This is to certify that the

thesis entitled

FORCE COMMUTATED DIRECT AC-DC CONVERTER
WITH CONTROL OF INPUT POWER FACTOR

presented by

John Gemmell

has been accepted towards fulfillment
of the requirements for

Masters degree in Electrical Engineering

Elias G Strangas

Elias Strangas
Major professor

Date 7/29/96

FORCE COMMUTATED DIRECT AC-DC CONVERTER WITH CONTROL OF INPUT POWER FACTOR

By

John Gemmell

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Electrical Engineering

1996

ABSTRACT

FORCE COMMUTATED DIRECT AC-DC CONVERTER
WITH CONTROL OF INPUT POWER FACTOR

By

John Gemmell

Various methods of implementing a force commutated direct AC-DC converter are reviewed. A new commutation algorithm for the AC-DC converter is derived and a suitable circuit topology chosen for implementation of a prototype converter. The results of simulations of the prototype design are presented with emphasis on regenerative capabilities of the converter. A physical converter was constructed and results of testing are presented. Operation of the physical converter is compared and contrasted with the results of simulation and theoretical expectations. Conclusions and the possible application of the converter are discussed.

ACKNOWLEDGEMENTS

The work contained herein would not have been possible except for the support and guidance of professor Elias Strangas. Initial ideas for the project were greatly influenced by John Vincent, an engineer at the National Superconducting Cyclotron Laboratory at MSU. I would also like to thank professors Wierzba and Schlueter for there time and effort in being on my committee.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

The conversion of power from the 60 or 50 Hertz form in which it is generated and distributed to other forms such as DC or variable frequency AC is required by a vast number of applications. An ideal power converter would provide the required output with minimal input current harmonics and control of the input power . The subject of this thesis deals with the theory and construction of a force commutated power converter to provide DC output with the control of input power factor.

1.1 CYCLOCONVERTERS AND DIRECT FREQUENCY CHANGERS

Power conversion from AC at a fixed frequency to DC or AC of an arbitrary frequency may be accomplished by a rectifier-filter or rectifier-filter-inverter topology respectively. Neither of these topologies provides a regenerative supply or allows for the control of the input power factor. The cyloconverter is a direct frequency changer (DFC) which eliminates the need for intermediate stages in the conversion process. Cyloconverters may be classified into naturally commutated direct frequency changers

(NCDFCs) and forced commutated direct frequency changers (FCDFCs). FCDFCs can provide both a regenerative supply and control of the input power factor.

Naturally commutated direct frequency changers (NCDFCs) are at present the most widely used. Operation of the single phase NCDFC may be described by two independently controlled rectifiers which can create an output voltage equal and opposite to the other rectifier. An AC output voltage is created by cycling between operation of the positive rectifier and negative rectifier. Due to the nature of the commutation scheme, the period of the output frequency of the converter must be an integral multiple of the period of the input frequency. Also, the harmonic content of the output voltage of the NCDFC cannot be improved. For these reasons the applications of NCDFCs has been limited to low speed motor drives.

Force commutated direct frequency changers (FCDFCs) resolve the problems of output frequency limitation and harmonics by allowing switching at frequencies much higher than the input frequency. A diagram of a three phase to single phase FCDFC is shown in Figure 1.1(a). Switches shown in the diagram must be bidirectional and capable of interupting load current on demand. A possible implementation of the bidirectional switch is shown in Figure 1.1(b). In the ideal case, only one switch is closed at any given time. The switches are operated in a cycle with period equal to the sum of the on times for each switch. Commutation of the switches is performed by a pulse width modulation (PWM) algorithm which equates the average output voltage over the switching period to the desired output value. FCDFCs allow bidirectional power flow and the control

algorithm may be developed to improve the input power factor. In general, input current harmonics are determined by the switching frequency.

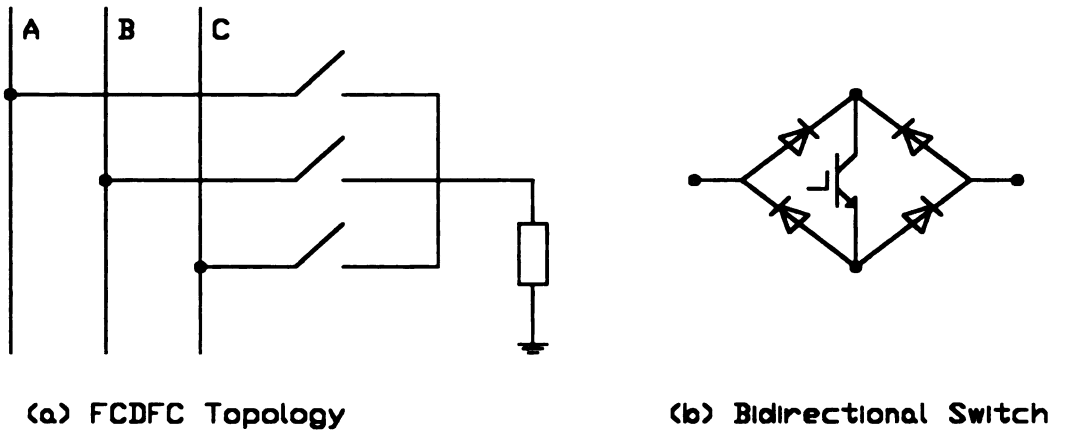


Figure 1.1 - General FCDFC Topology and Bidirectional Switch

Implementation of the FCDFC requires an algorithm which provides timing of the switches to produce the desired input and output. A general condition allowing the design of an algorithm based on the low frequency (fundamental) input and output behavior was presented by Alesina and Venturini[1] using a Fourier analysis approach. In the same paper, a method of generating the switching times was presented in terms of a low frequency modulation matrix. Other algorithms have been presented based on different approaches to the problem. A scalar algorithm, based on comparison of the input phase voltages has been suggested by Roy et. al.[2]. Another approach to the problem, presented by Kim and Eshani[3], is to factor the transfer function matrix into three parts.

Each factored term is then taken to represent a rectifier, filter, or inverter operation. A detailed discussion of each of the above methods is presented in Chapter 2.

A problem concerning the commutation of non-ideal switches must be addressed when considering the circuit topology for a FCDFC. For an inductive load, the current must be continuous. This may be acheived via a break-before-make commutation of the switches at the cost of adding large capacitors. If filter capacitors are to be avoided , a make-before-break commutation scheme must employed which would effectively short circuit the input phases for small periods. A circuit topology using series diodes was presented by Beasant, Beattie and Refsum[4] in which half the commutations are natural. For obvious reasons the scheme was named seminatural commutation. Other schemes which avoid the short circuit problem and reduce switching losses have been presented by Pan, Chen and Sheih[5]. These schemes require a more elaborate topology. A discussion of the circuit topologies discussed above may be found in Chapter 3.

1.2 THE FCDFC APPLIED AS A MAGNET DRIVER

Application of PWM to generate waveforms requires an ouput filter or a load which acts as an output filter. The filter must act as an integrator, since any PWM scheme equates the average of the output over a switching period to the desired value. A large inductance will serve this purpose. Therefore, a DC drive for a magnet would be a suitable application for a FCDFC. The regenerative capability of the FCDFC would also

allow energy stored within the magnet to be returned to the system upon denergization. Conventional DC supplies are nonregenerative and do not provide for control of the input power factor.

The original motivation for this project was provided by John Vincent, an engineer at the National Superconducting Cyclotron facility located on the Michigan State University campus. Large superconducting magnets are used at the facility to provide deflection and focusing of particle beams. Presently, conventional DC sources are used to drive the magnets and the energy stored is dissipated as heat. The application of a FCDFC as a regenerative DC supply could improve operating efficiency, provide reduction in low order current harmonics, and improve the power factor.

The chapters which follow deal with both the theorectical and practical aspects of the design and implementation of a FCDFC as a DC supply for a magnet. Chapter 2 covers the theorectical aspects of the switching control algorithms. Different approaches to the problem of generating the switching times are presented. A solution to the DC problem is presented using the low frequency modulation method. Chapter 3 deals with the physical implementation of the converter. Circuit topologies, commutation problems and the microcontroller interface are all discussed. Chapter 4 covers the results of computer simulations for the AC-DC converter. Ouput waveforms are presented. Chapter 5 details the experimental setup for the AC-DC converter and presents the results

of testing. Finally, Chapter 6 provides conclusions including the practicality of using FCDFCs for this application in the future.

Chapter 2

THEORECTICAL ASPECTS OF FORCE COMMUTATED DIRECT FREQUENCY CHANGERS

Theoretical considerations in the development of a switching algorithm for the FCDFC are discussed below. A rigorous development of conditions relating the input and output necessary for the existence of a switching algorithm were presented by Alesina and Venturini[1]. These conditions are discussed below in Section 2.1 using an intuitive approach. For a rigorous development of the conditions the reader is directed to the original paper[1]. Section 2.2 describes different algorithms presented in the literature and outlines solutions for the case of three phase input to a DC output.

2.1 FOURIER ANALYSIS OF THE GENERALIZED TRANSFORMATION

Conditions necessary for the existence of an algorithm to provide high frequency synthesis of p-outputs given n-inputs has been presented Alesina and Venturini[1]. An intuitive development of these conditions is presented below along with a discussion of the general switching topology for a FCDFC.

Assume a general FCDFC with p-inputs and n-outputs is fed from a set of sources represented by the vector $f_i(t)$ and is to produce a set of outputs represented by vector $f_o(t)$. Let the transformation be given by a switching matrix M(t) as follows,

$$f_o(t) = M(t) \cdot f_i(t) \tag{1}$$

The topology of the generalized switching arrangement labeling inputs, outputs and switches is shown in Figure-2.1. Elements of the switching matrix $M_{ij}(t)$, have a value of 1 if switch S_{ij} is on and zero if S_{ij} is off, where switch S_{ij} connects the ith input to the jth output. In the ideal operation, none of the input lines are ever shorted and none of the output lines are ever open circuited. This places a constraint on the elements of each row of the matrix,

$$\sum_{j=1}^{p} M_{i,j}(t) = 1 \tag{2}$$

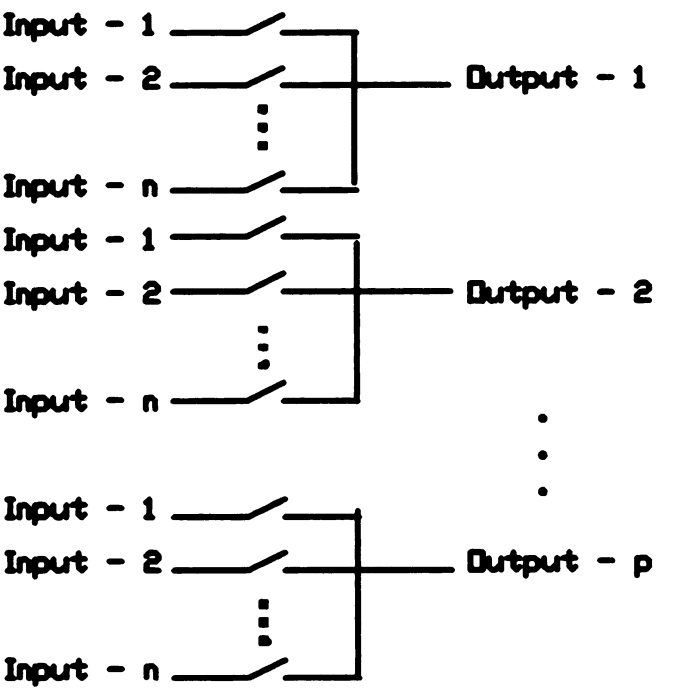


Figure 2.1 - General n-Input, p-Output Converter

Suppose that the desired output of the converter is given by vector, $f_d(t)$, which is bandwidth limited by ω_{bw}. Then the output can be synthesized if the switches can be operated in such a way (i.e. the $M_{ij}(t)$ determined) so that for arbitrary ε there exists ϖ such that,

$$\left|F_o(\omega) - F_d(\omega)\right| < \varepsilon \ , \quad \forall \omega < \varpi \ , \ \omega_{bw} < \varpi \tag{3}$$

where $F(\omega)$ is the Fourier transform of the function f(t) and $f_o(t)$ is given by equation (1). The relation (3) implies that the output spectrum of the converter will be correct for frequencies less than ω_{bw}. If an ideal low pass filter with cutoff frequency between ω_{bw} and ϖ is applied to the output of the converter, the desired output will be achieved. Although equation (3) states the problem, it does not provide insight as to whether the problem admits a solution. The following existence theorem states that the problem admits a solution if and only if ,

$$\inf\{f_i(t)\} \le f_d(t) \le \sup\{f_i(t)\} \tag{4}$$

Therefore the only requirement for a solution to exist is that the functions to be synthesized take on only values between the set of inputs. The existence theorem stated above agrees well with intuition but a rigorous proof may be found in reference [1].

The existence theorem indicates when a solution, the switching matrix M(t), exists but does not provide help in finding a solution. The definition of a low frequency modulation matrix, m(t), corresponding to the switching matrix, M(t), aids greatly in finding a solution. The low frequency modulation matrix may be introduced intuitively by considering a single input, single output converter having a perfect output filter with cutoff frequency ω_f. Suppose the switch is operated at constant frequency $\omega_s >> \omega_f$ with slowly varying pulsewidth. Let the kth pulsewidth interval be denoted by Δ_k , and the kth

switching interval by Δ_k. If the pulsewidth then varies according to equation (5) then the filtered output would be approximated by equation (6).

$$|\Delta'_k| = m(t_k) \cdot |\Delta_k| \qquad \text{for some} \qquad t_k \in \Delta_k \tag{5}$$

$$f_o(t) = m(t) \cdot f_i(t) \tag{6}$$

The argument above provides an intuitive argument in justifying the low frequency modulation function. For a rigorous discussion the reader is again directed to reference [1]. In the n-input, p-output case the above argument directly results in a low frequency modulation matrix of dimension nxp identical to the switching matrix. Each element of the matrix, m(t), corresponds to a particular switch in the converter. The value of $m_{i,j}(t)$ may then be regarded as the average time that switch $S_{i,j}$ is closed on an arbitrarily small interval centered at time t. A constraint corresponding to equation (2) for the switching matrix applies as well to the low frequency modulation matrix in addition to the contraint imposed by equation (6). The constraints on $m_{i,j}(t)$ are summarized below,

$$0 \le m_{i,j}(t) \le 1 \qquad \forall t \tag{7}$$

$$\sum_i m_{i,j}(t) = 1 \qquad \forall t \tag{8}$$

The low frequency modulation matrix is of great utility in finding an appropriate switching algorithm for a FCDFC. Determination of a valid low frequency modulation matrix requires the simultaneous solution to equations (6) through (8). However, this set of equations may not have a unique solution. The following section investigates various methods of how the matrix m(t) may be determined. The different approaches are then applied to the problem of the direct AC to DC converter.

