi}(t) - v_{abl}(t), \quad (1.1a)$$

$$L_f \cdot di_{bci}(t)/dt = v_{bci}(t) - v_{bcl}(t), \quad (1.1b)$$

$$L_f \cdot di_{cai}(t)/dt = v_{cai}(t) - v_{cal}(t), \quad (1.1c)$$

$$i_{abi}(t) - i_{abl}(t) - 3 \cdot C_f \cdot dv_{abl}(t)/dt = 0, \quad (1.2a)$$

$$i_{bci}(t) - i_{bcl}(t) - 3 \cdot C_f \cdot dv_{bcl}(t)/dt = 0, \quad (1.2b)$$

and

$$i_{cai}(t) - i_{cal}(t) - 3 \cdot C_f \cdot dv_{cal}(t)/dt = 0, \quad (1.2c)$$

where $\{v_{abl}, v_{bcl}, v_{cal}\}$ are the line-to-line load voltages, $\{v_{abi}, v_{bci}, v_{cai}\}$ are the line-to-line inverter output voltages, L_f is the filter circuit series inductance, and C_f is the filter circuit line-to-line capacitance. Defining $\{i_{al}, i_{bl}, i_{cl}\}$ as the load phase currents and $\{i_{ai}, i_{bi}, i_{ci}\}$ as the inverter output currents, the following terms are defined:

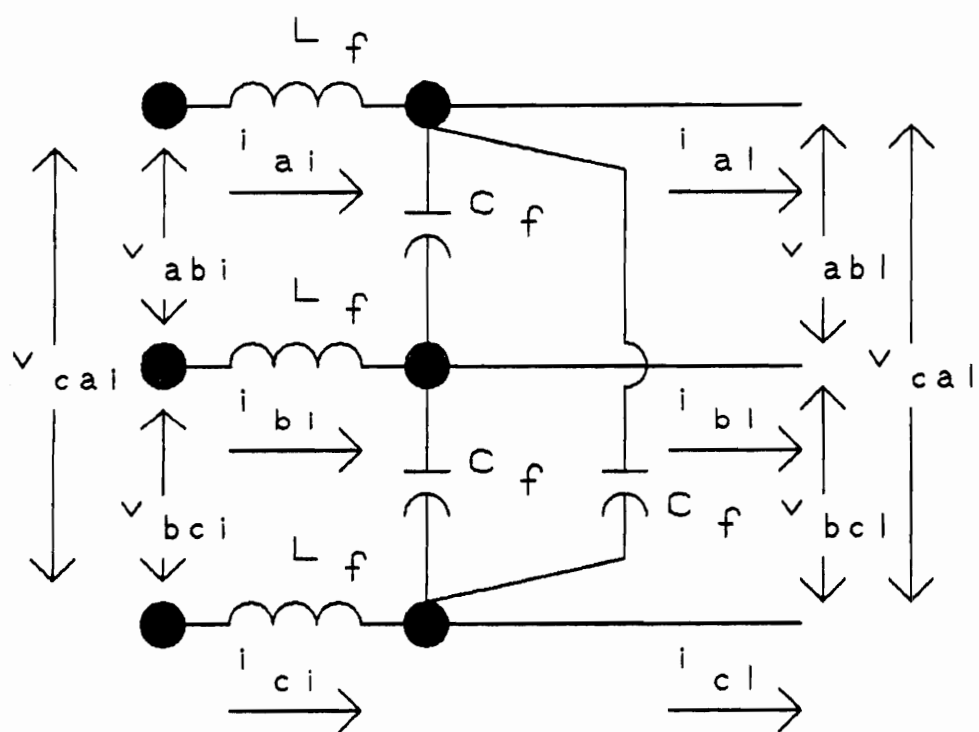


FIGURE 1.2: OUTPUT FILTER CIRCUIT

$$i_{abl} = i_{al} - i_{bl}, \quad (1.3a)$$

$$i_{bcl} = i_{bl} - i_{cl}, \quad (1.3b)$$

$$i_{cal} = i_{cl} - i_{al}, \quad (1.3c)$$

$$i_{abi} = i_{ai} - i_{bi}, \quad (1.4a)$$

$$i_{bci} = i_{bi} - i_{ci}, \quad (1.4b)$$

and

$$i_{cai} = i_{ci} - i_{ai}. \quad (1.4c)$$

Using (1.2a), (1.2b), and (1.2c) to eliminate the inverter output current variables from the output filter state equations, (1.1a), (1.1b), and (1.1c) are re-written in the form

$$\begin{aligned} d^2 v_{abl}(t)/dt^2 = & (2\pi \cdot f_f)^2 \cdot [v_{abi}(t) - v_{abl}(t)] - \\ & (2\pi \cdot f_f)^2 \cdot L_f \cdot di_{abl}(t)/dt, \end{aligned} \quad (1.5a)$$

$$\begin{aligned} d^2 v_{bcl}(t)/dt^2 = & (2\pi \cdot f_f)^2 \cdot [v_{bci}(t) - v_{bcl}(t)] - \\ & (2\pi \cdot f_f)^2 \cdot L_f \cdot di_{bcl}(t)/dt, \end{aligned} \quad (1.5b)$$

and

$$\begin{aligned} d^2 v_{cal}(t)/dt^2 = & (2\pi \cdot f_f)^2 \cdot [v_{cai}(t) - v_{cal}(t)] - \\ & (2\pi \cdot f_f)^2 \cdot L_f \cdot di_{cal}(t)/dt, \end{aligned} \quad (1.5c)$$

where $2\pi \cdot f_f = 1/(3 \cdot L_f \cdot C_f)$. The Fourier Transforms of (1.5a), (1.5b), and (1.5c) are

$$\begin{aligned} v_{abl}(f) = & [1/(1 - 12 \cdot \pi^2 \cdot f^2 \cdot L_f \cdot C_f)] \cdot [v_{abi}(f) - \\ & j \cdot 2\pi \cdot f \cdot L_f \cdot i_{abl}(f)], \end{aligned} \quad (1.6a)$$

$$\begin{aligned} v_{bcl}(f) = & [1/(1 - 12 \cdot \pi^2 \cdot f^2 \cdot L_f \cdot C_f)] \cdot [v_{bci}(f) - \\ & j \cdot 2\pi \cdot f \cdot L_f \cdot i_{bcl}(f)], \end{aligned} \quad (1.6b)$$

and

$$v_{cal}(f) = [1/(1 - 12 \cdot \pi^2 \cdot f^2 \cdot L_f \cdot C_f)] \cdot [v_{cai}(f) -$$

$$j \cdot 2\pi \cdot f \cdot L_f \cdot i_{cal}(f)], \quad (1.6c)$$

where the Fourier Transform of a function $g(t)$ is computed from the expression

$$g(f) = \int_{-\infty}^{\infty} g(t) \cdot \exp(-j \cdot 2\pi \cdot f \cdot t) dt, \quad j = \sqrt{-1}.$$

The terms

$$v_{abi}(f)/(1 - 12 \cdot \pi^2 \cdot f^2 \cdot L_f \cdot C_f),$$

$$v_{bci}(f)/(1 - 12 \cdot \pi^2 \cdot f^2 \cdot L_f \cdot C_f), \text{ and}$$

$$v_{cai}(f)/(1 - 12 \cdot \pi^2 \cdot f^2 \cdot L_f \cdot C_f),$$

are the no-load UPS line-to-line load voltages. Inverter output waveform harmonic distortion suppression requirements determine the value of the product term $L_f \cdot C_f$. Typically, each inverter output voltage harmonic component is attenuated to a normalized maximum value, for example 0.03, where normalization is performed using the fundamental frequency component as the reference. The product $L_f \cdot C_f$ is then computed based on the attenuation of the lowest order harmonic frequency component f_c in the ideal inverter line-to-line output voltage waveform using the equation $L_f \cdot C_f = 11.444/(2\pi \cdot f_c)^2$ where $1/(1 - 12 \cdot \pi^2 \cdot f_c^2 \cdot L_f \cdot C_f) = -0.03$. The value of f_c is dependent on the inverter switching frequency f_s and the PWM technique employed. f_c and f_s are typically harmonics of the inverter fundamental (modulation) frequency f_m . f_s is a triplen harmonic.

The value of L_f determines the value of RMS inverter output harmonic current due to inverter output voltage harmonic distortion. It is typical for L_f to be chosen based on minimizing the total inductor

volt-amperes due to both fundamental and harmonic inverter output currents. As with the computation of the product L_f and C_f , the value of L_f is dependent on the inverter switching frequency and PWM technique employed.

The application of UPS load alters the no-load UPS output voltages due to the voltage drops

$$j \cdot 2\pi \cdot f \cdot L_f \cdot i_{abl}(f) / (1 - 12 \cdot \pi^2 \cdot f^2 \cdot L_f \cdot C_f),$$

$$j \cdot 2\pi \cdot f \cdot L_f \cdot i_{bcl}(f) / (1 - 12 \cdot \pi^2 \cdot f^2 \cdot L_f \cdot C_f), \text{ and}$$

$$j \cdot 2\pi \cdot f \cdot L_f \cdot i_{cal}(f) / (1 - 12 \cdot \pi^2 \cdot f^2 \cdot L_f \cdot C_f).$$

Balanced and passive (linear) UPS loads which typically have unity or lagging power factors cause equal load voltage magnitude reductions and lagging phaseshifts. In a three-phase closed-loop UPS load voltage control scheme, the inverter line-to-line output voltages $\{v_{abi}, v_{bci}, v_{cai}\}$ are modified in a symmetrical (balanced) manner to compensate for the balanced passive load-generated magnitude and phaseshift variations.

Unbalanced and nonlinear UPS loads create more difficult three-phase load voltage control problems. Unbalanced passive loads cause unequal line-to-line load voltage magnitude and phaseshift variations. Three-phase closed-loop load voltage control which compensates for the unbalanced passive load-generated magnitude and phaseshift variations forces the generation of asymmetrical (or unbalanced) three-phase line-to-line inverter output voltages.

Nonlinear UPS loads such as rectifier and triac-controlled loads generate UPS load currents with significant low-order harmonic currents.

Third harmonic load currents occur with single-phase switch-mode power supply loads. Three-phase nonlinear loads generate significant 5th and 7th harmonic load currents. The magnitude of the harmonic voltage drops across the complex impedance

$$j \cdot 2\pi \cdot f \cdot L_f / (1 - 12\pi^2 \cdot f^2 \cdot L_f \cdot C_f)$$

due to the harmonic load current are dependent on the placement of the output filter pole and the size of L_f . Prudent output filter design avoids placement of the output filter pole at a frequency corresponding to the harmonics associated with typical nonlinear loads. Three-phase closed-loop load voltage control which compensates for the nonlinear load-generated harmonic distortion forces the selective generation of harmonic voltages in the three-phase line-to-line inverter output voltages which in turn causes the inverter (rather than the output filter capacitors) to source the load harmonic currents. The generation of triplen harmonic inverter output line-to-line voltages requires unbalanced three-phase inverter switching.

Load-generated fundamental load voltage magnitude variations, phaseshifts, and harmonic distortion together with inverter DC supply voltage variations (and harmonic distortion) and inverter switching distortion are the primary disturbance mechanisms in a closed-loop UPS load voltage control scheme. Several closed-loop control methods have been proposed (references [1][2][3][4][5]) which have been designed to deal with load-generated disturbances.

In the hysteresis control method presented in reference [1], the PWM inverter switching mode is determined by the comparison of a

sinusoidal reference with the generated load voltage waveform. The resulting error signal after processing by a proportional-derivative controller is input to a hysteresis element which in turn generates the PWM inverter switching control signals. The performance of this method is dependent on the inverter switching frequency where higher switching frequencies correspond to improved performance. Dependent on the window width of the hysteresis element, the inverter switching frequency, and the harmonic frequency content of the load current, a ripple voltage is impressed on the sinusoidal waveform generated by the inverter. Hysteresis control methods are used in low-power inverters where high inverter switching frequencies (in the order of tens of kiloHertz) are practical. The highest practical inverter switching frequency is determined by power semiconductor switching delays and losses, as well as, limitations placed on the generation of electro-magnetic interference (EMI) produced as a result of inverter switching.

An alternate method of achieving a fast-response closed-loop load voltage control scheme is implemented through the use of dead-beat control techniques (references [2][3][4][5]). These methods utilize the inverse transfer function (or inverse dynamics) of the output filter circuit to design the inverter controller. The controller outputs are the modulation commands applied to the PWM inverter switch control logic (where the command signals either control the inverter output waveform pulsewidth per switching period directly or through a conventional sinusoidal PWM technique such as carrier or space vector PWM). Unlike the hysteresis method where the inverter switching frequency can vary

(dependent on the load characteristics), these control schemes operate with a fixed inverter switching frequency. Though high inverter switching frequencies improve the performance of these control schemes, it is not necessary to utilize switching frequencies of the magnitude required by the hysteresis method.

In addition to the inverter switching frequency magnitude, controller design using the output filter inverse dynamics is dependent on the filter component values (L_f and C_f) and knowledge of the mechanisms creating load voltage waveform distortion. In a digital controller design, the discrete-time version of the equations describing output filter operation is used where the interval of discretization is dependent on the inverter switching frequency and modulation sample method. Digital controller gains are dependent on the output filter circuit component values (their tolerances), as well as, the inverter switching frequency magnitude. Additionally, prediction of disturbances in the form of the load current and DC power supply voltage variations is required due to the inherent delay built into the process of synthesizing a sinusoidal waveform in the output filter circuit, and the delay associated with the controller execution speed.

Other methods of compensating for inverter/filter/load system disturbances involve the modification of the inverter output voltage waveform spectral characteristics. Reference [6] describes a method which utilizes an optimized fixed-pattern inverter switching scheme (optimized with respect to the suppression of low-order harmonics) in conjunction with the regulation of the DC voltage supplied to the

inverter. The suppression (or elimination) of low-order inverter output voltage waveform harmonics through the use of a fixed optimized inverter switching pattern increases the lowest harmonic frequency ω_c at which the output filter circuit is designed to attenuate inverter output waveform harmonics. By increasing ω_c , the value of the product term $L_f \cdot C_f$ increases, and the value of L_f decreases due to the decrease in the filter inductor harmonic volt-amperes. As a result of increasing ω_c through the use of an optimized inverter switching pattern, the magnitude of the impedance

$$2\pi \cdot f \cdot L_f / (1 - 12\pi^2 \cdot f^2 \cdot L_f \cdot C_f)$$

at the inverter fundamental frequency f_m , as well as, at the low-order harmonics associated with nonlinear UPS loads is reduced without increasing the inverter switching frequency. This in turn reduces the adverse effects on the three-phase load voltage waveforms created by unbalanced and nonlinear UPS load currents.

Reference [7] describes a method designed to suppress the influence of DC power supply voltage harmonic frequency distortion. As in the method described in [6], an optimized fixed-pattern inverter switching scheme is used (in this case, a switching strategy based on selective harmonic elimination), enabling the reduction of the output filter inductance value L_f and the increase of the filter's pole frequency. The adverse effects of unbalanced and nonlinear loads are reduced as in the method described in [6]. Unlike the method in reference [6], inverter output voltage control is implemented through the use of a modulation index command which selects the programmed PWM

switching pattern stored in ROM (read-only memory). Programmed PWM switching data stored in ROM is generated from off-line computations (as described in references [8] and [9]) for discrete modulation index values.

Using the method described in [7], the modulation index command value which selects the inverter switching pattern stored in ROM is varied in a manner which suppresses the inverter output voltage distortion due to the inverter DC power supply ripple voltage. Defining v_{rv} as the DC power supply instantaneous ripple voltage, the modulation index command is multiplied by the factor $1/[1 + v_{rv}/V_{dc}]$ where V_{dc} is the nominal DC power supply voltage.

Another method of dealing with inverter/filter/load system disturbances in the form of load-generated harmonic distortion is the addition of harmonic traps or some form of active filter to the existing output filter design. This approach is typically feasible in medium- to high-power systems where the associated added cost may be tolerated. Additions to the existing output filter circuit in the form of tuned traps and active filters do not resolve load voltage waveform control problems associated with unbalanced UPS loads.

The reduction of the filter impedance magnitude at the fundamental frequency and harmonics associated with nonlinear UPS loads is a step in the direction towards the solution of load voltage waveform control problems associated with unbalanced and nonlinear loads. Impedance reduction can be achieved by increasing the inverter switching frequency. Due to practical limitations such as power semiconductor

switching delays and losses and the generation of electromagnetic interference (EMI), reduction of the filter impedance magnitude in this manner is limited with regard to medium- and high-power UPS applications.

Through the use of an optimized inverter, PWM technique filter impedance reduction is achieved without a significant increase in the inverter switching frequency.

1.2 Survey of PWM Waveform Generation Techniques

Methods of generating inverter output waveform switching angles are based either on some form of direct control by the load voltage controller (as in the case of a hysteresis controller) or on the use of a feedforward PWM technique. Historically, the development of feedforward PWM waveform generation techniques has been associated primarily with the control of polyphase AC motors. PWM technique development has been directed towards the following concerns regarding the performance of a polyphase AC motor under inverter control:

- i.) Linear control of the fundamental frequency voltage implies linear control of motor speed and torque. Efforts to extend the linear range of voltage control have been made to minimize the adverse effects of the transition from linear operation to quasi-squarewave operation in order to maintain a constant Volts/Hertz ratio at higher motor speeds. The effects associated with the transition from the linear range to quasi-squarewave operation

include pulse-dropping (with corresponding jumps in fundamental frequency voltage magnitude), and increased low-order harmonic frequency generation (which contributes to the generation of motor torque pulsations).

- ii.) The harmonic frequency components in the inverter output voltage waveform produce harmonic currents in the motor. These harmonic currents in turn create copper losses within the motor, as well as, adversely affect motor performance by producing torque pulsations. Suppression of low-order harmonic frequency components which is primarily responsible for the minimization of the magnitude of the inverter output harmonic current has been an objective in the continued development of PWM waveform generation techniques.

Feedforward PWM waveform control techniques belong to one of three main categories: carrier PWM, optimized switching strategies, and space vector PWM. A discussion of each method follows.

1.2.1 Carrier PWM

Carrier PWM techniques are presently the most widely used methods of inverter output waveform control due to their relative simplicity. The suboscillation or triangulation method (or naturally-sampled sinusoidal PWM) is the best known, and is commonly used as the benchmark in the evaluation of other PWM techniques. In carrier PWM waveform generation processes, a modulation reference waveform (typically a

sinusoid) is compared directly to a carrier waveform (typically a bipolar isosceles triangular waveform) where the intersection of these two waveforms determines the PWM waveform switching angles (or waveform transition times). Referring to the ideal inverter output phase voltage pulse waveform depicted in figure 1.3, the naturally-sampled PWM method generates waveform transition times τ_k and τ_{k+1} based on the solution of the equations

$$-1.0 + 4.0 \cdot [\tau_k - k \cdot t_s] / t_s = M \cdot \sin[2\pi \cdot f_m \cdot \tau_k + \theta], \quad (1.7a)$$

$$\tau_k = t_s \cdot \{k + 0.25 + 0.25 \cdot M \cdot \sin[2\pi \cdot f_m \cdot \tau_k + \theta]\} \text{ and} \quad (1.7b)$$

$$3.0 - 4.0 \cdot [\tau_{k+1} - k \cdot t_s] / t_s = M \cdot \sin[2\pi \cdot f_m \cdot \tau_{k+1} + \theta], \quad (1.8a)$$

$$\tau_{k+1} = t_s \cdot \{k + 0.75 - 0.25 \cdot M \cdot \sin[2\pi \cdot f_m \cdot \tau_{k+1} + \theta]\}. \quad (1.8b)$$

where k is an integer, M is the normalized amplitude of the modulation reference waveform (normalized with respect to the carrier waveform peak amplitude), t_s is the inverter switching frequency period, f_m is the modulation waveform frequency, and θ is the phase-displacement between the modulation and carrier waveforms. M is also commonly referred to as the modulation index. Implementation of the naturally-sampled PWM method is realized in analog circuit designs where an analog level comparator is used to make the direct comparison between the modulation and carrier waveforms.

1.2.2 PWM Regular Sampling Methods

In microprocessor-based PWM controllers using carrier PWM techniques, the modulation reference waveform is sampled at a rate equal

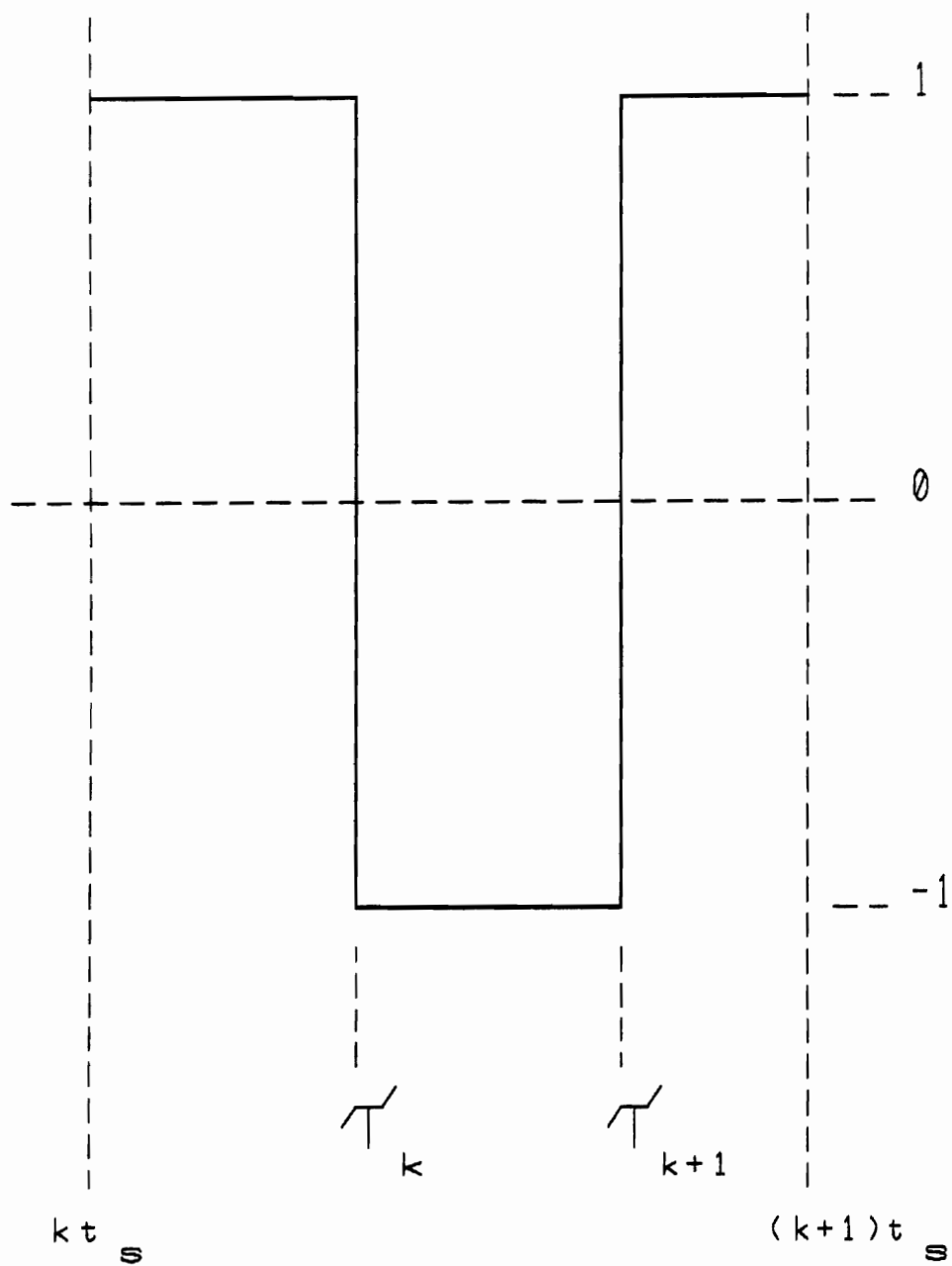


FIGURE 1.3: IDEAL INVERTER OUTPUT PULSE WAVEFORM

to either the inverter switching frequency or twice the switching frequency. Modulation waveform sampling is typically synchronized with the carrier waveform (sampling is regular or uniform).

1.2.2.1 Symmetric Regular Sampling

Using symmetric regular sampling, the modulation waveform is sampled once per inverter switching period. Using this method of sampling and sampling at a time which coincides with the negative peak of the carrier waveform, waveform transition times generated by the carrier PWM process are derived using the equations

$$-1.0 + 4.0 \cdot [\tau_k - k \cdot t_s] / t_s = M \cdot \sin[2\pi \cdot f_m \cdot k \cdot t_s + \theta], \quad (1.9a)$$

$$\tau_k = t_s \cdot \{k + 0.25 + 0.25 \cdot M \cdot \sin[2\pi \cdot f_m \cdot k \cdot t_s + \theta]\} \text{ and} \quad (1.9b)$$

$$3.0 - 4.0 \cdot [\tau_{k+1} - k \cdot t_s] / t_s = M \cdot \sin[2\pi \cdot f_m \cdot k \cdot t_s + \theta], \quad (1.10a)$$

$$\tau_{k+1} = t_s \cdot \{k + 0.75 - 0.25 \cdot M \cdot \sin[2\pi \cdot f_m \cdot k \cdot t_s + \theta]\}. \quad (1.10b)$$

Sampling at a time which coincides with the positive peak of the carrier waveform results in the equations

$$-1.0 + 4.0 \cdot [\tau_k - k \cdot t_s] / t_s = M \cdot \sin[2\pi \cdot f_m \cdot (k + 0.5) \cdot t_s + \theta], \quad (1.11a)$$

$$\tau_k = t_s \cdot \{k + 0.25 + 0.25 \cdot M \cdot \sin[2\pi \cdot f_m \cdot (k + 0.5) \cdot t_s + \theta]\} \quad (1.11b)$$

and

$$3.0 - 4.0 \cdot [\tau_{k+1} - k \cdot t_s] / t_s = M \cdot \sin[2\pi \cdot f_m \cdot (k + 0.5) \cdot t_s + \theta] \quad (1.12a)$$

$$\tau_{k+1} = t_s \cdot \{k + 0.75 - 0.25 \cdot M \cdot \sin[2\pi \cdot f_m \cdot (k + 0.5) \cdot t_s + \theta]\}. \quad (1.12b)$$

1.2.2.2 Asymmetric Regular Sampling

Using asymmetric regular sampling, the modulation waveform is sampled twice per inverter switching period. Using this method of sampling and sampling at a time which coincides with the peaks of the carrier waveform, waveform transition times generated by the carrier PWM process are derived using the equations

$$-1.0 + 4.0 \cdot [\tau_k - k \cdot t_s] / t_s = M \cdot \sin[2\pi \cdot f_m \cdot k \cdot t_s + \theta], \quad (1.13a)$$

$$\tau_k = t_s \cdot \{k + 0.25 + 0.25 \cdot M \cdot \sin[2\pi \cdot f_m \cdot k \cdot t_s + \theta]\} \quad (1.13b)$$

and

$$3.0 - 4.0 \cdot [\tau_{k+1} - k \cdot t_s] / t_s = M \cdot \sin[2\pi \cdot f_m \cdot (k + 0.5) \cdot t_s + \theta], \quad (1.14a)$$

$$\tau_{k+1} = t_s \cdot \{k + 0.75 - 0.25 \cdot M \cdot \sin[2\pi \cdot f_m \cdot (k + 0.5) \cdot t_s + \theta]\}. \quad (1.14b)$$

Sampling at a time which coincides with the zero-crossings of the carrier waveform, waveform transition times generated by the carrier PWM process are derived using the equations

$$-1.0 + 4.0 \cdot [\tau_k - k \cdot t_s] / t_s = M \cdot \sin[2\pi \cdot f_m \cdot (k + 0.25) \cdot t_s + \theta], \quad (1.15a)$$

$$\tau_k = t_s \cdot \{k + 0.25 + 0.25 \cdot M \cdot \sin[2\pi \cdot f_m \cdot (k + 0.25) \cdot t_s + \theta]\} \text{ and} \quad (1.15b)$$

$$3.0 - 4.0 \cdot [\tau_{k+1} - k \cdot t_s] / t_s = M \cdot \sin[2\pi \cdot f_m \cdot (k + 0.75) \cdot t_s + \theta], \quad (1.16a)$$

$$\tau_{k+1} = t_s \cdot \{k + 0.75 - 0.25 \cdot M \cdot \sin[2\pi \cdot f_m \cdot (k + 0.75) \cdot t_s + \theta]\}. \quad (1.16b)$$

Compared to equations (1.7b) and (1.8b) describing the generation of v_k and v_{k+1} using naturally-sampled PWM, equations associated with regular sampled techniques are comparatively simple algebraic equations requiring far less computational overhead, and as such, are easier to implement in a microprocessor-based PWM controller. Another benefit of regular sampling of the modulation waveform is that the absolute value of the sampled modulation waveform slope is always less than that of the carrier waveform. As a result, the pulse generation process is well-defined in each inverter switching period. This is not always the case with regard to naturally-sampled PWM when instantaneous changes in the modulation waveform amplitude are made.

Regular sampling techniques are also applied to space vector PWM and optimized switching strategies. Typically, asymmetric regular sampling techniques are applied since the sampled modulation waveform associated with carrier PWM (or sampled modulation commands signals associated with other PWM processes) contain more spectral information than symmetric regular sampling techniques.

Due to their filtering characteristics, regular sampling techniques alter the relationship between modulation commands and the corresponding inverter output voltage waveform spectral characteristics. Defining $v_i^*(f)$ as the Fourier Transform of the modulation command signal where

$$v_i^*(f) = \sum_{k=-\infty}^{\infty} a_k \cdot \delta(f - k \cdot f_m), \quad (1.17)$$

where f_m is the modulation frequency, f is the frequency variable, k is an integer, a_k is the modulation command signal Fourier coefficient corresponding to the frequency $f = k \cdot f_m$, and $\delta()$ is the Dirac delta function, the Fourier Transform of the sampled modulation command signal is given by the expression

$$\int_{-\infty}^{\infty} v_i^*(f - \Omega) \cdot \sum_{n=-\infty}^{\infty} \delta(\Omega - n/t_0) d\Omega \cdot \sin(\pi \cdot t_0 \cdot f) / (\pi \cdot t_0 \cdot f), \quad (1.18)$$

where n is an integer, Ω is a variable of integration, and t_0 is the modulation command signal sampling interval. The Fourier Transform of the modulation command is modified by the sampling process in two ways:

- i.) The amplitudes of the modulation command spectral components are modified by the $\sin(\pi \cdot t_0 \cdot f) / (\pi \cdot t_0 \cdot f)$ function.
- ii.) Frequency components of the orders $k \cdot f_m \pm n/t_0$ ($n > 0$) are added to the modulation command Fourier Transform. An aliasing (or waveform distortion) effect occurs if the modulation command signals contain spectral components at frequencies greater than twice the sampling frequency.

Regular sampling techniques convey a nonlinear characteristic to the PWM waveform generation process transfer function with regard to selectively-generating harmonic components in the inverter output voltage waveform. Increasing the inverter switching frequency and the sampling frequency minimizes the nonlinear effects of the $\sin(\pi \cdot t_0 \cdot f) / (\pi \cdot t_0 \cdot f)$ term.

1.2.3 Modulation Waveform Modification Using Added Third-Harmonic Distortion

Using the sinusoidal PWM methods described in the last section, the line-to-line inverter output voltage fundamental has a peak value of $0.866 \cdot V_{dc}$ (V_{dc} is the inverter DC supply voltage) when the modulation index is equal to 1.0. In a polyphase induction motor drive system, the DC supply for the voltage-source inverter is a three-phase uncontrolled rectifier which generates an average DC voltage output equal to $1.35 \cdot V_{in}$ (where V_{in} is the RMS line-to-line AC supply voltage feeding the rectifier). The maximum RMS inverter output line-to-line voltage at the fundamental frequency is then $0.827 \cdot V_{in}$. If the inverter load is a standard motor, the peak inverter output voltage available when the modulation index is 1.0 is not sufficient to operate the motor at its rated speed. Houldsworth and Grant [10] proposed a method of increasing the inverter output voltage fundamental component by adding a third-harmonic component to the modulation waveform. A modulation waveform described by the expression

$$m(t) = M \cdot \{\sin[2\pi \cdot f_m \cdot t + \theta] + (1/6) \cdot \sin[6\pi \cdot f_m \cdot t + 3 \cdot \theta]\} \quad (1.19)$$

increases the linear fundamental voltage control range by 15%. The third-harmonic component which appears in the inverter leg output voltage waveform is absent in the line-to-line waveform due to triplen-harmonic cancellation in a balanced three-phase system.

Another benefit obtained from the addition of third-harmonic distortion to the modulation waveform is the modification of output

waveform spectra in a fashion that reduces the RMS inverter output harmonic current [11][12]. The expression for the modulation waveform with added third-harmonic distortion which achieves this objective is

$$m(t) = M \cdot \{\sin[2\pi \cdot f_m \cdot t + \theta] + (1/4) \cdot \sin[6\pi \cdot f_m \cdot t + 3 \cdot \theta]\}. \quad (1.20)$$

1.2.4 Carrier PWM Inverter Output Waveform Spectral Characteristics

Characteristic of synchronous PWM processes (where the ratio f_s/f_m is an integer) is the presence of discrete harmonic components (sideband frequencies of the carrier frequency and its multiples) in the inverter output waveform. Asynchronous PWM processes which occur in connection with variable-speed AC motor drives generate output waveforms with continuous spectral characteristics. PWM inverters used in UPS systems typically employ synchronous modulation techniques in order to avoid generating harmonic frequencies other than the carrier sideband frequencies normally generated with synchronous modulation.

Naturally-sampled carrier PWM generates waveforms with the lowest-order harmonic frequencies at $f_m \cdot (p \pm 2)$ where $p = f_s/f_m$. Regular sampling techniques cause an additional two lowest-order harmonic frequencies located at $f_m \cdot (p \pm 4)$ to be generated. The waveform spectral characteristics associated with asymmetric regular sampled carrier PWM more closely resembles the spectral characteristics associated with naturally-sampled carrier due to the additional modulation waveform data obtained by sampling at twice the inverter switching frequency.

The addition of third harmonic distortion to the modulation waveform also causes the generation of $f_m \cdot (p \pm 4)$ harmonics in the output waveform. In contrast to the other carrier PWM techniques examined, the magnitude of the lowest-order harmonic components are reduced which corresponds to the generation of lower RMS inverter output harmonic currents.

1.2.5 Optimized Switching Strategies

With regard to their use in variable-speed drives, carrier PWM methods operating in an asynchronous modulation mode and with low f_s/f_m ratios generate low-order harmonics in the inverter output waveform not generated in a synchronous modulation modes which degrade both inverter and motor performance. PWM methods such as selective harmonic elimination and other optimal modulation strategies (collectively referred to as optimized switching strategies) were devised initially with the idea of avoiding this problem associated with the use of carrier PWM in an asynchronous modulation mode. Due to the synchronization between the output waveform pulse pattern and the desired fundamental frequency sinusoidal inverter output waveform, the output waveform is ideally free of the low-order harmonics generated in an asynchronous modulation mode.

The typical implementation of this group of modulation techniques is in the form of programmed waveforms. Optimized inverter switching patterns are computed off-line. Programmed waveforms (in the form of an

inverter switching pattern defined in periodic intervals such as in quarter cycle intervals), each corresponding to a discrete command value (such as a modulation index command), are subsequently stored in the memory circuitry of a digital PWM controller for use in real-time inverter output waveform control.

The process of deriving inverter leg output voltage waveform switching angles typically begins with the assumption that the output waveform has quarter- and half-wave symmetry. Subsequently, the switching angles in the first positive quarter cycle of the inverter output waveform are the dependent variables. Given the quarter-cycle symmetry assumption, the Fourier series coefficient equation for the inverter leg circuit output voltage waveform referenced to the inverter DC power supply midpoint is

$$a_n = [4/(n \cdot \pi)] \cdot [1 + 2 \cdot \sum_{k=1}^K (-1)^k \cdot \cos(n \cdot \beta_k)] \quad (1.21)$$

where a_n is the Fourier series coefficient associated with the n^{th} output waveform harmonic component, K is the number of switching angles in the first positive quarter cycle, and β_k is the k^{th} inverter output waveform switching angle ($0 < \beta_1 < \beta_2 < \dots \beta_K < \pi/2$).

Mathematical techniques associated with the generation of switching angle solutions are described in references [8][9] (selective harmonic elimination) and reference [13] (a general method for optimizing inverter output waveforms based on the minimization of a objective or cost function). Typical cost functions applied to switching

angle solutions include the minimization of inverter output harmonic current RMS values, and, torque and speed ripples.

The use of programmed waveforms has several drawbacks. An obvious drawback is that the conditions under which the waveform switching angles were derived (for example, either ideal inverter switching or use of a specific load circuit model) are not necessarily the conditions existing in an operating system. The effects of the resulting waveform errors become more pronounced as the number of output waveform pulses increase. Other problems associated with the use of programmed waveforms are:

- (i) As the output waveform pulse number increases, the modulation index decreases in inverse proportion. The modulation index command range also decreases with the pulse number in inverse proportion. The required changes in the output waveform pulse number in the attempt to expand the modulation index command range results in a far more complex control scheme.
- (ii) The basis on which programmed waveforms are generated is the assumption that waveforms are periodic. This is inconsistent with the behavior of inverter output waveform generation given fast-response closed-loop control of inverter switching. If the modulation index command is varied by a fast-response closed-loop controller, the resulting inverter output waveforms could be quite different from the optimum waveforms desired.

A more efficient application of the programmed waveform approach is through the use of a fixed optimized inverter switching pattern with fundamental frequency voltage control performed through inverter DC power supply voltage control. Typically, the three-phase inverter switching pattern is devised to selectively eliminate low-order harmonics (5^{th} , 7^{th} , 11^{th} , 13^{th} , ...). In an UPS inverter application, the use of this approach is designed to modify the output filter design in fashion which reduces the effect of UPS load-generated distortion of the load voltage waveforms. Additionally, elimination of harmonic frequency pairs (for example, 5^{th} , 7^{th}) leads to a corresponding elimination of inverter input DC current harmonics (the 6^{th} given the elimination of the inverter output voltage 5^{th} and 7^{th}). The value of eliminating DC supply current harmonic components in this fashion is limited because low-order harmonic elimination simply moves the DC current harmonic spectra to higher frequency locations aggravating potential EMI problems, and second harmonic DC currents resulting from unbalanced UPS loads are not eliminated by this process. The use of an inverter DC power supply voltage controller in the technique described in reference [6] isolates the DC current harmonics generated at the inverter DC inputs from the remainder of the DC link circuit (the battery and the UPS AC/DC converter). This additional power circuitry increases the complexity of the UPS design which is not desired given concerns regarding UPS reliability. UPS systems are designed to tolerate mains power disturbances such as high-speed mains voltage spikes which would typically damage the critical UPS load. The design of a DC/DC converter

capable of maintaining a high degree of UPS reliability, particularly at medium- and high-power levels, is a factor that weighs against its use in an UPS application. High-frequency DC/DC converters in medium- and high-power UPS systems aggravate the problem of suppressing EMI generated by the UPS making the use of the solid-state UPS in an EMI-sensitive environment less attractive.

1.2.6 Space Vector PWM

A different approach to three-phase PWM waveform generation is based on the representation of inverter output voltages in the two-phase stationary frame of reference. The transformation from a three-phase to two-phase frame of reference is described by the matrix equation

$$\begin{bmatrix} e_d \\ e_q \end{bmatrix} = \sqrt{(2/3)} \cdot \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \quad (1.22)$$

where $\{e_a, e_b, e_c\}$ are the three-phase components, and $\{e_d, e_q\}$ are the corresponding two-phase components. Assuming ideal three-phase inverter bridge switching, there are seven possible normalized inverter output voltage states where a normalized inverter output voltage state is defined either in terms of the inverter output phase-to-ground voltages, $[v_{ai}/V_{dc}, v_{bi}/V_{dc}, v_{ci}/V_{dc}]$, or in terms of the inverter output phase-to-phase voltages, $[v_{abi}/V_{dc}, v_{bci}/V_{dc}, v_{cai}/V_{dc}]$. A functional relationship between the normalized inverter output voltage states and the signals controlling the conduction state of the semiconductor switches in the

inverter bridge circuit is based on the operation of the three inverter leg circuits given the ideal inverter switching assumption. A typical inverter leg circuit is depicted in figure 1.4. In figure 1.5, a switching logic table describing the inverter leg circuit operation assuming ideal switching is depicted. In figure 1.6, the eight possible three-phase inverter bridge conduction states assuming ideal switching are depicted. In the UPS inverter system, a normalized inverter output voltage state in the form of $[v_{abi}/V_{dc}, v_{bci}/V_{dc}, v_{cai}/V_{dc}]$ is used. The three-phase vector $[v_{abi}, v_{bci}, v_{cai}]^T$ is converted to the vector $[V_{ds}^{(k)}, V_{qs}^{(k)}]^T$ in the stationary d-q frame. $[V_{ds}^{(k)}, V_{qs}^{(k)}]^T$ is described in each of the eight inverter conduction states, $1 \leq k \leq 8$, by the following set of equations:

$$\text{State 1: } V_{ds}^{(1)} = \sqrt{3/2} \cdot V_{dc} \quad V_{qs}^{(1)} = 1/\sqrt{2} \cdot V_{dc} \quad (1.23a)$$

$$\text{State 2: } V_{ds}^{(2)} = 0 \quad V_{qs}^{(2)} = \sqrt{2} \cdot V_{dc} \quad (1.23b)$$

$$\text{State 3: } V_{ds}^{(3)} = -\sqrt{3/2} \cdot V_{dc} \quad V_{qs}^{(3)} = 1/\sqrt{2} \cdot V_{dc} \quad (1.23c)$$

$$\text{State 4: } V_{ds}^{(4)} = -\sqrt{3/2} \cdot V_{dc} \quad V_{qs}^{(4)} = -1/\sqrt{2} \cdot V_{dc} \quad (1.23d)$$

$$\text{State 5: } V_{ds}^{(5)} = 0 \quad V_{qs}^{(5)} = -\sqrt{2} \cdot V_{dc} \quad (1.23e)$$

$$\text{State 6: } V_{ds}^{(6)} = \sqrt{3/2} \cdot V_{dc} \quad V_{qs}^{(6)} = -1/\sqrt{2} \cdot V_{dc} \quad (1.23f)$$

$$\text{State 7: } V_{ds}^{(7)} = 0 \quad V_{qs}^{(7)} = 0 \quad (1.23g)$$

$$\text{State 8: } V_{ds}^{(8)} = 0 \quad V_{qs}^{(8)} = 0 \quad (1.23h)$$

where V_{ds} is the d-axis inverter phase-to-phase output voltage, and V_{qs} is the q-axis inverter phase-to-phase output voltage. The normalized inverter output voltage vector, $[V_{ds}^{(k)}/V_{dc}, V_{qs}^{(k)}/V_{dc}]^T$, is commonly referred to as a space vector.

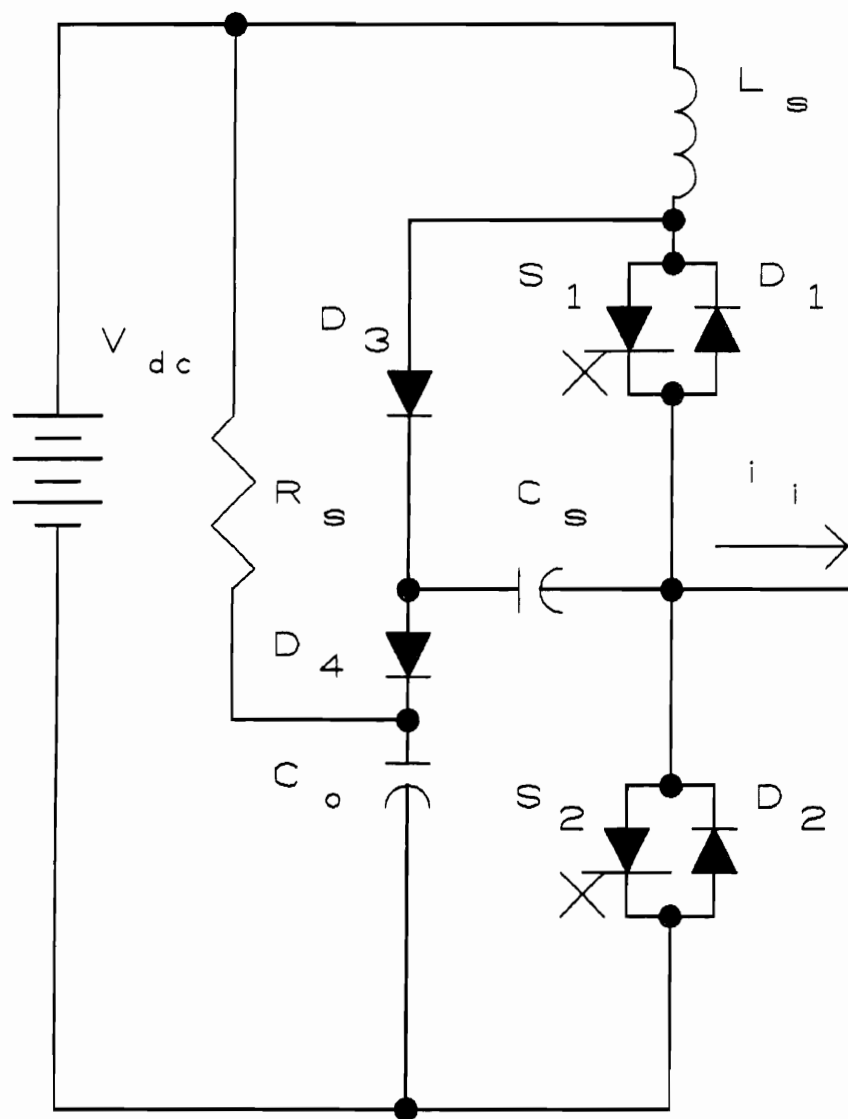


FIGURE 1.4: INVERTER LEG CIRCUIT

S1 STATE	S2 STATE	LOAD CURRENT SENSE	OUTPUT VOLTAGE SENSE
0	0	0	?
1	0	0	1
0	1	0	-1
0	0	1	-1
1	0	1	1
0	1	1	-1
0	0	-1	1
1	0	-1	1
0	1	-1	-1

0 => SWITCH OFF OR ZERO LOAD CURRENT
 1 => SWITCH ON, POSITIVE OUTPUT VOLTAGE OR LOAD CURRENT SENSE
 -1 => NEGATIVE OUTPUT VOLTAGE OR LOAD CURRENT SENSE

FIGURE 1.5: INVERTER LEG CIRCUIT SWITCHING LOGIC TABLE

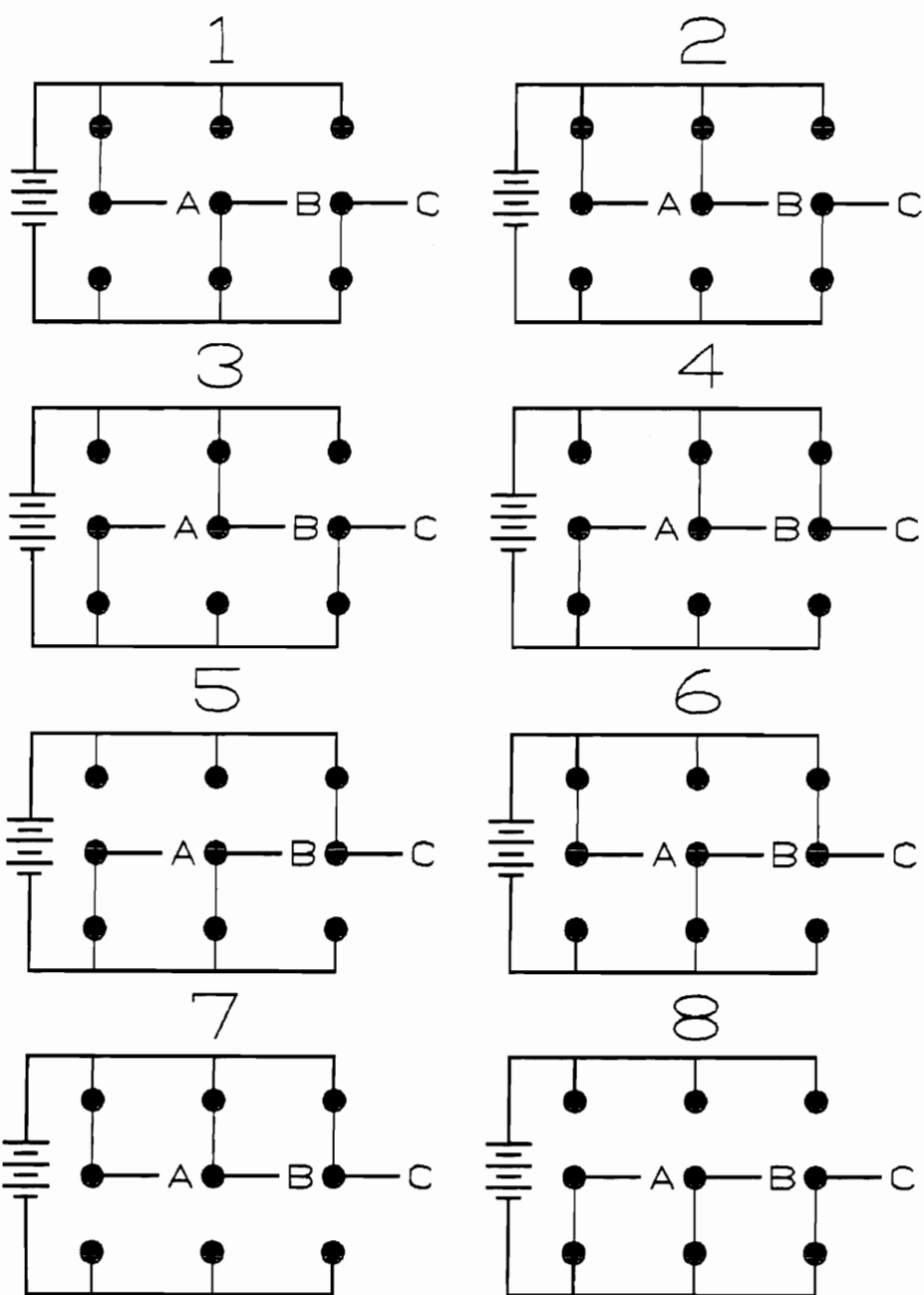


FIGURE 1.6: INVERTER SWITCHING STATES

A typical method of generating three-phase PWM inverter switching patterns based on the use of space vectors uniformly samples a normalized inverter output voltage command vector $\underline{v}_i^*(t)/V_{dc} = [v_{di}^*(t)/V_{dc}, v_{qi}^*(t)/V_{dc}]^T$ at $1/2f_s$ intervals, and approximates this sampled vector by a sequence of three space vectors having a time average value in the sampling interval that equals the sampled value of $\underline{v}_i^*(t)/V_{dc}$. The implementation of this method is described in the following section. In reference [14], an alternate space vector PWM method utilizing non-uniform command vector sampling is described where the inverter switching pattern is optimized with respect to minimizing the inverter output RMS harmonic current.

The description of three-phase inverter bridge switching in terms of space vectors enables the application of closed-loop control techniques where the inverter switching pattern is generated directly by the states of the inverter's load circuit. For example, the inverter's load circuit is described by a state equation of the form

$$d\underline{x}(t)/dt = \underline{A} \cdot \underline{x}(t) + \underline{B} \cdot \underline{u}(t) \quad (1.24)$$

and the corresponding state transition equation

$$\underline{x}(t) = \underline{\Phi}(t, t_0) \cdot [\underline{x}(t_0) + \int_{t_0}^t \underline{\Phi}(t_0, \tau) \cdot \underline{B} \cdot \underline{u}(\tau) d\tau] \quad (1.25)$$

where

$$\underline{\Phi}(t, t_0) = \exp \left[\int_{t_0}^t \underline{A} d\tau \right],$$

$\underline{x}$ is the vector of load circuit states, and $\underline{u}$ is the vector of inverter output voltages and inputs to the load circuit, τ is an integration variable. When the three-phase variables in equation (1.25) are transformed to the stationary d-q frame and normalized with respect to V_{dc} , the vector $\underline{u}/V_{dc}$ may be described in terms of inverter switching space vectors. Given a load circuit state command vector $\underline{x}^*(t)$, a controller is constructed where sequences of inverter switching space vectors are selected based on the commanded load circuit state transition and on the criterion applied to the error $[\underline{x}^*(t) - \underline{x}(t)]$. For example, the selection of a switching space vector in the interval $t_0 \leq t \leq (t_0 + \sigma)$ can be based on satisfying the condition

$$[\underline{x}^*(t) - \underline{x}(t)] \cdot (d/dt) \cdot [\underline{x}^*(t) - \underline{x}(t)] < 0 \text{ at } t = t_0 + \sigma.$$

1.2.7 Space Vector PWM Implementation

References [15] and [16] describe an implementation based on the modulation/carrier waveform comparison technique associated with carrier PWM where the modulation waveform $v_i^*(t)$ is described by the equation

$$\begin{aligned} v_i^*(t) &= 2 \cdot M \cdot \sin(2\pi \cdot f_m \cdot t) && \text{for } 0^\circ \leq 2\pi \cdot f_m \cdot t \leq 30^\circ \\ &= (2/\sqrt{3}) \cdot M \cdot \sin(2\pi \cdot f_m \cdot t + 30^\circ) && \text{for } 30^\circ \leq 2\pi \cdot f_m \cdot t \leq 90^\circ \end{aligned} \quad (1.26)$$

M is the modulation index. The modulation waveform is assumed to have quarter- and half-wave symmetry. Carrier PWM techniques which are designed to control the operation of a single inverter leg circuit

require three modulation circuits in a three-phase inverter application. In contrast, space vector PWM can be implemented with a single modulation circuit (microprocessor-based) where a single space vector modulation command $\{v_{di}^*, v_{qi}^*\}$ is required for three-phase inverter switching control. This implementation conforms readily to the structure of three-phase PWM controllers based on two-phase reference frames (which includes the majority of variable-speed AC motor controllers).

The least complex space vector PWM implementation derives inverter switching patterns based on the solution of the vector equation

$$2 \cdot f_s \cdot \int_{n/f_s}^{(n + \frac{1}{2})/f_s} \underline{v}_i(\Omega) d\Omega = \underline{\phi}_i^* \quad (1.27)$$

where the command vector $\underline{\phi}_i^* = \{\phi_{di}^*, \phi_{qi}^*\}$, and a $1/2f_s$ asymmetric regular sampling interval are used. The left-hand side of equation (1.27) is equal to the mean inverter output voltage in the n^{th} $1/2f_s$ interval. In each $1/2f_s$ interval (or modulation sample interval), the inverter output waveform vector $\underline{v}_i(t)$ is generated from the execution of three inverter switching states where one of the three states is the zero vector (either state 7 or 8).

The selection of switching states is based on the premise that inverter switches change state only once during a modulation sample interval. This switching agenda dictates the selection of inverter switching states in the following manner:

- i.) States 7 and 8 are executed in an alternating fashion each modulation sample interval.
- ii.) Assuming the zero vector is executed first in a modulation sample

period, the order of the remaining two switching states is determined by the zero vector executed. For example, if state 7 is executed in an interval, then the next switching state executed will have two switches connected to the positive DC rail. The last switching state will have two switches connected to the negative DC rail.

iii.) Given the command vector $\underline{v}_i^*$, the selection of the two non-zero vectors corresponds to the two vectors adjacent to the command vector. Of course, if $\underline{v}_i^*$ coincides with a non-zero vector, then the execution of the second non-zero vector does not occur.

The algorithm for the selection of the zero vector is simple. Selection of the non-zero vectors is somewhat more complex. The following algorithm is an example of how the selection process would be implemented:

```

if  $v_{di}^* = 0$  and  $v_{qi}^* = 0$ 
    then use only states 7 or 8
if  $v_{di}^* \geq 0$  and  $v_{qi}^* \geq 0$ 
{
    and if  $v_{qi}^* \geq v_{di}^* / \sqrt{3}$ 
        then use states 1 and 2
    else
        use states 1 and 6
}
if  $v_{di}^* \leq 0$  and  $v_{qi}^* \geq 0$ 

```

```

{
    and if  $v_{qi}^* \geq |v_{di}^*|/\sqrt{3}$ 
        then use states 2 and 3
    else
        use states 3 and 4
}

if  $v_{di}^* \leq 0$  and  $v_{qi}^* \leq 0$ 
{
    and if  $|v_{qi}^*| \geq |v_{di}^*|/\sqrt{3}$ 
        then use states 4 and 5
    else
        use states 3 and 4
}

if  $v_{di}^* \geq 0$  and  $v_{qi}^* \leq 0$ 
{
    and if  $|v_{qi}^*| \geq |v_{di}^*|/\sqrt{3}$ 
        then use states 5 and 6
    else
        use states 1 and 6
}

```

Based on the relative absolute magnitudes of the d- and q-axis commands, the position of the command vector is determined. Once the three switching vectors have been identified, the corresponding switching state times are computed. Equation (1.27) is reduced to the vector equation

$$2 \cdot f_s \cdot [\underline{v}_s^{(k)} \cdot t_k + \underline{v}_s^{(k+1)} \cdot t_{k+1}] = \underline{v}_i^* \quad (1.28)$$

where $\underline{v}_s^{(k)}$ and $\underline{v}_s^{(k+1)}$ are the adjacent non-zero vectors, and, t_k and t_{k+1} are the corresponding state times. t_k and t_{k+1} are solutions of the pair of equations obtained from the previous vector equation. Finally, the zero vector state time t_0 is the result of the difference equation

$$t_0 = (1/2f_s) - t_k - t_{k+1}. \quad (1.29)$$

To illustrate the space vector PWM technique described previously, the following example is given. In the interval $t_0 \leq t \leq t_0 + 1/2f_s$, an inverter output phase-to-phase sinusoidal voltage command vector is defined as

$$\begin{aligned} \underline{v}_i^*(t_0) &= \{v_{di}^*(t_0), v_{qi}^*(t_0)\} \\ &= \{A \cdot \sin(2\pi \cdot f_m \cdot t_0) \cdot V_{dc}, -A \cdot \cos(2\pi \cdot f_m \cdot t_0) \cdot V_{dc}\} \end{aligned}$$

where $0 \leq A \leq \sqrt{2}$. t_k and t_{k+1} are solutions to the set of equations corresponding to equation (1.28)

$$\begin{aligned} t_k &= (1/2f_s) \cdot (1/\sqrt{3}) \cdot [v_{di}^*(t_0) \cdot v_{qs}^{(k+1)} - v_{qi}^*(t_0) \cdot v_{ds}^{(k+1)}], \\ t_{k+1} &= (1/2f_s) \cdot (1/\sqrt{3}) \cdot [v_{qi}^*(t_0) \cdot v_{ds}^{(k)} - v_{di}^*(t_0) \cdot v_{qs}^{(k)}]. \end{aligned}$$

Selecting $A = 1$, $2\pi \cdot f_m \cdot t_0 = \pi/6$, and $V_{dc} = 1$, the adjacent switching state vectors are

$$\begin{aligned} v_{ds}^{(k)} &= v_{ds}^{(5)} = 0, & v_{qs}^{(k)} &= v_{qs}^{(5)} = -\sqrt{2}, \\ v_{ds}^{(k+1)} &= v_{ds}^{(6)} = \sqrt{3}/2, & v_{qs}^{(k+1)} &= v_{qs}^{(6)} = -1/\sqrt{2}, \end{aligned}$$

t_k , t_{k+1} , and t_0 are computed as $t_k = 0.229 \cdot (1/2f_s)$, $t_{k+1} = 0.408 \cdot (1/2f_s)$, and $t_0 = 0.363 \cdot (1/2f_s)$.

In a practical implementation, the tasks of selecting adjacent non-zero switching state vectors, the zero switching state vector, and the computation of t_k , t_{k+1} , and t_0 may be simplified by the pre-

computation of space vector PWM parameters and storage of these parameters in Read-Only Memory (ROM). The phase-to-phase inverter output voltage command vector $\underline{v}_i^*$ is normalized with respect to the value $\sqrt{2} \cdot V_{dc}$. In a fixed-point digital controller, the normalized $\underline{v}_i^*$ is represented by a signed integer number where the normalized v_{di}^* and v_{qi}^* are concatenated to form the lower and upper nibbles, bytes, or words of this signed number. For example, a 16-bit signed value for a normalized $\underline{v}_i^*$ would be represented by integer values in the range $\{-32767, 32768\}$. The corresponding normalized v_{di}^* and v_{qi}^* would be 8-bit signed values represented by integer values in the range $\{-128, 127\}$. A command vector in this format would generate the ROM address of the stored space vector PWM parameters corresponding to $\underline{v}_i^*$. ROM data would be in the form of t_k and t_{k+1} normalized with respect to $1/2f_s$, and the three-phase inverter bridge switching commands corresponding to the switching vectors $\underline{v}_s(k)$ and $\underline{v}_s(k+1)$. A ROM addressed by a 16-bit wide integer will store space vector PWM parameters corresponding to 65536 discrete normalized $\underline{v}_i^*$ vector commands where the normalized v_{di}^* and v_{qi}^* are discretized in terms of 256 uniformly incremented levels in the range $\{-1, 127/128\}$. The use of pre-computed space vector PWM parameters stored in ROM reduces the computational overhead involved with the use of equation (1.28) and the process of logically selecting $\underline{v}_s(k)$ and $\underline{v}_s(k+1)$.

Sinusoidal space vector PWM, like suboptimal carrier PWM techniques (modulation waveform with added third harmonic distortion), generates a maximum line-to-line fundamental frequency voltage in the

linear control range which is 15% higher than the conventional carrier PWM technique.

1.2.8 An Evaluation of Sinusoidal PWM Techniques

A basis for the evaluation of optimized switching strategies, carrier, and space vector PWM techniques with regard to their application in UPS inverters is the output filter circuit design required to either eliminate or suppress inverter output voltage harmonics. Typical UPS load voltage specifications limit the total load voltage harmonic distortion to 5% with a 3% limit on distortion due to individual harmonics. Conservative output filter circuit design methods such as the computation of the $L_f \cdot C_f$ product based on

$$|1/(1 - 12\pi^2 \cdot f_c^2 \cdot L_f \cdot C_f)| \leq 0.03,$$

f_c is the frequency of the lowest inverter output waveform harmonic, result in a filter design where nearly distortion-free load voltage waveforms at no-load are generated.

L_f is selected based on minimizing the volt-amperes of the filter inductor due to both the fundamental and harmonic currents. This volt-ampere value is defined by the expression

$$\frac{1}{2} \cdot \sum_{k=1}^{\infty} v_i^2(k \cdot f_m) / (k \cdot 2\pi \cdot f_m \cdot L_f) + I_1^2(f_m) \cdot 2\pi \cdot f_m \cdot L_f \quad (1.30)$$

where $v_i(k \cdot f_m)$ is the k^{th} harmonic inverter line-to-line output voltage, $I_1(f_m)$ is the RMS fundamental inductor current, f_m is the fundamental frequency, and n is an integer corresponding to the harmonic present in the inverter output voltage waveform. $I_1(f_m)$ is the RMS value

of the load and filter capacitor fundamental currents. L_f resulting in the minimum inductor volt-amperes is computed from the expression

$$L_f = \{1/[(\sqrt{2} \cdot 2\pi \cdot f_m \cdot I_1(f_m))]\} \cdot \sqrt{\sum_{k=1}^{\infty} v_i^2(k \cdot f_m)/k} \quad (1.31)$$

derived by equating the derivative of (1.30) with respect to L_f to zero and solving for L_f .

Some design flexibility exists with regard to the gain factor assigned to $|1/(1 - 12\pi^2 \cdot f_c^2 \cdot L_f \cdot C_f)|$. Though the assignment of a 0.03 gain factor results in a relatively distortion-free load voltage waveform given no-load conditions, this is done at the expense of reducing the filter pole frequency. In order to reduce the adverse effects of unbalanced UPS loads, it is desirable to reduce the filter impedance magnitude $2\pi \cdot f \cdot L_f/[1 - 12\pi^2 \cdot f_c^2 \cdot L_f \cdot C_f]$ at the fundamental ($f = f_m$). Assigning a higher gain factor to $|1/(1 - 12\pi^2 \cdot f_c^2 \cdot L_f \cdot C_f)|$ is necessary to accomplish this objective.

Inverter output line-to-line voltage waveform spectral information at a fixed modulation index corresponding to

- i.) asymmetric regular sampled sinusoidal carrier PWM,
- ii.) asymmetric regular sampled sinusoidal carrier PWM with 25% added third-harmonic distortion,
- iii.) space vector PWM, and
- iv.) an optimized switching strategy described in [6]

is depicted in figures 1.7a through 1.7d. Fundamental and harmonic magnitudes are normalized with respect to the inverter DC power supply voltage. A 0.75 modulation index operating point and an inverter

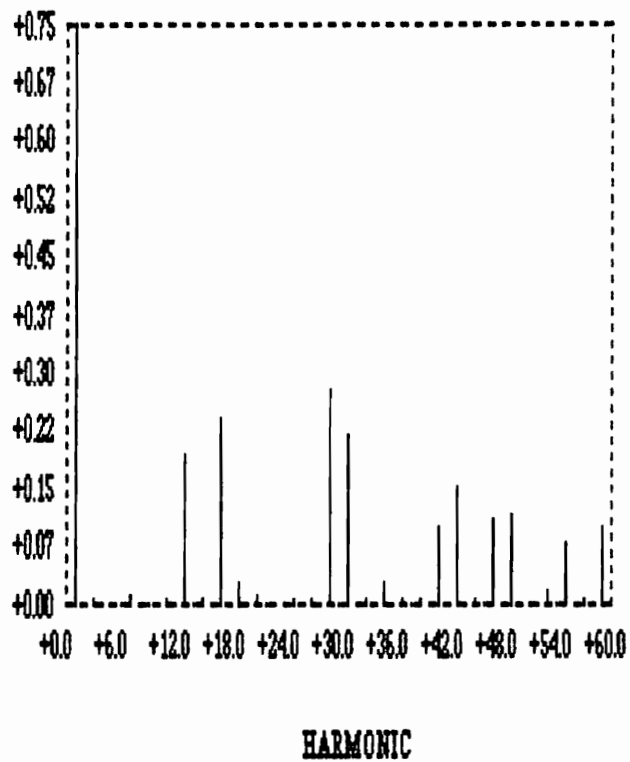


Figure 1.7a: Inverter Output Line-to-Line Voltage Waveform Spectral Information, Asymmetric Regular Sampled Sinusoidal PWM Technique, Magnitudes Normalized with Respect to the Inverter DC Power Supply Voltage.

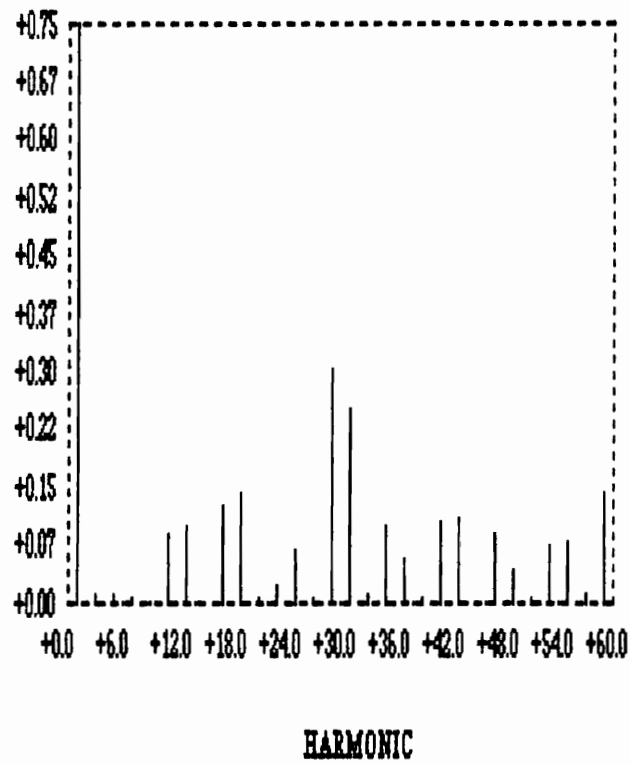


Figure 1.7b: Inverter Output Line-to-Line Voltage Waveform Spectral Information, Asymmetric Regular Sampled Sinusoidal PWM with 25% Added Third-Harmonic Distortion, Magnitudes Normalized with Respect to the Inverter DC Power Supply Voltage.

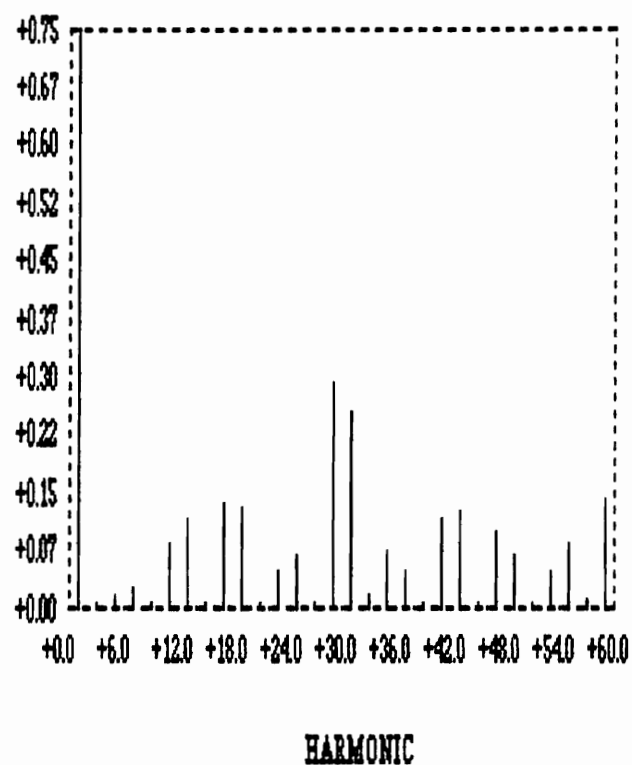


Figure 1.7c: Inverter Output Line-to-Line Voltage Waveform Spectral Information, Space Vector Sinusoidal PWM Technique, Magnitudes Normalized with Respect to the Inverter DC Power Supply Voltage.

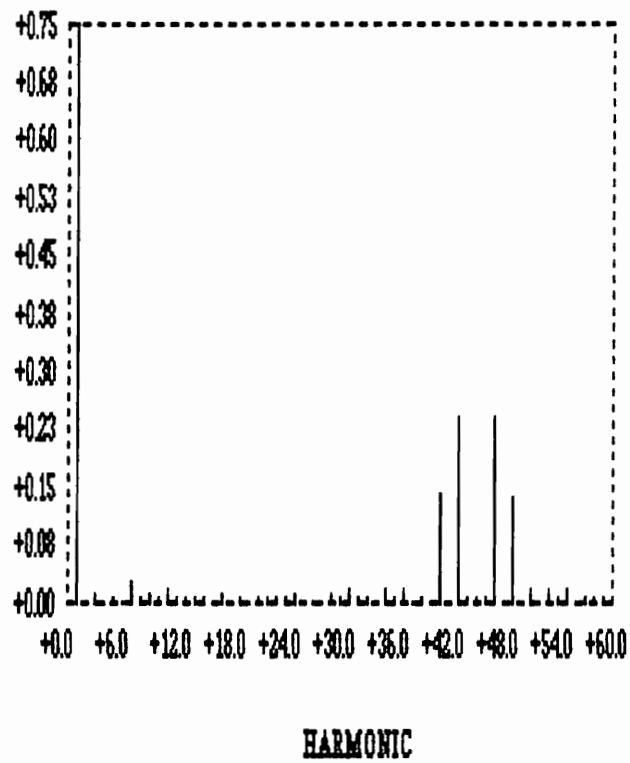


Figure 1.7d: Inverter Output Line-to-Line Voltage Waveform Spectral Information, Optimized Fixed-Pattern Sinusoidal PWM Technique, Magnitudes Normalized with Respect to the Inverter DC Power Supply Voltage.

switching frequency equal to $15 \cdot f_m$ were chosen for the carrier and space vector PWM waveform generation processes. In figures 1.8a through 1.8c, plots of

$$\sqrt{\sum_{k=1}^{\infty} v_{in}^2(k \cdot f_m)/k}$$

verses the corresponding modulation index command are depicted for inverter output line-to-line voltages generated by

- i.) asymmetric regular sampled sinusoidal carrier PWM,
- ii.) asymmetric regular sampled sinusoidal carrier PWM with 25% added third-harmonic distortion,
- iii.) space vector PWM,

where $v_{in}(k \cdot f_m)$ is the inverter output line-to-line k^{th} harmonic voltage normalized with respect to v_{dc} .

The waveform spectral characteristics associated with the asymmetric regular sampled sinusoidal carrier PWM process with 25% added third-harmonic distortion and the space vector PWM process are similar. Further comparisons of these two PWM processes at higher modulation index values reveal the fact that this similarity is maintained. This observation is not unexpected given the basis for the development of this techniques. The development of suboptimal carrier PWM techniques described in [11] and [12] is based on the reduction of the term

$$\sqrt{\sum_{k=1}^{\infty} v_i^2(k \cdot f_m)/k}$$

which is proportional to the total inverter output harmonic current distortion or THD. This reduction is observed in comparison with the THD

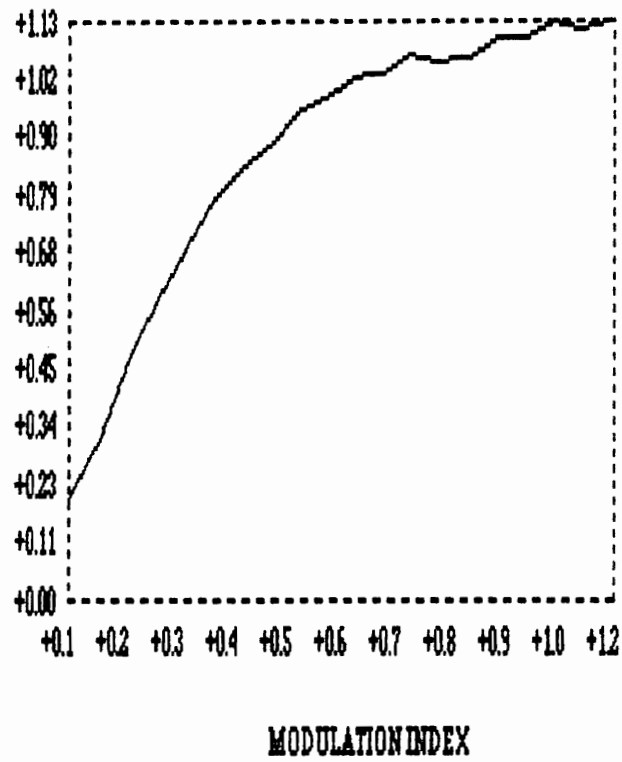


Figure 1.8a: Plot of $\sqrt{\sum_{k=1}^{\infty} v_{in}^2(k \cdot f_m)/k}$ (x10 Scale) Verses the Modulation Index Command Given to the Asymmetric Regular Sampled Sinusoidal PWM Waveform Generation Process.

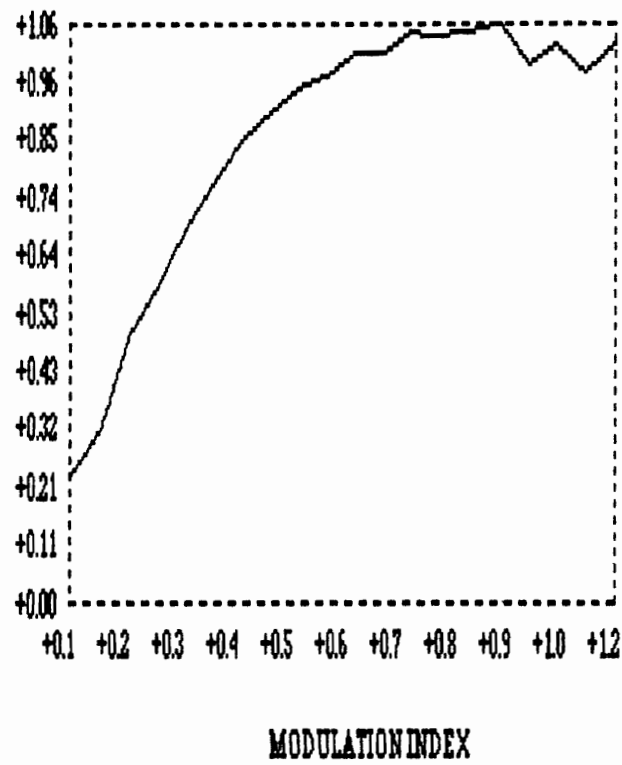


Figure 1.8b: Plot of $\sqrt{\sum_{k=1}^{\infty} v_{in}^2(k \cdot f_m)/k}$ (x10 Scale) Verses the Modulation Index Command Given to the Asymmetric Regular Sampled Sinusoidal PWM Waveform Generation Process with 25% Added Third-Harmonic Distortion.

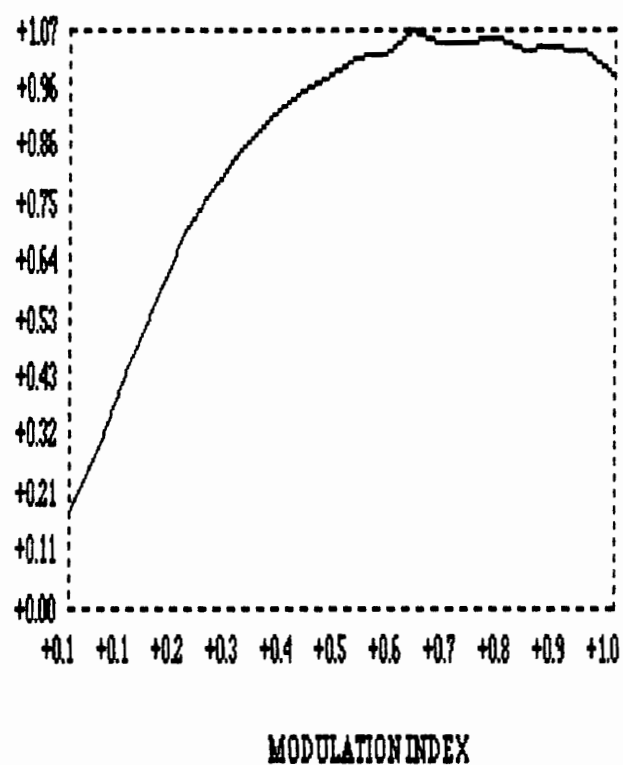


Figure 1.8c: Plot of $\sqrt{\sum_{k=1}^{\infty} v_{in}^2(k \cdot f_m)/k}$ (x10 Scale) Verses the Modulation Index Command Given to the Space Vector Sinusoidal PWM Waveform Generation Process.

given the use of the asymmetric regular sampled sinusoidal carrier PWM process without added third-harmonic distortion. Space vector PWM (described in [15]) generates inverter switching states and corresponding state times based on minimizing within $1/2f_s$ intervals the integral

$$2 \cdot f_s \cdot \int_{t_0}^{t_0 + 1/2f_s} [\underline{v}_i(\Omega) - \underline{v}_i^*(\Omega)] d\Omega$$

where $\underline{v}_i$ and $\underline{v}_i^*$ are the actual and command inverter output voltage space vectors respectively. This term is proportional to the inverter output current deviation in the interval $\{t_0, t_0 + 1/2f_s\}$ where this deviation is the result of harmonic current flow through the inductance connecting the inverter output to the remainder of the inverter load circuitry (the filter inductance L_f in an UPS inverter). Minimization of this integral is equivalent to the minimization of the inverter output current THD

$$\sqrt{\sum_{k > 1}^{\infty} v_{i1}^2(k \cdot f_m)/k}.$$

Using either the suboptimal carrier PWM technique or space vector PWM, the output filter L_f value derived using (1.31) may be reduced in comparison to the use of the conventional carrier sinusoidal PWM technique. From the waveform spectral data in figure 1.7a, the conventional sinusoidal PWM technique produces the result

$$\sqrt{\sum_{k > 1}^{61} v_{iN}^2(k \cdot f_m)/k} = 0.109$$

where this value was computed using harmonic data up to the 61th harmonic. In contrast, the suboptimal carrier and space vector PWM spectral data produce the result

$$\sqrt{\sum_{k=1}^{61} v_{in}^2(k \cdot f_m)/k} = 0.104.$$

A 4.6% smaller filter inductor value may be specified using either the suboptimal carrier or space vector PWM technique. This reduction in the value of L_f reduces the adverse effects of unbalanced UPS loads on the load voltage waveforms by reducing the filter impedance magnitude at the fundamental.

A further reduction in the filter inductor value is achieved with the use of the optimized switching strategy. The switching strategy is a derivative of the sinusoidal carrier PWM process with added third-harmonic distortion where inverter fundamental voltage control is achieved by controlling the inverter DC power supply voltage. The mean inverter switching frequency (mean value evaluated over the inverter fundamental period) using this PWM technique is approximately two-fifths the carrier frequency. The lowest-order inverter output waveform harmonics are located at $f_s - 4 \cdot f_m$, $f_s - 2 \cdot f_m$, $f_s + 2 \cdot f_m$, and $f_s + 4 \cdot f_m$ where f_s is the carrier frequency and f_m is the inverter fundamental frequency. Using a carrier frequency of $45 \cdot f_m$ and an inverter DC power supply voltage which is 65% of the value used in the carrier and space vector PWM waveform generation processes, the result

$$\sqrt{\sum_{k=1}^{121} v_{in}^2(k \cdot f_m)/k} = 0.060$$

is obtained where this value was computed using harmonic data up to the 121th harmonic. A 45% filter inductor value reduction is realized compared to the conventional carrier PWM technique. The mean inverter switching frequency is approximately $18 \cdot f_m$. Compared to this result, the inductor value reduction achieved using either the suboptimal carrier or space vector PWM technique is modest. Additionally with an inverter switching frequency which is in the proximity of the switching frequency used in the carrier and space vector PWM techniques, the lowest-order inverter output waveform harmonic is at $41 \cdot f_m$ compared to $11 \cdot f_m$ in either the suboptimal carrier or space vector PWM waveforms enabling the output filter pole frequency to be increased substantially further reducing the filter impedance magnitude.