2.2 ALGORITHMS TO GENERATE FCDFC SWITCHING STATES

An introduction to the problem of finding a suitable switching algorithm was presented in the last section with the discussion of the low frequency modulation matrix. The solution to the set of constraint equations in terms of the matrix elements will only be unique and consistent if the number of constraining equations is equivalent to the number of switches. If the problem is under determined, additional constraints may be imposed which provide beneficial results, such as control of the input power factor. The section which follows presents three different methods to the problem of determining the switching state matrix. Two of the methods use an approach based on solution of vector equations containing the low frequency modulation matrix. The third method generates the switching times directly via a scalar algorithm.

2.2.1 LOW FREQUENCY MODULATION METHOD

Constraints on the low frequency modulation matrix must be maintained for any suitable switching state matrix. In general, the set of constraining equations will not be sufficient for the problem to be uniquely determined. Additional desired constraints on current and input power can be asserted which allow the system of equations to be determined. The fundamental constraining voltage equations are given by (6) and (8) with an additional current continuity equation relating the input and output current vectors. These fundamental equations are written below in vector form.

$$\underline{V}_o(t) = \underline{\underline{m}}(t) \cdot \underline{V}_i(t) \tag{9}$$

$$\underline{I}_i(t) = \underline{\underline{m}}^T(t) \cdot \underline{I}_o(t) \tag{10}$$

$$\underline{\underline{m}} \cdot \underline{1}_n = \underline{1}_p \tag{11}$$

where the vectors in equation (11) represent column vectors with all elements unity of dimension n (number of inputs) and p (number of outputs) respectively.

The discussion will now be directed to the solution of the vector equations above for the case of the direct AC-DC converter with topology as shown in Figure-2.2. The

figure shows the case of 3-inputs and 2-outputs corresponding to the practical case of three phase AC as input and the positive and negative DC terminals as output. Writing the fundamental equations (9) through (11) as one system of equations we arrive at equation (12) below.

$$\begin{bmatrix} V_a & V_b & V_c & 0 & 0 & 0 \\ 0 & 0 & 0 & V_a & V_b & V_c \\ I_o & 0 & 0 & -I_o & 0 & 0 \\ 0 & I_o & 0 & 0 & -I_o & 0 \\ 0 & 0 & I_o & 0 & 0 & -I_o \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} m_{1,1} \\ m_{1,2} \\ m_{1,3} \\ m_{2,1} \\ m_{2,2} \\ m_{2,3} \end{bmatrix} = \begin{bmatrix} V_o/2 \\ -V_o/2 \\ I_a \\ I_b \\ I_c \\ 1 \\ 1 \end{bmatrix} \tag{12}$$

The system can now be reduced by Gaussian elimination. Subtracting row 3 above multiplied by the appropriate constant to eliminate column one yields equation (13).

$$\begin{bmatrix} V_b & V_c & V_a & 0 & 0 \\ 0 & 0 & V_a & V_b & V_c \\ I_o & 0 & 0 & -I_o & 0 \\ 0 & I_o & 0 & 0 & -I_o \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} m_{1,2} \\ m_{1,3} \\ m_{2,1} \\ m_{2,2} \\ m_{2,3} \end{bmatrix} = \begin{bmatrix} V_o/2 - I_a V_a / I_o \\ -V_o/2 \\ I_b \\ I_c \\ 1 - I_a / I_o \\ 1 \end{bmatrix} \tag{13}$$

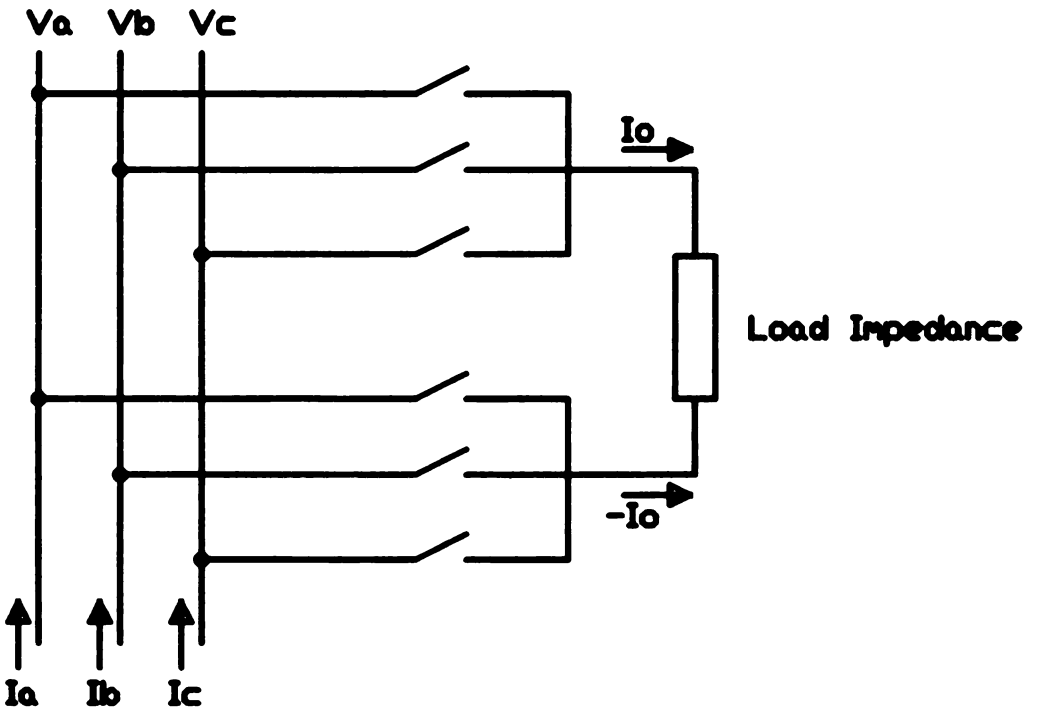


Figure 2.2 - Direct AC-DC Converter Topology

Further reduction of the system of equations is next performed using row 3 of equation (13) to again eliminate the first column resulting in the following equation.

$$\begin{bmatrix} V_c & V_a & V_b & 0 \\ 0 & V_a & V_b & V_c \\ I_o & 0 & 0 & -I_o \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} m_{1,3} \\ m_{2,1} \\ m_{2,2} \\ m_{2,3} \end{bmatrix} = \begin{bmatrix} V_o/2 - \left(I_a V_a + I_b V_b\right)/I_o \\ -V_o/2 \\ I_c \\ 1 - \left(I_a + I_b\right)/I_o \\ 1 \end{bmatrix} \tag{14}$$

Reduction of the equation above once more will finally produce a form which provides useful insight into the problem. Using row 4 of equation (14) to eliminate column 1 yields equation (15). Equation (15) can be viewed as four linear algebraic equations in three unknowns. For this set of equations to be consistent, it is immediately obvious that the relations of equations (16) and (17) must be satisfied. These last two relations are merely a statement of conservation of energy and Kirchoff's current law applied to the converter. Since the input and output voltages and currents must satisfy these relations, we are left with equation (18).

$$\begin{bmatrix} V_a & V_b & V_c \\ V_a & V_b & V_c \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} m_{2,1} \\ m_{2,2} \\ m_{2,3} \end{bmatrix} = \begin{bmatrix} V_o/2 - \left(I_a V_a + I_b V_b + I_c V_c\right)/I_o \\ -V_o/2 \\ 1 - (I_a + I_b + I_c)/I_o \\ 1 \end{bmatrix} \tag{15}$$

$$V_o I_o = I_a V_a + I_b V_b + I_c V_c \tag{16}$$

$$0 = I_a + I_b + I_c \tag{17}$$

$$\begin{bmatrix} V_a & V_b & V_c \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} m_{2,1} \\ m_{2,2} \\ m_{2,3} \end{bmatrix} = \begin{bmatrix} -V_o / 2 \\ 1 \end{bmatrix} \tag{18}$$

We are now free to choose constraints on $m_{2,1}$, $m_{2,2}$, $m_{2,3}$ subject to equation (18). As yet we have not stated any relations between the input and output voltages and currents. Given a balanced set of three phase input voltages, we would like to specify the modulation matrix elements in such a manner that the input currents are also a balanced three phase set at a specified power factor. The general form of the desired input voltages and currents is as shown in equations (19) and (20). Input currents chosen in such a manner will automatically satisfy the requirement of equation (17). The power conservation equation of (16) can now be expressed in terms of the balanced three phase power relation as stated in equation (21).

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} V_i \cos(\omega \cdot t) \\ V_i \cos(\omega \cdot t + 2\pi / 3) \\ V_i \cos(\omega \cdot t + 4\pi / 3) \end{bmatrix} \tag{19}$$

$$\begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} = \begin{bmatrix} I_i \cos(\omega \cdot t + \Theta) \\ I_i \cos(\omega \cdot t + 2\pi / 3 + \Theta) \\ I_i \cos(\omega \cdot t + 4\pi / 3 + \Theta) \end{bmatrix} \tag{20}$$

$$V_o I_o = 3 V_i I_i \cos(\Theta) / 2 \tag{21}$$

Now that the input quantities have been specified, relations for $m_{2,1}$, $m_{2,2}$, $m_{2,3}$ may be found subject to equation (18). Remembering that the solution will not be unique, we are free to choose a solution. It is easily shown that a solution consistent with (18) through (21) is given by equation (22).

$$\begin{bmatrix} m_{2,1} \\ m_{2,2} \\ m_{2,3} \end{bmatrix} = \begin{bmatrix} 1/3(1 - \Omega \cos(\omega \cdot t + \Theta)) \\ 1/3(1 - \Omega \cos(\omega \cdot t + 2\pi / 3 + \Theta)) \\ 1/3(1 - \Omega \cos(\omega \cdot t + 4\pi / 3 + \Theta)) \end{bmatrix} \tag{22}$$

where, $\Omega = V_o / V_i \cos(\Theta)$

The next step in finding the solution will be to find relations for $m_{1,1}$, $m_{1,2}$, $m_{1,3}$ consistent with the fundamental equations as expressed in (12), the input quantities as stated in (19) and (20) and the previously determined values for $m_{2,1}$, $m_{2,2}$, $m_{2,3}$ found in (22). Again a

solution is found by inspection and verified. It can be easily shown that the solution expressed in (23) below is consistent with all requirements. The final expression for the modulation matrix has dimension 2x3 and may then be expressed as shown in (24) in terms of the elements of (22) and (23) .

$$\begin{bmatrix} m_{1,1} \\ m_{1,2} \\ m_{1,3} \end{bmatrix} = \begin{bmatrix} 1/3(1+\Omega\cos(\omega \cdot t + \Theta)) \\ 1/3(1+\Omega\cos(\omega \cdot t + 2\pi/3 + \Theta)) \\ 1/3(1+\Omega\cos(\omega \cdot t + 4\pi/3 + \Theta)) \end{bmatrix} \tag{23}$$

$$\underline{\underline{m}}(t) = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} \\ m_{2,1} & m_{2,2} & m_{2,3} \end{bmatrix} \tag{24}$$

The modulation matrix elements must also be constrained by () to lie between zero and one for all times. The allowed values of Ω which may be used are such that $0 < \Omega < 1$, corresponding to a maximum voltage transfer ratio of $V_o/V_i\cos(\Theta)$.

Although the above development follows the general approach of reference [1], a different variation for developing the modulation matrix will now be presented. The impetus for pursuing this method lies in the microprocessor hardware implementation which is explained in Chapter 3.

The underlying concept in this new approach lies in viewing the output as a single voltage created by all six switches. In the previous discussion, the output voltage was considered as being created by two separate output phases, each being determined by a set of three switches. The output across the load was then viewed as the difference between the constant voltages created at the two output phases. While the old method impressed phase voltages at the outputs, the new method will directly treat the output voltage in terms of phase to phase quantities. The method previously developed shall be refered to as the phase to neutral algorithm, while the new method shall be denoted as the phase to phase algorithm. The new system will now be viewed as having one output and six inputs as described by equations (25) and (26) below.

$$\begin{bmatrix} V_{ab} & V_{bc} & V_{ca} & V_{ba} & V_{cb} & V_{ac} \end{bmatrix} \cdot \begin{bmatrix} m_{ab} \\ m_{bc} \\ m_{ca} \\ m_{ba} \\ m_{cb} \\ m_{ac} \end{bmatrix} = V_o \tag{25}$$

$$I_o = \begin{bmatrix} m_{ab} & m_{bc} & m_{ca} & m_{ba} & m_{cb} & m_{ac} \end{bmatrix} \cdot \begin{bmatrix} I_{ab} \\ I_{bc} \\ I_{ca} \\ I_{ba} \\ I_{cb} \\ I_{ac} \end{bmatrix} \tag{26}$$

The elements of the modulation matrix for the phase to phase algorithm correspond to a set of two switches as opposed to a single switch for the phase to neutral method. Using a constraint equation corresponding to equation (11) and writing V_{ji} as $-V_{ij}$ and I_{ji} as $-I_{ij}$ we may write the entire system as shown in (27) below.

$$\begin{bmatrix} V_{ab} & V_{bc} & V_{ca} & -V_{ab} & -V_{bc} & -V_{ca} \\ 1 & 1 & 1 & 1 & 1 & 1 \\ I_o & 0 & 0 & -I_o & 0 & 0 \\ 0 & I_o & 0 & 0 & -I_o & 0 \\ 0 & 0 & I_o & 0 & 0 & -I_o \end{bmatrix} \cdot \begin{bmatrix} m_{ab} \\ m_{bc} \\ m_{ca} \\ m_{ba} \\ m_{cb} \\ m_{ac} \end{bmatrix} = \begin{bmatrix} V_o \\ 1 \\ I_{ab} \\ I_{bc} \\ I_{ca} \end{bmatrix} \tag{27}$$

Reducing the system above by Gaussian elimination yields the following equations.

$$V_o I_o = V_{ab} I_{ab} + V_{bc} I_{bc} + V_{ca} I_{ca} \tag{28}$$

$$m_{ab} + m_{bc} + m_{ca} = (1 - (I_{ab} + I_{bc} + I_{ca}))/2 \tag{29}$$

Equation (28) is just a statement of conservation of energy and will automatically be satisfied. If the input currents and voltages are constrained to be balanced three phase, then the currents no longer appear in (29). The following equations then are constraints on the matrix elements.

$$m_{ab} + m_{bc} + m_{ca} = 1/2 \tag{30}$$

$$m_{ba} + m_{cb} + m_{ac} = 1/2 \tag{31}$$

As before we shall write the voltages and currents as balanced three phase sinusoids. These relations are written below as equations (32) and (33).

$$\begin{bmatrix} V_{ab} \\ V_{bc} \\ V_{ca} \end{bmatrix} = \begin{bmatrix} V_i \cos(\omega \cdot t) \\ V_i \cos(\omega \cdot t + 2\pi / 3) \\ V_i \cos(\omega \cdot t + 4\pi / 3) \end{bmatrix} \tag{32}$$

$$\begin{bmatrix} I_{ab} \\ I_{bc} \\ I_{ca} \end{bmatrix} = \begin{bmatrix} I_i \cos(\omega \cdot t + \Theta) \\ I_i \cos(\omega \cdot t + 2\pi / 3 + \Theta) \\ I_i \cos(\omega \cdot t + 4\pi / 3 + \Theta) \end{bmatrix} \tag{33}$$

Using equations (32) and (33) we may rewrite the last three rows of (27) as the three equations shown below as (34).