In comparison to the carrier and space vector PWM techniques, the use of the optimized switching strategy results in an output filter design which is superior with respect to reducing the adverse effects of unbalanced and nonlinear loads. The cost of implementing the optimized switching strategy is the addition of a DC/DC converter between the battery and inverter DC input. Despite the fact that the development of DC/DC conversion techniques is well advanced, the addition of semiconductor power circuitry to an UPS design is not easily implemented given concerns regarding system reliability and the generation of additional electromagnetic interference (particularly at high inverter output power levels). Another concern (discussed in the next section) is inverter switching distortion due to switching delays, dead-time insertion, and power semiconductor conduction voltage drops. Harmonic

components on the order of the 5th, 7th, 11th, ... are added to the inverter output line-to-line voltage waveform as a result of this distortion effect. The magnitude of these components is dependent on the inverter switching period. Though the technique in reference [6] has a mean switching frequency which is approximately two-fifths the carrier frequency, portions of the inverter output waveform are generated by inverter switching at the carrier frequency increasing the magnitude of the harmonic components due to switching distortion. The added inverter output voltage harmonic distortion is either tolerated or the filter pole frequency is reduced in order to attenuate sufficiently the low-order harmonics generated by inverter switching distortion (a move which is counter-productive with regard to reducing the filter impedance magnitude). Reduction of inverter switching distortion effects requires reductions in the switching delays and dead-time insertion implying the use of power transistor semiconductor technology (bipolar, field-effect, insulated-gate bipolar). The application of this technology is dependent on the inverter circuit current and voltage stresses (inverter output power level) and system reliability considerations. An additional constraint applied to the use of a specific semiconductor technology is the environment in which the UPS operates. It is not uncommon for UPS systems to operate in environments which are not as well controlled in terms of temperature and humidity as in the case of data-processing centers. For example, a 50° C ambient air temperature in an uncontrolled environment is not uncommon in the specification of an UPS application. Dissipation of heat generated by semiconductor switching and conduction

losses is typically achieved by forced-air cooling, therefore, the maximum semiconductor current and switching frequency are functions of the thermal circuit characteristics of the semiconductor and heatsink given forced-air cooling. The operation of a power semiconductor at its maximum junction temperature is not desirable from the perspective of system reliability, and so, the actual semiconductor current and switching frequency limits are more conservative.

Despite their inferior open-loop spectral characteristics compared to an optimized switching strategy, the consideration of either the suboptimal carrier or space vector PWM technique remains a valid alternative given practical limitations associated with high-power three-phase UPS system design. Compensation for unbalanced and nonlinear UPS load-induced output voltage waveform distortion using either of these techniques becomes an objective associated with the design of the closed-loop load voltage controller.

Given that the spectral performance with sinusoidal modulation of the suboptimal carrier and space vector PWM techniques are similar, the choice of either technique is based on its application in a closed-loop control scheme. High-performance three-phase inverter controllers are microprocessor- or DSP-based systems typically constructed on a two-phase frame of reference (such as the stationary d-q frame). The space vector PWM process may be defined in the same reference frame, and so, it can be incorporated into the controller software without the necessity of performing the additional conversion from the d-q frame to a three-phase frame (required in the case of implementing a carrier PWM

technique in conjunction with a controller constructed in the d-q frame). This advantage of space vector PWM is more readily appreciated in connection with variable-frequency three-phase inverters where the process of converting d-q values to three-phase values is dependent on the inverter modulation frequency value.

The principal stimulus for the use of space vector PWM in a fixed-frequency (or static) three-phase inverter is the fact the inverter switching function (in the form of discrete switching states and corresponding state time intervals) and the remainder of the inverter control loop (the output filter circuit and the inverter controller) may be described directly in the same frame of reference. UPS system currents, voltages, and associated commands (such as the load voltage commands) may be described in intervals corresponding to the inverter switching state time intervals in the form of sampled d-q values. This method of describing the inverter/filter/load system has enabled the development of an adaptive intelligent inverter controller described in detail in later sections which is constructed without reference to either the inverter modulation or switching frequencies. An inverter controller of this type may be applied without design modifications in different static three-phase inverter applications where inverter modulation and switching frequencies vary with the application.

1.2.9 Software Inverter Simulations Using Open-Loop Carrier and Space Vector Sinusoidal PWM Techniques

Inverter/load circuit simulations were run in open-loop using the carrier and space vector sinusoidal PWM techniques examined in the previous section operating at the same inverter DC power supply and load voltages. The simulation software is described in appendices 1 through 5.

A common balanced passive load model is devised for the comparison of sinusoidal modulation methods. See Appendix 3 for more detail. Comparisons are made with an unity power factor load. With a 900 Hertz inverter switching frequency f_s selected to reflect GTO inverter operation and a 60 Hertz inverter modulation frequency f_m , the lowest order inverter output voltage harmonic generated by these modulation processes will be at 660 Hertz or $f_c = 11 \cdot f_m$. Refer to Appendix 4 for details of the emulation of these modulation processes in the simulation. A 0.75 modulation index operating point was chosen. The modulation index was chosen based on the following considerations:

- i.) The DC supply is a 180-cell lead-calcium battery with a float voltage of 405 Volts DC and an end-voltage of 315 Volts DC. Sufficient range is allocated to the linear control of the fundamental output voltage to compensate for the DC power supply voltage variation.
- ii.) A 15% inverter output overvoltage adjustment is desired.

A GTO is employed as the switch in the inverter model. Refer to Appendix 1 for more detail on the switch and diode models used in the simulation. Device specifications pertinent to the operation of this switch in the simulation are given as follows:

- i.) Turn-on time = 4 μ s,
- ii.) Turn-off time = 8 μ s,
- iii.) Switch on-state voltage = 3.6 Volts.

Additionally, a dead-time interval of 20 μ s is inserted in the PWM switch control signals. With a simulation time resolution of 8.333 μ s, the effects of diode reverse recovery and snubber circuit operation on the inverter leg circuit output waveform is not expected to be observed. This simulation time resolution is chosen for the purpose of obtaining waveform information per fundamental period within a reasonable length of time. Appendix 5 describes the operation of the inverter/load simulation software in detail. A 7.003 ohm line-to-line load resistive is used in the passive load model. The output filter design utilizes the filter component values $L_f = 325 \mu$ H and $C_f = 540 \mu$ F where L_f is computed from (1.31) using the spectral characteristics of the inverter output line-to-line voltage waveform generated by the space vector PWM technique. The filter pole frequency is placed to avoid the proximity of the 5th harmonic generated by inverter switching distortion while providing sufficient attenuation of the inverter output waveform harmonics. Each inverter/load simulation was run until the simulation

information reached steady-state values. Simulation results are given in tables 1.1 through 1.3.

As anticipated from the output filter L_f computations in the previous section, the suboptimal carrier and space vector PWM techniques have lower inverter output current distortion values compared to conventional sinusoidal carrier PWM. Consistent with other comparisons of PWM techniques, for example [17], this tabulation of simulation results includes a inverter DC power supply input current total harmonic distortion value for each simulation. It should be noted that this figure of merit has bearing only with regard to the application of balanced passive loads.

Lower harmonic distortion values for the inverter DC power supply and output currents translate to lower heat losses due to harmonic current flow in the inverter power semiconductors. Based on the open-loop simulation results given a balanced passive load model, the use of suboptimal carrier and space vector PWM techniques should result in inverter operation which is more efficient in terms of power conversion. Additionally, lower inverter DC power supply RMS harmonic currents translate to lower harmonic current stresses in the DC power supply filter capacitors and the UPS battery.

Nonlinear loads cause an increase the inverter DC power supply and output RMS harmonic currents particularly when closed-loop, sub-cycle control of the load voltage waveforms is performed through the PWM inverter output voltage waveform process. Subsequently, reductions in RMS harmonic currents realized by the use of either suboptimal carrier

**Table 1.1: Asymmetric Regular Sampled Sinusoidal
PWM Simulation Results**

Modulation Process	=	Asymmetric Regular Sampled Sinusoidal PWM
Modulation Index	=	0.75
Modulation Frequency	=	60 Hertz
Inverter Switching Frequency	=	900 Hertz
DC Supply Voltage	=	405 Volts DC
DC Power Supply RMS Current	=	62.709 Amperes
<u>DC Power Supply Current Total Harmonic Distortion</u>		<u>89.6%</u>
GTO RMS Current	=	53.380 Amperes
Diode RMS Current	=	36.956 Amperes
Inverter Output RMS Current	=	98.167 Amperes
<u>Inverter Output Current Total Harmonic Distortion</u>		<u>34.5%</u>
Load Line-to-Line RMS Voltage	=	191.873 Volts
<u>Load Voltage Total Harmonic Distortion</u>	<u>=</u>	<u>2.5%</u>

**Table 1.2: Asymmetric Regular Sampled Sinusoidal
PWM With 25% Added Third-Harmonic
Distortion Simulation**

Modulation Process	=	Asymmetric Regular Sampled Sinusoidal PWM With 25% Added Third-Harmonic Distortion
Modulation Index	=	0.75
Modulation Frequency	=	60 Hertz
Inverter Switching Frequency	=	900 Hertz
DC Supply Voltage	=	405 Volts DC
DC Power Supply RMS Current	=	60.638 Amperes
<u>DC Power Supply Current Total Harmonic Distortion</u>	=	<u>88.9%</u>
GTO RMS Current	=	53.058 Amperes
Diode RMS Current	=	36.280 Amperes
Inverter Output RMS Current	=	96.067 Amperes
<u>Inverter Output Current Total Harmonic Distortion</u>	=	<u>32.2%</u>
Load Line-to-Line RMS Voltage	=	191.873 Volts
<u>Load Voltage Total Harmonic Distortion</u>	=	<u>2.3%</u>

Table 1.3: Space Vector Sinusoidal PWM Simulation

Modulation Process	=	Space Vector Sinusoidal PWM
Modulation Index	=	0.75
Modulation Frequency	=	60 Hertz
Inverter Switching Frequency	=	900 Hertz
DC Supply Voltage	=	405 Volts DC
DC Power Supply RMS Current	=	59.883 Amperes
<u>DC Power Supply Current Total Harmonic Distortion =</u>		<u>88.8%</u>
GTO RMS Current	=	52.942 Amperes
Diode RMS Current	=	36.529 Amperes
Inverter Output RMS Current	=	96.678 Amperes
<u>Inverter Output Current Total Harmonic Distortion =</u>		<u>32.4%</u>
Load Line-to-Line RMS Voltage	=	189.010 Volts
<u>Load Voltage Total Harmonic Distortion =</u>		<u>2.5%</u>

or space vector PWM can be made insignificant given nonlinear load currents with significant harmonic distortion.

1.3 Inverter Switching Distortion

An example of a voltage-source inverter leg circuit is depicted in figure 1.4 along with a table depicted in figure 1.5 describing the steady-state circuit operation given the semiconductor switch states and the load current sense. An assumption commonly made with regard to inverter switching is that switch operation is ideal, i.e., switching is instantaneous, no voltage drops during switch conduction, and no switch leakage currents when the switch is open. Additionally, it is assumed that the inverter's DC supply voltage is constant and without harmonic frequency voltage distortion. This assumption is useful with regard to simplifying the description of PWM inverter switching patterns and switching control techniques.

Actual waveform generation is influenced by the realistic behavior of power semiconductor devices, the operation of snubber circuitry, errors in the PWM control logic, and the inverter DC supply voltage characteristics. The deviation of actual inverter leg operation from the ideal model may be viewed in terms of inverter output voltage waveform distortion. Numerous research papers, [18] to [24], have been devoted to this topic.

The mechanisms responsible for waveform errors are briefly described as follows:

- i.) The operation of a semiconductor switch is delayed by turn-on (t_{dg}) and turn-off (t_{dq}) times. In devices such as SCR's, GTO's, bipolar transistors, and IGBT's the turn-off operation ends with a current-tailing effect. To avoid simultaneous conduction of the positive- and negative-rail switches, a dead-time interval t_D is inserted between the turn off of one switch and the turn-on of the other switch. The insertion of this dead time interval sets up the condition that the load current sense dictates the polarity of the inverter output voltage waveform (see table in figure 1.5). The delays associated with the semiconductor switch modify the length of the positive and negative level pulse widths where turn-on delay decreases the pulse width, and the turn-off delay increases the pulse width.
- ii.) The commutation of load current between the DC power supply rails is influenced by the snubber circuit design particularly with regard to the selection of the snubber inductance L_S . The rate of load current commutation is also influenced by the reverse recovery characteristics of diodes D_1 and D_2 . The slope of the output voltage waveform level transition has a finite value as a result.
- iii.) The output waveform amplitude is modulated by voltage drops across the semiconductor switches and diodes. DC power supply voltage distortion and variation contribute to the output waveform amplitude modulation.

Inverter output voltage waveform distortion associated with time delays and load current commutation may be viewed as the addition of voltage harmonics created by the deviations from the inverter output voltage waveform generated by ideal inverter switching. Inverter output voltage waveform amplitude modulation modifies the output voltage waveform spectral characteristics in a fashion described by the convolution of the ideal inverter output voltage Fourier Transform with the Fourier Transform of the function describing the amplitude modulation process.

1.3.1 Inverter Leg Switching with Positive-Sense Load Current

With reference to figure 1.4, the switching operation of the inverter leg circuit with load current flow out of the leg output node (positive-sense load current) is described as follows:

- i.) Inverter leg load current flows from the leg output node to the load circuit and switch S_1 is enabled (and so load current flows through S_1). The PWM control signal for S_1 goes logic 0 (switch disable). After a switch turn-off delay t_{dq} due to the time required to remove base (or gate) charge, S_1 is disabled. Load current flow is diverted to the snubber circuit components D_3 and C_S , and diode D_2 . In this interval of time, the leg output waveform makes a level transition from the positive to the negative. The slope of this transition is determined by the

snubber circuit component values, the turn-off characteristics of S_1 particularly with regard to the current tailing effect, and the turn-on characteristics of D_2 . After a fall time period, t_F , the leg output waveform level reaches its steady-state negative value where load current commutation via D_2 to the negative-rail is completed. Given the load current sense, the turn-on process of S_2 does not affect the output waveform level when D_2 carries the load current.

- ii.) Inverter leg load current flows from the leg output node to the load circuit and the PWM control signal to S_2 goes to logic 0 (switch disable). The leg load current continues to flow through D_2 (the output waveform level remains unchanged). After a dead-time delay, t_D , the PWM control signal for S_1 makes the transition from 0 to 1 (switch enable). After a turn-on delay t_{dg} , current flow through S_1 begins. The rate-of-rise of S_1 current is limited by the snubber inductance L_S , and in-turn, limits the rate of load current commutation from the negative to positive rail. After a rise time period, t_R , the leg output waveform level reaches its steady-state positive value (load current commutation via S_1 to the positive-rail is completed).

Figure 1.9 depicts the normalized ideal and actual leg output waveforms given positive-sense load current. The following observations are made regarding the waveforms depicted in this figure:

- i.) The generation of output waveforms generates waveform errors or

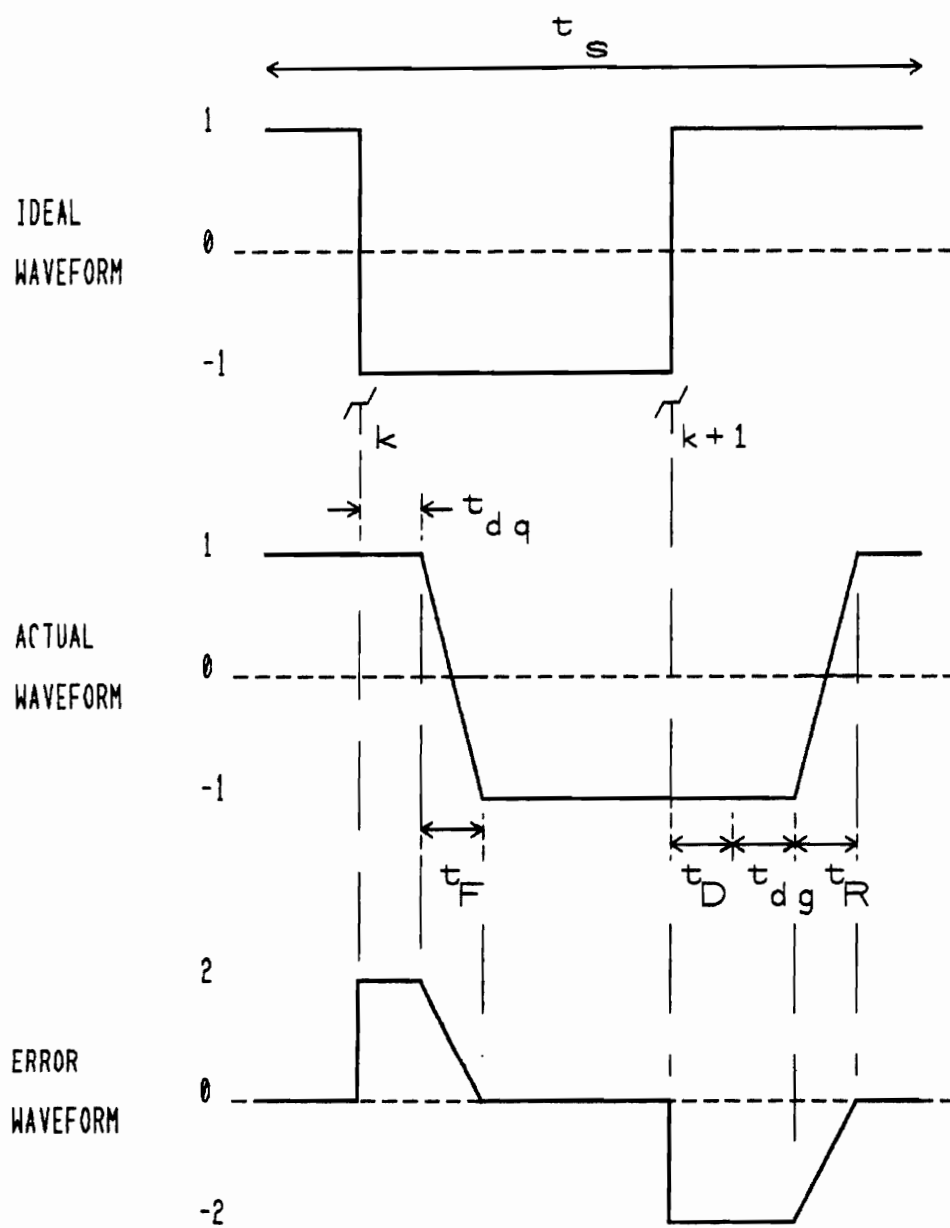


FIGURE 1.9: POSITIVE-SENSE CURRENT DISTORTION

distortion which may be modeled as an error waveform added to the ideal waveform. Spectral analysis of the error waveform yields output waveform spectral distortion information.

- ii.) Dead time, turn-on, and turn-off delays associated with switch S_1 create rectangular error pulses which have a polarity opposite of the ideal waveform level magnitude coinciding with the error pulse.
- iii.) Output waveform with fall time errors have a positive polarity. Output waveform with rise time errors have a negative polarity.
- iv.) Error waveforms are synchronized to the ideal waveform transition times, τ_k and τ_{k+1} .

1.3.2 Inverter Leg Switching with Negative-Sense Load Current

The switching operation of the inverter leg with load current flow into the leg output node (negative-sense load current) is described as follows:

- i.) Inverter leg load current flows into the leg output node and switch S_1 is enabled. Due to the load current sense, D_1 carries the load current, and the output waveform polarity is positive. The waveform remains in this state after the PWM control signal for S_1 goes logic 0 (switch disable). After a dead time delay, t_{dg} , the PWM control signal for S_2 goes from logic 0 to logic 1 (switch enabled). After a turn-on time delay, t_{dg} , S_2 begins to conduct.

S_2 current is the sum of the load current, the snubber capacitor charging current, and D_1 reverse recovery current. The rate-of-rise of S_2 current is limited by the snubber inductance L_s , and in-turn, limits the rate of load current commutation from the positive to negative rail. In this interval of time, the leg output waveform makes a level transition from the positive to the negative. After a fall time period, t_F , the leg output waveform level reaches its steady-state negative value (load current commutation via S_2 to the negative-rail is completed).

- ii.) Inverter leg load current flows into the leg output node and the PWM control signal to S_2 goes to logic 0 (switch disable). After a turn-off delay, t_{dq} , the output waveform rises to a positive level in an interval t_F . During this interval, load current flow is diverted from discharging the snubber capacitor to the positive-rail via D_1 . The current commutation rate is determined by the snubber inductance, L_s . Given the load current sense, the S_1 turn-on process does not affect the waveform level.

Figure 1.10 depicts the ideal and actual leg output waveforms given negative-sense load current. The observations made for the error waveform in the positive-sense load current case hold for the negative load current.

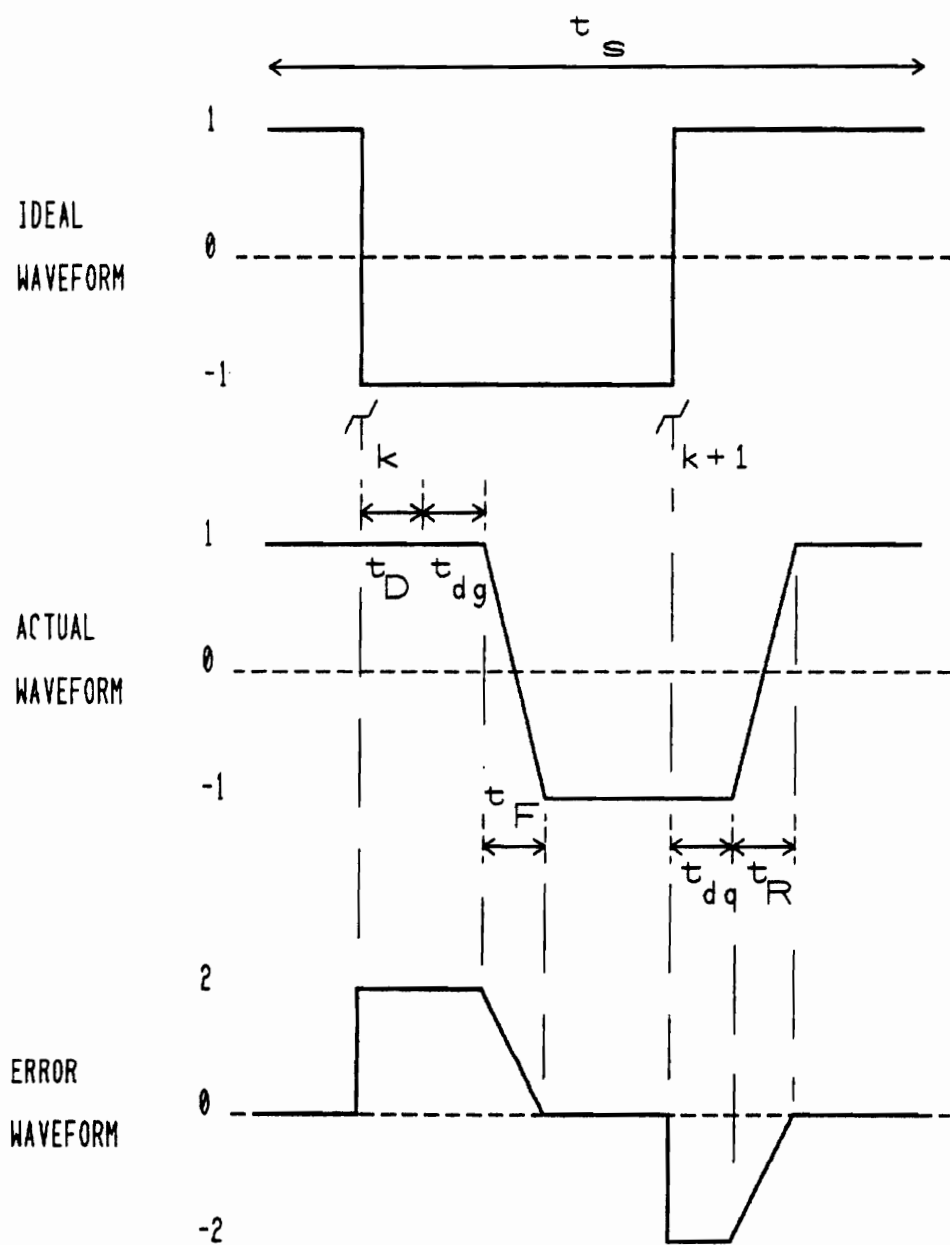


FIGURE 1.10: NEGATIVE-SENSE CURRENT DISTORTION

1.3.3 Fundamental Voltage Modification By Inverter Switching Distortion

The mean normalized error waveform voltage per switching period given positive-sense load current is expressed as

$$v_{erp} = 2 \cdot [t_{dq} - t_{dg} - t_D] / t_s, \quad (1.32)$$

where t_s is the inverter switching period, and $t_F \approx t_R$ (an assumption made based on the fact that the rate of load current commutation depends on the snubber inductance value L_S). The mean normalized error waveform voltage per switching period given negative load current is expressed as

$$v_{ern} = 2 \cdot [t_D + t_{dg} - t_{dq}] / t_s = -v_{erp}. \quad (1.33)$$

Given that $t_D > t_{dg}$ and $t_D > t_{dq}$, the previous equations reveal that the mean error voltage per switching period opposes the load current flow.

The effect of switching distortion on the fundamental component is analyzed by regarding the modification of the average inverter output waveform, $v_o(t)$, by the average error waveform, $v_{er}(t)$. Both waveforms are formed by using the average waveform value per switching period. As $f_s \rightarrow \infty$, $v_o(t)$ approaches the shape of the modulation waveform used by the PWM process to generate the ideal inverter output waveform. The normalized error waveform is defined by the equation

$$v_{er}(t) = \{2 \cdot [t_{dq} - t_{dg} - t_D] / t_s\} \cdot \tanh[G \cdot i_i(t)] \quad (1.34)$$

where $G \rightarrow \infty$, $i_i(t)$ is the instantaneous inverter leg load current, and $\tanh[G \cdot i_i(t)]$ takes on a value of 1 or -1 depending on the polarity of $i_i(t)$. The polarity of $i_i(t)$ is positive when current flow is from the leg circuit to the load circuit. If the load current waveform is a

sinusoid or any periodic odd function with only two zero crossings per waveform period, the mean error waveform described by equation (1.34) is a square waveform 180° degrees out of phase with the load current waveform. The phase-displacement of this square error waveform relative to the inverter modulation waveform is a function of the inverter output current power factor. The error waveform subtracts the fundamental and harmonic components of the phase-shifted square waveform (odd harmonic orders beginning with the third) from the modulation waveform. In addition to modifying the magnitude of the fundamental component, the square waveform harmonic frequencies will appear in the inverter leg output voltage spectrum (3rd, 5th, 7th, ...). For even integral values of f_s/f_m , low order even harmonics are generated in the inverter output waveforms as well. Assuming that the switching distortion is symmetrical with regard to all three phases, triplen harmonics will be eliminated from the line-to-neutral and line-to-line waveforms.

General observations regarding inverter switching distortion are given as follows:

- i.) The fundamental component of the inverter output waveform is reduced in magnitude by the delays associated with inverter switching. The dead-time interval (t_D) is the most significant contributor to this effect. In the case of sinusoidal load current, it is evident that the most severe reduction in fundamental frequency component amplitude occurs when the load has an unity power factor (the error square waveform is 180° out of phase with the modulation waveform).

- ii.) Evident from the absolute mean normalized error voltage per switching period, $[t_D + t_{dg} - t_{dq}]/t_s$, the distortion added to the modulation waveform is proportional to the inverter switching frequency. One factor determining the upper switching frequency limit is the degree of low-order harmonic distortion in the inverter output waveform resulting from the distortion added to the modulation waveform.
- iii.) Rise and fall time waveform distortion contributes to the high frequency components of the inverter output voltage spectrum. The process of creating finite slope transitions in the output waveform reduces the magnitude of higher order frequencies (which is beneficial). Typically, the harmonic frequencies at which waveform rise and fall times have an effect are not significant compared to the fundamental and low-order harmonic distortion created by switching delays.

Though the delay created by dead-time is the most significant factor in switching distortion, delays created by the control system and the interfacing hardware contribute to the switching distortion process as well. One significant contribution may come from the devices used to isolate the PWM gate generation logic from the control system. Though delays of this type ideally phase-shift the output waveforms without adding distortion, propagation delay variations in such devices can create additional distortion emphasized by increasing inverter switching frequencies.

1.4 Summary of PWM Waveform Generation Techniques

The magnitude of the inverter output voltage total harmonic distortion (THD) is defined by

$$\sqrt{\sum_{k > 1}^{\infty} v_{ik}^2 / k},$$

v_{ik} is the k^{th} harmonic voltage, is a common figure of merit for PWM waveform generation techniques in either variable-frequency (motor drives) or fixed-frequency (static UPS) applications. In UPS applications, THD determines the output filter line inductance or L_f . Additionally, the lowest-order harmonic determines the placement of the output filter pole. Due to the adverse effects of unbalanced and nonlinear UPS loads on the load voltage waveforms in a three-phase system, reduction of L_f is desirable.

Filter impedance magnitude reduction is achieved by either increasing the inverter switching frequency or utilizing a PWM technique which spectrally-shapes the inverter output waveform in a manner which reduces the filter impedance. Though increasing the switching frequency improves the control response of the inverter by increasing PWM process sampling rate and reduces the inherent delay associated with the synthesis of the load voltage waveform from the PWM inverter output waveform, inverter switching distortion effects (fundamental voltage reduction and low-order harmonic generation) are magnified by increasing the inverter switching frequency. Given the constraints of reducing the output filter impedance while maintaining load voltage distortion limits

under balanced passive load conditions, the presence of low-order inverter output waveform harmonics (for example, the 5th and 7th) due to inverter switching distortion may force an increase in the output filter series impedance due to the reduction of the output filter pole frequency in order to meet load voltage waveform distortion specifications.

Other motivations to reduce the inverter switching frequency are the reduction of inverter switching losses (particularly given the fact that the environment in which an UPS and the reduction of generated electromagnetic interference (EMI). The effects EMI has on sensitive systems (communications and medical imaging systems) is dependent on the characteristics of the sensitive system, the UPS DC power supply and inverter harmonic frequencies, and the packaging and design of the UPS. The flow of harmonic current generates an AC magnetic field around the conductor which in turn can induce AC currents in conductors in other UPS or critical load circuits. Containment of these AC fields by shielding the conductors (magnetic shielding) is a method for minimizing adverse EMI effects on sensitive critical loads. The success of this method is dependent on the physical size of the UPS system. In high-power three-phase UPS systems, the battery is physically much larger than the remainder of the UPS system, and so, it is typically outside of the enclosure(s) containing the remainder of the UPS circuitry. Though not impossible, magnetic shielding in the high-power UPS battery circuits can prove to be expensive.

Compared to the benchmark conventional sinusoid carrier PWM technique, the use of either suboptimal carrier or space vector PWM results in a lower computed L_f value. Further significant reduction in L_f is achieved through the use of an optimized fixed-pattern inverter switching strategy designed to eliminate low-order harmonics. Additionally, the filter pole may be placed at a higher frequency due to the shift of inverter output waveform harmonics upwards in frequency.

The cost of implementing the optimized fixed-pattern inverter switching strategy is the addition of a DC/DC converter between the battery and the inverter DC input. The inverter output voltage waveforms generated through the use of programmed inverter switching patterns based on selective harmonic elimination are subject to the adverse effects of switching delays and dead-time insertion resulting in the generation of low-order harmonics in the inverter output waveform not anticipated in the process of generating inverter switching patterns off-line.

1.5 Survey of Closed-Loop Inverter Control Techniques

With the use of low-distortion PWM techniques either in the form of programmed waveforms generated by methods described in [8] and [9] or a modified carrier PWM technique described in [6], significant reductions in the output filter series impedance magnitude are realized. As a result, unequal load voltage magnitudes and phaseshifts due to

unbalanced loads and waveform harmonic distortion due to nonlinear loads are minimized. Closed-loop load voltage control based on controlling the load voltage amplitude only (average control) is then justified.

Alternative methods of closed-loop load voltage control are based on subcycle control of inverter switching with the objective of minimizing load voltage waveform amplitude and phase-displacement errors in relation to a sinusoidal load voltage reference waveform. The discussion of closed-loop three-phase UPS inverter output waveform control begins with the consideration of the inverter load model, the output filter circuit depicted in figure 1.2, described by the two-phase stationary d-q frame state equation

$$d\mathbf{x}(t)/dt = \mathbf{A} \cdot \mathbf{x}(t) + \mathbf{B} \cdot \mathbf{v}_i(t) + \mathbf{C} \cdot \mathbf{w}(t) \quad (1.35)$$

where

$$\mathbf{x} = \begin{bmatrix} v_{dl} \\ v_{ql} \\ (d/dt) \cdot v_{dl} \\ (d/dt) \cdot v_{ql} \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -(2\pi \cdot f_f)^2 & 0 & 0 & 0 \\ 0 & -(2\pi \cdot f_f)^2 & 0 & 0 \end{bmatrix},$$

$$\mathbf{v}_i = \begin{bmatrix} v_{di} \\ v_{qi} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ (2\pi \cdot f_f)^2 & 0 \\ 0 & (2\pi \cdot f_f)^2 \end{bmatrix},$$

$$\mathbf{w} = \begin{bmatrix} (d/dt) \cdot i_{dl}^{(1)} \\ (d/dt) \cdot i_{ql}^{(1)} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -L_f \cdot (2\pi \cdot f_f)^2 & 0 \\ 0 & -L_f \cdot (2\pi \cdot f_f)^2 \end{bmatrix}.$$

Defining $\{i_{ai}, i_{bi}, i_{ci}\}$ as the inverter output phase currents, and $\{i_{al}, i_{bl}, i_{cl}\}$ as the UPS load phase currents, the d-q frame current variables $\{i_{di}^{(1)}, i_{qi}^{(1)}\}$ and $\{i_{dl}^{(1)}, i_{ql}^{(1)}\}$ are transformations of the three-phase difference expressions

$$\{i_{ai} - i_{bi}, i_{bi} - i_{ci}, i_{ci} - i_{ai}\} \text{ and}$$

$$\{i_{al} - i_{bl}, i_{bl} - i_{cl}, i_{cl} - i_{al}\}$$

respectively. The inverter output voltage vector $\underline{v}_i$ is ideally described as one of eight inverter switching states $\underline{v}_s^{(k)}$ depicted in figure 1.6 in a corresponding switching state time interval t_{sk} where $0 \leq k \leq 8$.

Subcycle control of inverter switching is implemented through the selection of inverter switching states and corresponding inverter switching state times within intervals which are synchronized to inverter switching process. In the case of closed-loop UPS load voltage control, the selection of inverter switching states and switching state times in each inverter switching period is based on minimizing the sampled instantaneous load voltage errors where the maximum sampling rate is at the inverter switching frequency. Given the initial state $\underline{x}(t_0)$, the transition of filter circuit states $\underline{x}(t)$ in an interval $\{t_0, t_1\}$ is described by the equation

$$\begin{aligned} \underline{x}(t_1) = & \Phi(t_1 - t_0) \cdot \underline{x}(t_0) + \int_{t_0}^{t_1} \Phi(t_1 - \tau) \cdot C \cdot \underline{w}(\tau) d\tau + \\ & \int_{t_0}^{t_1} \Phi(t_1 - \tau) \cdot B \cdot \underline{v}_i(\tau) d\tau, \end{aligned} \quad (1.36)$$

where $\Phi(t)$ is the output filter circuit state transition matrix.

With the use of linear control techniques, the length of the interval $\{t_0, t_1\}$ is uniform through-out the inverter output waveform period. Given a PWM waveform generation process utilizing asymmetric regular sampling (corresponding to the maximum sampling rate), the length of this interval is $1/2f_s$. Defining $T = 1/2f_s$, equation (1.36) is re-written in a discrete-time form

$$\begin{aligned} \underline{x}(k+1) = & \Phi(T) \cdot \underline{x}(k) + \int_0^T \Phi(T-\tau) \cdot C \, d\tau \cdot \underline{w}(k) + \\ & \int_0^T \Phi(T-\tau) \cdot B \, d\tau \cdot \underline{v}_i(k), \end{aligned} \quad (1.37)$$

where $t_0 = k \cdot T$, $t_1 = (k+1) \cdot T$, $\underline{x}(k+1) = \underline{x}[(k+1) \cdot T]$, $\underline{x}(k) = \underline{x}(k \cdot T)$, and $\underline{w}(k)$ is the sampled system disturbance at $t = t_0$. $\underline{v}_i(k)$ is the sampled inverter output voltage at $t = t_0$ where this voltage is regarded as a continuous function of time. Given uniform sampling, $\Phi(T)$ and the integral expressions in (1.37) are constants.

Given a target filter state $\underline{x}^*(k+1)$ (which includes the UPS load voltages), the inverter output voltages $\underline{v}_i^*(k)$ required to synthesize the target filter states are derived from the equation

$$\underline{v}_i^*(k) = (B')^{-1} \cdot [\underline{x}^*(k+1) - \Phi(T) \cdot \underline{x}(k) - C' \cdot \underline{w}(k)] \quad (1.38)$$

where

$$C' = \int_0^T \Phi(T-\tau) \cdot C \, d\tau \quad (1.39a)$$

and

$$B' = \int_0^T \Phi(T - \tau) \cdot B \, d\tau. \quad (1.39b)$$

Equation (1.38) describes, in general, the dead-beat control algorithm utilized in UPS inverter control techniques described in [2], [4], and [5]. Based on sampled filter state feedback $\{\underline{x}(k), \underline{w}(k)\}$, knowledge of the output filter circuit design, controller sampling rate, and a target filter state command $\underline{x}^*(k + 1)$, inverter switching in uniform intervals of length T is controlled with the objective of generating the sampled inverter output voltage $\underline{v}_i^*(k)$. From the perspective of the controller, (1.38) is normalized with respect to the inverter DC power supply voltage or

$$\underline{v}_{in}^*(k) = (B')^{-1} \cdot [\underline{x}_n^*(k + 1) - \Phi(T) \cdot \underline{x}_n(k) - C' \cdot \underline{w}_n(k)] \quad (1.40)$$

where $\underline{v}_{in}^*(k) = \underline{v}_i^*(k)/v_{dc}$, $\underline{x}_n^*(k + 1) = \underline{x}^*(k + 1)/v_{dc}$, $\underline{x}_n(k) = \underline{x}(k)/v_{dc}$, $\underline{w}_n(k) = \underline{w}(k)/v_{dc}$, and v_{dc} is the inverter DC power supply voltage.

The implementation of equation (1.40) in a controller design requires practical considerations regarding the conversion of $\underline{v}_{in}^*(k)$ to inverter switching commands. In the single-phase dead-beat control methods described in [2] and [3], The equivalent of $\underline{v}_{in}^*(k)$ in a single-phase system is expressed in terms of the inverter output voltage pulse waveform polarity and corresponding pulsewidth. A single-phase version of equation (1.40) is used to derive the waveform pulsewidth and the waveform polarity is the polarity of the left-hand side of (1.40). Single-phase inverter switching is the direct result of the controller's output. In the three-phase dead-beat control methods described in [4]

and [5], $\underline{v}_{in}^*(k)$ is a continuous-time d-q frame inverter output voltage command converted to three single-phase inverter output voltage commands for use by carrier PWM control logic.

In order to generate the target filter states $\underline{x}_n^*(k + 1)$ with minimum instantaneous errors, it is necessary to generate the inverter output voltage command $\underline{v}_{in}^*(k)$ in advance of its execution interval $\{k \cdot T, (k + 1) \cdot T\}$. In a microprocessor or DSP-based controller, the dead-beat control algorithm is modified to compensate for the controller computational delays and the inverter switching command execution delay. The discrete state transition time is increased to $2 \cdot T$ requiring the use of constant matrix values

$$C'' = \int_0^{2 \cdot T} \Phi(2 \cdot T - \tau) \cdot C \, d\tau \quad (1.41a)$$

and

$$B'' = \int_0^{2 \cdot T} \Phi(2 \cdot T - \tau) \cdot B \, d\tau. \quad (1.41b)$$

In the interval $\{k - 1, k\}$, the inverter output voltage commands are computed using the equation

$$\underline{v}_{in}^*(k) = (B'')^{-1} \cdot [\underline{x}_n^*(k + 1) - \Phi(2 \cdot T) \cdot \underline{x}_n(k - 1) - C'' \cdot \underline{w}_n^{(p)}(k)] \quad (1.42a)$$

where $\underline{w}_n^{(p)}(k)$ is the predicted sampled (at $k \cdot T$) load current values. Likewise in the interval $\{k, k + 1\}$, the inverter output voltage commands are computed using the equation

$$\underline{v}_{in}^*(k + 1) = (B'')^{-1} \cdot [\underline{x}_n^*(k + 2) - \Phi(2 \cdot T) \cdot \underline{x}_n(k) - C'' \cdot \underline{w}_n^{(p)}(k + 1)]. \quad (1.42b)$$

In [4] and [5], a method of load current prediction is described using previous load current samples.

The performance of dead-beat control methods is dependent on system parameter variations (tolerances associated with L_f and C_f), the effects of inverter switching distortion, inverter DC power supply voltage variations, and inaccuracies associated with controller operation. In digital controllers where fixed-point computations are used, the bit-length of the control variables can contribute to errors in controller operation. Additionally, sampled filter circuit feedback may contain errors either as a result of analog-to-digital conversion quantization errors or harmonic distortion in the feedback signals. In order to eliminate the steady-state load voltage errors, an augmentation to the dead-beat controller in the form of either a proportional-integral (PI) controller or a repetitive controller described in [3] is necessary.

In the three-phase controllers constructed in the synchronous d-q frame described in [4] and [5], fundamental frequency steady-state errors are DC values. Steady-state errors due to output filter components tolerance, reduction of the inverter output fundamental voltage due to switching distortion, or a steady-state change in the inverter DC power supply voltage are successfully eliminated by the use of a PI controller in the inverter controllers described in [4] and [5]. On the other hand, PI controllers can not eliminate errors due to

independent periodic disturbances such as harmonic UPS load currents. In the three-phase controller techniques described in [4] and [5], compensation for load voltage waveform harmonic distortion is a function of the dead-beat controller.

The repetitive controller described in [3] is used to eliminate steady-state load voltage errors due to periodic load disturbances. This controller application is described in a single-phase inverter application. The repetitive controller function is described by the expression

$$u_2(k \cdot T) = c_1 \cdot \epsilon_v[(k + 1 - n) \cdot T] + c_2 \cdot \sum_{i=1}^{\infty} \epsilon_v[(k + 1 - i \cdot n) \cdot T] \quad (1.43)$$

where T is the controller's sampling interval, $u_2(k \cdot T)$ is the controller's output, $\epsilon_v(k \cdot T)$ is the sampled load voltage error (relative to a sinusoidal waveform reference), $\{c_1, c_2\}$ are controller gains, and n is the ratio between the inverter modulation and controller sampling frequencies. n is an integer equal to the ratio between the controller sampling frequency and the inverter fundamental frequency.

The output of the repetitive controller is added to the output of a dead-beat controller where $u_1(k \cdot T)$ is defined as the dead-beat controller output. At $t = k \cdot T$, the sum $u_1(k \cdot T) + u_2(k \cdot T)$ is the inverter output voltage command to the PWM control logic. Assuming that load voltage errors are created by periodic disturbances (frequencies greater than the fundamental but less than one-half the controller sampling frequency), the addition of the weighted sum of previous load voltage errors sampled at intervals $(k + 1 - i \cdot n) \cdot T$, $i \geq 1$, to the dead-beat

controller output $u_1(k \cdot T)$ serves to reduce the sampled load voltage at $t = k \cdot T$.

The repetitive controller concept is also based on the assumption that the sampled AC inverter modulation command signal $[u_1(k \cdot T) + u_2(k \cdot T)]$ is linearly transformed into a corresponding inverter output voltage waveform. It should be noted that the PWM sampling process is not linear with regard to modulation commands with more than one frequency component, subsequently the assumption that the modulation command signal $[u_1(k \cdot T) + u_2(k \cdot T)]$ is linearly transformed into a corresponding inverter output voltage waveform is not valid. From the discussion of PWM sampling techniques in section 1.2.2, the description of the PWM sampling process nonlinearity is repeated. Defining $v_i^*(f)$ as the Fourier Transform of the modulation command signal,

$$v_i^*(f) \equiv \sum_{k=-\infty}^{\infty} a_k \cdot \delta(f - k \cdot f_m), \text{ where } f_m \text{ is the modulation frequency,}$$

k is an integer, $\delta()$ is the Dirac delta function, f is the frequency variable, a_k is the k^{th} Fourier coefficient of $v_i^*(f)$, and T as the sampling interval, the Fourier Transform of the sampled modulation command signal is given by the expression,

$$\int_{-\infty}^{\infty} v_i^*(f - \Omega) \cdot \sum_{n=-\infty}^{\infty} \delta(\Omega - n/T) d\Omega \cdot \sin(\pi \cdot T \cdot f) / (\pi \cdot T \cdot f) \quad (1.44)$$

where Ω is an integration variable. The modulation command Fourier Transform is modified by the sampling process in two ways:

- i.) The modulation command spectral components are modified by the $\sin(\pi \cdot T \cdot f) / (\pi \cdot T \cdot f)$ function.

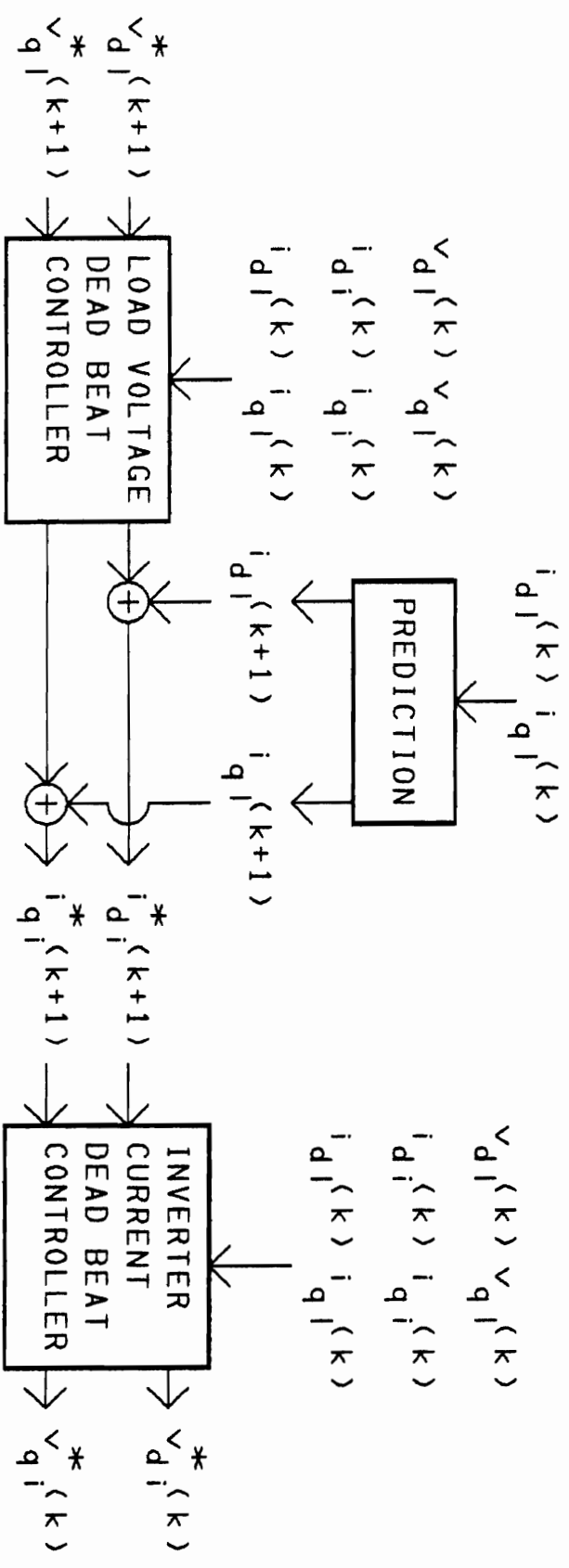


FIGURE 1.11: THREE-PHASE LINEAR CONTROLLER

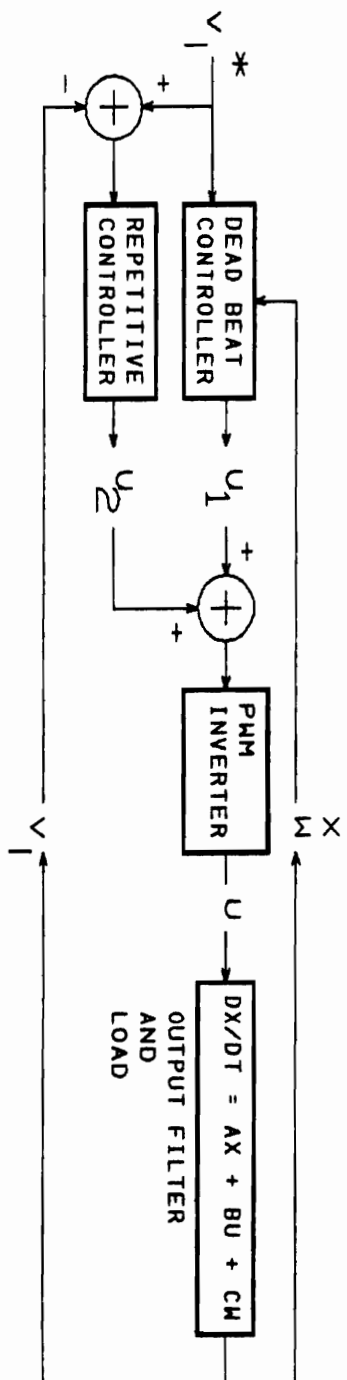


FIGURE 1.12: REPETITIVE CONTROLLER APPLICATION

ii.) Frequency components of the orders $k \cdot f_m \pm n/T$ ($n > 0$) are added to the modulation command Fourier Transform. An aliasing (or waveform distortion) effect occurs if the modulation command signals contain spectral components at frequencies greater than twice the sampling frequency.

1.5.1 Nonlinear Controllers

Nonlinear control techniques such as adaptive hysteresis described in [1] and Variable Structure Strategies (VSS) are alternative methods of UPS inverter control. The advantages of nonlinear controllers are insensitivity to system parameter variations, external disturbances, and modeling errors. These advantages are realized at the cost of relatively high controller sampling rates (high inverter switching frequencies).

The principal objective of the VSS control technique is to force the trajectory of the system to a surface, known as the sliding surface, so that regulation takes place on this sliding surface. The control laws are designed so that the system trajectory always reaches the sliding surface. Once on the sliding surface, the control structure is changed discontinuously to maintain the system on the sliding surface. At this point, the system is in the sliding mode. The structure is changed according to a preselected switching logic, where the change is based on the instantaneous values of system states evaluated along the trajectory. Typical switching hypersurfaces in the d-q frame include

$$s_1 \equiv i_{di}(t) - i_{di}^*(t) \text{ and} \quad (1.45)$$

$$s_2 \equiv i_{qi}(t) - i_{qi}^*(t), \text{ or} \quad (1.46)$$

$$s_3 \equiv v_{dl}(t) - v_{dl}^*(t) \text{ and} \quad (1.47)$$

$$s_4 \equiv v_{ql}(t) - v_{ql}^*(t). \quad (1.48)$$

The sufficient condition for the sliding mode to exist on the i^{th} hypersurface $s_i(t)$ is given by the expression

$$\lim_{s_i(t) \rightarrow 0} \{s_i(t) \cdot ds_i(t)/dt\} < 0. \quad (1.49)$$

Gain values associated with a specific VSS controller form, which change values discontinuously in order to achieve a sliding mode, are derived from equation (1.54) in the form of inequalities. Examples of the application of the VSS technique with regard to the control of brushless DC motors are given in [25] and [26].

To illustrate the VSS technique, the determination of controller gains $\{K_d, K_q\}$ for an inverter output current controller defined by the equations

$$v_{di}^*(t) = K_d \cdot \text{sgn}[s_1(t)], \quad (1.50)$$

$$v_{qi}^*(t) = K_q \cdot \text{sgn}[s_2(t)], \quad (1.51)$$

where

$$\begin{aligned} \text{sgn}(s_i) &= 1 \quad \text{for } s_i > 0 \\ &= -1 \quad \text{for } s_i < 0 \\ &= 0 \quad \text{for } s_i = 0, \end{aligned} \quad (1.52)$$

is undertaken. The following inequalities are derived:

$$\begin{aligned} s_1 \cdot ds_1/dt &= s_1 \cdot \{w \cdot i_{qi}^{(1)} + (1/L_f) \cdot [K_d \cdot \text{sgn}(s_1) - v_{dl}] - \\ &\quad di_{di}^{(1)}/dt\} < 0 \end{aligned} \quad (1.53)$$

$$s_2 \cdot ds_2/dt = s_2 \cdot \{-w \cdot i_{di}^{(1)} + (1/L_f) \cdot [K_q \cdot \text{sgn}(s_2) - v_{ql}] -$$

$$di_{di}^{(1)*}/dt \} < 0. \quad (1.54)$$

Current controller gains satisfy the inequalities

$$K_d < -(s_1/|s_1|) \cdot [w \cdot L_f \cdot i_{qi}^{(1)} - v_{d1} - L_f \cdot di_{di}^{(1)*}/dt], \text{ and} \quad (1.55)$$

$$K_q < (s_2/|s_2|) \cdot [w \cdot L_f \cdot i_{di}^{(1)} - v_{q1} - L_f \cdot i_{qi}^{(1)*}/dt], \quad (1.56)$$

where s_1 and s_2 are nonzero (gain values are irrelevant given zero errors). Current controller commands $\{i_{di}^{(1)*}, i_{qi}^{(1)*}\}$ are typically derived from a load voltage controller (outer control loop). The current controller outputs $\{v_{di}^*, v_{qi}^*\}$ are modulation commands to a space vector PWM process.

With the use of a finite sampling interval, the VSS does not slide along the surface $s_i(t) = 0$, rather the resulting trajectory chatters back and forth across this surface. Inverter DC supply and switching distortion, controller quantization effects, sensor delays, and nonlinear load-generated distortion contribute to this chattering effect. The upper limits on the controller gains are established by constraints on the trajectory variation magnitude and the need to maintain controller operation in the sliding mode.

1.6 Adaptive Intelligent Inverter Control Strategy

With increasing UPS inverter output power levels, practical limitations in the form of switching delays, switching losses, and the generation of electromagnetic interference force corresponding reductions of the inverter switching frequency (and the load voltage controller sampling rate). The effects of the inverter switching

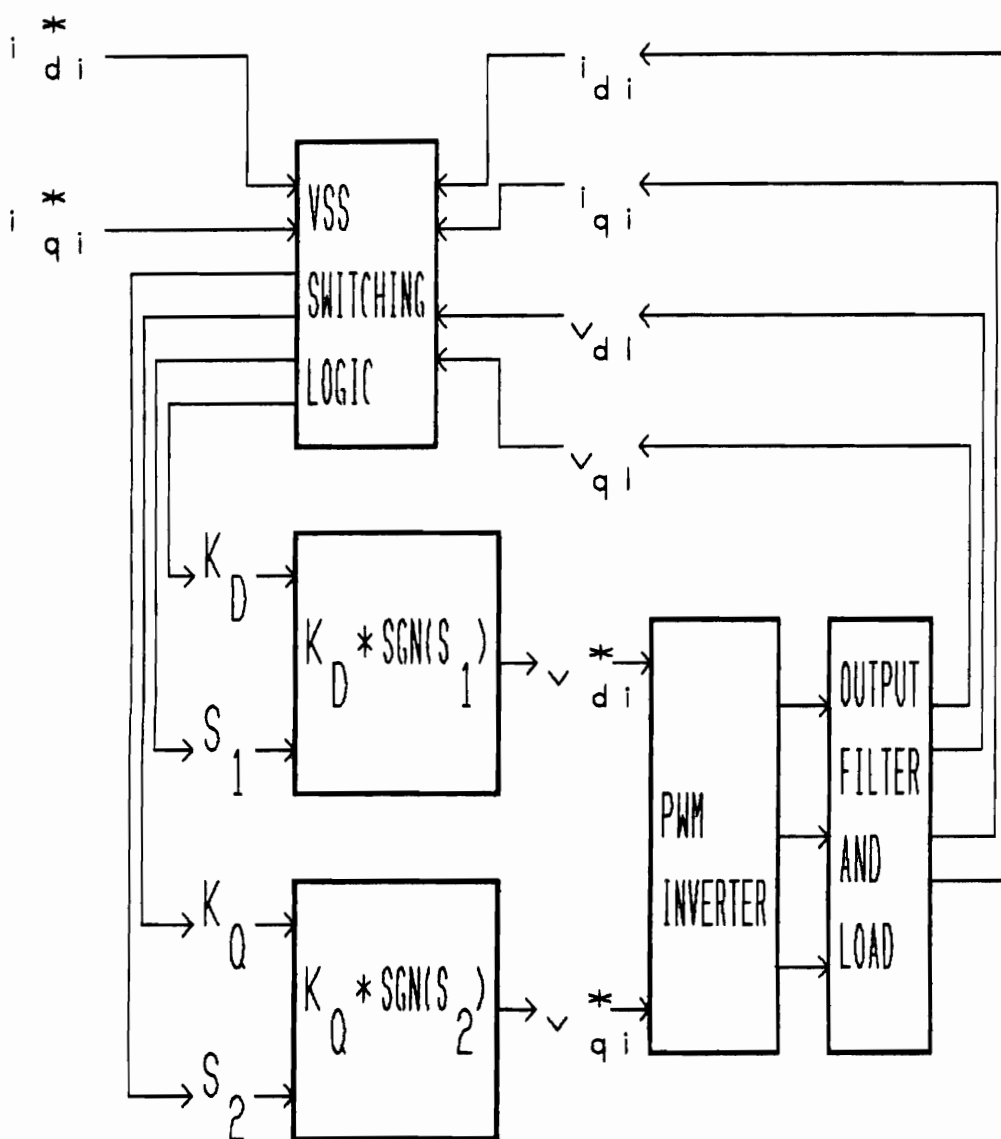


FIGURE 1.13: VSS INVERTER OUTPUT CURRENT CONTROLLER

frequency magnitude in the inverter waveform generation and control processes are summarized as follows:

- i.) the spectral characteristics of the ideal inverter output waveform are direct functions of the inverter switching frequency and the inverter PWM switching method. The output filter design (selection of L_f and C_f) which is typically based on load voltage waveform harmonic distortion limits, given balanced passive UPS loads, is a function of the inverter switching frequency. The filter series impedance magnitude $2\pi \cdot f \cdot L_f / (1 - 12\pi^2 \cdot f^2 \cdot L_f \cdot C_f)$ determines the degree unbalanced and nonlinear loads distort the UPS load voltage waveforms. The location of the output filter pole frequency determines the bandwidth of the load voltage controller (and its response transient or periodic load variations).
- ii.) As described in [8] and [9], direct control inverter switching with the objectives of controlling the output waveform fundamental and harmonic components is achieved through the solution of a set of nonlinear equations where the N line-to-line inverter output waveform switching angles $\{\beta_1, \beta_2, \dots, \beta_N\}$ within the output waveform fundamental frequency period for each phase are the dependent variables. The value of N is a function of the inverter switching frequency. Switching angle solutions for each phase (which may differ due to the connection of an unbalanced UPS load or an unbalanced filter circuit due to L_f and C_f tolerances) $\{\beta_1^{(0)}, \beta_2^{(0)}, \dots, \beta_N^{(0)}\}$ are typically best-fit solutions where an

exact solution is approached by increasing N (or increasing the inverter switching frequency).

- iii.) The PWM technique modulation command sampling process is nonlinear with regard to modulation commands and harmonic frequency components. Increasing the inverter switching frequency (or decreasing the sampling interval) diminishes this nonlinearity at lower-order harmonics of the inverter modulation frequency.
- iv.) The load voltage controller's sampling interval is decreased with increased inverter switching frequency. As a result, the output filter circuit state variable transition described by its discrete-time state transition equation (the basis for the load voltage controller design) given the presence of harmonic load currents and inverter DC power supply voltage harmonic distortion approaches the state transition described by the continuous-time state transition equation.

The best method of PWM inverter waveform generation and control depends on the UPS application. In low-power single-phase systems where relatively high inverter switching frequencies are practical (in the order of tens of kiloHertz), nonlinear control methods are applicable. Due to their insensitivity to output filter component value variations, modeling errors, and external disturbances given switching frequencies on the order described previously, nonlinear control techniques would appear to be the best choice in this application. Other single-phase control techniques as described in [2] and [3] are also applicable in

low-power systems particularly in the case where lower inverter switching frequencies are desired. These control techniques require the use of microprocessor-based controllers which in low-power single-phase inverter applications is not desirable with regard to the associated cost and complexity. An analog circuit control technique described in [27] based on model reference adaptive control is a lower cost alternative to the use of microprocessor-based dead-beat control techniques. On the other hand, the load model assumption made with regard to the implementation of this technique (the load is approximated as a pure inductance) is not necessarily valid given nonlinear single-phase UPS loads.

Three-phase PWM inverter systems are typical in medium- and high-power applications. A problem unique to three-phase systems is the adverse effect of unbalanced UPS loads on the load voltage waveforms (unequal voltage magnitudes and phaseshifts). Other conditions unique to three-phase systems are:

- i.) Inverter output waveform control with regard to fundamental voltage control and selective control of waveform harmonics is limited by three-phase inverter bridge switching constraints. The control of one inverter output voltage phase affects the other two phase voltages.
- ii.) In medium- and high-power UPS systems, disturbance-free transfer of load power flow between the UPS output and the mains (via the UPS system bypass circuit) is required. Typically in low-power single-phase systems, UPS maintenance is performed easily by

turning-off both the single-phase UPS and its critical load. This is not the case in three-phase UPS systems where the critical load operates without interruption despite the occasional need to service the UPS.

The output filter impedance is a critical factor with regard to either of these two conditions. The utility mains circuit is a nearly zero impedance circuit compared to the UPS output. Given the possibility of unbalanced and nonlinear UPS loads, the transfer of load power flow between a nearly-zero impedance source and a source with a higher impedance may lead to an undesirable disturbance in critical load operation.

In medium- and high-power three-phase UPS inverter applications, three general approaches are used to minimize the problems associated with unbalanced and nonlinear UPS loads:

- i.) The use of a fixed-pattern PWM technique as described in [6] and [7] with the objective of reducing the output filter series impedance. Average closed-loop load voltage control is employed with the use of this approach.
- ii.) Three-phase inverter output waveform control using the dead-beat control techniques described in [4] and [5] with the objective of compensating for load voltage distortion due to unbalanced and nonlinear loads through the control of inverter switching. The output filter design is not necessarily optimized with regard to reducing the output filter series impedance.

iii.) The addition of tuned traps or active filter circuits to the UPS output filter circuit.

Since it is desirable to minimize the UPS power circuit complexity at higher power levels in order to reduce cost and improve system reliability, the application of closed-loop inverter output waveform subcycle control techniques would appear to be the best approach to resolving load voltage distortion problems due to unbalanced and nonlinear loads.

In addition to closed-loop inverter output waveform subcycle control, a further improvement to a high-power UPS inverter design would be the generation of a low-distortion inverter output voltage waveform under balanced passive loads as in the case of the fixed-pattern PWM technique described in [6] enabling the reduction of the output filter series impedance. Because asymmetrical inverter output waveform generation is required for load voltage waveform compensation due to unbalanced loads, and selective inverter output waveform harmonic control is required for load voltage waveform compensation due to nonlinear loads, the use of optimized fixed pattern inverter output waveforms (as in [7]) is not considered. Though the use of space vector PWM in open-loop does not result in the significant reduction of the output filter series impedance as in the case of the PWM technique described in [6] or the optimized switching patterns used in the inverter control technique described in [7], this PWM technique is far

more flexible with regard to its implementation in subcycle closed-loop control methods.

An alternative method of linear inverter control techniques is the use of an adaptive intelligent controller structure which is capable of learning subcycle load voltage control with the objective of minimizing the instantaneous load voltage waveform errors relative to a three-phase set of sinusoidal load voltage reference waveforms. Though the minimization of load voltage waveform errors is the objective of conventional linear control techniques, these techniques are based on precise knowledge of the output filter design, load current characteristics, the inverter switching frequency (controller sampling rate), and assumptions regarding ideal inverter switching and the operation of an inverter as a linear power amplifier.

Artificial neural networks (ANN's) are mathematical systems designed to emulate the rudimentary learning and self-organization functions of the biological brain. The structure of a feed-forward ANN ,for example the structures depicted in figures 1.14a and 1.14b, connects network inputs and outputs through multiple linear or nonlinear neuron models, and processes these input/output data associations in a parallel distributed manner. A neuron unit (or node) consists of multiple inputs and one output where the output is a function of the weighted sum of inputs. For example, the output of the hidden layer in the two-layer ANN structure is described by the expression

$$h_q(t) = f\left[\sum_{i=1}^I w_{iq}^{(1)} \cdot x_i(t) - \theta\right] \quad (1.57)$$

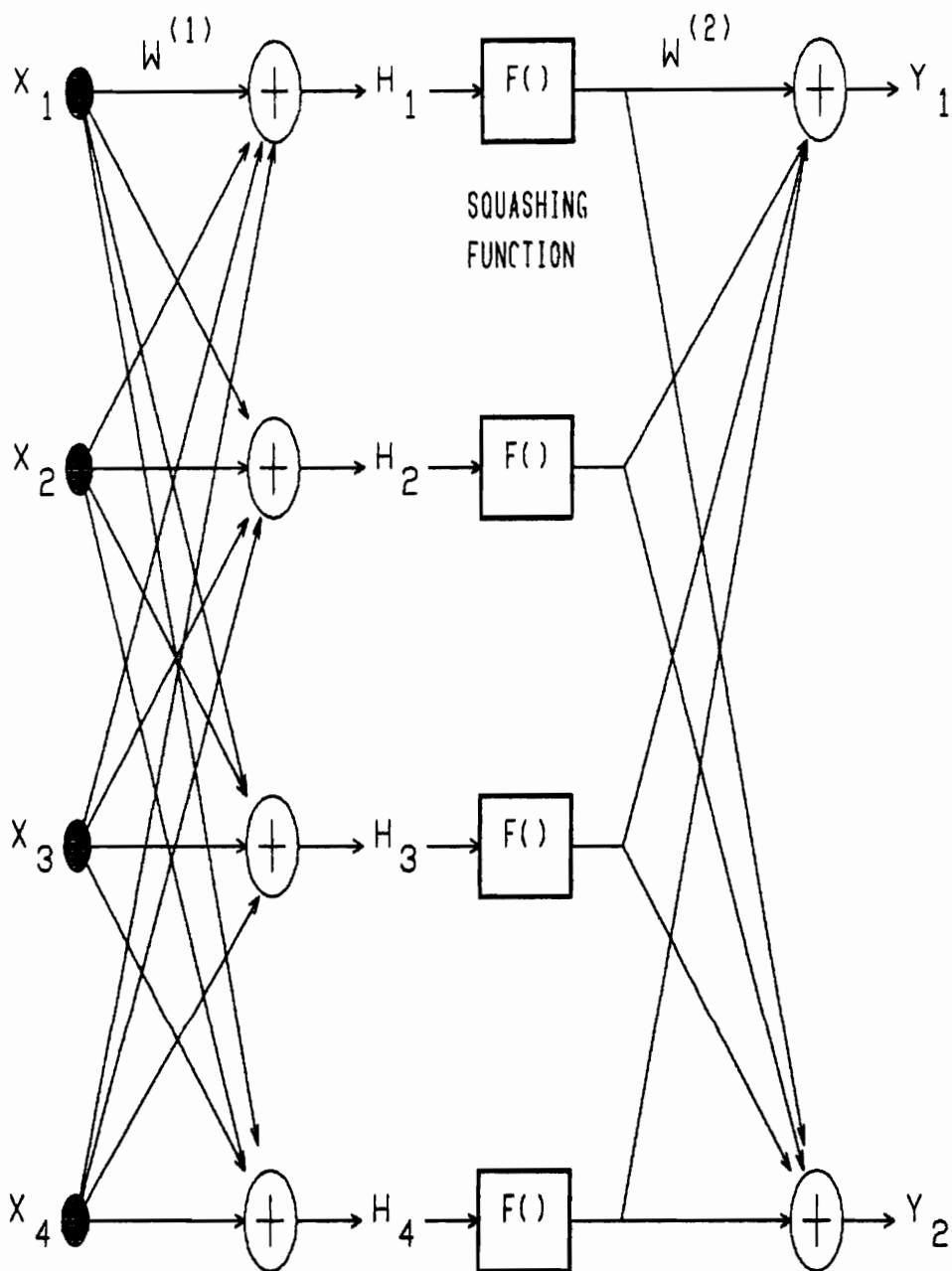


FIGURE 1.14A: TWO-LAYER ANN

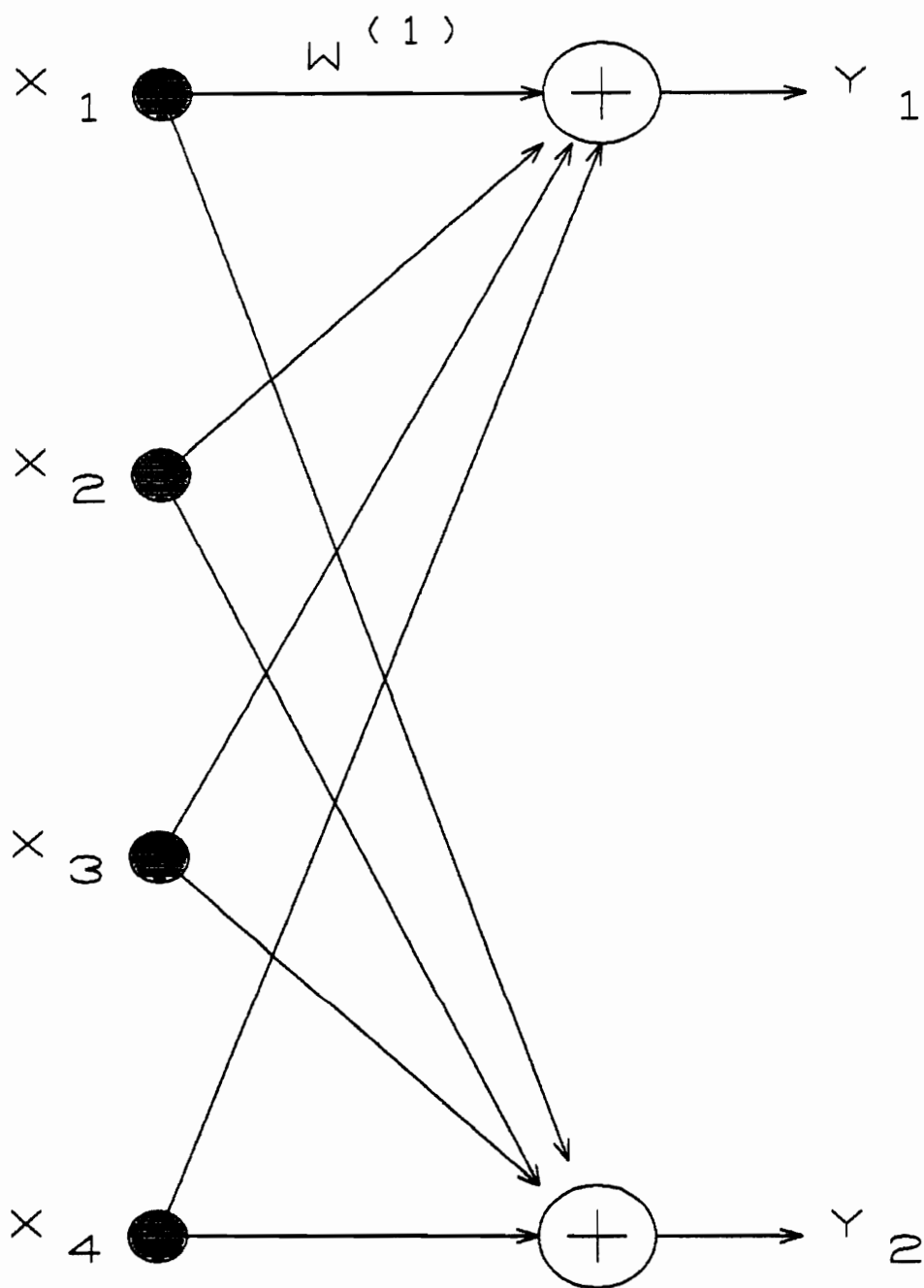


FIGURE 1.14B: ONE-LAYER ANN

where x_i is a network input, I is the number of network inputs, h_q is a hidden layer output, t is time, θ is a bias of the node, and $w_{iq}^{(1)}$ is the weight of the connection between the i^{th} input and the q^{th} hidden layer node output. When the function $f()$ is nonlinear, it is typically either a sigmoid function,

$$f(x) \equiv 1/[1 + e^{-Ax}] \quad 0 < f(x) < 1 \quad (1.58)$$

and A is a constant, or a hyperbolic tangent function,

$$f(x) \equiv \tanh[Ax] \quad -1 < f(x) < 1. \quad (1.59)$$

The function $f()$ often referred to as the squashing function serves to limit the node output and to impart additional nonlinear characteristics to the weighted sum.

In control applications, the attraction ANN's have are their ability to perform functional approximation of nonlinear systems and/or systems with uncertain structure in a fashion that is beyond the theoretical and practical capabilities of conventional optimal linear techniques. In the case of the PWM inverter/filter/load system, both linear and nonlinear characteristics system transfer functions exist. Additionally, output filter design which is based on several system variables (inverter switching frequency, PWM inverter waveform generation technique, and the UPS maximum power rating) may be regarded as a system with an uncertain structure if filter component tolerances are considered and if we desire to construct a load voltage controller that has the flexibility to operate in a wide range of UPS system designs.

Using the linear inverse dynamics of the output filter circuit, inverter output voltages based on the load voltage sinusoidal reference voltages $\{v_{abl}^*, v_{bcl}^*, v_{cal}^*\}$ and the measured load currents are derived.

In the interval $t_0 \leq t$, the load voltage state transition equations for the output filter circuit depicted in figure 1.2 are:

$$v_{abl}(t) = \cos[2\pi \cdot f_f \cdot (t - t_0)] \cdot v_{abl}(t_0) + \\ [1/2\pi \cdot f_f] \cdot \sin[2\pi \cdot f_f \cdot (t - t_0)] \cdot v_{abl}'(t_0) + \\ 2\pi \cdot f_f \cdot \int_{t_0}^t \sin[2\pi \cdot f_f \cdot (t - \tau)] \cdot [v_{abi}(\tau) - L_f \cdot i_{abl}'(\tau)] d\tau \quad (1.60a)$$

$$v_{bcl}(t) = \cos[2\pi \cdot f_f \cdot (t - t_0)] \cdot v_{bcl}(t_0) + \\ [1/2\pi \cdot f_f] \cdot \sin[2\pi \cdot f_f \cdot (t - t_0)] \cdot v_{bcl}'(t_0) + \\ 2\pi \cdot f_f \cdot \int_{t_0}^t \sin[2\pi \cdot f_f \cdot (t - \tau)] \cdot [v_{bci}(\tau) - L_f \cdot i_{bcl}'(\tau)] d\tau \quad (1.60b)$$

$$v_{cal}(t) = \cos[2\pi \cdot f_f \cdot (t - t_0)] \cdot v_{cal}(t_0) + \\ [1/2\pi \cdot f_f] \cdot \sin[2\pi \cdot f_f \cdot (t - t_0)] \cdot v_{cal}'(t_0) + \\ 2\pi \cdot f_f \cdot \int_{t_0}^t \sin[2\pi \cdot f_f \cdot (t - \tau)] \cdot [v_{cai}(\tau) - L_f \cdot i_{cal}'(\tau)] d\tau \quad (1.60c)$$

where $v_{abl}' = (d/dt) \cdot v_{abl}$, $v_{bcl}' = (d/dt) \cdot v_{bcl}$, $v_{cal}' = (d/dt) \cdot v_{cal}$,
 $i_{abl}' = (d/dt) \cdot i_{abl}$, $i_{bcl}' = (d/dt) \cdot i_{bcl}$, $i_{cal}' = (d/dt) \cdot i_{cal}$, and $2\pi \cdot f_f$
 $= 1/(3 \cdot L_f \cdot C_f)$.

Given an uniform discretization interval T where the smallest interval is obtained with $T = 1/2f_s$, the discrete-time output filter state transition equations are

$$\begin{aligned} v_{abl}[(k + 1) \cdot T] = & \cos(2\pi \cdot f_f \cdot T) \cdot v_{abl}(k \cdot T) + \\ & [1/(2\pi \cdot f_f)] \cdot \sin(2\pi \cdot f_f \cdot T) \cdot v_{abl}'(k \cdot T) + \\ & [1 - \cos(2\pi \cdot f_f \cdot T)] \cdot [v_{abi}(k \cdot T) - L_f \cdot i_{abl}'(k \cdot T)] \end{aligned} \quad (1.61a)$$

$$\begin{aligned} v_{bcl}[(k + 1) \cdot T] = & \cos(2\pi \cdot f_f \cdot T) \cdot v_{bcl}(k \cdot T) + \\ & [1/(2\pi \cdot f_f)] \cdot \sin(2\pi \cdot f_f \cdot T) \cdot v_{bcl}'(k \cdot T) + \\ & [1 - \cos(2\pi \cdot f_f \cdot T)] \cdot [v_{bci}(k \cdot T) - L_f \cdot i_{bcl}'(k \cdot T)] \end{aligned} \quad (1.61b)$$

$$\begin{aligned} v_{cal}[(k + 1) \cdot T] = & \cos(2\pi \cdot f_f \cdot T) \cdot v_{cal}(k \cdot T) + \\ & [1/(2\pi \cdot f_f)] \cdot \sin(2\pi \cdot f_f \cdot T) \cdot v_{cal}'(k \cdot T) + \\ & [1 - \cos(2\pi \cdot f_f \cdot T)] \cdot [v_{cai}(k \cdot T) - L_f \cdot i_{cal}'(k \cdot T)] \end{aligned} \quad (1.61c)$$

where k is an integer.

Given target load voltages $\{v_{abl}^*[(k + 1) \cdot T], v_{bcl}^*[(k + 1) \cdot T], v_{cal}^*[(k + 1) \cdot T]\}$, the inverter output voltages required to synthesize these target load voltages are

$$\begin{aligned} v_{abi}^*(k \cdot T) = & [1 - \cos(2\pi \cdot f_f \cdot T)]^{-1} \cdot \{ v_{abl}^*[(k + 1) \cdot T] - \\ & \cos(2\pi \cdot f_f \cdot T) \cdot v_{abl}(k \cdot T) - \\ & [1/(2\pi \cdot f_f)] \cdot \sin(2\pi \cdot f_f \cdot T) \cdot v_{abl}'(k \cdot T) \} + \\ & L_f \cdot i_{abl}'(k \cdot T) \end{aligned} \quad (1.62a)$$

$$\begin{aligned}
v_{bci}^*(k \cdot T) = & [1 - \cos(2\pi \cdot f_f \cdot T)]^{-1} \cdot \{ v_{bcl}^*[(k + 1) \cdot T] - \\
& \cos(2\pi \cdot f_f \cdot T) \cdot v_{bcl}(k \cdot T) - \\
& [1/(2\pi \cdot f_f)] \cdot \sin(\omega_f \cdot T) \cdot v_{bcl}'(k \cdot T) \} + \\
& L_f \cdot i_{bcl}'(k \cdot T)
\end{aligned} \tag{1.62b}$$

$$\begin{aligned}
v_{cai}^*(k \cdot T) = & [1 - \cos(2\pi \cdot f_f \cdot T)]^{-1} \cdot \{ v_{cal}^*[(k + 1) \cdot T] - \\
& \cos(2\pi \cdot f_f \cdot T) \cdot v_{cal}(k \cdot T) - \\
& [1/(2\pi \cdot f_f)] \cdot \sin(2\pi \cdot f_f \cdot T) \cdot v_{cal}'(k \cdot T) \} + \\
& L_f \cdot i_{cal}'(k \cdot T).
\end{aligned} \tag{1.62c}$$

An ANN-based load voltage controller whose outputs are the modulation commands $\{v_{abin}^* = v_{abi}^*/v_{dc}, v_{bcin}^* = v_{bci}^*/v_{dc}, v_{cain}^* = v_{cai}^*/v_{dc}\}$ supplied to the PWM inverter switching control logic performs a functional approximation of network input/output data sets which are defined by the closed-loop transfer function which includes the output filter linear transfer functions given in (1.62a,b,c) and the nonlinear transfer characteristics associated with the inverter PWM technique and inverter switching. The purpose of the ANN controller or neuro-controller is to learn the appropriate mapping between network inputs and outputs via connection weight value adjustments and/or modification of the ANN structure such that the instantaneous load voltage waveform errors relative to the corresponding load voltage sinusoidal references are minimized.

The unique properties of the ANN controller are listed as follows:

- i.) The core controller design is independent of UPS system parameters such as the inverter DC power supply voltage, inverter modulation and switching frequencies, and output filter design and component

values. Characteristic of nonlinear controllers, the ANN controller is relatively insensitive to system parameter variations and modeling errors. Unlike conventional nonlinear controllers, the ANN controller can operate at relatively low sampling rates or inverter switching frequencies.

- ii.) Delays associated with controller operation and load voltage waveform synthesis are inherently compensated.

The application of the inverter neurocontroller is intended for high-power three-phase inverters where switching frequencies are drastically reduced, in the range of 300 to 700 Hertz, given the practical limitations associated with switching losses and EMI generation.

1.7 Scope

The problem addressed in this thesis is the development of a practical inverter neurocontroller which could be implemented in the subcycle control of load voltage waveforms generated by high-power three-phase UPS systems where inverter switching frequencies are relatively low in the range of 540 to 900 Hertz. The investigation of this problem is carried out in two stages; computer simulation of this control concept in a three-phase inverter/filter/load system given passive and nonlinear load conditions and experimental evaluation of an actual neurocontroller design in a three-phase inverter/filter/load circuit. Simulation software developed for this thesis described in

appendices 1 through 5 utilized a three-phase inverter model using gate turn-off thyristors or GTO's with Undeland snubber circuits with no energy recovery. Passive and nonlinear or three-phase uncontrolled rectifier load models were utilized. Software modules incorporating the neurocontroller scheme and other conventional control schemes into the simulation were developed.

The development of a neurocontroller which could be realistically implemented in an UPS system limited the investigations to neural network structures with no hidden layers and with one hidden layer. A practical implementation of the neurocontroller was limited by the controller software execution time. Subsequently, the experimental evaluation of the neurocontroller concept utilized a neural network with no hidden layers. Chapter 2 of this thesis contains a survey of neurocontroller techniques and a detailed description of the inverter neurocontroller. Chapter 3 contains the results of both computer simulations and experiments performed using the neurocontroller concept. Conclusions regarding the development of the neurocontroller concept are given in Chapter 4.

II. Adaptive Intelligent Inverter Control Method

2.1 Introduction

Multi-layered neural networks (MNN's) were originally developed for pattern recognition problems where the network training data is stationary and a real-time learning process is not required. MNN's have learning and self-organization capabilities, and so, they can adapt to changes in training data, learning the characteristics defined in the training data set. MNN's are used to learn a mapping between an input and an output space, and synthesize an associative memory that retrieves the appropriate output when presented with the input and generalize when presented with new inputs. Moreover, because of their nonlinear nature, MNN's can perform functional approximations and signal-filtering operations that are beyond the capabilities of optimal linear techniques.

It is common for the structure of the neural network to be compared to a model of the biological brain. In general, the neural network structure consists of many linear or nonlinear neuron (or node) models where information is processed in a parallel distributed manner. The neuron unit depicted in figure 2.1 consists of multiple inputs, $\underline{u}$, and one output y where y is the weighted sum of the input signals

$$y(t) = f\left[\sum_{i=1}^N w_i \cdot u_i(t) - \theta\right] \quad (2.1)$$

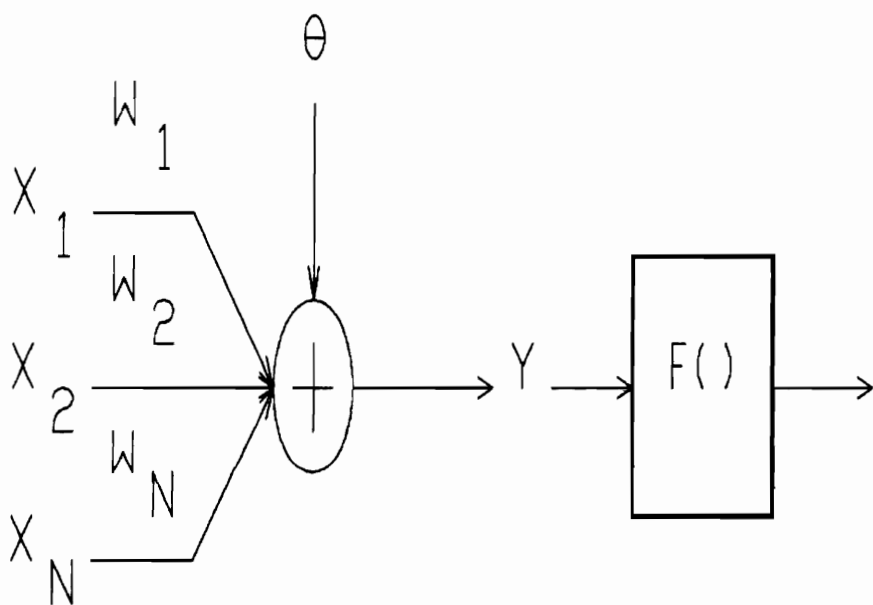


FIGURE 2.1: NEURON UNIT

where w_i is a connection weight, θ is a bias, and N is the number of inputs. $f()$ is the neuron output (or squashing) function which takes the following forms:

$f(x) = x$ or linear neuron function,

$f(x) = 1/[1 + e^{-ax}]$ or sigmoid function, and

$f(x) = [1 - e^{-ax}]/[1 + e^{-ax}]$ or hyperbolic tangent function.

Sigmoid and hyperbolic tangent squashing functions limit the neuron output to ranges $\{0,1\}$ and $\{-1,-1\}$ respectively. Additionally, these functions contribute to the nonlinearity of the neural network mapping.

Organization of the neural network is classified largely into two types: a recurrent or a feedforward network. In the recurrent network depicted in figure 2.2, multiple neurons are interconnected. The feedforward network has a hierarchical structure that consists of layers of neurons without interconnections between neurons in each layer. Signals flow in one direction from the network input to output. Figures 1.14A and 1.14B depict examples of feedforward neural networks. It is the feedforward neural network structure that is applied in the controller concept presented in this thesis.

In the discussion of feedforward neural network applications, it is common for the network to have one or more hidden layers between the network input and output layers. The term multi-layered neural network is applied to a neural network with at least one hidden layer. The role of the hidden layer is fundamental. A neural network without hidden layers will be unable to perform the necessary mappings between network

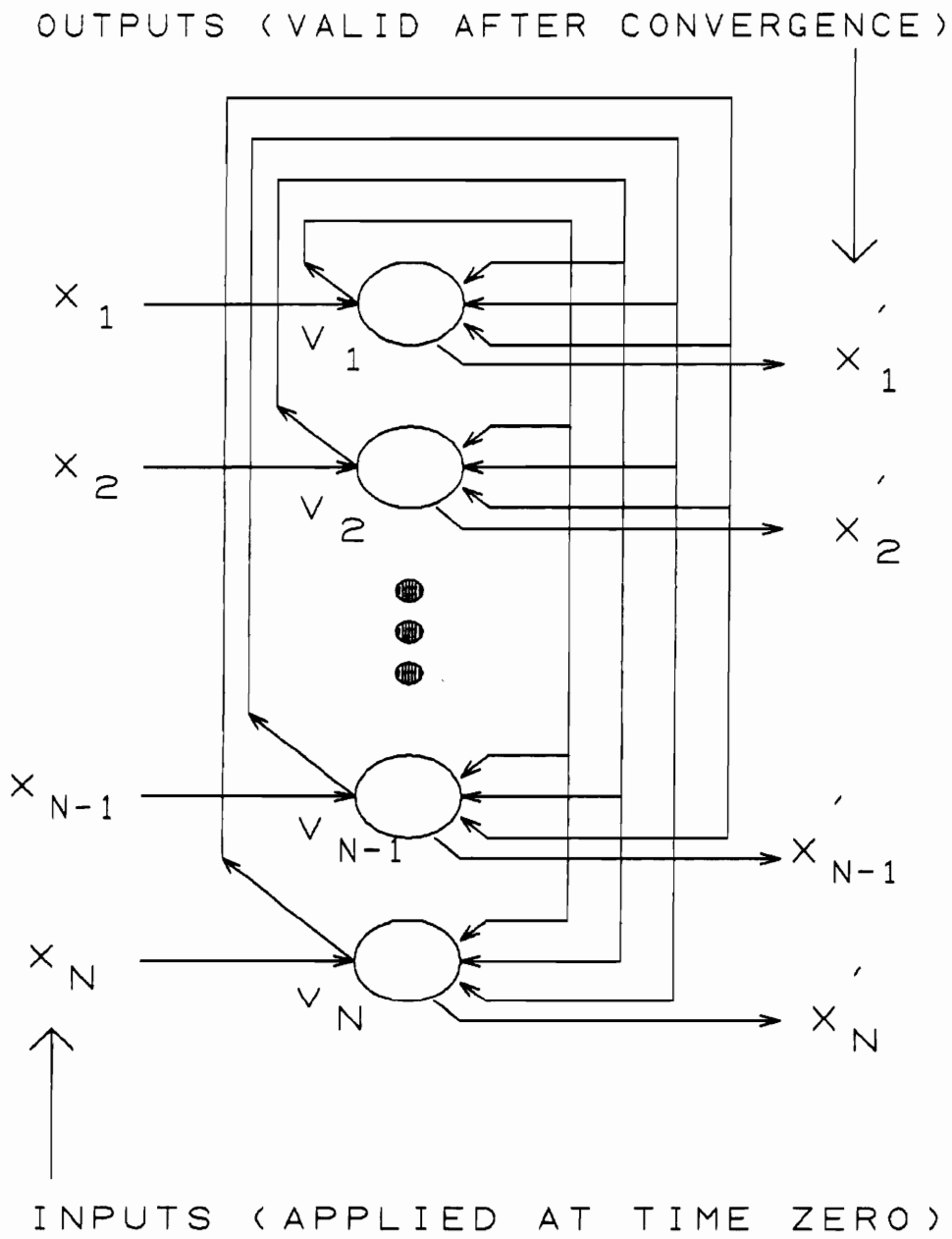


FIGURE 2.2: HOPFIELD NEURAL NETWORK

inputs and outputs associated with nonlinear processes such as the Boolean Exclusive-Or function.

Typical MNN application in a control application is either as a plant emulator or as a controller. A plant emulator is depicted in figure 2.3. Given the current plant input $\underline{u}(k)$ and q previous plant input samples, and p previous plant output samples, the plant emulator learns the plant transfer function $G()$ where

$$\begin{aligned} \underline{y}(k + 1) = G[\underline{y}(k), \underline{y}(k - 1), \dots, \underline{y}(k - p + 1), \\ \underline{u}(k), \underline{u}(k - 1), \dots, \underline{u}(k - q)]. \end{aligned} \quad (2.2)$$

The MNN learns a mapping between network inputs in the form of

$$\begin{aligned} \underline{y}(k), \underline{y}(k - 1), \dots, \underline{y}(k - p + 1), \text{ and} \\ \underline{u}(k), \underline{u}(k - 1), \dots, \underline{u}(k - q) \end{aligned}$$

and the network output $\underline{y}(k + 1)$ through the use of a learning technique (for example, the backpropagation algorithm) devised to minimize a norm of the emulation error

$$\underline{y}(k + 1) - \underline{y}_1(k)$$

where $\underline{y}_1(k)$ is the MNN output vector.

Of interest in this thesis is the use of a neural network as a controller (or neurocontroller) where figure 2.4 depicts a general neurocontroller configuration. Assuming that the plant is invertible, the plant input $\underline{u}(k)$ given the inverse dynamics of the plant is expressed as follows;

$$\begin{aligned} \underline{u}(k) = G^{-1}[\underline{r}(k + 1), \underline{y}(k), \dots, \underline{y}(k - p + 1), \\ \underline{u}(k - 1), \underline{u}(k - 2), \dots, \underline{u}(k - q)], \end{aligned} \quad (2.3)$$

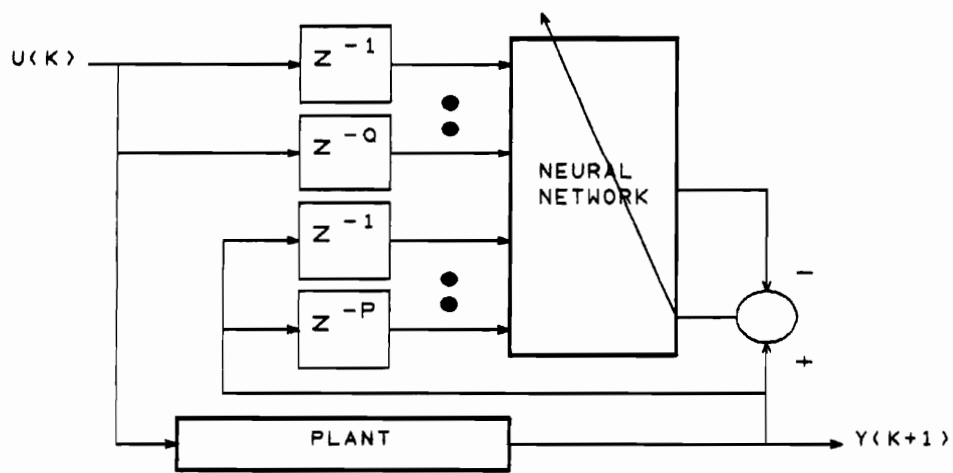


FIGURE 2.3: PLANT EMULATOR

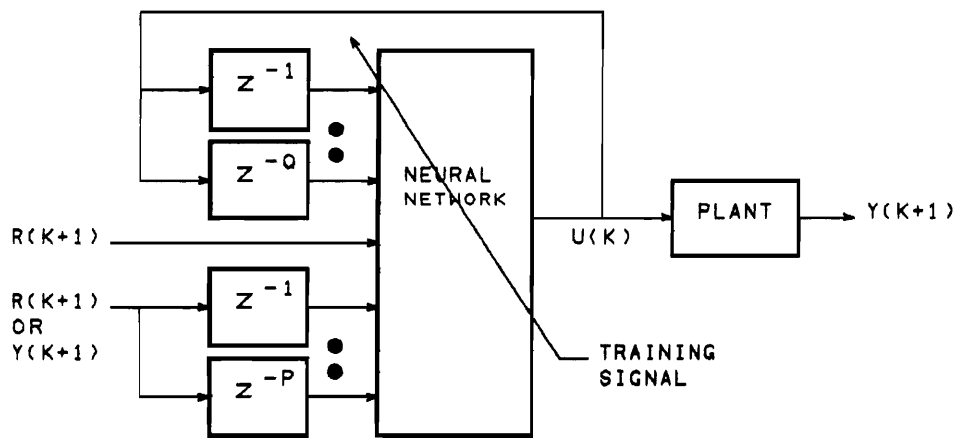


FIGURE 2.4: GENERAL NEUROCONTROLLER CONFIGURATION

where $\underline{r}(k + 1)$ is the desired plant output. The neurocontroller learns a mapping between network inputs in the form of

$$\underline{r}(k + 1), \underline{y}(k), \dots, \underline{y}(k - p + 1), \text{ and}$$

$$\underline{u}(k - 1), \underline{u}(k - 2), \dots, \underline{u}(k - q)$$

and the network output $\underline{u}(k)$ required to generate the desired plant output. The neurocontroller learns the inverse dynamics of the plant in a fashion that minimizes the plant output error

$$\underline{r}(k) - \underline{y}(k).$$

The plant in the UPS inverter/filter/load system is defined by the output filter circuit (depicted in figure 1.1) linear transfer characteristics and the inverter's nonlinear transfer characteristics (nonlinear with respect to simultaneous control of inverter output fundamental and harmonic voltages). From equations (1.6a) through (1.6c), the output filter inverse dynamics are described by the equations

$$v_{abi}^*(f) = [1 - 3 \cdot (2\pi \cdot f)^2 \cdot L_f \cdot C_f] \cdot v_{abl}^*(f) + j \cdot 2\pi \cdot f \cdot L_f \cdot i_{abl}(f), \quad (2.4a)$$

$$v_{bci}^*(f) = [1 - 3 \cdot (2\pi \cdot f)^2 \cdot L_f \cdot C_f] \cdot v_{bcl}^*(f) + j \cdot 2\pi \cdot f \cdot L_f \cdot i_{bcl}(f), \quad (2.4b)$$

and

$$v_{cai}^*(f) = [1 - 3 \cdot (2\pi \cdot f)^2 \cdot L_f \cdot C_f] \cdot v_{cal}^*(f) + j \cdot 2\pi \cdot f \cdot L_f \cdot i_{cal}(f), \quad (2.4c)$$

where $j = \sqrt{-1}$, f is the frequency, L_f is the output filter series inductance, C_f is the output filter line-to-line capacitance, $\{v_{abl}^*(f), v_{bcl}^*(f), v_{cal}^*(f)\}$ are the command line-to-line load voltages,

and $\{v_{abi}^*(f), v_{bci}^*(f), v_{cai}^*(f)\}$ are the line-to-line inverter output voltages based on the output inverse dynamics. An alternate set of discrete-time equations are derived from (1.66):

$$\begin{aligned} v_{abi}^*(k \cdot T) = & [1 - \cos(2\pi \cdot f_f \cdot T)]^{-1} \cdot \{ v_{abl}^*[(k + 1) \cdot T] - \\ & \cos(2\pi \cdot f_f \cdot T) \cdot v_{abl}(k \cdot T) - \\ & [1/(2\pi \cdot f_f)] \cdot \sin(2\pi \cdot f_f \cdot T) \cdot v_{abl}'(k \cdot T) \} + \\ & L_f \cdot i_{abl}'(k \cdot T) \end{aligned} \quad (2.5a)$$

$$\begin{aligned} v_{bci}^*(k \cdot T) = & [1 - \cos(2\pi \cdot f_f \cdot T)]^{-1} \cdot \{ v_{bcl}^*[(k + 1) \cdot T] - \\ & \cos(2\pi \cdot f_f \cdot T) \cdot v_{bcl}(k \cdot T) - \\ & [1/(2\pi \cdot f_f)] \cdot \sin(2\pi \cdot f_f \cdot T) \cdot v_{bcl}'(k \cdot T) \} + \\ & L_f \cdot i_{bcl}'(k \cdot T) \end{aligned} \quad (2.5b)$$

$$\begin{aligned} v_{cai}^*(k \cdot T) = & [1 - \cos(2\pi \cdot f_f \cdot T)]^{-1} \cdot \{ v_{cal}^*[(k + 1) \cdot T] - \\ & \cos(2\pi \cdot f_f \cdot T) \cdot v_{cal}(k \cdot T) - \\ & [1/(2\pi \cdot f_f)] \cdot \sin(2\pi \cdot f_f \cdot T) \cdot v_{cal}'(k \cdot T) \} + \\ & L_f \cdot i_{cal}'(k \cdot T). \end{aligned} \quad (2.5c)$$

where k is an integer, T is the inverter controller's sampling interval, $2\pi \cdot f_f = 1/(3 \cdot L_f \cdot C_f)$, $v_{abl}' = (d/dt) \cdot v_{abl}$, $v_{bcl}' = (d/dt) \cdot v_{bcl}$, $v_{cal}' = (d/dt) \cdot v_{cal}$, $i_{abl}' = (d/dt) \cdot i_{abl}$, $i_{bcl}' = (d/dt) \cdot i_{bcl}$, and $i_{cal}' = (d/dt) \cdot i_{cal}$.

The problem of simultaneously controlling the inverter output voltage fundamental magnitude while selectively generating or eliminating waveform harmonics is similar to the technique described in references [8] and [9]. A system of nonlinear equations with N dependent

variables $\{\beta_1, \beta_2, \dots, \beta_N\}$ (inverter output waveform switching angles) is expressed in the form

$$f_i(\beta_1, \beta_2, \dots, \beta_N) = 0, \quad i = 1, 2, \dots, N. \quad (2.6)$$

The normalized ideal inverter leg circuit output voltage waveform's (normalized with respect to $v_{dc}/2$) Fourier Series coefficients are,

$$a_n = (4/n\pi) \cdot [1 + 2 \cdot \sum_{k=1}^N (-1)^k \cdot \cos(n \cdot \beta_k)], \quad (2.7)$$

and they are equated to fundamental and harmonic magnitude commands a_n^* forming the set of nonlinear equations in the form of (2.6) or

$$f_i(\beta_1, \beta_2, \dots, \beta_N) = a_n^* - a_n = 0. \quad (2.8)$$

Given closed-loop load voltage control, a_n^* for $n = 1, 2, \dots, N$ and for each phase is obtained from the output filter inverse dynamics described in equations (2.4a) through (2.4c) where $w = n \cdot w_m$, w_m is the fundamental angular frequency.

The forward transfer characteristics of the ideal inverter PWM waveform generation process are linear when waveform control is limited to the fundamental component and inverter output waveform symmetry is maintained. This condition under closed-loop load voltage control corresponds to balanced passive UPS loads where $\{v_{abi}^*, v_{bci}^*, v_{cai}^*\}$ are symmetrical three-phase voltages. Load voltage controller PWM switching control logic level inverter modulation commands $\underline{v}_{abi}^* = v_{abi}^*/v_{dc}$, $\underline{v}_{bci}^* = v_{bci}^*/v_{dc}$, $\underline{v}_{cai}^* = v_{cai}^*/v_{dc}$, are linearly transformed by the inverter waveform generation process to the inverter output voltages

$\{v_{abi}^*, v_{bci}^*, v_{cai}^*\}$ where unwanted inverter-generated harmonics are suppressed by the output filter circuit.

Given the assumption that the inverter is a distortion-free linear power amplifier, the inverse dynamics of the inverter/filter system are described by equations proportional to either (2.4a) through (2.4c) or (2.5a) through (2.5c). Based on these equations, inputs to a neurocontroller used in a load voltage control application are:

- i.) load voltage commands $\{v_{abl}^*(k+1), v_{bcl}^*(k+1), v_{cal}^*(k+1)\}$,
- ii.) load voltage feedback $\{v_{abl}(k), v_{bcl}(k), v_{cal}(k)\}$, and
- iii.) load current feedback in the form of current samples $\{i_{la}(k), i_{lb}(k), i_{lc}(k)\}$ and previous samples $\{i_{la}(k-1), i_{lb}(k-1), i_{lc}(k-1)\}$.

The neurocontroller outputs are $\{\underline{v}_{abi}^*(k), \underline{v}_{bci}^*(k), \underline{v}_{cai}^*(k)\}$. The plant output errors are $\epsilon_{abl}(k) = v_{abl}^*(k) - v_{abl}(k)$, $\epsilon_{bcl}(k) = v_{bcl}^*(k) - v_{bcl}(k)$, and $\epsilon_{cal}(k) = v_{cal}^*(k) - v_{cal}(k)$. The inverter neurocontroller learns the inverse dynamics of the inverter/filter system in a fashion that minimizes these errors.

Due to the effects of inverter switching and inverter DC power supply harmonic voltage distortion, nonlinear UPS load currents, and three-phase inverter switching constraints associated with the generation of asymmetrical inverter output voltage waveforms (a condition resulting from load voltage waveform compensation with unbalanced UPS loads), the forward transfer characteristics of the inverter are no longer linear. Switching angle solutions to equation (2.8) $\{\beta_1^{(0)}, \beta_2^{(0)}, \dots, \beta_N^{(0)}\}$ for each phase are dependent on Fourier

Series coefficients commands a_n^* which are either unequal (phase-to-phase comparison) or are non-zero at harmonics typically associated with either inverter switching and DC power supply distortion and nonlinear load currents (for example, harmonics in the order of the 3th, 5th, 7th, ...). The ability of the inverter to selectively-generate harmonics in response to load voltage waveform compensation requirements is limited by the number of waveform switching angles per fundamental frequency cycle N (a function of the inverter switching frequency f_s). Under these conditions, solutions to the set of nonlinear equations described by (2.8) are typically best-fit solutions approaching exact solutions with increasing inverter switching frequency (or controller sampling rate).

The modulation commands $\{\underline{v}_{abi}^*(k), \underline{v}_{bci}^*(k), \underline{v}_{cai}^*(k)\}$ output from any closed-loop load voltage controller designed to minimize the steady-state load voltage waveform errors $\{\epsilon_{abl}(k), \epsilon_{bcl}(k), \epsilon_{cal}(k)\}$ should result in the generation of inverter output voltage waveforms described by the sets of switching angle $\{\beta_1^{(0)}, \beta_2^{(0)}, \dots, \beta_N^{(0)}\}$ solutions for each phase. The best performance of any closed-loop load voltage control technique with regard to the minimization of steady-state load voltage waveform error magnitudes is dependent on the PWM waveform generation process where optimum inverter output waveform generation in response to load voltage waveform compensation requirements is described by the waveform switching angles $\{\beta_1^{(0)}, \beta_2^{(0)}, \dots, \beta_N^{(0)}\}$.

The stimulus for the investigation of neurocontroller application in an UPS load voltage control system is the ability of the neurocontroller to learn the required inverter output waveform

compensation without a dependence on knowledge of the inverter/filter/load system in the form of parameters such as the inverter modulation and switching frequencies, the output filter component values, or an assumption made concerning the load model. The neurocontroller can adapt in an intelligent manner to a broad range of UPS system configurations requiring only one controller gain adjustment (in the form of the neural network learning-rate parameter). The two general functions of a conventional linear controller, fast response to sudden load changes and minimization of steady-state load voltage errors, are combined in the neurocontroller structure.

The target application of the inverter neurocontroller is in high-power three-phase UPS inverters where the inverter switching frequency is reduced by practical limitations resulting in relatively long controller sampling intervals.

2.2 Neurocontroller Configurations and Learning Techniques

Neurocontrollers learn by modifying the network connection weight values. A common learning technique is the backpropagation algorithm, a gradient descent method of finding the minimum of the objective function $\underline{E}(k) \equiv \frac{1}{2} \cdot [\underline{r}(k) - \underline{y}(k)]^2$. Connection weight $w_{ij}(k)$ is modified by the backpropagation algorithm in a manner described by the equation

$$w_{ij}(k + 1) = w_{ij}(k) - c(k) \cdot \delta \underline{E}(k) / \delta w_{ij} \quad (2.9)$$

where $c(k)$ is referred to as the learning-rate parameter. The term $\delta E(k)/\delta w_{ij}$ is defined based on the neurocontroller's learning process configuration.

In the direct adaptive control configuration depicted in figure 2.5, the exact computation of the term

$$\delta E(k)/\delta w_{ij} = - [\underline{r}(k) - \underline{y}(k)] \cdot \delta \underline{y}(k)/\delta w_{ij}$$

requires knowledge of the Jacobian of the plant. In many practical applications where the direction of the gradient of $E(k)$ is sufficient, the term $\delta E(k)/\delta w_{ij}$ can be estimated or replaced with the polarity of this derivative. This approach is similar to the one used in the inverter neurocontroller application.

A hybrid control scheme is depicted in figure 2.6. In this control scheme, the neurocontroller augments the operation of a conventional controller. When the absolute plant output error $|\underline{r}(k) - \underline{y}(k)|$ exceeds a preselected threshold, the conventional controller is switched into the control loop. The neurocontroller is disconnected and its learning process is continued. In this state, the neurocontroller learning process is in a direct inverse control configuration depicted in figure 2.7. The error $\underline{u}(k) - \underline{u}_1(k)$ is used in the backpropagation algorithm where $\underline{u}_1(k)$ is the neurocontroller's output. When the absolute value of this error is within a preselected threshold, the neurocontroller is reconnected, and the conventional controller is disconnected. The virtue of this approach is the fact that the neurocontroller's learning process has a minimum adverse effect on the system's performance.

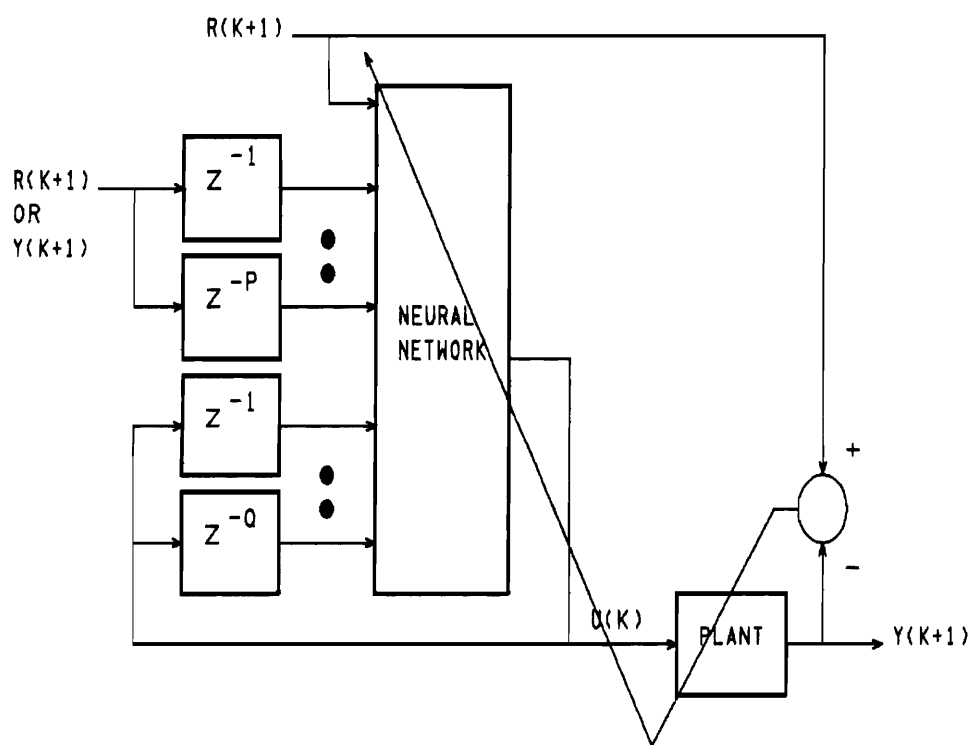


FIGURE 2.5: DIRECT ADAPTIVE CONTROL CONFIGURATION

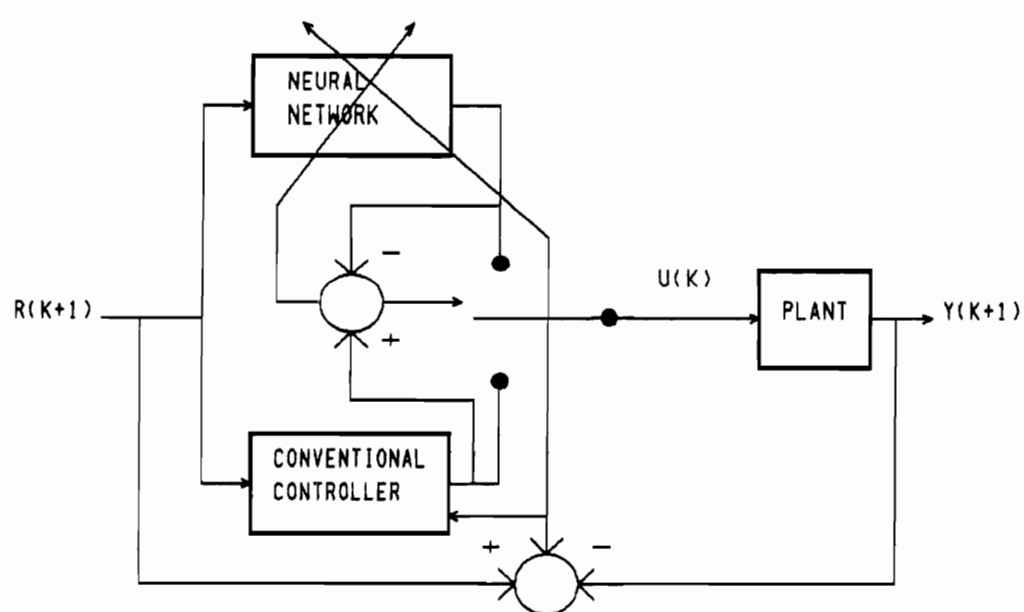


FIGURE 2.6: HYBRID CONTROL SCHEME

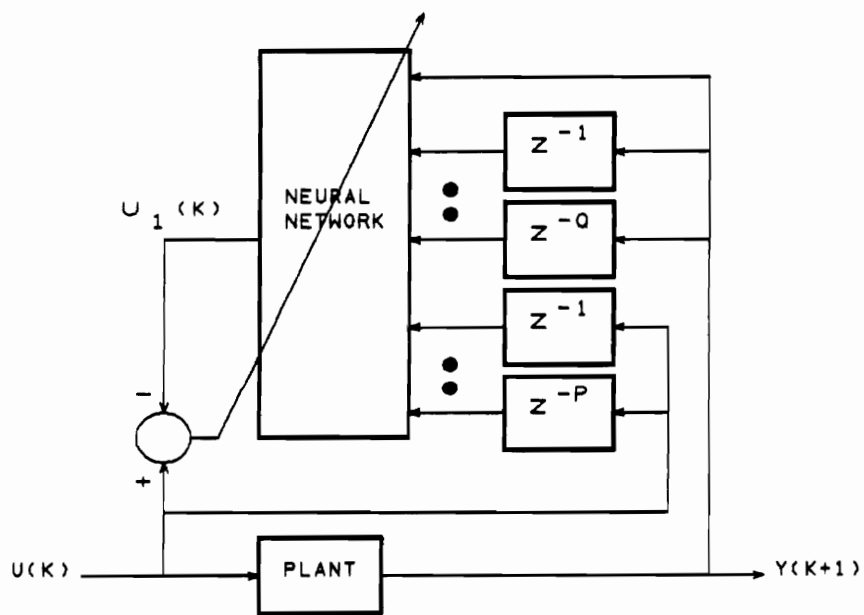


FIGURE 2.7: DIRECT INVERSE CONTROL CONFIGURATION

Other control configurations incorporate the use of a plant emulator. For example, in figure 2.8 an indirect learning architecture is depicted. The objective of the learning processes of both networks is to minimize the error $\underline{u}(k) - \underline{v}(k)$ where $\underline{v}(k)$ is the plant emulator output. By minimizing this error, the plant output error is reduced as well (an assumption given the existence of a plant transfer function which is invertible). The plant emulator learning process can also be used to compute the sensitivity of the error $\underline{r}(k) - \underline{y}(k)$ with respect to the controller's output $\underline{u}(k)$. The term $\delta \underline{y}(k) / \delta w_{ij}$ applied in the controller's learning process is derived from this sensitivity $\delta \underline{y}(k) / \delta \underline{u}(k)$. This approach referred to as indirect adaptive control is depicted in figure 2.9. Typically, the plant emulator function is initially trained off-line.

Numerous other control configurations have been proposed where a neural network is integrated into a more conventional adaptive controller scheme where knowledge of the plant model is at least partially available. Given the scope of this thesis, a discussion of these configurations will be avoided. Additionally, variations of the neurocontroller concept tend to increase in complexity. Implementation of neurocontroller configurations is limited by practical considerations associated with contemporary digital controller hardware. In applications such as robotic control, increased neurocontroller complexity may be acceptable given the nature of the application. Inverter control (and power conversion applications in general) operate at rates or frequencies which greatly exceed the rates robotic

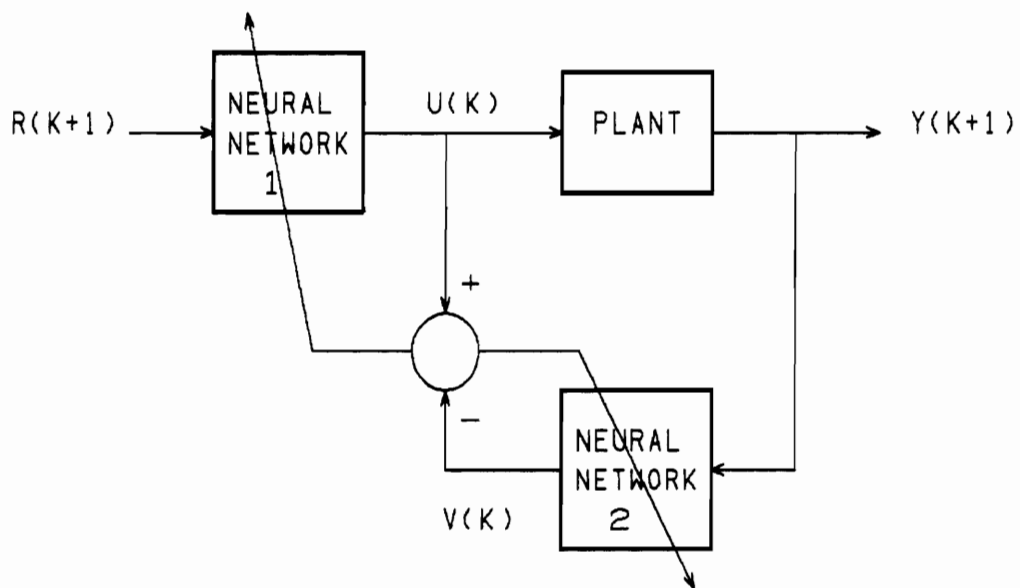


FIGURE 2.8: INDIRECT LEARNING ARCHITECTURE

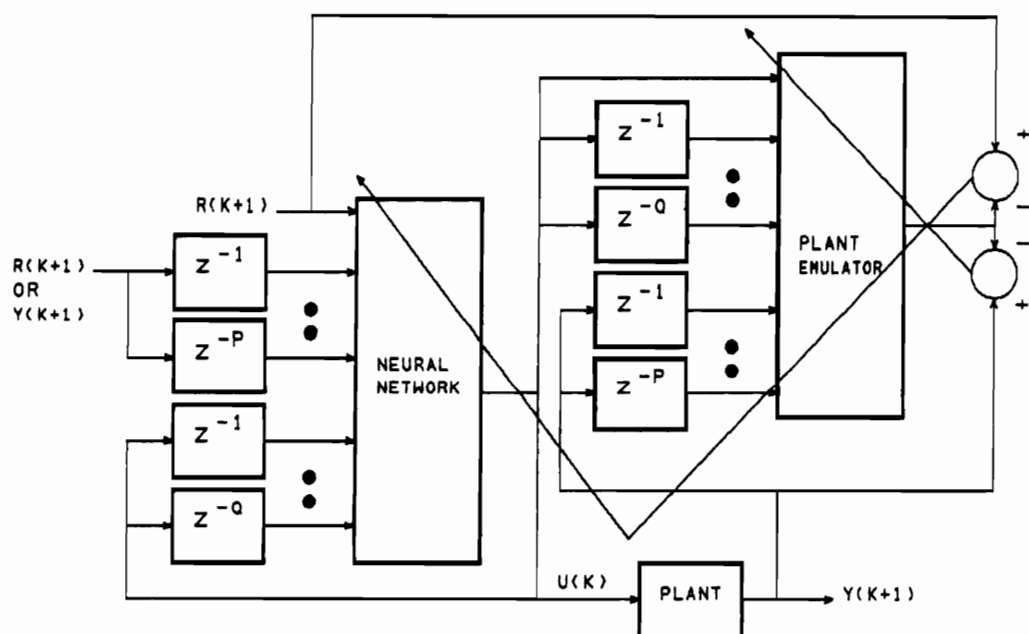


FIGURE 2.9: INDIRECT ADAPTIVE CONTROL CONFIGURATION

applications typically operate. Assumptions concerning the time-invariance of controller training data set (plant feedback) are not necessarily valid in the case of inverter control. Therefore, neurocontroller configurations with increased complexity could, in the case of inverter control, lack sufficient speed response to new learning situations to be of practical use.

The conventional backpropagation algorithm has two disadvantages which are not desirable in a control application:

- i.) The learning-rate parameter is typically made a small value such that the change in neural network weights cause the neurocontroller learning process to follow the true path of steepest descent. As a result, the learning process crawls along the descent path, requiring a large number of learning iterations to minimize the error objective.
- ii.) Given an objective function with local minimums in addition to an absolute minimum, convergence to a minimum other than the absolute one is a possibility depending on the initial connection weight values. This problem is displayed in the form of steady - state controller operation at a plant output error other than the desired plant output error, $\underline{r}(k) - \underline{y}(k) = \underline{0}$.

Additionally, it is typically assumed that the neurocontroller training data is reasonably stationary. In the event that on-line training of the neurocontroller invokes a significant response from a dynamic plant, this assumption is no longer valid. As a result, the gradient descent

process may be interrupted by changes in the training data set. Of course, one method to minimize this problem is to apply the hybrid control scheme depicted in figure 2.6. The singular advantage of the backpropagation technique is its relative simplicity with regard to its implementation in a practical neurocontroller design.

Alternate methods of neural network training are devised with the purpose of improving the network learning rate as well as avoid the problems associated with the use of the conventional backpropagation algorithm. For example, reference [28] describes an adaptive training neural network training method devised to enhance the learning process despite the use of nonstationary training data. The algorithm in this paper uses nonlinear programming methods to adapt to temporally nonstationary training data. A restriction placed on this method is that the training data's variation with time is slow.

Reference [29] proposes alternate method of neural network training which are devised to define the network structure and promote faster network learning. In reference [29], an estimate probability density function (PDF) is generated from the training data $\{\underline{x}, \underline{d}\}$ where $\underline{x}$ corresponds to the network input vector, and $\underline{d}$ corresponds to the desired network output vector. Given this estimate PDF, the conditional PDF and the conditional expected value of the network output vector given the network input vector, $E[\underline{y}|\underline{x}]$, can be computed. The neural network structure is based on the mathematical expression describing the conditional expected value. This reference also includes a discussion on the use of clustering techniques as a means of limiting the number of

nodes in the network structure. The reported advantages of this technique are:

- i.) The network learns in one pass through the training data and can generalize from examples as soon as they are stored.
- ii.) The estimate converges to the conditional mean regression surfaces as more examples are observed.
- iii.) The estimate is bounded by the minimum and maximum of the observations.
- iv.) The estimate can not converge to poor solutions corresponding to local minima of the error criterion.
- v.) The clustering versions of this algorithm limit the number of nodes in the network structure.

Reference [29] is given as an example of alternate neural network training algorithm because its methodology is relatively obvious. The neural network structure derived from this approach is based on an analytical basis rather than an arbitrary selection of neural network hidden units. Reference [30] is another example of a training algorithm with an analytical basis where decision trees are used to design the network structure, but its methodology is not as obvious particularly with regard to its use in a control application.

A critical limitation associated with the use of sophisticated learning schemes in control applications is the amount of digital controller execution time required by these schemes, as well as, the hardware requirements associated with their use. Control applications,

unlike applications such as pattern recognition, have restrictive time constraints associated with on-line learning. Though neurocontrollers are not known to have the fast-response characteristics of more conventional control methods, it is expected that new learning situations, a common event, are resolved in the minimum possible time interval. By necessity, learning algorithms in neurocontrollers are as brief as possible. Additionally, contemporary practical controller hardware designs are intentionally limited, as they generally do not have the design flexibility required by algorithms described in references [29] and [30], as well as other similar techniques.

2.3 The Inverter Neurocontroller Technique

Linear UPS inverter control methods are well-defined and established techniques. A competitive control technique, such as the neurocontroller technique, would have to offer an obvious advantage. Off-line neural network training as employed in a hybrid control scheme depicted in figure 2.6 or the direct inverse control scheme depicted in figure 2.7 merely trains or programs a neurocontroller to perform the control function which a linear controller could perform. The complexity of neurocontroller configurations such as the indirect learning architecture depicted in figure 2.8 and the indirect adaptive control scheme depicted in figure 2.9 are not justified given the practical constraints associated with implementing a inverter neurocontroller.

The inverter neurocontroller technique described in this thesis uses a variation of the direct adaptive control scheme depicted in figure 2.5. Based on the equations describing the output filter inverse dynamics, the forward transfer characteristics of the inverter neurocontroller are described by the equations

$$\begin{aligned}\underline{v}_{abi}^*(k \cdot T) = & g_{11}(k \cdot T) \cdot \underline{v}_{abl}^*[(k + 1) \cdot T] + \\ & g_{12}(k \cdot T) \cdot \underline{v}_{abl}(k \cdot T) + g_{13}(k \cdot T) \cdot \underline{v}_{abl}[(k - 1) \cdot T] + \\ & g_{14}(k \cdot T) \cdot \{\underline{i}_{abl}(k \cdot T) - \underline{i}_{abl}[(k - 1) \cdot T]\}, \quad (2.10a)\end{aligned}$$

$$\begin{aligned}\underline{v}_{bci}^*(k \cdot T) = & g_{21}(k \cdot T) \cdot \underline{v}_{bcl}^*[(k + 1) \cdot T] + \\ & g_{22}(k \cdot T) \cdot \underline{v}_{bcl}(k \cdot T) + g_{23}(k \cdot T) \cdot \underline{v}_{bcl}[(k - 1) \cdot T] + \\ & g_{24}(k \cdot T) \cdot \{\underline{i}_{bcl}(k \cdot T) - \underline{i}_{bcl}[(k - 1) \cdot T]\}, \quad (2.10b)\end{aligned}$$

$$\begin{aligned}\underline{v}_{cai}^*(k \cdot T) = & g_{31}(k \cdot T) \cdot \underline{v}_{cal}^*[(k + 1) \cdot T] + \\ & g_{32}(k \cdot T) \cdot \underline{v}_{cal}(k \cdot T) + g_{33}(k \cdot T) \cdot \underline{v}_{cal}[(k - 1) \cdot T] + \\ & g_{34}(k \cdot T) \cdot \{\underline{i}_{cal}(k \cdot T) - \underline{i}_{cal}[(k - 1) \cdot T]\}, \quad (2.10c)\end{aligned}$$

where k is an integer, T is the controller sampling interval, $\{\underline{v}_{abi}^*, \underline{v}_{bci}^*, \underline{v}_{cai}^*\}$ are the logic-level three-phase inverter modulation commands, $\{\underline{v}_{abl}^*, \underline{v}_{bcl}^*, \underline{v}_{cal}^*\}$ are the logic-level three-phase load voltage commands, $\{\underline{v}_{abl}, \underline{v}_{bcl}, \underline{v}_{cal}\}$ are the logic-level three-phase load voltage feedback signals, $\{\underline{i}_{abl}, \underline{i}_{bcl}, \underline{i}_{cal}\}$ are the logic-level three-phase load current feedback signals, and g_{ij} ($1 \leq i \leq 3$, $1 \leq j \leq 4$) is a controller gain. In a single-layer neurocontroller, the controller gain g_{ij} corresponds to a network connection weight. In a multi-layer neurocontroller, the controller gain g_{ij} corresponds to the total network input/output gain which is dependent on the connection weight

values and squashing function characteristics in the parallel paths taken by a network input signal to a specific network output. With the use of a hyperbolic tangent squashing function (a bipolar signal limiting function), the controller gain g_{ij} in a multi-layer neurocontroller is a nonlinear function of the neurocontroller inputs.

Initial investigation of the use of a neural network in an inverter control system utilized the control configuration depicted in figure 2.10. The current and voltage signals are defined in the d-q frame of reference where the following expression describes the transformation from a three-phase frame to the d-q frame:

$$\begin{bmatrix} e_d \\ e_q \end{bmatrix} = \sqrt{(2/3)} \cdot \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \quad (2.11)$$

Signals in this control scheme maintain their AC characteristics after this transformation.

A neurocontroller with one hidden layer and hyperbolic tangent squashing functions was utilized in this investigation. The use of a neural network with a hidden layer is consistent with the idea that hidden layers are required to enable the neural network to map nonlinear functions. The inverse dynamics of the inverter/filter system was expected to be a nonlinear system. A network version using six hidden layer nodes was found to perform in a satisfactory manner. Versions with four and eight hidden layer nodes were evaluated. The performance of the eight-node version did not display a distinct improvement over the

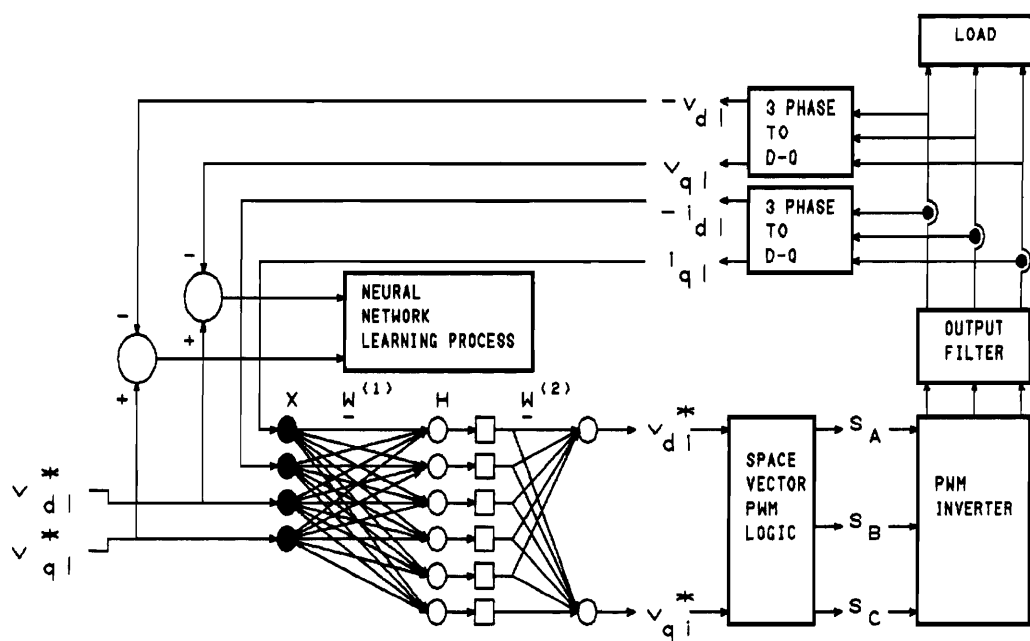


FIGURE 2.10: INITIAL NEUROCONTROLLER CONFIGURATION

six-node version. On the other hand, the four-node version required a longer training interval to reach a steady-state condition.

The operation of the neurocontroller is synchronized to the space vector PWM process. At discrete-time intervals ($1/2f_s$ in duration where f_s is the inverter switching frequency), the space vector PWM process converts the d-q frame modulation commands $\{\underline{v}_{di}^*(k), \underline{v}_{qi}^*(k)\}$ to three inverter switching control signals $\{S_a, S_b, S_c\}$ (each signal controls the switching of one inverter leg circuit). Due to the computational time required to implement this process, these switching control signals generated by the space vector PWM control logic in the k^{th} controller sampling interval are executed one controller sampling interval later. Output filter circuit current and voltage waveform synthesis corresponding to these inverter switching commands takes place during the $(k + 1)$ sampling interval resulting in load currents and voltages sampled by the controller at the beginning of the $(k + 2)$ interval. The sampled d-q frame load voltages and currents, $\{\underline{v}_{dl}(k), \underline{v}_{ql}(k)\}$ and $\{\underline{i}_{dl}^{(1)}(k), \underline{i}_{ql}^{(1)}(k)\}$, which result in the generation of inverter modulation commands $\{\underline{v}_{di}^*(k), \underline{v}_{qi}^*(k)\}$ are the result of inverter modulation commands generated two sampling intervals previously or $\{\underline{v}_{di}^*(k - 2), \underline{v}_{qi}^*(k - 2)\}$.

The neurocontroller, at the beginning of the k^{th} PWM sampling interval, samples the load voltage commands, load voltages and currents. Sampled d-q frame load currents and voltages are transformed prior to their application to the neural network in the following manner:

$$\underline{v}_l^{(q)}(k) = \{\underline{v}_{dl}^{(q)}(k), \underline{v}_{ql}^{(q)}(k)\} = \{\underline{v}_{ql}(k), -\underline{v}_{dl}(k)\} \quad (2.12)$$

and

$$\underline{i}_1(q)(k) = \{\underline{i}_{d1}(q)(k), \underline{i}_{q1}(q)(k)\} = \{\underline{i}_{q1}^{(1)}(k), -\underline{i}_{d1}^{(1)}(k)\}. \quad (2.13)$$

This transform phase-shifts the load current and voltage vectors by 90° placing these vectors in the same quadrant as the sampled d-q frame load voltage commands $\{\underline{v}_{d1}^*(k), \underline{v}_{q1}^*(k)\}$. The load voltage commands are in the following form:

$$\underline{v}_{d1}^*(t) = M \cdot \sin(2\pi f_m t) \quad (2.14)$$

and

$$\underline{v}_{q1}^*(t) = -M \cdot \cos(2\pi f_m t) \quad (2.15)$$

where M is the modulation index and f_m is the inverter's modulation frequency.

Using the neurocontroller configuration depicted in figure 2.10, network inputs are forward propagated through the neurocontroller in a manner described by the following equations:

$$h_u(k) = \sum_{i=1}^4 x_i(k) \cdot w_{iu}^{(1)}(k), \quad 1 \leq u \leq 6, \quad (2.16)$$

where $x_1(k) = \underline{v}_{d1}^*(k)$, $x_2(k) = \underline{v}_{q1}^*(k)$, $x_3(k) = \underline{i}_{d1}^{(q)}(k)$, $x_4(k) = \underline{i}_{q1}^{(q)}(k)$,

$$\underline{v}_{di}^*(k) = \sum_{u=1}^6 \tanh[h_u(k)] \cdot w_{ui}^{(2)}(k), \quad (2.17)$$

and

$$\underline{v}_{qi}^*(k) = \sum_{u=1}^6 \tanh[h_u(k)] \cdot w_{ui}^{(2)}(k) \quad (2.18)$$

where x_1 through x_4 are the neurocontroller inputs, $w_{iu}^{(1)}(k)$ are the network connection weights in the k^{th} interval associated with the connection of the network input and hidden layers, and $w_{uj}^{(2)}$ are the network connection weights in the k^{th} interval associated with the connection of the network hidden and output layers.

The conventional backpropagation algorithm was implemented in the neurocontroller's learning process. The objective of the learning process is to minimize the sampled load voltage errors given by

$$\underline{\epsilon}_{dv}(k) = \underline{v}_{dl}^*(k) - \underline{v}_{dl}^{(q)}(k) \quad (2.19)$$

and

$$\underline{\epsilon}_{qv}(k) = \underline{v}_{q1}^*(k) - \underline{v}_{q1}^{(q)}(k). \quad (2.20)$$

The backpropagation algorithm took into account the fact that the sampled load voltage errors were the result of the forward propagation of network inputs present two sampling intervals previously. Network weight updates are then generated using the following equations:

$$\delta w_{u1}^{(2)}(k) = \underline{\epsilon}_{dv}(k) \cdot \tanh[h_u(k-2)] \quad (2.21)$$

$$\delta w_{u2}^{(2)}(k) = \underline{\epsilon}_{qv}(k) \cdot \tanh[h_u(k-2)], \quad 1 \leq u \leq 6 \quad (2.22)$$

$$w_{u1}^{(2)}(k+1) = w_{u1}^{(2)}(k) + c(k) \cdot \delta w_{u1}^{(2)}(k) \quad (2.23)$$

$$w_{u2}^{(2)}(k+1) = w_{u2}^{(2)}(k) + c(k) \cdot \delta w_{u2}^{(2)}(k) \quad (2.24)$$

$$\delta w_{iu}^{(1)}(k) = [\underline{\epsilon}_{dv}(k) \cdot w_{u1}^{(2)}(k) + \underline{\epsilon}_{qv}(k) \cdot w_{u2}^{(2)}(k)] \cdot \text{sech}^2[h_u(k-2)] \cdot x_i(k-2) \quad (2.25)$$

$$w_{iu}^{(1)}(k+1) = w_{iu}^{(1)}(k) + c(k) \cdot \delta w_{iu}^{(1)}(k) \quad (2.26)$$

where $c(k)$ is the learning-rate parameter used in the k^{th} sampling interval. In each controller sampling interval, the learning process was executed after the forward propagation process, and prior to the

execution of the space vector PWM process. This is a critical factor which subsequently limited the implementation of this neurocontroller in an experimental controller design.

Using the inverter/filter/load simulation software developed for this thesis, this neurocontroller version was evaluated using both a balanced passive inductive load and a nonlinear load model. An inverter modulation frequency f_m of 60 Hertz and switching frequency f_s of 900 Hertz were selected for the simulations. Space vector PWM was used in all simulations. The output filter circuit used is the configuration depicted in figure 1.2. Additionally, a dead time interval of 60 μs was inserted in the inverter switching control signals which is consistent with the use of GTO and SCR technology. The neural network was initially trained off-line where the initial network weights were randomly selected. Off-line training was configured to minimize the errors

$$\underline{v}_{dl}^*(k) - \underline{v}_{di}^*(k) \text{ and } \underline{v}_{lq}^*(k) - \underline{v}_{iq}^*(k),$$

where the load currents network inputs were kept at zero. Once off-line training was completed, the actual inverter/filter/load simulations were run. Specific results of simulations and experiments are the topics of chapter 3.

One of the initial concerns regarding the application of the neurocontroller technique was the closed-loop response of the inverter/filter/load system during the neurocontroller learning process. In this control scheme, there are two control loops in operate:

- i.) The loop associated with the sampled load current feedback, an input to the neurocontroller's network, and

- ii.) The loop associated with the neurocontroller's learning process where sampled load voltage feedback is utilized.

The loop associated with the neurocontroller's learning process is a negative feedback loop. The loop associated with the sampled load current feedback is a positive feedback loop. Controller instability as well as cyclic variations of the load voltage error trajectories about the desired error surfaces $\underline{\epsilon}_{dv} = 0$ and $\underline{\epsilon}_{qv} = 0$ during steady-state controller operation are attributed primarily to the positive feedback effect of the load current loop.

Both neurocontroller simulations and experiments were begun with zero value initial conditions in the inverter/filter/load circuitry and without neurocontroller off-line training in a closed-loop circuit. Initially, the forward transfer characteristics of the neurocontroller do not match the forward transfer characteristics of the inverter/filter system, promoting the opportunity for unstable system operation due to the presence of positive load current feedback. Instability is avoided by the selection of the neurocontroller's learning-rate parameter and by the response of the output filter circuit which limits the load current rate of rise.

In the balanced passive load simulation using the neurocontroller depicted in figure 2.10, a learning-rate parameter value of 1.0 during the neurocontroller learning process was utilized without creating unstable system operation. In the remainder of the neurocontroller simulations and experiments where either nonlinear loads or less complex

neurocontroller structures (fewer hidden layer nodes or single-layer networks) were utilized, learning-rate parameter values were reduced significantly to avoid unstable system operation.

By reducing the learning-rate parameter value, the size of neurocontroller weight step changes during the learning process is reduced as well. This corresponds to a reduction in the step change of the controller gain associated with the load current loop during the learning process. In the neurocontroller simulations and experiments, the neurocontroller load current loop gain is initially a lower value than the value, L_f/T , associated with the output filter inverse dynamics described in equations (2.5a) through (2.5c):

$$L_f \cdot i_{abl}'(k \cdot T) = (L_f/T) \cdot \{i_{abl}(k \cdot T) - i_{abl}[(k - 1) \cdot T]\} \quad (2.27)$$

$$L_f \cdot i_{bcl}'(k \cdot T) = (L_f/T) \cdot \{i_{bcl}(k \cdot T) - i_{bcl}[(k - 1) \cdot T]\} \quad (2.28)$$

$$L_f \cdot i_{cal}'(k \cdot T) = (L_f/T) \cdot \{i_{cal}(k \cdot T) - i_{cal}[(k - 1) \cdot T]\}. \quad (2.29)$$

As the neurocontroller learning process proceeds, the neurocontroller's load current loop gain approaches the gain value described by the inverter/filter system inverse dynamics in steps. By limiting the gain step size, the convergence of the neurocontroller load current loop gain towards the corresponding inverter/filter inverse dynamics gain does not include an overshoot condition (similar to an underdamped response) avoiding possible system instability.

The manner in which the neurocontroller forward transfer characteristics converge to the corresponding characteristics of the inverter/filter system inverse dynamics during the neurocontroller learning process is also a function of the neurocontroller structure. A

general observation based on neurocontroller simulations and experiments is that the learning characteristics of less complex neurocontroller structures are more likely to result in a load current loop positive feedback gain which overshoots the corresponding inverter/filter inverse dynamics gain during the learning process.

Given the sensitivity of the neurocontroller learning process due to the presence of the positive feedback load current loop, use of the hybrid control scheme depicted in figure 2.6 would appear to be a logical solution to preventing system instability during the neurocontroller learning process. From the perspective of a practical implementation of this control scheme in an UPS system, the benefit obtained from the use of a neurocontroller is greatly diminished since the design of the conventional fast-response controller used in this control scheme is assumed to be based on knowledge of the output filter design and other system parameters. The adaptive intelligent control attributes of the neurocontroller are not utilized fully in this control scheme. An alternate solution to the use of the hybrid control scheme is the adjustment of the neurocontroller's learning-rate parameter. In the experimental neurocontroller, learning-rate parameter control is achieved through the manual adjustment of an external potentiometer.

Despite the fact that learning-rate parameters were reduced in simulations with less complex neurocontroller structures in order to avoid system stability problems, the initial neurocontroller learning process beginning at the simulation startup point typically required 10 to 15 modulation frequency cycles to achieve steady-state

inverter/filter/load system operation. Given a controller sampling rate at twice the inverter switching frequency and a maximum simulation inverter switching frequency of 1260 Hertz, a maximum of 630 neural network training iterations were observed between simulation startup and the steady-state neurocontroller operating point (minimum errors $\{\underline{\epsilon}_{dv}, \underline{\epsilon}_{qv}\}$). By basing the selection of neurocontroller inputs on the output filter circuit inverse dynamics, the number of required neural network training iterations are minimized (a desirable condition given practical implementation considerations). A larger number of training iterations would be expected with lower learning-rate parameter values.

2.3.1 Multi-Layer Neurocontroller Steady-State Operation

In steady-state where $\underline{\epsilon}_{dv}$ and $\underline{\epsilon}_{qv}$ are minimized, the three-phase neurocontroller forward transfer characteristics approximate the forward transfer characteristics given in the equations describing the inverter/filter circuit inverse dynamics. The use of a hidden layer with six hidden-layer nodes and hyperbolic tangent squashing functions in the neurocontroller depicted in figure 2.10 enables this neurocontroller to generate an internal nonlinear input/output mapping during its learning process. The nonlinear characteristics of this mapping correspond to the nonlinear forward transfer characteristics of the PWM inverter. From the results of the balanced passive load simulations depicted in figures 2.11 and 2.12, the neurocontroller depicted in figure 2.10 achieves the

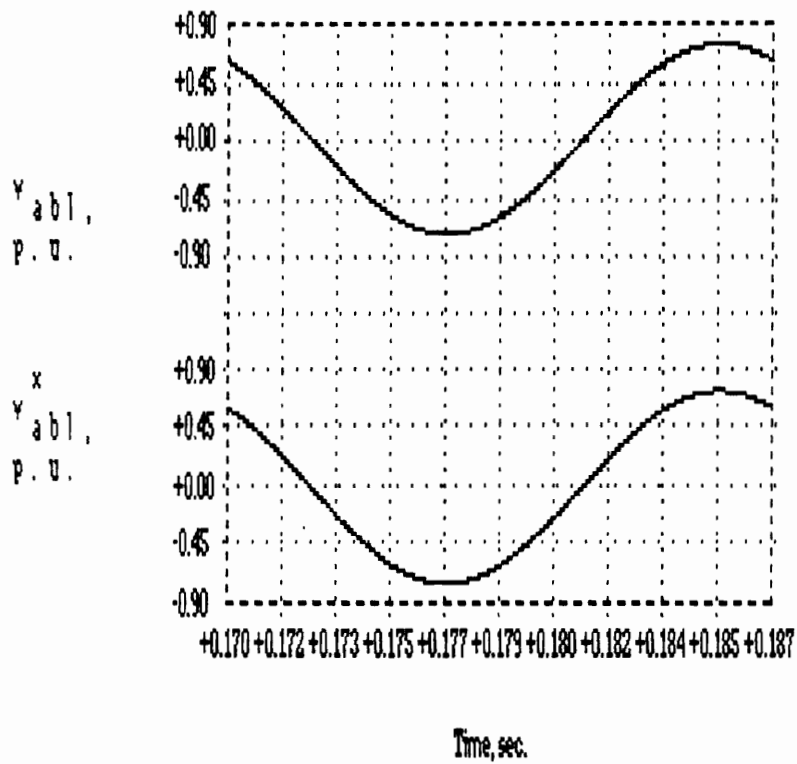


Figure 2.11: Phase A Line-to-Line Load and Corresponding Sinusoidal Reference Voltages Obtained from the Balanced Passive Load Simulation where $f_s = 1260$ Hz., $f_m = 60$ Hz., and $M = 0.75$.

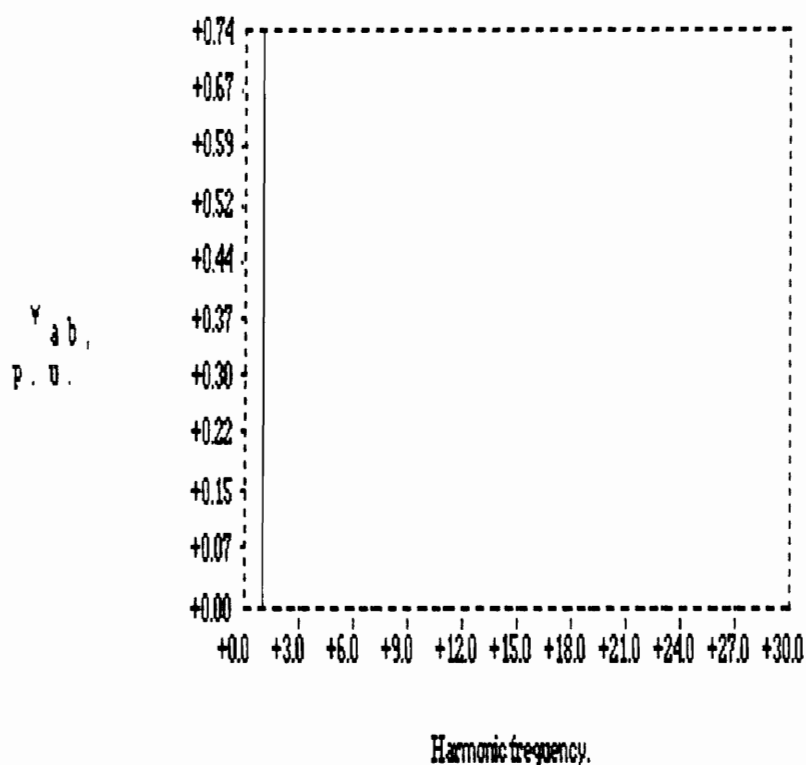


Figure 2.12: Phase A Line-to-Line Load Voltage Spectral Information, Balanced Passive Load Simulation, $f_s = 1260$ Hz., $f_m = 60$ Hz., and $M = 0.75$.

desired goal of regulating load voltage waveform amplitude and phase-displacement relative to the load voltage reference waveform minimizing the errors $\underline{\epsilon}_{dv}$ and $\underline{\epsilon}_{qv}$ without requiring prior knowledge of the output filter circuit design or system parameters such as the inverter modulation and switching frequency. With regard to balanced passive UPS loads, the concept of adaptive intelligent inverter control based on the neurocontroller configuration depicted in figure 2.10 was verified.

Using the neurocontroller depicted in figure 2.10, three-phase nonlinear load simulations were run at inverter switching frequencies $f_s = 540$ and 900 Hertz with the results depicted in figures 2.13 through 2.16. Due to observed controller instability during the learning process, the learning-rate parameter was set to a lower value (0.1 verses 1.0 in the balanced passive load simulation). The results obtained from these simulations are summarized as follows:

- i.) The load voltage fundamental voltage amplitude matches the corresponding load voltage reference waveform. The load voltage phase-displacement relative to the load voltage reference waveform is insignificant.
- ii.) Load voltage waveform harmonic distortion due to the nonlinear load is still evident.

For the purpose of comparison, additional nonlinear load simulations were run with the same system parameters (including the use of space vector PWM) using an average load voltage control technique and the three-phase version of the repetitive controller technique described

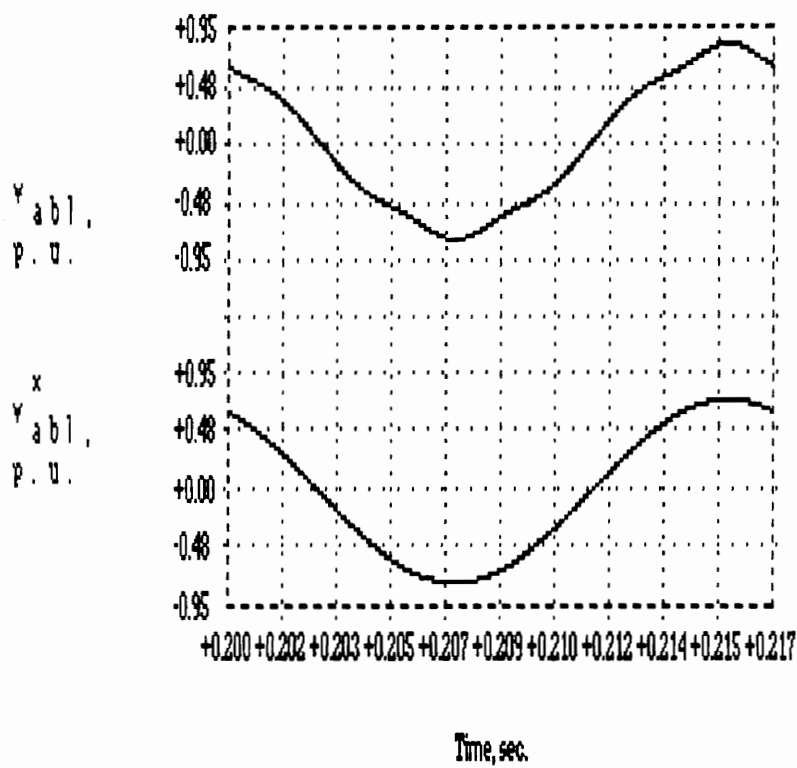


Figure 2.13: Nonlinear Load Simulation Phase A Line-to-Line Load and Reference Voltages, $f_s = 540$ Hz., $f_m = 60$ Hz., and $M = 0.75$.

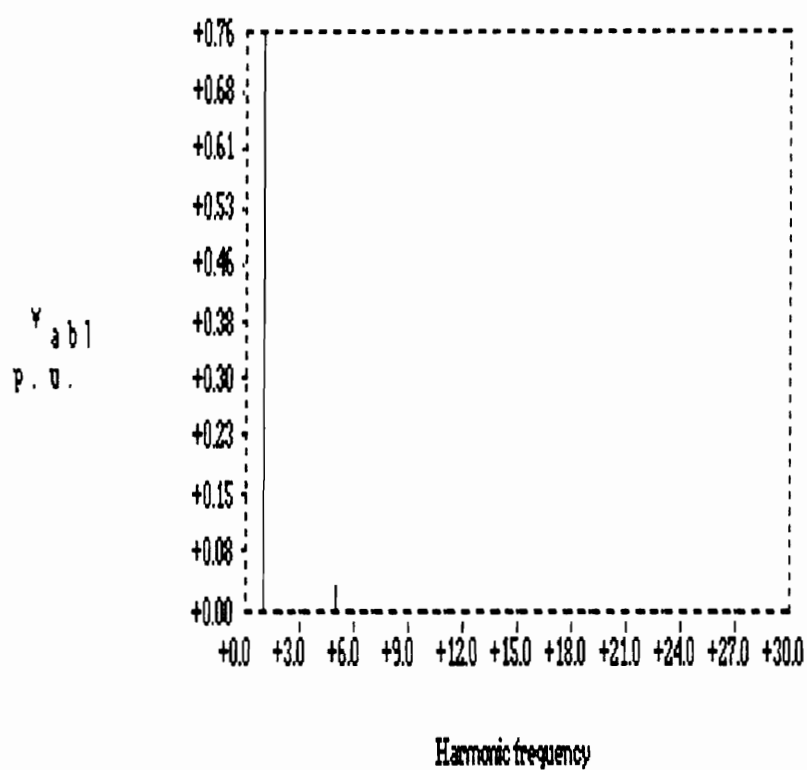


Figure 2.14: Nonlinear Load Simulation Phase A Line-to-Line Load Voltage Spectral Analysis $f_s = 540$ Hz., $f_m = 60$ Hz., and $M = 0.75$.

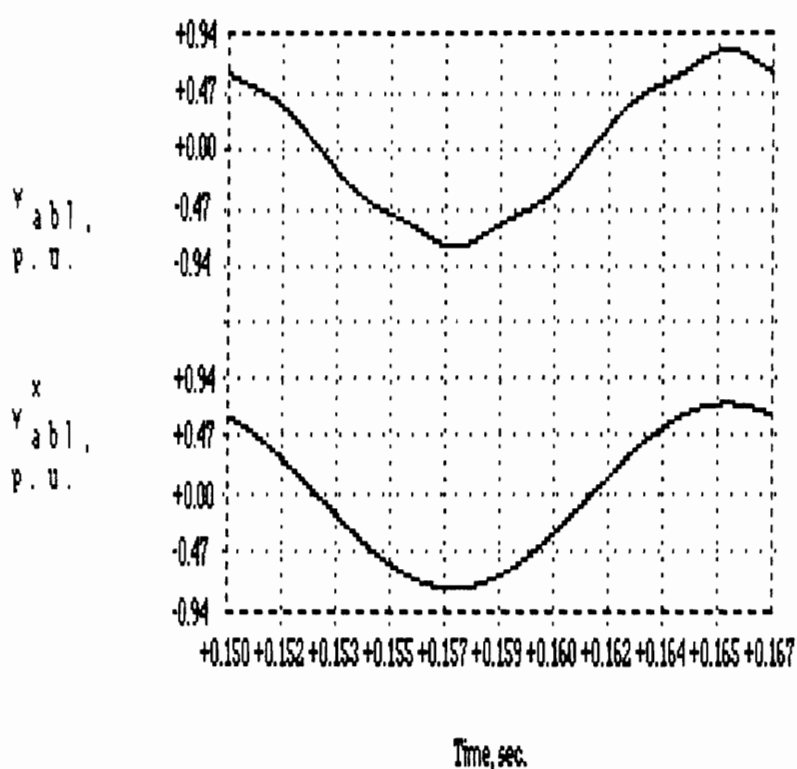


Figure 2.15: Nonlinear Load Simulation Phase A Line-to-Line Load and Reference Voltages $f_s = 900$ Hz., $f_m = 60$ Hz., and $M = 0.75$.

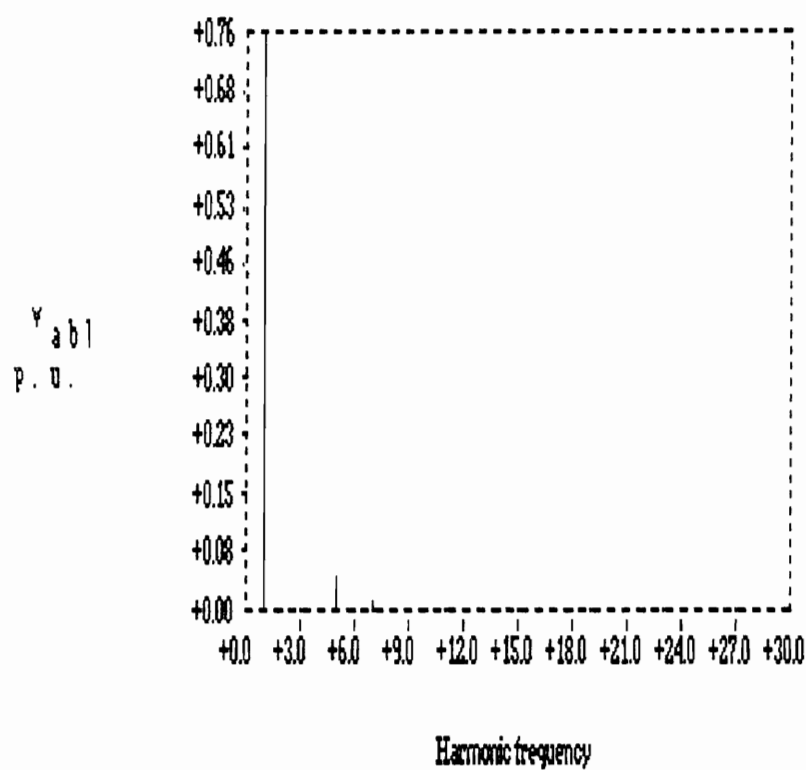


Figure 2.16: Nonlinear Load Simulation Phase A Line-to-Line Load Voltage Spectral Analysis $f_s = 900 \text{ Hz.}$, $f_m = 60 \text{ Hz.}$, and $M = 0.75$.

in reference [3]. Results of these simulations at $f_s = 540$ Hertz are depicted in figures 2.17 through 2.20. Tables 2.1 through 2.4 give load voltage and inverter output current harmonic distortion simulation data for each of the three control techniques studied. Data obtained from simulations using the average load voltage control technique serve as a benchmark in the comparison of repetitive controller and neurocontroller operation.

Both the neurocontroller and repetitive controller are designed to minimize the sampled load voltage waveform errors in steady-state. Nonlinear load simulation results corresponding to the use of these control techniques are similar. Despite having the capability of mapping nonlinear inverter/filter system characteristics particularly the nonlinear characteristics associated with the selective generation of inverter output voltage waveform harmonics, the ability of the neurocontroller with a single hidden layer and hyperbolic tangent squashing functions to suppress nonlinear load-generated load voltage waveform harmonic distortion appears to be the same as the repetitive controller which is based on linear control techniques.

Neurocontrollers with more hidden layers and nodes were not considered in later investigations given the anticipated difficulty in implementing more complex structures in an experimental controller. Additionally, the similarity in the steady-state performances of the single hidden layer neurocontroller and the repetitive controller at the relatively low inverter switching frequencies used in the simulations where $f_s = 540$ and 900 Hertz implies that the addition of hidden layers

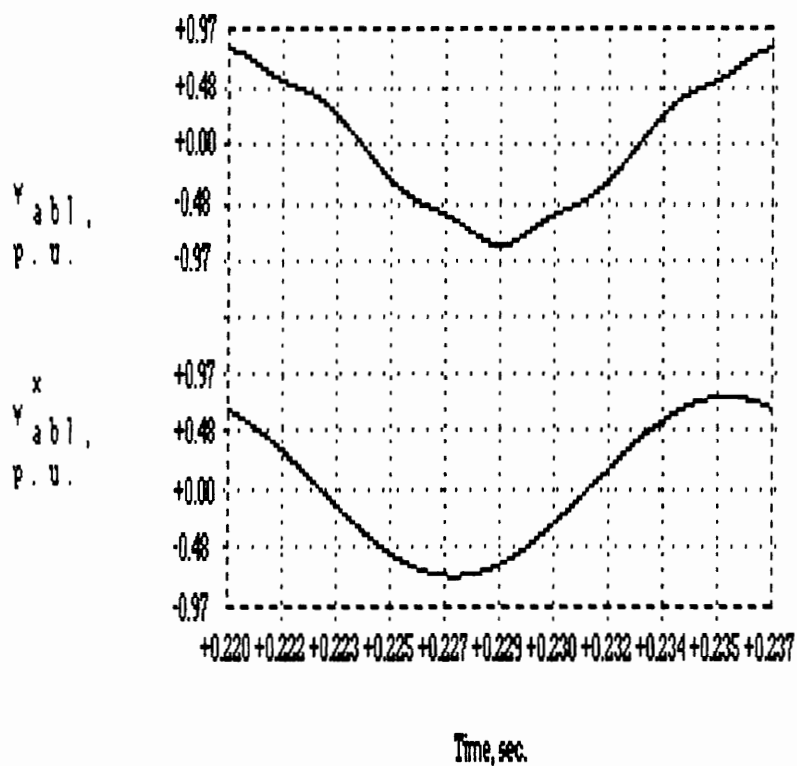


Figure 2.17: Nonlinear Load Simulation with Average Load Voltage Controller, Phase A Line-to-Line Load and Reference Voltages, $f_s = 540$ Hz, $f_m = 60$ Hz., and $M = 0.75$.

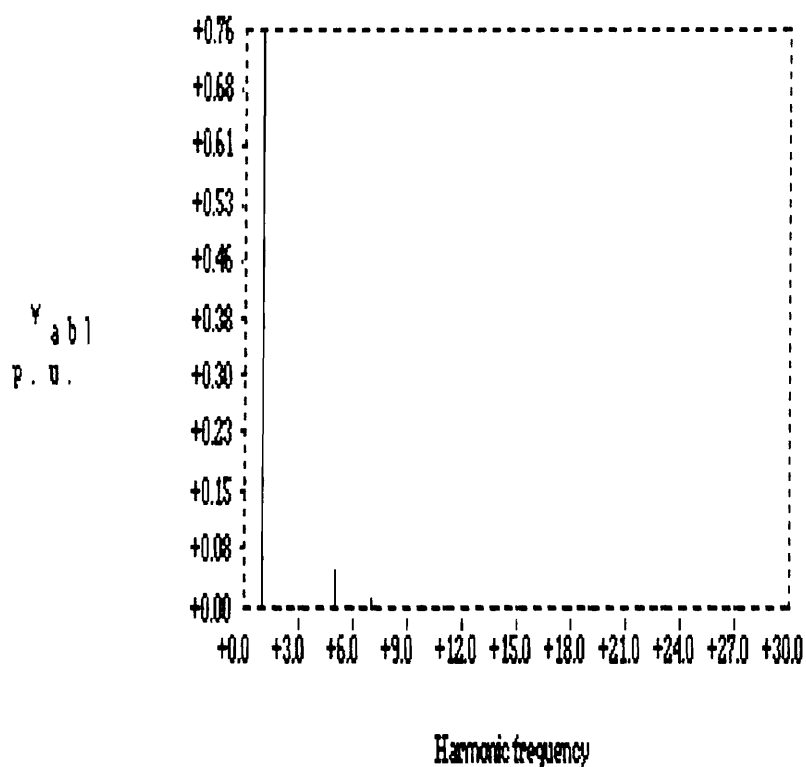


Figure 2.18: Nonlinear Load Simulation with an Average Load Voltage Controller, Phase A Line-to-Line Load Voltage Spectral Information, $f_s = 540$ Hz., $f_m = 60$ Hz., and $M = 0.75$.

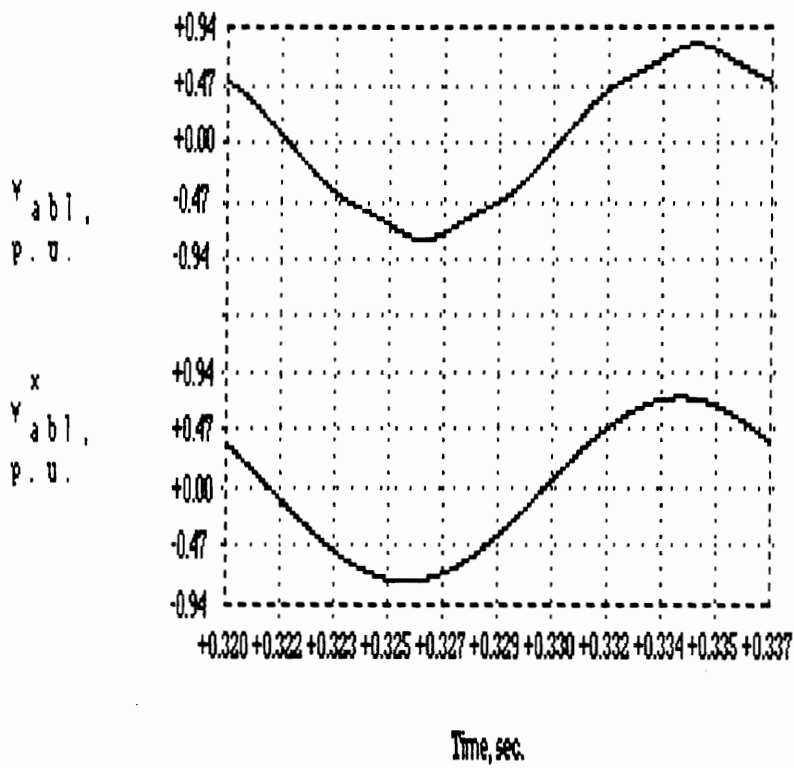


Figure 2.19: Nonlinear Load Simulation Using a Repetitive Load Voltage Controller, Phase A Line-to-Line Load and Reference Voltages, $f_s = 540$ Hz., $f_m = 60$ Hz., and $M = 0.75$.

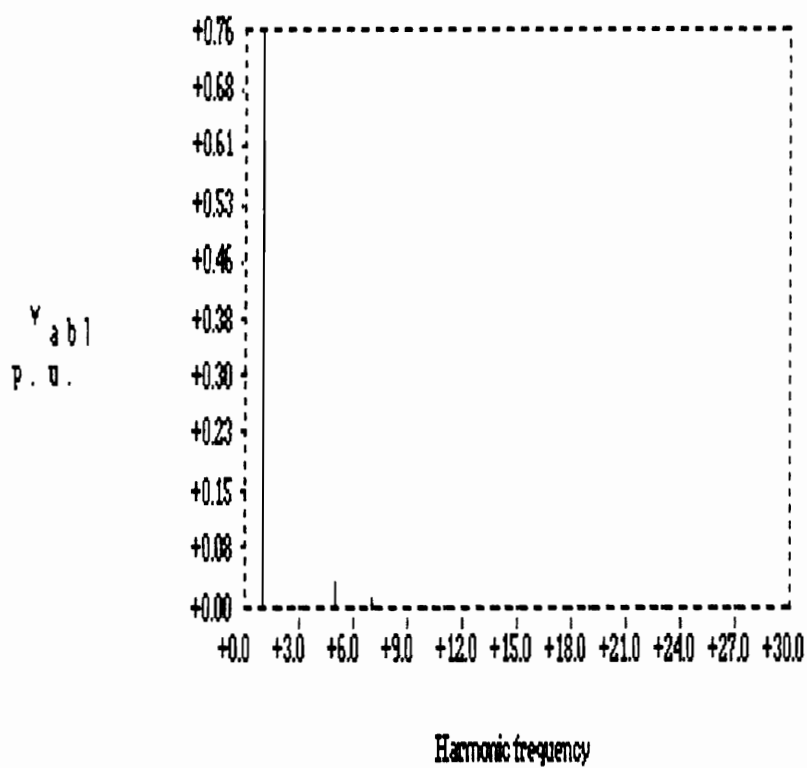


Figure 2.20: Nonlinear Load Simulation Using a Repetitive Load Voltage Controller, Phase A Line-to-Line Load Voltage Spectral Information, $f_s = 540$ Hz., $f_m = 60$ Hz., and $M = 0.75$.