$$\begin{aligned} m_{ab} - m_{ba} &= I_i \cos(\omega \cdot t + \Theta) / I_o \\ m_{bc} - m_{cb} &= I_i \cos(\omega \cdot t + 2\pi / 3 + \Theta) \\ m_{ca} - m_{ac} &= I_i \cos(\omega \cdot t + 4\pi / 3 + \Theta) \end{aligned} \tag{34}$$

Now we may find a solution by inspection which will satisfy all constraints. Using (30), (31) and (34) it is easily shown that a solution to the problem is given by the expression below. It should again be noted that the solution is not unique.

$$\begin{bmatrix} m_{ab} \\ m_{bc} \\ m_{ca} \\ m_{ba} \\ m_{cb} \\ m_{ac} \end{bmatrix} = \begin{bmatrix} (1+3I_i\cos(\omega\cdot t+\Theta)/I_o)/6 \\ (1+3I_i\cos(\omega\cdot t+2\pi/3+\Theta)/I_o)/6 \\ (1+3I_i\cos(\omega\cdot t+4\pi/3+\Theta)/I_o)/6 \\ (1-3I_i\cos(\omega\cdot t+\Theta)/I_o)/6 \\ (1-3I_i\cos(\omega\cdot t+2\pi/3+\Theta)/I_o)/6 \\ (1-3I_i\cos(\omega\cdot t+4\pi/3+\Theta)/I_o)/6 \end{bmatrix} \tag{35}$$

For the elements of the modulation matrix to be valid, they must only take on values between zero and one as expressed in (7). This inequality is expressed in terms of the input and output currents, however we would like the relationship in terms of input and output voltages. Using an expression for the three phase power we can then express the inequality in terms of voltages as shown in (36). The actual limit on the output voltages is then expressed by (37).

$$V_oI_o = 3V_iI_i\cos(\Theta)/2 \Rightarrow 3I_i/I_o = 2V_o/V_i\cos(\Theta) \tag{36}$$

$$V_o \le V_i\cos(\Theta)/2 \tag{37}$$

A comparison of the maximum voltage output possible from the previous development of the phase to neutral algorithm expressed in (22) through (24) to that of (37) shows that the second method yields a voltage transfer ratio of 0.87 with respect to that of the first method. Although this is a disadvantage, the phase to phase method can be implemented more easily using microprocessor based control. This subject is discussed in Chapter #3.

2.2.2 THREE STAGE RECTIFIER-FILTER-INVERTER METHOD

A variation of the low frequency modulation method is obtained by factoring the modulation matrix into three component matrices. Each matrix is then viewed as a rectifier, filter and inverter operation respectively. For this reason, the method is often called indirect PWM while that of the previous section is referred to as direct PWM. An investigation of the relative merits of indirect PWM versus direct PWM has been performed by Ziogas et al[6].

A detailed discussion for the direct conversion of 3-phase at given frequency to 3-phase at an arbitrary frequency was performed by Kim and Eshani [3]. The discussion below will be an adaptation of the method to the case of direct 3-phase AC to DC with two output terminals. Using the same methodology as reference [3] the output will as expressed in equation (38) below.

$$\underline{V}_o = \underbrace{\underline{h}_{inv} h_f(t) \underline{h}_{rect}(t)}_{\underline{m}(t)} \underline{V}_i(t) \tag{38}$$

The modulation matrix of section 2.2.1 for the phase to neutral algorithm will be analogous to the product of all three matrices in (38) above. The filter and rectifier functions will be equivalent to the AC output case and may take the form shown in (39) and (40) below [3]. The inverter function will have constant elements for the DC output case and is shown in equation (41).

$$\begin{aligned} h_f(t) = \sum_{\ell} \Bigg\{ & \frac{K(-1)^{\ell}}{\cos(\omega t)} \left[u(\omega t + \pi/6 - \pi\ell) - u(\omega t - \pi/6 - \pi\ell) \right] \\ & + \frac{K(-1)^{\ell}}{\cos(\omega t - 2\pi/3)} \left[u(\omega t - \pi/2 - \pi\ell) - u(\omega t - 5\pi/6 - \pi\ell) \right] \\ & + \frac{K(-1)^{\ell}}{\cos(\omega t - 4\pi/3)} \left[u(\omega t - 7\pi/6 - \pi\ell) - u(\omega t - 3\pi/2 - \pi\ell) \right] \Bigg\} \end{aligned} \tag{39}$$

$$\underline{h}_{rect}(t) = \begin{bmatrix} \sum_{\ell} \left[u(\omega t + \pi/6 - \pi\ell) - u(\omega t - \pi/6 - \pi\ell) \right](-1)^{\ell} \\ \sum_{\ell} \left[u(\omega t - \pi/2 - \pi\ell) - u(\omega t - 5\pi/6 - \pi\ell) \right](-1)^{\ell} \\ \sum_{\ell} \left[u(\omega t - 7\pi/6 - \pi\ell) - u(\omega t - 3\pi/2 - \pi\ell) \right](-1)^{\ell} \end{bmatrix} \tag{40}$$

where u(t) is the unit step function and $K=1/V_i$ is the gain of the filter.

$$\underline{h}_{inv} = \begin{bmatrix} V_o/2 \\ -V_o/2 \end{bmatrix} \tag{41}$$

Substituting (39) through (41) into (38) the DC voltage developed across the load is then given by the following expression.

$$V_o = \begin{bmatrix} 1 & -1 \end{bmatrix} \underline{V}_o = \begin{bmatrix} 1 & -1 \end{bmatrix} \underline{h}_{inv} h_f(t) \underline{h}_{rect}(t) \underline{V}_i(t) \tag{42}$$

It should be noted that the method above did not take into account the input current to the device and therefore will not provide for the control of input power factor. Although the approach uses the familiar analogies of rectifier and filter stages, the lack of ability to control the input power factor is a severe limitation.

2.2.3 SCALAR ALGORITHM

A scalar algorithm has been proposed and tested by Roy et al[4]. This scalar algorithm applies to only a balanced three phase set of input voltages. The method separates the domain of the input into regions over which two given input phases have opposite polarity to the remaining phase. For a set of balanced three phase inputs, the entire time domain may be divided into such regions. A derivation of the scalar algorithm is presented below.

The following equations are the basic scalar equations relating to output voltage and switching time intervals for any 3-phase DFC scheme.

$$V_o = \frac{V_k t_k + V_l t_l + V_m t_m}{T_S} \tag{43}$$

$$t_k + t_l + t_m = T_S \tag{44}$$

In equations (43) and (44), T_S represents the switching period, V_k the input voltage on phase k, t_k the switching pulsewidth of phase k, and k,l,m represents any permutation of the input phases A,B,C. Suppose that at time t input phase m has polarity opposite to both phases k and l. Also suppose that k and l are chosen such that equation (45) holds. Now t_k and t_l are chosen such that the equality in (46) is maintained. Note that (46) places a constraint on the problem and will allow the problem to have a unique solution.

$$0 \le \frac{V_k}{V_l} \le 1 \tag{45}$$

$$\frac{t_k}{t_l} = \frac{V_k}{V_l} \equiv \rho_{kl} \tag{46}$$

Substituting the above equation into the basic scalar DFC equations (43) and (44) and then solving for t_l , t_k , and t_m yields the equations below. Before the equations are

applied, the correct permutation of the input phases corresponding to k , l , and m must be determined in accordance with (45) and (46).

$$t_l = \frac{(V_o - V_m)T_S}{V_k \rho_{kl} + V_l - V_m(1+\rho_{kl})} \tag{47}$$

$$t_k = \rho_{kl} t_k \tag{48}$$

$$t_m = T_S - (1+\rho_{kl})t_l \tag{49}$$

Note that the duration over which a particular permutation remains valid corresponds to an angular change of $\pi/6$. The equation which applies to the pulsewidth of a particular switch changes with the same frequency.

The evaluation of power factor for the balanced 3-phase output case using the scalar algorithm can be found in reference [4]. For this case the scalar algorithm achieves a unity power factor. The following expression for the input current applying to the topology of Figure 2.2 can easily be derived from the previous discussion. Note that the input current will have the same phase as the input voltage only if the input power remains constant. Therefore the scalar algorithm applied to the DC output case will attain a unity

power factor only in steady state operation. The vector methods found using the low frequency modulation matrix in section 2.2.1 did not suffer this limitation.

$$I_l = \frac{V_o I_o V_l}{V_k^2 + V_l^2 + V_m^2} = \frac{V_o I_o V_l}{1.5 V_i^2} \tag{50}$$

Chapter 3

PHYSICAL IMPLEMENTATION OF THE FCDFC

The implementation of a FCDFC requires that the switches comprising the converter be commutated to change the connection of the output phases between any of the input phases. Switching algorithms, previously discussed, assumed ideal commutation of the switches so that the input phases are never shorted and the output phases never open circuited. In a practical device, ideal commutation of the switches can not be achieved. Either a 'make before break' or 'break before make' switching strategy must be employed. In the later case, large output filter capacitors must be employed if the load is highly inductive. If large output filter capacitors are to be avoided, a make before break commutation scheme must be used. This commutation scheme must tolerate the short circuit of the input phases for small periods. Circuit topologies of different commutation schemes and aspects of the physical implementation of FCDFCs are presented below.

3.1 CIRCUIT TOPOLOGIES AND COMMUTATION

Different circuit topologies and switching strategies have been presented in the literature which address the FCDFC commutation problem. The topologies which are discussed below provide an overview of different strategies. All assume 'make before break' commutation of the switches.

3.1.1 STAGGERED COMMUTATION SCHEME

A simple topology for the FCDFC is attained using bidirectional switches and is shown in Figure-3.1 for the single output case. The input filter capacitors must be included to allow switching of the input phases with a non-zero source inductance. A working implementation of the scheme has been achieved by numerous groups including Alesina and Venturini[1], and Pan and Chen[5].

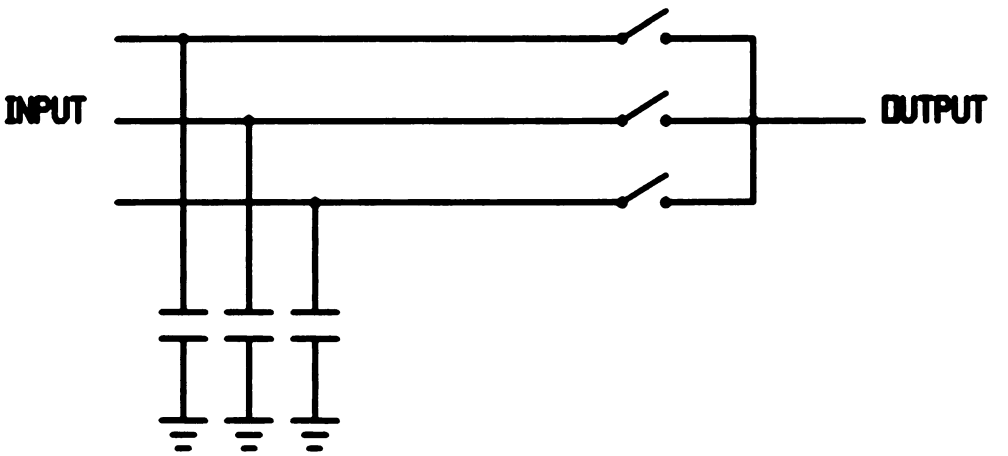


Figure 3.1 - Single Output FCDFC with Input Filter Capacitors

The commutation problem was solved by Alesina and Venturini in reference [1] using a strategy they termed "staggered commutation". Implementation of the bidirectional switch includes two antiparallel transistors. The staggered commutation scheme is described by Figure-3.2 and the sequence of events listed below.

1) Determine whether the switch turning on is at a lower voltage or higher voltage than the switch turning off. The two devices in the switches which allow a current flow outward from the lower voltage switch and inward to the higher voltage switch are termed "free wheeling " devices for the events of step (2) below.

2) Stagger the commutation by sequentially,

a) turning on the freewheeling portion of the incoming switch,

b) turning off the nonfreewheeling portion of the outgoing switch,

c) turning on the nonfreewheeling portion of the incoming switch,

d) turning off the freewheeling portion of the outgoing switch.

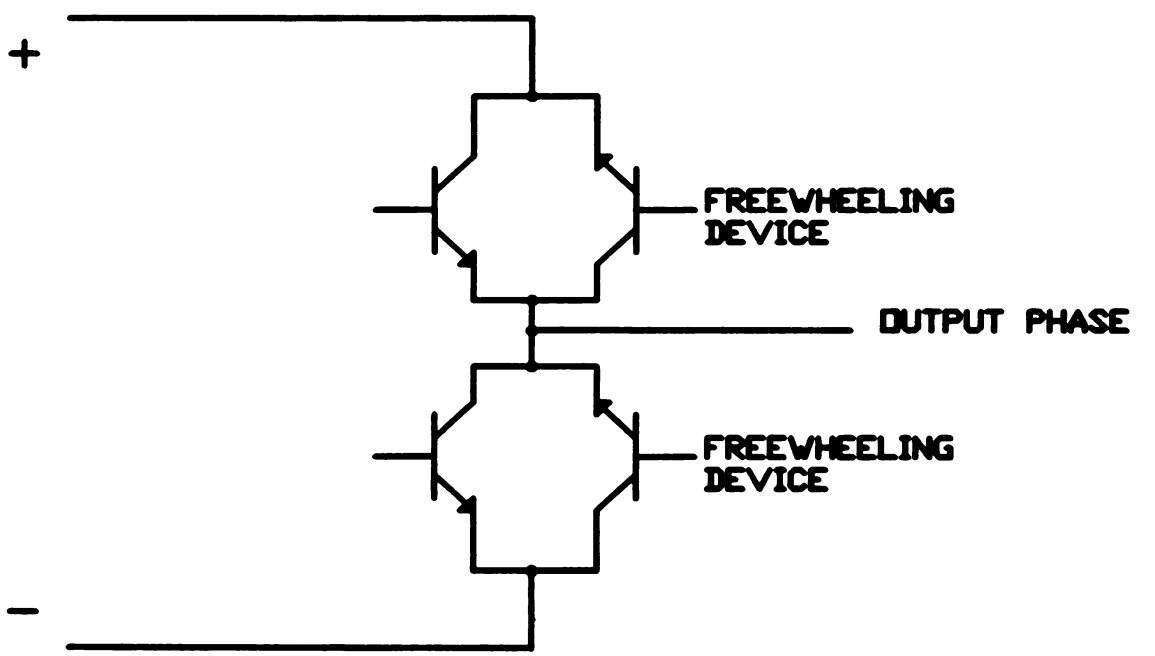


Figure 3.2 - Staggered Commutation Scheme

Operation of the "free wheeling" devices in the staggered commutation strategy are analogous to the operation of the free wheeling diodes in a standard inverter bridge. Therefore this type of commutation scheme does not require snubbering. A disadvantage of the scheme is the complicated switching strategy and the required comparison of the input phase voltages.