Table 2.1: Comparison of Controller Techniques, Nonlinear Load. $f_s = 540$ Hz, $f_m = 60$ Hz. Load Voltage Harmonic Components Normalized with Respect to the Fundamental Frequency Component.

CONTROLLER TECHNIQUE:	NEURO-CONTROLLER	AVERAGE	REPETITIVE
HARMONICS	P.U.	P.U.	P.U.
3	0.006	0.002	0.005
5	0.044	0.060	0.033
7	0.003	0.014	0.011
9	0.001	0.000	0.001
11	0.002	0.005	0.005
13	0.002	0.003	0.003
15	0.001	0.0	0.0
17	0.006	0.005	0.005
19	0.004	0.003	0.003
21	0.0	0.0	0.0
23	0.002	0.001	0.001
25	0.002	0.001	0.001
THD	0.046	0.062	0.036

Table 2.2: Comparison of Controller Techniques, Nonlinear Load. $f_s = 540$ Hz, $f_m = 60$ Hz. Inverter Output Current Harmonic Components Normalized with Respect to the Fundamental Frequency Component.

CONTROLLER TECHNIQUE:	NEURO-CONTROLLER	AVERAGE	REPETITIVE
HARMONICS	P.U.	P.U.	P.U.
3	0.005	0.003	0.014
5	0.062	0.094	0.016
7	0.051	0.043	0.077
9	0.000	0.001	0.002
11	0.042	0.035	0.062
13	0.021	0.045	0.012
15	0.002	0.001	0.002
17	0.083	0.069	0.077
19	0.065	0.049	0.064
21	0.001	0.001	0.0
23	0.032	0.026	0.026
25	0.027	0.018	0.030
THD	0.145	0.147	0.147

Table 2.3: Comparision of Controller Techniques, Nonlinear Load. $f_s = 900$ Hz, $f_m = 60$ Hz. Load Voltage Harmonic Components Normalized with Respect to the Fundamental Frequency Component.

CONTROLLER TECHNIQUE:	NEURO-CONTROLLER	AVERAGE	REPETITIVE
HARMONICS	P.U.	P.U.	P.U.
3	0.002	0.002	0.009
5	0.046	0.048	0.051
7	0.011	0.012	0.012
9	0.0	0.0	0.001
11	0.003	0.004	0.005
13	0.004	0.0	0.005
15	0.0	0.002	0.0
17	0.003	0.002	0.002
19	0.001	0.001	0.001
21	0.0	0.0	0.0
23	0.0	0.0	0.001
25	0.001	0.0	0.0
THD	0.048	0.05	0.054

Table 2.4: Comparison of Controller Techniques, Nonlinear Load. $f_s = 900$ Hz, $f_m = 60$ Hz. Inverter Output Current Harmonic Components Normalized with Respect to the Fundamental Frequency Component.

CONTROLLER TECHNIQUE:	NEURO-CONTROLLER	AVERAGE	REPETITIVE
HARMONICS	P.U.	P.U.	P.U.
3	0.001	0.010	0.004
5	0.039	0.045	0.056
7	0.004	0.011	0.013
9	0.001	0.002	0.001
11	0.024	0.027	0.023
13	0.030	0.030	0.033
15	0.0	0.001	0.001
17	0.028	0.027	0.031
19	0.020	0.022	0.021
21	0.001	0.001	0.001
23	0.005	0.009	0.008
25	0.008	0.010	0.006
THD	0.065	0.073	0.08

and nodes would not result in an improvement in the neurocontroller's ability to suppress nonlinear load-generated load voltage waveform harmonic distortion.

2.3.2 Single-Layer Neurocontrollers

The consideration of a single-layer neurocontroller was prompted by the difficulty in implementing the neurocontroller structure depicted in figure 2.10 in an experimental digital controller. There are numerous approaches to implementing the neurocontroller concept in hardware. A design involving analog circuitry was dismissed given the relative difficulty and expense of implementing design in this fashion. The parallel structure of the neural network suggests the use of individual processors for each node in the hidden and output layers. The functions of each node including the squashing function are basic, and so, a node processor design could be limited to an application-specific integrated circuit. For reasons which are not relevant to the topic of this thesis, a controller design utilizing serial microcontroller technology was executed.

Experimental controller software written in assembly language utilized equations (2.16) through (2.26) in the controller routines for the forward propagation and conventional backpropagation learning processes. In each controller sampling interval $1/2f_s$ in duration, both of these processes are to be executed along with the space vector PWM process. All controller math operations were performed in fixed-point

16-bit signed arithmetic. Hyperbolic secant and tangent function values were pre-computed and stored in look-up tables located in the controller Read-Only Memory (ROM).

The initial attempt to implement the neurocontroller structure depicted in figure 2.10 in an experimental controller resulted in controller execution times or combined forward propagation, backpropagation, and space vector PWM processes which greatly exceeded the controller sampling intervals corresponding to inverter switching frequencies $f_s = 540, 900, \text{ and } 1260$ Hertz. A neurocontroller structure using four hidden layer nodes was also implemented in an experimental controller, but the combined execution times were again outside the desired range dictated by the inverter switching frequencies.

Experimental neurocontroller software was written for a single-layer neurocontroller. The forward propagation process is described by the equations

$$\begin{aligned} \underline{v}_{di}^*(k) = & w_{11}(k) \cdot \underline{v}_{d1}^*(k) + w_{12}(k) \cdot \underline{v}_{q1}^*(k) + \\ & w_{13}(k) \cdot [\underline{i}_{d1}^{(q)}(k) - \underline{i}_{d1}^{(q)}(k-1)] + \\ & w_{14}(k) \cdot [\underline{i}_{q1}^{(q)}(k) - \underline{i}_{q1}^{(q)}(k-1)] \text{ and} \end{aligned} \quad (2.30)$$

$$\begin{aligned} \underline{v}_{qi}^*(k) = & w_{21}(k) \cdot \underline{v}_{d1}^*(k) + w_{22}(k) \cdot \underline{v}_{q1}^*(k) + \\ & w_{23}(k) \cdot [\underline{i}_{d1}^{(q)}(k) - \underline{i}_{d1}^{(q)}(k-1)] + \\ & w_{24}(k) \cdot [\underline{i}_{q1}^{(q)}(k) - \underline{i}_{q1}^{(q)}(k-1)], \end{aligned} \quad (2.31)$$

where $w_{ij}(k)$ are the network weights in the k^{th} controller sampling interval, $\{\underline{v}_{d1}^*(k), \underline{v}_{q1}^*(k)\}$ are the logic-level load voltage commands in the d-q frame, $\{\underline{i}_{d1}^{(q)}(k), \underline{i}_{q1}^{(q)}(k)\}$ are the logic-level load current feedback signals (90° phase shifted) in the d-q frame, and

$\{\underline{v}_{di}^*(k), \underline{v}_{qi}^*(k)\}$ are the logic-level inverter modulation commands in the d-q frame. The conventional backpropagation learning process is described by the equations

$$w_{11}(k + 1) = w_{11}(k) + c(k) \cdot \underline{\epsilon}_{dv}(k) \cdot \underline{v}_{d1}^*(k - 2) \quad (2.32)$$

$$w_{12}(k + 1) = w_{12}(k) + c(k) \cdot \underline{\epsilon}_{dv}(k) \cdot \underline{v}_{q1}^*(k - 2) \quad (2.33)$$

$$w_{13}(k + 1) = w_{13}(k) + c(k) \cdot \underline{\epsilon}_{dv}(k) \cdot [\underline{i}_{d1}^{(q)}(k - 2) - \underline{i}_{d1}^{(q)}(k - 3)] \quad (2.34)$$

$$w_{14}(k + 1) = w_{14}(k) + c(k) \cdot \underline{\epsilon}_{dv}(k) \cdot [\underline{i}_{q1}^{(q)}(k - 2) - \underline{i}_{q1}^{(q)}(k - 3)] \quad (2.35)$$

$$w_{21}(k + 1) = w_{21}(k) + c(k) \cdot \underline{\epsilon}_{qv}(k) \cdot \underline{v}_{d1}^*(k - 2) \quad (2.36)$$

$$w_{22}(k + 1) = w_{22}(k) + c(k) \cdot \underline{\epsilon}_{qv}(k) \cdot \underline{v}_{q1}^*(k - 2) \quad (2.37)$$

$$w_{23}(k + 1) = w_{23}(k) + c(k) \cdot \underline{\epsilon}_{qv}(k) \cdot [\underline{i}_{d1}^{(q)}(k - 2) - \underline{i}_{d1}^{(q)}(k - 3)] \quad (2.38)$$

and

$$w_{24}(k + 1) = w_{24}(k) + c(k) \cdot \underline{\epsilon}_{qv}(k) \cdot [\underline{i}_{q1}^{(q)}(k - 2) - \underline{i}_{q1}^{(q)}(k - 3)] \quad (2.39)$$

where

$$\underline{\epsilon}_{dv}(k) = \underline{v}_{d1}^*(k) - \underline{v}_{d1}^{(q)}(k), \quad (2.40)$$

$$\underline{\epsilon}_{qv}(k) = \underline{v}_{q1}^*(k) - \underline{v}_{q1}^{(q)}(k), \quad (2.41)$$

$\{\underline{v}_{d1}^{(q)}, \underline{v}_{q1}^{(q)}\}$ are the logic-level load voltage feedback 90° phase shifted in the d-q frame, and $c(k)$ is the learning-rate parameter value in the k^{th} controller sampling interval. Experimental controller software written using equations (2.30) through (2.41) had combined execution times which enabled the use of a maximum inverter switching frequency $f_s = 750$ Hertz. Subsequent computer simulation and

experimental investigations of the neurocontroller concept were made using the single-layer neurocontroller structure with this maximum inverter switching frequency and an inverter modulation frequency, f_m , of 50 Hertz.

By eliminating the neurocontroller hidden layer, the neurocontroller loses its capability of generating a fixed internal mapping of nonlinear inverter/filter system characteristics. Comparisons of the results from computer simulations and experiments using the single-layer neurocontroller with the simulation results associated with the multi-layer neurocontroller depicted in figure 2.10 have revealed that the presence of the hidden layer influences the neurocontroller's dynamic performance with regard to controller stability during the controller learning process and steady-state load voltage error characteristics.

Experiments and parallel computer simulations using the single-layer neurocontroller with the conventional backpropagation learning process utilized were run using balanced passive, unbalanced passive, and nonlinear load models. Experiments were conducted with and without the learning process enabled where load voltage error measurements without the learning process enabled served as a basis for comparing neurocontroller performance with the learning process enabled. Single-layer neurocontroller weights were initialized with $w_{11} = w_{22} = 1$, and the remainder of the network weights set to zero. Single-layer neurocontroller instability during the learning process was avoided by setting the learning-rate parameter to a relatively low value of 0.066.

In steady-state, the performance of the single-layer neurocontroller using the conventional backpropagation learning process is inferior to the performance of the multi-layer neurocontroller. This comparison is illustrated in the load voltage spectral information obtained from the balanced passive load simulation depicted in figure 2.21. A non-trivial steady-state fundamental load voltage amplitude error is generated, 0.79 verses the 0.75 command value. Additionally, significant low-order harmonics not typical with a balanced passive load are present in the load voltage waveform. Figure 2.21 indicates the presence of second and third harmonics in the load voltage waveform. Load voltage harmonic distortion at these frequencies is created by asymmetrical three-phase inverter output waveform generation and neurocontroller limit cycle operation. The frequency of the limit cycle oscillation is determined by the output filter pole frequency. The output filter design in the single-layer neurocontroller simulations and experiments utilizes the filter component values $L_f = 10 \text{ mH}$ and $C_f = 45 \text{ }\mu\text{F}$ which places the filter pole frequency at 137 Hertz or between the second and third harmonics.

The problem associated with steady-state single-layer neurocontroller operation using the conventional backpropagation learning process is created when either $d\underline{\underline{\epsilon}}_{dv}/dt = 0$ and $\underline{\underline{\epsilon}}_{dv}$ is not zero, or $d\underline{\underline{\epsilon}}_{qv}/dt = 0$ and $\underline{\underline{\epsilon}}_{qv}$ is not zero. When $d\underline{\underline{\epsilon}}_{dv}/dt = 0$ and $|\underline{\underline{\epsilon}}_{dv}| > 0$, the single-layer neurocontroller d-axis forward transfer characteristics defined in equation (2.30) do not change (or the learning process terminates). When $d\underline{\underline{\epsilon}}_{qv}/dt = 0$ and $|\underline{\underline{\epsilon}}_{qv}| > 0$, the single-layer

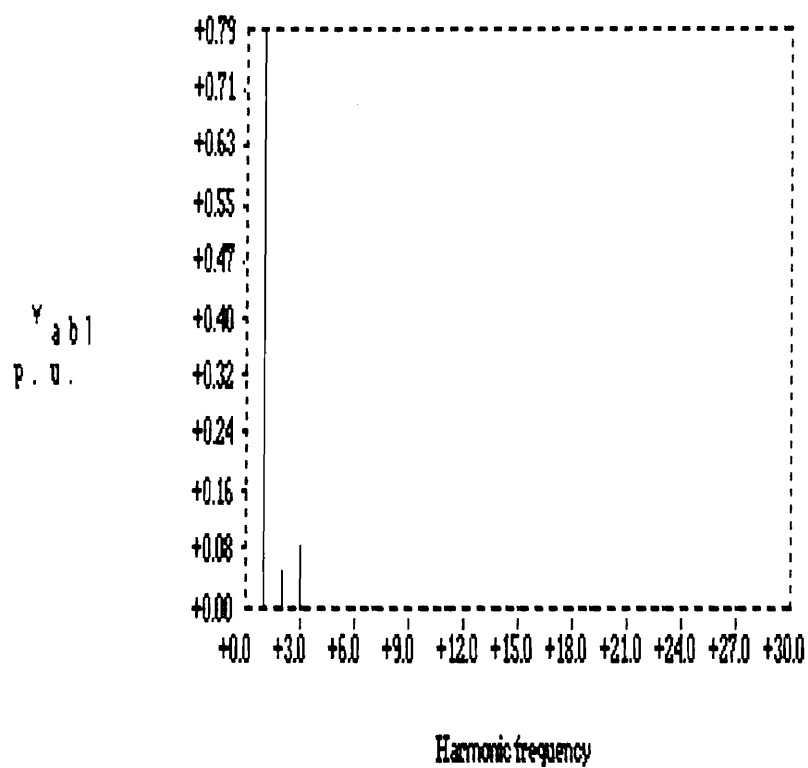


Figure 2.21: Phase A Line-to-Line Load Voltage Spectral Information Obtained from the Balanced Resistive Load Simulation, Conventional Backpropagation Learning Process Enabled.

neurocontroller q-axis forward transfer characteristics defined in equation (2.31) do not change. Limit cycle oscillation occurs as a result of one learning process terminating while the other learning process continues. When the steady-state load voltage errors are not zero, the conditions

$$d\underline{\epsilon}_{dv}/dt = 0 \quad (2.42)$$

and

$$d\underline{\epsilon}_{qv}/dt = 0 \quad (2.43)$$

can be realized in this control problem. The error functions are linear combinations of sine and cosine functions. Solutions to equations (2.42) and (2.43) different from the equations

$$\underline{\epsilon}_{dv} = 0 \quad (2.44)$$

and

$$\underline{\epsilon}_{qv} = 0 \quad (2.45)$$

are possible.

The fact that this problem associated with the single-layer neurocontroller is not evident in the multi-layer neurocontroller simulations is attributed to the fact that d- and q-axis multi-layer neurocontroller forward transfer characteristics described in equations (2.16) through (2.18) are not independent as in the case of the single-layer neurocontroller. The modification of connection weights $w_{iu}^{(1)}$ in the multi-layer neurocontroller modifies both the d- and q-axis forward transfer characteristics and the limit cycle operation evident in the single-layer neurocontroller is avoided. Steady-state controller instability may be created in the multi-layer neurocontroller,

particularly in the nonlinear load simulations, by increasing the learning-rate parameter value.

The conventional backpropagation learning process is known to converge towards a local minimum as in the case of solutions for (2.42) and (2.43) given corresponding non-zero errors. This event which would appear as a set of d- and q-axis steady-state fundamental load voltage errors was not evident in the multi-layer neurocontroller simulations. Given the load voltage commands in the form

$$\underline{v}_{d1}^*(t) = M \cdot \sin(2\pi f_m t) \text{ and} \quad (2.46)$$

$$\underline{v}_{q1}^*(t) = -M \cdot \cos(2\pi f_m t) \quad (2.47)$$

where M is the modulation index and f_m is the inverter's modulation frequency, simultaneous solutions to (2.42) and (2.43) occur only when $\underline{\epsilon}_{dv} = \underline{\epsilon}_{qv} = 0$. The coupling between the d- and q-axis forward transfer characteristics prevents the multi-layer neurocontroller from converging to a load voltage trajectory other than the desired zero error trajectory.

2.3.3 Modified Backpropagation Learning Process

The limit cycle generation problem in the single-layer neurocontroller operation was eliminated by modifying the backpropagation learning process. The new neurocontroller learning strategy is constructed on the principle that neurocontroller weight changes are made based on logical decisions involving the load voltage errors, the current network weights values, and the product term values

in equations (2.30) and (2.31). The following pseudo-software routine segment describes the w_{11} weight modification process using this new learning strategy given below:

```

if ( $\underline{\underline{\epsilon}}_{dv}(k) \geq 0$ )
    then
        if [ $w_{11}(k) \cdot \underline{\underline{v}}_{d1}^*(k - 2) \geq 0$ ]
            then
                 $w_{11}(k + 1) = w_{11}(k) + |\underline{\underline{\epsilon}}_{dv}(k) \cdot \underline{\underline{v}}_{d1}^*(k - 2)| \cdot$ 
                     $\text{sgn}[w_{11}(k)] \cdot c(k)$ 
            else
                 $w_{11}(k + 1) = w_{11}(k) - |\underline{\underline{\epsilon}}_{dv}(k) \cdot \underline{\underline{v}}_{d1}^*(k - 2)| \cdot$ 
                     $\text{sgn}[w_{11}(k)] \cdot c(k)$ 
        else if ( $\underline{\underline{\epsilon}}_{dv}(k) < 0$ )
            then
                if [ $w_{11}(k) \cdot \underline{\underline{v}}_{d1}^*(k - 2) \geq 0$ ]
                    then
                         $w_{11}(k + 1) = w_{11}(k) - |\underline{\underline{\epsilon}}_{dv}(k) \cdot \underline{\underline{v}}_{d1}^*(k - 2)| \cdot$ 
                             $\text{sgn}[w_{11}(k)] \cdot c(k)$ 
                    else
                         $w_{11}(k + 1) = w_{11}(k) + |\underline{\underline{\epsilon}}_{dv}(k) \cdot \underline{\underline{v}}_{d1}^*(k - 2)| \cdot$ 
                             $\text{sgn}[w_{11}(k)] \cdot c(k)$ 

```

The function $\text{sgn}(a)$ is defined as

$\text{sgn}(a) = 1$ for $a \geq 0$ and

$$\text{sgn}(a) = -1 \text{ for } a \leq 0.$$

The modification process for the remaining connection weights is similar. The load voltage error and neurocontroller input values used in the modification process for a specific neurocontroller weight is defined in the single-layer neurocontroller forward transfer equations (2.30) and (2.31). Using the software segment given previously as a template, the modified backpropagation algorithm for the other neurocontroller weights is derived by the appropriate substitution of weight terms, neurocontroller inputs, and the load voltage error term.

Using the conventional backpropagation learning process, the single-layer neurocontroller forward transfer characteristics are modified during the k^{th} controller sampling interval in a fashion described by the following equations:

$$\begin{aligned} \underline{v}_{di}^*(k) = & w_{11}(k-1) \cdot \underline{v}_{dl}^*(k) + \\ & w_{12}(k-1) \cdot \underline{v}_{ql}^*(k) + \\ & w_{13}(k-1) \cdot [\underline{i}_{dl}^{(q)}(k) - \underline{i}_{dl}^{(q)}(k-1)] + \\ & w_{14}(k-1) \cdot [\underline{i}_{ql}^{(q)}(k) - \underline{i}_{ql}^{(q)}(k-1)] + \\ & c(k-1) \cdot \underline{\epsilon}_{dv}(k-1) \cdot \left[\underline{v}_{dl}^*(k-3) \cdot \underline{v}_{dl}^*(k) + \right. \\ & \quad \underline{v}_{ql}^*(k-3) \cdot \underline{v}_{ql}^*(k) + \\ & \quad [\underline{i}_{dl}^{(q)}(k-3) - \underline{i}_{dl}^{(q)}(k-4)] \cdot \\ & \quad [\underline{i}_{dl}^{(q)}(k) - \underline{i}_{dl}^{(q)}(k-1)] + \\ & \quad [\underline{i}_{ql}^{(q)}(k-3) - \underline{i}_{ql}^{(q)}(k-4)] \cdot \\ & \quad \left. [\underline{i}_{ql}^{(q)}(k) - \underline{i}_{ql}^{(q)}(k-1)] \right] \end{aligned}$$

$$\left. \vphantom{\frac{1}{2}} \right] \quad (2.48)$$

and

$$\begin{aligned} \underline{v}_{qi}^*(k) = & w_{21}(k-1) \cdot \underline{v}_{d1}^*(k) + \\ & w_{22}(k-1) \cdot \underline{v}_{q1}^*(k) + \\ & w_{23}(k-1) \cdot [\underline{i}_{d1}(q)(k) - \underline{i}_{d1}(q)(k-1)] + \\ & w_{24}(k-1) \cdot [\underline{i}_{q1}(q)(k) - \underline{i}_{q1}(q)(k-1)] + \\ & c(k-1) \cdot \underline{\epsilon}_{qv}(k-1) \cdot \left[\underline{v}_{d1}^*(k-3) \cdot \underline{v}_{d1}^*(k) + \right. \\ & \quad \underline{v}_{q1}^*(k-3) \cdot \underline{v}_{q1}^*(k) + \\ & \quad [\underline{i}_{d1}(q)(k-3) - \underline{i}_{d1}(q)(k-4)] \cdot \\ & \quad [\underline{i}_{d1}(q)(k) - \underline{i}_{d1}(q)(k-1)] + \\ & \quad [\underline{i}_{q1}(q)(k-3) - \underline{i}_{q1}(q)(k-4)] \cdot \\ & \quad \left. [\underline{i}_{q1}(q)(k) - \underline{i}_{q1}(q)(k-1)] \right] \end{aligned} \quad (2.49)$$

Equations (2.48) and (2.49) were derived by substituting equations (2.32) through (2.39) into equations (2.30) and (2.31).

Given load voltage command signals defined by the equations (2.46) and (2.47), the term $[\underline{v}_{d1}^*(k-3) \cdot \underline{v}_{d1}^*(k) + \underline{v}_{q1}^*(k-3) \cdot \underline{v}_{q1}^*(k)]$ is a constant. Load current d-q frame feedback signals $\{\underline{i}_{d1}(q), \underline{i}_{q1}(q)\}$ generated from the three-phase to d-q frame transformation defined in equation (2.11) are AC signals. The inverter modulation command trajectory $\{\underline{v}_{di}^*(k), \underline{v}_{qi}^*(k)\}$ resulting from the single-layer neurocontroller operation using the conventional backpropagation process

is separated into two components; trajectory generation due to load voltage command inputs and load current positive feedback.

At no-load $\{i_{d1}(q) = 0, i_{q1}(q) = 0\}$, changes in the inverter modulation command trajectory $\{v_{di}^*(k), v_{qi}^*(k)\}$ are dependent on the product terms

$$c(k-1) \cdot \epsilon_{dv}(k-1) \cdot [v_{d1}^*(k-3) \cdot v_{d1}^*(k) + v_{q1}^*(k-3) \cdot v_{q1}^*(k)]$$

and

$$c(k-1) \cdot \epsilon_{qv}(k-1) \cdot [v_{d1}^*(k-3) \cdot v_{d1}^*(k) + v_{q1}^*(k-3) \cdot v_{q1}^*(k)].$$

Neurocontroller instability in the form of limit cycle oscillation is not likely (depending on the learning-rate parameter value) since only the controller negative feedback loop is in operation. Additionally, the event where the backpropagation process converges to a local minimum or when either

$$d\epsilon_{dv}/dt = 0 \text{ and } |\epsilon_{dv}| > 0 \text{ or}$$

$$d\epsilon_{qv}/dt = 0 \text{ and } |\epsilon_{qv}| > 0$$

is not likely since the term

$$v_{d1}^*(k-3) \cdot v_{d1}^*(k) + v_{q1}^*(k-3) \cdot v_{q1}^*(k)$$

is non-zero for non-zero modulation index values.

The observed limit cycle operation of the single-layer neurocontroller using the conventional backpropagation learning process is attributed to the load current positive feedback. In the balanced passive load simulation, second and third harmonic load voltage distortion creates corresponding load current distortion. Three-phase PWM inverters are not capable of selectively generating inverter output waveform harmonics other than harmonics in the order of $6 \cdot p \pm 1$ (p is

positive integer greater than one) in symmetrical three-phase inverter output voltage waveforms. Due to three-phase inverter switching limitations, load current harmonic positive feedback at harmonics other than those in the order of $6 \cdot p \pm 1$ does not compensate for the load voltage waveform harmonic distortion, and promotes the generation of the limit cycle oscillation observed in the single-layer neurocontroller balanced passive load simulations.

The modified backpropagation learning process changes the trajectory $\{\underline{v}_{di}^*(k), \underline{v}_{qi}^*(k)\}$ in the direction suggested by the load voltage error vector $\{\underline{\epsilon}_{dv}(k), \underline{\epsilon}_{qv}(k)\}$. For example, changes in the d-axis modulation command signals due to changes via connection weight modification in the corresponding single-layer neurocontroller forward transfer characteristics

$$\begin{aligned} \underline{v}_{di}^*(k) = & w_{11}(k) \cdot \underline{v}_{d1}^*(k) + w_{12}(k) \cdot \underline{v}_{q1}^*(k) + \\ & w_{13}(k) \cdot [\underline{i}_{d1}^{(q)}(k) - \underline{i}_{d1}^{(q)}(k-1)] + \\ & w_{14}(k) \cdot [\underline{i}_{q1}^{(q)}(k) - \underline{i}_{q1}^{(q)}(k-1)] \end{aligned} \quad (2.30)$$

are made such that changes in each product term of (2.30) have the same polarity as the load voltage error $\underline{\epsilon}_{dv}(k)$. Connection weight modification in this fashion forces a change in the d-axis component $\underline{v}_{di}^*(k)$ in the direction corresponding to the polarity of $\underline{\epsilon}_{dv}(k)$. With the use of the convention backpropagation method, a change in either $\underline{v}_{di}^*(k)$ or $\underline{v}_{qi}^*(k)$ is not limited in this fashion and can have a polarity opposing the direction suggested by the corresponding error $\underline{\epsilon}_{dv}(k)$ or $\underline{\epsilon}_{qv}(k)$.

In the modified backpropagation algorithm, modification of each neurocontroller connection weight is based on logical decisions or polarity tests of the corresponding load voltage error and the product term in which the weight is a component. Based on the result of the polarity tests, weight changes are made in a fashion such that the product term associated with the connection weight changes in value in the direction suggested by the corresponding load voltage error. The connection weight change increment is the absolute value of the increment used in the conventional backpropagation algorithm.

Using the modified backpropagation algorithm in the single-layer neurocontroller, the balanced passive load simulation was again run with results of this simulation depicted in figure 2.22 (load voltage waveform spectral information). The steady-state fundamental load voltage error is eliminated. Second harmonic generation (limit cycle oscillation) is eliminated as well. The subsequent single-layer neurocontroller, network depicted in figure 2.23, simulations and experiments were performed using the modified backpropagation learning process.

2.4 Experimental Neurocontroller Design

Figure 2.24 depicts a block diagram of the neurocontroller circuitry. The functions of the neurocontroller circuitry are enumerated as follows:

- i.) Space vector PWM process; modulation commands $\{\underline{v}_{di}^*(k), \underline{v}_{qi}^*(k)\}$

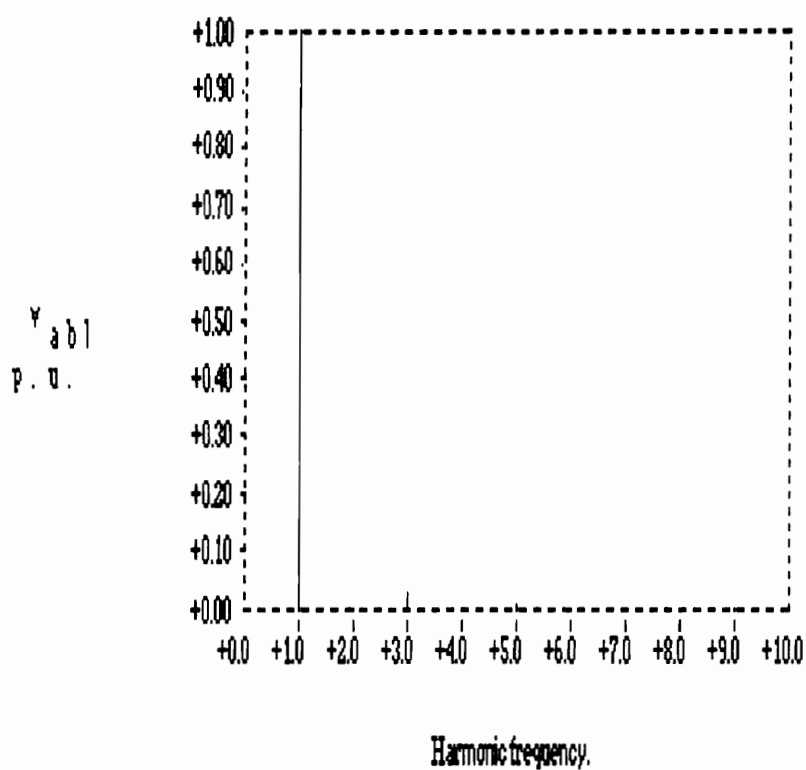


Figure 2.22: Q-Axis Load Voltage Spectral Information, Balanced Resistive Load Simulation, Modified Backpropagation Learning Process Enabled.

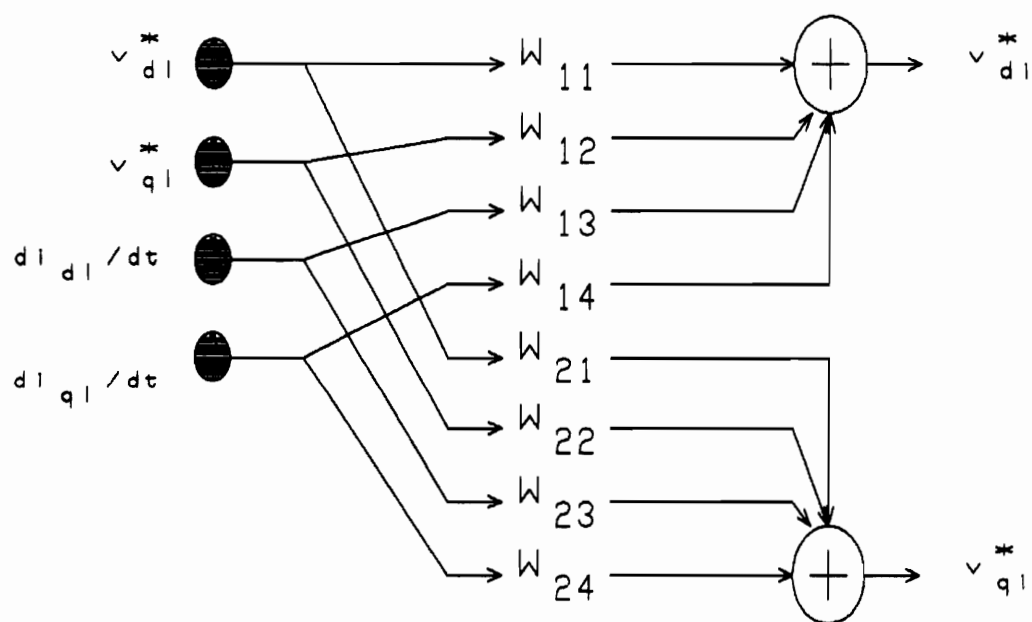


FIGURE 2.23: FINAL NETWORK VERSION

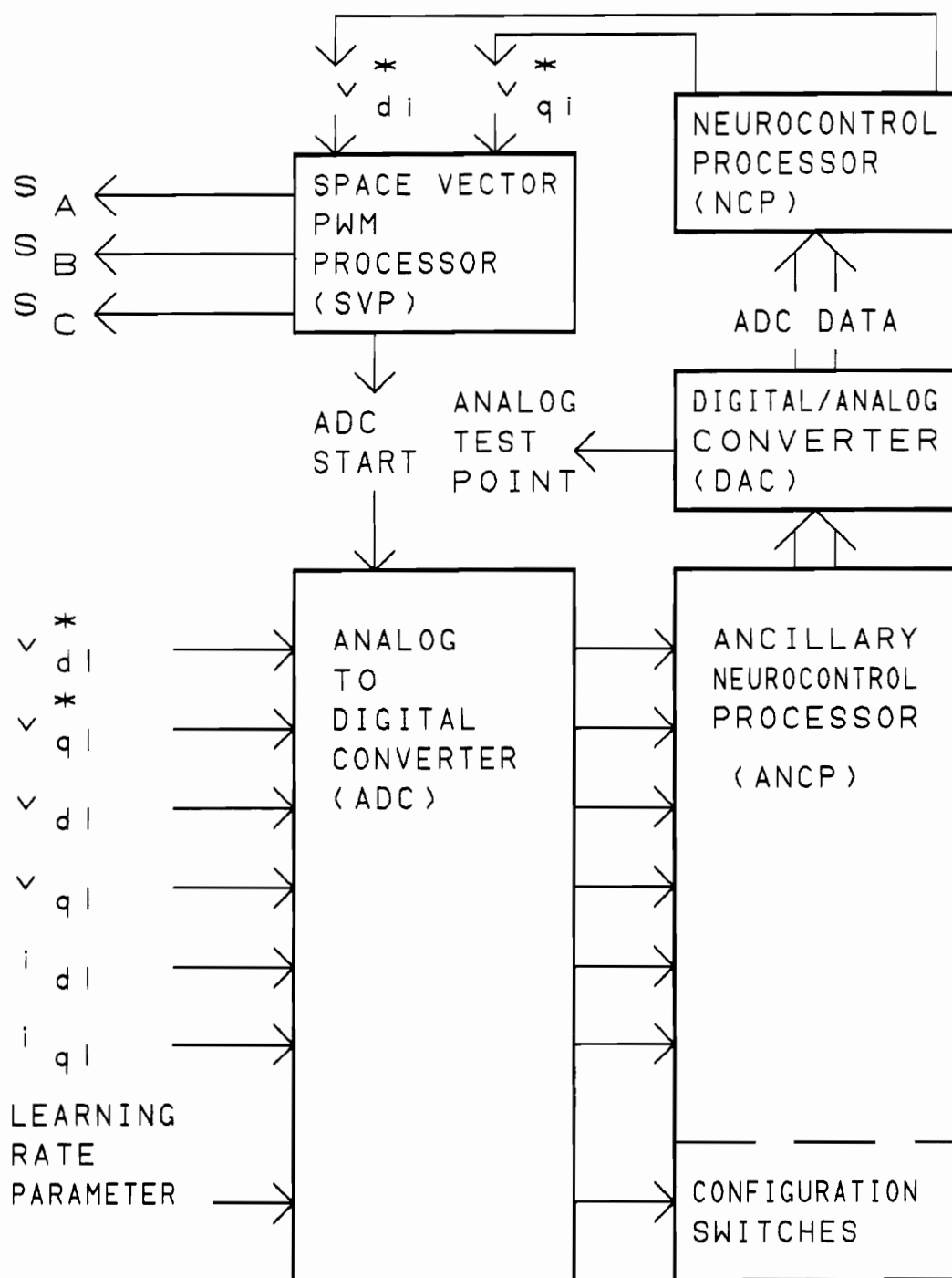


FIGURE 2.24: NEUROCONTROLLER CIRCUIT BLOCK DIAGRAM

are sampled at $1/2f_s$ intervals. The sequence of inverter switching states and corresponding state times are computed. This information is stored in a queue for execution during the proceeding PWM sampling interval. Independent of other controller functions (i.e. an interrupt routine), inverter switching states are output to the inverter switching control logic via an optical fiber interface in the form of three inverter leg circuit control signals (binary signals). All other controller functions are synchronized to the inverter switching signals generation process.

At the beginning of a PWM sample interval, the process of sampling and analog-to-digital conversion of controller input data is triggered. This conversion process when completed, triggers the neural network forward propagation and learning processes. At the termination of the forward propagation process, modulation commands are passed to the space vector PWM process and the computation of inverter switching states and state times is initiated. The microcontroller designated as SVP performs all Space vector PWM process functions and the generation of the start signal for the analog-to-digital conversion process.

- ii.) In order to minimize controller execution time, the seven channels of analog data input to the controller are converted simultaneously. The analog-to-digital conversion (ADC) circuit operates in a stand-alone mode triggered into operation by a conversion start signal generated by the SVP at the beginning of

the PWM sample interval. The seven analog signals input to the controller are:

- a.) $\underline{v}_{dl}^*(k)$ and $\underline{v}_{ql}^*(k)$
- b.) $\underline{i}_{dl}^{(q)}(k)$ and $\underline{i}_{ql}^{(q)}(k)$
- c.) $\underline{v}_{dl}^{(q)}(k)$ and $\underline{v}_{ql}^{(q)}(k)$
- d.) learning-rate parameter.

Three-phase load current and voltage feedback are converted to d-q frame currents and voltages via analog circuitry within the controller circuitry. The learning-rate parameter is an external potentiometer adjustment. The load voltage commands are generated by a sinusoidal load voltage generator within the controller circuitry. An external potentiometer adjustment controls the amplitudes of these reference signals or the potentiometer controls the modulation index value M . When the conversion process is completed (27 μ s conversion time), the ADC circuitry emits an interrupt signal to the processors involved with the execution of the neural network functions.

- iii.) The processor designated as ANCP inputs converted analog data and passes it directly to the processor designated as NCP. The NCP processor executes the forward propagation and learning processes. Converted analog data input by the ANCP processor is passed to the NCP processor via an eight-bit parallel port connecting both processors. The ANCP also inputs configuration switch data and outputs selected converted analog input data to a digital-to-analog converter (DAC) which in turn generates an analog signal

that can be monitored externally. Configuration switch data selects the converted analog data send to the DAC and either enables or disables the controller's learning process.

- iv.) At the termination of the forward propagation process executed in the NCP, the NCP writes the modulation commands to a 16-bit wide parallel port which in turn addresses EPROM's in the SVP circuitry containing pre-computed data regarding the inverter switching states and state times. The modulation commands are signed eight-bit numbers which are combined to form the 16-bit EPROM address. EPROM data at a given address corresponds to the modulation commands used to generate this address. The NCP write signal to this 16-bit port is an interrupt signal to the SVP triggering the process of computing inverter switching states and state times.
- v.) The controller's learning process is executed after the forward propagation process and concurrent with the space vector PWM process triggered by the output of the modulation signals. The neural network weights are updated and used in the next forward propagation process.

The software structure of the controller dictates the minimum PWM sample interval $1/2f_s$ (or the maximum inverter switching frequency f_s). This minimum sample interval is either the combined forward propagation and space vector PWM process execution times or the combined forward propagation and learning process execution times (the longer of the

two). In the final version of the experimental neurocontroller, the maximum inverter switching frequency was 750 Hertz.

Additional external circuitry (external to the neurocontroller board) was required to generate the six PWM inverter switching control signals and insert 60 μ s of dead-time in each switching control signal.

In appendix 8, detailed circuit diagrams and a theory of operation for the experiment inverter neurocontroller is given.

III. SIMULATION AND EXPERIMENTAL RESULTS

3.1 Overview of Experimental and Simulation Studies

During the initial studies of the neurocontroller concept, the inverter neurocontroller utilized the conventional backpropagation learning algorithm and a single hidden layer. A hidden layer was included in the neurocontroller structure based on the assumption that a nonlinear neurocontroller input/output mapping was necessary. The use of the conventional backpropagation learning algorithm was consistent with other studies of neurocontrollers using on-line learning processes [32]. Simulation studies of this neurocontroller configuration were made using balanced passive and nonlinear load models. Simulation parameters were selected to reflect operation of an inverter using gate turn-off thyristors (GTO's). Inverter switching frequencies in the range of 540 to 1260 Hertz were utilized. Simulations were run with zero initial load currents and voltages.

The balanced passive load simulation results verified the ability of this inverter neurocontroller configuration to effectively eliminate the steady-state load voltage waveform errors

$$\epsilon_{dv} = v_{dl}^* - v_{dl} \quad (3.1)$$

$$\epsilon_{qv} = v_{ql}^* - v_{ql} \quad (3.2)$$

It was determined through the simulation studies that six hidden layer nodes were sufficient to minimize the errors ϵ_{dv} and ϵ_{qv} . Additional hidden layer nodes did not result in discernible improvements.

Nonlinear load simulations were run at inverter switching frequencies $f_s = 540$ Hertz and $f_s = 900$ Hertz. Nonlinear load simulations utilizing average load voltage control and the repetitive controller [3] were also run with the same simulation parameters. Results from these simulations served as a basis for the evaluation of this neurocontroller configuration given the nonlinear load condition. The observed steady-state performance of the neurocontroller with a hidden layer and using the conventional backpropagation learning algorithm was at least as good as the observed performance of the repetitive controller in the nonlinear load simulations.

Three potential problems regarding the application of the inverter neurocontroller in a direct adaptive control configuration are:

- i.) The interval of time required by the neurocontroller to minimize the errors ϵ_{dv} and ϵ_{qv} given zero initial load current and voltage conditions.
- ii.) The magnitude of the steady-state ϵ_{dv} and ϵ_{qv} errors given the use of relatively low inverter switching frequencies.
- iii.) The implementation of the neurocontroller outside of a simulated inverter/filter/load system.

The initial simulation-based evaluations of an inverter neurocontroller using a hidden layer and the conventional backpropagation learning algorithm revealed that the first two problems were not significant

given the appropriate choice of the neurocontroller learning-rate parameter. In the balanced passive load simulations, a learning-rate parameter value of 1.0 was utilized. In the nonlinear load simulations, the learning-rate parameter value had to be reduced to 0.1 due to the controller instability observed in steady-state with the use of higher learning-rate parameter values. The interval of time required by the neurocontroller to minimize the errors ϵ_{dv} and ϵ_{qv} is dependent on the inverter switching frequency and the learning-rate parameter. Based on the simulation results, this interval of time was comparable to the corresponding intervals in the repetitive controller simulations.

Consideration of the third problem lead to the study of neurocontrollers without hidden layers. An experimental neurocontroller was constructed based on a network structure without hidden layers. Experiments and simulations were performed using this neurocontroller structure given balanced passive, unbalanced passive, and nonlinear loads. Additionally, simulations using the repetitive controller were run given similar inverter/filter/load conditions.

Using the conventional backpropagation learning algorithm, the neurocontroller without hidden layers, or a single-layer neurocontroller, exhibited a significant steady-state limit cycle oscillation in each of the load experiments. The same condition was observed in the previous multi-layer neurocontroller simulations when:

- i.) The learning-rate parameter was increased appreciably beyond 0.1 in the nonlinear load simulations.
- ii.) Fewer than six hidden layer nodes were utilized in the

neurocontroller structure.

A modified backpropagation learning algorithm was devised for the single-layer neurocontroller with the objective of eliminating the limit-cycle instability condition. Subsequent experimental and simulation studies of the single-layer neurocontroller were based on the use of this learning algorithm.

The evaluation of the experimental single-layer neurocontroller steady-state performance was based on:

- i.) A comparison of instantaneous load voltage waveform steady-state errors measured in an open-loop mode, or learning process disabled, and errors measured in a closed-loop mode, or learning process enabled.
- ii.) A comparison of the steady-state load voltage waveform to the corresponding load voltage reference waveform.
- iii.) A comparison of experiment's load voltage spectral characteristics to the corresponding simulation load voltage spectral characteristics.

The experimental neurocontroller steady-state performance given a nonlinear load was compared to the repetitive controller's performance given the same load condition. As in the case of the multi-layer neurocontroller, the experimental single-layer neurocontroller's performance was at least as good as the performance of the repetitive controller. A conclusion drawn from this observation, as well as from the observed steady-state single-layer neurocontroller performance in the balanced and unbalanced passive load experiments, is that the value

of hidden layers in an inverter neurocontroller structure is questionable.

Finally, two dynamic control experiments were run with the single-layer controller where the instantaneous load voltage waveform errors were observed given the conditions:

- i.) DC link voltage variation from a nominal value (200 volts) to the minimum value (150 volts) corresponding to a 1.0 modulation index.
- ii.) 100% step load application.

The former experiment verified the neurocontroller's ability to perform closed-loop load voltage control given the maximum DC link voltage variation. The later experiment verified the use of load current feedback as neurocontroller inputs with the object of improving the controller's response to sudden load changes.

3.2 Initial Evaluation of Neurocontroller Operation

Using inverter/filter/load circuit simulation software, the initial neuro-controller concept, network with one hidden layer and six hidden layer nodes was evaluated. The control concept was tested using a balanced passive load model. Figures 3.1 and 3.2 depict the results obtained from the inverter/filter/load simulation where results are normalized using the DC power supply voltage where the inverter output fundamental frequency (f_m) was selected to be 60 Hertz and the inverter switching frequency (f_s) was selected to be 1260 Hertz. Other simulation parameters are given as follows:

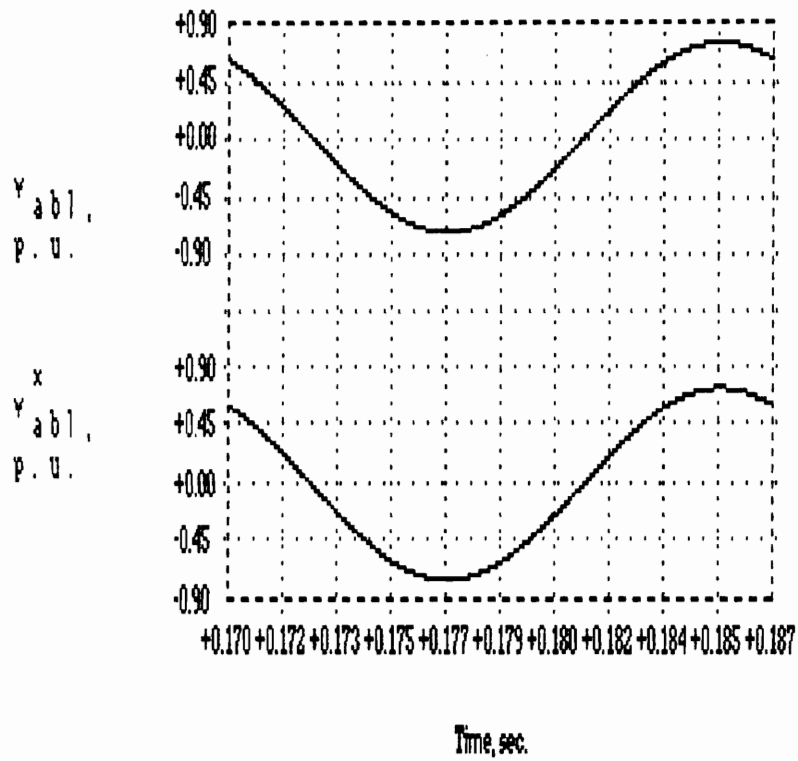


Figure 3.1: Phase A Line-to-Line Load and Corresponding Sinusoidal Reference Voltages Obtained from the Balanced Passive Load Simulation where $f_s = 1260$ Hz., $f_m = 60$ Hz., and $M = 0.75$.

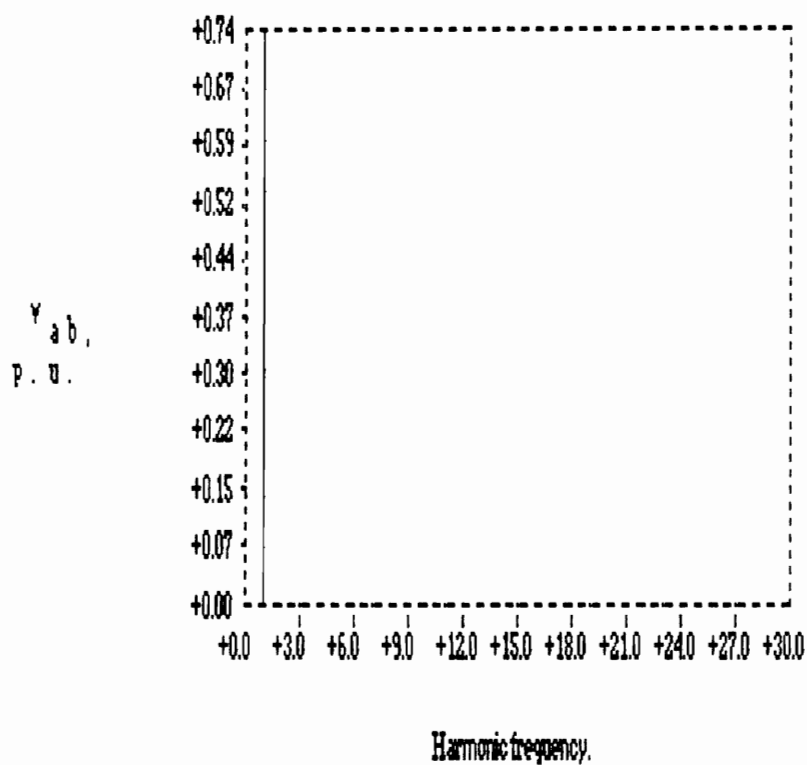


Figure 3.2: Phase A Line-to-Line Load Voltage Spectral Information, Balanced Passive Load Simulation, $f_s = 1260$ Hz., $f_m = 60$ Hz., and $M = 0.75$.

$L_f = 1688 \mu\text{H}$, $C_f = 132 \mu\text{F}$

Modulation index (M) = 0.75

DC supply voltage = 405 VDC

Line-to-line load resistance = 7.003Ω

Inverter switch turn-on time = $4 \mu\text{s}$

Inverter switch turn-off time = $8 \mu\text{s}$

Inverter switch on-state voltage = 3.6 volts.

A $20 \mu\text{s}$ dead-time was inserted in the inverter switch control signals. A learning-rate parameter of 1.0 was used in the passive load simulation. A simulation interval step of $8.333 \mu\text{s}$ was used, and each simulation was run until a steady-state condition was observed.

Ten modulation frequency periods after starting this simulation, the load voltage waveforms were compared to their corresponding sinusoidal references where figure 3.1 depicts the phase A waveforms in this comparison. The comparisons confirmed the anticipated operation of the neurocontroller with regard to the minimization of the load voltage errors $[v_{d1}^*(t) - v_{d1}(t)]$ and $[v_{q1}^*(t) - v_{q1}(t)]$.

Figure 3.2 depicts the spectral characteristics of the phase A line-to-line load voltage associated with this simulation. A negligible amount of harmonic distortion is observed in the load voltage waveform.

3.2.1 Nonlinear Load Simulations

Nonlinear load current waveforms of the type depicted in figure 3.3 contain significant low-order harmonic frequency components, particularly at the 5th and 7th harmonics. Nonlinear load simulations were run using the initial neuro-controller method at $f_s = 540$ Hertz and 900 Hertz. The modulation index command was the same as with the balanced passive load simulation. The output filter component values were selected to be $L_f = 1849 \mu\text{H}$ and $C_f = 360 \mu\text{F}$. A learning-rate parameter value of 0.1 was used. With the application of the nonlinear load model, a lower learning-rate parameter was required in this simulation due to the unstable operation displayed by the neurocontroller with a 1.0 learning-rate parameter. Simulations were run until steady-state operation was observed.

Figures 3.4 through 3.7 depict the results of nonlinear load simulations given inverter switching frequencies $f_s = 540$ Hz. and $f_s = 900$ Hz. (results obtained ten modulation frequency periods after the simulation start in the $f_s = 900$ Hz. simulation, 15 modulation frequency periods in the $f_s = 540$ Hz simulation). As in the case of the balanced passive load simulation, the neurocontroller minimizes the load voltage waveform fundamental frequency component amplitude and phase-displacement errors with respect to the sinusoidal reference. Harmonic distortion in the load voltage waveform due to the nonlinear load current is still present.

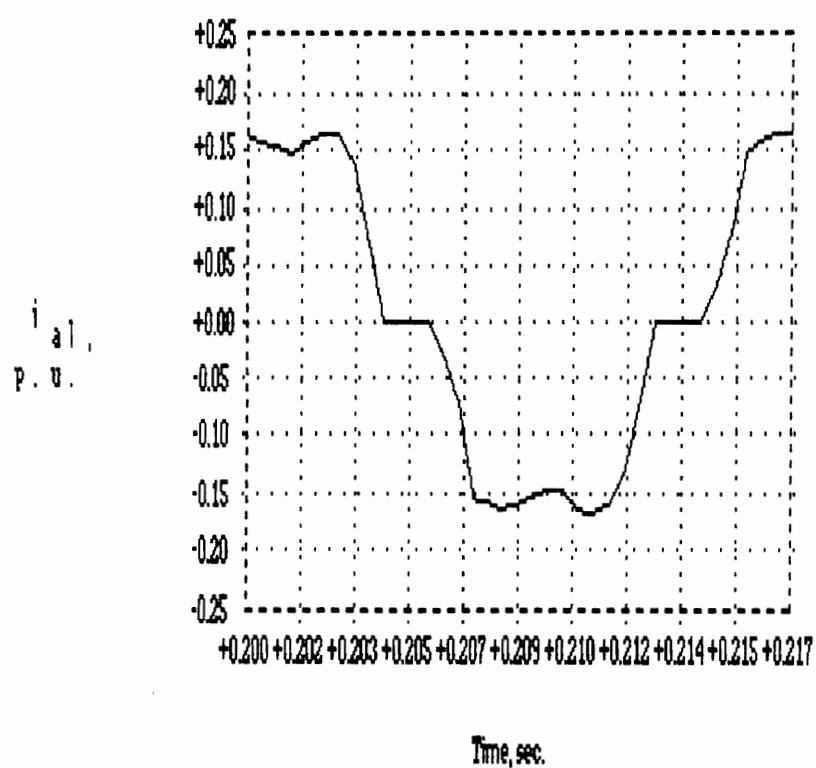


Figure 3.3: Nonlinear Load Current Generated by An Uncontrolled Three-Phase Rectifier

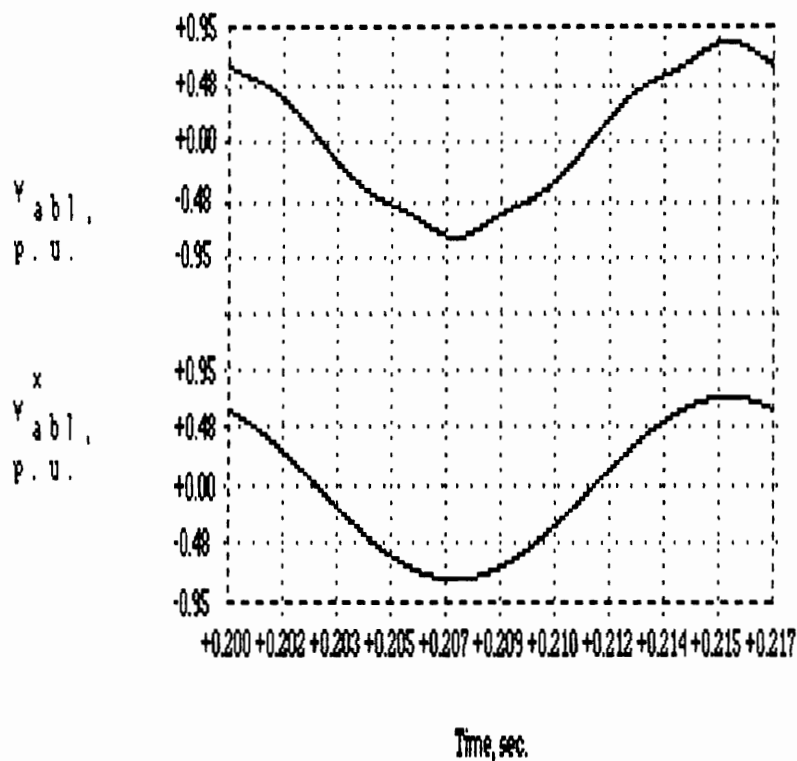


Figure 3.4: Nonlinear Load Simulation Phase A Line-to-Line Load and Reference Voltages, $f_s = 540$ Hz., $f_m = 60$ Hz., and $M = 0.75$.

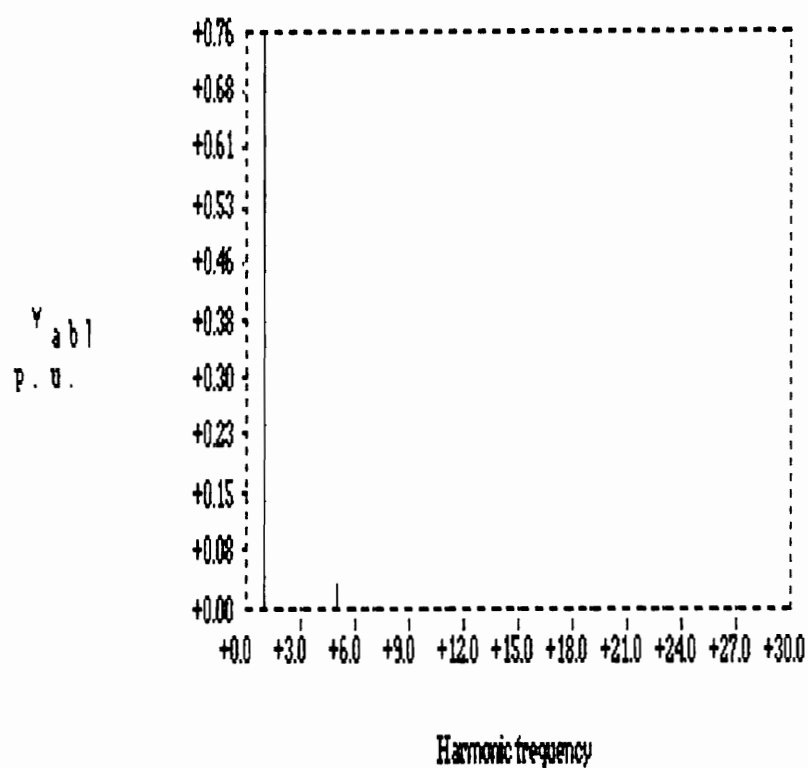


Figure 3.5: Nonlinear Load Simulation Phase A Line-to-Line Load Voltage Spectral Analysis $f_s = 540$ Hz., $f_m = 60$ Hz., and $M = 0.75$.

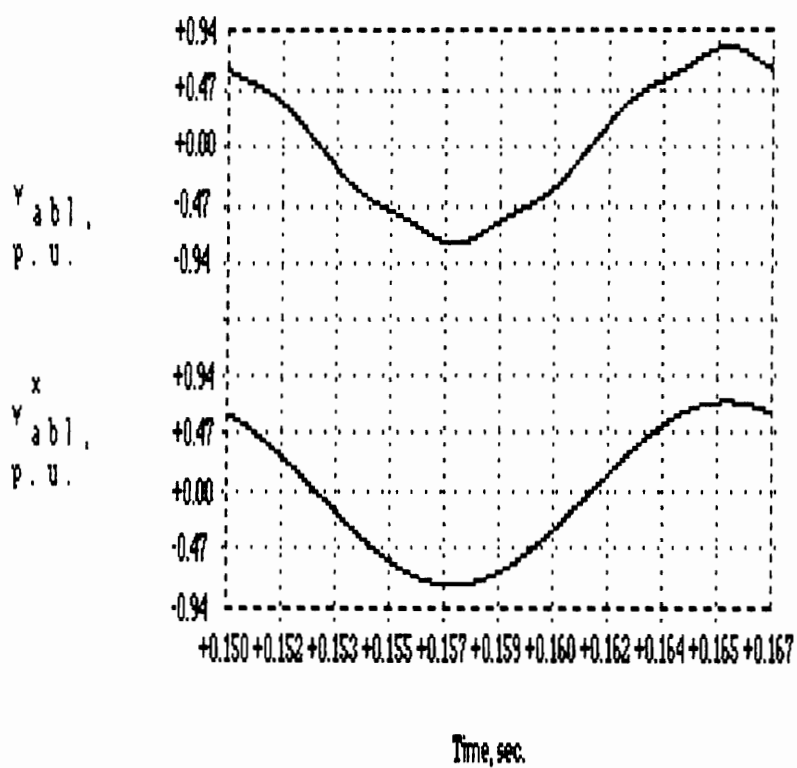


Figure 3.6: Nonlinear Load Simulation Phase A Line-to-Line Load and Reference Voltages $f_s = 900$ Hz., $f_m = 60$ Hz., and $M = 0.75$.

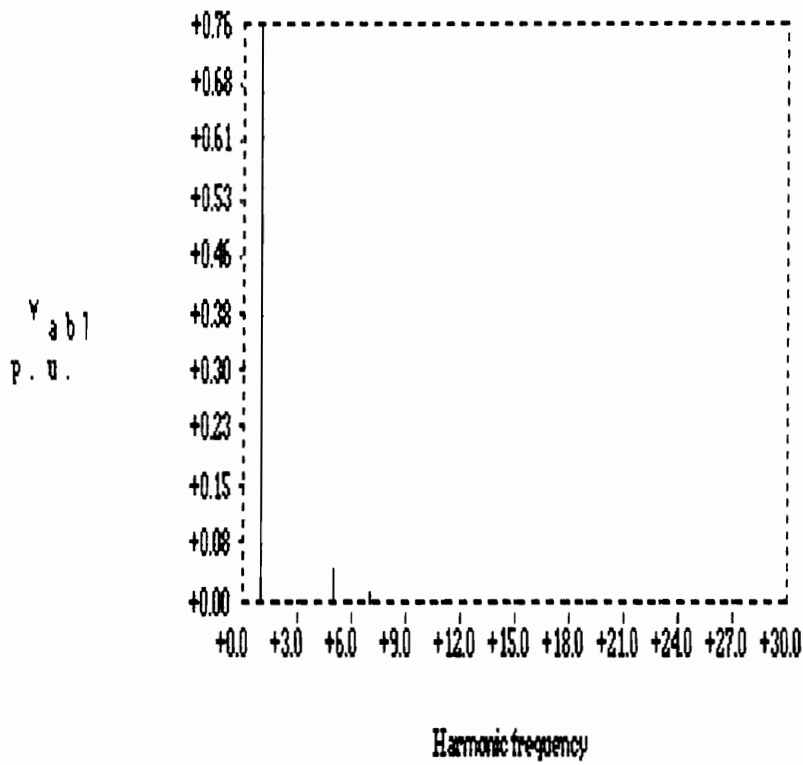


Figure 3.7: Nonlinear Load Simulation Phase A Line-to-Line Load Voltage Spectral Analysis $f_s = 900$ Hz., $f_m = 60$ Hz., and $M = 0.75$.

In the next section, the neurocontroller performance with regard to minimizing load voltage harmonic distortion due to nonlinear loads is compared to other methods of inverter control.

3.2.2 Comparisions with Conventional Inverter Control Methods

Nonlinear load simulations were run using an average load voltage controller and the repetitive load voltage controller described in reference [3]. As in the simulations of neurocontroller operation, the inverter switching pattern is derived through the use of the space vector PWM process. The equations defining the three-phase d-q frame repetitive controller operation are

$$u_{2d}(k \cdot T) = c_1 \cdot \epsilon_{dv}[(k + 1 - n) \cdot T] + c_2 \cdot \sum_{i=1}^{\infty} \epsilon_{dv}[(k + 1 - i \cdot n) \cdot T] \quad (3.3)$$

and

$$u_{2q}(k \cdot T) = c_1 \cdot \epsilon_{qv}[(k + 1 - n) \cdot T] + c_2 \cdot \sum_{i=1}^{\infty} \epsilon_{qv}[(k + 1 - i \cdot n) \cdot T] \quad (3.4)$$

where $\{u_{2d}, u_{2q}\}$ are the controller outputs, $\{\epsilon_{dv}, \epsilon_{qv}\}$ are the load voltage errors, $\{c_1, c_2\}$ are the controller gains, T is the controller sampling interval, k is a positive integer, and n is equal to the ratio $2 \cdot f_s / f_m$. The repetitive controller gains in the nonlinear load simulations were set to $c_1 = 0$ and $c_2 = 0.1$. Figures 3.8 through 3.11 depict the results of these simulations where $f_s = 540$ Hz.

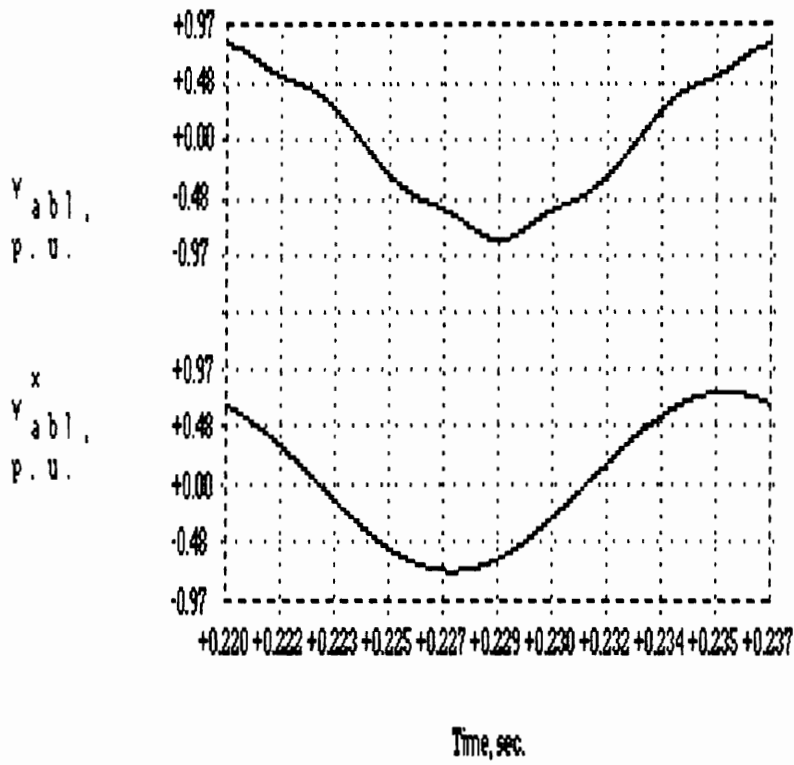


Figure 3.8: Nonlinear Load Simulation with Average Load Voltage Controller, Phase A Line-to-Line Load and Reference Voltages, $f_s = 540$ Hz, $f_m = 60$ Hz., and $M = 0.75$.

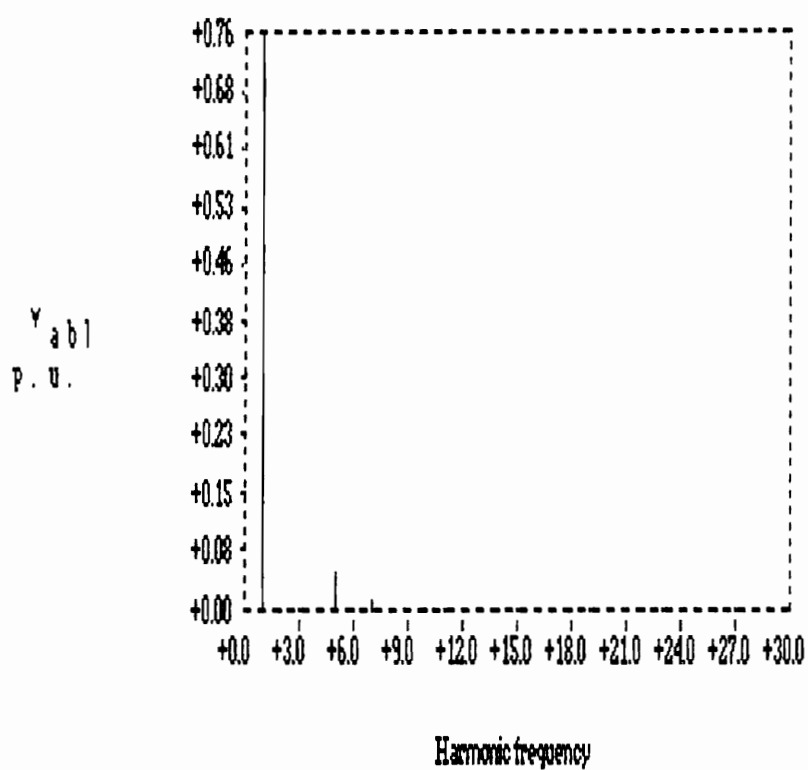


Figure 3.9: Nonlinear Load Simulation with an Average Load Voltage Controller, Phase A Line-to-Line Load Voltage Spectral Information, $f_s = 540$ Hz., $f_m = 60$ Hz., and $M = 0.75$.

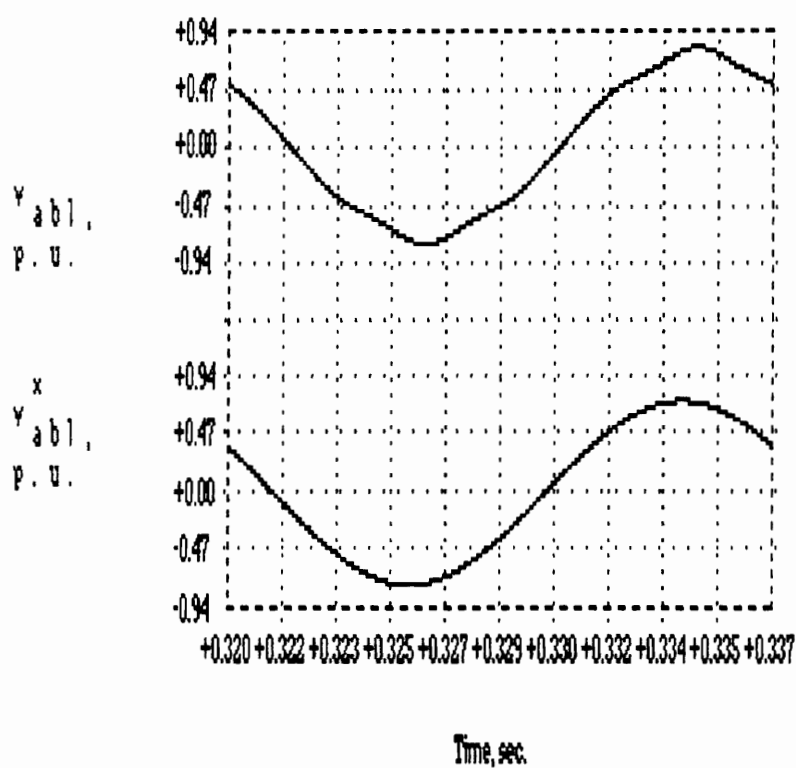


Figure 3.10: Nonlinear Load Simulation Using a Repetitive Load Voltage Controller, Phase A Line-to-Line Load and Reference Voltages, $f_s = 540$ Hz., $f_m = 60$ Hz., and $M = 0.75$.

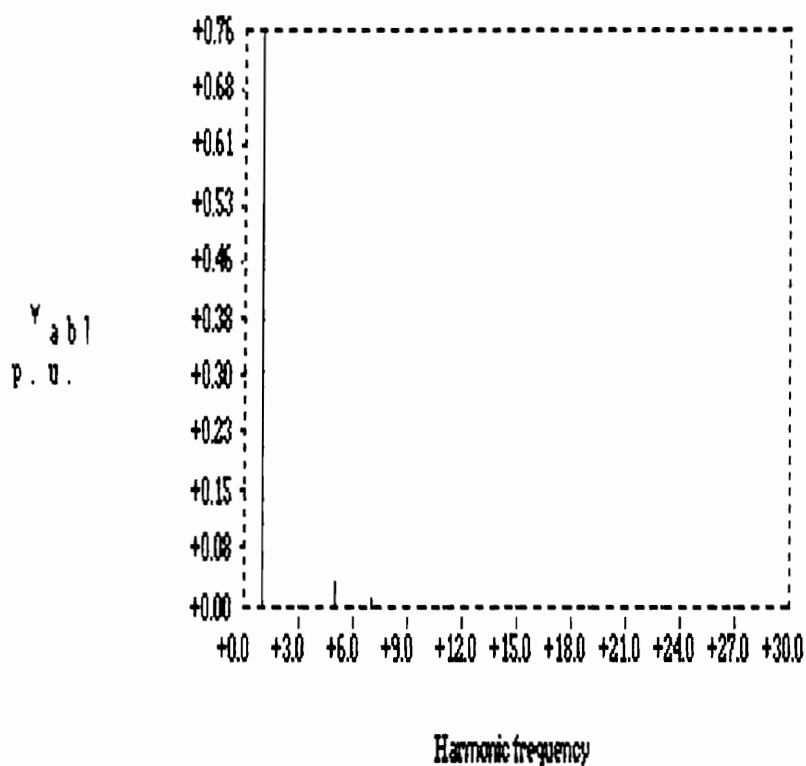


Figure 3.11: Nonlinear Load Simulation Using a Repetitive Load Voltage Controller, Phase A Line-to-Line Load Voltage Spectral Information, $f_s = 540$ Hz., $f_m = 60$ Hz., and $M = 0.75$.

The average load voltage controller results serve as the benchmarks for the performance of both the repetitive controller and the neurocontroller. The scope of average load voltage control is limited to the control of the load voltage fundamental frequency component amplitude. Subsequently, waveform phase-displacements due to controller delays, load current flow through the output filter series inductance L_f , and the inherent delay in the filter circuit waveform synthesis process are present in the load voltage waveforms resulting from the use of this method. The cumulative phase-displacement relative to the sinusoidal reference waveform is evident in figure 3.8. The average controller does not compensate for load voltage harmonic distortion, and so, the harmonic distortion due to nonlinear loading in the corresponding load voltage waveforms represents the uncompensated load voltage harmonic distortion.

The performance of the repetitive controller is similar to the performance of the neurocontroller. This is to be expected since the objective of the repetitive controller is to eliminate the steady-state instantaneous load voltage waveform errors (the same objective as in the case of neurocontroller operation). The primary difference between the two methods is the fact that neurocontroller operation achieves this objective without knowledge of the output filter circuit design or the inverter switching frequency.

Tables 3.1 through 3.4 give the results of the nonlinear load simulations involving the average controller, repetitive controller, and neurocontroller methods.

Table 3.1: Comparison of Controller Techniques, Nonlinear Load. $f_s = 540$ Hz, $f_m = 60$ Hz. Load Voltage Harmonic Components Normalized with Respect to the Fundamental Frequency Component.

CONTROLLER TECHNIQUE:	NEURO-CONTROLLER	AVERAGE	REPETITIVE
HARMONICS	P.U.	P.U.	P.U.
3	0.006	0.002	0.005
5	0.044	0.060	0.033
7	0.003	0.014	0.011
9	0.001	0.000	0.001
11	0.002	0.005	0.005
13	0.002	0.003	0.003
15	0.001	0.0	0.0
17	0.006	0.005	0.005
19	0.004	0.003	0.003
21	0.0	0.0	0.0
23	0.002	0.001	0.001
25	0.002	0.001	0.001
THD	0.046	0.062	0.036

Table 3.2: Comparison of Controller Techniques, Nonlinear Load. $f_s = 540$ Hz, $f_m = 60$ Hz. Inverter Output Current Harmonic Components Normalized with Respect to the Fundamental Frequency Component.

CONTROLLER TECHNIQUE:	NEURO-CONTROLLER	AVERAGE	REPETITIVE
HARMONICS	P.U.	P.U.	P.U.
3	0.005	0.003	0.014
5	0.062	0.094	0.016
7	0.051	0.043	0.077
9	0.000	0.001	0.002
11	0.042	0.035	0.062
13	0.021	0.045	0.012
15	0.002	0.001	0.002
17	0.083	0.069	0.077
19	0.065	0.049	0.064
21	0.001	0.001	0.0
23	0.032	0.026	0.026
25	0.027	0.018	0.030
THD	0.145	0.147	0.147

Table 3.3: Comparison of Controller Techniques, Nonlinear Load. $f_s = 900$ Hz, $f_m = 60$ Hz. Load Voltage Harmonic Components Normalized with Respect to the Fundamental Frequency Component.

CONTROLLER TECHNIQUE:	NEURO-CONTROLLER	AVERAGE	REPETITIVE
HARMONICS	P.U.	P.U.	P.U.
3	0.002	0.002	0.009
5	0.046	0.048	0.051
7	0.011	0.012	0.012
9	0.0	0.0	0.001
11	0.003	0.004	0.005
13	0.004	0.0	0.005
15	0.0	0.002	0.0
17	0.003	0.002	0.002
19	0.001	0.001	0.001
21	0.0	0.0	0.0
23	0.0	0.0	0.001
25	0.001	0.0	0.0
THD	0.048	0.05	0.054

Table 3.4: Comparison of Controller Techniques, Nonlinear Load. $f_s = 900$ Hz, $f_m = 60$ Hz. Inverter Output Current Harmonic Components Normalized with Respect to the Fundamental Frequency Component.

CONTROLLER TECHNIQUE:	NEURO-CONTROLLER	AVERAGE	REPETITIVE
HARMONICS	P.U.	P.U.	P.U.
3	0.001	0.010	0.004
5	0.039	0.045	0.056
7	0.004	0.011	0.013
9	0.001	0.002	0.001
11	0.024	0.027	0.023
13	0.030	0.030	0.033
15	0.0	0.001	0.001
17	0.028	0.027	0.031
19	0.020	0.022	0.021
21	0.001	0.001	0.001
23	0.005	0.009	0.008
25	0.008	0.010	0.006
THD	0.065	0.073	0.08

At $f_s = 540$ Hertz, the load voltage total harmonic distortion (THD) is the lowest using the repetitive controller method. The neurocontroller method is second in this contest primarily as a result of the 5th harmonic component magnitude. With regard to the 7th, 11th, and 13th harmonic frequency components, the neurocontroller method has the advantage in the suppression of load voltage waveform harmonic distortion. The neurocontroller method at $f_s = 540$ Hertz exhibits a superior inverter output current THD (Table 3.2) compared to the other conventional controller methods. This translates into decreased heat losses in the inverter circuit due to the flow of harmonic frequency inverter output current.

At $f_s = 900$ Hertz, neurocontroller performance with respect to the load voltage and inverter output current THD is consistent with the quality of the results obtained in the $f_s = 540$ Hertz simulation. On the other hand, the repetitive controller performance appears to have degraded in this simulation.

The increase in the repetitive controller's sampling rate would imply an improvement in this controller's ability to compensate for nonlinear load-generated load voltage waveform distortion. This would be the case if the nonlinear load current spectral characteristics were not dependent on the instantaneous load voltage waveform characteristics or the load currents are modeled as independent current sources. The spectral characteristics of nonlinear load currents given load models such as the uncontrolled three-phase rectifier vary with changes in the three-phase load voltages due to controller activity.

The repetitive controller eliminates the steady-state load voltage errors given the assumption that the disturbances distorting the load voltages are periodic and the disturbance frequencies are less than f_s . The performance of the repetitive controller can be compromised if load voltage errors are not periodic as in the case of nonlinear load current variations due to controller activity. Non-zero steady-state load voltage errors occur as a result.

The performance of the repetitive controller in the $f_s = 900$ Hertz nonlinear load simulation is worst than the performance of the average controller. This implies that the repetitive controller activity is creating the non-periodic response described previously from the nonlinear load model. Given that the repetitive controller operation at 540 Hertz did not exhibit this problem, resolution of this problem at 900 Hertz would be achieved by intentionally slowing down the rate of repetitive controller activity. This would be done by either applying a low-pass filter to the load model current and voltage feedback or by reducing the controller gains.

As noted previously, the neurocontroller's learning-rate parameter, analogous to the repetitive controller c_2 gain, was lowered from a value of 1.0 to 0.1 with the application of the nonlinear load model. The load current positive feedback in the neurocontroller which led to unstable system operation was the reason for the reduction of the learning-rate parameter.

The neurocontroller concept is not based on assumptions made concerning inverter/filter/load circuit operation. As in the case of

other nonlinear controller methods, the neurocontroller is relatively insensitive to parameter variations and system modeling errors.

3.3 Evaluation of An Experimental Neurocontroller

The implementation of the neurocontroller concept utilized a neural network without a hidden layer. The network learning process in the first operational neurocontroller employed the conventional backpropagation algorithm. As in the evaluation of the initial neurocontroller version, inverter/filter/load simulations were run using this controller version. Additionally, experiments were performed with the operational controller which paralleled the computer simulations. Experiments were performed with a modulation frequency (f_m) of 50 Hertz, a modulation index command of 0.75, and an inverter switching frequency (f_s) of 750 Hertz.

The outputs of the neuro-controller board, inverter leg circuit switching commands $\{S_a, S_b, S_c\}$, were applied to the switching control of a three-phase bridge inverter using FET's as the switching elements. $\{S_a, S_b, S_c\}$ were converted to the required six FET gate-to-source control voltages with a 60 μ s dead time interval inserted in each of these six control voltages. A Undeland snubber configuration was employed in the inverter bridge circuit (details depicted in figure 1.4).

An output filter circuit depicted in figure 1.2 was fabricated using 10 mH inductors (L_f value) and 45 μ F AC capacitors (C_f value). This configuration resulted in a filter pole frequency of 137 Hertz, and

a 1.15 output filter gain at 50 Hertz. Variations between the actual inductor and capacitor component values did not exceed 10%.

Three steady-state load conditions were examined:

- i.) A balanced resistive three-phase delta-connected load,
- ii.) An unbalanced resistive load (two-phase load), and
- iii.) A nonlinear load (an uncontrolled three-phase rectifier with a resistive load and L-C output filter).

Steady-state experiments were performed with the controller board operating in an open-loop mode (learning process disabled and initial ANN connection weights in use), and with the controller board in the closed-loop mode (learning process enabled). Measurements in the open-loop mode were taken for the purpose of establishing a benchmark in evaluating the neurocontroller performance.

Additional experiments were performed with regard to the examination of controller performance under dynamic conditions. First with the controller operating in a closed-loop mode, the inverter's DC power supply voltage was varied between 150 and 200 volts DC. Second with the controller again operating in a closed-loop mode and the balanced passive load connected, a 100% step load was applied. In both of these experiments, the load voltage error was measured in relation to the applied disturbance.

Waveform measurements of load and reference voltages, load currents, and load voltage errors were made from points within the neurocontroller circuitry. Waveform amplitudes are on the order of levels typical of analog electronic circuitry. Load voltage error

measurement and display is an auxiliary function of the neurocontroller circuitry (a function of the ANCP processor).

3.3.1 Balanced Resistive Load Experiment Results

With the neurocontroller operating in an open-loop mode and a balanced resistive load connected, the following inverter/filter circuit measurements were made:

$$\begin{array}{lll} V_{dc} = 200.0 \text{ volts} & I_{dc} = 1.08 \text{ amps} & \\ V_{abl} = 107.5 \text{ volts} & V_{bcl} = 108.2 \text{ volts} & V_{cal} = 104.6 \text{ volts} \\ I_{al} = 1.0 \text{ amps} & I_{bl} = 1.0 \text{ amps} & I_{cl} = 1.0 \text{ amps.} \end{array}$$

Corresponding measurements of the load voltage and sinusoidal reference waveforms are depicted in figures 3.12 and 3.13. In figure 3.14, the d- and q-axis load voltage error waveforms are depicted.

With the neurocontroller operating in a closed-loop mode, the conventional backpropagation learning algorithm utilized, the learning-rate parameter set to 0.066, and a balanced resistive load connected, the following inverter/filter circuit measurements were made:

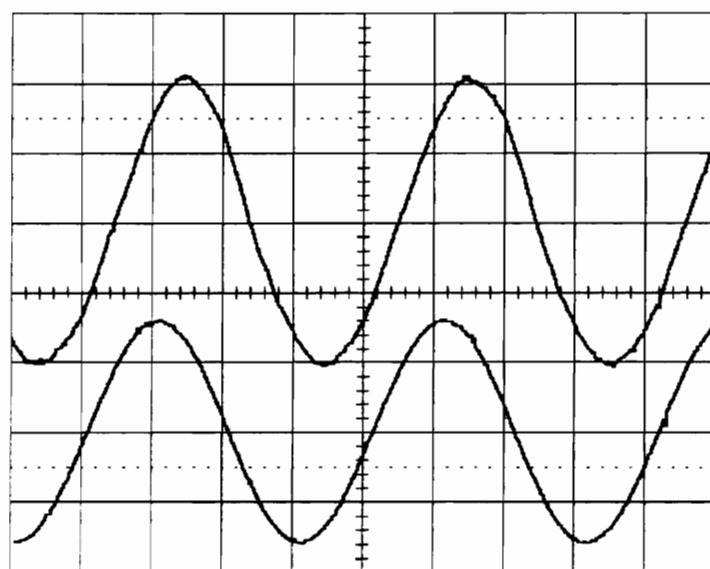
$$\begin{array}{lll} V_{dc} = 200.0 \text{ volts} & I_{dc} = 1.09 \text{ amps} & \\ V_{abl} = 107.5 \text{ volts} & V_{bcl} = 106.5 \text{ volts} & V_{cal} = 106.6 \text{ volts} \\ I_{al} = 1.0 \text{ amps} & I_{bl} = 1.0 \text{ amps} & I_{cl} = 1.0 \text{ amps.} \end{array}$$

Corresponding measurements of the load voltage and sinusoidal reference waveforms are depicted in figures 3.15 and 3.16. In figure 3.17, the d- and q-axis load voltage error waveforms are depicted.

20-Sep-93
11:34:09

1: Eres(1)
5 ms
1.00 V

2: Eres(2)
5 ms
1.00 V



5 ms BWL

1 .1 V AC ∞

2 .1 V AC ∞  **2** DC 0.00 V

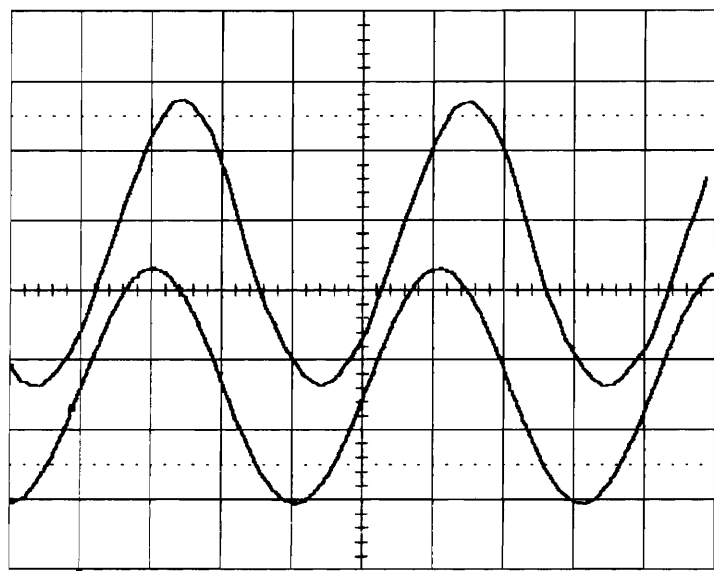
☐ STOPPED

Figure 3.12: D-Axis Load Voltage (Top Trace) Verses D-Axis Load Voltage Reference (Bottom Trace), Balanced Resistive Load, Controller Operating in an Open-Loop Mode.

17-Sep-93
13:42:17

A: Eres(1)
5 ms
1.00 V

B: Eres(2)
5 ms
1.00 V



5 ms BWL

1 .1 V AC

2 .1 V AC

2 DC 0.00 V

□ STOPPED

Figure 3.13: Q-Axis Load Voltage (Top Trace) Verses Q-Axis Load Voltage Reference (Bottom Trace), Balanced Resistive Load, Controller Operating in an Open-Loop Mode.

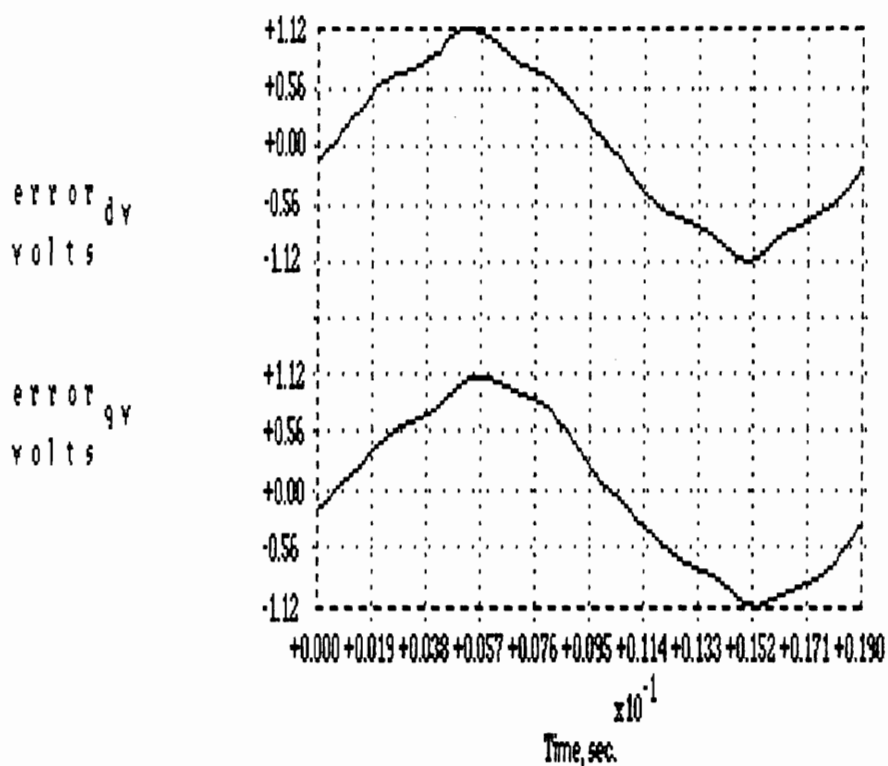


Figure 3.14: D-Axis Load Voltage Error (Top Trace) Verses Q-Axis Load Voltage Error (Bottom Trace), Balanced Resistive Load, Controller Operating in an Open-Loop Mode.

17-Sep-93
14:31:41

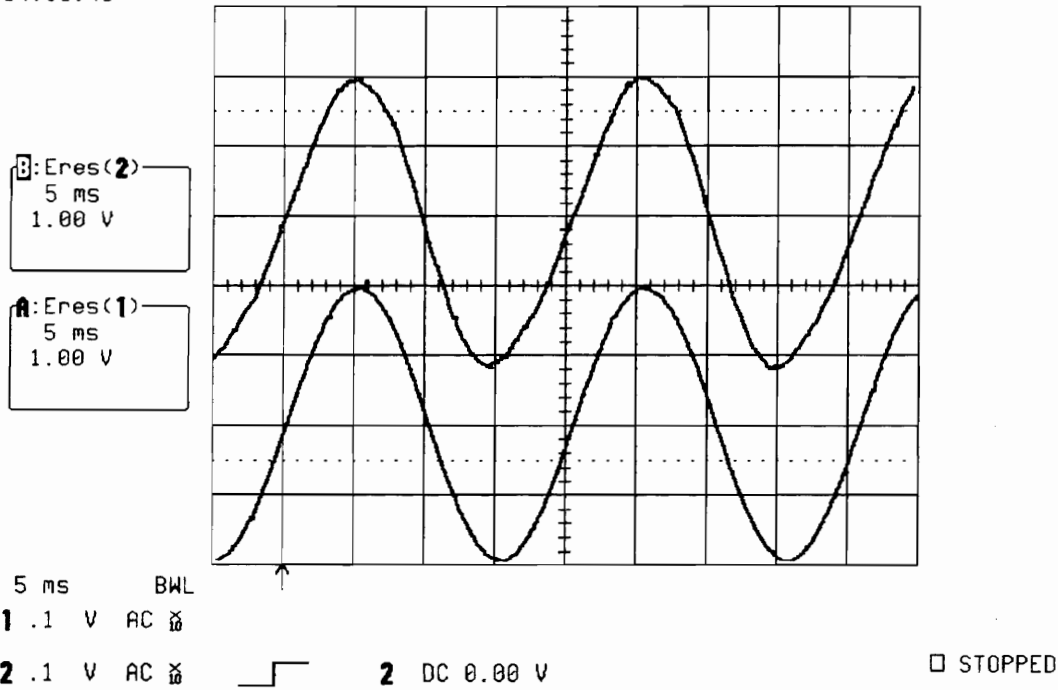


Figure 3.15: D-Axis Load Voltage (Top Trace) Verses D-Axis Load Voltage Reference (Bottom Trace), Balanced Resistive Load, Controller Operating in a Closed-Loop Mode Using the Conventional Backpropagation Algorithm.

17-Sep-93
14:36:00

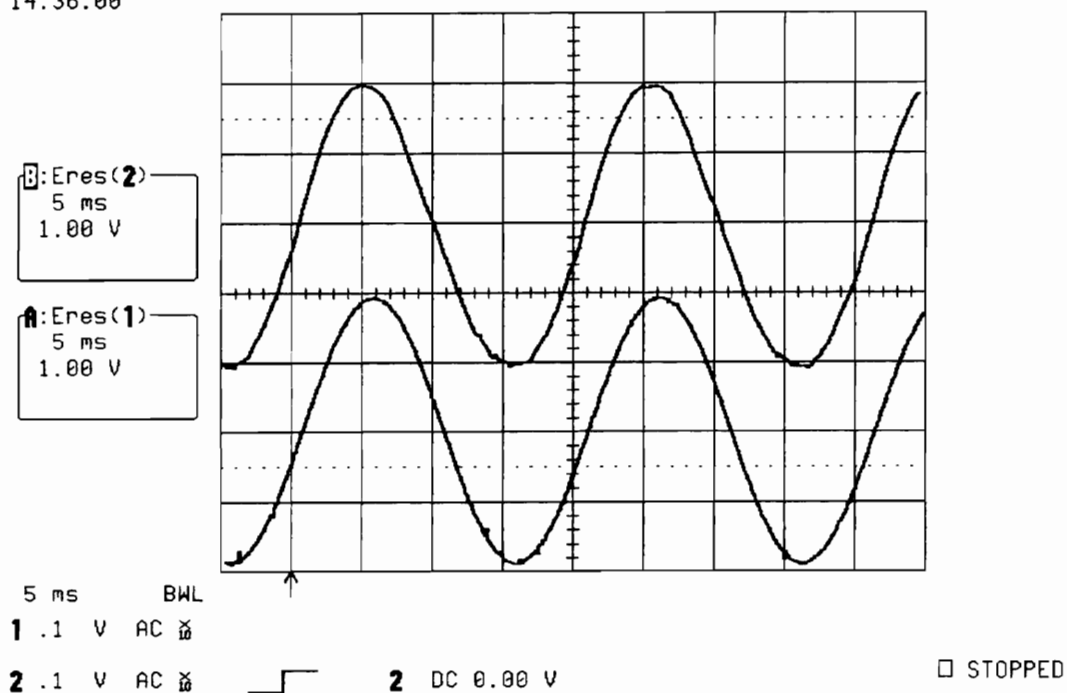


Figure 3.16: Q-Axis Load Voltage (Top Trace) Verses Q-Axis Load Voltage Reference (Bottom Trace), Balanced Resistive Load, Controller Operating in a Closed-Loop Mode Using the Conventional Backpropagation Algorithm.

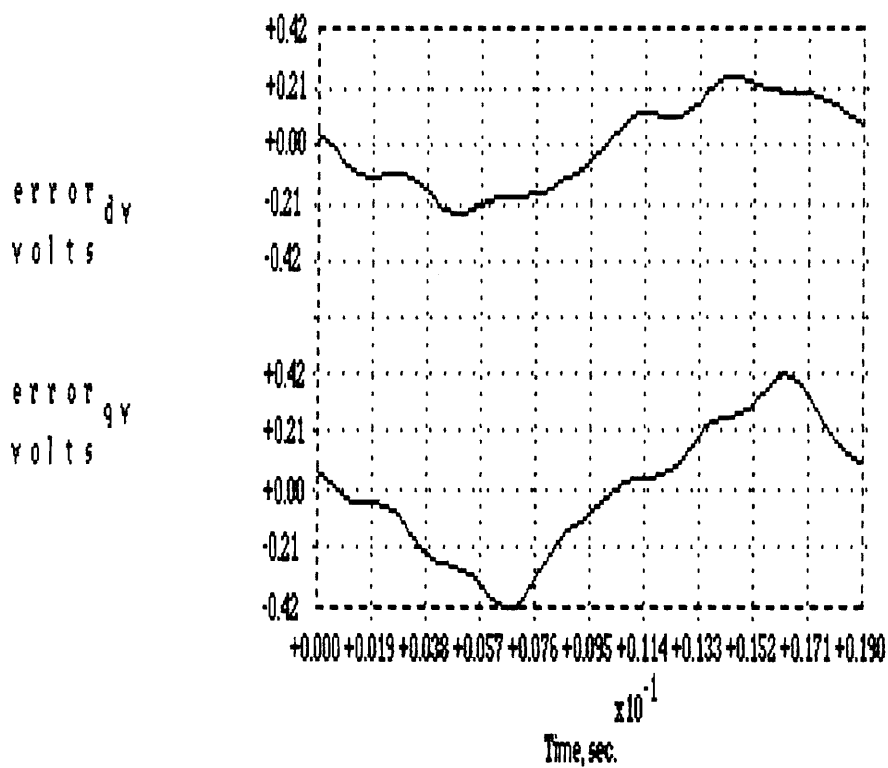


Figure 3.17: D-Axis Load Voltage Error (Top Trace) Verses Q-Axis Load Voltage Error (Bottom Trace), Balanced Resistive Load, Controller Operating in a Closed-Loop Mode Using the Conventional Backpropagation Algorithm.

A comparison of open- and closed-loop load voltage errors depicted in figures 3.14 and 3.17 reveals a reduction of instantaneous load voltage errors expected with closed-loop controller operation. Ideally, load voltage errors should be eliminated. Non-zero instantaneous load voltage errors exist in the closed-loop controller mode due to the following factors:

- i.) Controller and space vector PWM processes utilize 8-bit binary values and fixed-point arithmetic (16-bit) in their respective software routines. Quantization errors due to the limited precision occur. One process where the effect of these types of errors are the most noticeable is the generation of the inverter leg control signals $\{S_a, S_b, S_c\}$.
- ii.) The adjustment of the inverter modulation frequency (f_m) is separate from the inverter switching frequency (f_s) setting. As a result, the ratio f_s/f_m is not necessarily an integer, and the controller input current and voltage sampling process may not be periodic with respect to the modulation frequency period ($1/f_m$).
- iii.) The inverter DC power supply (a single-phase uncontrolled rectifier with an L-C filter and fed from the 60 Hertz mains) has a small but detectable ripple voltage (primary harmonic component at 120 Hertz). Also, this supply is not particularly stiff and inverter load changes have resulted in 10% changes in the DC supply voltage value.
- iv.) The inverter switching control with regard to following a

trajectory determined by the inverter load inverse dynamics is limited. The inverter switching frequency and inverter switching distortion effects, particularly with regard to the effects of dead-time insertion, are examples of the limiting factors.

- v.) Analog-to-digital converters (ADC) introduce errors into the load voltage and current feedback signals due to nonlinearity in the converter's forward transfer function. The degree of ADC nonlinearity is dependent on the operational quality of the ADC utilized in the controller. In the controller constructed for these experiments, the ADC converts analog signals to unsigned 8-bit binary values where the analog signals were offset by +2.5 volts prior to conversion. This offset was removed from the converted feedback signals in the controller's program. Any ADC nonlinearity would distort the converted feedback signals, preventing load voltage error elimination.
- vi.) The order in which the forward propagation and learning processes are executed in each controller sampling period is significant with regard to the steady-state phase-displacement of the load voltage waveforms in relation to the corresponding reference waveforms. When the learning process is executed before the forward propagation process in the controller sampling period, changes in the controller's weights due to the learning process alter the inverter switching pattern executed in the following controller sampling period, and the steady-state load voltage waveform phase-displacement is small. If the learning process is

executed after the forward propagation process in the controller sampling period, changes in the controller's weights due to the learning process alter the inverter switching pattern executed two controller sampling periods later, and the steady-state load voltage waveform will have a leading phase-displacement in relation to the reference waveform.

Condition (vi.) is the primary contributor to non-zero instantaneous load voltage errors in the controller experiments due to the fact that the controller program executes the forward propagation process before the learning process. This order of process execution was necessary due to controller execution time limitations associated with generating the PWM inverter switching commands executed in the following controller sampling period. The forward propagation process generates the inverter output voltage commands v_{di}^* and v_{qi}^* which are in turn converted to inverter switching commands and switching state intervals by the space vector PWM processor (SVP). The execution time of the controller's learning process prevented execution of the learning process prior to the forward propagation process.

In the neurocontroller simulations where the learning process is executed before the forward propagation process, the leading load voltage waveform phase-displacement did not occur. When the neurocontroller simulation software is altered so that the learning process execution follows the forward propagation process, load voltage

error waveforms and the load voltage phase-displacements compare favorably to the experimental results.

Another source of load voltage error in the controller closed-loop mode was found in connection with the use of the conventional backpropagation learning algorithm in a neural network configuration without hidden layers. Obtained from the computer simulation which paralleled this experiment, the spectral information associated with the phase A line-to-line load voltage resulting from closed-loop operation is depicted in figure 3.18. Compared to the results obtained from the initial neurocontroller simulation (neural network with a hidden layer) depicted in figure 3.2, the load voltage waveform resulting from this neurocontroller configuration (without hidden units) contains significant second and third harmonic frequency components, as well as, a fundamental frequency component which is higher than the desired 0.75 amplitude. In section 2.3.3, the source of the low-order harmonic generation and fundamental voltage error is described. A modified backpropagation learning algorithm devised to eliminate this problem is described in this section.

Using this new learning algorithm in the experimental controller, the balanced resistive load experiment was performed again. Figure 3.19 depicts the q-axis load and reference voltages measured in this experiment. Figure 3.20 depicts the same set of voltages obtained from the corresponding balanced resistive load simulation where these voltages have been normalized with respect to the peak value of v_{q1}^* . Figure 3.21A depicts the open- and closed-loop q-axis load voltage

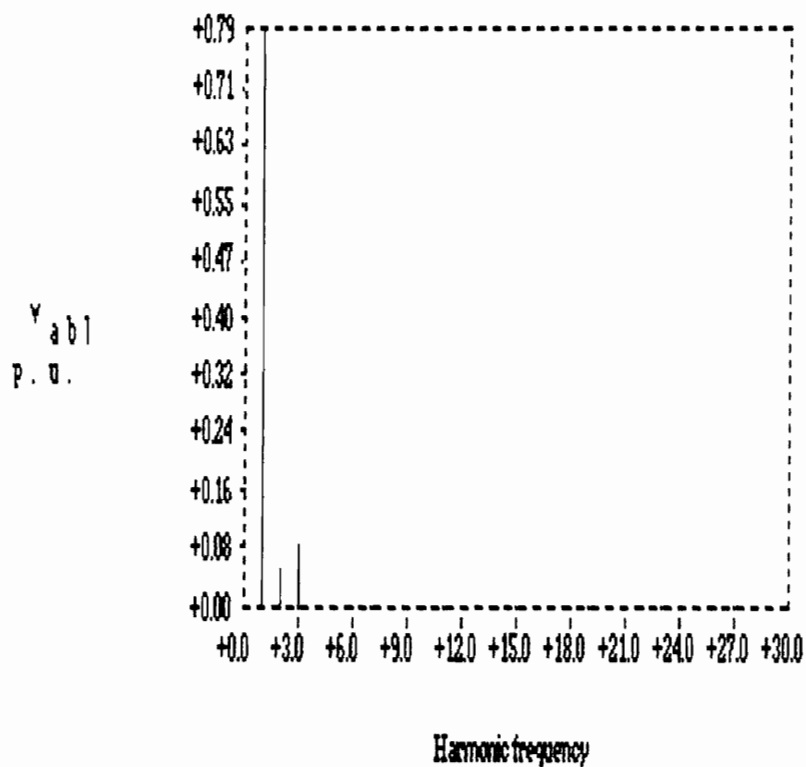
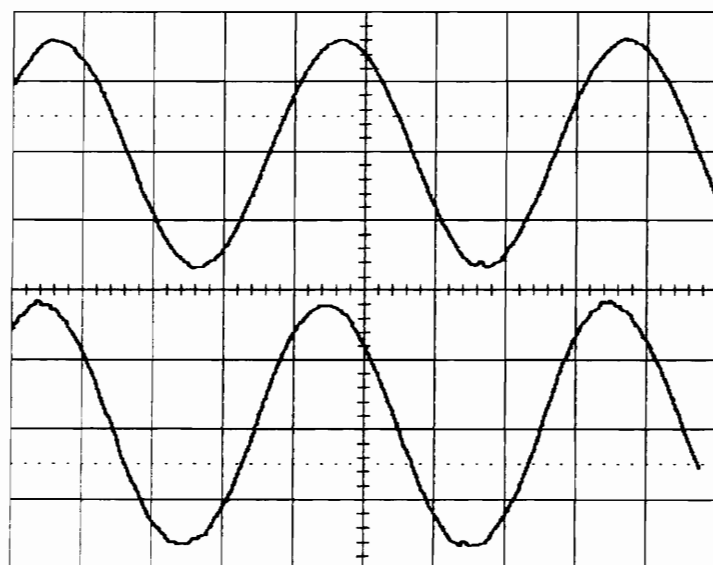


Figure 3.18: Phase A Line-to-Line Load Voltage Spectral Information Obtained from the Balanced Resistive Load Simulation, Conventional Backpropagation Learning Process Enabled.

29-Oct-93
13:41:27

B:Eres(2)
5 ms
1.00 V

A:Eres(1)
5 ms
1.00 V



5 ms

1 .1 V AC

2 .1 V AC



2 DC 3.32 V

□ STOPPED

Figure 3.19: Q-Axis Load Voltage Reference (Top Trace) Verses the Q-Axis Load Voltage in the Balanced Resistive Load Experiment, Modified Backpropagation Learning Process Enabled.

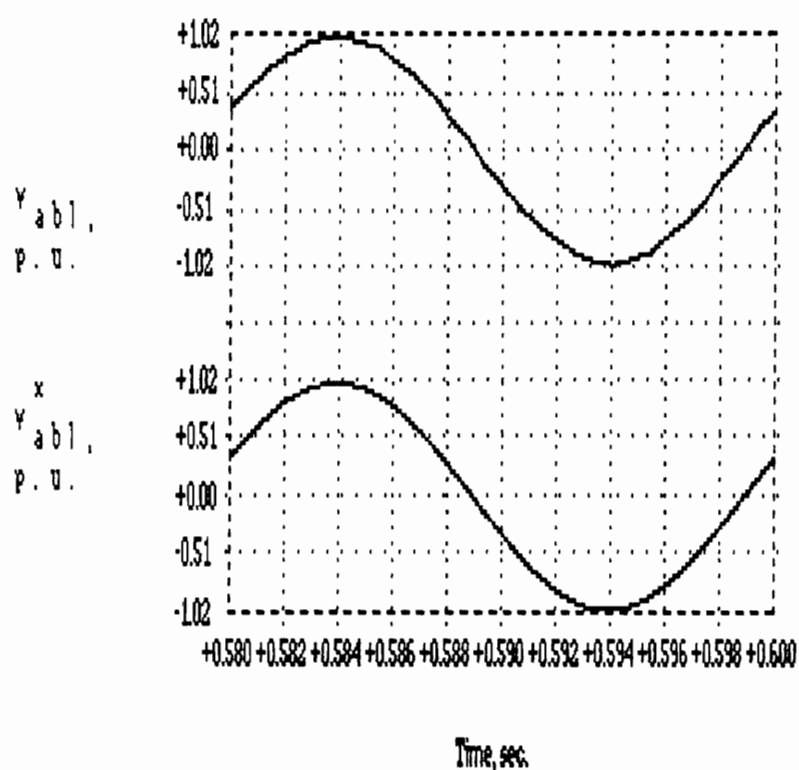


Figure 3.20: Q-Axis Load Voltage (Top Trace) Verses the Q-Axis Load Voltage Reference (Bottom Trace) in the Balanced Resistive Load Simulation, Modified Backpropagation Learning Process Enabled.

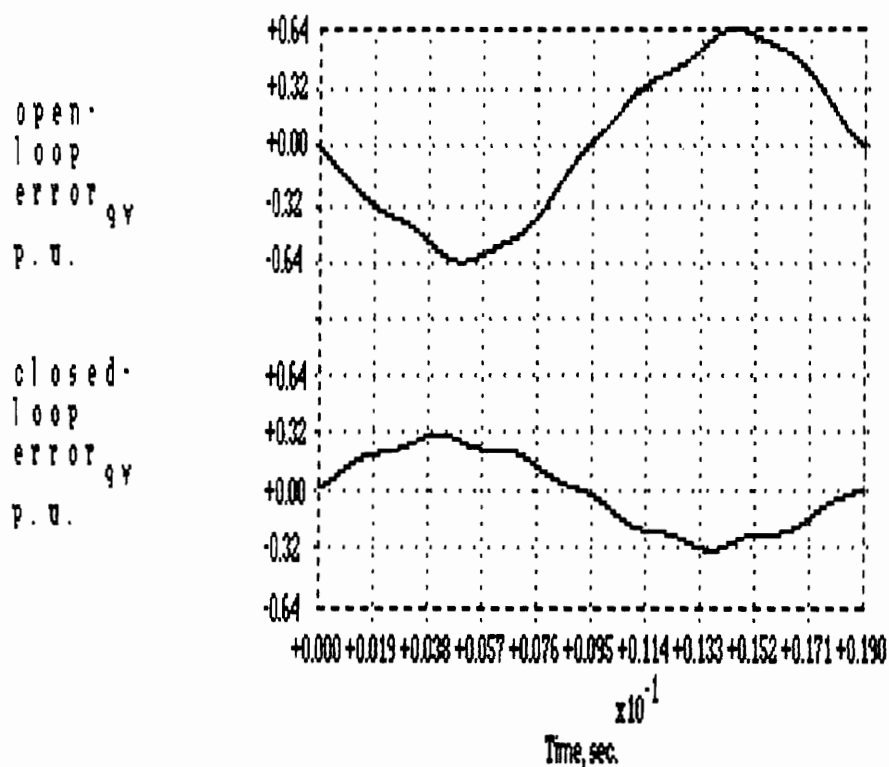


Figure 3.21A: Open-Loop Normalized Q-Axis Load Voltage Error (Top Trace) Verses the Closed-Loop Normalized Q-Axis Load Voltage Error (Bottom Trace), Balanced Resistive Load Experiment.

errors normalized with respect to the peak value of v_{q1}^* where the closed-loop results were obtained using the modified backpropagation learning algorithm. Figures 3.22 and 3.23 depict the q-axis load voltage waveform spectral information, normalized to the fundamental voltage amplitude, obtained from the balanced resistive load simulation and experiment respectively.

Both experimental and simulation results demonstrate the elimination of the second harmonic load voltage waveform distortion encountered with the use of the conventional backpropagation learning algorithm. The per unit steady-state fundamental load voltage amplitude was reduced from 1.151 in the open-loop experiment to 1.005 in the closed-loop experiment. The measured fundamental load voltage phase-displacement relative to the reference voltage was changed from a lagging 37.2° in the open-loop mode to a leading 17.2° in the closed-loop mode. The observed steady-state leading load voltage waveform phase-displacement is due to the one controller sampling period learning process delay created by executing the learning process after the forward propagation process in each controller sampling period. In figure 3.21B, the steady-state q-axis load voltage errors obtained from the balanced resistive load simulations are depicted where the top trace corresponds to load voltage errors without a learning process delay, or the learning process is executed first in the controller sampling period, and the bottom trace corresponds to load voltage errors with a one sampling period delay, or the learning process is executed after the forward propagation process. Simulation results where a learning process

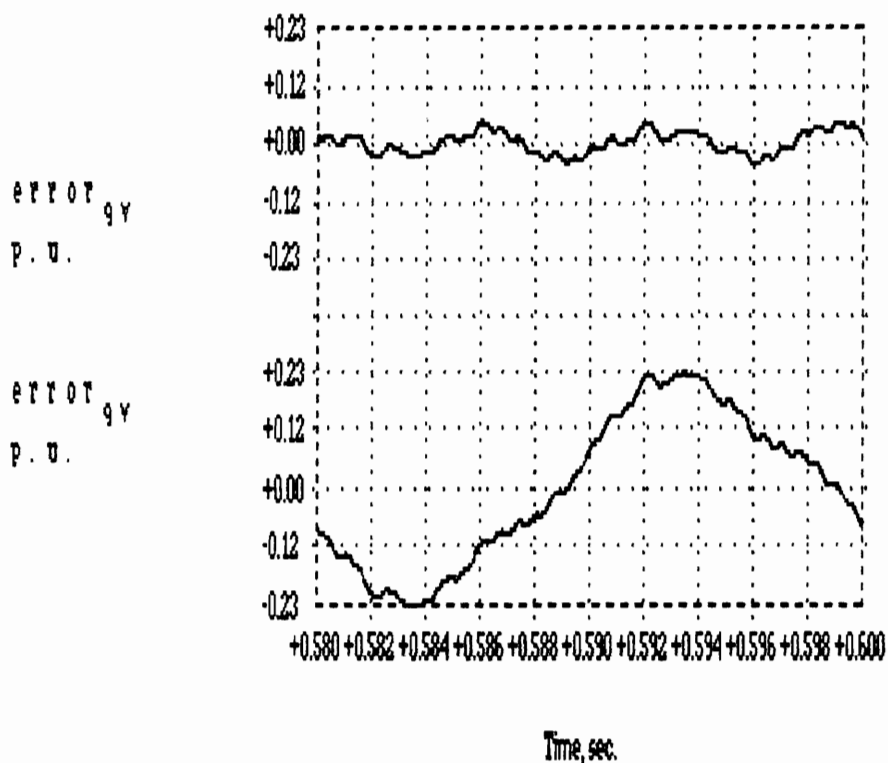


Figure 3.21B: Q-Axis Load Voltage Error without a Learning Process Delay (Top Trace) Verses the Q-Axis Load Voltage Error with One Controller Sampling Period Learning Process Delay (Bottom Trace), Balanced Resistive Load Simulation.

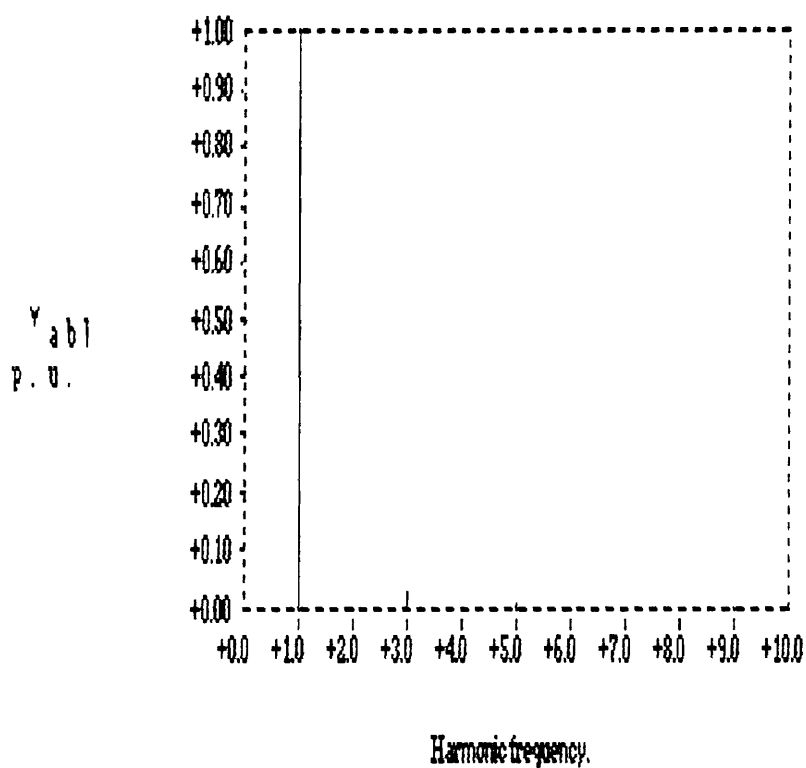


Figure 3.22: Q-Axis Load Voltage Spectral Information, Balanced Resistive Load Simulation, Modified Backpropagation Learning Process Enabled.

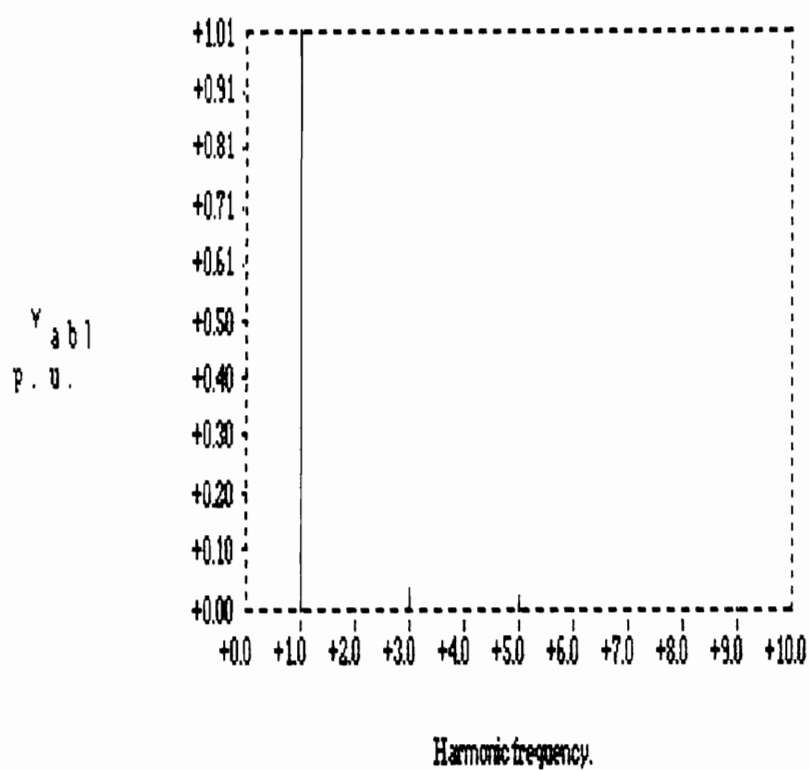


Figure 3.23: Q-Axis Load Voltage Spectral Information, Balanced Resistive Load Experiment, Modified Backpropagation Learning Process Enabled.

delay is involved correspond closely to the errors observed in the balanced resistive load experiment.

Given that the elimination of the learning process delay in the neurocontroller simulation reduced the observed load voltage errors further and eliminated the leading load voltage waveform phase-displacement, the same technique would affect the experimental controller's performance in the same fashion. The implementation of a controller program which executes the learning process first each controller sampling period would require faster processor technology, or equivalently, shorter forward propagation and learning process execution times.

With regard to the load voltage waveform fundamental and harmonic frequency characteristics, the experimental controller's performance corresponds closely to the simulation results.

3.3.2 Unbalanced Resistive Load Experiment Results

With the neurocontroller operating in an open-loop mode and an unbalanced resistive load connected, the following inverter/filter circuit measurements were made:

$V_{dc} = 200.0$ volts	$I_{dc} = 0.75$ amps	
$V_{abl} = 107.7$ volts	$V_{bcl} = 109.1$ volts	$V_{cal} = 99.4$ volts
$I_{al} = 0.6$ amps	$I_{bl} = 0.96$ amps.	

Corresponding measurements of the load voltage and sinusoidal reference waveforms are depicted in figures 3.24 and 3.25. In figure 3.26, the d- and q-axis load voltage error waveforms are depicted.

With the neurocontroller operating in a closed-loop mode, the modified backpropagation learning algorithm utilized, the learning-rate parameter set to 0.066, and an unbalanced resistive load connected, the following inverter/filter circuit measurements were made:

$$\begin{aligned} V_{dc} &= 200.0 \text{ volts} & I_{dc} &= 0.75 \text{ amps} \\ V_{abl} &= 104.8 \text{ volts} & V_{bcl} &= 104.2 \text{ volts} & V_{cal} &= 101.6 \text{ volts} \\ I_{al} &= 0.6 \text{ amps} & I_{bl} &= 0.96 \text{ amps.} \end{aligned}$$

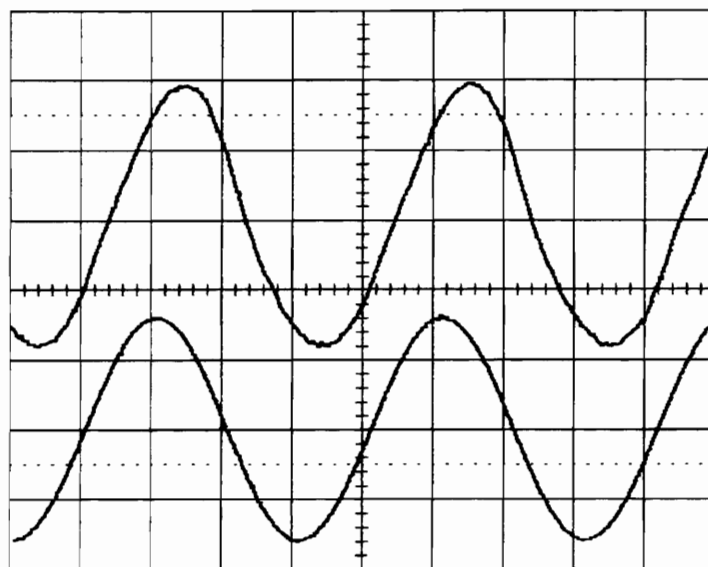
Corresponding measurements of the q-axis load voltage and sinusoidal reference waveforms are depicted in figure 3.27. In figure 3.28, the q-axis load voltage and reference waveforms, normalized with respect to the peak reference waveform amplitude, obtained from the unbalanced resistive load simulation are depicted. In figure 3.29, the normalized open- and closed-loop q-axis load voltage errors obtained from the unbalanced resistive load experiment is depicted. In figures 3.30 and 3.31, the normalized closed-loop load voltage waveform spectral characteristics obtained from the unbalanced resistive load simulation and experiment are depicted.

As in the case of the balanced load experiment, the load voltage errors are reduced with the learning process enabled. The load voltage waveform amplitude normalized with respect to the reference waveform peak amplitude is reduced from a 1.251 per unit value in open-loop to 0.998 per unit value in closed-loop. The load voltage waveform phase-

20-Sep-93
11:42:53

1:Eres(1)
5 ms
1.00 V

2:Eres(2)
5 ms
1.00 V



5 ms BWL

1 .1 V AC

2 .1 V AC **2** DC 0.00 V

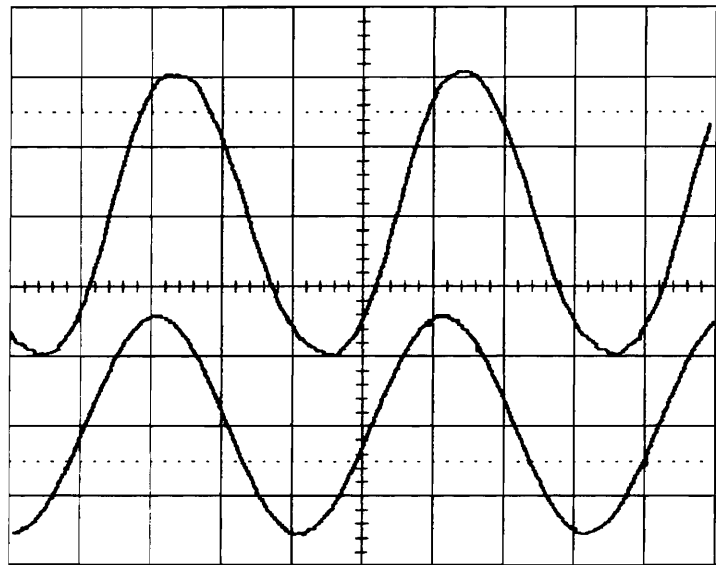
☐ STOPPED

Figure 3.24: D-Axis Load Voltage (Top Trace) Verses D-Axis Load Voltage Reference Waveform (Bottom Trace): Unbalanced Resistive Load, Controller Learning Process Disabled.

20-Sep-93
11:49:31

A:Eres(1)
5 ms
1.00 V

B:Eres(2)
5 ms
1.00 V



5 ms BWL

1 .1 V AC

2 .1 V AC 2 DC 0.00 V

□ STOPPED

Figure 3.25: Q-Axis Load Voltage (Top Trace) Verses Q-Axis Load Voltage Reference Waveform (Bottom Trace): Unbalanced Resistive Load, Controller Learning Process Disabled.

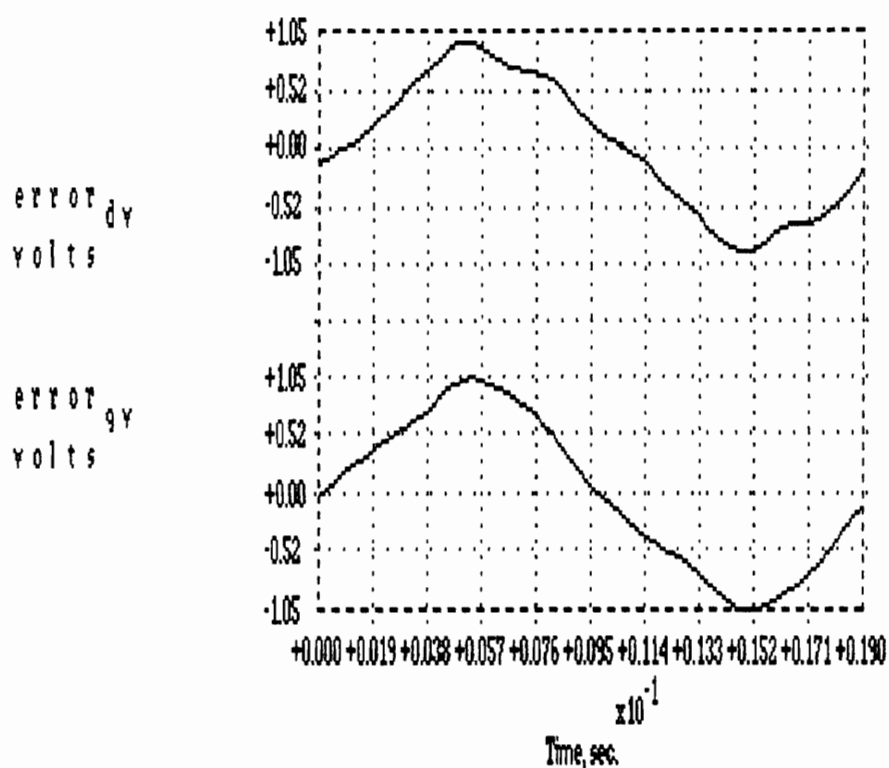
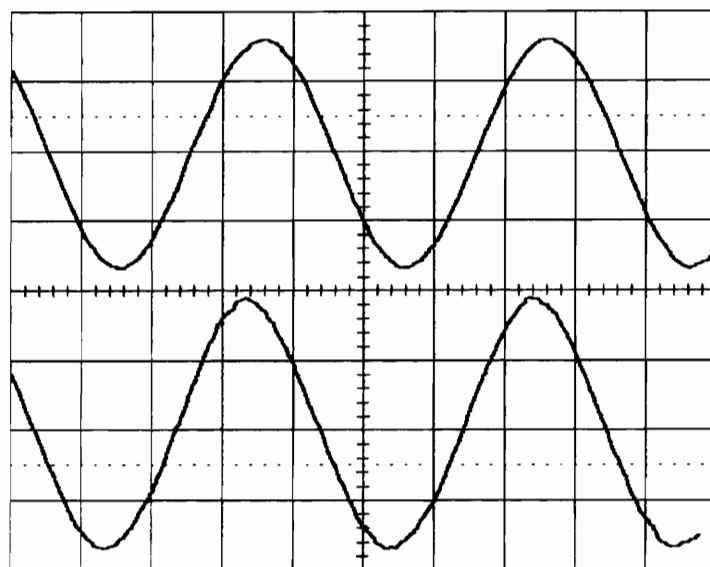


Figure 3.26: D-Axis Load Voltage Error (Top Trace) Verses Q-Axis Load Voltage Error (Bottom Trace), Unbalanced Resistive Load, Controller Operating in an Open-Loop Mode.

29-Oct-93
13:53:46

8: Eres(2)
5 ms
1.00 V

8: Eres(1)
5 ms
1.00 V



5 ms

1 .1 V AC

2 .1 V AC



2 AC 3.32 V

STOPPED

Figure 3.27: Q-Axis Load Voltage Reference (Top Trace) Verses the Q-Axis Load Voltage in the Unbalanced Resistive Load Experiment, Modified Backpropagation Learning Process Enabled.

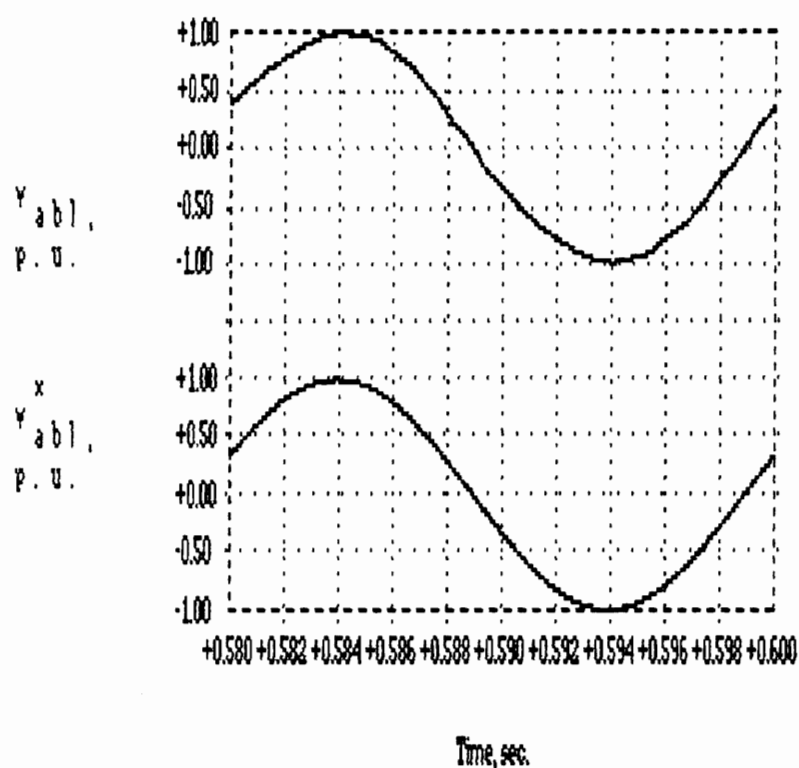


Figure 3.28: Q-Axis Load Voltage (Top Trace) Verses the Q-Axis Load Voltage Reference (Bottom Trace) in the Unbalanced Resistive Load Simulation, Modified Backpropagation Learning Process Enabled.

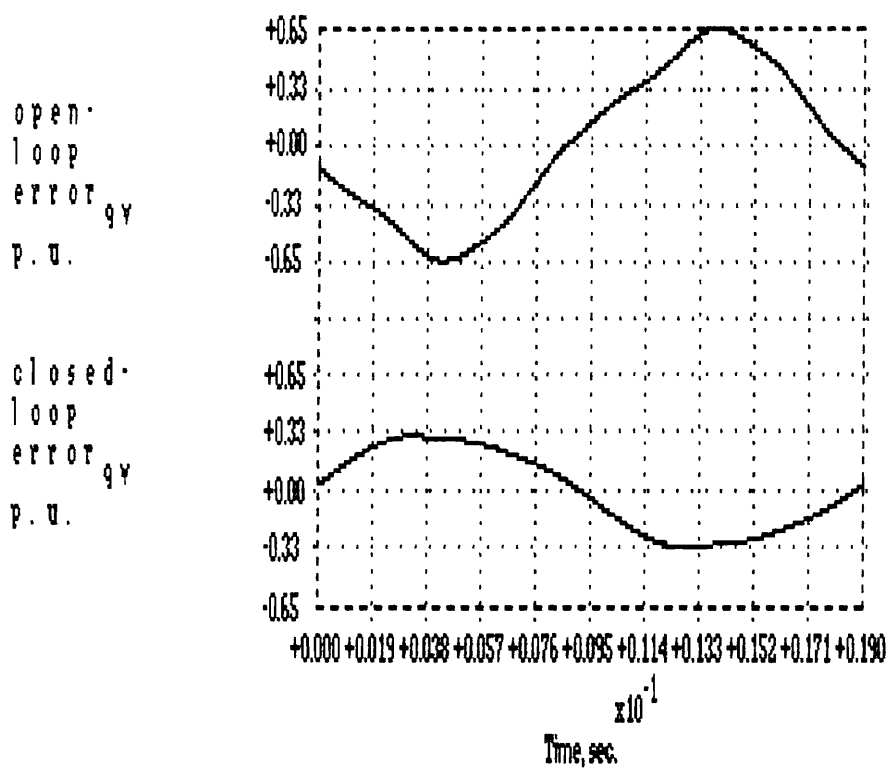


Figure 3.29: Open-Loop Normalized Q-Axis Load Voltage Error (Top Trace) Verses the Closed-Loop Normalized Q-Axis Load Voltage Error (Bottom Trace), Unbalanced Resistive Load Experiment.

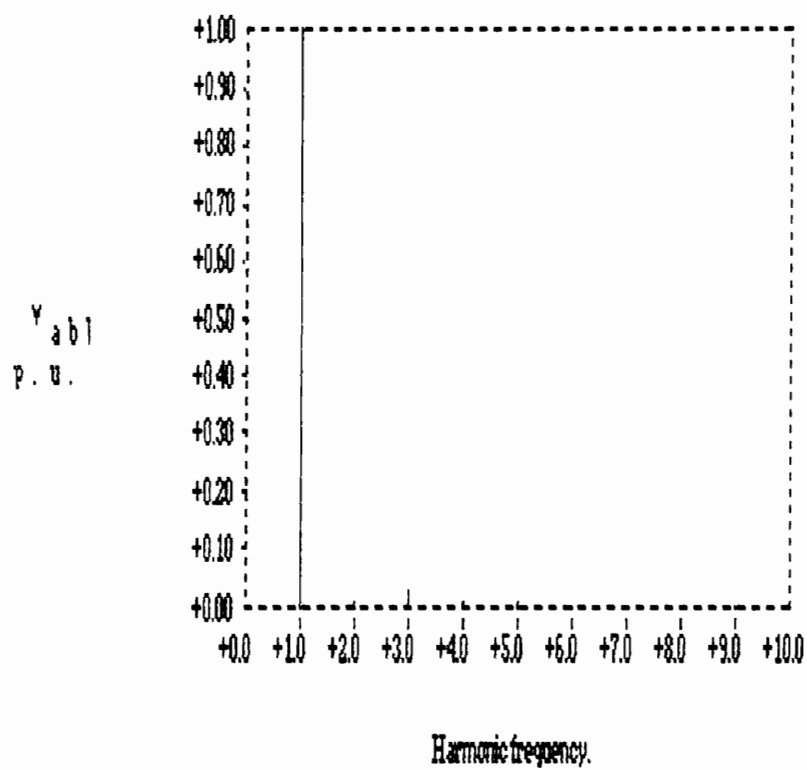


Figure 3.30: Q-Axis Load Voltage Spectral Information, Unbalanced Resistive Load Simulation, Modified Backpropagation Learning Process Enabled.

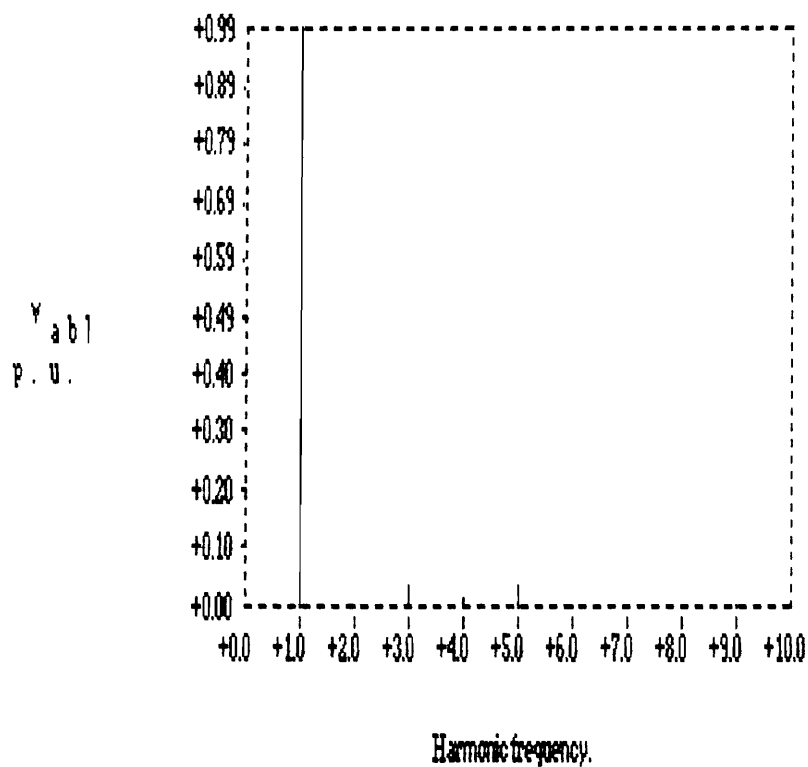


Figure 3.31: Q-Axis Load Voltage Spectral Information, Unbalanced Resistive Load Experiment, Modified Backpropagation Learning Process Enabled.

displacement in relation to the reference waveform changed from a lagging 36.1° degrees in open-loop to a leading 18.6° in closed-loop. As in the case of the balanced load experiment, the leading load voltage phase displacement is created by the execution of the controller learning process after the forward propagation process in each sampling period. The closed-loop load voltage errors and phase-displacement value are nearly equivalent to the values observed in the balanced load experiment. The load voltage and reference waveforms depicted in figure 3.28 are obtained from a simulation operating without a learning process delay, and so, a leading load voltage waveform phase-displacement is not evident. Load voltage waveform spectral characteristics obtained from the simulation and experiment are similar.

The unbalanced resistive load experiment tests the controller's ability to generate asymmetrical three-phase inverter output voltage waveforms. Unbalanced loads create imbalances in the load voltage waveform amplitude and phase-displacement. In response to closed-loop control, the modulation commands $\{v_{di}^*(t), v_{qi}^*(t)\}$ generated by the controller no longer follow a circular trajectory typical in balanced load conditions. The instantaneous load voltage error results demonstrate the controller's ability to follow a non-circular trajectory generated by an unbalanced load condition.

3.3.3 Nonlinear Load Experiment Results

With the neurocontroller operating in an open-loop mode and a nonlinear load connected, the following inverter/filter circuit measurements were made:

$$\begin{aligned} V_{dc} &= 200.0 \text{ volts} & I_{dc} &= 0.5 \text{ amps} \\ V_{abl} &= 107.5 \text{ volts} & V_{bcl} &= 104.3 \text{ volts} & V_{cal} &= 109.4 \text{ volts} \\ I_{al} &= 0.65 \text{ amps} & I_{bl} &= 0.7 \text{ amps.} \end{aligned}$$

Corresponding measurements of the load voltage and sinusoidal reference waveforms are depicted in figures 3.32 and 3.33. In figure 3.34, the d- and q-axis load voltage error waveforms are depicted. In figure 3.35, the open-loop load voltage waveform spectral information, normalized with respect to the reference waveform peak amplitude, is depicted.

With the neurocontroller operating in a closed-loop mode, the modified backpropagation learning algorithm utilized, the learning-rate parameter set to 0.066, and a nonlinear load connected, the following inverter/filter circuit measurements were made:

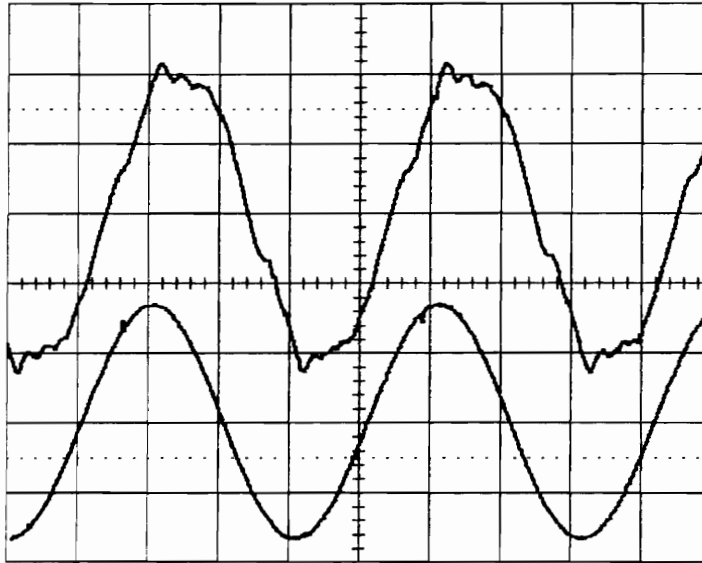
$$\begin{aligned} V_{dc} &= 200.0 \text{ volts} & I_{dc} &= 0.5 \text{ amps} \\ V_{abl} &= 106.5 \text{ volts} & V_{bcl} &= 109.6 \text{ volts} & V_{cal} &= 104.5 \text{ volts} \\ I_{al} &= 0.7 \text{ amps} & I_{bl} &= 0.7 \text{ amps.} \end{aligned}$$

Corresponding measurements of the load voltage and sinusoidal reference waveforms are depicted in figure 3.36. In figure 3.37, the closed-loop q-axis load voltage and reference waveforms, normalized with respect to the reference waveform peak amplitude, obtained from the nonlinear load simulation are depicted.

20-Sep-93
13:16:45

A:Eres(1)
5 ms
1.00 V

B:Eres(2)
5 ms
1.00 V



5 ms BWL

1 .1 V AC

2 .1 V AC



2 DC 0.00 V

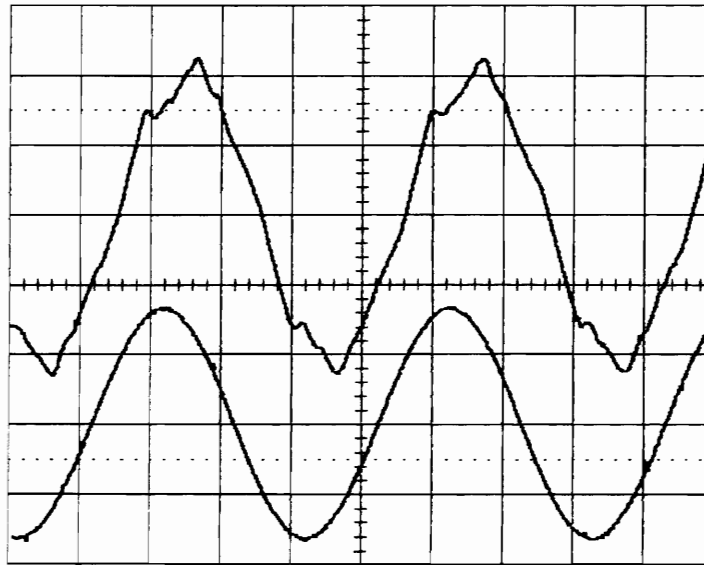
☐ STOPPED

Figure 3.32: D-Axis Load Voltage (Top Trace) Verses D-Axis Load Voltage Reference Waveform (Bottom Trace): Nonlinear Load, Controller Learning Process Disabled.

20-Sep-93
13:21:50

A:Eres(1)
5 ms
1.00 V

B:Eres(2)
5 ms
1.00 V



5 ms BWL

1 .1 V AC

2 .1 V AC



2 DC 0.00 V

□ STOPPED

Figure 3.33: Q-Axis Load Voltage (Top Trace) Verses Q-Axis Load Voltage Reference Waveform (Bottom Trace): Nonlinear Load, Controller Learning Process Disabled.

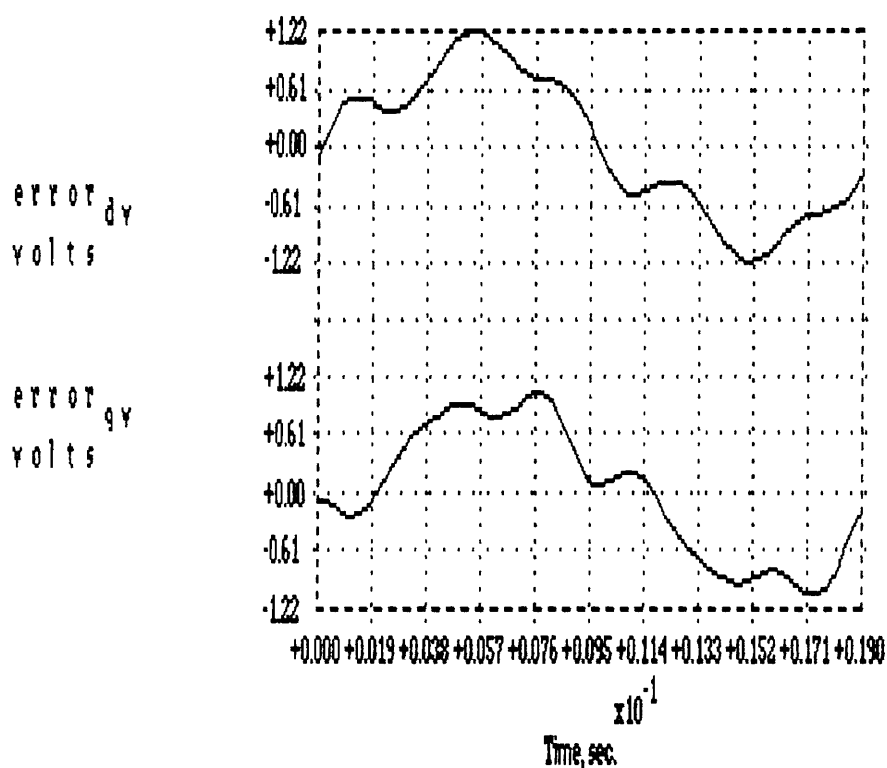


Figure 3.34: D-Axis Load Voltage Error (Top Trace) Verses Q-Axis Load Voltage Error (Bottom Trace), Nonlinear Load, Controller Operating in an Open-Loop Mode.

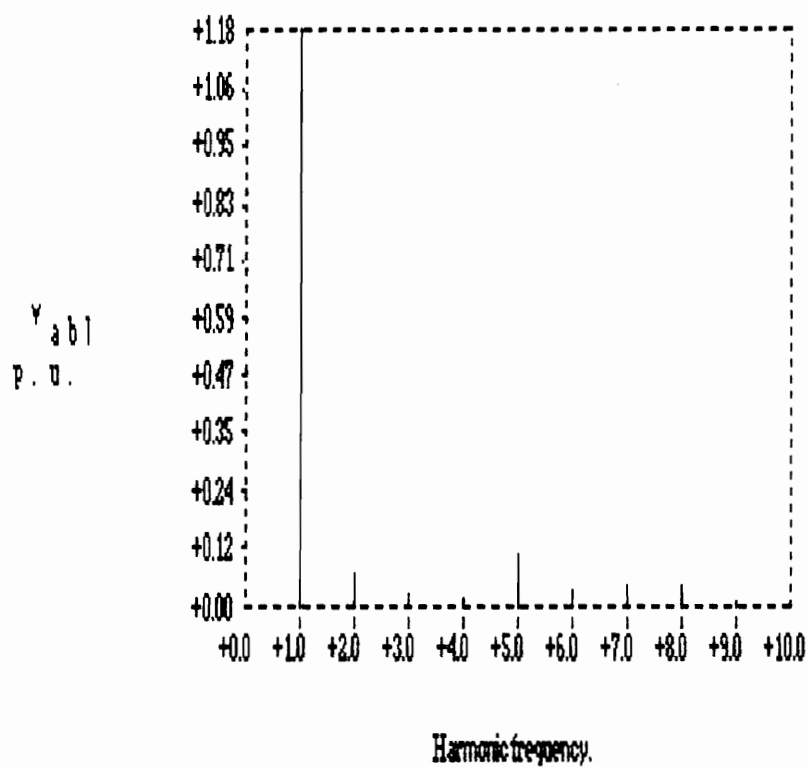
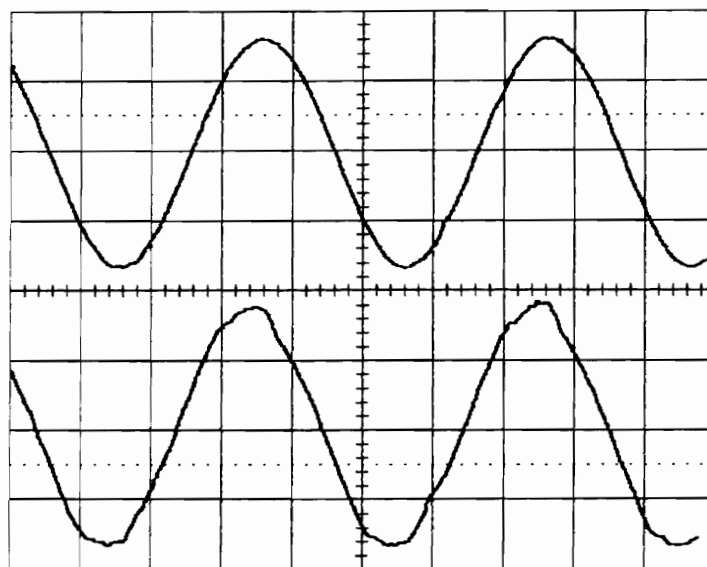


Figure 3.35: Q-Axis Load Voltage Spectral Information, Nonlinear Load Experiment, Controller Operating in an Open-Loop Mode.

29-Oct-93
14:02:22

B:Eres(2)
5 ms
1.00 V

A:Eres(1)
5 ms
1.00 V



5 ms

1 .1 V AC

2 .1 V AC



2 AC 3.32 V

□ STOPPED

Figure 3.36: Q-Axis Load Voltage Reference (Top Trace) Verses the Q-Axis Load Voltage in the Nonlinear Load Experiment, Controller Learning Process Enabled and the Modified Backpropagation Learning Strategy Employed.

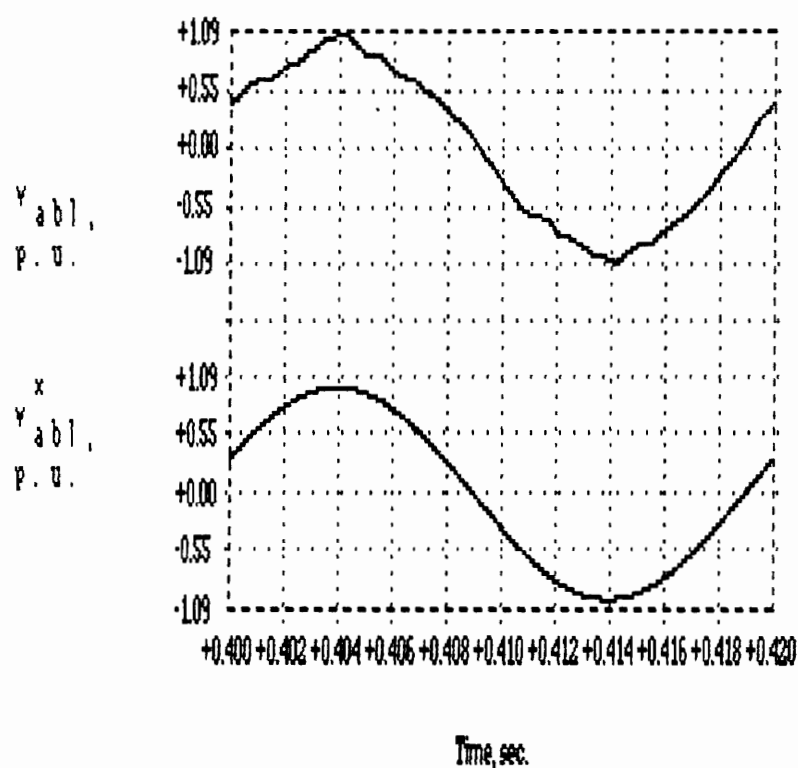


Figure 3.37: Q-Axis Load Voltage (Top Trace) Verses the Q-Axis Load Voltage Reference (Bottom Trace) in the Nonlinear Load Simulation, Modified Backpropagation Learning Process Enabled.

In figure 3.38, the open- and closed-loop q-axis load voltage errors, normalized with respect to the reference waveform peak amplitude, are depicted. In figures 3.39 and 3.40, normalized closed-loop load voltage waveform spectral characteristics obtained from the nonlinear load simulation and experiment are depicted.

With a nonlinear load connected, the load voltage waveform harmonic distortion is an important point of evaluation. A tabulation of the normalized open- and closed-loop load voltage spectral characteristics, depicted figures 3.35 and 3.40 respectively, is given in table 3.5. In the closed-loop mode, a significant reduction of the 5th harmonic is observed. Reductions in the 2nd, 6th, 7th, 8th, and 9th harmonics are observed as well. The closed-loop load voltage THD is significantly reduced. Load voltage waveform spectral characteristics obtained from the nonlinear load simulation conform closely to the experimental results.

In the closed-loop mode, the normalized fundamental load voltage amplitude is reduced from its 1.182 per unit value in the open-loop mode to a 0.989 per unit value. As in the case of the balanced and unbalanced resistive load experiment, the closed-loop fundamental load voltage waveform has a leading phase-displacement, 17.5°, in relation to the reference waveform. This is the result of the execution of the forward propagation process before the learning process in each controller sampling period. The closed-loop load voltage error waveform depicted in figure 3.38 is similar to the corresponding waveforms in the obtained in the resistive load experiments.

Table 3.5: Comparison of the Open- and Closed-Loop Load Voltage Waveform Spectral Characteristics, Nonlinear Load Experiment. Load Voltage Harmonic Components Normalized with Respect to the Reference Waveform Peak Amplitude.

HARMONICS	OPEN-LOOP P.U.	CLOSED-LOOP P.U.
1	1.182	0.989
2	0.063	0.029
3	0.021	0.036
4	0.013	0.016
5	0.105	0.033
6	0.037	0.031
7	0.040	0.018
8	0.043	0.015
9	0.010	0.009
10	0.011	0.018
THD	0.110	0.005

To conclude the discussion of nonlinear load experiment and simulation results using a neurocontroller without hidden layers, a comparison to the repetitive controller given the same load model is made. In figure 3.41, the normalized closed-loop load voltage errors obtained from the nonlinear load simulations using the repetitive controller and the neurocontroller without hidden layers are depicted. In the neurocontroller simulation, no learning process delay was included. In figure 3.42, the load voltage waveform spectral characteristics, normalized with respect to the reference waveform peak amplitude, obtained from the nonlinear load simulation using the repetitive controller is depicted. As in the case of the neurocontroller with a hidden layer, the steady-state performance of the neurocontroller without a hidden layer and using the modified backpropagation learning algorithm with no learning process delay is at least as good as the performance of the repetitive controller. With regard to the fundamental load voltage amplitude and waveform harmonic distortion, this observation can be applied to the performance of the experimental neurocontroller.

The closed-loop leading load voltage waveform phase-displacement observed in the neurocontroller experiments is the primary deviation from the performance observed in the repetitive controller simulation. The cause of this deviation, a controller sampling period learning process delay has been established, and its resolution verified through neurocontroller simulations.

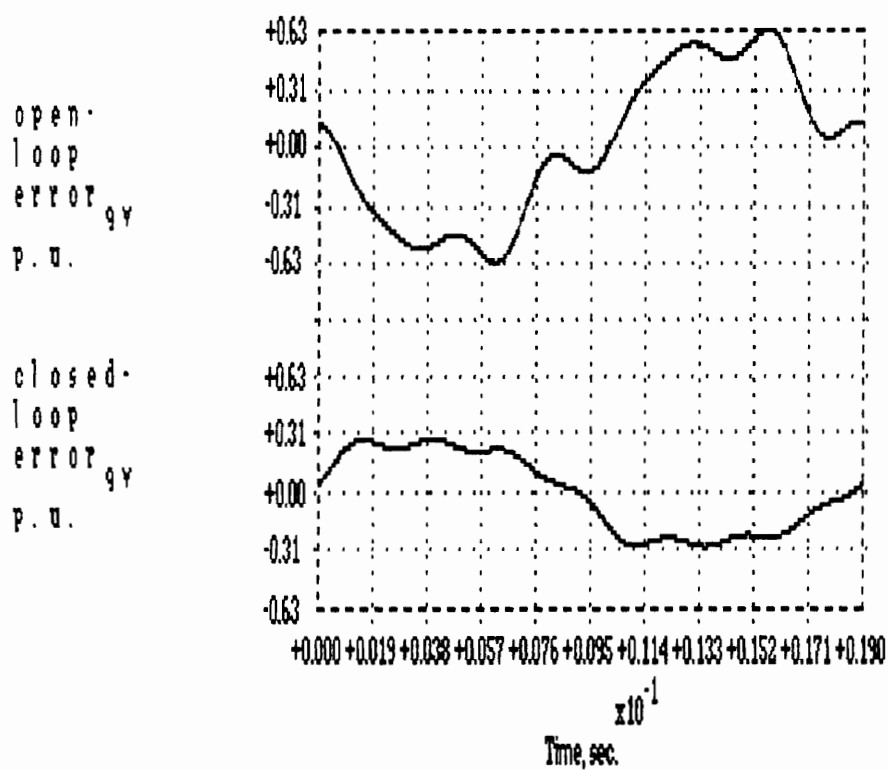


Figure 3.38: Open-Loop Normalized Q-Axis Load Voltage Error (Top Trace) Verses the Closed-Loop Normalized Q-Axis Load Voltage Error (Bottom Trace), Nonlinear Load Experiment.

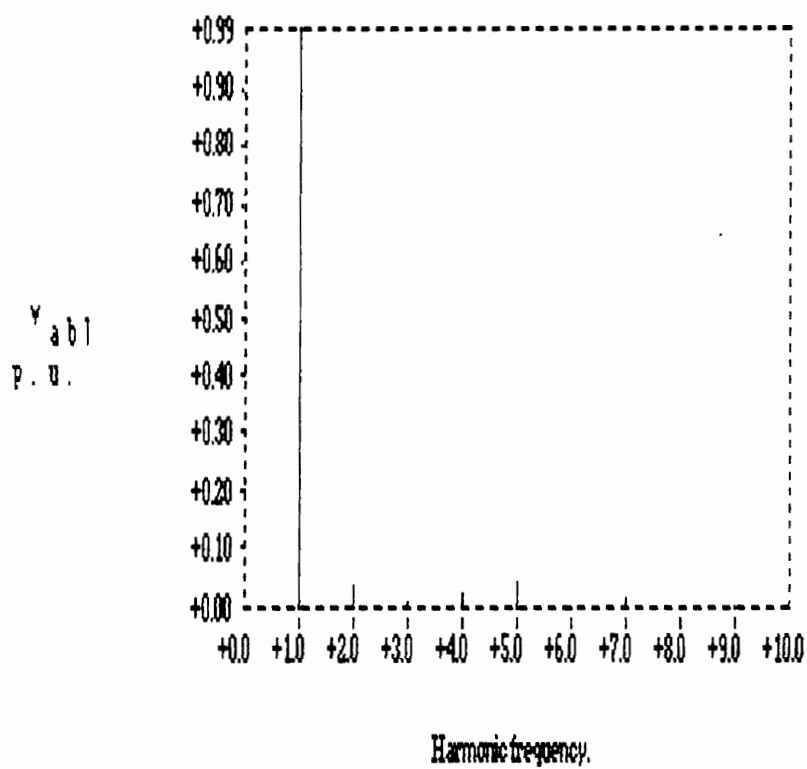


Figure 3.39: Q-Axis Load Voltage Spectral Information, Nonlinear Load Simulation, Modified Backpropagation Learning Process Enabled.

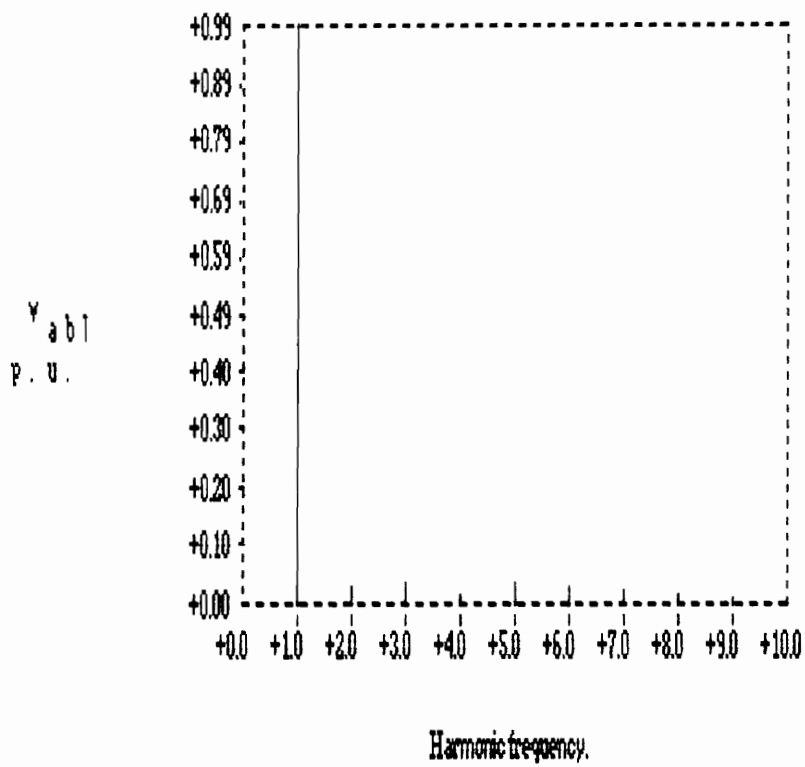


Figure 3.40: Q-Axis Load Voltage Spectral Information, Nonlinear Load Experiment, Modified Backpropagation Learning Process Enabled.

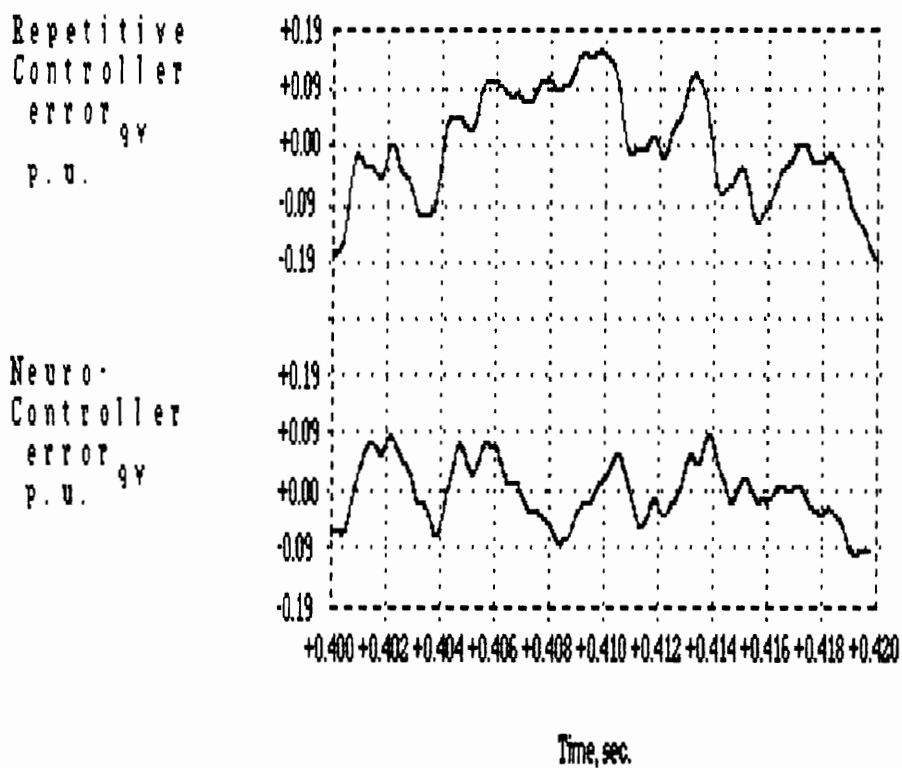


Figure 3.41: Closed-Loop Normalized Load Voltage Errors, Repetitive Controller (Top Trace) and Neurocontroller (Bottom Trace), Nonlinear Load Simulation.

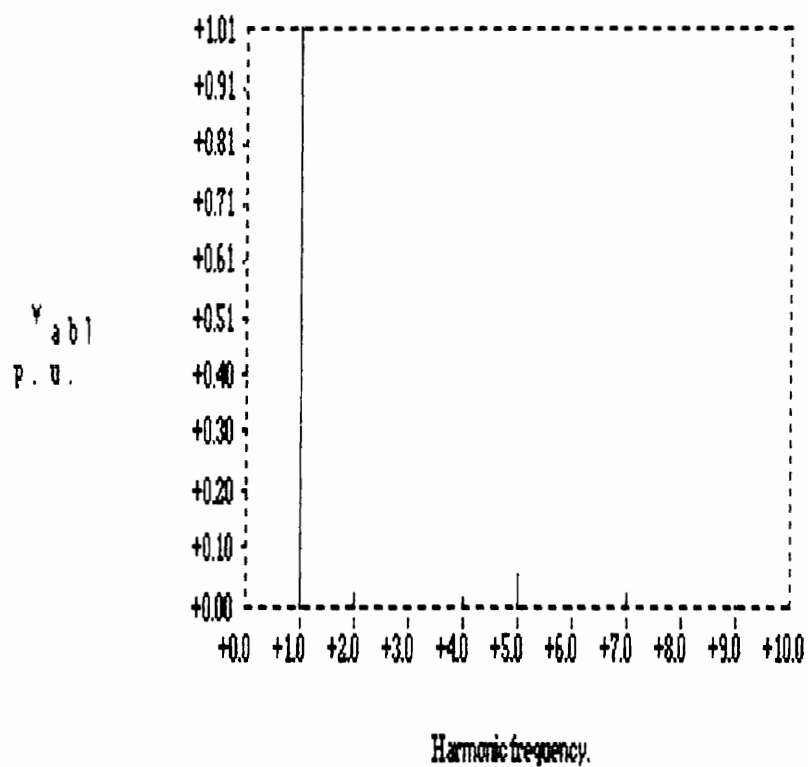


Figure 3.42: Q-Axis Load Voltage Spectral Information, Nonlinear Load Simulation, Repetitive Controller Utilized.