3.1.2 THE ACTIVE CHOKE

Another more elaborate topology introduced by Beasant and Refsum [4] which uses bidirectional switches is shown in Figure 3.3. The essential idea is to provide an interface circuit between the load and the switch matrix which would appear as zero impedance to load currents but appear as an infinite impedance to short circuit currents caused by the commutation. The multiwinding transformer and diodes rectify the flux in the core due to currents entering from the three phase source. Additional windings which carry only load current create a flux within the core which opposes the flux created by the load current flowing through the coils of the input phase windings. Therefore the current which circulates between phases will cause a rectified flux in the core with no opposing flux from the compensating windings since this no similar current flows through the load. The impedance seen connecting the load across any of the source inputs will be only the leakage reactance. However short circuits of the source due to commutation will see magnetizing reactance. The above discussion of course assumes that the turns of the input windings and compensating coils are properly matched.

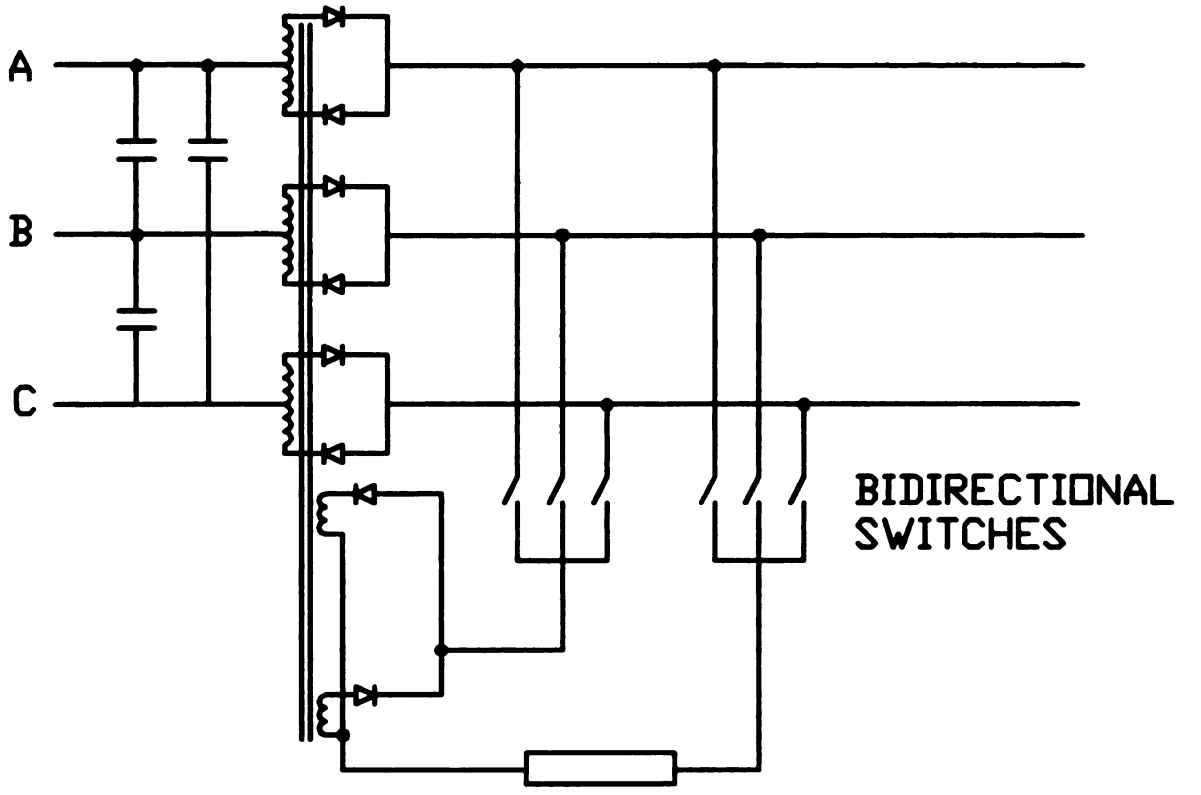


Figure 3.3 - Active Choke Topology

3.1.3 SEMINATURAL COMMUTATION

A simpler FCDFC topology was also investigated in [4] which solves the commutation problem. The topology requires only unidirectional switches which may be implemented using any transistor with the appropriate voltage, current and switching capability. Diodes are used in series with the transistors and provide reverse blocking during commutation. The scheme is illustrated in Figure 3.4 below. Two separate stages are used to provide 4-quadrant operation. The commutation control is provided by only

one stage at a time. A current zero crossing detector is required to control the transfer of commutation between the positive and negative conducting stages.

When the output is to be commutated between source phases two situations occur with equal frequency. The first situation is that the outgoing source phase is at a higher voltage than the incoming source phase. When the incoming phase is switched on, the diode on the incoming phase is reversed biased and does not conduct. The outgoing phase is then opened by the controlling commutation signal. Therefore the commutation of the switch for this situation is forced. Consider now the second possibility where the outgoing source phase is at a lower voltage than the incoming phase. When the incoming switch initially closes, the diode is forward conducting but reverse biased. If the overlap of the 'make before break' commutation exceeds the reverse recovery time of the diode then the switch commutation is natural. Therefore, for this topology half the commutations are provided naturally from the series diodes. For this reason the scheme has been called "seminatural commutation". The scheme requires relatively few components and has a simple commutation scheme. The only complication in the commutation is the zero current crossing detection to provide for 4-quadrant operation. For two quadrant operation, the commutation scheme is extremely simple and the required components are minimal. For this reason, the scheme was adopted for the experimental work of this thesis.

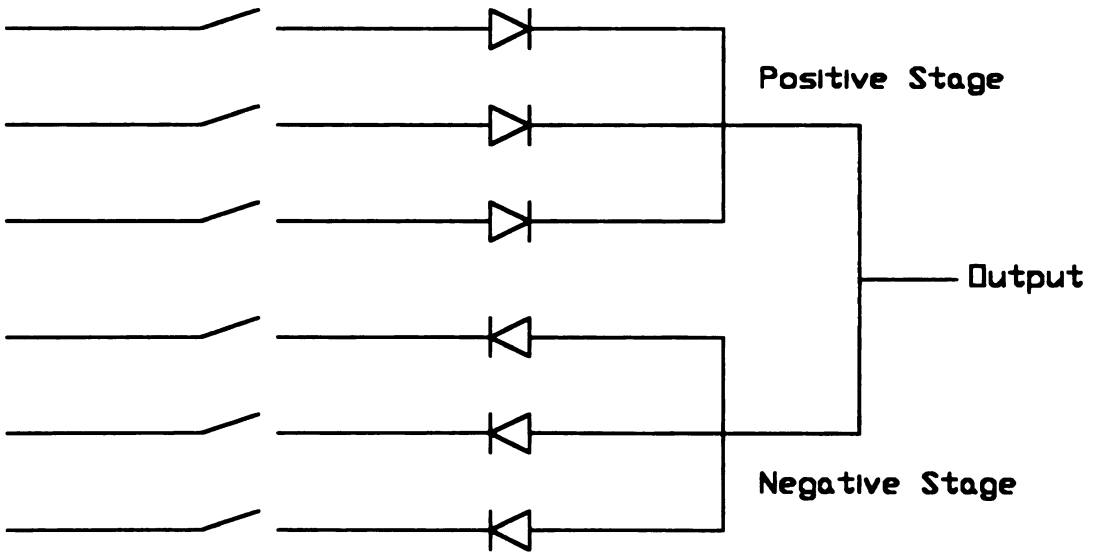


Figure 3.4 - Seminatural Commutation Topology

3.1.4 LOW SWITCHING LOSS TOPOLOGY

The last circuit topology to be discussed is also the most complicated. The topology to be presented here is based entirely on the work of Pan et.al. [5]. Central to the scheme is the implementation of the bidirectional switch shown in Figure 3.5. The switch uses a rectifier bridge, similar to the simple bidirectional switch of Figure 1.2, but implements the unidirectional switch with two transistor-diode conducting paths and a capacitor. Figure 3.6 (a) through (d) show the four different conducting states which are attainable. In the initial turn on configuration (a), the transistors conduct until the capacitor is discharged from its previous blocking voltage. The switch then enters the

steady state turn on configuration (b) during which the capacitor plays no role. When the transistors are then turned off, an initial turn off state (c) ensues in which the diodes conduct and the capacitor is again charged. After the capacitor is completely charged the switch enters the steady state off configuration (d). The capacitor acts as a snubber during commutation preventing voltage spikes across the transistors.

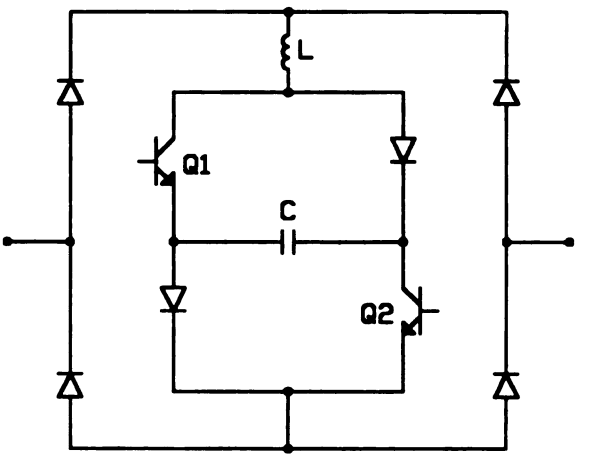


Figure 3.5 - Zero Loss Bidirectional Switch

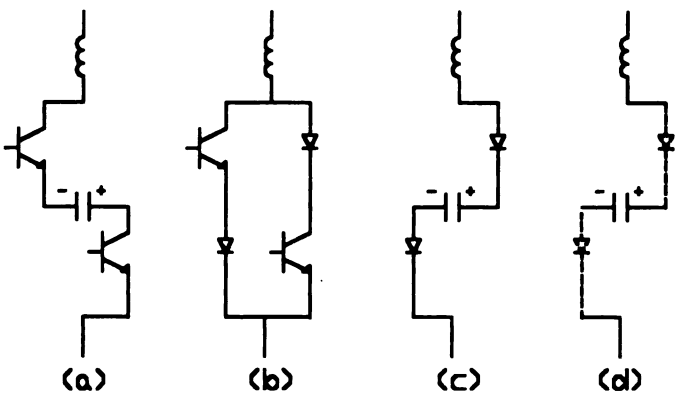


Figure 3.6 - Zero Loss Switch Conduction States

During initial turn on, the blocking voltage across the capacitor implies that the initial current through the switch will be zero. Therefore zero current switching occurs at turn on. At turn off, the voltage across each transistor is zero if the capacitor is completely discharged. Therefore zero voltage switching occurs at turnoff. So if the time constants of the LC circuit is such that the capacitor is able to fully charge and discharge within the switching interval, zero switching loss will be achieved. Reference [5] shows that the condition for the above to be true is as shown in (51) below where the S and R subscripts denote the switching and resonant LC circuit frequencies respectively.

$$f_R > \frac{f_S}{2} \tag{51}$$

In addition to zero loss, the bidirectional switch proposed above has advantage of a simple commutation scheme. Both transistors in the switch are simultaneously signaled off and on. No separate logic is required to determine modes of operation dependent on voltage or current polarity as found in the staggered and seminatural commutation schemes. The only disadvantage of the proposed switch is the additional components required by the topology.

3.2 POWER ELECTRONIC COMPONENTS

Electronic components with suitable ratings and abilities must be found to implement the topologies of the last section. Rectifier diodes and transistors used as unidirectional switches are all that is required.

Power switching diodes must be capable of carrying the maximum current required by the FCDFC and be able to commutate at a frequency well beyond the switching frequency used by the algorithm. Line frequency rectifier diodes are unacceptable. The speed of a diode is determined by t_{rr} , the time it takes for charge-carriers to be swept out of the junction when reverse biased creating a depletion region. The actual time is dependent upon conditions external to the diode. Standard conditions include initial current of πI_f with a change in current of $25\mu A$/second. Conversely the reverse recovery ability of a diode may be characterized by Q_{rr} , the charge which must be swept out of the junction to create the depletion region. The value of Q_{rr} is independent of conditions external to the diode. Available power switching diodes are categorized based their reverse recovery time. Fast recovery diodes have t_{rr} values of several microseconds or less. Ultrafast recovery diodes are available with t_{rr} values as low as a few nanoseconds. The choice of acceptable diode recovery time is dependent upon the switching speed used in the commutation. If distortions due to diode reverse recovery are to be avoided, the reverse recovery time must be small with respect to the switching period. Also, certain topologies, such as seminatural commutation, place requirements on the maximum reverse recovery time.

The present choice for high power switching is the insulated gate bipolar transistor (IGBT). The IGBT has a structure which provides a low on-state voltage similar to a BJT with input drive requirements similar to a MOSFET. IGBTs are currently available capable of switching frequencies of 30kHz . A SPICE model for the IGBT will be discussed in Chapter 4. Packaging available for IGBTs is similar to other power transistors. Available case styles include pin out, and flat pack configurations.

Drivers for discrete IGBTs are widely available from a number of manufacturers. These drivers have minimal power requirements. Most driver circuits contain optoisolators to provide isolation of the commutation signal from the power circuit. Each isolated IGBT will require a driver supplied by an isolated power supply.

3.3 SNUBBER CAPACITORS AND LOW INDUCTANCE BUS WORK

The circuit topologies of section 3.1 did not address parasitic inductance of electrical connections and capacitors. In the discussion of those schemes it was assumed that the electrical connections from transistors are ideal, without any inductance. The transistor could then interrupt arbitrary current without introducing a di/dt voltage spike. It was also assumed that capacitors used in these schemes could support current discontinuities. Obviously, any physical electrical connection or device will contain some parasitic inductance. Extremely low inductance connections may be designed by sandwiching thin copper bars together. This type of bus work also has the added benefit

of reducing stray magnetic flux and EMI. A discussion of the design of low inductance bus may be found reference [?]. Specialized kits for designing prototype bus connections are available. These kits allow the fabrication of specialized bus work with little effort.

Low inductance capacitors are now available for snubbers and applications such as the FCDFC topologies discussed earlier. Besides the normal microfarad rating, these capacitors have an additional rating in units of kV/μsec. This rating reflects the capacitor units ability to conduct current imposed by voltage spikes. . With regard to the design of a FCDFC, the inductance of bus work and capacitors must be made small enough to ensure the proper operation of the circuit topology chosen.

3.4 MICROCONTROLLER AND INTERFACE ELECTRONICS

Algorithms for creating the switching states of a FCDFC have been discussed, however a method of physically generating the control signals has not been addressed. Although the discussion which follows is directed toward the low frequency modulation method, the ideas apply equally well to any FCDFC algorithm. The switching states may be implemented using the parallel port of a microprocessor. Each pin of the parallel port provides the control signal for one switch. Switching states are commuted by calculation of the pulsewidth for a given switching state as a fraction of T_s, the total switching period. The fractional value above is the evaluation of the appropriate element of the low frequency modulation matrix at that particular time. Since we must write to all pins of the parallel port simultaneously, we would prefer an algorithm which views all

switch positions as a single state. The phase-to-neutral algorithm found in section 2.2.1 does not achieve this goal. The states for the switches associated with the positive output are found independently of the states found for the negative output. For this case, the sequence of parallel port states commutated within a switching period will change. This complicates the generation of a switching sequence. The phase to phase algorithm, also developed in section 2.2.1, finds the positions of all switches as a single state. The switching sequence will then be constant within the switching period. A sequence may then be determined which minimizes the number of commutations of each switch within the switching period. Table 3.1 below shows an ordering of states for the phase to phase algorithm which minimizes the number of commutations. In the table, the P and N subscripts denote switches connected to either the positive or negative output terminal. The '+' and '-' symbols denote a conducting or nonconducting state respectively. Conducting states shown determine the of the six switches of Figure 3.4.

Table 3.1 - Phase-to-Phase Algorithm Switching States

State Position	A_P	B_P	C_P	A_N	B_N	C_N
1	+	-	-	-	+	-
2	+	-	-	-	-	+
3	-	+	-	-	-	+
4	-	+	-	+	-	-
5	-	-	+	+	-	-
6	-	-	+	-	+	-

The microcontroller commutates between switching states by calculating pulsewidths for each particular state and loading these states to an internal timer.

Simultaneously the switching state is written to the parallel port. When the timer has reached the value which has been loaded, an interrupt is generated which causes the time for the next state to be loaded and the process repeats. If a table of switching states versus switching times is loaded into memory for an entire cycle of the input, then no calculations need be performed.

It is inherent in every FCDFC algorithm that the switching sequence be synchronized with the input power source. Using a microcontroller this may be achieved by setting an external pin to generate an interrupt. The interrupt would then cause the program to begin the switching sequence at the appropriate point. A zero crossing detector can be used to generate the interrupt.

The microcontroller will generate control signals to switch directly between the different states. However, we wish to implement a 'make before break' switching strategy. Therefore a delay circuit is needed to provide for this requirement. The delay circuit will ensure that the load is never open circuited. A block diagram of the describing the entire system including microcontroller, zero crossing detector and delay circuitry is shown in Figure 3.7. A description of the experimental setup used in this thesis is contained in Chapter 5. A description and schematics of the circuits used for the zero crossing detector and delay circuitry may be found there.

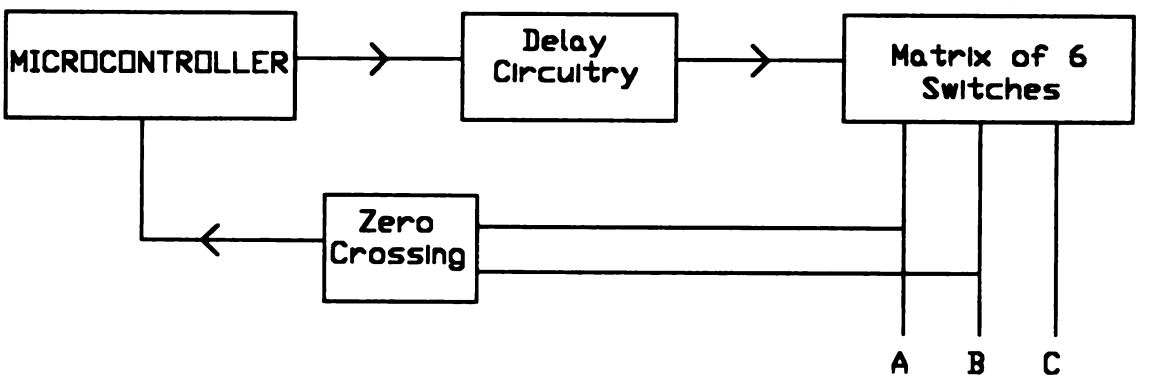


Figure 3.7 - Experimental FCDFC Block Diagram

Chapter 4

COMPUTER SIMULATIONS

The previous chapters covered theoretical aspects of FCDFC algorithms and several circuit topologies by which the converter could be implemented. To test the operation of the converter a given topology must be implemented and an algorithm chosen. This chapter deals with the simulation of the converter using PSpice. The seminatural commutation scheme of Chapter 3 will be adopted and the low frequency modulation method chosen as the algorithm. First however, a discussion of a single output converter using IGBTs is presented. The impetus for this discussion is use of IGBTs in the experimental implentation of the converter in Chapter 5.

4.1 SPICE MODEL FOR THE IGBT

The Insulated Gate Bipolar Transistor (IGBT) has a structure which combines the favorable input drive and switching characteristics of the MOSFET and the low on state voltage of the BJT. The symbol for the IGBT is shown below in Figure 4.1 along with a model for the device [8]. Using PSpice, the model shown has been adapted suitably well to describe the characteristic curves for Toshiba MG75J1BS11 device used in the experimental work of Chapter 5. The model has been applied in the single output seminatural commutation topology shown in Figure 4.2 below.

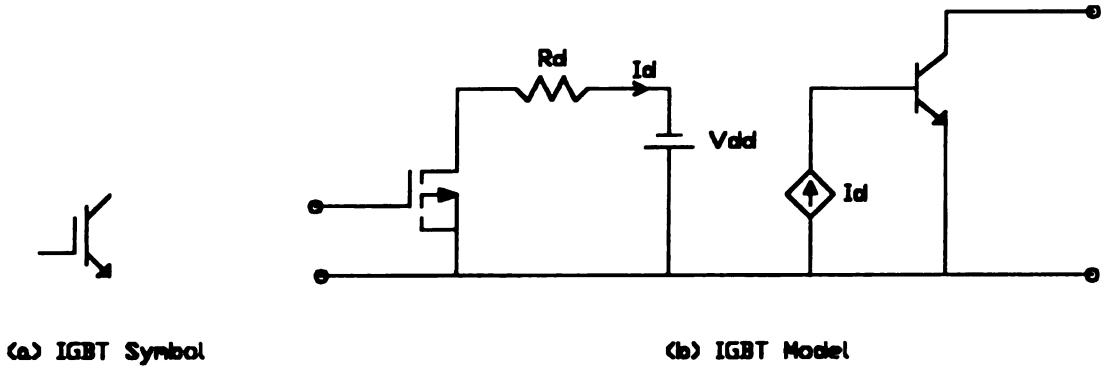


Figure 4.1 - IGBT Symbol and Model

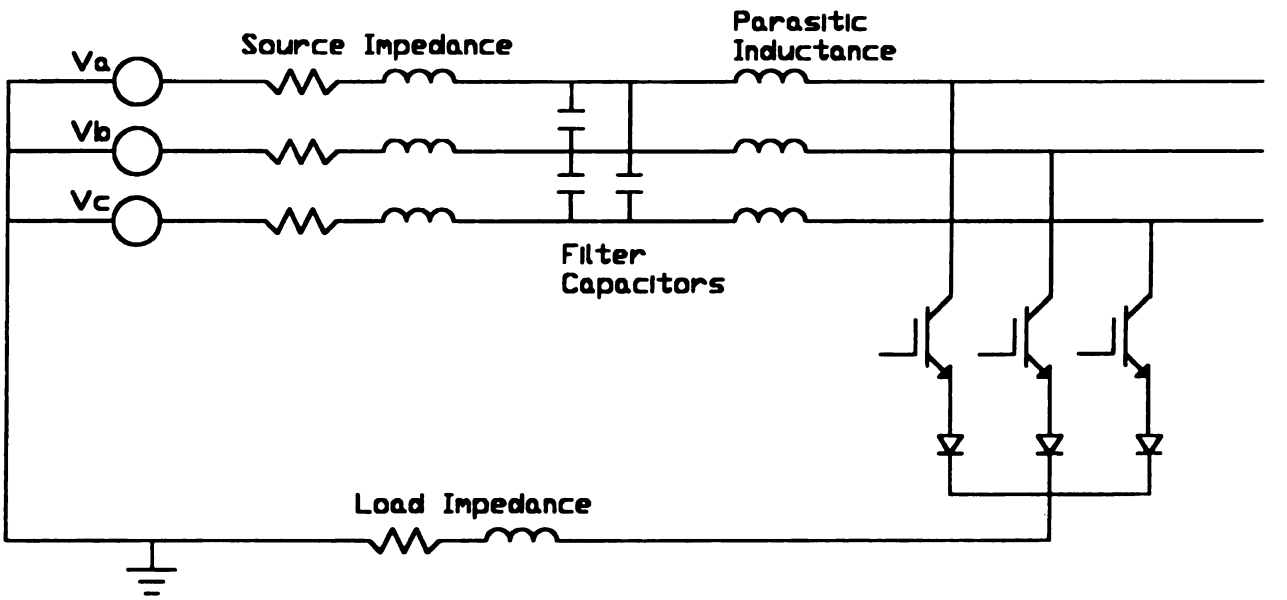


Figure 4.2 - Seminatural Commutation Single Output Topology

The single output topology was used to test the IGBT model in the seminatural commutation scheme due to convergence problems encountered in the two output scheme. Although it will not simulate the actual converter, the single output case will simulate the operation of the IGBT in the circuit topology. The main reason for simulating the actual device used is to test for voltage spikes caused by the commutation. Figure 4.3 shows the output of the simulation for the single output converter connected to a 208 VAC source. A parasitic inductance of 0.15μH was used. The collector-emitter voltage was found displayed spikes exceeding the maximum phase-to-phase voltage by approximately 100V. These spikes however do not coincide with overvoltages due to commutation since they occur when the IGBT is turned on. The lower graph in Figure 4.3 shows the current ramping due to the application of the DC component created by the converter. This behavior in the simulation agrees well with the theory.

Throughout this thesis, commutation of switches in PSpice was achieved by the piecewise linear model (PWL). Inputs to the model were generated by a C program which calculated the PWL corners in terms of the FCDFC algorithm and output these to a text file for each switch. The text files were then pasted into a PSpice circuit file. Although tedious, the process yielded useful results.

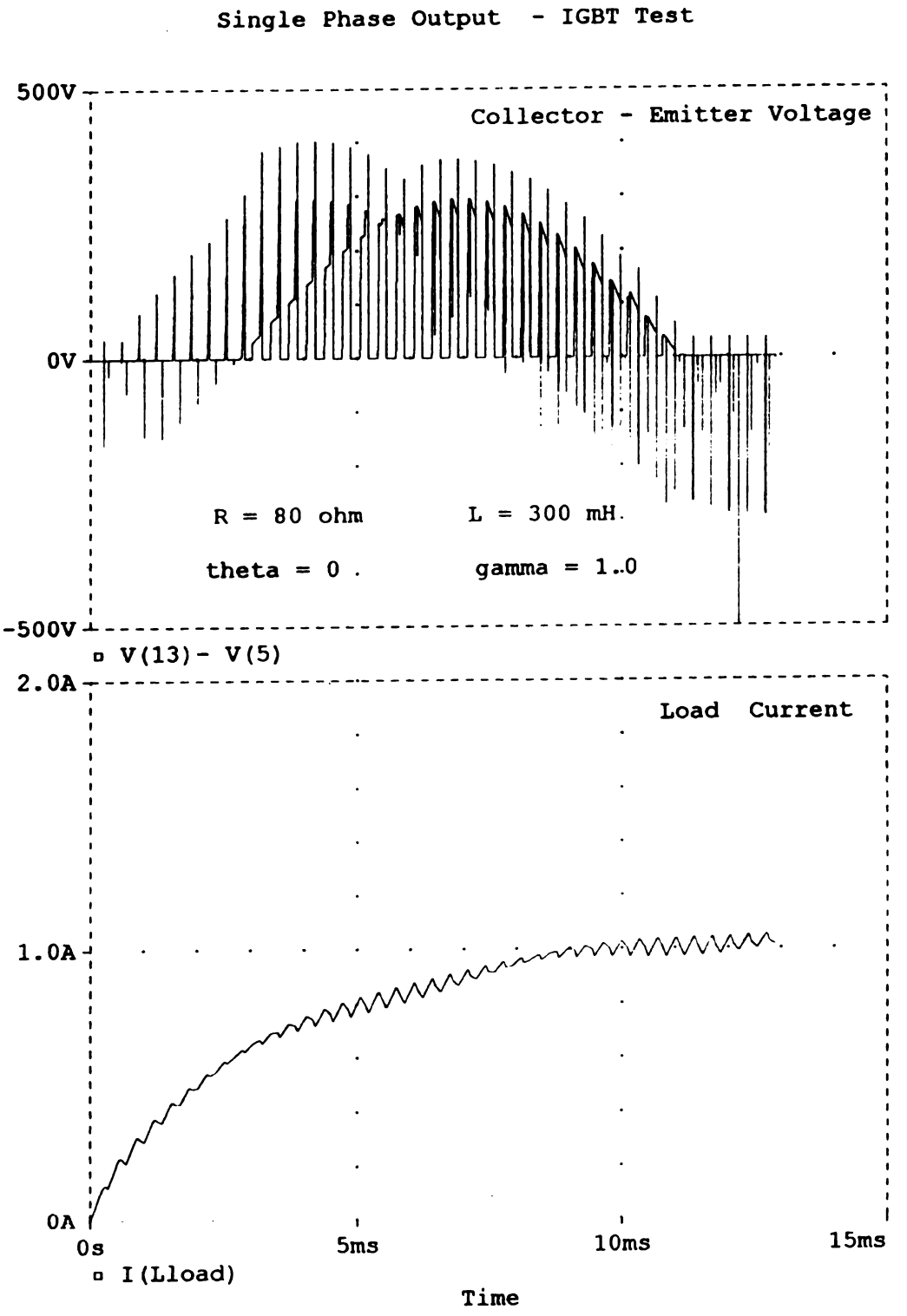


Figure 4.3 - Single Output Converter IGBT Simulation

4.2 SPICE CIRCUIT FOR THE AC-DC CONVERTER

Observing the IGBT model of Figure 4.1 it can be seen that the output characteristics of the device are simulated by the BJT. Since modeling of the drive circuit is not required, the MOSFET and sources may be viewed as part of the drive circuitry. Therefore, in the simulation of the two output direct DC converter BJTs were used in place of IGBTs. Care was taken to ensure that rise and fall times of the circuit were within the limits for the physical IGBTs to be used in the converter. This allowed the overall operation of the converter to be simulated. The circuit used to simulate the two output converter using BJTs is shown below. The circuit topology is seen to be that of the seminatural commutation scheme.