3.3.4 Dynamic Control Experiments

Figure 3.43 depicts the measurement of instantaneous load voltage errors with a balanced passive load and variable inverter DC supply voltage (variation from 200 volts DC to 150 volts). The maximum instantaneous errors decrease in the range between 200 and 150 volts DC. This improvement in control activity is due to the superior inverter output waveform spectral characteristics at higher modulation index values (magnitude reduction of the low-order harmonic frequencies present in the inverter output waveforms). This improvement does not continue when the DC supply voltage is 150 volts (theoretically the maximum limit of linear voltage control). Inverter switching distortion effects reduce the maximum limit of linear voltage control, and so, inverter output voltage control given a 150 volt DC supply value is now in a nonlinear mode. The instantaneous load voltage errors are slightly higher at this point compared to the errors measured at 200 volts DC.

The mechanism in this neurocontroller implementation which regulates the load voltages given DC power supply variations is the learning process modification of neural network weights connecting the load voltage commands $\{v_{dl}^*, v_{ql}^*\}$ to the network outputs (inverter modulation commands). The response of the controller with regard to DC supply voltage variations is dependent on the inverter switching frequency, the learning-rate parameter, and the output filter response characteristics.

Application of a load is a test of the controller's transient response characteristics. Depicted in figure 3.44 are the measurements of the instantaneous load voltage error and the load current waveform at the time of the 100% block load increase. The duration of the controller's forced response (based on the load voltage error waveform) appears to be on the order of three cycles. The maximum instantaneous load voltage errors during the forced response is slightly larger than twice the maximum errors prior to the step load. With the step application of load, the inverter DC power supply voltage falls from 200 volts DC to 180 volts DC (a soft inverter DC power supply).

In the final neurocontroller version, load current rate-of-change feedback $\{di_{d1}(t)/dt, di_{q1}(t)/dt\}$ are neural network inputs propagated forward in the network along with the load voltage commands to generate modulation commands $\{v_{di}^*, v_{qi}^*\}$. The sampled load current rate-of-change is derived from the difference equations

$$i_{d1}(k) - i_{d1}(k - 1) \quad (3.5)$$

$$i_{q1}(k) - i_{q1}(k - 1). \quad (3.6)$$

From figure 3.44, it is observed that the controller's response (based on the error waveform) begins prior to a change in the load current (approximately one cycle in advance). In this interval of time, the mechanism which regulates the load voltages is the learning process modification of network weights connecting the load voltage commands $\{v_{d1}^*, v_{q1}^*\}$. The controller's response in this interval is determined by the output filter response characteristics and the controller's learning-rate parameter (which is set to 0.066).

Once changes in the load current are realized, then corrections in the modulation command values are made directly by the load current feedback. The controller response during and after the rise of load current is the result of load current feedback, the output filter response and the modification of network weights by the learning process. After the load current increase begins, the load voltage errors return to their instantaneous values prior to load application after an interval of one cycle.

Compared to reported performance of dead-beat control techniques (for example reference [2]), the response time of the neurocontroller in the step load experiment appears to be comparatively slow. On the other hand, the ideal conditions on which dead-beat control techniques achieve this fast response to sudden load changes are not necessarily present in an actual step load event. Two characteristics of this particular experiment which represent a departure from an ideal step load event are described as follows:

- i.) The observed increase of load current is delayed one cycle in reference to the observed changes in the instantaneous load voltage error waveform. The additional load circuitry connected during the step load increase consisted of conventional electric light bulbs which have a cold filament resistance that is somewhat lower than the filament resistance in steady-state operation. As a consequence, the actual initial step load application was much greater than 100%. The load current increase was delayed by the output filter's response to the low initial load resistance. Given

the output filter capacitance $C_f = 45 \mu\text{F}$ (70.7Ω impedance at 50 Hertz), the filter inductance $L_f = 10 \text{ mH}$, and the initial (cold filament) load resistance of 10Ω , it is evident why the increase in load current was not immediate. The observed load current increase corresponds to the transition of the additional load filament resistance from its initial low value to its steady-state value (approximately 108Ω).

- ii.) The inverter DC supply voltage dropped from 200 volts DC prior to the load step to 180 volts DC after the load current increase.

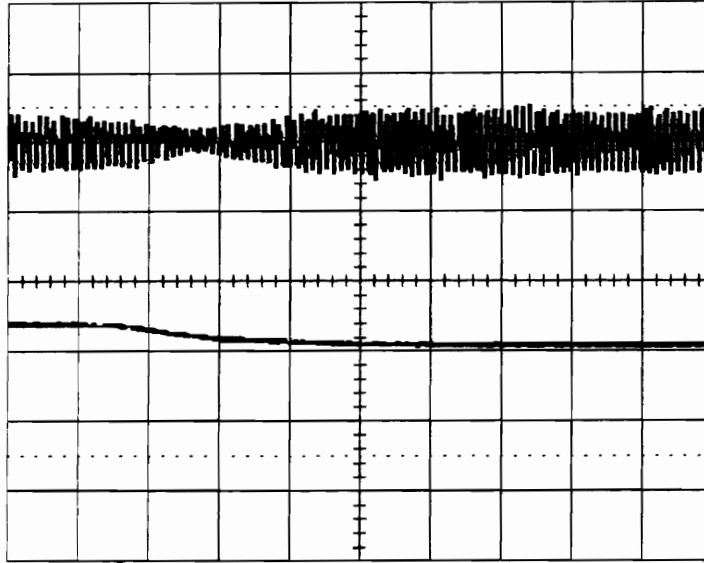
Load step increases involving initially low surge load impedances and the inverter DC supply voltage variation due to the eventual load current increase force inverter output waveform control into a nonlinear mode. The operation of linear control techniques on which dead beat control techniques are based is not well-defined in the case where inverter fundamental voltage control is in a nonlinear mode.

The observed slower neurocontroller response observed in this load step experiment is a result of the neurocontroller's learning process, where the load current increase delay and the inverter DC power supply voltage variation create new controller learning situations.

24-Sep-93
14:57:59

B:Eres(2)
.2 s
5.0 V

A:Eres(1)
.2 s
0.50 V



.2 s BWL
1 50 mV AC

2 .5 V DC 1 AC 1.68 V

□ STOPPED

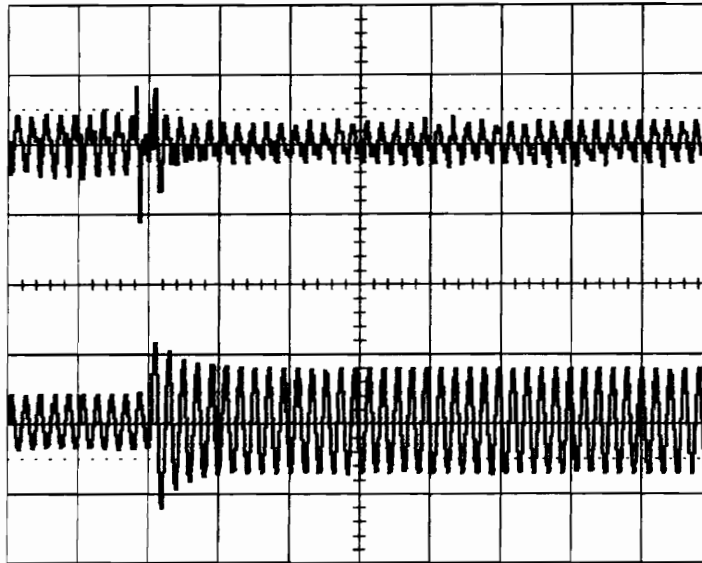
Figure 3.43: Load Voltage Instantaneous Error (Top Trace) Verses the DC Link Voltage (Bottom Trace): DC Power Supply Voltage Reduction from 200 VDC to 150 VDC, Controller Learning Process is Enabled.

24-Sep-93

13:53:13

B:Eres(2)
.1 s
5.0 V

A:Eres(1)
.1 s
0.50 V



.1 s BWL

1 50 mV AC

2 .5 V AC

1 AC 1.68 V

□ STOPPED

Figure 3.44: Load Voltage Instantaneous Error (Top Trace) Verses the Load Current (Bottom Trace): 100% Load Increase, Controller Learning Process Enabled.

IV. THESIS SUMMARY

4.1 Overview of Neurocontroller Application

An adaptive intelligent inverter control technique using a single-layer neural network and a modified backpropagation learning process has been described. A variation of the direct adaptive control configuration depicted in figure 2.5 was employed. Computer simulations and experiments using balanced passive, unbalanced passive, and nonlinear three-phase loads have shown that the steady-state performance of this control technique is similar to the performance of the repetitive controller [3], a conventional linear control technique devised to eliminate steady-state load voltage waveform errors.

The selection of sampled neurocontroller inputs $\underline{v}_{dl}^*(k + 1)$, $\underline{v}_{ql}^*(k + 1)$, $\underline{i}_{dl}^{(q)}(k) - \underline{i}_{dl}^{(q)}(k - 1)$, and $\underline{i}_{ql}^{(q)}(k) - \underline{i}_{ql}^{(q)}(k - 1)$ corresponds to the independent variables in the equations describing the output filter circuit, depicted in figure 1.2, inverse dynamics where k is a positive integer, $\{\underline{v}_{dl}^*, \underline{v}_{ql}^*\}$ are the logic-level load voltage commands in the d-q frame, and $\{\underline{i}_{dl}^{(q)}, \underline{i}_{ql}^{(q)}\}$ are the logic-level load current feedback signals in the d-q frame phase shifted 90°. The neurocontroller learning process based on minimizing the load voltage errors

$$\underline{\epsilon}_{dv}(k) = \underline{v}_{dl}^*(k) - \underline{v}_{dl}^{(q)}(k) \text{ and} \quad (4.1a)$$

$$\underline{\epsilon}_{qv}(k) = \underline{v}_{ql}^*(k) - \underline{v}_{ql}^{(q)}(k), \quad (4.1b)$$

where $\{\underline{v}_{dl}^{(q)}, \underline{v}_{ql}^{(q)}\}$ are the logic-level load voltage feedback signals in the d-q frame phase shifted 90° , generates a mapping between the neurocontroller inputs and outputs $\{\underline{v}_{di}^*(k), \underline{v}_{qi}^*(k)\}$, space vector PWM process modulation commands, in steady-state which approximates the inverse dynamics of the inverter/filter circuit. Once the neurocontroller learning process is completed, the neurocontroller performance should be similar to the performance of dead-beat inverter control techniques [2][3][4][5] which are designed based on knowledge of the output filter circuit state transition equations. In addition to the observed reduction of steady-state load voltage errors in the computer simulations and experiments, the response of the neurocontroller in the block load experiments conforms to the anticipated response of an inverter controller utilizing the output filter inverse dynamics for the purpose of load voltage waveform compensation.

The neurocontroller learning process is synchronized to the forward propagation and space vector PWM processes where neurocontroller connection weight modification occurs at $1/2f_s$ intervals where f_s is the inverter switching frequency. Initial neurocontroller learning processes where learning-rate parameter values were set to the maximum values without causing instability required 10 to 15 fundamental frequency cycles to reach steady-state from an inverter/filter/load system cold startup. Given the use of inverter switching frequencies in the range of 540 to 1260 Hertz and the response of the corresponding output filter circuit designs, the interval of time required for the neurocontroller learning process to reach steady-state compares favorably to the

corresponding response of an inverter/filter system utilizing a conventional control technique. The relative brevity of the inverter neurocontroller learning process is attributed to the selection of neurocontroller inputs and the fact that the structure of the plant or output filter circuit inverse dynamics is known.

The characteristics of inverter neurocontroller operation in a direct adaptive control configuration using on-line training are different from a neural network application in a pattern recognition problem where network training is performed off-line. In the pattern recognition problem, the network training data utilized describes a stationary process. Changes in the network weights due to the learning process are made with the objective of minimizing a cost function which is independent of time and changes in the neural network forward transfer characteristics. This is not the case in the operation of an inverter neurocontroller in a closed-loop system utilizing on-line training.

A problem encountered with the use of an inverter neurocontroller utilizing on-line training in a closed-loop system is the inability of the controller to reach the desired load voltage trajectory in steady-state. This inability to reach the desired trajectory results in system instability in the form of limit cycle oscillation. Stability problems associated with the application of the neurocontroller technique are created by two mechanisms:

- i.) Subcycle control of inverter output waveform generation resulting in asymmetrical three-phase inverter output waveforms, and

ii.) The mismatch between inverter neurocontroller forward transfer characteristics and the actual inverter/filter circuit inverse dynamics during the neurocontroller learning process.

Asymmetrical three-phase waveforms contain triplen harmonics and even harmonics in the case of extremely asymmetrical waveforms. Given the relatively low inverter switching frequencies used in the computer simulations and experiments, the pole in the corresponding output filter circuit response is in the proximity of the subharmonics generated by asymmetrical inverter output voltage waveforms. For example in the neurocontroller experiments where the inverter modulation frequency is 50 Hertz and the switching frequency is 750 Hertz, the output filter circuit ($L_f = 10$ mH, $C_f = 45$ μ F) pole frequency is at 137 Hertz.

The limit cycle oscillation and steady-state fundamental load voltage errors observed in experiments and simulations using the single-layer neurocontroller employing the conventional backpropagation learning process is attributed to the generation of asymmetrical inverter output voltage waveforms. The single-layer neurocontroller d- and q-axis forward transfer functions described in the equations

$$\begin{aligned} \underline{v}_{di}^*(k) = & w_{11}(k) \cdot \underline{v}_{d1}^*(k) + w_{12}(k) \cdot \underline{v}_{q1}^*(k) + \\ & w_{13}(k) \cdot [\underline{i}_{d1}^{(q)}(k) - \underline{i}_{d1}^{(q)}(k-1)] + \\ & w_{14}(k) \cdot [\underline{i}_{q1}^{(q)}(k) - \underline{i}_{q1}^{(q)}(k-1)] \text{ and} \end{aligned} \quad (4.2a)$$

$$\begin{aligned} \underline{v}_{qi}^*(k) = & w_{21}(k) \cdot \underline{v}_{d1}^*(k) + w_{22}(k) \cdot \underline{v}_{q1}^*(k) + \\ & w_{23}(k) \cdot [\underline{i}_{d1}^{(q)}(k) - \underline{i}_{d1}^{(q)}(k-1)] + \\ & w_{24}(k) \cdot [\underline{i}_{q1}^{(q)}(k) - \underline{i}_{q1}^{(q)}(k-1)], \end{aligned} \quad (4.2b)$$

are decoupled. Changes in d- and q-axis components of the inverter modulation command trajectory $\{\underline{v}_{di}^*(k), \underline{v}_{qi}^*(k)\}$ as the result of the backpropagation process are decoupled as well. The events where the conventional backpropagation process converges to a local minimum are defined by the conditions

$$d\underline{\epsilon}_{dv}(t)/dt = 0 \text{ and } |\underline{\epsilon}_{dv}(t)| > 0 \quad (4.3a)$$

$$d\underline{\epsilon}_{qv}(t)/dt = 0 \text{ and } |\underline{\epsilon}_{qv}(t)| > 0. \quad (4.3b)$$

Asymmetrical three-phase inverter output voltage waveform generation is a result of the occurrence of either condition defined in (4.2a) or (4.2b) but not both simultaneously. Load current positive feedback sustains low-order subharmonic generation in the load voltage waveforms due to the proximity of the output filter pole frequency to the load voltage waveform second and third harmonics.

A modified backpropagation learning algorithm was devised which eliminated the limit cycle oscillation and steady-state fundamental load voltage errors observed with the use of the single-layer neurocontroller. This new learning algorithm is based on the principle that direction changes in the inverter modulation command trajectory $\{\underline{v}_{di}^*(k), \underline{v}_{qi}^*(k)\}$ due to the learning process are determined only by the direction suggested by the load voltage errors $\{\underline{\epsilon}_{dv}(k), \underline{\epsilon}_{qv}(k)\}$. Direction changes in the trajectory $\{\underline{v}_{di}^*(k), \underline{v}_{qi}^*(k)\}$ are influenced by negative feedback only. Subsequently, the minimization of load voltage errors $\{\underline{\epsilon}_{dv}(k), \underline{\epsilon}_{qv}(k)\}$ forces the convergence of the load voltage trajectory $\{\underline{v}_{d1}(k), \underline{v}_{q1}(k)\}$ to the desired trajectory $\{\underline{v}_{d1}^*(k), \underline{v}_{q1}^*(k)\}$.

In contrast, direction changes in the trajectory $\{\underline{v}_{di}^*(k), \underline{v}_{qi}^*(k)\}$ with the use of the conventional backpropagation algorithm were influenced by the polarity of the load current positive feedback as well. As observed in the simulations and experiments using the single-layer neurocontroller employing the conventional backpropagation algorithm, the influence of load current positive feedback in conjunction with the location of the pole associated with the output filter response prevailed over the negative feedback process typically associated with neurocontroller learning.

The presence of load current positive feedback is also a factor in the selection of the neurocontroller learning-rate parameter value. Unlike conventional control techniques, the neurocontroller forward transfer characteristics are not fixed, but vary with time in response to the sampled load voltage errors $\{\underline{\epsilon}_{dv}(k), \underline{\epsilon}_{qv}(k)\}$. In the case of conventional dead-beat controllers, a mismatch in the controller's forward transfer characteristics and the output filter circuit inverse dynamics is known to create steady-state load voltage errors and may cause the controller/inverter/filter system to become unstable.

This condition is common with the use of the neurocontroller particularly during the initial neurocontroller learning process. Learning-rate parameter values were manually set based on the observed steady-state performance of the neurocontroller (particularly with regard to system stability). Stable neurocontroller operation was typically achieved by lowering the learning-rate parameter value. An exception to this statement is demonstrated by the performance of the

multi-layer neurocontroller (single hidden layer with six nodes) in the balanced passive load simulation. A 1.0 learning-rate parameter was successfully utilized in this application as compared to 0.1 and 0.066 values in the nonlinear load simulations and the experiments with the single-layer neurocontroller. In a more complex neurocontroller structure (the use of hidden units), the rate of convergence associated with the learning process is inherently lower which enhances the stability of the system after the neurocontroller learning process reaches steady-state. Additionally, the load current harmonics generated by nonlinear loads promote system instability due to the presence of load current positive feedback in the neurocontroller learning process.

4.2 Contributions and Conclusions

The contributions made in this thesis are:

- i.) A dedicated study of the application of the neurocontroller technique utilizing on-line training to the control of load voltage waveforms generated by an uninterruptible power supply (UPS),
- ii.) The identification of the controller instability problem associated with the use of the conventional backpropagation algorithm,
- iii.) The development of a modified backpropagation learning algorithm

devised to eliminate system stability problems resulting from the presence of load current positive feedback in the inverter neurocontroller learning process, and

- iv.) Experimental validation of the proposed neurocontroller in an UPS with passive and nonlinear loads.

A true adaptive intelligent inverter control technique has been demonstrated which enables the inverter controller to have the desirable characteristic of insensitivity to system parameter variations and modeling errors or equivalently, the control technique learns the desired inverter control function without knowledge of the output filter component values, inverter modulation and switching frequencies. Despite the use of relatively low inverter switching frequencies, the initial neurocontroller learning process response time compares favorably with the corresponding responses of conventional inverter control methods. Without the use of load current prediction techniques or controller gains based on knowledge of the output filter circuit design, inverter modulation and switching frequencies, the neurocontroller minimizes the steady-state load voltage errors inherently compensating for delays in controller and PWM process operation. The implementation of the neurocontroller technique was made without the use of specialized digital controller hardware, though the performance of this technique in terms of speed could be enhanced with the use of an application specific integrated circuit design.

The development of an adaptive intelligent inverter control technique based on the application of a neurocontroller was limited by the objective of producing a control technique which could be implemented in a practical fashion. The use of a variation of the direct adaptive control configuration and on-line neurocontroller training was judged to be the most likely approach to achieving this objective. As a consequence of implementation limitations, the final neurocontroller version utilized a single-layer neural network.

Alternate approaches to the application of the neurocontroller technique by which the observed system stability problems could have presumably been avoided would be the use of the hybrid control scheme depicted in figure 2.6 or the indirect adaptive control configuration depicted in figure 2.9. The use of the hybrid control scheme would have rendered the use of the neurocontroller as redundant since the conventional fast response controller used to generate neurocontroller training data is based on the assumption that inverter/filter system parameters are known precisely. Implementation problems ruled out the use of the indirect adaptive control configuration, as well as, other neurocontroller techniques involving more sophisticated learning strategies.

The use of the UPS inverter neurocontroller technique as a practical alternative to other conventional UPS inverter control techniques is realistically dependent on the specific UPS application. Its application is targeted for higher power UPS systems where inverter

switching frequencies are reduced due to switching delays and losses and limitations on the generation of electromagnetic interference.

4.3 Suggestions for Future Study

Though single-layer neurocontrollers do not have the capability of generating a nonlinear input/output mapping which is the basis for the use of multi-layer neurocontrollers, their forward transfer characteristics are readily defined in precise mathematical expressions enabling an analytical evaluation of neurocontroller performance in a closed-loop system utilizing on-line neurocontroller training. A similar evaluation of a multi-layer neurocontroller structure is typically tedious, particularly given the fact that the number of hidden layers and hidden layer nodes would be adjusted in order to synthesize a more precise neurocontroller nonlinear input/output mapping.

It is suggested that future neurocontroller development utilize multiple parallel single-layer units where a nonlinear controller input/output mapping is approximated by selectively activating single layer units whose forward transfer characteristics yield a minimum plant output error. The learning process would be selective as well where the weights for a given single-layer unit are updated if the weight change does not exceed a specified maximum deviation from the initial weight values. Single-layer units would be added if none of the existing units are updated based on the maximum weight deviation limit.

Further study is required with regard to the selection of the learning-rate parameter value based on controller stability considerations. With regard to this problem, selection of this parameter based on satisfying the equations

$$\underline{\underline{\epsilon}}_{dv}(t) \cdot d\underline{\underline{\epsilon}}_{dv}(t)/dt < 0 \quad (4.4a)$$

$$\underline{\underline{\epsilon}}_{qv}(t) \cdot d\underline{\underline{\epsilon}}_{qv}(t)/dt < 0 \quad (4.4b)$$

each controller sampling period is anticipated.

GTO and Diode Models for Inverter Leg Circuit Simulation

A1.1 Overview of GTO Operation

The name thyristor is generic for devices composed of four silicon layers (p-emitter connected to the anode, a n-base, a p-base, and a n-emitter connected to the cathode) and exhibiting a regenerative internal mechanism that latches the device in the on-state. The GTO (gate turn-off thyristor) has a highly interdigitated gate electrode structure, leaving the cathode as a large number of separate islands or fingers. With this structure, a large negative gate current terminates the internal regenerative process, turning off the GTO. A consequence of the gate electrode structure is that the GTO's anode-cathode dv/dt is inferior to that of a conventional thyristor. To overcome this problem, the gate-cathode junction is reverse-biased during the GTO off-state.

To ensure uniform current conduction in all cathode fingers during the turn-on process, an overdrive positive gate current pulse is required (along with a di/dt reactor limiting the rate-of-rise of anode current). During the GTO on-state, a maintaining gate current is

required to ensure uniform current conduction in all cathode fingers in the event of an increase of GTO load current.

During the gate-assisted turn-off process, the gate drive draws a high reverse current from the gate (the peak current value is typically 20 to 25% of the anode current value prior to the turn-off process). During the initial stage of gate-assisted turn-off, the p-base charge is extracted as the reverse gate current increases to its peak value. At its peak value, the regenerative process is interrupted where the increase in reverse gate current is linear in order to minimize gate losses during turn-off. When the negative gate current initially interrupts the regenerative process near the gate contact, the cathode current is focused into the region between the gate fingers. GTO turn-off begins when the active region supporting anode current conduction is reduced to a small filament. To prevent a destructive thermal runaway during turn-off due to the active region reduction, a turn-off snubber is required to divert anode current flow. As the GTO begins to turn off, the gate-cathode junction goes into avalanche. Once this occurs, further increases in reverse gate current are diverted through the avalanching region of the junction and do not contribute to the turn-off process. This event limits the maximum anode current that can be turn-off.

After the GTO load current is diverted into the snubber circuit, the anode current falls quickly to its tail current value. Tail current is the resulting anode diffusion current flow (which is exponential in shape) due to the n-base excess hole charge which remains after all the p-base charge is consumed. The gate and cathode at turnoff do not

significantly affect the tail current characteristics since this junction does not influence the wide, low concentration n-base region of the GTO. Tail current is very dependent on the dv/dt of the reapplied device voltage following forced commutation. Tail current characteristics are important since its magnitude and duration influences the device's switching speed in the inverter bridge configuration, and makes a significant contribution to device power loss during turn-off. When the gate-cathode junction comes out of avalanche, the tail and gate currents are the same.

A1.2 GTO Simulation

A GTO simulation has been designed to emulate GTO operation given data available from device data specifications. In order to reduce the simulation overhead, the gate drive circuit operation is assumed to be consistent with the operating conditions given in device specifications. The models applied to the different GTO modes of operation affect the electrical characteristics of the device between the anode and cathode terminals.

The GTO model depicted in figure A1.1 consists of two branches, one containing a resistor and the other containing an independent voltage source. The resistor value is modulated in a manner consistent with device conduction characteristics in each mode of operation. The independent voltage source becomes active only during the on-state mode.

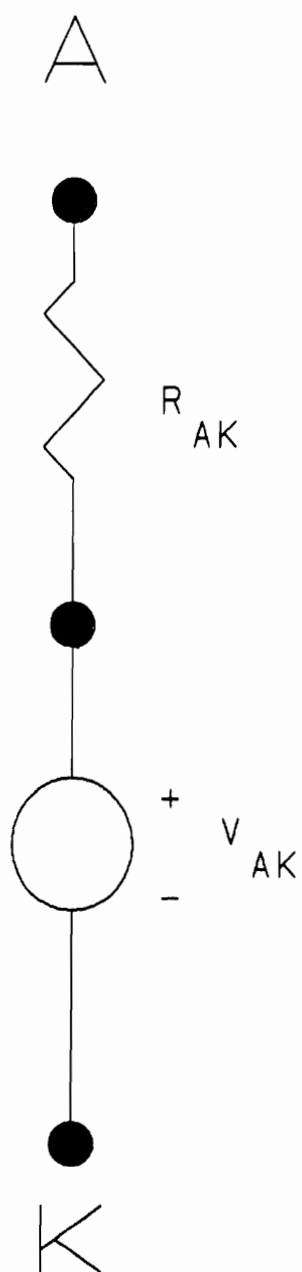


FIGURE A1.1: GTO MODEL

A1.2.1 GTO Off-State Model

In its off-state, the GTO is modeled as a resistance whose value is derived from the ratio V_{DRM}/I_{DRM} , where V_{DRM} is the peak repetitive forward blocking voltage and I_{DRM} is the peak forward leakage current. This model is applied to the forward blocking event, as well as, the reverse blocking event. The reverse blocking event corresponds to the anti-parallel diode forward conduction state, and so, diode forward conduction model characteristics will govern the event.

A1.2.2 GTO Turn-On Model

Device data given along with the maximum turn-on time, t_{gt} , include the on-state test current, $I_T(\text{TEST})$, the off-state test voltage, $V_D(\text{TEST})$, and the critical rate-of-rise of on-state current, dI_T/dt . It will be assumed that the device anode current rises at the critical rate with no contribution from the snubber capacitance. The anode current rise time is derived from the equation

$$t_{gr} = I_T(\text{TEST}) / (dI_T/dt). \quad (\text{A1.1})$$

The turn-on delay is derived from the equation

$$t_{gd} = t_{gt} - t_{gr}. \quad (\text{A1.2})$$

During t_{gd} , the GTO model remains in its off-state. During the anode current rise time, the GTO model is a variable resistance whose value is given by the equation

$$R_{AK} = \frac{V_{D(TEST)} \cdot [1 - (t - t_{gd})/t_{gr}] + V_{TM}}{I_{T(TEST)} \cdot (t - t_{gd})/t_{gr} + I_{DRM}}, \quad (A1.3)$$

where V_{TM} is the peak on-state voltage and $t_{gd} \leq t \leq t_{gt}$. During t_{gr} , a linear anode current rise and device voltage decline is assumed. The GTO model switches to its on-state version when $t = t_{gt}$.

A1.2.3 GTO On-State Model

In its on-state, the GTO anode-cathode voltage is equal to the voltage drop across both of its forward-biased junctions (1.4 volts) and the voltage drop dependent on the anode current magnitude. The on-state model is an independent voltage source (1.4 volts) with a series resistance whose value is given by the expression

$$(V_{TM} - 1.4)/I_{TM},$$

where V_{TM} and I_{TM} are the maximum device on-state voltage and current.

A1.2.4 GTO Turn-Off Model

During gate-assisted turn-off, the GTO model goes through three stages of operation; a turn-off delay, t_{dq} , anode current fall time, t_{fq} , and a tail current period, t_z . The following assumptions are made with regard to the development of this model:

- i.) The reverse gate current slope, dI_{RG}/dt , is constant and is equal to the value given in the device data associated with the turn-off time specification.

- ii.) The snubber circuit component values, L_S and C_S , are equal to the values given in the device data associated with the turn-off time specification, and the rate-of-rise of device voltage during turn-off is equal to the critical value given in the device data.
- iii.) The maximum tail current, I_{ZM} , is equal to the maximum reverse gate current, I_{RGM} . This assumption is made based on the condition that the maximum reverse gate current flow is maintained at a constant value by the gate drive circuit inductance during t_{fq} .

On-state anode current flow is interrupted when the extracted p-base charge equals Q_{GQ} (the area under the reverse gate current curve during the t_{dq} interval). Given the constant dI_{RG}/dt assumption, t_{dq} is also derived from I_{RGM} or

$$I_{RGM} = I_{T(ON\_STATE)}/G_{OFF}, \quad (A1.4)$$

and

$$t_{dq} = I_{RGM}/(dI_{RG}/dt), \quad (A1.5)$$

where $I_{T(ON\_STATE)}$ is the anode current prior to the turn-off process, and G_{OFF} is the turn-off gain. The dependence of t_{dq} on the anode current prior to turn-off is apparent. During t_{dq} , the GTO model remains in its on-state.

t_{fq} is obtained directly from device data specifications (if available), or derived from the snubber circuit values and the critical rate-of-rise of device voltage. Given the assumption that the device voltage rise is linear during t_{fq} and equal to the critical value, dV_D/dt , the anode current during t_{fq} is given by the equation

$$I_A(t) = I_T(\text{ON\_STATE}) \cdot \cos[(t - t_{dq})/\sqrt{L_S C_S}], \quad t_{dq} \leq t \leq t_{dq} + t_{fq}. \quad (\text{A1.6})$$

At the end of the anode current fall time, the anode current is equal to the maximum reverse gate current, and so, t_{fq} is derived from the equation

$$t_{fq} = \sqrt{L_S C_S} \cdot \cos^{-1}(1/G_{OFF}). \quad (\text{A1.7})$$

Given that the turn-off gain is dependent on the anode current prior to turn-off, the anode current fall time is dependent on the anode current value as well.

During t_{fq} , the GTO is modeled as a variable resistance whose value is given by the equation

$$R_{AK} = \frac{V_{TM} + dV_D/dt \cdot (t - t_{dq})}{I_{TM} \cdot \cos[(t - t_{dq})/\sqrt{L_S C_S}]}, \quad t_{dq} \leq t \leq t_{dq} + t_{fq}. \quad (\text{A1.8})$$

After t_{fq} , the tail current is equal to the reverse gate current flow after gate-cathode junction recovery. The turn-off delay and anode current fall time depend on the anode current prior to the turn-off process. On the other hand, the tail current period, t_z , will be regarded as a constant interval in this model. The tail current period is derived from the maximum turn-off time, and the delay and fall times corresponding to the test circuit conditions or

$$t_z = t_{gq} - t_{dq}(\text{TEST}) - t_{fq}(\text{TEST}), \quad (\text{A1.9})$$

where $t_{dq}(\text{TEST})$ and $t_{fq}(\text{TEST})$ are computed given $I_T = I_{TM}$.

The tail current is exponential in shape and described by the equation

$$I_Z(t) = I_{RGM} \cdot \exp[-(t - t_{dq} - t_{fq})/\tau_z], \quad (A1.10)$$

where $t_{dq} + t_{fq} \leq t \leq t_{dq} + t_{fq} + t_z$ and $\tau_z \approx t_z/5$ where $I_Z(t_z) \approx 0$. During t_z , the device voltage continues its rise at the critical rate to its off-state voltage.

During the tail current period, the GTO is modeled as a variable resistance. The variable resistance model is executed in two stages:

i.) The resistance value while the device voltage is rising is given by

$$R_{AK} = \frac{V_{TM} + dV_D/dt \cdot (t - t_{dq})}{I_{RGM} \cdot \exp[-(t - t_{dq} - t_{fq})/\tau_z]}, \quad t_{dq} + t_{fq} \leq t \leq t_{dq} + t_{z1},$$

where $t_{z1} = (V_{D(TEST)} - V_{TM})/(dV_D/dt)$, $V_{D(TEST)}$ is the device off-state voltage given in the turn-off time test specifications. I_{RGM} is the maximum reverse gate current required to interrupt the maximum rated anode current, I_{TM} .

ii.) Once the device voltage has risen to its off-state value, the resistance value (for $t_{dq} + t_{z1} \leq t \leq t_{dq} + t_{fq} + t_z$) is given by

$$R_{AK} = \frac{V_{D(TEST)}}{I_{RGM} \cdot \exp[-(t - t_{dq} - t_{fq})/\tau_z]}. \quad (A1.11)$$

The GTO turn-off model process terminates when $t = t_{dq} + t_{fq} + t_z$. The GTO model switches over to its off-state model afterwards.

A1.2.5 GTO Reverse Blocking State

In the inverter circuit, GTO reverse voltage is limited by the diode in anti-parallel with the GTO. A diode forward conduction event causes the GTO model to be switched to the off-state model (diode conduction governs the circuit response). The rules for a transition from the off-state model after the termination of the diode conduction event are discussed in the next section.

A1.2.6 GTO Simulation Software

The previous discussion of a GTO model has been incorporated into the software module written in the C programming language. This module operates on one GTO, and so, six calls to this module are required per inverter simulation step in order to operate all six GTO's in the inverter bridge. The module requires as input the following items;

- i.) GTO model data (one data structure per GTO),
- ii.) the PWM control signal (1 => on and 0 => off),
- iii.) the anode current,
- iv.) the anti-parallel diode current, and
- v.) the simulation step time.

The turn-on or turn-off process is initiated at the level transitions of the PWM control signal. A positive transition triggers the turn-on process, and a negative transition triggers the turn-off

process. Both the turn-on and turn-off processes are driven by their respective process timers which are reset when either process is initiated. The process timer is incremented if the process is enabled. Time comparisons using these timers trigger changes in the GTO model.

The negative anode/cathode voltage causes the GTO model to enter its off-state. After the reverse voltage event terminates, two courses of action may be taken;

- i.) If the GTO model is in its turn-on process (which includes its on-state) when diode conduction terminates, the turn-on process is set to the anode current rise time interval.
- ii.) If the GTO model is in its turn-off process when diode conduction terminates, the GTO model remains in its off-state (it is assumed that diode conduction completes the turn-off process).

Additional delays are added to the turn-on and turn-off processes to account for delays caused by the gate drive circuitry.

A1.3 Overview of Power Diode Operation

The bipolar diode (rather than the Schottky diode) will be used in the inverter leg simulation. Of interest in the development of a diode model are the diode operating characteristics during forward conduction and reverse recovery. During steady-state forward conduction, the diode forward voltage drop increases with forward current. This increase can be considered linear up to the diode's rated current. Above rated

current, the current-voltage relationship becomes somewhat nonlinear (current-voltage slope flattens out).

Diode reverse recovery is the most significant characteristic of diode operation in the inverter leg circuit. Large values of diode reverse recovery current and the generation of high reverse diode voltages place stresses on other components in the inverter leg circuit. Waveforms depicting the diode reverse recovery process are shown in figure A1.2. Diode reverse recovery behavior is dependent on the circuit in which the diode is applied. With regard to the inverter leg circuit simulation, the inverter leg current-limiting reactor is part of the diode circuit during reverse recovery, and it limits the rate-of-rise of reverse recovery current. During the initial stage of diode reverse recovery ($t < t_a$), the reverse diode rises to a maximum value I_{RM} . This maximum value is proportional to the recovered charge, Q_R , and the rate-of-rise of reverse current, dI_R/dt , as given in the following equation

$$I_{RM} = K_2 \cdot [Q_R \cdot dI_R/dt]^{1/2}, \quad (A1.12)$$

where K_2 is a constant. The previous equation implies that the maximum reverse recovery current is a function of the recovered charge and the slope of the reverse recovery current. The reverse recovery time, t_{rr} , is also proportional to the recovered charge and the rate-of-rise of reverse current as given in the following equation

$$t_{rr} = K_3 \cdot [Q_R / (dI_R/dt)]^{1/2}, \quad (A1.13)$$

where K_3 is a constant.

During reverse recovery, the reverse diode current does not remove all of the charge stored during forward conduction; some of it

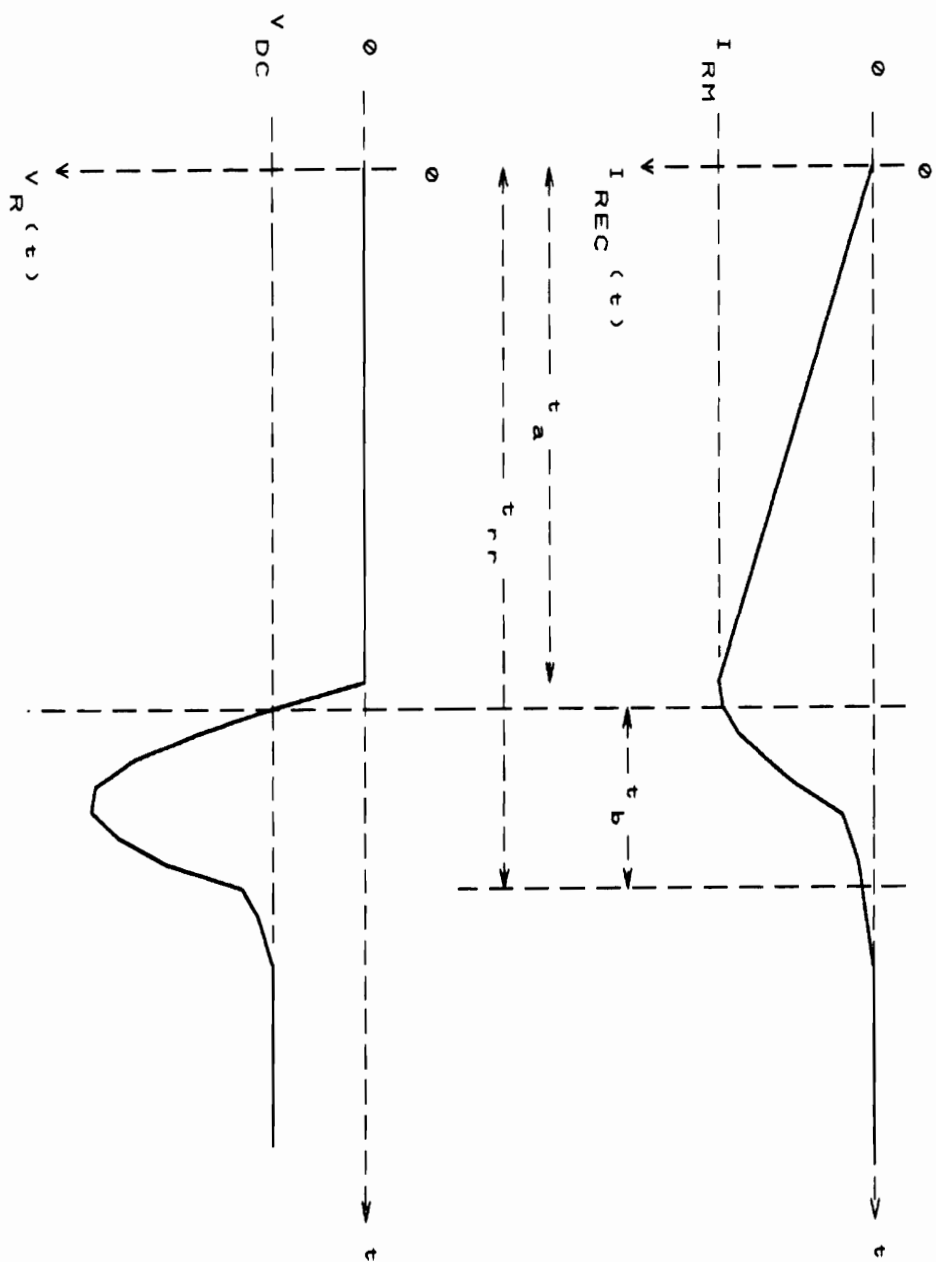


FIGURE A1.2: DIODE REVERSE RECOVERY CURRENT AND VOLTAGE WAVEFORMS

recombines. How much recombines depends on the magnitude of the reverse current compared to that of the forward current prior to recovery. If the reverse current, I_R , is much greater than the previous forward current, almost all the stored charge is recovered in a time much less than τ_a , and only a small amount will be removed by recombination. For $I_R \leq I_F$, a substantial fraction of the stored charge will recombine. This later event is more likely in the case of typical inverter leg operation.

With reference to the waveforms depicted in figure A1.2, the diode recovery parameters are described as follows;

- t_a - the point in time where the recovery current reaches its maximum value, I_{RM} . The diode voltage is near zero during this period. The current-limiting reactor in the inverter leg circuit limits the rate-of-rise of reverse current to a constant value during this interval. By t_a , the positive charge stored in the diode during diode forward conduction is extracted. The diode voltage begins a negative rise from zero (with a corresponding increase in negative charge stored in the diode).
- t_b - the time lapse from I_{RM} to the intersection with the time axis of a line through I_{RM} and $0.25 \cdot I_{RM}$. A snap factor, $s = t_b/t_a$, is an indicator of the diode recovery characteristics. Low values of s ($t_a \gg t_b$) correspond to higher peak reverse diode voltages during the t_b interval (given the presence of the current-limiting reactor). Larger values of s imply a softer diode recovery (or lower peak reverse diode voltage).

The reverse recovery time, t_{rr} , and recovered charge, Q_R , are given in diode data specifications along with the test circuit configuration and test parameters. t_a , t_b , and I_{RM} are not always supplied directly (these parameters are dependent on the diode circuit configuration).

The diode model developed is shown in figure A1.3 (later sections will describe this model in detail).

A1.3.1 Diode Model Capacitance

Diode model capacitance is constant in all modes with the exception of reverse recovery for $t < t_a$. This capacitance value, C_{DF} , is derived from the maximum recovered charge, Q_{RR} , and diode forward voltage drop, V_F , or

$$Q_{RR} = C_{DF} \cdot V_F. \quad (A1.14)$$

Q_{RR} is typically specified along with the forward diode current. V_F is the diode forward voltage drop corresponding to the diode forward current specified with Q_{RR} .

During reverse recovery ($t < t_a$), the voltage across the diode is nearly zero. During this period of time, the model capacitance is set to a very large value. This is consistent with the definition of diode capacitance (assuming an abrupt junction) when the diode is reverse-biased or

$$C_{DIODE} = K_1 \cdot V_R^{-1/2}, \quad (A1.15)$$

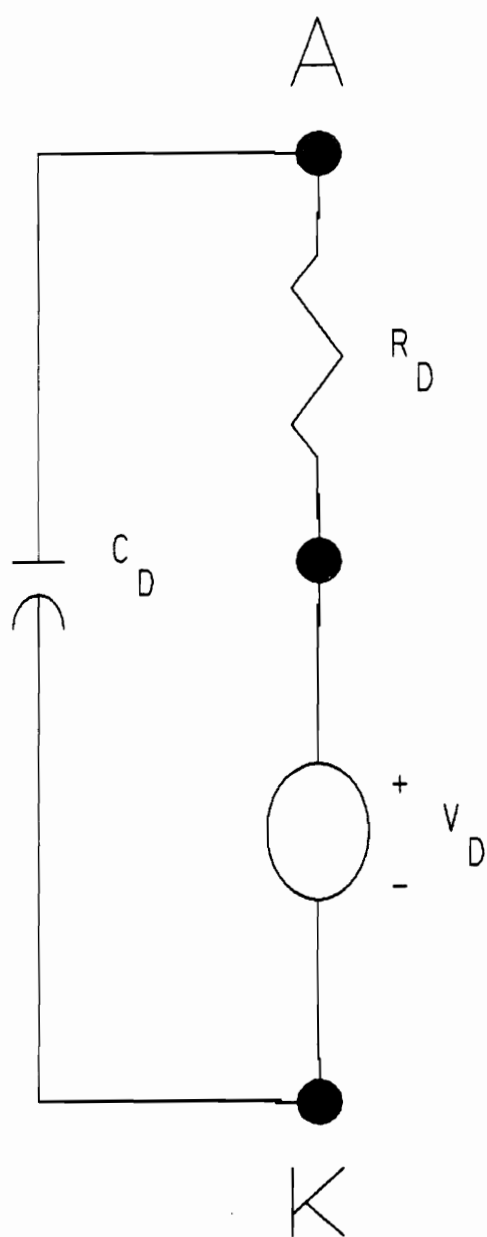


FIGURE A1.3: DIODE MODEL

where K_1 is a constant, and V_R is the algebraic sum of the unbiased barrier voltage and the externally-applied reverse voltage.

A1.3.2 Diode Model Resistance

During diode reverse-blocking (negative capacitance branch voltage), the model resistance is derived from the expression

$$R_{\text{DIODE}} = V_R / I_R, \quad (\text{A1.16})$$

where V_R is the maximum DC reverse voltage and I_R is the reverse diode current at V_R .

While the model capacitance branch voltage is negative or less than the unbiased barrier voltage (0.7 volt), the model resistance is equal to the reverse-blocking value. When the capacitance branch voltage is equal to or greater than the barrier voltage, the model voltage source becomes active (and equal to 0.7 volt) and the model resistance is derived from the equation

$$R_{\text{DIODE}} = (V_F - 0.7) / I_F, \quad (\text{A1.17})$$

where V_F is the forward voltage given forward current I_F (data obtained from device data specifications).

A1.3.3 Power Diode Simulation Software

The previous discussion of a power diode model has been incorporated into the software module written in the C programming language. This module operates on one diode, and so, six calls to this

module are required per inverter simulation step in order to operate all six diodes in the inverter bridge. The module requires as input the following items;

- i.) Diode model data,
- ii.) the model capacitance forward voltage,
- iii.) the model capacitance forward charge, and
- iv.) the diode forward current.

Model activity is determined by the polarity of the diode forward current. Positive current places the model in its forward conduction mode. In this mode, model activity is directed by the magnitude of the model capacitance forward voltage (which is equivalent to the diode forward voltage). For diode forward voltage less than 0.7 volt, the diode is placed in its blocking state. Otherwise, the diode is placed in its forward conduction state.

Negative current places the diode model in either its reverse recovery or blocking state. The model enters its reverse recovery state if the model capacitance forward charge is greater than zero. Otherwise, the model is placed in its blocking state.

Appendix 2

The Inverter Leg Model

Using the diode and GTO models described in appendix 1, the inverter leg circuit depicted in figure A2.1 is transformed into the network depicted in figure A2.2. Four matrix equations describing the network component relationships, Kirchhoff's voltage and current laws are given below:

$$M \cdot v - N \cdot i - P \cdot x = e, \quad (A2.1)$$

where

$M =$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{d4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{d8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{d12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{d16} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_s \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

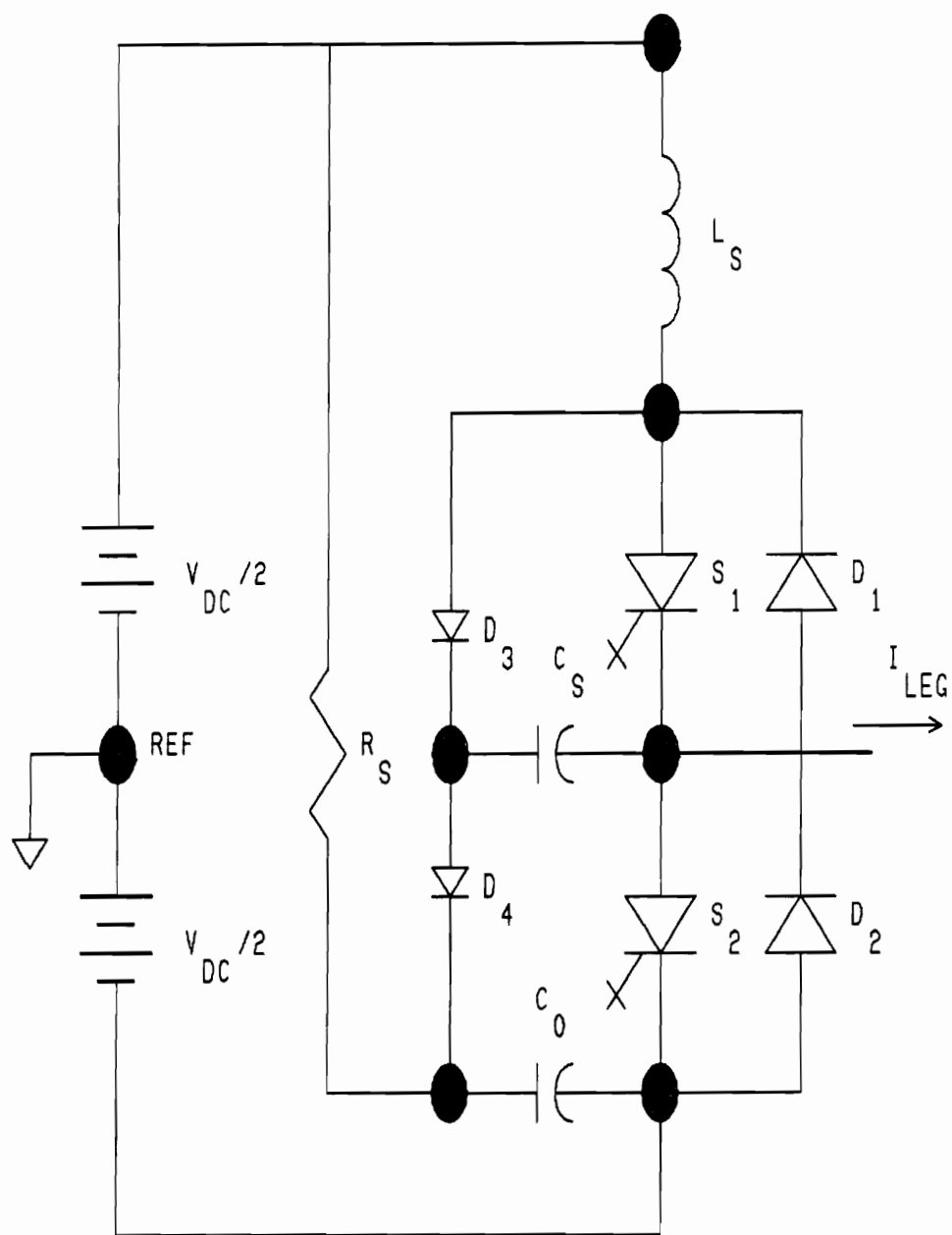


FIGURE A2.1: INVERTER LEG CIRCUIT

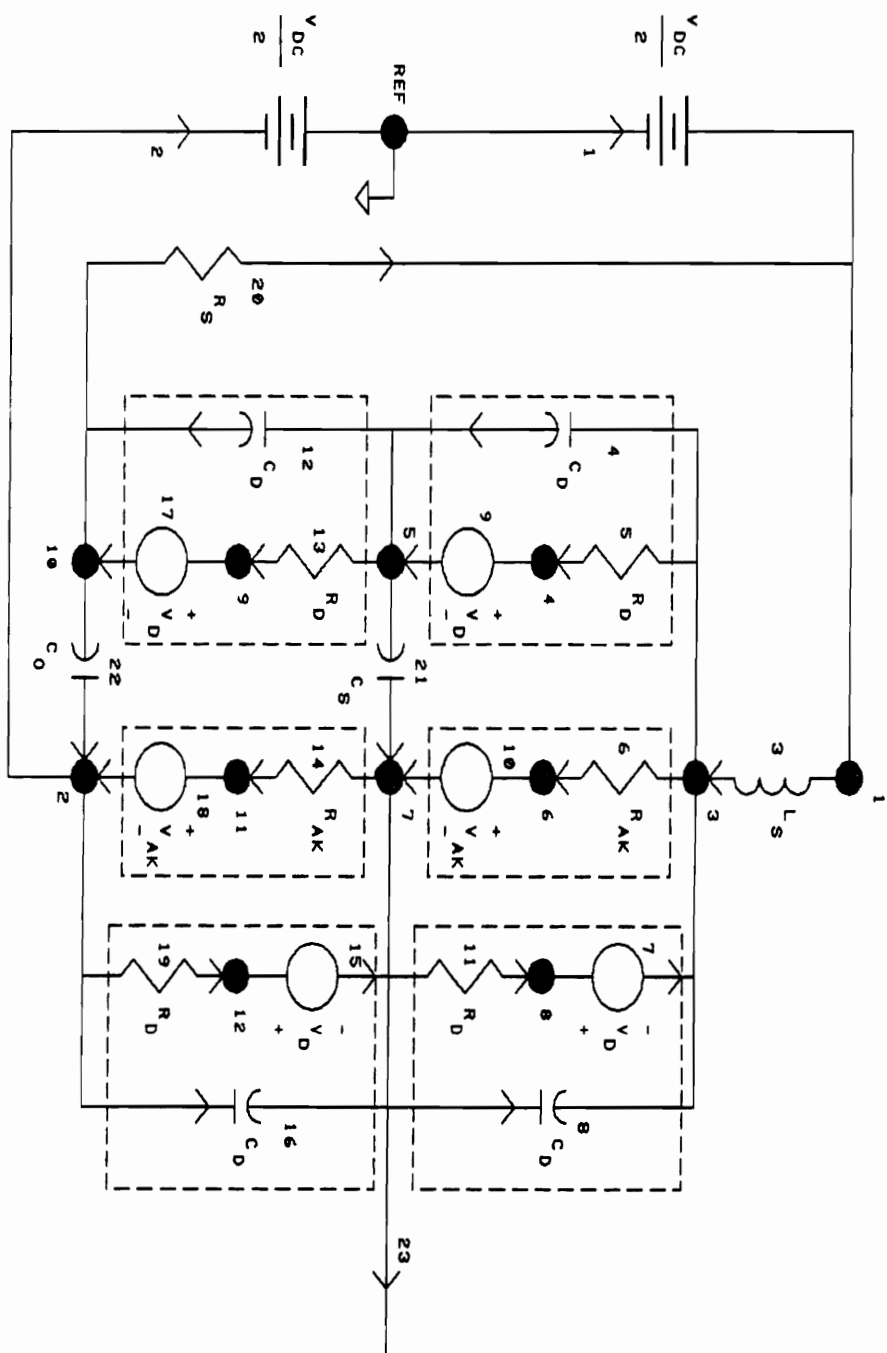


FIGURE A2.2: INVERTER LEG MODEL

N =

1

$$i = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \\ I_7 \\ I_8 \\ I_9 \\ I_{10} \\ I_{11} \\ I_{12} \\ I_{13} \\ I_{14} \\ I_{15} \\ I_{16} \\ I_{17} \\ I_{18} \\ I_{19} \\ I_{20} \\ I_{21} \\ I_{22} \\ I_{23} \end{bmatrix}, \text{ branch currents,}$$

$$P \cdot x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \phi_3 \\ q_4 \\ q_8 \\ q_{12} \\ q_{16} \\ q_{21} \\ q_{22} \end{bmatrix},$$

$$\begin{aligned} \Phi_3 &= L_S \cdot I_3, \\ q_4 &= C_{d4} \cdot V_4, \\ q_8 &= C_{d8} \cdot V_8, \\ q_{12} &= C_{d12} \cdot V_{12}, \\ q_{16} &= C_{d16} \cdot V_{16}, \\ q_{21} &= C_S \cdot V_{21}, \\ q_{22} &= C_O \cdot V_{22}, \end{aligned}$$

$$e =$$

$$\begin{bmatrix} -V_{dc}/2 \\ -V_{dc}/2 \\ 0 \\ 0 \\ 0 \\ 0 \\ V_{d7} \\ 0 \\ V_{d9} \\ V_{AK10} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ V_{d15} \\ 0 \\ V_{d17} \\ V_{AK18} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ I_{LEG} \end{bmatrix}, \text{ source vector.}$$

$$\mathbf{K}_v \cdot \mathbf{v} + \mathbf{K}_i \cdot \mathbf{i} - (d/dt) \cdot \mathbf{x} = 0, \quad (\text{A2.2})$$

where

$$K_V =$$

[illegible]

[illegible]

where

[illegible]

where

$A' =$

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$v_n =$

$$\begin{bmatrix} v_{n1} \\ v_{n2} \\ v_{n3} \\ v_{n4} \\ v_{n5} \\ v_{n6} \\ v_{n7} \\ v_{n8} \\ v_{n9} \\ v_{n10} \\ v_{n11} \\ v_{n12} \end{bmatrix}, \text{ node voltages.}$$

Solutions for the inverter leg network node voltages, branch currents and voltages, and circuit charges and fluxes are obtained from the matrix equation

$$\begin{bmatrix} 0 & A & 0 & 0 \\ A' & 0 & -U & 0 \\ 0 & -N & M & -P \\ 0 & K_i & K_v & -U \cdot d/dt \end{bmatrix} \cdot \begin{bmatrix} v_n \\ i \\ v \\ x \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ e \\ 0 \end{bmatrix}. \quad (A2.5)$$

Using the discrete-time approximation for the first derivative

$$(d/dt)x \approx [x(n) - x(n - 1)]/\delta t$$

where δt is the discrete-time interval, the solution of the previous matrix equation is expressed as

$$\begin{bmatrix} v_n \\ i \\ v \\ x \end{bmatrix} = \begin{bmatrix} 0 & A & 0 & 0 \\ A' & 0 & -U & 0 \\ 0 & -N & M & -P \\ 0 & K_i & K_v & -U/\delta t \end{bmatrix}^{-1} \cdot \begin{bmatrix} 0 \\ 0 \\ e \\ -x(n - 1)/\delta t \end{bmatrix}. \quad (A2.6)$$

A software module written in the C programming language generates the solution to equation (A2.6) for each of the three inverter legs in the three phase bridge. The solutions are returned to the calling routine via a pointer to the vector group

$$\begin{bmatrix} v_n \\ i \\ v \\ x \end{bmatrix}_{\text{PHASE A}}, \begin{bmatrix} v_n \\ i \\ v \\ x \end{bmatrix}_{\text{PHASE B}}, \begin{bmatrix} v_n \\ i \\ v \\ x \end{bmatrix}_{\text{PHASE C}}.$$

Inputs to this software module are listed below:

- i.) Snubber component values C_s , C_o , L_s , and R_s .
- ii.) The DC power supply voltage, V_{dc} .
- iii.) The inverter leg load currents, I_{LEG} .
- iv.) The individual diode and GTO model values (pointers to model data structures).

Appendix 5 describes in detail the software design of the inverter/load simulation of which this software module is a component.

Appendix 3

Three-Phase Load Models

A3.1 Passive Inductive Load Model

Figure A3.1 depicts the passive inductive load model. In addition to the inverter's output filter, the load consists of three parallel RL circuits connected line-to-line across the filter's output bus.

As in the analysis of inverter leg operation given in appendix 2, four matrix equations describing the network component relationships, Kirchhoff's voltage and current laws are given below:

$$M \cdot v - N \cdot i - P \cdot x = e, \quad (A3.1)$$

where

$M =$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_f & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_f & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_f & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

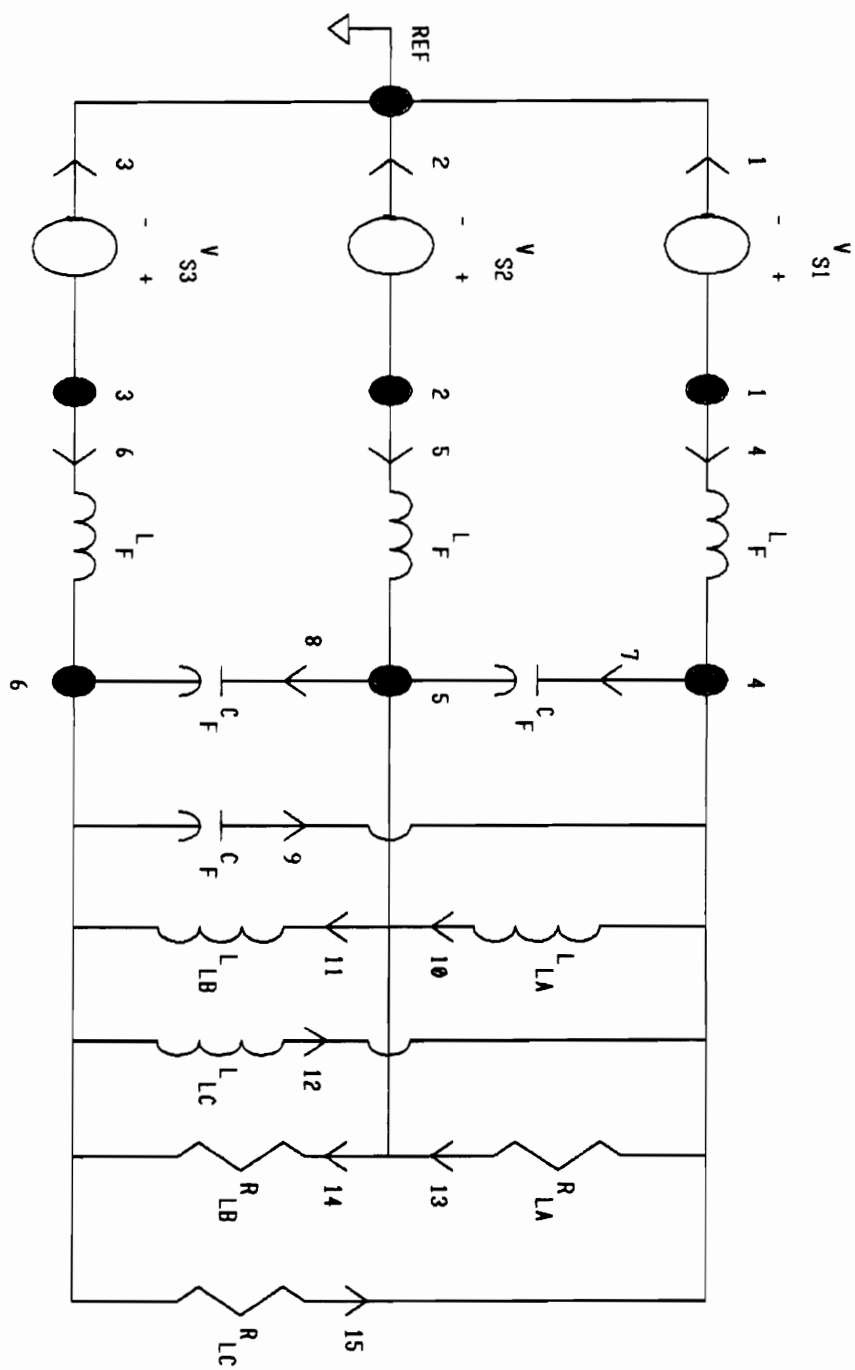


FIGURE A3.1: PASSIVE INDUCTIVE LOAD MODEL

$\mathbf{v} =$

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \\ v_{11} \\ v_{12} \\ v_{13} \\ v_{14} \\ v_{15} \end{bmatrix}, \text{ branch voltages,}$$

$\mathbf{N} =$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -L_f & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -L_f & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -L_f & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -L_{LA} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -L_{LB} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -L_{LC} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & R_{LA} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & R_{LB} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & R_{LC} & 0 \end{bmatrix},$$

i =

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \\ I_7 \\ I_8 \\ I_9 \\ I_{10} \\ I_{11} \\ I_{12} \\ I_{13} \\ I_{14} \\ I_{15} \end{bmatrix}, \text{ branch currents,}$$

P·x =

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Phi_4 \\ \Phi_5 \\ \Phi_6 \\ q_7 \\ q_8 \\ q_9 \\ \Phi_{10} \\ \Phi_{11} \\ \Phi_{12} \end{bmatrix},$$

$$\begin{aligned} \Phi_4 &= L_f \cdot I_4, \\ \Phi_5 &= L_f \cdot I_5, \\ \Phi_6 &= L_f \cdot I_6, \\ q_7 &= C_f \cdot V_7, \\ q_8 &= C_f \cdot V_8, \\ q_9 &= C_f \cdot V_9, \\ \Phi_{10} &= L_{LA} \cdot I_{10}, \\ \Phi_{11} &= L_{LB} \cdot I_{11}, \\ \Phi_{12} &= L_{LC} \cdot I_{12}, \end{aligned}$$

e =

$$\begin{bmatrix} v_{s1} \\ v_{s2} \\ v_{s3} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \text{ source vector.}$$

$$K_v \cdot v + K_i \cdot i - (d/dt) \cdot x = 0, \quad (A3.2)$$

where

$K_v =$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

$K_i =$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$A \cdot i = 0, \quad (A3.3)$$

where

$A =$

$$\begin{bmatrix} -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 1 & -1 & 0 & 1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \end{bmatrix},$$

$$A' \cdot v_n = v, \quad (A3.4)$$

where

$A' =$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix},$$

$v_n =$

$$\begin{bmatrix} v_{n1} \\ v_{n2} \\ v_{n3} \\ v_{n4} \\ v_{n5} \\ v_{n6} \end{bmatrix}, \text{ node voltages.}$$

As in the analysis of inverter bridge operation, the four matrix equations are combined into one matrix equation:

$$\begin{bmatrix} 0 & A & 0 & 0 \\ A' & 0 & -U & 0 \\ 0 & -N & M & -P \\ 0 & K_i & K_v & -U \cdot d/dt \end{bmatrix} \cdot \begin{bmatrix} v_n \\ i \\ v \\ x \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ e \\ 0 \end{bmatrix}. \quad (A3.5)$$

Discrete-time solutions for the load network branch currents and voltages, node voltages, charges and fluxes are obtained by solving:

$$\begin{bmatrix} v_n \\ i \\ v \\ x \end{bmatrix} = \begin{bmatrix} 0 & A & 0 & 0 \\ A' & 0 & -U & 0 \\ 0 & -N & M & -P \\ 0 & K_i & K_v & -U/\delta t \end{bmatrix}^{-1} \cdot \begin{bmatrix} 0 \\ 0 \\ e \\ -x(n-1)/\delta t \end{bmatrix}, \quad (A3.6)$$

where δt is the discrete-time interval. A software module written in the C programming language performs the solution of equation (A3.6).

A3.2 Nonlinear Load Model

Figure A3.2 depicts a typical nonlinear load configuration used to examine the operation of a three-phase inverter. In addition to the inverter's output filter, the load consists of an uncontrolled three-phase rectifier with a L-C filter and AC input line inductances.

The same technique used to analyze inverter and passive load operation is applied to the analysis of this load type. The individual matrices associated with this load type are given below:

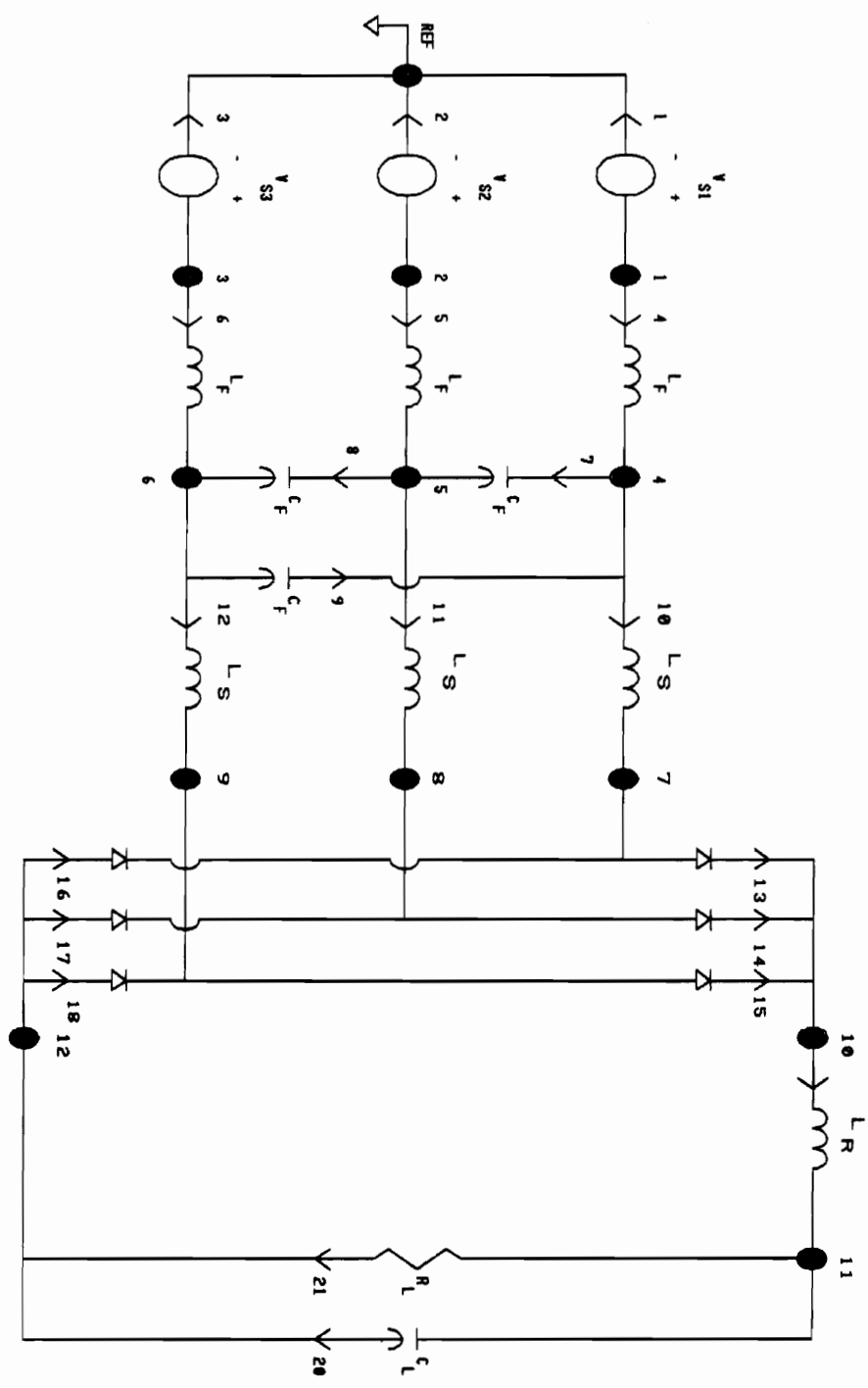


FIGURE A3.2: NONLINEAR LOAD MODEL

[illegible]

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \\ v_{11} \\ v_{12} \\ v_{13} \\ v_{14} \\ v_{15} \\ v_{16} \\ v_{17} \\ v_{18} \\ v_{19} \\ v_{20} \\ v_{21} \end{bmatrix}, \text{ branch voltages,}$$

[illegible]

R_{10} through R_{15} are the uncontrolled rectifier bridge diode resistances,

$$\mathbf{i} = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \\ I_7 \\ I_8 \\ I_9 \\ I_{10} \\ I_{11} \\ I_{12} \\ I_{13} \\ I_{14} \\ I_{15} \\ I_{16} \\ I_{17} \\ I_{18} \\ I_{19} \\ I_{20} \\ I_{21} \end{bmatrix}, \text{ branch currents.}$$

$$P \cdot x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Phi_4 \\ \Phi_5 \\ \Phi_6 \\ q_7 \\ q_8 \\ q_9 \\ \Phi_{10} \\ \Phi_{11} \\ \Phi_{12} \\ \Phi_{19} \\ q_{20} \end{bmatrix},$$

$$\Phi_4 = L_f \cdot I_4, \quad \Phi_{10} = L_s \cdot I_{10}, \quad q_7 = C_f \cdot V_7,$$

$$\Phi_5 = L_f \cdot I_5, \quad \Phi_{11} = L_s \cdot I_{11}, \quad q_8 = C_f \cdot V_8,$$

$$\Phi_6 = L_f \cdot I_6, \quad \Phi_{12} = L_s \cdot I_{12}, \quad q_9 = C_f \cdot V_9, \quad q_{20} = C_L \cdot V_{20},$$

e =

$$\begin{bmatrix} v_{s1} \\ v_{s2} \\ v_{s3} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

, source vector,

[illegible]

[illegible]

$$\begin{bmatrix} -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & 1 & 1 \end{bmatrix}$$

$A' =$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix},$$

$v_n =$

$$\begin{bmatrix} v_{n1} \\ v_{n2} \\ v_{n3} \\ v_{n4} \\ v_{n5} \\ v_{n6} \\ v_{n7} \\ v_{n8} \\ v_{n9} \\ v_{n10} \\ v_{n11} \\ v_{n12} \end{bmatrix}, \text{ node voltages.}$$

A software module written in the C programming language performs the solution of equation A4.6 for the nonlinear load simulation. A routine in this module controls the values of resistances R_{10} through R_{15} in a fashion which mimics diode operation where these resistances correspond to the branches in which the diodes are located.

A3.3 Load Model Function

In Appendix 5, the function of the inverter/load software simulation is discussed. These load models along with the inverter bridge simulation are key elements of the inverter/load simulation. Load model voltages V_{s1} , V_{s2} , and V_{s3} are the inverter leg output node voltages where both the inverter bridge and load are referenced to the DC power supply mid-point. Load model currents in branches 1, 2, and 3 are also the inverter leg output currents. Therefore, the inverter bridge simulation supplies to the load simulation its source voltages, while in response, the load simulation supplies to the inverter bridge simulation its load current.

In the passive inductive load model, the load resistance and power factor on each line-to-line load element may be set individually (if an unbalanced load is desired). In the nonlinear load model, the rectifier load resistance and capacitance may be set to values corresponding to a desired load characteristic. A header file included with each load simulation module describes the data structure which would contain this information.

Load branch and node values, in addition to an analysis of inverter/load operation, are used in closed-loop control system simulations.

Simulation Software Implementation of PWM Techniques

A4.1 Carrier PWM Methods

Carrier PWM methods generate PWM gating signals (signals controlling the inverter leg switches) by comparing a modulation waveform (typically a sinusoid) with a carrier waveform (typically a triangular waveform). The result of the comparison is a two-level PWM waveform, the waveform level corresponds to the polarity of the difference between instantaneous modulation and carrier waveform values. For example, if $C(t)$ is the function describing the carrier waveform and $M(t)$ is the function describing the modulation waveform, the PWM waveform generation process could be described by the term

$$\tanh\{G \cdot [C(t) - M(t)]\}, \quad G \rightarrow \infty,$$

where the PWM waveform level is either +1 or -1 depending on the polarity of the difference $C(t) - M(t)$.

Each of the routines written to mimic carrier PWM methods input the following parameter list:

- f_s - the inverter switching or carrier frequency in Hertz.
- f_m - the modulation waveform fundamental frequency in Hertz.
- M - the fundamental frequency modulation index.
- $M3$ - the third-harmonic distortion injection index.
- phase - the modulation waveform phase displacement.

t - simulation time in seconds.

In each routine, the carrier waveform is a bipolar triangular waveform (-1 and +1 peak amplitudes) generated by the set of equations:

$$-1.0 + 4.0 \cdot [t \text{ MOD } t_s] / t_s, \text{ for } [t \text{ MOD } t_s] \leq t_s / 2.0, \quad (\text{A4.1})$$

$$3.0 - 4.0 \cdot [t \text{ MOD } t_s] / t_s, \text{ for } [t \text{ MOD } t_s] \geq t_s / 2.0, \quad (\text{A4.2})$$

where $t_s = 1/f_s$. The sinusoidal modulation waveform is generated from the following equation:

$$M \cdot \{\sin(2\pi f_m t + \text{phase}) + M3 \cdot \sin(6\pi f_m t + 3 \cdot \text{phase})\}. \quad (\text{A4.3})$$

Third-harmonic distortion is added to the sinusoidal waveform via the M3 factor. Note that in the sampled PWM techniques, the time variable in equation (A4.3) is replaced by the waveform sample time.

Five software routines have been written to mimic the natural sampled technique, two versions of the symmetric regular sampled technique, and two versions of the asymmetric regular sampled technique:

NSPWM() - routine executing the natural sampled technique.

SSPWM1() - routine executing the symmetric regular sampled technique, modulation waveform sampling occurs at the carrier waveform negative peak.

SSPWM2() - routine executing the symmetric regular sampled technique, modulation waveform sampling occurs at the carrier waveform positive peak.

ASPWM1() - routine executing the asymmetric regular sampled technique, modulation waveform sampling occurs at the carrier waveform peaks.

ASPWM2() - routine executing the asymmetric regular sampled technique, modulation waveform sampling occurs at the carrier waveform zero-crossings.

The negative peak of the carrier waveform is aligned with the start of each carrier period. Routines SSPWM1() and ASPWM1() mimic modulation waveform real-time sampling. Routines SSPWM2() and ASPWM2() mimic modulation waveform sampling as typically executed from waveform data stored in ROM.

Each routine returns a +1 if the modulation waveform value is greater than or equal to the carrier waveform value at time t , otherwise -1 is returned.

A4.2 Three-Phase Carrier PWM Gating Generation

The routines NSPWM(), SSPWM1(), SSPWM2(), ASPWM1(), and ASPWM2() generate PWM bipolar gating signals for one inverter leg. Routine INVERTER() generates PWM bipolar gating signals via carrier-comparison PWM techniques for a three-phase inverter bridge. Input parameters passed to INVERTER() are f_s , f_m , M , $M3$, phase, t , and a pointer, mod_func, to one of the previously described PWM signal generation functions. INVERTER() makes three calls to the PWM signal generation function where the modulation function is phase-shifted by $2\pi/3$ radians after each function call. The return value of each PWM signal generation function call (either +1 or -1) is stored in the array INVERTER_STATES (INVERTER_STATES[0] contains the phase A PWM signal, INVERTER_STATES[1]

contains the phase B PWM signal, and INVERTER_STATES[2] contains the phase C PWM signal. The routine returns a pointer to array INVERTER_STATES.

A4.3 Space Vector Modulation

The load current flow path through the inverter leg is determined by the conduction state of the power semiconductor switches and diodes, and the inverter leg load current sense. Setting aside semiconductor switch rise and fall times, snubber circuit operation, and diode reverse recovery, the primary load current flow paths are through either the semiconductor switches or the anti-parallel diodes where eight unique path configurations describe three-phase inverter bridge operation. If the inverter is assumed to have ideal switching characteristics (instantaneous switching with no voltage drops or leakage currents), the load current flow paths could be described by a depiction of toggle switches connecting the load to either DC power supply rail (depicted in figure A4.1). Though the ideal switching assumption is inaccurate from the perspective of realistic inverter leg operation, it serves to describe the objectives of the PWM technique implemented.

A different approach to three-phase PWM waveform generation is based on the representation of inverter output voltages in the two-phase frame of reference (d-q system). The transformation from a three-phase to two-phase frame of reference is described by the matrix equation

$$\begin{bmatrix} e_d \\ e_q \end{bmatrix} = \sqrt{2/3} \cdot \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \quad (\text{A4.4})$$

where $\{e_a, e_b, e_c\}$ are the three-phase components, and $\{e_d, e_q\}$ are the corresponding two-phase components. Ideally, inverter output voltages result from one of eight unique inverter switching states depicted in figure A4.1. Using the transformation given in equation (A4.4), the inverter line-to-line output voltages in the d-q frame corresponding to these inverter switching states are given as follows:

State 1:	$v_{ds}^{(1)} = \sqrt{3/2} \cdot V_{dc}$	$v_{qs}^{(1)} = 1/\sqrt{2} \cdot V_{dc}$
State 2:	$v_{ds}^{(2)} = 0$	$v_{qs}^{(2)} = \sqrt{2} \cdot V_{dc}$
State 3:	$v_{ds}^{(3)} = -\sqrt{3/2} \cdot V_{dc}$	$v_{qs}^{(3)} = 1/\sqrt{2} \cdot V_{dc}$
State 4:	$v_{ds}^{(4)} = -\sqrt{3/2} \cdot V_{dc}$	$v_{qs}^{(4)} = -1/\sqrt{2} \cdot V_{dc}$
State 5:	$v_{ds}^{(5)} = 0$	$v_{qs}^{(5)} = -\sqrt{2} \cdot V_{dc}$
State 6:	$v_{ds}^{(6)} = \sqrt{3/2} \cdot V_{dc}$	$v_{qs}^{(6)} = -1/\sqrt{2} \cdot V_{dc}$
State 7:	$v_{ds}^{(7)} = 0$	$v_{qs}^{(7)} = 0$
State 8:	$v_{ds}^{(8)} = 0$	$v_{qs}^{(8)} = 0.$

V_{dc} is the inverter DC power supply voltage. The inverter output d-q voltages for each inverter switching state k , $\{v_{ds}^{(k)}, v_{qs}^{(k)}\}$, are commonly referred to as space vectors. The space vector PWM technique employed in the simulation software converts modulation vector commands $\underline{v}_i^*(t) \equiv \{v_{di}^*(t), v_{qi}^*(t)\}$ to inverter switching states and corresponding state times in PWM sampling intervals equal to $1/2f_s$ (where f_s is the inverter switching frequency).