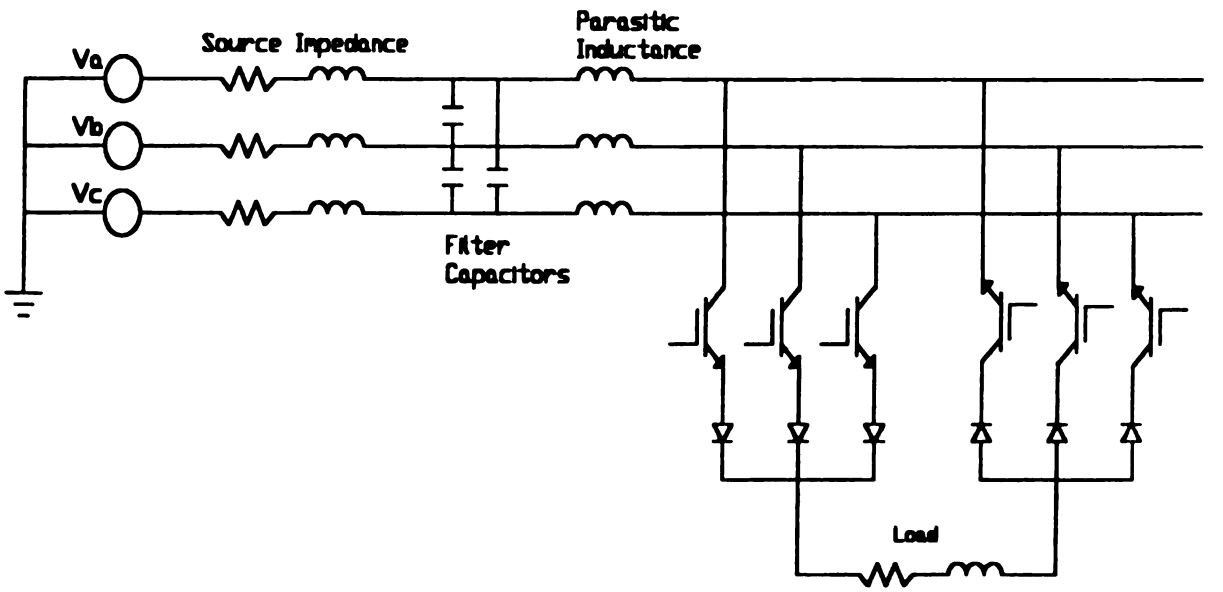


Figure 4.4 - Two Output Seminatural Converter Topology

4.3 OUTPUT WAVEFORMS

The circuit of Figure 4.4 was entered into a PSpice circuit file. Several simulations were obtained using different sets of PWL corner data to control the commutation. A load inductance of 25 mH, load resistance of 6.8 Ω and 12VAC phase to neutral source were used to match the experimental converter. Each simulation began with zero initial load current. Therefore the current in the load should increase as an LR series circuit with a DC voltage applied. This behavior was confirmed in each simulation.

Figure 4.5 is a simulation of the converter using the phase to neutral algorithm discussed in Chapter 2. The upper and lower graphs show the voltage across the load resistor for switching frequencies of 2kHz and 5kHz respectively. Cyclic fluctuations in the harmonics can be seen due to the independent commutation of the positive and negative output terminals. Output voltages obtained agree with the theory when the collector-emitter saturation voltages of the two series transistors are taken into account. Figure 4.6 shows an identical simulation for the phase to phase algorithm. No fluctuation in the harmonics is evident. The output voltage obtained is approximately a factor of $\sqrt{3}/2$ less than the phase to neutral algorithm as predicted.

Figure 4.7 demonstrates the regenerative capability of the FCDFC using the phase to phase algorithm. In each graph of the figure, the voltage is turned on to the positive maximum and then switched to a lower voltage. For a γ of zero, the DC component of the output voltage should also be zero. Therefore the decay of the load resistor voltage in

the upper graph of Figure 4.7 should match the time constant of the load. This agrees well with the simulation. For the middle graph, the voltage is switched to -0.5 of the maximum. The rate of decay of the load resistor voltage is seen to exceed that of the natural time constant. This indicates that power is being transferred from the load to the source. The lower graph displays the decay of the load resistor voltage when the output voltage is changed to a negative maximum. This corresponds to the maximum rate that energy may be removed from the load.

The simulations which are presented here will be compared with the results of the experimental converter presented in the next chapter. These simulations have verified the correct operation of both the phase to phase and phase switching algorithm and the seminatural commutation circuit topology.

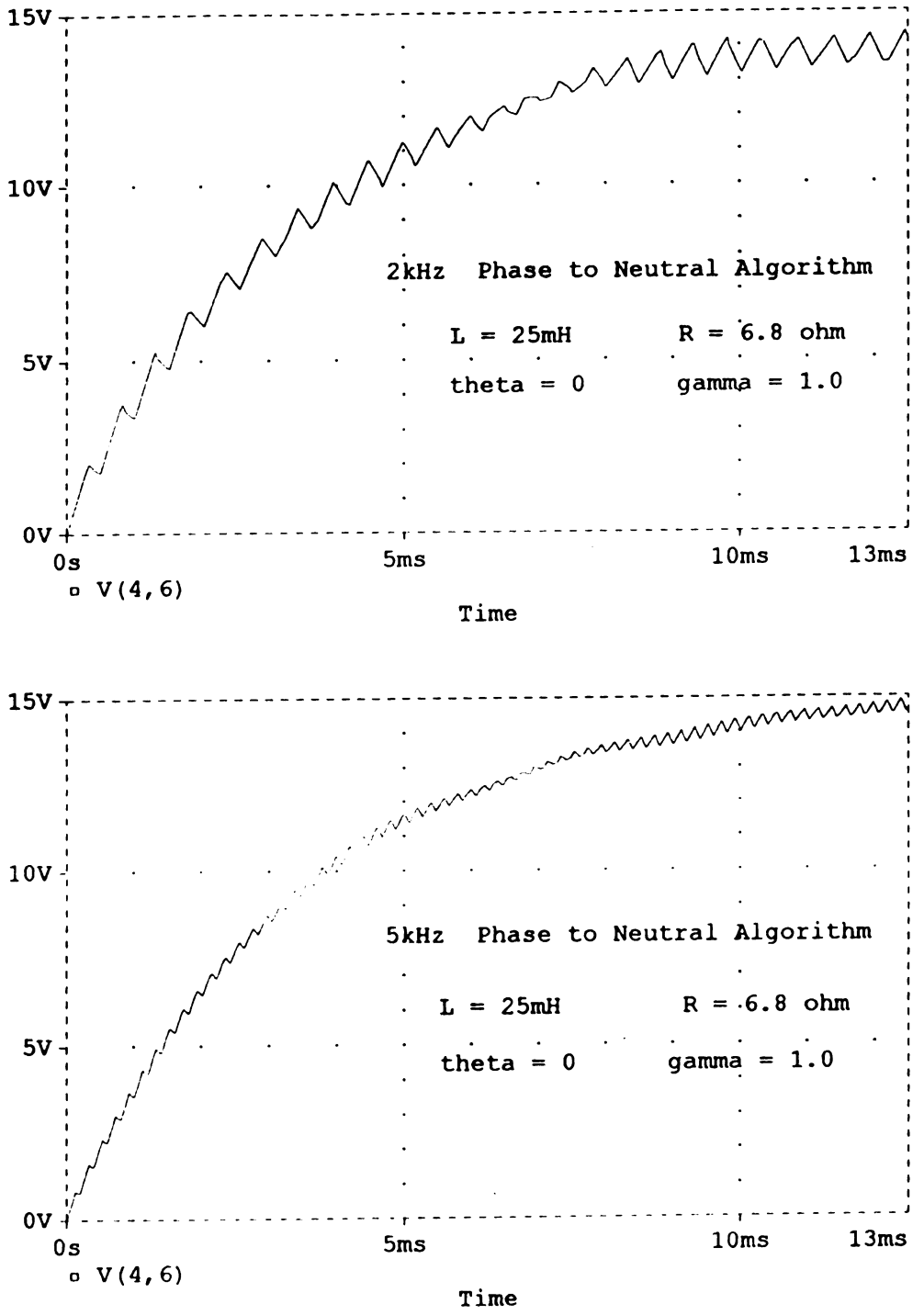


Figure 4.5 - Phase to Neutral Algorithm Simulation

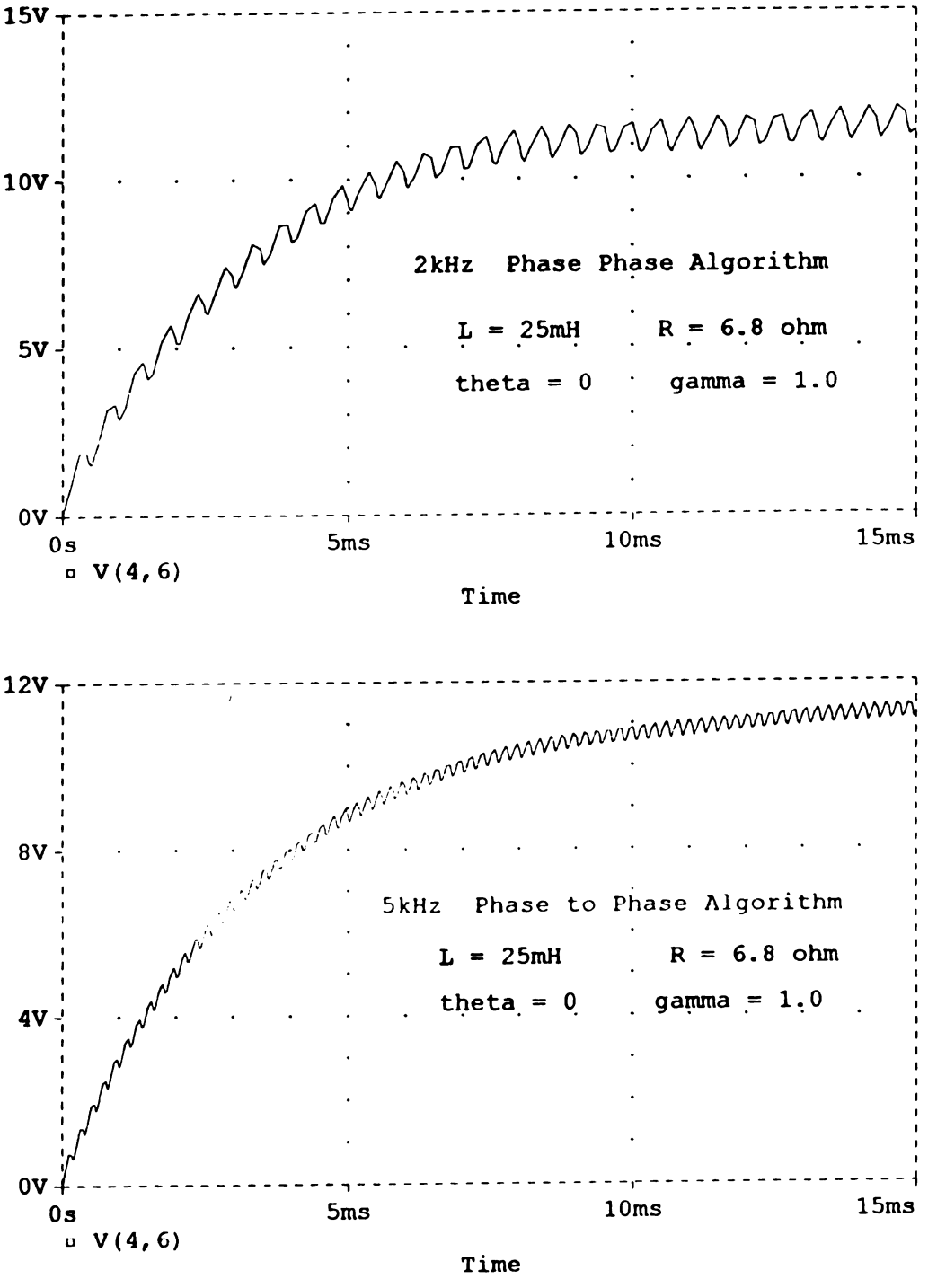


Figure 4.6 - Phase to Phase Algorithm Simulation

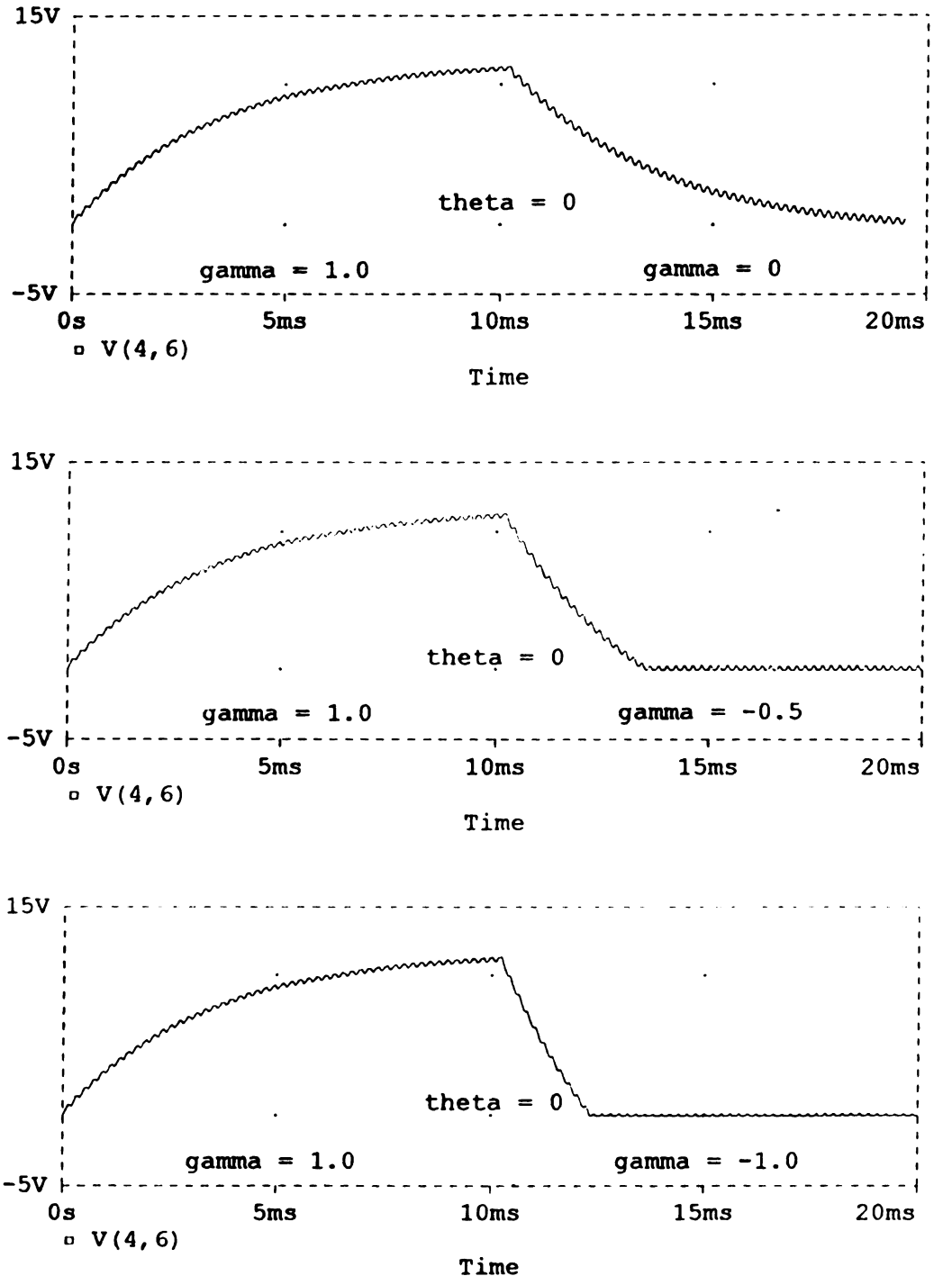


Figure 4.7 - Demonstration of the Regenerative Capability of the FCDFC

Chapter 5

EXPERIMENTAL SETUP AND TESTING

Operation of a direct AC-DC converter was verified in the last chapter using PSpice. The circuit topology used in the simulation was that of the seminatural commutation scheme of Beasant [4] with the phase to phase switching algorithm developed in Chapter 2. An experimental converter was developed based on the same circuit toplogy and switching scheme. The following pages detail the design and testing of the experimental converter.

5.1 DESCRIPTION OF HARDWARE DESIGN

The seminatural commutation scheme requires six transistors and diodes to implement the desired 2-quadrant supply with two outputs. Transistors used were 75A, 600V Toshiba MG75J1BS11 IGBTs. Diodes used were stud mount 40A, 500ns t_{rr}, 600V International Rectifier 40HFL60S05 and 40HFLR60S05. The two different part numbers correspond to cathode to stud and anode to stud configurations respectively. The ratings of these devices far exceed the requirements of testing, however the original scope of the project was to develop a prototype 20A, 208VAC converter for testing at the National Superconducting Cyclotron facility at Michigan State University. Although only low power testing was performed, the converter developed should be capable of operating with a 208 VAC 3 phase input and 20 amperes DC output.

For low power testing of the inductor, a source was provided by three wye connected 120/12V , 2A voltage transformers. The largest inductor available for testing was 25 mH. This inductor was used in series with a 6.8 Ω resistor. Input filter capacitors were provided by three 20 μF metal can capacitors. Snubbing of the input was provided by two 1 μF low inductance capacitors donated by the Illinois Capacitor Company. To avoid the use of snubber capacitors across each IGBT, low inductance bus work was required. A Proto Bus™ kit purchased from Eldre Corporation was used in the design of the bus. The kit includes tinned copper bus bar with equally spaced connection tabs, adhesive insulting tape and plastic connectors. Figure 5.1 below shows the design of the low inductance bus including connections of diodes, IGBTs and input snubbing capacitors.

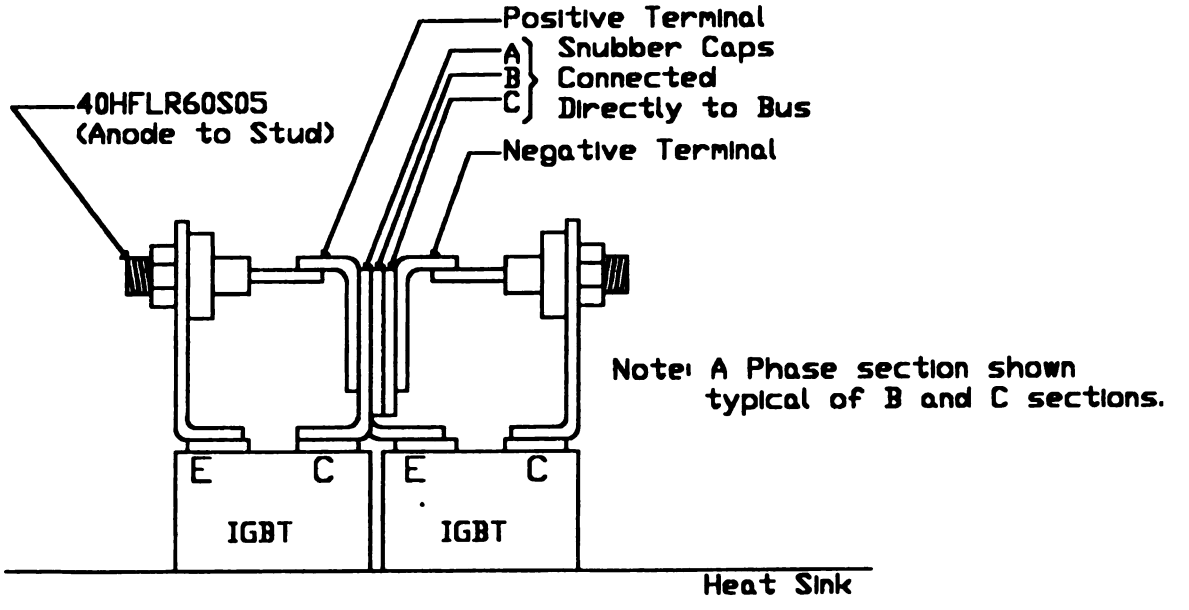


Figure 5.1 - Low Inductance Bus

5.2 MICROCONTROLLER OPERATION AND PROGRAMMING

Operation of the converter was controlled by a Little Giant™ microcontroller manufactured by Z-World Systems. The microcontroller operates at 9.216 MHz using a Z180 processor. A 16 bit parallel port on the controller may be addressed in two 8 bit sections. Each pin on a given port may configured for input or output. A pin configured for input may be set to generate a vectored interrupt. Control of switches was provided by six pins of the parallel port configured as outputs in the upper 8 bit division. A single bit of the lower 8 bit division was configured as input to generate an interrupt used in synchronizing the switching algorithm with the input voltage source. A simple op-amp voltage zero crossing detector was used to provide synchronization.

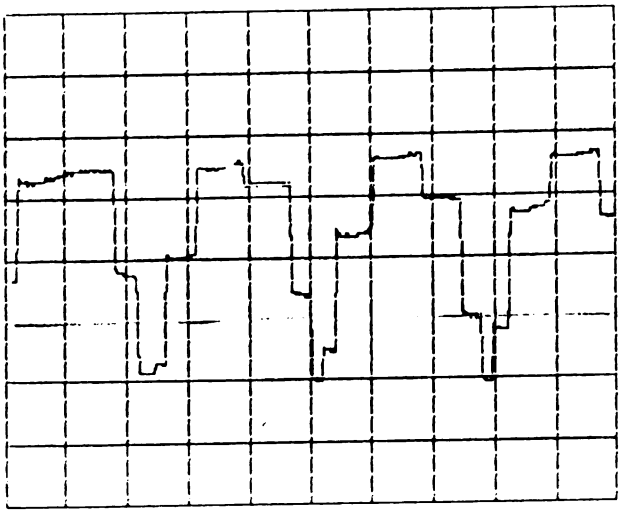
The phase to phase switching algorithm was implemented in the microcontroller program by a lookup table providing timer constants and switching states. A switching state is retrieved from the table and written to the parallel port while the corresponding time is loaded to the timer. An interrupt is generated upon time out causing the next switching state in the table to be written and the next timer value loaded. The sequence repeats until a synchronizing interrupt is generated causing the process to reset to the top of the table. The table is intentionally made longer than one power system cycle so that a synchronizing interrupt will always occur before the end of the table is reached. Values for the table are generated by the program at start up using C functions provided by the compiler. The switching state sequence and synchronizing interrupts were all written in

assembler to make the program faster. A minimum switching state pulsewidth of approximately 15 μs has been achieved. This minimum pulsewidth could cause deviations from the results expected from the algorithm. More will be said of this problem when the output oscilloscope traces are reviewed. The actual microcontroller program may be found in Appendix A.

5.3 RESULTS OF TESTING

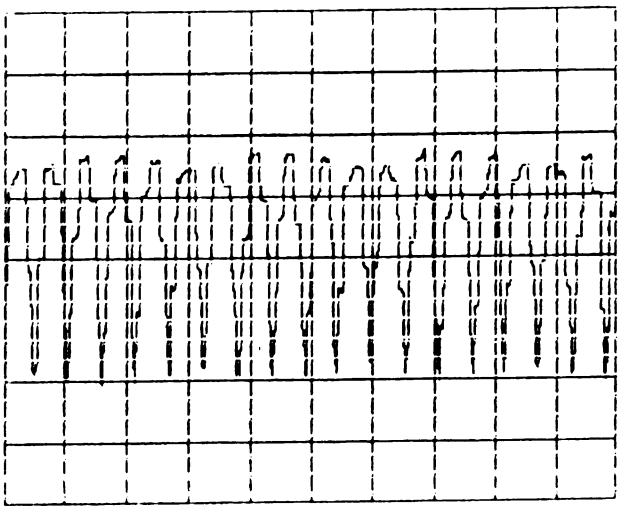
The converter developed was operated applying various parameters in the phase to phase algorithm. Output was observed and plotted using a digital oscilloscope. The total output voltage of the converter is shown in Figure 5.2(a) and (b) for the parameters shown. The waveforms agree well with the expected operation of the converter. Current through the load was monitored by observing the voltage across the series load resistor as was done in simulation. In all cases the DC component observed was approximately 2/3 of that expected from simulation and theory. Figure 5.3(a) shows the load resistor voltage for a gamma of 1.0 and a unity power factor. The voltage expected from simulation would be approximately 12.5V. An actual load resistor voltage of about 8.5V was observed. Operation of the converter for a variation in power factor angle is shown in Figure 5.4 with identical values. The relative change in the output is as expected from theory. In hindsight, it would have been informative to observe the voltage and current waveforms at the 120V terminal of the source transformers to verify the input power factor. For a final verification of the converter operation, the load resistor voltage was

observed for different values of gamma using identical power factor values. The output agreed well with theory as is shown in Figure 5.5(a) and (b).



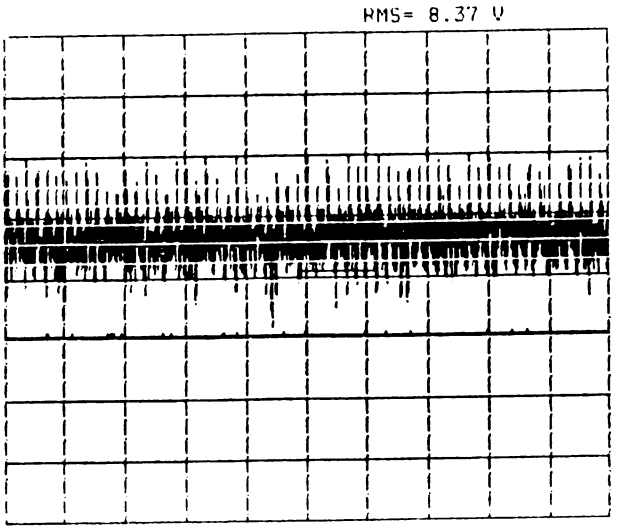
20V/Div 200μs/Div T_S=2000Hz
$\theta = 0$, $\gamma = 1.0$

(a)



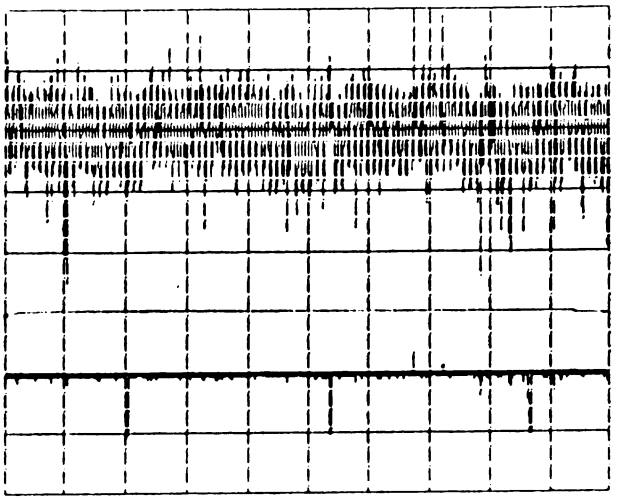
(b)

Figure 5.2 - Converter Output Voltage



5V/Div 50ms/Div T_S=2000Hz
$\theta = 0$, $\gamma = 1.0$

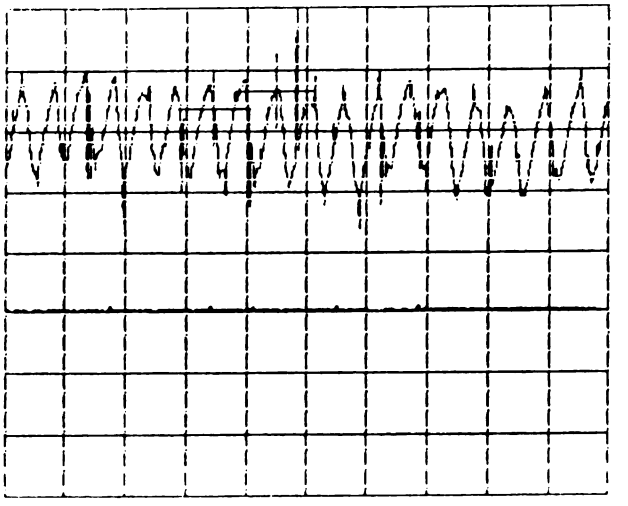
(a)



Top = 2V/Div Bottom 5V/Div 10ms/Div T_S=2000Hz
$\theta = 0$, $\gamma = 1.0$

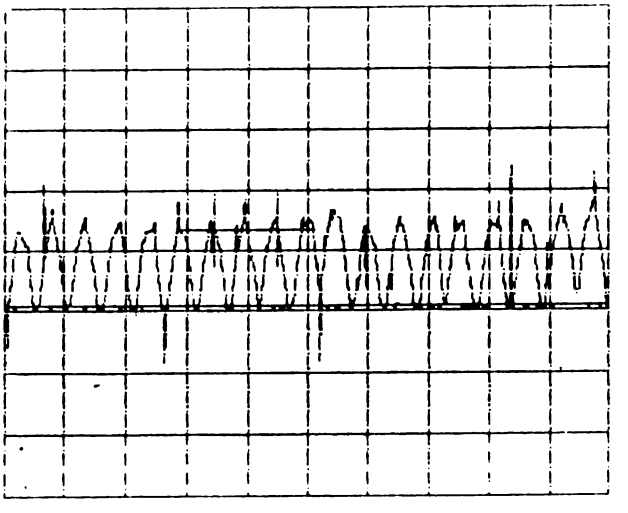
(b)

Figure 5.3 - Load Resistor Voltage



2V/Div 1ms/Div T_S=2000Hz
$\theta = 30$, $\gamma = 1.0$

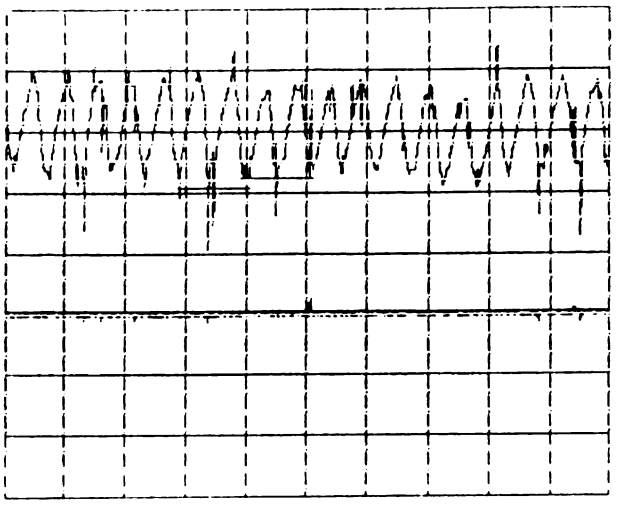
(a)



2V/Div 1ms/Div T_S=2000Hz
$\theta = 60$, $\gamma = 1.0$

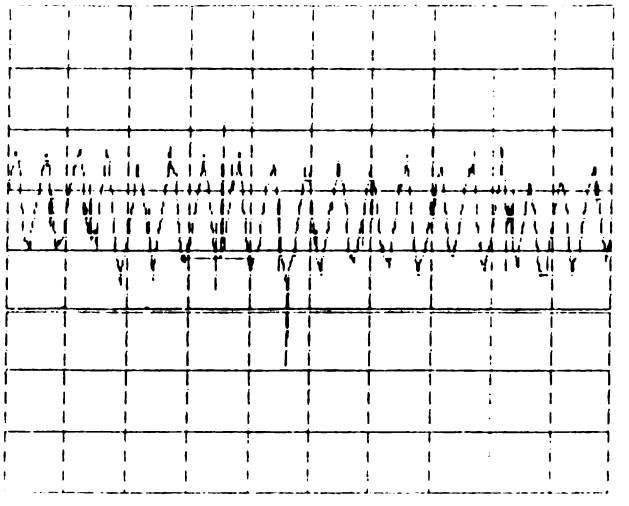
(b)

Figure 5.4 - Load Resistor Voltage Versus θ



2V/Div 1ms/Div T_S=2000Hz
$\theta = 0, \quad \gamma = 0.8$

(a)



20V/Div 1ms/Div T_S=2000Hz
$\theta = 0, \quad \gamma = 0.4$

(b)

Figure 5.5 - Load Resistor Voltage Versus γ

Chapter 6

CONCLUSIONS AND COMMENTS

Various methods of implementing FCDFCs have been presented with regard to the construction of a direct AC-DC converter. In Chapter 2, two different algorithms using the low frequency modulation were presented. The first method used the traditional approach, viewing each output terminal separately in terms of the phase to neutral voltages of the source. The second method derived a new algorithm based only on the voltage between the two output terminals. This new algorithm developed allowed for a simpler microcontroller implementation of the switching states. Adopting the new switching algorithm, a circuit topology was chosen for the development of the prototype converter. The converter design and switching algorithm were then simulated to confirm theoretical expectations. The discussion which follows reviews, compares and contrasts the results obtained from theory, simulation and testing.