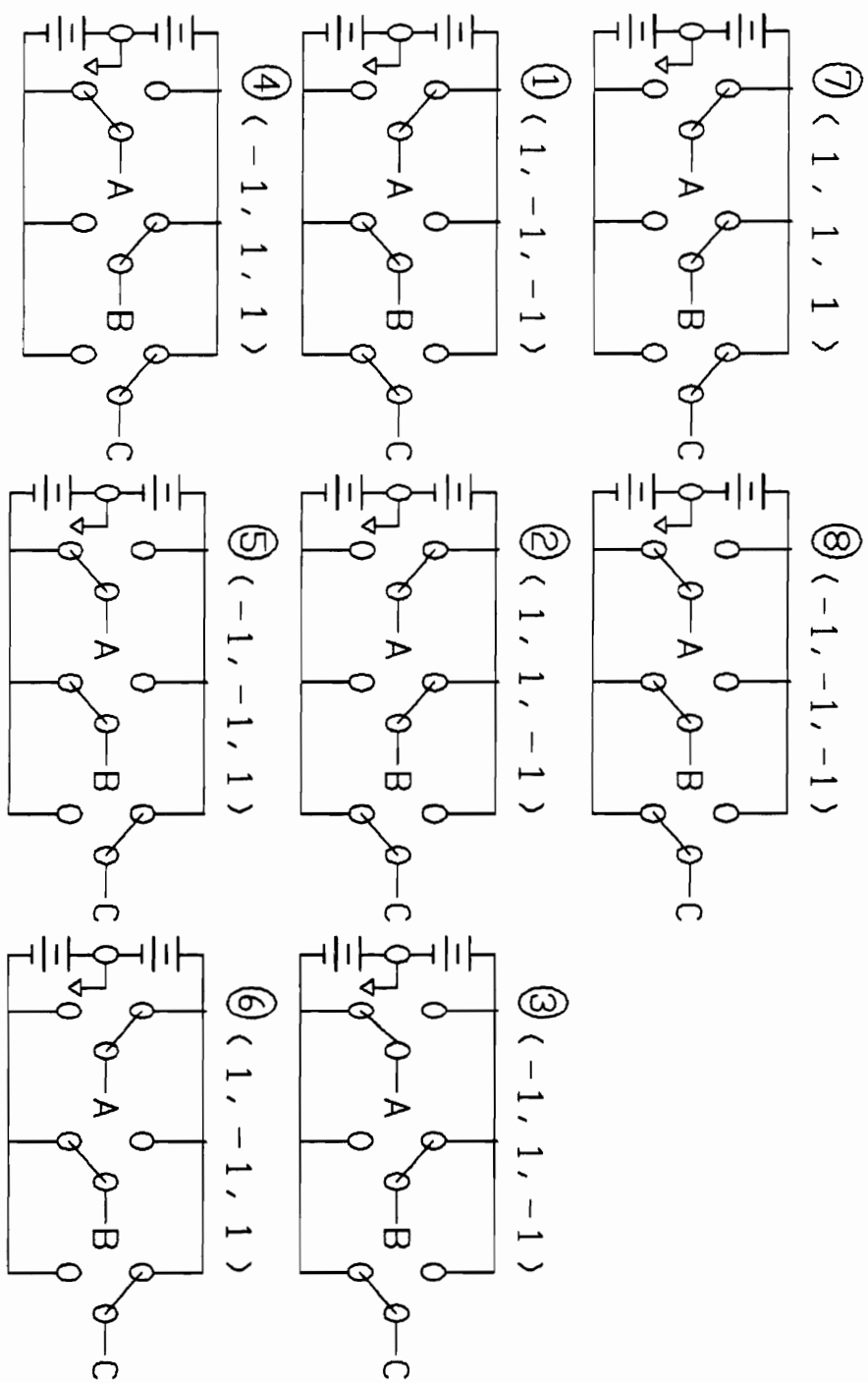


FIGURE A4.1: INVERTER SWITCH CONFIGURATIONS

The least complex space vector PWM implementation derives inverter switching patterns based on the solution of the vector equation

$$2 \cdot f_s \cdot \int_{n \cdot t_s}^{(n + \frac{1}{2}) \cdot t_s} \underline{v}_i(\Omega) d\Omega = \underline{v}_i^*(n \cdot t_s) \quad (\text{A4.5})$$

where the command vector $\underline{v}_i^*(n \cdot t_s) \equiv \{v_{id}^*(n \cdot t_s), v_{iq}^*(n \cdot t_s)\}$, and a $1/2f_s$ asymmetric regular sampling interval is used. The left-hand side of equation (A4.5) is equal to the mean inverter output voltage in the n^{th} $1/2f_s$ interval. In each $1/2f_s$ interval (or modulation sample interval), the inverter output waveform vector $\underline{v}_i(t)$ is generated from the execution of three inverter switching states where one of the three states is the zero vector (either state 7 or 8).

The selection of switching states is based on the premise that inverter switches change state only once during a modulation sample interval. This switching agenda dictates the selection of inverter switching states in the following manner:

- i.) States 7 and 8 are executed in an alternating fashion each modulation sample interval.
- ii.) Assuming the zero vector is executed first in a modulation sample period, the order of the remaining two switching states is determined by the zero vector executed. For example, if state 7 is executed in an interval, then the next switching state executed will have two switches connected to the positive DC rail. The last switching state will have two switches connected to the negative DC rail.
- iii.) Given the command vector $\underline{v}_i^*(n \cdot t_s)$, the selection of the two

non-zero vectors corresponds to the two vectors adjacent to the command vector. Of course, if $\underline{v}_i^*(n \cdot t_s)$ coincides with a non-zero vector, then the execution of the second non-zero vector does not occur.

The algorithm for the selection of the zero vector is simple. Selection of the non-zero vectors is somewhat more complex. The following algorithm is an example of how the selection process would be implemented:

```

if  $v_{di}^* = 0$  and  $v_{qi}^* = 0$ 
    then use only states 7 or 8
if  $v_{di}^* \geq 0$  and  $v_{qi}^* \geq 0$ 
{
    and if  $v_{qi}^* \geq v_{di}^*/\sqrt{3}$ 
        then use states 1 and 2
    else
        use states 1 and 6
}
if  $v_{di}^* \leq 0$  and  $v_{qi}^* \geq 0$ 
{
    and if  $v_{qi}^* \geq |v_{di}^*|/\sqrt{3}$ 
        then use states 2 and 3
    else
        use states 3 and 4
}

```

```

if  $v_{di}^* \leq 0$  and  $v_{qi}^* \leq 0$ 
{
    and if  $|v_{qi}^*| \geq |v_{di}^*|/\sqrt{3}$ 
        then use states 4 and 5
    else
        use states 3 and 4
}
if  $v_{di}^* \geq 0$  and  $v_{qi}^* \leq 0$ 
{
    and if  $|v_{qi}^*| \geq |v_{di}^*|/\sqrt{3}$ 
        then use states 5 and 6
    else
        use states 1 and 6
}

```

Based on the relative absolute magnitudes of the d- and q-axis commands, the position of the command vector is determined. Once the three switching vectors have been identified, the corresponding switching state times are computed. Equation (A4.5) is reduced to the vector equation

$$2 \cdot f_s \cdot [\underline{V}_s^{(k)} \cdot t_k + \underline{V}_s^{(k+1)} \cdot t_{k+1}] = \underline{v}_i^*(n \cdot t_s) \quad (A4.6)$$

where $\underline{V}_s^{(k)}$ and $\underline{V}_s^{(k+1)}$ are the adjacent non-zero vectors, and, t_k and t_{k+1} are the corresponding state times. t_k and t_{k+1} are solutions of the pair of equations obtained from the previous vector equation. Finally, the zero vector state time t_0 is the result of the difference equation

$$t_0 = (1/2f_s) - t_k - t_{k+1}. \quad (A4.7)$$

Software module SVM.C contains the routine which mimics the space vector modulation process. The following parameters are input to routine SVM():

$\underline{v}_i^*$ - modulation command vector.

f_s - the inverter switching frequency in Hertz.

f_m - the inverter output waveform modulation frequency in Hertz.

t - simulation time in seconds.

The modulation command vector is sampled by the routine twice per switching period (at the beginning and midpoint of this period). This sampling technique mimics asymmetric regular sampling in real-time. As in the case of routine INVERTER(), routine SVM() returns a pointer to an array of three bipolar logic values corresponding to the three-phase PWM control signals for phases A, B, and C.

A4.4 Summary

The primary objective in writing the modules which mimic carrier and space vector modulation processes was to produce a driver to control the GTO models in the inverter leg simulation. These routines may be used to directly generate inverter output waveforms if the ideal switching assumption is applied. The Fourier analysis of such waveforms yields the spectral distortion inherent to a specific modulation technique. This has been the traditional approach to the study of modulation processes. On the other hand, the other distortion mechanisms present in the inverter/load system are not considered if the ideal switching assumption is applied.

Inverter/Load Simulation Software

The objective of software design is the same as that of hardware design, efficient and effective design performance. For this reason, the C programming language was selected to implement the software design of the inverter/load simulation.

A5.1 Simulation Software Input

All simulation process data is organized into one of several data structures defined in the following header files:

- INVSIM.H - inverter/load simulation configuration data.
- DIODE.H - diode model data.
- GTO.H - GTO model data.
- INVERTER.H - three-phase inverter model data.
- LOAD.H - load model data.

A structure is a collection of one or more variables, possibly of different types, grouped together under a single name for the purpose of efficient software design (In Pascal, structures are referred to as records). Structures help to organize complicated data, particularly in large programs, because in many situations they permit a group of related variables to be treated as a unit instead of separate entities. Though the use of the structure definitions given in the header files listed above, the overhead involved in transferring program data between

routines, as well as, to and from an external program interface is greatly reduced.

A5.2 GTO Model Data Structure

GTO model data is organized into data structures, one structure per GTO in the inverter bridge. The format of this structure is given in a header file designated as GTO.H. Data in each structure may be categorized into three major groups:

- Data obtained from an external input such as from a file or console input. Device data specifications and PWM control signal delays are placed in this category.
- Model data generated by the GTO model initialization routine. This data is generated from the device data specifications.
- Data utilized by the GTO model driver and inverter leg simulation routines.

Data obtained from device data specifications are listed as follows:

- T_{gt} - GTO turn-on time.
- T_{gq} - GTO turn-off time.
- V_{drm} - Peak repetitive forward blocking voltage.
- I_{drm} - Peak forward anode leakage current.
- V_{tm} - Peak device on-state voltage.
- I_{tgq} - Peak anode on-state current.
- V_{test} - Anode-cathode voltage during turn-on and turn-off

time measurement.

- I_{test} - Anode current during turn-on and turn-off time measurement.
- dV_{dt} - Critical rate-of-rise of device voltage.
- dI_{dt} - Critical rate-of-rise of anode current.
- dI_{rg_dt} - Rate-of-rise of reverse gate current specified with the GTO turn-off time.
- C_S - Snubber capacitance specified with GTO turn-off time and power-loss measurements.
- L_S - Inductance in series with the specified snubber capacitance.
- G_{off} - GTO turn-off gain.

In addition to device data specifications, two structure variables require external input:

- pwn_on_delay - PWM control signal on-level delay time. This variable represents the time delays imposed on the GTO gating signal when the signal makes its transition from an off to on level.
- pwm_off_delay - PWM control signal off-level delay time. This variable represents the time delays imposed on the GTO gating signal when the signal makes its transition from an on to off level.

Though these variables are not strictly part of the GTO device model, they directly affect the operation of the GTO. Their placement in the GTO model data structure is then appropriate.

Using the device specification data, the following variables in the data structure are derived:

- | T_{gr} - Anode current rise time during GTO turn-on.
- | T_{gd} - GTO turn-on delay time.
- | I_{rgm} - Maximum reverse gate current.
- | T_{dq} - GTO turn-off delay.
- | T_{fq} - GTO anode current fall time during turn-off.
- | T_z - GTO tail current period.
- | T_{z1} - GTO anode-cathode voltage rise time during turn-off.

The remaining data structure variables are utilized by the GTO model driver routine:

- | $turn\_on\_enable$ - GTO turn-on process switch.
- | $turn\_off\_enable$ - GTO turn-off process switch.
- | $turn\_on\_timer$ - Timer used to trigger events in the GTO turn-on model.
- | $turn\_off\_timer$ - Timer used to trigger events in the GTO turn-off model.
- | R_{AK} - GTO model resistance branch value.
- | V_{AK} - GTO model independent voltage source branch value.

R_{AK} and V_{AK} are used by the inverter leg simulation routine, and are modified by the GTO model driver.

During simulation start-up, memory is allocated for six of these data structures. After memory allocation, some method of loading device data specifications and PWM control signal delays into each of the six

structures must be implemented (this is an issue of interfacing the program to the console. Though important, it is not strictly part of the simulation process). Routine INIT_GTO_MODEL() is then executed for each of the six GTO model data structures (this routine loads the remainder of the structure variables).

Pointers to memory allocated for the six GTO model data structures are loaded into the S1A, S2A, S1B, S2B, S1C, and S2C variables of the inverter model structure (see header file INVERTER.H).

A5.3 Diode Model Data Structure

Each inverter leg uses four diodes (refer to figure A2.1), where diodes D_1 and D_2 are the diodes in anti-parallel with the GTO's, and diodes D_3 and D_4 are part of the leg snubber circuitry. For each diode model, there is a data structure as defined in header file DIODE.H.

Elements of this structure are defined as follows:

C_d - diode model capacitor.

R_d - diode model resistor.

V_d - diode model independent voltage source.

C_{df} - model capacitor constant capacitance value.

R_f - model resistor forward conduction resistance.

R_r - model resistor reverse-blocking resistance.

Values for variables C_{df} , R_f , and R_r come from device data specifications associated with the diodes implemented in the inverter

leg design. Variables C_d , R_d , and V_d are modified by the diode model driver routine, and are used by the inverter leg simulation routine.

During simulation start-up, memory is allocated for twelve of these data structures. After memory allocation, some method of loading C_{df} , R_f , and R_r values into each of the twelve structures must be implemented (as in the case of the GTO model, this is an issue of interfacing the simulation program to the console).

Pointers to memory allocated for the twelve diode model data structures are loaded into the D1A, D2A, D3A, D4A, D1B, D2B, D3B, D4B, D1C, D2C, D3C, and D4C variables of the inverter model structure (see header file INVERTER.H).

A5.4 Inverter Model Data Structure

Data used by the inverter bridge simulation routine is stored in a structure defined in header file designated as INVERTER.H. In addition to the pointers to GTO and diode model data structures, this structure contains the following variables:

- V_{dc} - The DC power supply voltage.
- I_a, I_b, I_c - The inverter leg load currents.
- L_s - The inverter leg snubber inductance.
- C_s - The inverter leg snubber capacitance.
- C_o - Snubber capacitance used in the snubber energy dissipation/recovery circuit (see figure A2.1).
- R_s - Snubber circuit resistance used in the snubber

dissipation/recovery circuit (see figure A2.1).

On simulation start-up, memory is allocated for one of these structures. The previously defined variables are initialized by a console interface routine.

A5.5 Load Model Data Structure

A header file designated as LOAD.H contains structure definitions for passive and non-linear three-phase load types. Structure type `NONLINEAR_LOAD_3` is used for non-linear load data variables which are defined as follows:

- L_f - Inverter series filter inductance.
- C_f - Inverter filter line-to-line capacitance.
- R_1 - Uncontrolled three-phase rectifier load resistance.
- C_1 - Uncontrolled three-phase rectifier load capacitance.

Refer to figure A3.2 for a depiction of the nonlinear load circuitry.

Structure type `LINEAR_LOAD_3` is used for passive inductive load variables which are defined as follows:

- L_f - Inverter series filter inductance.
- C_f - Inverter filter line-to-line capacitance.
- R_{1_A} , R_{1_B} , R_{1_C} - Line-to-line load resistances.
- PF_A , PF_B , PF_C - Load inductive power factors (given per phase).
- f_m - Load nameplate current/voltage fundamental frequency in Hertz.

Refer to figure A3.1 for a depiction of the passive inductive load circuitry.

On simulation start-up, memory is allocated for the data structure associated with the load type used in the simulation. Structure data is initialized by a console interface routine. The pointer to the memory allocated for this structure is loaded into the variable `load_func` in the `simulation_data` structure defined in header file designated as `INVSIM.H`.

A5.6 Inverter/Load Simulation Data Structure

Header file `INVSIM.H` contains the definitions for the data structure used by the inverter/load simulation routine `SIMULATION()`. The contents of this structure is used to configure the simulation process. The variables in this structure are defined as follows:

- | f_s - Inverter switching frequency in Hertz.
- | f_m - Modulation process fundamental frequency.
- | M - Modulation index.
- | $M3$ - Third harmonic distortion injection index.
- | `phase` - Inverter modulation process phase displacement in radians.
- | `inverter_func` - Function pointer to the inverter emulation program.
- | `mod_func` - Function pointer to the modulation method.
- | `load_func` - Function pointer to the load simulation.

▪ `load_data` - Pointer to the load model data.

▪ `sample_strategy` - modulation index sample strategy indication.

This variable is used by routine `M_sampler()` in software module `INVERTER.C` to determine the method used to sample the modulation index command prior to its use by the modulation process.

▪ `del_t` - Simulation time increment in seconds.

On simulation start-up, memory is allocated for one structure of the type described in `INVSIM.H`. All variables in this structure, except for `load_func`, are initialized from a console interface routine.

A5.7 Simulation Start-up

A summary of a simulation start-up format is given as follows:

- i.) Allocate memory for data structures defined in `INVSIM.H`, `LOAD.H`, and `INVERTER.H`.
- ii.) Allocate memory for six structures of the type defined in `GTO.H`, and twelve structures of the type defined in `DIODE.H`.
- iii.) Execute routine `INIT_GTO_MODEL()` for each of the six GTO model data structures allocated.
- iv.) Store pointers to GTO model data structures into the `S1A` through `S2C` variable locations of the structure defined in `INVERTER.H`.
- v.) Store pointers to diode model data structures into the `D1A` through `D4C` variable locations of the structure defined in `INVERTER.H`.

- vi.) At this point, input from either the keyboard or from a file is required. Assuming that a program exists for this purpose, input from either the keyboard or a file (input from the console) is used to initialize structure variables as described in the previous sections. Typically, electrical component values and time delays would be entered directly. The selection of modulation technique, modulation index sampling strategy, and load model type would be handled typically through some form of menu selection, with the appropriate entry into the corresponding data structure resulting.
- vii.) Initialize diode model structure variables R_d , C_d , and V_d with R_r , C_{df} , and 0 respectively.
- viii.) Based on the console input for the variable `load_func` in the structure defined in the header file `INVSIM.H`, either the routine `INITIALIZE_PASSIVE_LOAD_3()` or `INITIALIZE_NONLINEAR_LOAD_3()` is executed in order to allocate memory for the load model. Routine `INITIALIZE_INVERTER_LEG()` is executed in order to allocate memory for the inverter bridge model.
- ix.) Based on the console input for the variable `load_func` in the structure defined in header file `INVSIM.H`, the variable `load_data` is loaded with the pointer to the corresponding load model data structure.
- x.) Prior to executing the simulation, data arrays containing load and inverter circuit information (currents and voltages) are initialized by the calling program (program calling the simulation

routine). Typically, data from the previous simulation process is loaded into these arrays. If no previous simulation process was executed, or previous data is not used, the calling program should initialize data arrays with values corresponding to a cold-start of the simulation process.

After these events are completed, the program is set to execute the inverter/load simulation process.

A5.8 Inverter/Load Simulation Process

The inverter/load simulation is executed by a function call to routine `SIMULATION()`. Figure A5.1 diagrams the processes operating in this simulation. Inputs to this routine are pointers to the simulation data structure and the inverter model structure, and the simulation time in seconds.

The inverter bridge simulation is the first process executed. Inverter leg load currents are loaded with the load model inputs currents obtained from the previous load model simulation. Note that an array `Y` contains the node and branch voltages, branch currents, fluxes and charges computed from the last load simulation. If the load model is the nonlinear load, `Y[8]`, `Y[9]`, `Y[10]` contain the load model input currents. If the load model is the passive inductive load, `Y[6]`, `Y[7]`, and `Y[8]` contain the load model input currents. The inverter bridge simulation is executed by making a function call to routine

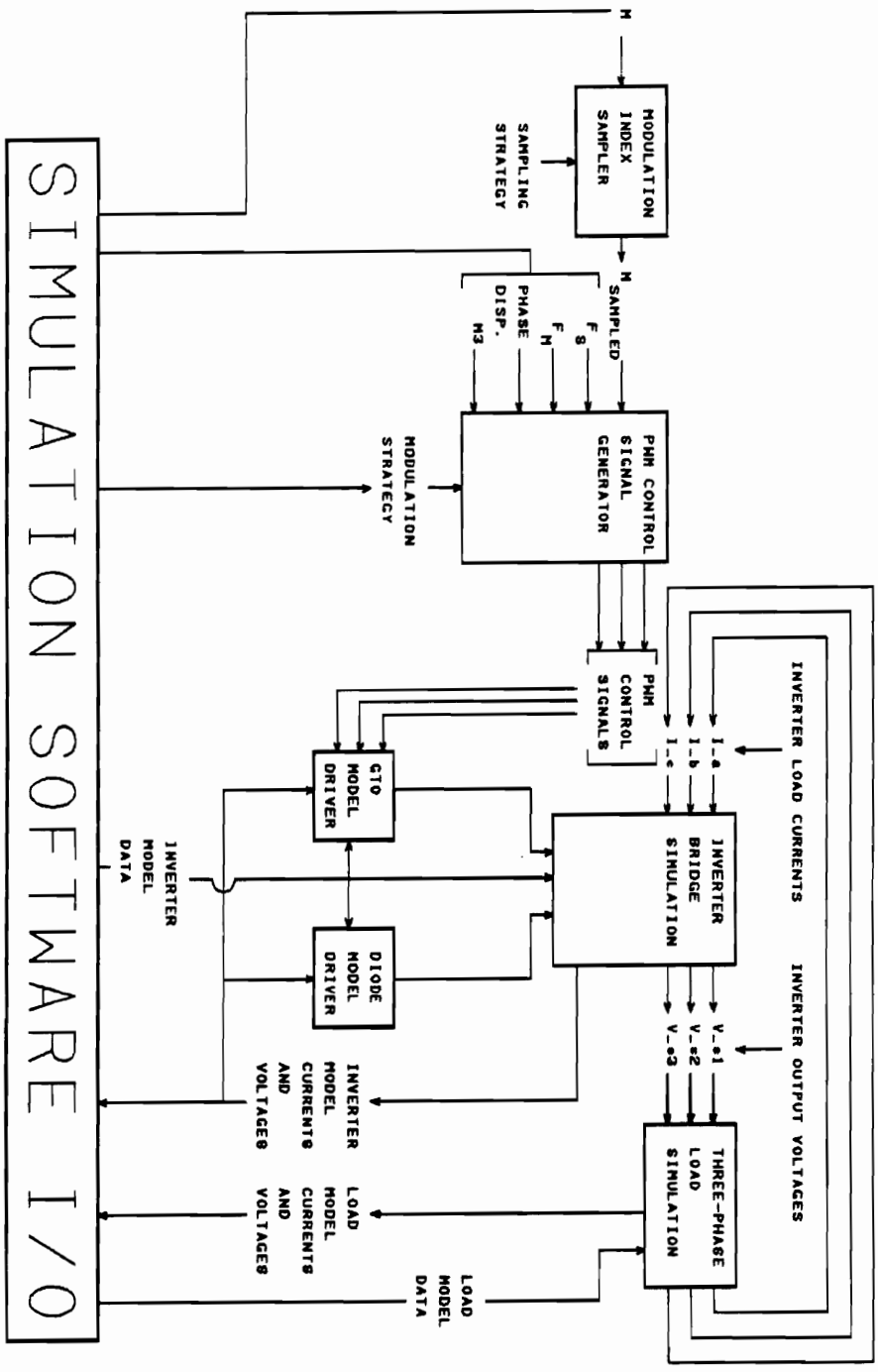


FIGURE A5.1: SIMULATION SOFTWARE STRUCTURE

INVERTER_LEG_SIMULATION()). This routine returns a pointer to an array containing the inverter bridge node and branch voltages, branch currents, fluxes and charges. This pointer initializes an array pointer X1 which points to the phase A leg circuit data. Array pointers X2 and X3 are initialized with the addresses of X1[65] and X2[65] respectively. These pointers point to phase B and C leg circuit data respectively.

After running the inverter bridge simulation, the load model simulation is executed (routine pointed to by load_func). Load model voltages and currents are stored in an array pointed to by the array pointer Y. Along with the load model data, the inverter leg output node voltages (in X1[6], X2[6], and X3[6]) are passed to the load model simulation.

After running the load model simulation, the modulation index is sampled in a fashion indicated by the structure variable sample_strategy. The routine M_sampler() executes the sampling process (note that if the sample_strategy = NATURAL_SAMPLING, the modulation index is not sampled). A control strategy is then executed. If the control strategy selected is the open-loop mode (no closed-loop control strategy selected), the modulation index is passed directly to an inverter modulation process. Otherwise, program control is passed to the routine executing the selected control strategy. The selected control strategy either outputs a modulation index command to a selected modulation process, or executes a specific modulation process as part of the control strategy routine.

If the pointer to the inverter emulation process, structure variable `inverter_func`, is equal to `svm`, the space vector modulation process is executed. Otherwise, one of the carrier PWM routines is executed; `mod_func` points to the function executing the modulation process. Routine `INVERTER()` executes the carrier PWM process in a three-phase manner. Both `SVM()` and `INVERTER()` return pointers to an array containing the three bipolar logic levels representing the order of a three-phase PWM signal generator.

Note: The carrier PWM modulation processes `NSPWM()`, `SSPWM1()`, `SSPWM2()`, `ASPWM1()`, and `ASPWM2()` assume specific modulation index sampling strategies as listed below:

`NSPWM(): NATURAL_SAMPLING`

`SSPWM1(): NEGATIVE_PEAK_SAMPLING`

`SSPWM2(): POSITIVE_PEAK_SAMPLING`

`ASPWM1(): PEAK_SAMPLING`

`ASPWM2(): ZERO_CROSS_SAMPLING`

As a precaution against the selection of an inappropriate sampling technique, the input from the console of a specific carrier PWM technique should automatically select the modulation index strategy. There is no restriction on sampling strategy when the Space Vector modulation process is executed (though typically either the equivalent of either `PEAK_SAMPLING` or `ZERO_CROSS_SAMPLING` is used).

After the modulation process is executed, all twelve diode models are updated using inverter bridge model simulation results. Using the PWM control signals generated by the modulation process (in array

pwm_signals) and inverter bridge model simulation results, the six GTO models are updated. Note that the three bipolar logic signals in array pwm_signals are converted to six unipolar (0 or 1) logic signals. Negative logic signals enable the negative-rail GTO (logic 1) and disable the positive-rail GTO (logic 0). The reverse is true for positive logic signals.

Finally, the simulation program executes the program control routine SIM_KEEPER(), and increments the simulation time t by the simulation interval time del_t if SIM_KEEPER() returns zero. The simulation then repeats the inverter/load simulation. If routine SIM_KEEPER() returns a non-zero value, the simulation process is terminated.

A5.9 The Program Control Routine

SIM_KEEPER() is the run-time interface between the inverter/load simulation and other routines such as console I/O. Passed to SIM_KEEPER() are pointers to inverter bridge and load model simulation data results (currents and voltages), the bipolar logic signal output array from the modulation process, and the simulation time. Two primary uses of this routine are listed below:

- Data passed to the routine is either stored on disk for later use and/or used in a simulation of closed-loop control for the inverter. If a closed-loop control simulation is executed, a

function call to the routine generating the simulation is made from SIM_KEEPER().

- Run-time control of the simulation from the console may be executed through SIM_KEEPER().

As in the case of simulation data input from the console, several different approaches may be taken to implement the functions of the routine SIM_KEEPER().

Appendix 6

The Modified Backpropagation Learning Algorithm

A6.1 Preliminaries

The inverter neurocontroller is implemented in a microprocessor-based system. The controller design is made in the d-q frame where the transformation from the three-phase to the stationary d-q frame is performed using the equation

$$\begin{bmatrix} e_d \\ e_q \end{bmatrix} = \sqrt{(2/3)} \cdot \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \quad (\text{A6.1})$$

where $\{e_a, e_b, e_c\}$ are the three-phase components, and $\{e_d, e_q\}$ are the corresponding d-q frame components. Neurocontroller processes are executed within uniform intervals equal to $1/2f_s$ where f_s is the inverter switching frequency.

The sampled inverter neurocontroller inputs are the logic-level load voltage commands $\{\underline{v}_{dl}^*(k), \underline{v}_{ql}^*(k)\}$, and the logic-level load current feedback in the form of

$$\{\underline{i}_{dl}^{(q)}(k) - \underline{i}_{dl}^{(q)}(k-1), \underline{i}_{ql}^{(q)}(k) - \underline{i}_{ql}^{(q)}(k-1)\}$$

where $\{\underline{i}_{dl}^{(q)}(k), \underline{i}_{ql}^{(q)}(k)\} = \{\underline{i}_{ql}^{(1)}(k), -\underline{i}_{dl}^{(1)}(k)\}$, and the load currents $\{\underline{i}_{dl}^{(1)}(k), \underline{i}_{ql}^{(1)}(k)\}$ are the d-q transformations of the three-phase load current difference terms $\{[i_{1a}(t) - i_{1b}(t)], [i_{1b}(t) - i_{1c}(t)], [i_{1c}(t) - i_{1a}(t)]\}$ at $t = k \cdot T$. The load voltage commands are in the form

$$\underline{v}_{dl}^*(k) = M \cdot \sin(2\pi \cdot f_m \cdot k \cdot T) \quad (\text{A6.2})$$

and

$$\underline{v}_{q1}^*(k) = -M \cdot \cos(2\pi \cdot f_m \cdot k \cdot T) \quad (\text{A6.3})$$

where M is the modulation index, $T = 1/2f_s$, and f_m is the inverter's modulation frequency. Inverter neurocontroller outputs during the k^{th} controller sampling period are the inverter modulation commands $\{\underline{v}_{di}^*(k), \underline{v}_{qi}^*(k)\}$ supplied to the space vector PWM process.

The single-layer inverter neurocontroller discrete-time forward transfer characteristics are described by the equations

$$\begin{aligned} \underline{v}_{di}^*(k) = & w_{11}(k) \cdot \underline{v}_{d1}^*(k) + w_{12}(k) \cdot \underline{v}_{q1}^*(k) + \\ & w_{13}(k) \cdot [\underline{i}_{d1}^{(q)}(k) - \underline{i}_{d1}^{(q)}(k-1)] + \\ & w_{14}(k) \cdot [\underline{i}_{q1}^{(q)}(k) - \underline{i}_{q1}^{(q)}(k-1)] \text{ and} \end{aligned} \quad (\text{A6.4})$$

$$\begin{aligned} \underline{v}_{qi}^*(k) = & w_{21}(k) \cdot \underline{v}_{d1}^*(k) + w_{22}(k) \cdot \underline{v}_{q1}^*(k) + \\ & w_{23}(k) \cdot [\underline{i}_{d1}^{(q)}(k) - \underline{i}_{d1}^{(q)}(k-1)] + \\ & w_{24}(k) \cdot [\underline{i}_{q1}^{(q)}(k) - \underline{i}_{q1}^{(q)}(k-1)], \end{aligned} \quad (\text{A6.5})$$

where k is a positive integer, $w_{ij}(k)$ are the neurocontroller weights in the k^{th} controller sampling interval. Via the neurocontroller learning process, neurocontroller weights are modified with the objective of reducing the load voltage errors

$$\underline{\epsilon}_{dv}(k) = \underline{v}_{d1}^*(k) - \underline{v}_{d1}^{(q)}(k) \quad (\text{A6.6})$$

and

$$\underline{\epsilon}_{qv}(k) = \underline{v}_{q1}^*(k) - \underline{v}_{q1}^{(q)}(k) \quad (\text{A6.7})$$

where $\{\underline{v}_{d1}^{(q)}(k), \underline{v}_{q1}^{(q)}(k)\}$ are the sampled logic-level load voltages and $\{\underline{v}_{d1}^{(q)}(k), \underline{v}_{q1}^{(q)}(k)\} = \{\underline{v}_{q1}(k), -\underline{v}_{d1}(k)\}$.

Inverter output voltage waveforms resulting from the neurocontroller outputs $\{\underline{v}_{di}^*(k), \underline{v}_{qi}^*(k)\}$, the space vector PWM process,

and three-phase inverter switching synthesize load voltages in the output filter (depicted in figure 1.2) as described by the discrete-time state transition equations

$$\begin{aligned}\underline{v}_{dl}(k+1) = & \cos(2\pi \cdot f_f \cdot T) \cdot \underline{v}_{dl}(k) + \\ & [1/(2\pi \cdot f_f)] \cdot \sin(2\pi \cdot f_f \cdot T) \cdot \underline{v}_{dl}'(k) + \\ & [1 - \cos(2\pi \cdot f_f \cdot T)] \cdot [\underline{v}_{di}(k) - L_f \cdot \underline{i}_{dl}^{(1)'}(k)]\end{aligned}\quad (A6.8)$$

$$\begin{aligned}\underline{v}_{ql}(k+1) = & \cos(2\pi \cdot f_f \cdot T) \cdot \underline{v}_{ql}(k) + \\ & [1/(2\pi \cdot f_f)] \cdot \sin(2\pi \cdot f_f \cdot T) \cdot \underline{v}_{ql}'(k) + \\ & [1 - \cos(2\pi \cdot f_f \cdot T)] \cdot [\underline{v}_{qi}(k) - L_f \cdot \underline{i}_{ql}^{(1)'}(k)]\end{aligned}\quad (A6.9)$$

where T is the discretization interval, L_f is the output filter series inductance, C_f is the filter line-to-line capacitance, $2\pi \cdot f_f = 1/(3 \cdot L_f \cdot C_f)$, $\{\underline{v}_{dl}(k), \underline{v}_{ql}(k)\}$ are the normalized load voltages, $\{\underline{v}_{dl}'(k), \underline{v}_{ql}'(k)\} = (d/dt) \cdot \{\underline{v}_{dl}(k), \underline{v}_{ql}(k)\}$, $\{\underline{v}_{di}(k), \underline{v}_{qi}(k)\}$ are the normalized inverter output voltages, $\{\underline{i}_{dl}^{(1)'}(k), \underline{i}_{ql}^{(1)'}(k)\}$ are the d-q transformations of $(d/dt) \cdot \{[i_{1a}(t) - i_{1b}(t)], [i_{1b}(t) - i_{1c}(t)], [i_{1c}(t) - i_{1a}(t)]\}$ at $t = k \cdot T$, and the currents and voltages are normalized with respect to the inverter DC power supply voltage. The inverter output voltages $\{\underline{v}_{di}(k), \underline{v}_{qi}(k)\}$ are generated from inverter modulation commands $\{\underline{v}_{di}^*(k-1), \underline{v}_{qi}^*(k-1)\}$ where a one controller sampling interval delay in modulation command execution is realized due to space vector PWM process. Inverter modulation commands $\{\underline{v}_{di}^*(k-1), \underline{v}_{qi}^*(k-1)\}$ result in load voltages $\{\underline{v}_{dl}(k+1), \underline{v}_{ql}(k+1)\}$.

The neurocontroller learning process modifies the neurocontroller weights and the inverter modulation command trajectory $\{\underline{v}_{di}^*(k), \underline{v}_{qi}^*(k)\}$ with the objective of minimizing the function $\frac{1}{2} \cdot [\underline{\epsilon}_{dv}^2 + \underline{\epsilon}_{qv}^2]$. Changes

in the load voltage trajectory $\{\underline{v}_{d1}(k), \underline{v}_{q1}(k)\}$ during the neurocontroller learning process due to the inverter modulation command trajectory $\{\underline{v}_{di}^*(k), \underline{v}_{qi}^*(k)\}$ are dependent on the inverter forward transfer characteristics and the output filter response. Assuming ideal inverter operation or no switching distortion effects, load voltage state transition equations including single-layer neurocontroller forward transfer characteristics are given as follows:

$$\begin{aligned} \underline{v}_{d1}(k) = & \cos(2\pi \cdot f_f \cdot T) \cdot \underline{v}_{d1}(k-1) + \\ & [1/(2\pi \cdot f_f)] \cdot \sin(2\pi \cdot f_f \cdot T) \cdot \underline{v}_{d1}'(k-1) + \\ & [1 - \cos(2\pi \cdot f_f \cdot T)] \cdot \\ & \{ w_{11}(k-3) \cdot \underline{v}_{d1}^*(k-2) + \\ & w_{12}(k-3) \cdot \underline{v}_{q1}^*(k-2) + \\ & w_{13}(k-3) \cdot [\underline{i}_{d1}(k-2) - \underline{i}_{d1}(k-3)] + \\ & w_{14}(k-3) \cdot [\underline{i}_{q1}(k-2) - \underline{i}_{q1}(k-3)] - \\ & L_f \cdot \underline{i}_{d1}^{(1)'}(k-1) + \\ & \delta w_{11}(k-2) \cdot \underline{v}_{d1}^*(k-2) + \\ & \delta w_{12}(k-2) \cdot \underline{v}_{q1}^*(k-2) + \\ & \delta w_{13}(k-2) \cdot [\underline{i}_{d1}^{(1)}(k-2) - \underline{i}_{d1}^{(1)}(k-3)] + \\ & \delta w_{14}(k-2) \cdot [\underline{i}_{q1}^{(1)}(k-2) - \underline{i}_{q1}^{(1)}(k-3)] \} \quad (A6.10) \end{aligned}$$

$$\begin{aligned} \underline{v}_{q1}(k) = & \cos(2\pi \cdot f_f \cdot T) \cdot \underline{v}_{q1}(k-1) + \\ & [1/(2\pi \cdot f_f)] \cdot \sin(2\pi \cdot f_f \cdot T) \cdot \underline{v}_{q1}'(k-1) + \\ & [1 - \cos(2\pi \cdot f_f \cdot T)] \cdot \\ & \{ w_{21}(k-3) \cdot \underline{v}_{d1}^*(k-2) + \\ & w_{22}(k-3) \cdot \underline{v}_{q1}^*(k-2) + \\ & w_{23}(k-3) \cdot [\underline{i}_{d1}(k-2) - \underline{i}_{d1}(k-3)] + \end{aligned}$$

$$\begin{aligned}
& w_{24}(k-3) \cdot [\underline{i}_{q1}(k-2) - \underline{i}_{q1}(k-3)] - \\
& L_f \cdot \underline{i}_{q1}^{(1)}(k-1) + \\
& \delta w_{21}(k-2) \cdot \underline{v}_{d1}^*(k-2) + \\
& \delta w_{22}(k-2) \cdot \underline{v}_{q1}^*(k-2) + \\
& \delta w_{23}(k-2) \cdot [\underline{i}_{d1}^{(1)}(k-2) - \underline{i}_{d1}^{(1)}(k-3)] + \\
& \delta w_{24}(k-2) \cdot [\underline{i}_{q1}^{(1)}(k-2) - \underline{i}_{q1}^{(1)}(k-3)] \} \quad (A6.11)
\end{aligned}$$

where δw_{ij} are the neurocontroller weight modifications due to the learning process where $w_{ij}(k+1) = w_{ij}(k) + \delta w_{ij}(k)$. Neurocontroller weight modifications based on load voltage errors $\{\underline{\epsilon}_{dv}(k), \underline{\epsilon}_{qv}(k)\}$ do not affect the load voltage trajectory until after a delay of two controller sampling intervals. Additionally, the output filter response in the form of the term $[1 - \cos(2\pi \cdot f_f \cdot T)]$ attenuates the change due to the weight modifications. Changes created by the neurocontroller learning process in the load voltage trajectory are described by the equations

$$\begin{aligned}
\delta \underline{v}_{d1}(k) = [1 - \cos(2\pi \cdot f_f \cdot T)] \cdot \\
\{ \delta w_{11}(k-2) \cdot \underline{v}_{d1}^*(k-2) + \\
\delta w_{12}(k-2) \cdot \underline{v}_{q1}^*(k-2) + \\
\delta w_{13}(k-2) \cdot [\underline{i}_{d1}^{(1)}(k-2) - \underline{i}_{d1}^{(1)}(k-3)] + \\
\delta w_{14}(k-2) \cdot [\underline{i}_{q1}^{(1)}(k-2) - \underline{i}_{q1}^{(1)}(k-3)] \} \quad (A6.12)
\end{aligned}$$

and

$$\begin{aligned}
\delta \underline{v}_{q1}(k) = [1 - \cos(2\pi \cdot f_f \cdot T)] \cdot \\
\{ \delta w_{21}(k-2) \cdot \underline{v}_{d1}^*(k-2) + \\
\delta w_{22}(k-2) \cdot \underline{v}_{q1}^*(k-2) + \\
\delta w_{23}(k-2) \cdot [\underline{i}_{d1}^{(1)}(k-2) - \underline{i}_{d1}^{(1)}(k-3)] + \\
\delta w_{24}(k-2) \cdot [\underline{i}_{q1}^{(1)}(k-2) - \underline{i}_{q1}^{(1)}(k-3)] \}. \quad (A6.13)
\end{aligned}$$

A6.2 The Conventional Backpropagation Algorithm

Using the conventional backpropagation learning algorithm, neurocontroller weights are modified based on the use of the equations

$$w_{ij}(k + 1) = w_{ij}(k) - c(k) \cdot \frac{1}{2} \cdot \delta [\underline{\epsilon}_{dv}^2 + \underline{\epsilon}_{qv}^2] / \delta w_{ij}, \quad (A6.14)$$

$$w_{1j}(k + 1) = w_{1j}(k) + c(k) \cdot \underline{\epsilon}_{dv}(k) \cdot \delta v_{d1} / \delta w_{1j}, \quad (A6.15)$$

and

$$w_{2j}(k + 1) = w_{2j}(k) + c(k) \cdot \underline{\epsilon}_{qv}(k) \cdot \delta v_{q1} / \delta w_{2j} \quad (A6.16)$$

where $c(k)$ is the learning-rate parameter. If ideal PWM inverter operation is assumed, $\{\delta v_{d1} / \delta w_{1j}, \delta v_{q1} / \delta w_{2j}\} = \alpha \cdot \{\delta v_{di}^* / \delta w_{1j}, \delta v_{qi}^* / \delta w_{2j}\}$ where α is a constant. Neurocontroller weight modifications each k^{th} controller sampling interval are described by the equations

$$\delta w_{11}(k) = c(k) \cdot \underline{\epsilon}_{dv}(k) \cdot \underline{v}_{d1}^*(k - 2) \quad (A6.17)$$

$$\delta w_{12}(k) = c(k) \cdot \underline{\epsilon}_{dv}(k) \cdot \underline{v}_{q1}^*(k - 2) \quad (A6.18)$$

$$\delta w_{13}(k) = c(k) \cdot \underline{\epsilon}_{dv}(k) \cdot [\underline{i}_{d1}^{(q)}(k - 2) - \underline{i}_{d1}^{(q)}(k - 3)] \quad (A6.19)$$

$$\delta w_{14}(k) = c(k) \cdot \underline{\epsilon}_{dv}(k) \cdot [\underline{i}_{q1}^{(q)}(k - 2) - \underline{i}_{q1}^{(q)}(k - 3)] \quad (A6.20)$$

$$\delta w_{21}(k) = c(k) \cdot \underline{\epsilon}_{qv}(k) \cdot \underline{v}_{d1}^*(k - 2) \quad (A6.21)$$

$$\delta w_{22}(k) = c(k) \cdot \underline{\epsilon}_{qv}(k) \cdot \underline{v}_{q1}^*(k - 2) \quad (A6.22)$$

$$\delta w_{23}(k) = c(k) \cdot \underline{\epsilon}_{qv}(k) \cdot [\underline{i}_{d1}^{(q)}(k - 2) - \underline{i}_{d1}^{(q)}(k - 3)] \quad (A6.23)$$

and

$$\delta w_{24}(k) = c(k) \cdot \underline{\epsilon}_{qv}(k) \cdot [\underline{i}_{q1}^{(q)}(k - 2) - \underline{i}_{q1}^{(q)}(k - 3)] \quad (A6.24)$$

where the constant α is included in the learning-rate parameter value. The neurocontroller inputs in equations (A6.17) through (A6.24) correspond to the generation of load voltages $\{\underline{v}_{ld}(k), \underline{v}_{lq}(k)\}$ and the

resulting errors $\{\underline{\epsilon}_{vd}(k), \underline{\epsilon}_{vq}(k)\}$. Changes in the load voltage trajectory in terms of the neurocontroller learning process are described by the equations

$$\begin{aligned} \delta \underline{v}_{d1}(k) = & [1 - \cos(2\pi \cdot f_f \cdot T)] \cdot c(k-2) \cdot \underline{\epsilon}_{dv}(k-2) \cdot \\ & \{ \underline{v}_{d1}^*(k-4) \cdot \underline{v}_{d1}^*(k-2) + \\ & \underline{v}_{q1}^*(k-4) \cdot \underline{v}_{q1}^*(k-2) + \\ & [\underline{i}_{d1}^{(1)}(k-4) - \underline{i}_{d1}^{(1)}(k-5)] \cdot \\ & [\underline{i}_{d1}^{(1)}(k-2) - \underline{i}_{d1}^{(1)}(k-3)] + \\ & [\underline{i}_{q1}^{(1)}(k-4) - \underline{i}_{q1}^{(1)}(k-5)] \cdot \\ & [\underline{i}_{q1}^{(1)}(k-2) - \underline{i}_{q1}^{(1)}(k-3)] \} \end{aligned} \quad (A6.25)$$

and

$$\begin{aligned} \delta \underline{v}_{q1}(k) = & [1 - \cos(2\pi \cdot f_f \cdot T)] \cdot c(k-2) \cdot \underline{\epsilon}_{qv}(k-2) \cdot \\ & \{ \underline{v}_{d1}^*(k-4) \cdot \underline{v}_{d1}^*(k-2) + \\ & \underline{v}_{q1}^*(k-4) \cdot \underline{v}_{q1}^*(k-2) + \\ & [\underline{i}_{d1}^{(1)}(k-4) - \underline{i}_{d1}^{(1)}(k-5)] \cdot \\ & [\underline{i}_{d1}^{(1)}(k-2) - \underline{i}_{d1}^{(1)}(k-3)] + \\ & [\underline{i}_{q1}^{(1)}(k-4) - \underline{i}_{q1}^{(1)}(k-5)] \cdot \\ & [\underline{i}_{q1}^{(1)}(k-2) - \underline{i}_{q1}^{(1)}(k-3)] \}. \end{aligned} \quad (A6.26)$$

As described by equations (A6.25) and (A6.26), load voltage trajectory changes are functions of the learning-rate parameter, the previous load voltage errors, and load current feedback. Ideally, changes in the load voltage trajectory during the neurocontroller learning process are always made in the direction suggested by the load voltage errors $\{\underline{\epsilon}_{dv}(k), \underline{\epsilon}_{qv}(k)\}$. Given this condition, the load voltage trajectory will eventually reach the desired trajectory

$\{\underline{v}_{d1}^*(k), \underline{v}_{q1}^*(k)\}$. The steady-state convergence of the load voltage trajectory to the desired trajectory $\{\delta \underline{v}_{d1}(k) = 0, \delta \underline{v}_{q1}(k) = 0\}$ is dependent on the learning-rate parameter value and the load current harmonic content.

The use of the conventional backpropagation algorithm does not ensure the condition that load voltage trajectory changes are always made in the direction suggested by $\{\underline{\epsilon}_{dv}(k), \underline{\epsilon}_{qv}(k)\}$. Two problems result from this situation:

- i.) The steady-state load voltage trajectory does not reach the desired trajectory $\{\underline{v}_{d1}^*(k), \underline{v}_{q1}^*(k)\}$.
- ii.) Positive load current feedback creates a steady-state limit cycle oscillation in the trajectory $\{\delta \underline{v}_{d1}(k), \delta \underline{v}_{q1}(k)\}$.

In computer simulations and experiments with the single-layer neurocontroller using the conventional backpropagation learning process and balanced passive loads, steady-state load voltage errors and waveform second and third harmonics were observed.

A6.3 The Modified Backpropagation Algorithm

The modified backpropagation algorithm was devised to force load trajectory changes in the direction suggested by $\{\underline{\epsilon}_{dv}(k), \underline{\epsilon}_{qv}(k)\}$ in each controller sampling interval. Weights are updated in a fashion such that

$$\text{sgn}[\delta \underline{v}_{di}^*(k)] = \text{sgn}[\underline{\epsilon}_{dv}(k)] \quad (\text{A6.27})$$

and

$$\text{sgn}[\delta \underline{v}_{qi}^*(k)] = \text{sgn}[\underline{\epsilon}_{qv}(k)] \quad (\text{A6.28})$$

where the function $\text{sgn}(a)$ is defined as

$$\text{sgn}(a) = 1 \text{ for } a \geq 0 \quad (\text{A6.29})$$

and

$$\text{sgn}(a) = -1 \text{ for } a < 0. \quad (\text{A6.30})$$

Based on the equations (A6.27) and (A6.28), weight changes are constrained in a manner described as follows:

$$\text{sgn}[\delta w_{11}(k) \cdot \underline{v}_{d1}^*(k)] = \text{sgn}[\underline{\epsilon}_{dv}(k)] \quad (\text{A6.31a})$$

$$\text{sgn}[\delta w_{12}(k) \cdot \underline{v}_{q1}^*(k)] = \text{sgn}[\underline{\epsilon}_{dv}(k)] \quad (\text{A6.31b})$$

$$\text{sgn}\{\delta w_{13}(k) \cdot [\underline{i}_{d1}(q)(k) - \underline{i}_{d1}(q)(k-1)]\} = \text{sgn}[\underline{\epsilon}_{dv}(k)] \quad (\text{A6.31c})$$

$$\text{sgn}\{\delta w_{14}(k) \cdot [\underline{i}_{q1}(q)(k) - \underline{i}_{q1}(q)(k-1)]\} = \text{sgn}[\underline{\epsilon}_{dv}(k)] \quad (\text{A6.31d})$$

$$\text{sgn}[\delta w_{21}(k) \cdot \underline{v}_{d1}^*(k)] = \text{sgn}[\underline{\epsilon}_{qv}(k)] \quad (\text{A6.31e})$$

$$\text{sgn}[\delta w_{22}(k) \cdot \underline{v}_{q1}^*(k)] = \text{sgn}[\underline{\epsilon}_{qv}(k)] \quad (\text{A6.31f})$$

$$\text{sgn}\{\delta w_{23}(k) \cdot [\underline{i}_{d1}(q)(k) - \underline{i}_{d1}(q)(k-1)]\} = \text{sgn}[\underline{\epsilon}_{qv}(k)] \quad (\text{A6.31g})$$

$$\text{sgn}\{\delta w_{24}(k) \cdot [\underline{i}_{q1}(q)(k) - \underline{i}_{q1}(q)(k-1)]\} = \text{sgn}[\underline{\epsilon}_{qv}(k)]. \quad (\text{A6.31h})$$

The magnitude of the individual weight changes per controller sampling interval is the same as in the conventional backpropagation algorithm.

Weight updates are described by the equations

$$\begin{aligned} \delta w_{11}(k) &= c(k) \cdot \underline{\epsilon}_{dv}(k) \cdot \\ &\quad |\underline{v}_{d1}^*(k-2)| \cdot \text{sgn}[\underline{v}_{d1}^*(k)] \end{aligned} \quad (\text{A6.32})$$

$$\begin{aligned} \delta w_{12}(k) &= c(k) \cdot \underline{\epsilon}_{dv}(k) \cdot \\ &\quad |\underline{v}_{q1}^*(k-2)| \cdot \text{sgn}[\underline{v}_{q1}^*(k)] \end{aligned} \quad (\text{A6.33})$$

$$\begin{aligned} \delta w_{13}(k) &= c(k) \cdot \underline{\epsilon}_{dv}(k) \cdot \\ &\quad |[\underline{i}_{d1}(q)(k-2) - \underline{i}_{d1}(q)(k-3)]| \cdot \\ &\quad \text{sgn}[\underline{i}_{d1}(q)(k) - \underline{i}_{d1}(q)(k-1)] \end{aligned} \quad (\text{A6.34})$$

$$\begin{aligned}\delta w_{14}(k) &= c(k) \cdot \underline{\epsilon}_{dv}(k) \cdot \\ &| [\underline{i}_{q1}^{(q)}(k-2) - \underline{i}_{q1}^{(q)}(k-3)] | \cdot \\ &\text{sgn}\{[\underline{i}_{q1}^{(q)}(k) - \underline{i}_{q1}^{(q)}(k-1)]\}\end{aligned}\quad (\text{A6.35})$$

$$\begin{aligned}\delta w_{21}(k) &= c(k) \cdot \underline{\epsilon}_{qv}(k) \cdot \\ &| \underline{v}_{d1}^*(k-2) | \cdot \text{sgn}[\underline{v}_{d1}^*(k)]\end{aligned}\quad (\text{A6.36})$$

$$\begin{aligned}\delta w_{22}(k) &= c(k) \cdot \underline{\epsilon}_{qv}(k) \cdot \\ &| \underline{v}_{q1}^*(k-2) | \cdot \text{sgn}[\underline{v}_{q1}^*(k)]\end{aligned}\quad (\text{A6.37})$$

$$\begin{aligned}\delta w_{23}(k) &= c(k) \cdot \underline{\epsilon}_{qv}(k) \cdot \\ &| [\underline{i}_{d1}^{(q)}(k-2) - \underline{i}_{d1}^{(q)}(k-3)] | \cdot \\ &\text{sgn}\{[\underline{i}_{d1}^{(q)}(k) - \underline{i}_{d1}^{(q)}(k-1)]\}\end{aligned}\quad (\text{A6.38})$$

$$\begin{aligned}\delta w_{24}(k) &= c(k) \cdot \underline{\epsilon}_{qv}(k) \cdot \\ &| [\underline{i}_{q1}^{(q)}(k-2) - \underline{i}_{q1}^{(q)}(k-3)] | \cdot \\ &\text{sgn}\{[\underline{i}_{q1}^{(q)}(k) - \underline{i}_{q1}^{(q)}(k-1)]\}.\end{aligned}\quad (\text{A6.39})$$

Changes in the load voltage trajectory in terms of the neurocontroller learning process are described by the equations

$$\begin{aligned}\delta \underline{v}_{d1}(k) &= [1 - \cos(2\pi \cdot f_f \cdot T)] \cdot c(k-2) \cdot \underline{\epsilon}_{dv}(k-2) \cdot \\ &\{ | \underline{v}_{d1}^*(k-4) \cdot \underline{v}_{d1}^*(k-2) | + \\ &| \underline{v}_{q1}^*(k-4) \cdot \underline{v}_{q1}^*(k-2) | + \\ &| [\underline{i}_{d1}^{(1)}(k-4) - \underline{i}_{d1}^{(1)}(k-5)] \cdot \\ &[\underline{i}_{d1}^{(1)}(k-2) - \underline{i}_{d1}^{(1)}(k-3)] | + \\ &| [\underline{i}_{q1}^{(1)}(k-4) - \underline{i}_{q1}^{(1)}(k-5)] \cdot \\ &[\underline{i}_{q1}^{(1)}(k-2) - \underline{i}_{q1}^{(1)}(k-3)] | \} \end{aligned}\quad (\text{A6.40})$$

and

$$\begin{aligned}\delta \underline{v}_{q1}(k) &= [1 - \cos(2\pi \cdot f_f \cdot T)] \cdot c(k-2) \cdot \underline{\epsilon}_{qv}(k-2) \cdot \\ &\{ | \underline{v}_{d1}^*(k-4) \cdot \underline{v}_{d1}^*(k-2) | +\end{aligned}$$

$$\begin{aligned}
& \underline{v}_{q1}^*(k-4) \cdot \underline{v}_{q1}^*(k-2) + \\
& | [\underline{i}_{d1}^{(1)}(k-4) - \underline{i}_{d1}^{(1)}(k-5)] \cdot \\
& [\underline{i}_{d1}^{(1)}(k-2) - \underline{i}_{d1}^{(1)}(k-3)] | + \\
& | [\underline{i}_{q1}^{(1)}(k-4) - \underline{i}_{q1}^{(1)}(k-5)] \cdot \\
& [\underline{i}_{q1}^{(1)}(k-2) - \underline{i}_{q1}^{(1)}(k-3)] | \}. \quad (A6.41)
\end{aligned}$$

Changes in the trajectory $\{\delta \underline{v}_{d1}(k), \delta \underline{v}_{q1}(k)\}$ are always made in the direction suggested by the errors $\{\underline{\epsilon}_{dv}(k), \underline{\epsilon}_{qv}(k)\}$. The influence of the load current feedback is limited to $\{\delta \underline{v}_{d1}(k), \delta \underline{v}_{q1}(k)\}$ amplitude variation. Given at non-zero learning-rate parameter, the load voltage trajectory $\{\underline{v}_{d1}(k), \underline{v}_{q1}(k)\}$ can reach the desired trajectory $\{\underline{v}_{d1}^*(k), \underline{v}_{q1}^*(k)\}$, though the issue of steady-state trajectory convergence is still dependent on the learning-rate parameter value and the load current harmonic content.

In computer simulations and experiments with the single-layer neurocontroller using the modified backpropagation algorithm, the steady-state load fundamental voltage errors and waveform second harmonic generation were eliminated.

Appendix 7

Output Filter Circuit Analysis

The typical three-phase output filter circuit utilized in a sinusoidal voltage uninterruptible power supply or UPS is depicted in figure A7.1. The following equations are derived from an analysis of this circuit:

$$i_{ai} - C_f \cdot (d/dt) \cdot [v_{a1} - v_{b1}] + C_f \cdot (d/dt) \cdot [v_{c1} - v_{a1}] - i_{a1} = 0 \quad (A7.1a)$$

$$i_{bi} - C_f \cdot (d/dt) \cdot [v_{b1} - v_{c1}] + C_f \cdot (d/dt) \cdot [v_{a1} - v_{b1}] - i_{b1} = 0 \quad (A7.1b)$$

$$i_{ci} - C_f \cdot (d/dt) \cdot [v_{c1} - v_{a1}] + C_f \cdot (d/dt) \cdot [v_{b1} - v_{c1}] - i_{c1} = 0 \quad (A7.1c)$$

$$L_f \cdot (d/dt) \cdot i_{ai} = v_{ai} - v_{a1} \quad (A7.2a)$$

$$L_f \cdot (d/dt) \cdot i_{bi} = v_{bi} - v_{b1} \quad (A7.2b)$$

$$L_f \cdot (d/dt) \cdot i_{ci} = v_{ci} - v_{c1} \quad (A7.2c)$$

where i_{ai} , i_{bi} , i_{ci} are the inverter output phase currents, i_{a1} , i_{b1} , i_{c1} are the load phase currents, v_{ai} , v_{bi} , v_{ci} are the inverter output phase-to-ground voltages, and v_{a1} , v_{b1} , v_{c1} are the load phase-to-ground voltages. Defining the phase-to-phase load voltages $v_{abl} = v_{a1} - v_{b1}$, $v_{bcl} = v_{b1} - v_{c1}$, and $v_{cal} = v_{c1} - v_{a1}$, equations (A7.1a) through (A7.1c) are re-written using phase-to-phase load voltage variables:

$$3 \cdot C_f \cdot (d/dt) \cdot v_{abl} = [i_{ai} - i_{bi}] - [i_{a1} - i_{b1}] \quad (A7.3a)$$

$$3 \cdot C_f \cdot (d/dt) \cdot v_{bcl} = [i_{bi} - i_{ci}] - [i_{b1} - i_{c1}] \quad (A7.3b)$$

$$3 \cdot C_f \cdot (d/dt) \cdot v_{cal} = [i_{ci} - i_{ai}] - [i_{c1} - i_{a1}]. \quad (A7.3c)$$

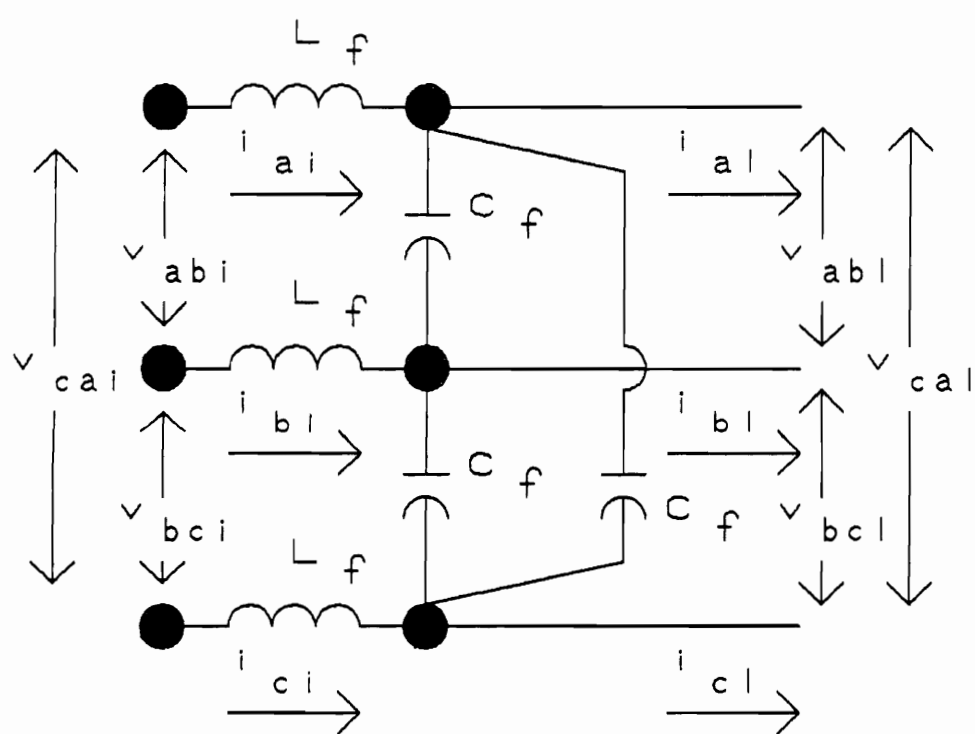


FIGURE A7.1: OUTPUT FILTER CIRCUIT

Defining the phase-to-phase inverter output voltages $v_{abi} = v_{ai} - v_{bi}$, $v_{bci} = v_{bi} - v_{ci}$, and $v_{cai} = v_{ci} - v_{ai}$, equations (A7.2a) through (A7.2c) are re-written using phase-to-phase inverter output voltage variables:

$$L_f \cdot (d/dt) \cdot [i_{ai} - i_{bi}] = v_{abi} - v_{abl} \quad (A7.4a)$$

$$L_f \cdot (d/dt) \cdot [i_{bi} - i_{ci}] = v_{bci} - v_{bcl} \quad (A7.4b)$$

$$L_f \cdot (d/dt) \cdot [i_{ci} - i_{ai}] = v_{cai} - v_{cal}. \quad (A7.4c)$$

The current difference terms are re-defined as follows:

$$i_{abi} = i_{ai} - i_{bi} \quad (A7.5a)$$

$$i_{bci} = i_{bi} - i_{ci} \quad (A7.5b)$$

$$i_{cai} = i_{ci} - i_{ai} \quad (A7.5c)$$

$$i_{abl} = i_{al} - i_{bl} \quad (A7.5d)$$

$$i_{bcl} = i_{bl} - i_{cl} \quad (A7.5e)$$

$$i_{cal} = i_{cl} - i_{al}. \quad (A7.5f)$$

The output filter circuit state equations are:

$$(d/dt) \cdot i_{abi} = (1/L_f) \cdot [v_{abi} - v_{abl}] \quad (A7.6a)$$

$$(d/dt) \cdot i_{bci} = (1/L_f) \cdot [v_{bci} - v_{bcl}] \quad (A7.6b)$$

$$(d/dt) \cdot i_{cai} = (1/L_f) \cdot [v_{cai} - v_{cal}] \quad (A7.6c)$$

$$(d/dt) \cdot v_{abl} = [1/(3 \cdot C_f)] \cdot [i_{abi} - i_{abl}] \quad (A7.6d)$$

$$(d/dt) \cdot v_{bcl} = [1/(3 \cdot C_f)] \cdot [i_{bci} - i_{bcl}] \quad (A7.6e)$$

$$(d/dt) \cdot v_{cal} = [1/(3 \cdot C_f)] \cdot [i_{cai} - i_{cal}]. \quad (A7.6f)$$

A dependence on the variables i_{abi} , i_{bci} , and i_{cai} is eliminated by algebraic manipulation yielding the equations:

$$(d^2/dt^2) \cdot v_{abl} = (2\pi \cdot f_f)^2 \cdot [(v_{abi} - v_{abl}) - L_f \cdot (d/dt) \cdot i_{abl}] \quad (A7.7a)$$

$$(d^2/dt^2) \cdot v_{bcl} = (2\pi \cdot f_f)^2 \cdot [(v_{bci} - v_{bcl}) - L_f \cdot (d/dt) \cdot i_{bcl}] \quad (A7.7b)$$

$$(d^2/dt^2) \cdot v_{cal} = (2\pi \cdot f_f)^2 \cdot [(v_{cai} - v_{cal}) - L_f \cdot (d/dt) \cdot i_{cal}] \quad (A7.7c)$$

where $2\pi \cdot f_f = 1/\sqrt{3 \cdot L_f \cdot C_f}$.

The Fourier Transforms of (A7.7a), (A7.7b), and (A.7c) are

$$v_{abl}(f) = [1/(1 - 12 \cdot \pi^2 \cdot f^2 \cdot L_f \cdot C_f)] \cdot [v_{abi}(f) - j \cdot 2\pi \cdot f \cdot L_f \cdot i_{abl}(f)], \quad (A7.8a)$$

$$v_{bcl}(f) = [1/(1 - 12 \cdot \pi^2 \cdot f^2 \cdot L_f \cdot C_f)] \cdot [v_{bci}(f) - j \cdot 2\pi \cdot f \cdot L_f \cdot i_{bcl}(f)], \quad (A7.8b)$$

$$v_{cal}(f) = [1/(1 - 12 \cdot \pi^2 \cdot f^2 \cdot L_f \cdot C_f)] \cdot [v_{cai}(f) - j \cdot 2\pi \cdot f \cdot L_f \cdot i_{cal}(f)]. \quad (A7.8c)$$

In the form of a matrix equation, the output filter circuit state equations are expressed as

$$(d/dt) \cdot \underline{x} = \underline{A} \cdot \underline{x} + \underline{B} \cdot \underline{v_i} + \underline{C} \cdot \underline{w} \quad (A7.9)$$

where

$$\underline{x} = \begin{bmatrix} v_{abl} \\ v_{bcl} \\ v_{cal} \\ (d/dt) \cdot v_{abl} \\ (d/dt) \cdot v_{bcl} \\ (d/dt) \cdot v_{cal} \end{bmatrix}, \quad \underline{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -(2\pi \cdot f_f)^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & -(2\pi \cdot f_f)^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -(2\pi \cdot f_f)^2 & 0 & 0 & 0 \end{bmatrix},$$

$$\underline{v_i} = \begin{bmatrix} v_{abi} \\ v_{bci} \\ v_{cai} \end{bmatrix}, \quad \underline{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ (2\pi \cdot f_f)^2 & 0 & 0 \\ 0 & (2\pi \cdot f_f)^2 & 0 \\ 0 & 0 & (2\pi \cdot f_f)^2 \end{bmatrix},$$

$$\underline{w} = \begin{bmatrix} (d/dt) \cdot i_{abl} \\ (d/dt) \cdot i_{bcl} \\ (d/dt) \cdot i_{cal} \end{bmatrix}, \quad \underline{C} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -L_f \cdot (2\pi \cdot f_f)^2 & 0 & 0 \\ 0 & -L_f \cdot (2\pi \cdot f_f)^2 & 0 \\ 0 & 0 & -L_f \cdot (2\pi \cdot f_f)^2 \end{bmatrix}$$

where $\underline{x}$ is a vector of output filter circuit states. Taking the Laplace Transform of equation (A7.9), the following equation is obtained after algebraic manipulation:

$$\underline{x}(s) = (s \cdot \underline{I} - \underline{A})^{-1} \cdot \underline{x}(t_0) + (s \cdot \underline{I} - \underline{A})^{-1} \cdot \underline{B} \cdot \underline{v}_i(s) + (s \cdot \underline{I} - \underline{A})^{-1} \cdot \underline{C} \cdot \underline{w}(s) \quad (A7.10)$$

where

$$(s \cdot \underline{I} - \underline{A})^{-1} = \begin{bmatrix} s & 0 & 0 & 1 & 0 & 0 \\ 0 & s & 0 & 0 & 1 & 0 \\ 0 & 0 & s & 0 & 0 & 1 \\ -(2\pi \cdot f_f)^2 & 0 & 0 & s & 0 & 0 \\ 0 & -(2\pi \cdot f_f)^2 & 0 & 0 & s & 0 \\ 0 & 0 & -(2\pi \cdot f_f)^2 & 0 & 0 & s \end{bmatrix} \cdot [s^2 + (2\pi \cdot f_f)^2]^{-1}$$

$$(s \cdot \underline{I} - \underline{A})^{-1} \cdot \underline{B} = \begin{bmatrix} (2\pi \cdot f_f)^2 & 0 & 0 \\ 0 & (2\pi \cdot f_f)^2 & 0 \\ 0 & 0 & (2\pi \cdot f_f)^2 \\ s \cdot (2\pi \cdot f_f)^2 & 0 & 0 \\ 0 & s \cdot (2\pi \cdot f_f)^2 & 0 \\ 0 & 0 & s \cdot (2\pi \cdot f_f)^2 \end{bmatrix} \cdot [s^2 + (2\pi \cdot f_f)^2]^{-1}$$

$$(s \cdot \underline{I} - \underline{A})^{-1} \cdot \underline{C} = \begin{bmatrix} -L_f \cdot (2\pi \cdot f_f)^2 & 0 & 0 \\ 0 & -L_f \cdot (2\pi \cdot f_f)^2 & 0 \\ 0 & 0 & -L_f \cdot (2\pi \cdot f_f)^2 \\ -L_f \cdot s \cdot (2\pi \cdot f_f)^2 & 0 & 0 \\ 0 & -L_f \cdot s \cdot (2\pi \cdot f_f)^2 & 0 \\ 0 & 0 & -L_f \cdot s \cdot (2\pi \cdot f_f)^2 \end{bmatrix} \cdot [s^2 + (2\pi \cdot f_f)^2]^{-1}.$$

$\underline{x}(t_0)$ is the vector of output filter states at time t_0 . Taking the inverse Laplace Transform of equation (A7.10) yields the load voltage state transition equations for time $t \geq t_0$:

$$\begin{aligned} v_{abl}(t) = & \cos[(2\pi \cdot f_f) \cdot (t - t_0)] \cdot v_{abl}(t_0) + \\ & [1/(2\pi \cdot f_f)] \cdot \sin[(2\pi \cdot f_f) \cdot (t - t_0)] \cdot v_{abl}'(t_0) + \\ & (2\pi \cdot f_f) \cdot \int_{t_0}^t \sin[(2\pi \cdot f_f) \cdot (t - \tau)] \cdot \left[v_{abi}(\tau) - \right. \\ & \left. L_f \cdot i_{abl}'(\tau) \right] d\tau \end{aligned} \quad (A7.11a)$$

$$\begin{aligned} v_{bcl}(t) = & \cos[(2\pi \cdot f_f) \cdot (t - t_0)] \cdot v_{bcl}(t_0) + \\ & [1/(2\pi \cdot f_f)] \cdot \sin[(2\pi \cdot f_f) \cdot (t - t_0)] \cdot v_{bcl}'(t_0) + \\ & (2\pi \cdot f_f) \cdot \int_{t_0}^t \sin[(2\pi \cdot f_f) \cdot (t - \tau)] \cdot \left[v_{bci}(\tau) - \right. \\ & \left. L_f \cdot i_{bcl}'(\tau) \right] d\tau \end{aligned} \quad (A7.11b)$$

$$\begin{aligned} v_{cal}(t) = & \cos[(2\pi \cdot f_f) \cdot (t - t_0)] \cdot v_{cal}(t_0) + \\ & [1/(2\pi \cdot f_f)] \cdot \sin[(2\pi \cdot f_f) \cdot (t - t_0)] \cdot v_{cal}'(t_0) + \\ & (2\pi \cdot f_f) \cdot \int_{t_0}^t \sin[(2\pi \cdot f_f) \cdot (t - \tau)] \cdot \left[v_{cai}(\tau) - \right. \\ & \left. L_f \cdot i_{cal}'(\tau) \right] d\tau \end{aligned} \quad (A7.11c)$$

where $v_{abl}(t_0)$, $v_{bcl}(t_0)$, $v_{cal}(t_0)$ are the phase-to-phase load voltages at $t = t_0$, $v_{abl}' = (d/dt) \cdot v_{abl}$, $v_{bcl}' = (d/dt) \cdot v_{bcl}$, $v_{cal}' =$

$(d/dt) \cdot v_{cal}, i_{abl}' = (d/dt) \cdot i_{abl}, i_{bcl}' = (d/dt) \cdot i_{bcl}, i_{cal}' =$
 $(d/dt) \cdot i_{cal},$ and τ is a variable of integration.

Experimental Neurocontroller Design

A8.1 Design Philosophy

Implementation of the neurocontroller concept was performed using a serial microprocessor technology. The primary problem encountered with the experimental controller design was the required execution times of the processes involved in the controller operation. These processes are listed as follows:

- i.) Space vector PWM process.
- ii.) Load circuit feedback, load voltage command, and learning-rate parameter analog-to-digital conversion process.
- iii.) Neurocontroller forward propagation and learning processes.

It was necessary for the space vector PWM process to operate without delays created by the other controller processes. A separate processor designated as SVP was assigned the function of generating the inverter switching control signals based on the modulation commands v_{di}^* and v_{qi}^* which are outputs of the neurocontroller. The SVP also generates a analog-to-digital conversion start signal at the beginning of the controller sampling period which is synchronized to the inverter modulation sample period. In the experimental controller constructed, a inverter switching frequency of 750 Hertz was utilized which translated to a 1500 Hertz controller sampling rate.

To minimize delays in converting analog data to a digital format, an analog-to-digital converter (ADC) circuit capable of converting seven analog channels simultaneously in 27 μ s was constructed. ADC operation is initiated by the conversion start signal generated by the SVP once each controller sampling period. At the end of the conversion process, the ADC generates an end-of-conversion signal which is used to initiate the neurocontroller forward propagation and learning processes.

A processor designated as ANCP was assigned the function of load circuit feedback, load voltage command, and learning-rate parameter input from the ADC circuit. Additionally, the ANCP reads controller configuration switch data and outputs data to a digital-to-analog converter port used to generate analog signals for a test point. Configuration switch 7 enables/disables the controller learning process. The ANCP sets the learning-rate parameter value to zero if the learning process is disabled.

A processor designated as NCP was assigned the neurocontroller forward propagation and learning process functions. ADC data input by the ANCP is passed to the NCP via the interconnection of the ANCP and NCP processors' parallel ports (port 1). The modulation commands generated by the neurocontroller forward propagation process are passed to the SVP where the inverter switching commands for the next controller sampling period are generated. To avoid delays in generating the inverter switching control signals, the neurocontroller learning process was executed after the forward propagation process.

Figure A8.1 depicts the processor circuitry common to the SVP, ANCP, and NCP. The processor technology used in this design utilized a 16-bit wide address/data bus. Program code was stored in two ROM's where even address code and odd address code was stored in separate ROM's.

All three processors shared the same power supply, reset, and clock generation circuitry depicted in figure A8.2. A power-on reset signal with a duration of one second was applied to all three processors. Figure A8.3 depicts address decode circuitry associated with each processor.

Controller processes are initiated by the analog-to-digital conversion start signal (AD_START) generated by the SVP at the start of the controller sampling period. The neurocontroller forward propagation and learning processes are executed later in the controller sampling period after the analog-to-digital conversion process is completed. All controller processes are executed within one controller sampling period. This condition limits the practical implementation of the inverter neurocontroller.

A8.2 PWM Inverter Switching Signal Generation

The neurocontroller circuitry generates three inverter switching control signals, S_A , S_B , and S_C which correspond to the switching states of the three inverter leg circuits. A logic 1 signal corresponds to enabling the inverter leg switch connected to the positive DC rail,

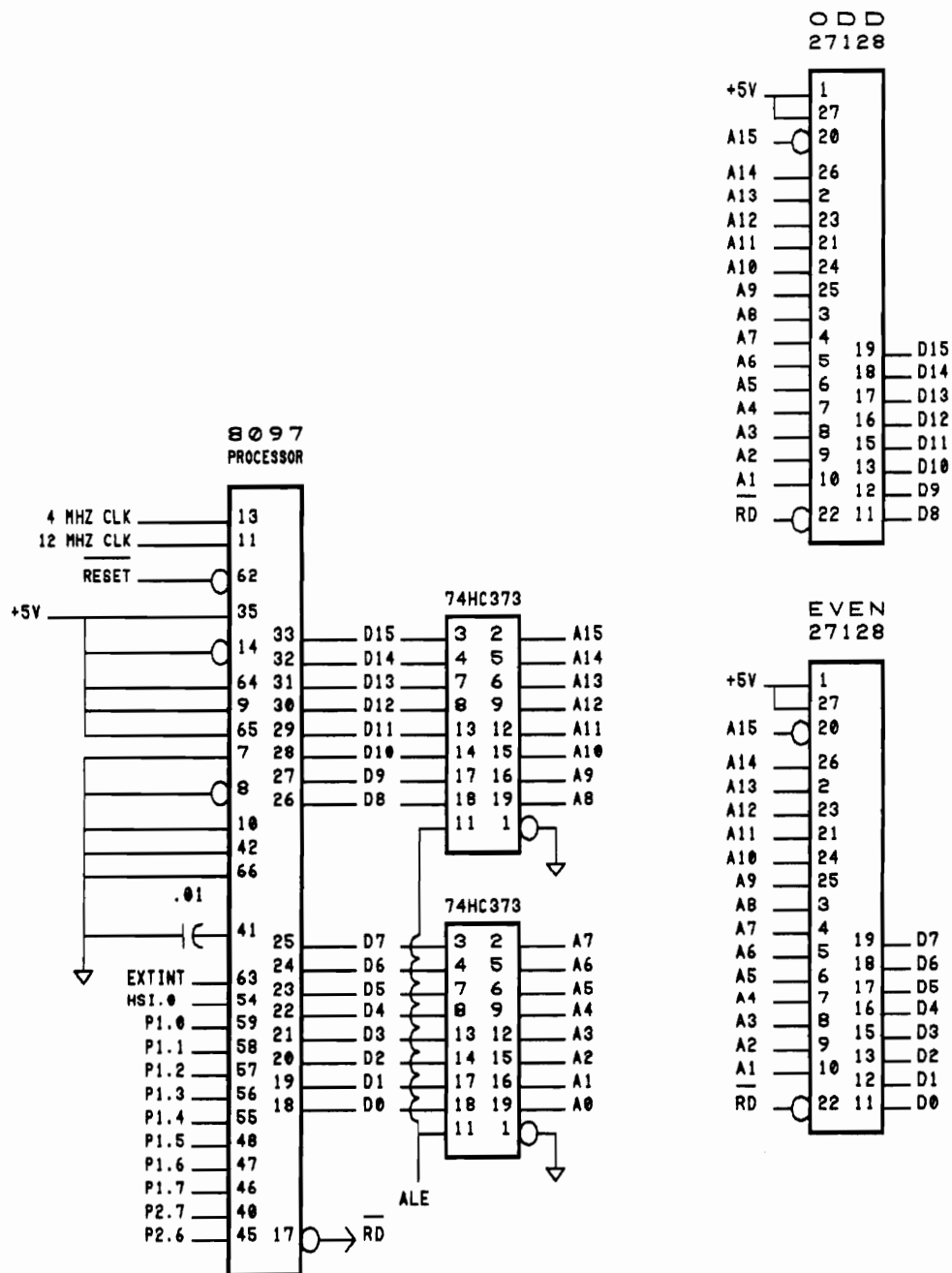


FIGURE A8.1: PROCESSOR ADDRESS/DATA BUS CIRCUITRY

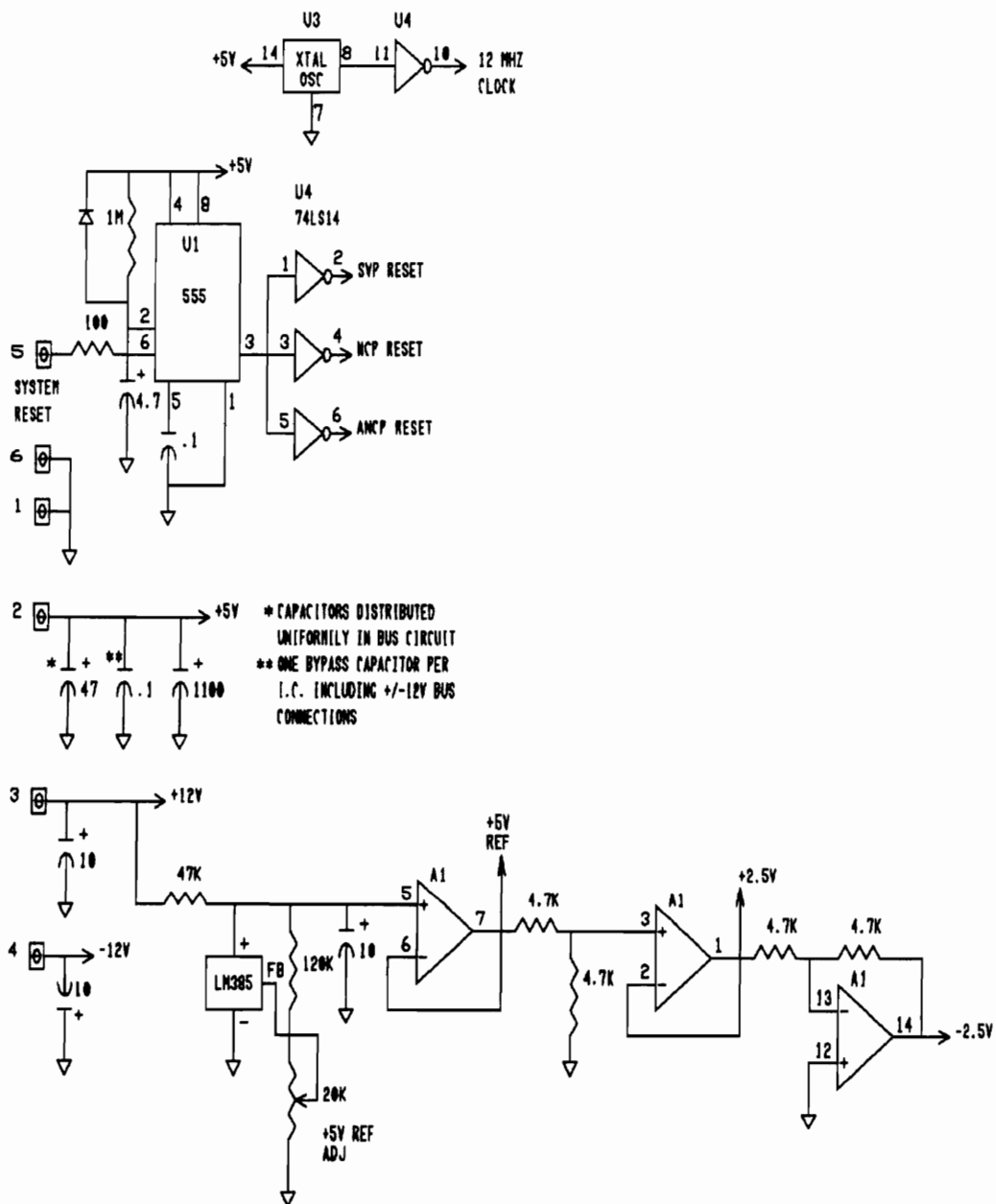


FIGURE A8.2: LOW-VOLTAGE POWER SUPPLY, RESET, CLOCK CIRCUITRY

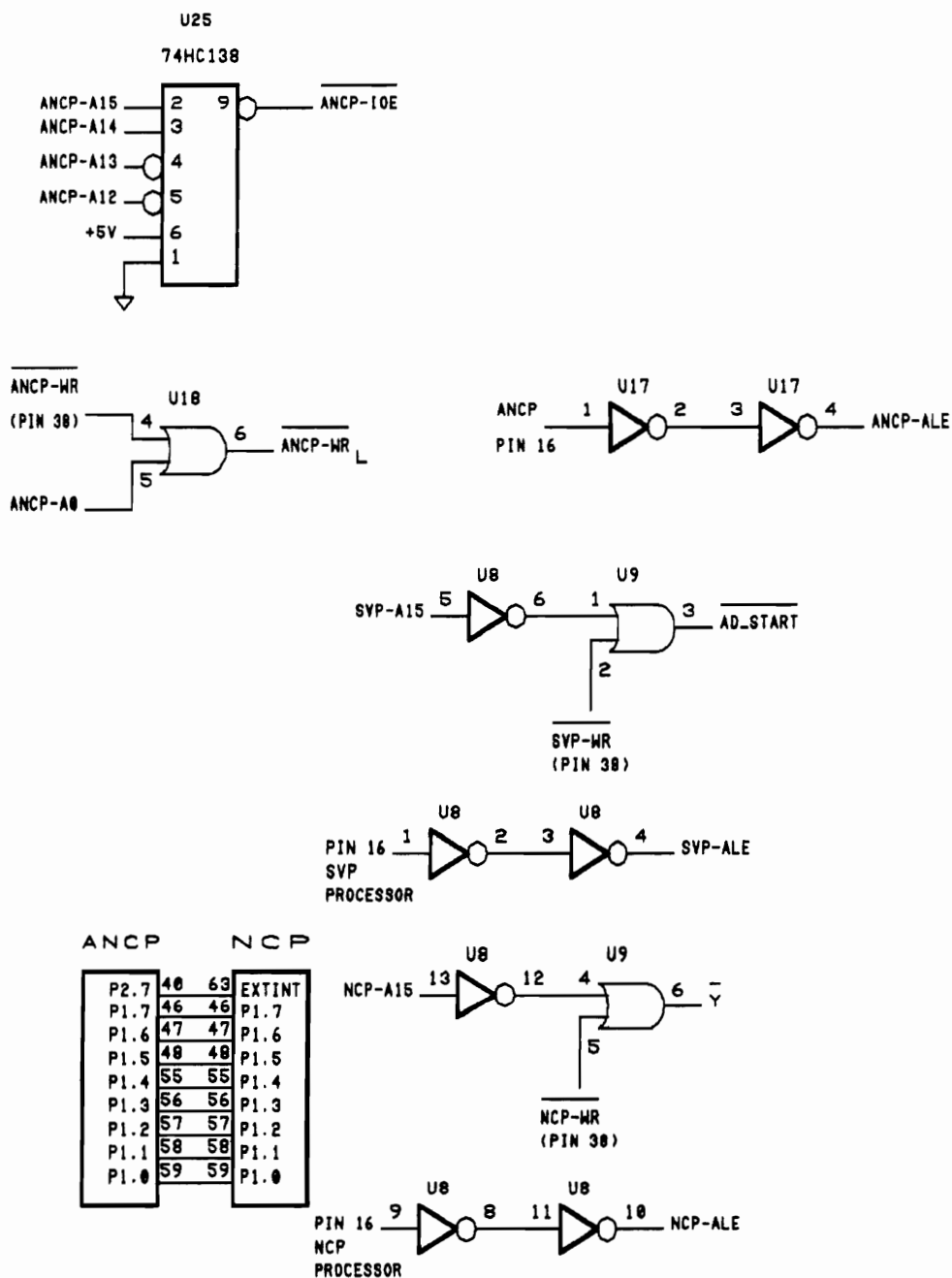


FIGURE A8.3: PROCESSOR MEMORY ADDRESS DECODE CIRCUITRY

while a logic 0 signal corresponds to enabling the inverter leg switch connected to the negative DC rail.

Signals corresponding to S_A , S_B , and S_C are output from the SVP at P1.0, P1.1, and P1.2 (SVP port 1 bits). These logic signals are transmitted to external circuitry via an optical fiber interface depicted in figure A8.4. S_A , S_B , and S_C signals are converted to the six inverter switch control signals by generating the logical complements of S_A , S_B , and S_C . A 60 μ s dead-time is inserted in each of the six inverter switch control signals through the use of the circuitry depicted in figures A8.5A and A8.5B. The logic-level inverter switch control signals are isolated from the inverter bridge gate drive circuitry by 6N136 high-speed opto-couplers.

The logical states of S_A , S_B , and S_C are represented respectively by bits 0, 1, and 2 in all SVP registers handling inverter switching state data.

The logical states of P1.0, P1.1, and P1.2 are updated during SVP software timer interrupt routines. The SVP has a high-speed output unit consisting of a 16-bit counter driven by the processor clock and a high-speed output unit command memory capable of storing eight high-speed output unit commands. Software timer interrupts are initiated by storing the timer interrupt command in the command memory along with the counter value corresponding to the interrupt time.

Software timer 0 interrupt occurs at $1/2f_s$ intervals. This interrupt routine loads the command for the next software timer 0

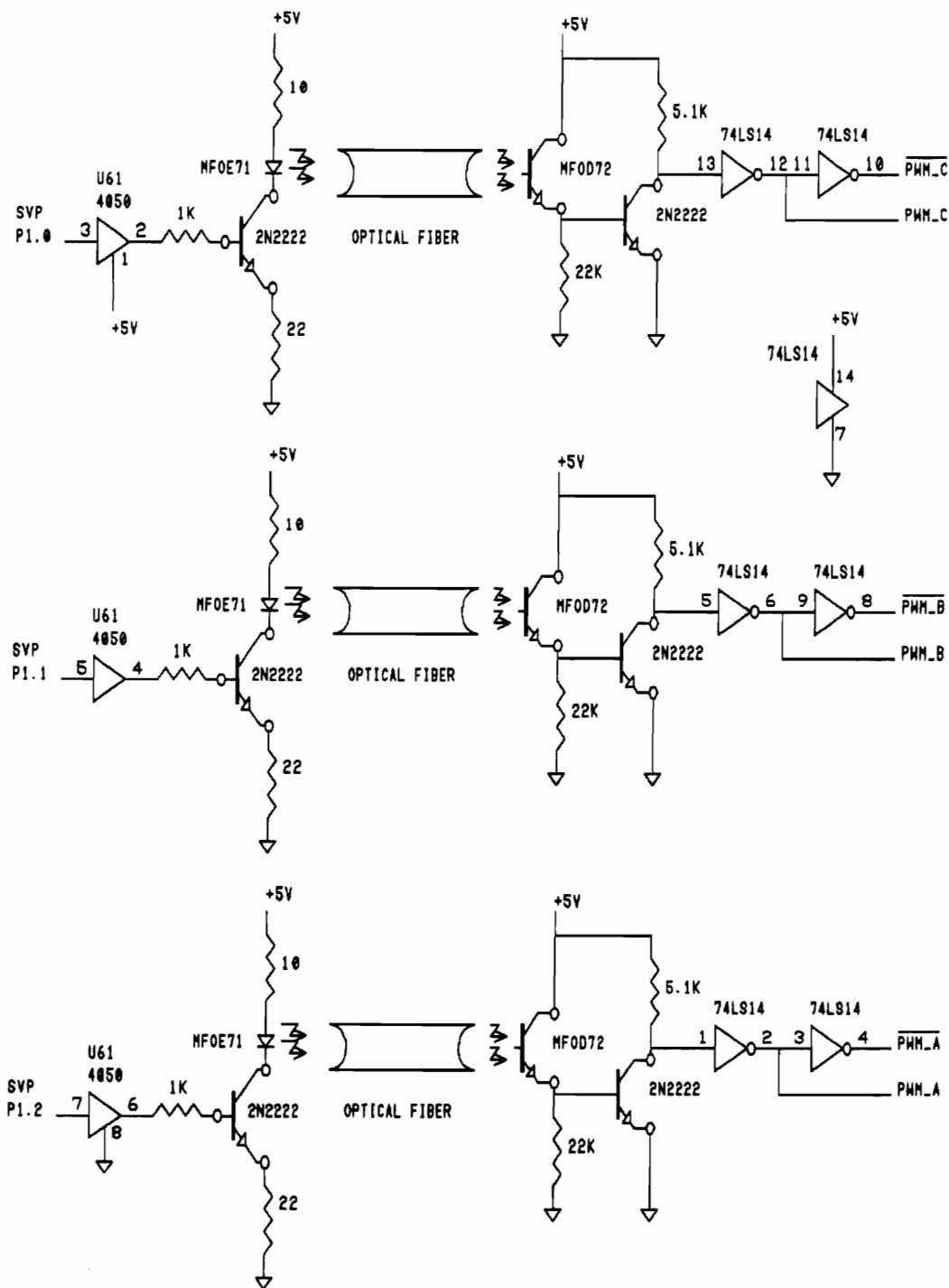


FIGURE A8.4: INVERTER SWITCH CONTROL SIGNAL INTERFACE

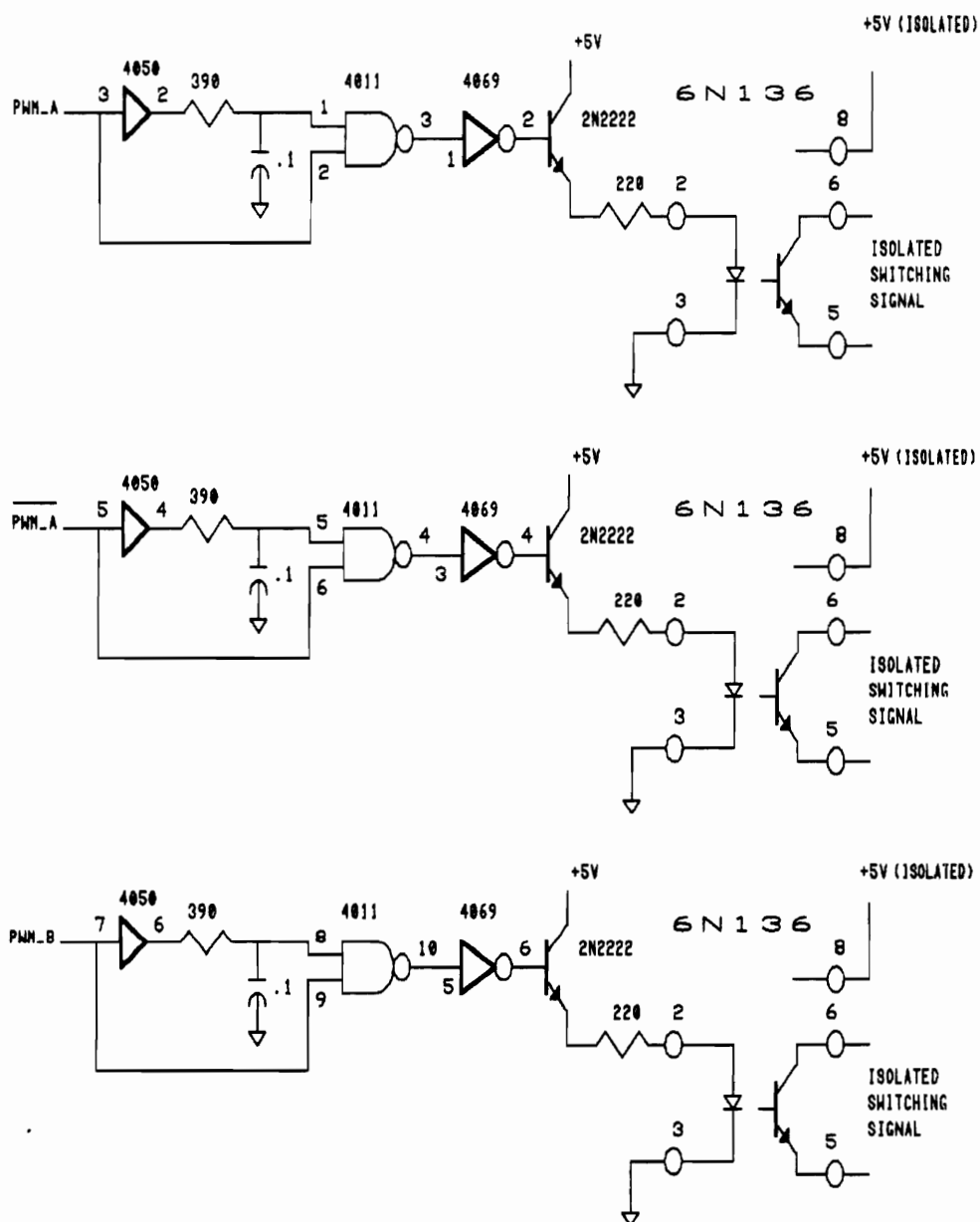


FIGURE A8.5A: DEAD-TIME INSERTION CIRCUITRY

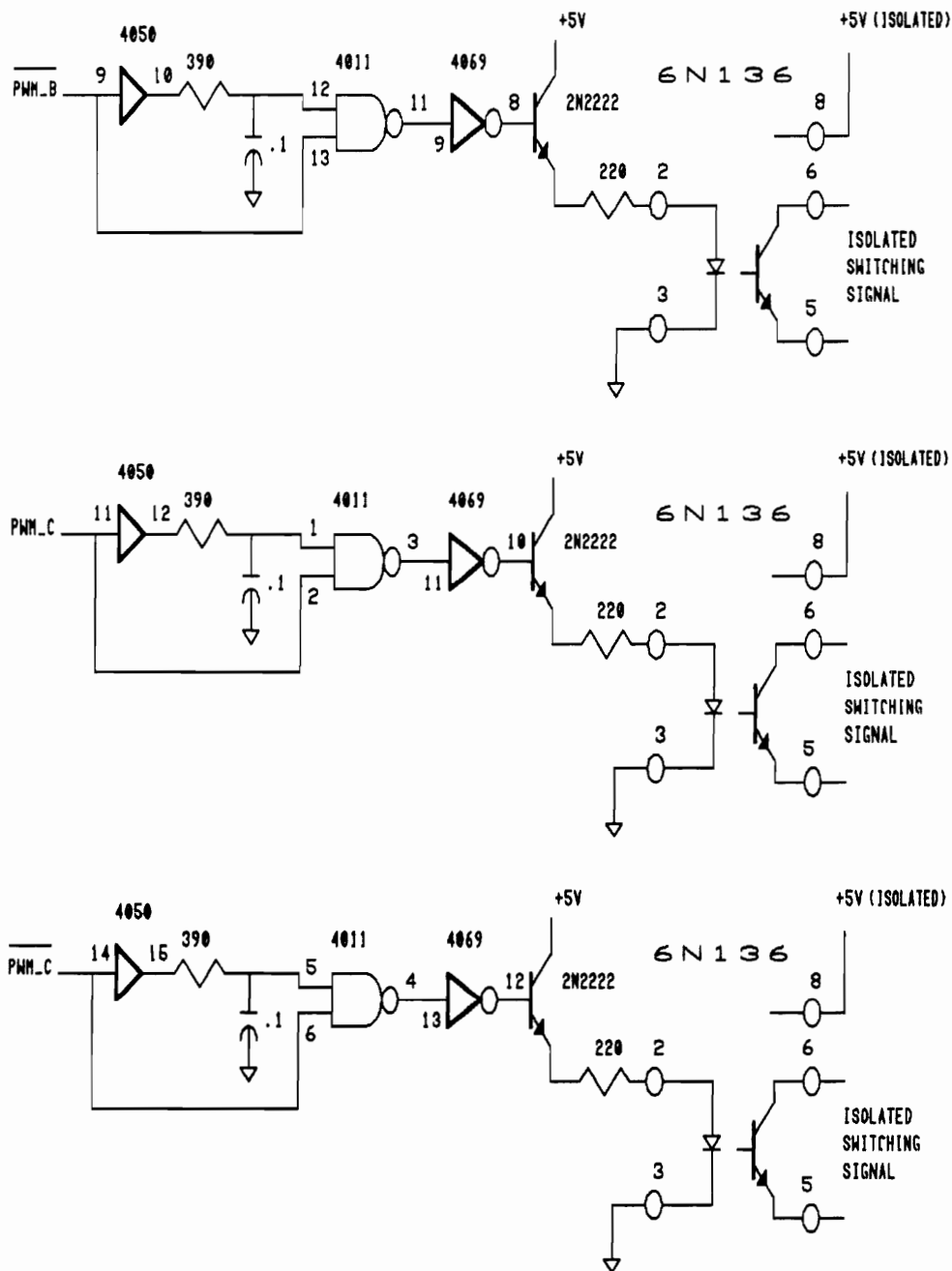


FIGURE A8.5B: DEAD-TIME INSERTION CIRCUITRY

interrupt and the next interrupt time. Software timer 0 interrupt begins the inverter modulation and controller sampling periods.

A8.3 Space Vector PWM Implementation

The least complex space vector PWM implementation derives inverter switching patterns based on the solution of the vector equation

$$2 \cdot f_s \cdot \int_{n/f_s}^{(n + \frac{1}{2})/f_s} \underline{v}_i(\Omega) d\Omega = 2 \cdot f_s \cdot \int_{n/f_s}^{(n + \frac{1}{2})/f_s} \underline{v}_i^* d\Omega \quad (\text{A8.1})$$

where the command vector $\underline{v}_i^* = \{v_{di}^*, v_{qi}^*\}$, and a $1/2f_s$ asymmetric regular sampling interval are used. The left-hand side of equation (A8.1) is equal to the mean inverter output voltage in the n^{th} $1/2f_s$ interval. In each $1/2f_s$ interval (or modulation sample interval), the inverter output waveform vector $\underline{v}_i(t)$ is generated from the execution of three inverter switching states where one of the three states is the zero vector (either state 7 or 8).

The normalized inverter output voltage space vectors are listed as follows:

$$\text{State 1: } \underline{v}_{ds}(1) = \sqrt{3}/2 \quad \underline{v}_{qs}(1) = 1/\sqrt{2} \quad (\text{A8.2a})$$

$$\text{State 2: } \underline{v}_{ds}(2) = 0 \quad \underline{v}_{qs}(2) = \sqrt{2} \quad (\text{A8.2b})$$

$$\text{State 3: } \underline{v}_{ds}(3) = -\sqrt{3}/2 \quad \underline{v}_{qs}(3) = 1/\sqrt{2} \quad (\text{A8.2c})$$

$$\text{State 4: } \underline{v}_{ds}(4) = -\sqrt{3}/2 \quad \underline{v}_{qs}(4) = -1/\sqrt{2} \quad (\text{A8.2d})$$

$$\text{State 5: } \underline{v}_{ds}(5) = 0 \quad \underline{v}_{qs}(5) = -\sqrt{2} \quad (\text{A8.2e})$$

$$\text{State 6: } \underline{v}_{ds}(6) = \sqrt{3}/2 \quad \underline{v}_{qs}(6) = -1/\sqrt{2} \quad (\text{A8.2f})$$

$$\text{State 7: } \underline{v}_{ds}(7) = 0 \quad \underline{v}_{qs}(7) = 0 \quad (\text{A8.2g})$$

$$\text{State 8: } \underline{v}_{ds}(8) = 0 \quad \underline{v}_{qs}(8) = 0 \quad (\text{A8.2h})$$

where $\underline{v}_{ds}^{(k)} = v_{ds}^{(k)}/V_{dc}$ and $\underline{v}_{qs}^{(k)} = v_{qs}^{(k)}/V_{dc}$.

The selection of switching states is based on the premise that inverter switches change state only once during a modulation sample interval. This switching agenda dictates the selection of inverter switching states in the following manner:

- i.) States 7 and 8 (zero vectors) are executed in an alternating fashion each modulation sample interval.
- ii.) Assuming the zero vector is executed first in a modulation sample period, the order of the remaining two switching states is determined by the zero vector executed. For example, if state 7 is executed in an interval, then the next switching state executed will have two switches connected to the positive DC rail. The last switching state will have two switches connected to the negative DC rail.
- iii.) Given the command vector $\underline{v}_i^*$, the selection of the two non-zero vectors corresponds to the two vectors adjacent to the command vector. Of course, if $\underline{v}_i^*$ coincides with a non-zero vector, then the execution of the second non-zero vector does not occur.

The algorithm for the selection of the zero vector is simple. Selection of the non-zero vectors is somewhat more complex. The following algorithm is an example of how the selection process would be implemented:

if $v_{di}^* = 0$ and $v_{qi}^* = 0$
then use only states 7 or 8

```

if  $v_{di}^* \geq 0$  and  $v_{qi}^* \geq 0$ 
{
    and if  $v_{qi}^* \geq v_{di}^*/\sqrt{3}$ 
        then use states 1 and 2
    else
        use states 1 and 6
}

if  $v_{di}^* \leq 0$  and  $v_{qi}^* \geq 0$ 
{
    and if  $v_{qi}^* \geq |v_{di}^*|/\sqrt{3}$ 
        then use states 2 and 3
    else
        use states 3 and 4
}

if  $v_{di}^* \leq 0$  and  $v_{qi}^* \leq 0$ 
{
    and if  $|v_{qi}^*| \geq |v_{di}^*|/\sqrt{3}$ 
        then use states 4 and 5
    else
        use states 3 and 4
}

if  $v_{di}^* \geq 0$  and  $v_{qi}^* \leq 0$ 
{
    and if  $|v_{qi}^*| \geq |v_{di}^*|/\sqrt{3}$ 
        then use states 5 and 6

```

```

else
    use states 1 and 6
}

```

Based on the relative absolute magnitudes of the d- and q-axis commands, the position of the command vector is determined. Once the three switching vectors have been identified, the corresponding switching state times are computed. Equation (A8.1) is reduced to the vector equation

$$2 \cdot f_s \cdot [\underline{v}_s^{(k)} \cdot t_k + \underline{v}_s^{(k+1)} \cdot t_{k+1}] = \underline{v}_i^* \quad (\text{A8.3})$$

where $\underline{v}_s^{(k)}$ and $\underline{v}_s^{(k+1)}$ are the adjacent non-zero vectors, and, t_k and t_{k+1} are the corresponding state times. t_k and t_{k+1} are solutions of the pair of equations obtained from the previous vector equation. Finally, the zero vector state time t_0 is the result of the difference equation

$$t_0 = (1/2f_s) - t_k - t_{k+1}. \quad (\text{A8.4})$$

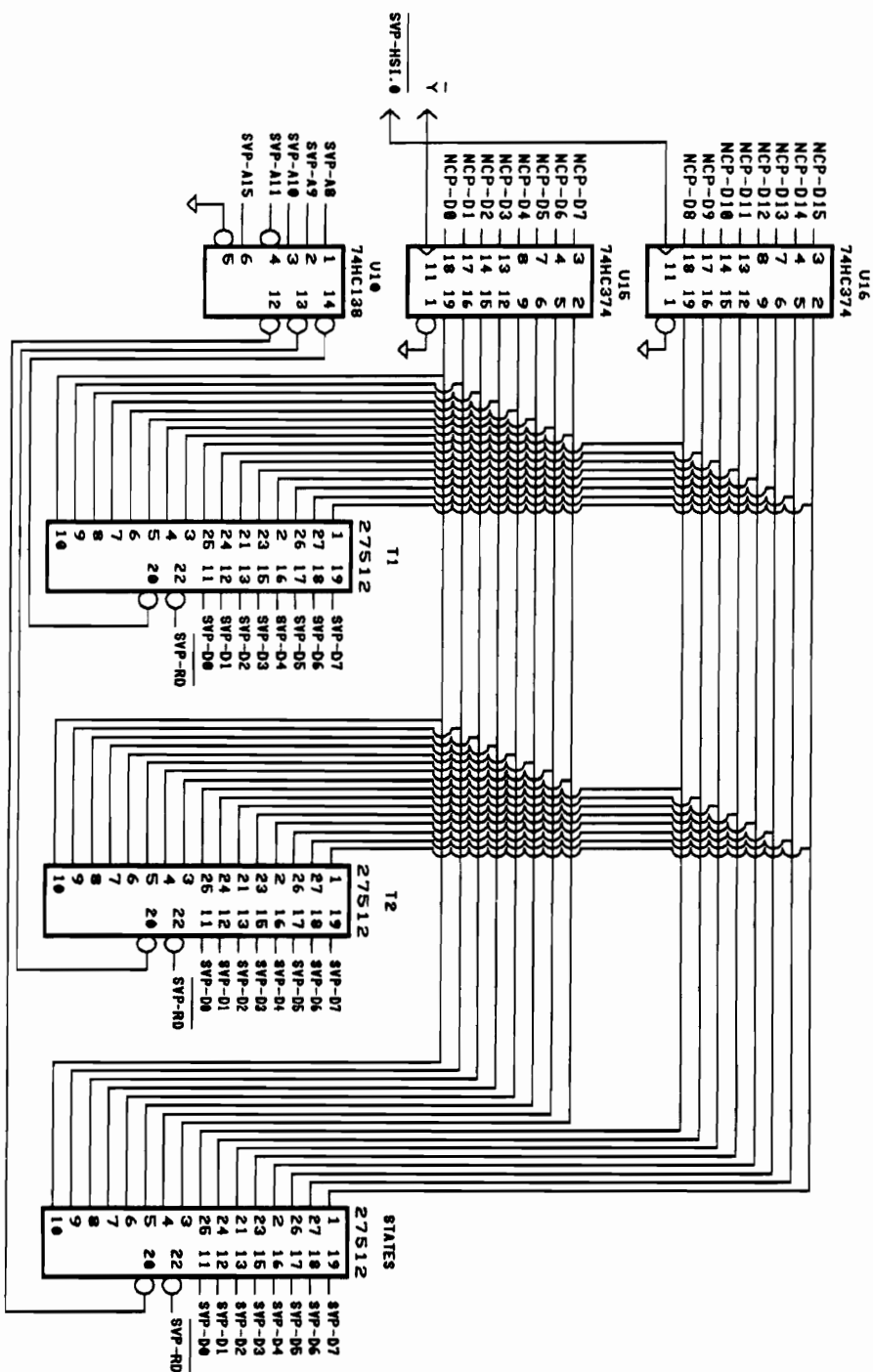
t_k and t_{k+1} are solutions to the set of equations corresponding to equation (A8.3)

$$t_k = (1/2f_s) \cdot (1/\sqrt{3}) \cdot [v_{di}^*(t_0) \cdot v_{qs}^{(k+1)} - v_{qi}^*(t_0) \cdot v_{ds}^{(k+1)}] \quad (\text{A8.5})$$

$$t_{k+1} = (1/2f_s) \cdot (1/\sqrt{3}) \cdot [v_{qi}^*(t_0) \cdot v_{ds}^{(k)} - v_{di}^*(t_0) \cdot v_{qs}^{(k)}] \quad (\text{A8.6})$$

where $v_{di}^*(t_0)$ and $v_{qi}^*(t_0)$ are the inverter output voltage commands at time t_0 .

The experimental neurocontroller design avoids the computational overhead involved in selecting inverter switching states and deriving t_k and t_{k+1} by utilizing pre-computed space vector PWM switching data stored in ROM. Figure A8.6 depicts the space vector PWM data ROM circuitry. At the completion of the forward propagation process executed



in the NCP, $v_{di}^*(t_0)$ and $v_{qi}^*(t_0)$ are written to eight-bit latches U15 and U16. The latch enable signal Y is also connected to the HSI.0 pin on the SVP processor. The $v_{di}^*(t_0)$ and $v_{qi}^*(t_0)$ write operation generates a HSI.0 interrupt in the SVP.

$v_{di}^*(t_0)$ and $v_{qi}^*(t_0)$ are generated by the NCP forward propagation process as eight-bit signed integers. 16-bit space vector PWM data ROM addresses are formed from $v_{di}^*(t_0)$ and $v_{qi}^*(t_0)$ where $v_{di}^*(t_0)$ corresponds to the address low-byte. Each ROM contains 65536 values for t_k (T1 ROM), t_{k+1} (T2 ROM), and inverter switching states corresponding to t_k and t_{k+1} (STATES ROM). t_k and t_{k+1} are normalized with respect to $2f_s$ and stored as 8-bit unsigned numbers. Inverter switching state data is also stored as a 8-bit format described as follows:

- i.) Bits 0-2: switching state corresponding to t_k .
- ii.) Bits 3-5: switching state corresponding to t_{k+1} .
- iii.) Bit 6: parity bit for bits 0-2, logic 1 if two or more bits are 1.
- iv.) Bit 7: parity bit for bits 3-5, logic 1 if two or more bits are 1.

When the HSI.0 interrupt is serviced, ROM data is read from SVP memory addresses 8100 HEX (T1 ROM), 8200 HEX (T2 ROM), and 8300 HEX (STATES ROM). t_k and t_{k+1} read from ROM are transformed from their normalized values to real-time values by the use of the equations

$$t_k \cdot (1/2f_s) / 256 \quad (A8.7)$$

$$t_{k+1} \cdot (1/2f_s) / 256 \quad (A8.8)$$

where $1/2f_s$ is stored in SVP program code as a 8-bit unsigned number. The zero vector state time t_0 is then computed from the difference equation (A8.4). t_0 , t_k , and t_{k+1} are used to compute software timer

interrupt times. The PWM inverter switch control signals are generated by the controller during the SVP software timer interrupt routines.

A SVP register designated as `LAST_ZERO_STATE` contains the number of the last zero inverter switching state executed (either state 7 or 8). If `LAST_ZERO_STATE = 7`, then state 8 zero vector (bit pattern 000) will be the next zero vector executed. If `LAST_ZERO_STATE = 8`, then state 7 zero vector (bit pattern 111) will be the next zero vector executed. The order of execution of the non-zero switching vectors given in the 8-bit number read from the STATES ROM is determined by the zero vector selected. If state 7 is selected, then the non-zero switching state whose parity bit is 1 is executed after the zero vector. If state 8 is selected, then the non-zero switching state whose parity bit is 0 is executed after the zero vector.

Zero and non-zero inverter switching commands are loaded into a switching command queue. t_0 and either t_k or t_{k+1} are used to compute software timer 1 and 2 interrupt times. A SVP register designated as `TIMER1_CLONE` contains the software timer 0 interrupt time. Software timer 1 interrupt time is computed by adding `TIMER1_CLONE` to t_0 . Software timer 2 interrupt time is computed by adding the software timer 1 interrupt time to either t_k or t_{k+1} where the selection of either t_k or t_{k+1} is based on the order non-zero vectors are executed. After the software timer interrupt commands have been given to the SVP high-speed command unit, the HSI.0 interrupt routine is completed.

A8.4 SVP Software Timer Interrupts

During routines servicing SVP software timer interrupt 0, 1, and 2, the logical states of S_A , S_B , and S_C are updated. During each interrupt routine, P1.0, P1.1, and P1.2 are loaded with inverter switching state commands derived in the HSI.0 interrupt routine and stored in a switching command queue. In addition to updating P1.0, P1.1, and P1.2 logical states with the zero vector command, software timer 0 interrupt generates an analog-to-digital conversion start signal (AD_START).

Software timer 1 and 2 interrupts are always serviced after the software timer 0 interrupt. Software timer interrupts have the highest priority, minimizing delays in generating the inverter switch control signals.

Within each controller sampling period, software timer 0 interrupt is always generated. Software timer 1 or 2 interrupts are not generated if t_k or t_{k+1} are less than a minimum switching state interval length.

A8.5 Analog-to-Digital Conversion Circuitry

Figures A8.7 and A8.8 depict the analog-to-digital converter circuitry. Seven ADC0831 8-bit serial I/O A/D converter were selected for this application. This converter requires requires external clock and chip select signals to perform a conversion operation. ADC0831 signal timing is depicted in figure A8.9.

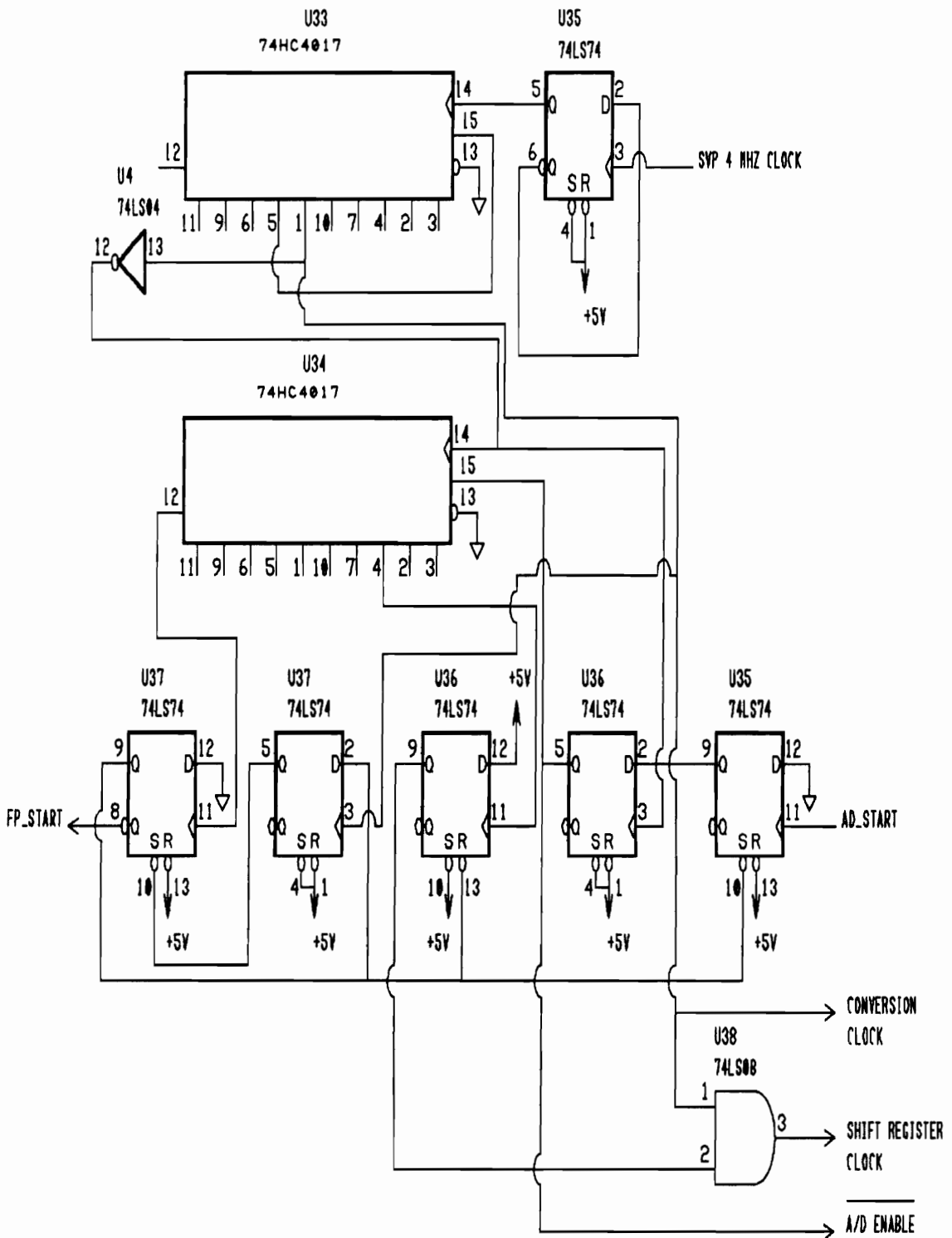
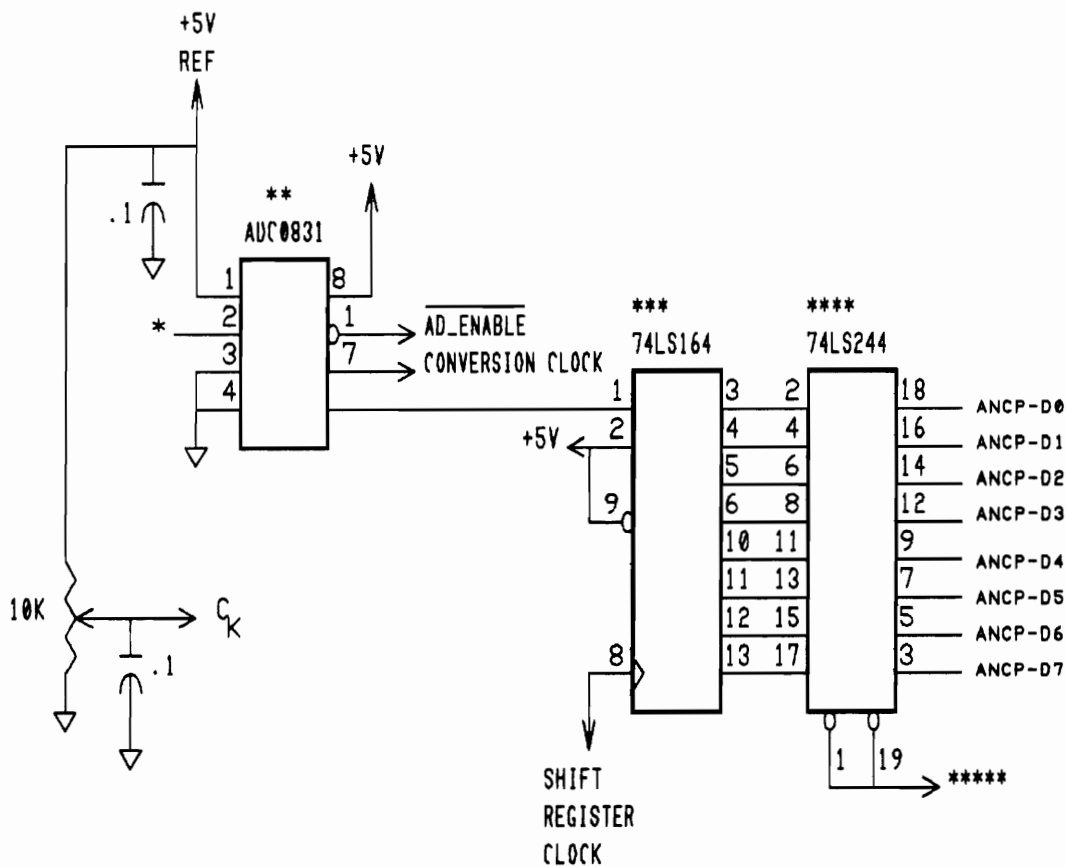


FIGURE A8.7: ANALOG-TO-DIGITAL CONVERSION CONTROL CIRCUITRY



*	**	***	****	*****
✓ _{q1}	U54	U39	U40	U26-15
✓ _{d1}	U53	U41	U42	U26-14
✓ _{q1}	U59	U43	U44	U26-13
✓ _{d1}	U56	U45	U46	U26-12
✓ _{q1}	U58	U47	U48	U26-11
✓ _{d1}	U57	U49	U50	U26-10
C _K	U55	U51	U52	U26-9

FIGURE A8.8: ANALOG-TO-DIGITAL CONVERTERS AND INTERFACE

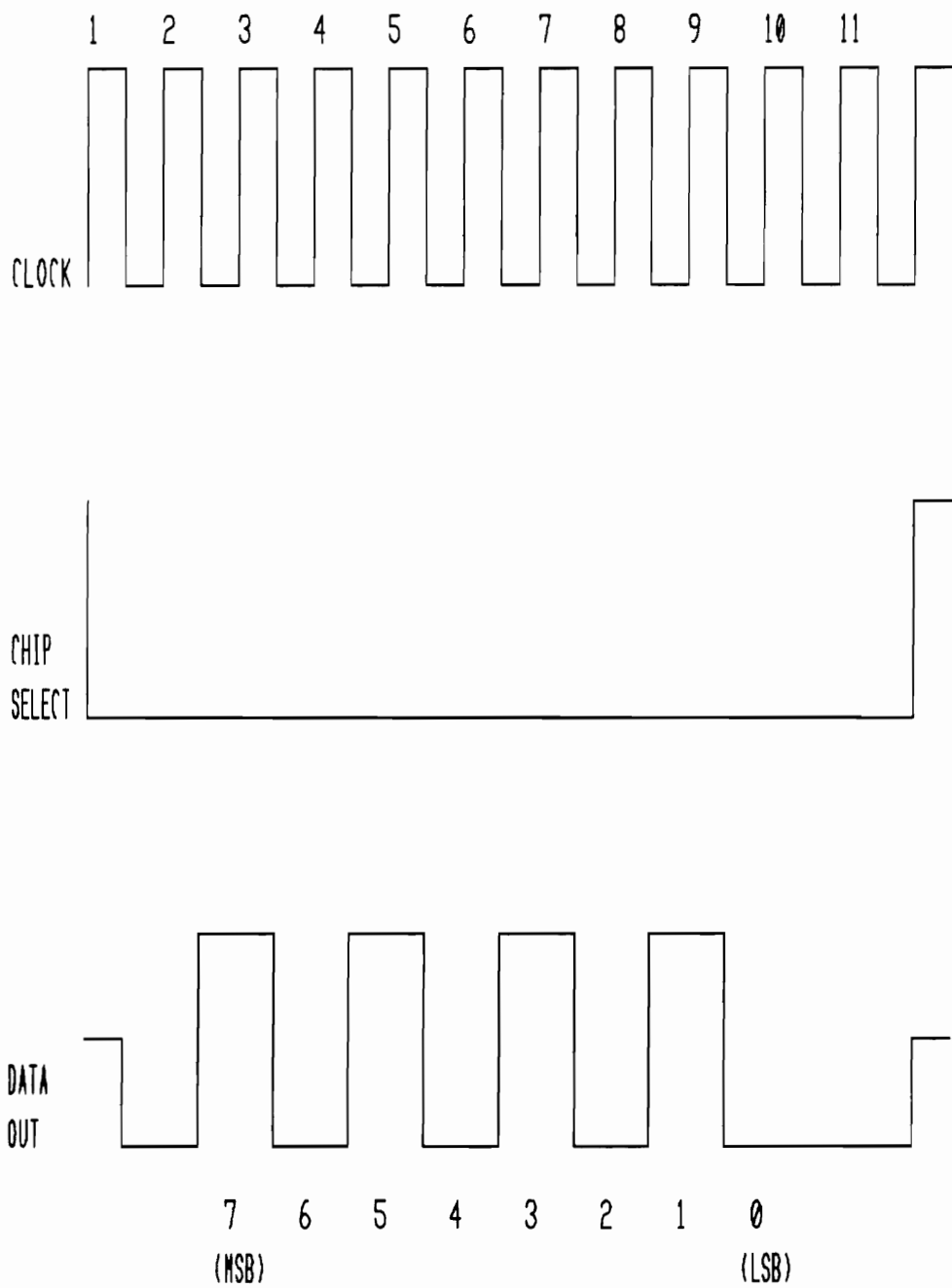


FIGURE A8.9: ADC0831 TIMING

The SVP generates an analog-to-digital conversion start signal (AD_START) at the beginning of the controller sampling interval. The SVP 4 MHZ clock is reduced to a 250 KHZ clock signal (CONVERSION_CLOCK) via a 74HC4017 counter (U33). The inverted state of AD_START is latched by CONVERSION_CLOCK and is present at U36-5 (AD_ENABLE). AD_ENABLE is the chip enable signal for all seven ADC0831 A/D converters.

The logic 0 AD_ENABLE enables another 74HC4017 counter (U34) which counts the number of CONVERSION_CLOCK pulses generated after AD_ENABLE becomes logic 0. After the second CONVERSION_CLOCK pulse is generated, U34-4 goes to a logic 1, setting U36-9 and enabling the SHIFT_REGISTER_CLOCK via U38.

The next eight CONVERSION_CLOCK pulses shift A/D data out of each ADC0831 into a corresponding serial-in/parallel-out shift register (74LS164). The most significant bit of the A/D data is shifted out first. When counter U34-12 goes to a logic 1, U37-8 (FP_START) goes to a logic 1, U37-9 goes to a logic 0, AD_ENABLE goes to a logic 1 (disabling the ADC0831's), and the SHIFT_REGISTER_CLOCK is disabled. FP_START signal is connected to the HSI.0 pins on the ANCP and NCP. The analog-to-digital conversion circuitry is reset to the state prior to the generation of the AD_START signal.

A8.6 Analog Interface Circuitry

Analog signals input to the inverter neurocontroller are listed as follows:

i.) $v_{abl}, v_{bcl}, v_{cal}$ - Three-phase line-to-line load voltage feedback.

ii.) i_{al}, i_{bl} - Two load phase currents.

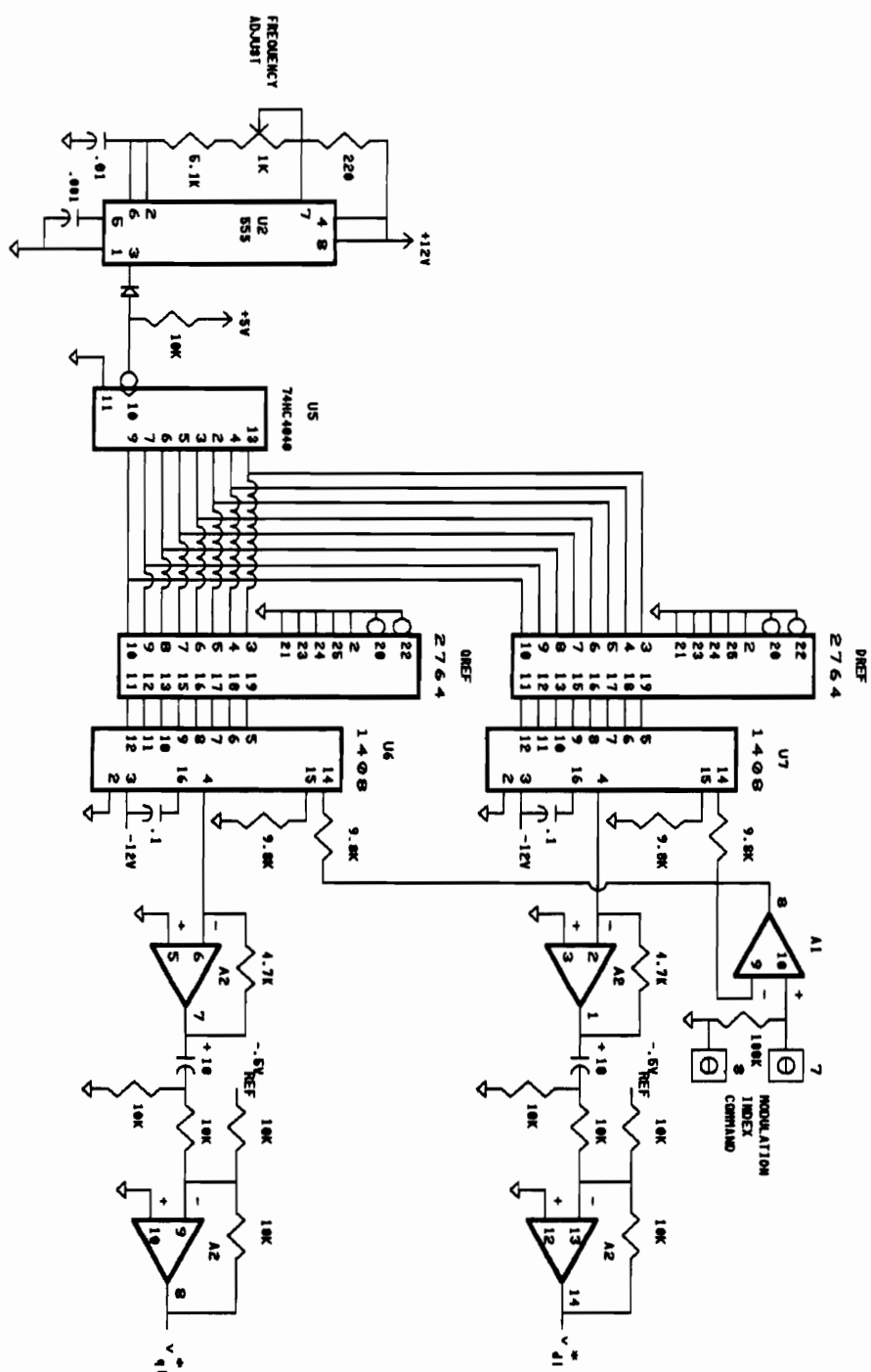
iii.) v_{dl}^*, v_{ql}^* - Line-to-line d-q frame load voltage commands.

iv.) c_k - Neurocontroller learning-rate parameter.

v_{dl}^*, v_{ql}^* , and c_k are generated on the neurocontroller circuit board. c_k input is obtained from the voltage at the wiper of a potentiometer used to manually adjust the learning-rate parameter.

v_{dl}^*, v_{ql}^* are generated by a sinusoidal voltage reference generator circuit depicted in figure A8.10. A free-running oscillator (U2) generates a clock for a binary counter (U5) which in turn generates an eight-bit address for ROM's designated as DREF and QREF. The output frequency of U2 is 256 times the inverter modulation frequency. ROM's DREF and QREF contain sine and cosine waveform data points respectively. DREF and QREF ROM data outputs are converted to analog sine and cosine waveform signals via corresponding digital-to-analog converter (DAC) circuits. The DAC circuit output signals, which are unipolar, are converted to bipolar signals by capacitively coupling the DAC outputs to the final stage of the reference generator circuit. The bipolar sinusoidal reference waveforms are offset by +2.5 volts prior to their application to the analog-to-digital conversion circuitry.

The amplitude of the sinusoidal reference waveforms is controlled by an external modulation index signal (0 to 10 volts) where a 10 volt signal corresponds to the maximum sinusoidal reference waveform amplitude.



From the load current feedback i_{a1} and i_{b1} , the third load current phase i_{c1} is synthesized (in the load current sensor circuit external to the neurocontroller circuitry). Three-phase load circuit feedback is converted to the d-q frame feedback using the analog interface circuitry depicted in figures A8.11A and A8.11B. As in the case of the load voltage commands v_{d1}^* and v_{q1}^* , the bipolar feedback signals v_{d1} , v_{q1} , i_{d1} , and i_{q1} are offset by +2.5 volts. This enables the analog-to-digital conversion circuitry, which is able to convert signals in the range of 0 to +5 volts, to process the analog feedback data without distortion.

A8.7 ANCP Function

When the FP_START signal generated by the analog-to-digital conversion (ADC) circuit goes to a logic 1 state, a HSI.0 interrupt is generated in the ANCP. The ANCP HSI.0 interrupt routine reads ADC output data from the following ANCP memory-mapped ports:

- i.) C000 HEX - v_{q1}^*
- ii.) C100 HEX - v_{d1}^*
- iii.) C200 HEX - i_{q1}
- iv.) C300 HEX - i_{d1}
- v.) C400 HEX - v_{q1}
- vi.) C500 HEX - v_{d1}
- vii.) C600 HEX - c_k .

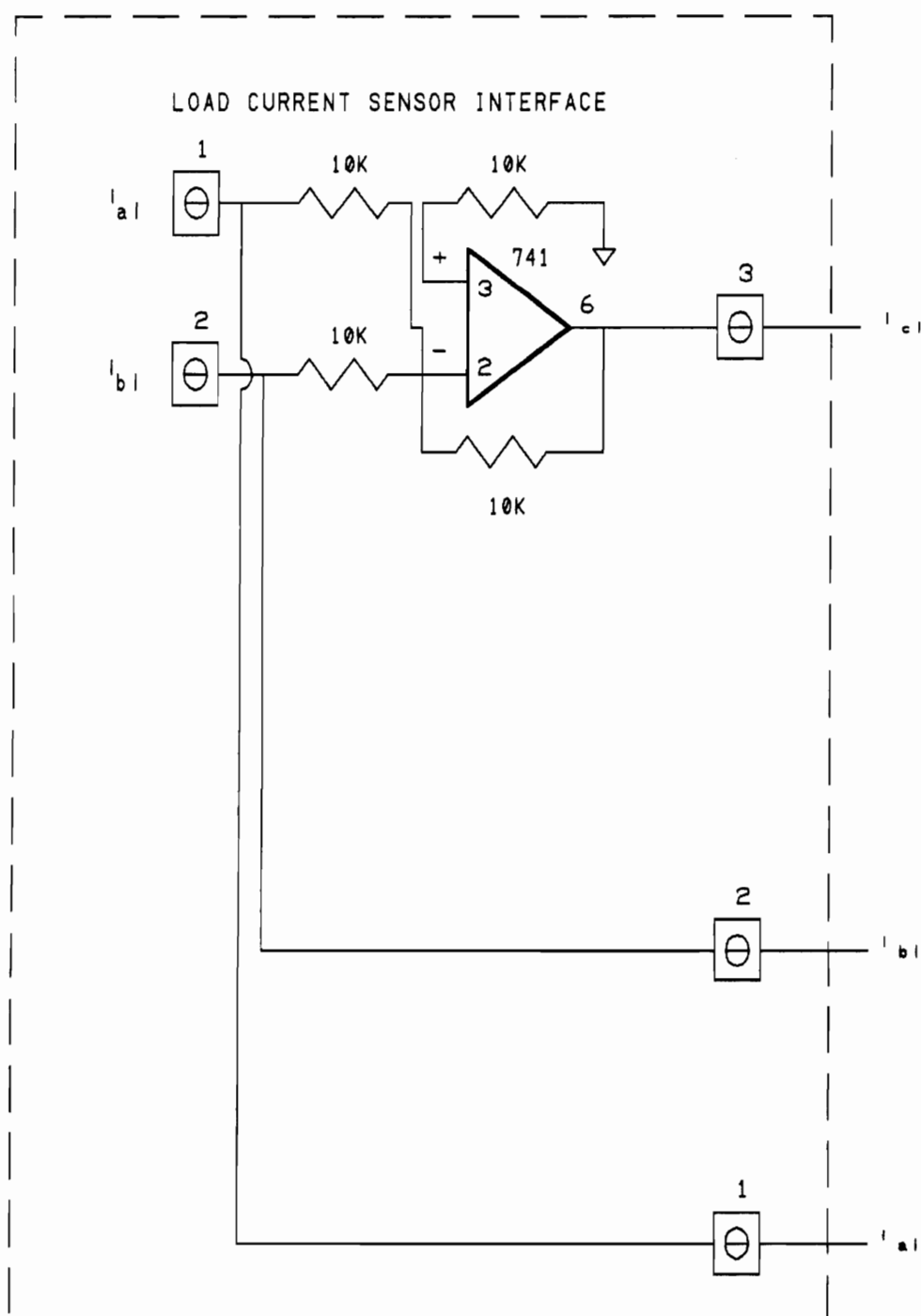


FIGURE A8.11A: ANALOG INTERFACE CIRCUITRY

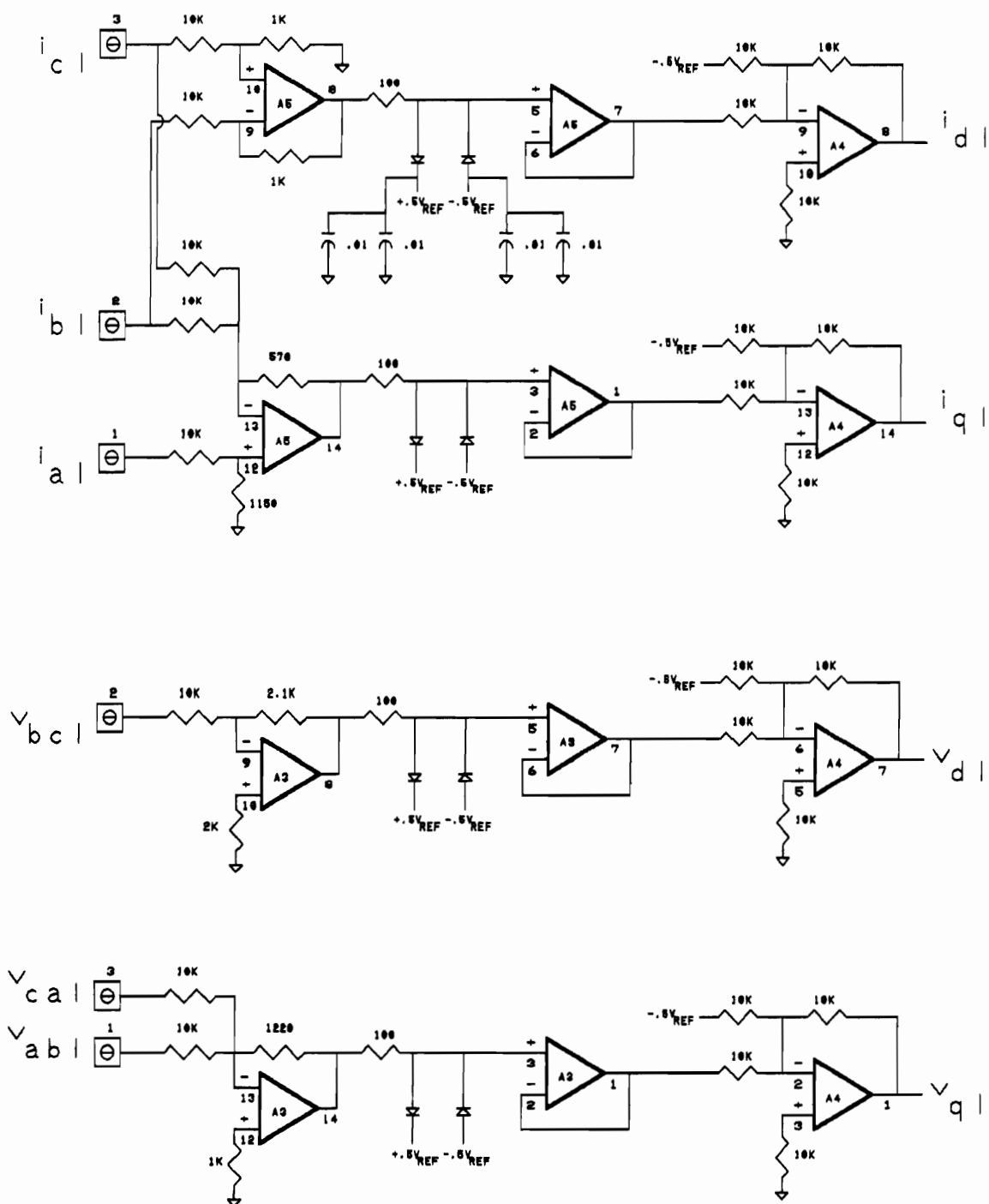


FIGURE A8.11B: ANALOG INTERFACE CIRCUITRY

ADC data is input to the ANCP as 8-bit unsigned numbers. Once all seven analog inputs are read, the ANCP reads the status of the controller configuration switches located at ANCP memory address C700 HEX. The state of configuration switch 8 either enables or disables the neurocontroller learning process. If the learning process is disabled, the learning-rate parameter read from the ADC is set to zero.

The ANCP passes the analog data to the NCP via its parallel port 1. An analog data byte is loaded into port 1 and the ANCP P2.7 output is set to a logic 1 for 1 μ s. P2.7 on the ANCP is connected to the EXTINT pin on the NCP. The logic 1 generated at the ANCP P2.7 generates an external interrupt in the NCP which causes the NCP to read the analog data byte present at the ANCP parallel port 1. The parallel ports of the ANCP and NCP are connected pin-for-pin. After the NCP reads the analog data byte, the ANCP repeats the process for the remaining six analog data bytes.

After the ANCP has passed all seven analog data bytes to the NCP, the ANCP outputs to a digital-to-analog converter (DAC) port at memory address C000 HEX the analog data byte read from the ADC specified by the states of configuration switches 1, 2, and 3. The DAC port circuitry is depicted in figure A8.12. The states of configuration switches 1, 2, and 3 correspond to the lowest three bits of the byte read from the configuration switch port where switch 1 state is in bit 0. A logic 0 bit state corresponds to a switch closure. The following table describes the analog data byte selection process:

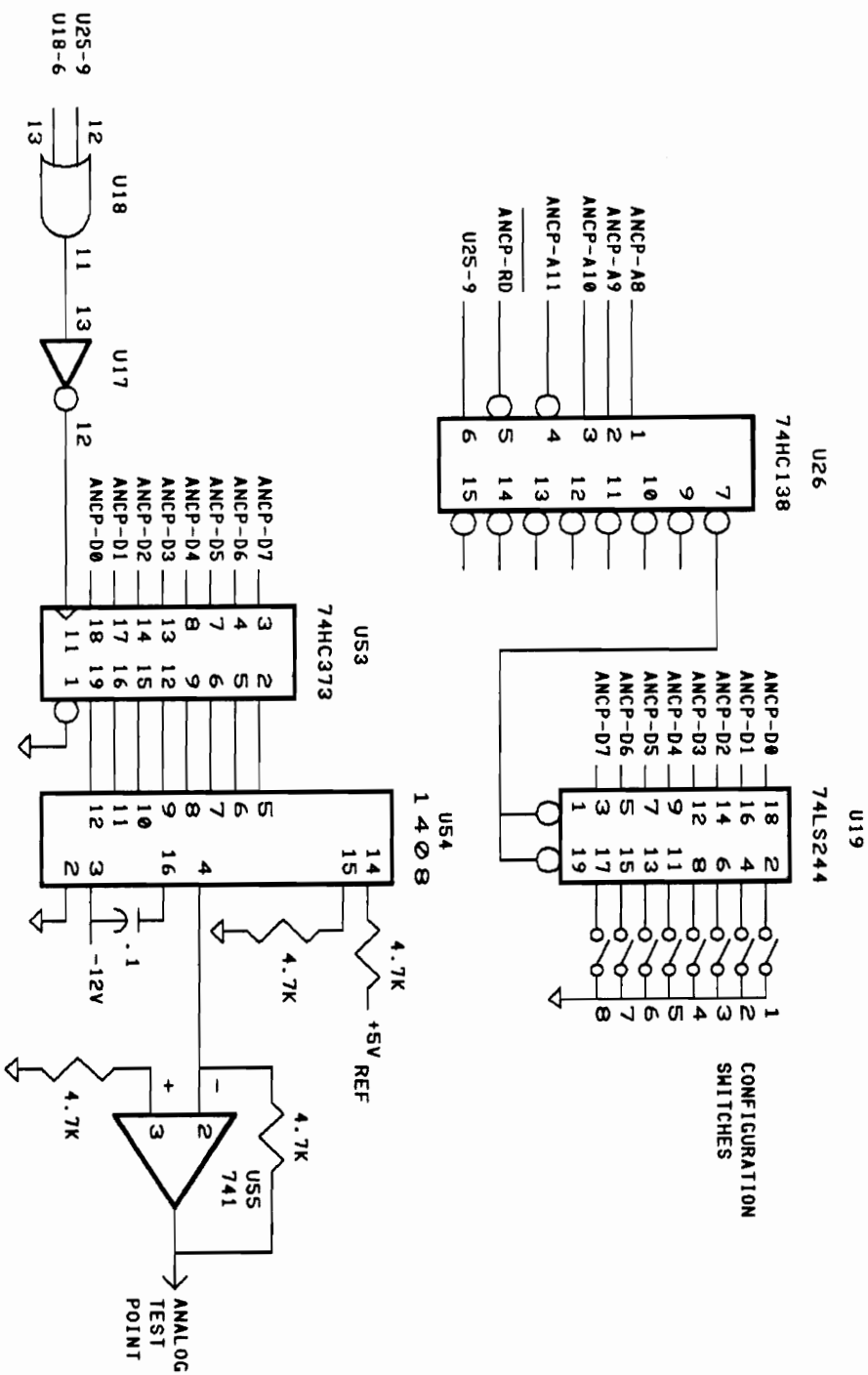


FIGURE A8.12: DAC AND CONFIGURATION SWITCH PORT

ANALOG DATA	SWITCH 1	SWITCH 2	SWITCH 3
v_{q1}^*	OPEN	OPEN	OPEN
v_{d1}^*	CLOSED	OPEN	OPEN
i_{d1}	OPEN	CLOSED	OPEN
i_{q1}	CLOSED	CLOSED	OPEN
v_{q1}	OPEN	OPEN	CLOSED
v_{d1}	CLOSED	OPEN	CLOSED
c_k	OPEN	CLOSED	CLOSED

The output of this DAC port is connected to an analog signal test point.

A8.9 NCP Function

When the FP_START signal generated by the analog-to-digital conversion (ADC) circuit goes to a logic 1 state, a HSI.0 interrupt is generated in the NCP, and the NCP EXTINT interrupt is enabled. Initially in the NCP HSI.0 interrupt routine, the NCP waits for an EXTINT interrupt which signals the availability of an ADC data byte at the NCP parallel port 1. The EXTINT interrupt routine reads and stores the analog data byte in the parallel port 1. After seven analog data bytes are input, the EXTINT interrupt is disabled. Current and voltage analog data bytes are converted to 8-bit signed numbers by subtracting 128 from each value. This is equivalent to eliminating the 2.5 volt offset added to the analog current and voltage signals in the analog interface and reference generator circuitry.

After the analog data conversion from unsigned to signed numbers, the load current difference values $[i_{d1}(k) - i_{d1}(k - 1)]$ and $[i_{q1}(k) - i_{q1}(k - 1)]$ are computed. The forward propagation algorithm based on the equations

$$\begin{aligned} v_{di}^*(k) = & w_{11}(k) \cdot v_{d1}^*(k) + w_{12}(k) \cdot v_{q1}^*(k) + \\ & w_{13}(k) \cdot [i_{d1}(k) - i_{d1}(k - 1)] + \\ & w_{14}(k) \cdot [i_{q1}(k) - i_{q1}(k - 1)] \end{aligned} \quad (A8.9)$$

$$\begin{aligned} v_{qi}^*(k) = & w_{21}(k) \cdot v_{d1}^*(k) + w_{22}(k) \cdot v_{q1}^*(k) + \\ & w_{23}(k) \cdot [i_{d1}(k) - i_{d1}(k - 1)] + \\ & w_{24}(k) \cdot [i_{q1}(k) - i_{q1}(k - 1)] \end{aligned} \quad (A8.10)$$

is executed. At the initial controller power-up, the neurocontroller weights w_{11} and w_{22} are set to a value of 1.0. All other weights are set to zero. This implies that initially the inverter output voltage commands are equivalent to the load voltage commands.

Neurocontroller weights are expressed as 16-bit signed numbers where the weights are normalized to 8192. Neurocontroller weight values are limited to a range $-4.0 \leq w \leq 4.0$. The neurocontroller outputs, $v_{di}^*(k)$ and $v_{qi}^*(k)$, and inputs are expressed as 8-bit signed numbers where inputs and outputs are normalized to 128. Controller addition and multiplication operations are performed using fixed-point mathematics.

When the forward propagation process is completed, $v_{di}^*(k)$ and $v_{qi}^*(k)$ are written to eight-bit data latches U15 and U16. This operation creates an HSI.0 interrupt in the SVP.

The neurocontroller learning process is executed after the forward propagation process. The load voltage errors

$$\epsilon_{dv}(k) = v_{d1}^*(k) - v_{d1}(k) \quad (A8.11)$$

$$\epsilon_{qv}(k) = v_{q1}^*(k) - v_{q1}(k) \quad (A8.12)$$

are computed, and used to update the current set of neurocontroller weights. If the learning process was disabled, the learning-rate parameter is set to zero and no change in neurocontroller weights occurs.

The learning process algorithm limits weight changes in a fashion that keeps neurocontroller weight values the range $-4.0 \leq w \leq 4.0$. If any neurocontroller weight change causes the weight to exceed this range, the weight update process is aborted and no weight changes are implemented. This function has been implemented to avoid unpredictable learning process results due to the use of fixed-point mathematics.

After the learning process is completed, the NCP HSI.0 interrupt routine is completed, and the NCP waits for another HSI.0 interrupt.

A8.10 Experimental Neurocontroller Design

Summary

The parallel structure of neural networks are more readily implemented in hardware constructed with multiple parallel processors. Ideally, each processor would be assigned the forward propagation and learning process functions of one node. A neural network hardware implementation of this type would realize minimum forward propagation and learning process execution times.

As in other conventional digital controller algorithms, the speed at which the entire controller algorithm is executed is a critical factor. Of all the processes executed in the neurocontroller, the learning process consumes the largest amount of execution time, roughly four times the execution time of the forward propagation process. This comparison of execution times is dependent on the size of the neural network utilized in the neurocontroller. Larger network structures and more complex learning algorithms will create longer execution times.

The total execution time of the learning process is the primary limitation to the size of the neural network used in the controller. In an on-line trained controller, the learning process must be executed each controller sampling period. The execution of this process should not create delays in the space vector PWM process, and so, its execution after the forward propagation process was necessary given the processor technology utilized.

A lesser but still critical factor in the digital controller performance is the total analog-to-digital conversion time. Parallel conversion of seven analog channels was necessary to minimize this conversion time.

A neural network implementation using multiple parallel processors is limited by the cost of such an implementation. A practical implementation of a neural network using multiple parallel processors would most likely be based on custom-designed very large scale integrated circuits. This approach is feasible if the size of the neural

network has been established, and the use of the custom VLSI is in large quantities reducing the cost of developing the custom VLSI.

Though the use of a serial processor technology in a neurocontroller design limits the rate neural network forward propagation and learning processes are executed, this approach is more adaptable with regard to its practical implementation. The custom VLSI approach limits the neurocontroller design to a specific application. By using serial processor hardware in a neurocontroller design, a practical neurocontroller hardware design may be developed which may be applied to several different applications.

Future improvements in a neurocontroller design should employ faster processor technology with the objective of executing the learning process before the forward propagation process.

A8.11 A General Procedure for Single-Layer Neurocontroller Implementations

In a practical control problem, the plant information is typically available with regard to the plant state equations' dependent and independent variables. In addition to the plant output commands, the number of neurocontroller inputs required is determined by the number of variables given by equations describing the plant inverse dynamics. The number of neurocontroller connection weights is determined by the number of neurocontroller inputs and outputs.

After establishing the neurocontroller size, the number of math operations required to perform the forward propagation and learning processes is derived. The minimum number of math operations required in the forward propagation process are $N_i \cdot N_o$ additions and $N_i \cdot N_o$ multiplications where N_i is the number of neurocontroller inputs and N_o is the number of neurocontroller outputs. Using the modified backpropagation algorithm, the minimum math operations required are listed as follows:

- i.) Two subtraction and multiplication operations are needed to compute the terms $c_k \cdot \epsilon_{dv}$ and $c_k \cdot \epsilon_{qv}$ where c_k is the learning-rate parameter.
- ii.) $N_i \cdot N_o$ multiplications are required to compute the connection weight changes δw .
- iii.) $N_i \cdot N_o$ additions are required to update the connections weights where $w(k) = w(k - 1) + \delta w$.

Addition math operations will be required to limit connection weight changes each learning iteration if fixed-point mathematics is used. This is necessary to prevent an unintended condition weight value overflow condition.

The minimum number of math operations required to execute the forward propagation and learning processes is used to determine the minimum execution times for these processes given a specified processor technology. The sum of these process execution times and the analog-to-digital conversion time is an approximation of the minimum controller

sampling period. Other processor operations such as memory data transfers and interrupt servicing are added to this sum.

Controller operations other than the operations associated with the neural network such as analog-to-digital conversion and the space vector PWM process should be executed in independent processors or in a hardware-based circuit in order to minimize controller processing delays.

REFERENCES

- [1] Atsuo Kawamura and Richard Hoft, "Instantaneous Feedback Controlled PWM Inverter with Adaptive Hysteresis", IEEE Transactions on Industry Applications, Vol. IA-20, No. 4, pp. 769-775, July/August 1984.
- [2] Kalyan P. Gokhale, Atsuo Kawamura, and Richard G. Hoft, "Dead Beat Microprocessor Control of PWM Inverter for Sinusoidal Output Waveform Synthesis", IEEE Transactions on Industry Applications, Vol. IA-23, No. 5, pp. 901-910, September/October 1987.
- [3] Toshimasa Haneyoshi, Atsuo Kawamura, and Richard G. Hoft, "Waveform Compensation of PWM Inverter with Cyclic Fluctuating Loads", IEEE Transactions on Industry Applications, Vol. 24, No. 4, pp. 582-589, July/August 1988.
- [4] Takao Kawabata, Takeshi Miyashita, and Yushin Yamamoto, "Dead Beat Control of Three Phase PWM Inverter", IEEE Transactions on Power Electronics, Vol. 5, No. 1, pp. 21-28, January, 1990.
- [5] Takao Kawabata, Takeshi Miyashita, Yushin Yamamoto, "Digital Control Of Three-Phase PWM Inverter With LC Filter", IEEE Transactions on Power Electronics, Vol. 6, No. 1, pp. 62-72, January, 1991.
- [6] Michael A. Boost and Phoivos D. Ziogas, "Towards a Zero-Output Impedance UPS System", IEEE Transactions on Industry Applications, Vol. 25, No. 3, pp. 408-418, May/June 1990.
- [7] Prasad N. Enjeti and Wajiha Shireen, "A New Technique to Reject DC-Link Voltage Ripple for Inverters Operating on Programmed PWM Waveforms", IEEE Transactions on Power Electronics, Vol. 7, No. 1, pp. 171-180, January 1992.
- [8] Hasmukh S. Patel and Richard G. Hoft, "Generalized Techniques of Harmonic Elimination and Voltage Control in Thyristor Inverters: Part I - Harmonic Elimination", IEEE Transactions on Industry Applications, Vol. IA-9, No. 3, pp. 310-317, May/June 1973.
- [9] Hasmukh S. Patel and Richard G. Hoft, "Generalized Techniques of Harmonic Elimination and Voltage Control in Thyristor Inverters: Part II - Voltage Control", IEEE Transactions on Industry Applications, Vol. IA-10, No. 5, pp. 666-673, September/October 1974.
- [10] John A. Houldsworth, Duncan A. Grant, "The Use of Harmonic Distortion to Increase the Output Voltage of a Three-Phase PWM Inverter", IEEE Transactions on Industry Applications, Vol. IA-20, No. 5, pp. 1224-1228, September/October 1984.

REFERENCES

- [11] S.R. Bowes, A. Midoun, "Suboptimal Switching Strategies for Microprocessor-Controlled PWM Inverter Drives", IEE Proceedings, Vol. 132, Pt. B, No. 3, pp. 133-148, May, 1985.
- [12] J.T. Boys, and S.J. Walton, "A Loss Minimized Sinusoidal PWM Inverter", IEE Proceedings, Vol. 132, Pt. B, No. 5, pp. 260-268, September, 1985.
- [13] Giuseppe S. Buja, "Optimum Output Waveforms in PWM Inverters", IEEE Transactions on Industry Applications, Vol. IA-16, No. 6, pp. 830-836, November/December 1980.
- [14] Joachim Holtz, "Optimal Pulsewidth Modulation for AC Servos and Low-Cost Industrial Drives", IEEE Industrial Application Society Annual Meeting, Houston, Texas, 1992.
- [15] Heinz Willi Van Der Broeck, Hans-Christoph Skudelny, Georg Viktor Stanke, "Analysis and Realization of a Pulsewidth Modulator Based on Voltage Space Vectors", IEEE Transactions on Industry Applications, Vol. 24, No. 1, pp. 142-150, January/February 1988.
- [16] J.T. Boys and P.G. Handley, "Harmonic Analysis of Space Vector Modulated PWM Waveforms", IEE Proceedings, Vol. 137, Pt. B, No. 4, pp. 197-204, July 1990.
- [17] Michael A. Boost and Phoivos D. Ziogas, "State-of-the-Art Carrier PWM Techniques: A Critical Evaluation", IEEE Transactions on Industry Applications, Vol. 24, No. 2, pp. 271-280, March/April 1988.
- [18] Ira J. Pitel, "Spectral Errors in the Application of Pulsewidth Modulated Waveforms", IEEE Transactions on Industry Applications, Vol. IA-17, No. 3, pp. 289-295, May/June 1981.
- [19] Duncan A. Grant, John A. Houldsworth, "The Effect of Word Length on the Harmonic Content of Microprocessor-Based PWM Waveform Generators", IEEE Transactions on Industry Applications, Vol. IA-21, No. 1, pp. 218-225, January/February 1985.
- [20] P.D. Evans, P.R. Close, "Harmonic Distortion in PWM Inverter Output Waveforms", IEE Proceedings, Vol. 134, Pt. B, No. 4, pp. 224-232, July, 1987.
- [21] Yoshihiro Murai, Tomofumi Watanabe, and Harumitu Iwasaki, "Waveform Distortion and Correction Circuit for PWM Inverters with Switching Lag-Times", IEEE Transactions on Industry Applications, Vol. IA-23, No. 5, pp. 881-886, September/October 1987.

REFERENCES

- [22] R.C. Dodson, P.D. Evans, H. Tabatabaei, S.C. Harley, "Compensating for Dead Time Degradation of PWM Inverter Waveforms", IEE Proceedings, Vol. 137, Pt. B, No. 2, pp. 73-81, March 1990.
- [23] Seung-Gi Jeong, Min-Ho Park, "The Analysis and Compensation of Dead-Time Effects in PWM Inverters", IEEE Transactions on Industrial Electronics, Vol. 38, No. 2, pp. 108-114, April, 1991.
- [24] Takashi Sukegawa, Kenzo Kamiyama, Katsuhiko Mizuno, Takayuki Matsui, "Fully Digital, Vector-Controlled PWM VSI-FED AC Drives with An Inverter Dead-Time Compensation Strategy", IEEE Transactions on Industry Applications, Vol. 27, No.3, pp. 552-559, May/June 1991.
- [25] Hideki Hashimoto, Hajime Yamamoto, Shigekazu Yanagisawa, and Fumio Harashima, "Brushless Servo Motor Control Using Variable Structure Approach", IEEE Transactions on Industry Applications, Vol. 24, No. 1, pp. 160-170, January/February 1988.
- [26] Khiang-Wee Lim, Teck-Seng Low, M.F. Rahman, and Liang-Boon Wee, "A Discrete Time Variable Structure Controller for a Brushless DC Motor Drive", IEEE Transactions on Industrial Electronics, Vol. 38, No. 2, pp. 102-107, April 1991.
- [27] Geza Joos, Phoivos D. Ziogas, and Donato Vincenti, "A Model Reference Adaptive PWM Technique", IEEE Transactions on Power Electronics, Vol. 5, No. 4, pp. 485-494, October 1990.
- [28] Dong C. Park, Mohamed A. El-Sharkawi, and Robert J. Marks II, "An Adaptively Trained Neural Network", IEEE Transactions on Neural Networks, Vol. 2, No. 3, pp. 334-345, May 1991.
- [29] Donald F. Specht, "A General Regression Neural Network", IEEE Transactions on Neural Networks, Vol. 2, No. 6, pp. 568-576, November 1991.
- [30] Richard P. Brent, "Fast Training Algorithms for Multilayer Neural Nets", IEEE Transactions on Neural Networks, Vol. 2, No. 3, pp. 346-354, May 1991.
- [31] Toshio Fukuda and Takanori Shibata, "Theory and Applications of Neural Networks for Industrial Control Systems", IEEE Transactions on Industrial Electronics, Vol. 39, No. 6, pp. 472-489, December 1992.
- [32] Julio Tanomaru and Sigeru Omatu, "Process Control by On-Lined Trained Neural Controllers", IEEE Transactions on Industrial Electronics, Vol. 39, No. 6, pp. 511-521, December 1992.

REFERENCES

- [33] Youji Iiguni, Hideaki Sakai, and Hidekatsu Tokumaru, "A Nonlinear Regulator Design in the Presence of System Uncertainties Using Multilayered Neural Networks", IEEE Transactions on Neural Networks, Vol. 2, No. 4, pp. 410-417, July, 1991.
- [34] Kumpati S. Narendra and Kannan Parthasarathy, "Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks", IEEE Transactions on Neural Networks, Vol. 2, No. 2, pp. 252-262, March 1991.
- [35] S.R. Bowes, M.J. Mount, "Microprocessor Control of PWM Inverters", IEE Proceedings, Vol. 128, Pt. B, No. 6, pp.293-305, November 1981.
- [36] Maurizio Mazzucchelli, Luciano Puglisi, and Giuseppe Sciutto, "PWM Systems in Power Converters: An Extension of the Subharmonic Method", IEEE Transactions on Industrial Electronics and Control Instrumentation, Vol. IECI-28, No.4, pp. 315-322, November 1981.
- [37] Giuseppe Franzò, Maurizio Mazzucchelli, Luciano Puglisi, and Giuseppe Sciutio, "Analysis of PWM Techniques Using Uniform Sampling in Variable-Speed Electrical Drives with Large Speed Ranges", IEEE Transactions on Industry Applications, Vol. IA-21, No.4, pp. 966-974, July/August 1985.
- [38] Julian Richardson and Osman T. Kukrer, "Implementation of a PWM Regular Sampling Strategy for AC Drives", IEEE Transactions on Power Electronics, Vol. 6, No. 4, pp. 645-655, October 1991.
- [39] John G. Kassakian, Martin F. Schlecht, George C. Verghese, "Principles of Power Electronics", Addison-Wesley Publishing, 1991.
- [40] Roger J. Bassett and Colin Smith, "A GTO Tutorial", PCIM, July, 1989.
- [41] B.W. Williams, J.K. Goodfellow, and F.V.P. Robinson, "GTO Thyristor and Bipolar Transistor Cascode Switches", IEE Proceedings, Vol. 137, Pt. B, No. 3, pp. 141-153, May 1990.
- [42] L.O. Eriksson, J.F. Donlon, and R.S. Chokhawala, "Assignment of Turn-On and Turn-Off Power and Energy Losses in a GTO", IEEE Transactions on Industry Applications, Vol. 27, No. 3, pp. 507-513, May/June 1991.

References

- [43] R.S. Chokhawala and E.I. Carroll, "A Snubber Design Tool for P-N Junction Reverse Recovery Using a More Accurate Simulation of the Reverse Recovery Waveform", IEEE Transactions on Industry Applications, Vol. 27, No. 1, pp. 74-84, January/February 1991.
- [44] B.W. Kernighan and D.M. Ritchie, "The C Programming Language", Prentice-Hall, Inc.

VITA

Lemuel Gregory Barnes III was born in Wattsville, Virginia on November 27, 1951. After graduating from the Prince George County (Virginia) public high school in June, 1970, he pursued undergraduate studies in Electrical Engineering at Virginia Polytechnic Institute & State University, receiving a B.S. degree in Electrical Engineering in June, 1975. During his undergraduate years, he was a member of the Virginia Tech Corp of Cadets (G squadron), and completed the Co-operative Education Program in Electrical Engineering. During his participation in the Co-operative Education Program, he was employed as a Process Control Engineer at the General Electric Company's Industrial Control Products Division in Charlottesville, Virginia.

Between 1975 and 1990, he worked in industry as an electrical engineer in several technical and design positions. From 1975 to 1983, his work involved technical support and design of high-power three-phase uninterruptible power supplies (both solid-state and rotary systems) and diesel-electric generator emergency power supplies. From 1983 to 1990, his work in industry shifted from electric power systems to the hardware and software design of microprocessor-based control and monitoring systems. In December 1985, he received his M.S. degree in Electrical Engineering from Virginia Polytechnic Institute & State University.

He joined the Ph.D. program in Electrical Engineering at Virginia Polytechnic Institute & State University in August 1990.

Lemuel G. Barnes III