6.1 COMPARISON OF SIMULATION AND TESTING

The results of testing deviated from the theory and simulation with regard to both the DC component of the output voltage and the magnitude of the ripple due to switchin harmonics. Both of these comments can be ascertained from the figures of Chapter 4 and Chapter 5. The ripple voltage magnitude depends strongly on the value of load inductance used. The load inductor used was made especially for the project by Osborne. Its value

of inductance was not provided and was determined from measurement using an LR circuit. A value of 25mH was computed from the measured time constant of the circuit for a known resistance value. It is possible that this measurement is a source of error. Another more definite source of error in both switching harmonics and DC level is the minimum commutation time of a switch due to interrupt latencies. The algorithm used will generate arbitrarily small pulsewidths which cannot be achieved with the microcontroller. This will cause the switching period to be longer than expected, decreasing the effective switching frequency and therefore increasing harmonics. The minimum commutation time will also cause phases with polarity opposite that of the desired output to be commutated on for extended periods. The outcome will be a lower DC component of the output. Figure 5.2(a) shows that the actual switching period is approximately 550μs compared with the desired switching period of 500μs.

Except for the problems mentioned above, the converter performed well during low power testing. Variation of the relative voltage with power displacement θ, and voltage utilization factor γ, agreed well with theory. Operation of the converter was verified for changes in these parameters of the algorithm. Performance of the low inductance bus with the seminatural commutation circuit topology proved exceptional. No commutation induced voltage spikes were observed across the collector-emitter terminals of the IGBTs and continuity of the load was maintained.

6.2 SUMMARY AND CONCLUSIONS

The converter developed performed well with the exception of the voltage output magnitude and ripple. These problems may be greatly reduced by modifying the microcontroller program to discard minimum pulsewidths. Switching to a given state would then be skipped if the calculated pulsewidth was less than a predetermined value. Both voltage output magnitude and ripple should improve with this modification. A copy of the microcontroller program used without the modification is provided in Appendix A as a reference.

Components of the prototype converter should allow its use for an input source voltage of 208V and an output current of 20A. High power testing was not performed due to both time constraints and lack of a suitable test load. The converter remains assembled in the MSU Machines Laboratory. High power testing of the device could be performed in the future if a suitable load is obtained.

Another future development which would leverage the work performed in this thesis would be the conversion of the device for four quadrant operation. The seminatural commutation topology used would require that the power circuit hardware be duplicated with the opposite current polarity. The only other component required would be a zero current crossing detector to provide control logic between the positive and negative current polarity devices.

The useful application of the direct AC-DC converter depends upon the load providing filtering of the switching harmonics. Future application of the device as a driver for superconducting magnets used in nuclear experimentation was discussed in the introduction. This would be an ideal application and could be pursued at the National Superconducting Cyclotron facility on the MSU campus.

APPENDIX A

```
/*************************************************************************/
/*                                                                       */
/*                    Program 2PH_DFC.c                                  */
/*                                                                       */
/*                                                                       */
/*************************************************************************/

/************** Function Declarations ****************************/

int prt1_init (int tc);      // initialize prt1
int tab(int a, float A);      // Pulsewidth for state AB
int tac(int b, float B);      // Pulsewidth for state AC
int tbc(int c, float C);              // Pulsewidth for state BC
int tba(int a, float C);              // Pulsewidth for state BA
int tcb(int c, float C);              // Pulsewidth for state CB
int tca(int c, float C);              // Pulsewidth for state CA
void tabulate();          // Table Generator

/********************** Define Global Variables ****************************/

#define TDE1 1            // PRT ch1 down-count enable bit
#define TIE1 5                  // PRT ch1 interrupt enable bit


/************** Variables for Main and Output Functions ************/

static float pi;
static int pwmin;
static float theta;
static float AMPL;    // DC output voltage amplitude
static int offset;          // Column offset to toggle between zero state
static int j;        // Counter timer
static struct tab {       //**** Data structure of PWM times and states
        char state;       // Switching state
  int widthAMPL;     // Switching time for DC voltage
        int widthZero;            // Switching time for Zero voltage
} table[240];        //**** End of data structure
```

```
/*************************************************************************/
/*                    Main Program                    */
/*************************************************************************/

nodebug main(){
        pi=3.1416;
        AMPL=0.99;
         // DC voltage input
        pwmin=10;
         // Minimum pulsewidth
        theta=-45*pi/180;
                        // Input power factor
  tabulate();                   // Generate Switching Table


/***    Set Port B for Output          ***/

  outport(PIOCB,0xcf);              // Set PIO B to mode 3
  outport(PIOCB,0x00);               // Set all bits as output


/***    Set Port A for Interrupt       ***/

  outport(PIOCA,0xcf);              // Set PIO A to mode 3
  outport(PIOCA,0x08);               // Set only bit 4 as input
        outport(PIOCA,PIOA_VEC);          // Set interrupt vector
        outport(PIOCA,0x97);              // Enable interrupt for bit 4 low
        outport(PIOCA,0xf7);              // Mask bit 4 as interrupt

        outport(PIODA,0x08);               // Write data to set bit 4 high



/***    Initialize time step counter    ***/

        j=0;        /*** Note that later j will be loaded to register DE ***/
        offset=1;         /*** Column offset initialized to 1st column       ***/


/***    Initilaize PRT 1                ***/
        prt1_init(10);                  // switch timing interrupts
```

```
/**** Interrupt wait loop  ***/

#asm nodebug
loop:
        nop
        jp c,loop               ; loop and wait for interrupts
#endasm
}


/*************************************************************************/
/*          Define Pulsewidth Interrupt Service Routine          */
/*************************************************************************/

#INT_VEC PRT1_VEC pwm

#asm nodebug
pwm::
        di
        push af             ; save register pair af
        push hl                                 ; save register pair hl
        push de                                         ; save register pair de
        ld      hl,table          ; load PWM switching table address to hl
        ld      de,(j)                          ; load time count to de
        add     hl,de           ; add time count offset in de to hl
        ld      b,(hl)            ; load b with current switching state
        out0  (42H),b           ; write switching state to PIO B
        ld      bc,(offset)         ; load bc with column offset to define voltage
        add   hl,bc           ; add offset to hl
        ld      b,(hl)            ; load b with current switching time
        out0    (14H),b         ; load switching time to PRT1 timer register
        ld      hl,04B0H            ; load hl with table length
        sbc     hl,de           ; subtract current position in table
        jp      nz,loop1            ; if table length<>current position then jump
        ld      de,0000H            ;* else reset current position for next cycle
        jp      loop2             ;* and skip incrementing position in de
loop1:
        inc     de
        inc     de
        inc     de                                      ; increment de by length of data
element
        inc     de
        inc     de
```

```
loop2:
        ld      (j),de                          ; save time counter value
        in0     l,(10H)         ;* read from TCR at 10H and
        in0     l,(14H)         ;* from TMDR1L to enable PWM timer
        pop     de                                      ; restore de
        pop     hl                                      ; restore hl
        pop     af                                      ; restore af
        ei
        reti            ; return from interrupt
#endasm



/*************************************************************************/
/*      Define Synchronization Interrupt Service Routine      */
/*************************************************************************/

#INT_VEC PIOA_VEC sag

#asm
sag::
        di              ; disable interrupts
        push    af
        push    bc                                      ; save registers
        push    de
        push    hl
        ld      de,0000H        ; reset current position for synchronization
        ld      (j),de                          ; save time counter value
        pop     hl
        pop     de
        pop     bc                                      ; restore registers
        pop     af
        ei              ; enable interrupts
        reti            ; return from interrupts
#endasm
```

```
/*********************************************************************/
/*            Define Port 1 Initialization Function        */
/*********************************************************************/

int prt1_init (int tc)
{
        DI();
  IRES (TCR, TDE1);            /* disable count down */
  IRES (TCR, TIE1);           /* disable interrupts */
  outport (TMDR1L, tc);        /* set low byte of data reg */
  outport (TMDR1H, tc >> 8);    /* set high byte of data reg */
  outport (RLDR1H, tc >> 8);    /* set reload counter high byte value */
  outport (RLDR1L, tc);        /* set reload counter low byte value*/
  ISET (TCR, TDE1);            /* enable count down */
  ISET (TCR, TIE1);           /* enable interrupts */
  EI ();                    /* enable interrupts */
}


/*********************************************************************/
/*          Define pulsewidth time functions              */
/*********************************************************************/


/*************************************************/
/* Function:  int ta(int k,float gamma)       */
/* Purpose:  Computes vector A switching time. */
/*************************************************/
int tab(int k,float gamma)
{
 int pw;                // switching pulse width, output variable
 float t;                  // time for calculation
 float Ts;               // switching period

/***  Set Switching Period to 1/80 of a power system cycle.  ***/
  Ts=384;        // Set switching period to (1/60)/(20/9.216MHz)/40=192

  t=k*(Ts/2+10)*20/9216000;                                    // Calculate
time
  pw=floor((1+gamma*cos(2*pi*60*t +theta))*Ts/6);                    // Calculate
switching period for A phase
  if(pw<pwmin) pw=pwmin;
  return(pw);
}
```

```
int tac(int k,float gamma)
{
 int pw;                                                    // switching pulse
width, output variable
 float t;                                                   // time for calculation
 float Ts;                                                  // switching period

/*** Set Switching Period to 1/80 of a power system cycle. ***/
  Ts=384;        // Set switching period to (1/60)/(20/9.216MHz)/80=96

  t=k*(Ts/2+10)*20/9216000;
                 // Calculate time
  pw=floor((1-gamma*cos(2*pi*60*t+4*pi/3 +theta))*Ts/6);        // Calculate switching
period for B phase
  if(pw<pwmin) pw=pwmin;
  return(pw);
}

int tbc(int k,float gamma)
{
 int pw;                                                    // switching pulse
width, output variable
 float t;                                                   // time for calculation
 float Ts;                                                  // switching period

/*** Set Switching Period to 1/80 of a power system cycle. ***/
  Ts=384;        // Set switching period to (1/60)/(20/9.216MHz)/80=96

  t=k*(Ts/2+10)*20/9216000;
                 // Calculate time
  pw=floor((1+gamma*cos(2*pi*60*t+2*pi/3 +theta))*Ts/6);        // Calculate switching
period for B phase
  if(pw<pwmin) pw=pwmin;
  return(pw);
}

int tba(int k,float gamma)
{
 int pw;                                                    // switching pulse
width, output variable
 float t;                                                   // time for calculation
 float Ts;                                                  // switching period
```

```
/*** Set Switching Period to 1/80 of a power system cycle.  ***/
 Ts=384;        // Set switching period to (1/60)/(20/9.216MHz)/80=96

 t=k*(Ts/2+10)*20/9216000;
                // Calculate time
 pw=floor((1-gamma*cos(2*pi*60*t +theta))*Ts/6);        // Calculate switching period
for B phase
 if(pw<pwmin) pw=pwmin;
 return(pw);
}

int tca(int k,float gamma)
{
 int pw;                                                    // switching pulse
width, output variable
 float t;                                          // time for calculation
 float Ts;                                                  // switching period

/*** Set Switching Period to 1/80 of a power system cycle.  ***/
 Ts=384;        // Set switching period to (1/60)/(20/9.216MHz)/80=96

 t=k*(Ts/2+10)*20/9216000;
                // Calculate time
 pw=floor((1+gamma*cos(2*pi*60*t+4*pi/3 +theta))*Ts/6);        // Calculate switching
period for B phase
 if(pw<pwmin) pw=pwmin;
 return(pw);
}

int tcb(int k,float gamma)
{
 int pw;                                                    // switching pulse
width, output variable
 float t;                                          // time for calculation
 float Ts;                                                  // switching period

/*** Set Switching Period to 1/80 of a power system cycle.  ***/
 Ts=384;        // Set switching period to (1/60)/(20/9.216MHz)/80=96

 t=k*(Ts/2+10)*20/9216000;
                // Calculate time
 pw=floor((1-gamma*cos(2*pi*60*t+2*pi/3 +theta))*Ts/6);        // Calculate switching
period for B phase
 if(pw<pwmin) pw=pwmin;
```

```
 return(pw);
}


/*****************************************************************/
/*      Function:  void tabulate()                               */
/*****************************************************************/
void tabulate()
{
 int test;
 static  int i;                                          // time step counter
 static  int Ts;                                         // switching period
 static  char s_table[6];                              // switching state table
          s_table[0]='\xDB';                                    // state AB
    s_table[1]='\xD7';                                  // state AC
    s_table[2]='\xB7';                                  // state BC
                s_table[3]='\xBD';                                      // state
BA
                s_table[4]='\x7D';                                      // state
CA
                s_table[5]='\x7B';                                      // state
CB


/***             Generate the switching table          ***/
 for(i=0; i<240; i=i+6){

        table[i].state=s_table[0];
        table[i+1].state=s_table[1];
        table[i+2].state=s_table[2];


        table[i].widthAMPL=tab(i/3, AMPL);
        table[i].widthZero=tab(i/3, 0);

        table[i+1].widthAMPL=tac(i/3, AMPL);
        table[i+1].widthZero=tac(i/3, 0);

        table[i+2].widthAMPL=tbc(i/3, AMPL);
        table[i+2].widthZero=tbc(i/3, 0);


        table[i+3].state=s_table[3];
        table[i+4].state=s_table[4];
        table[i+5].state=s_table[5];
```

```
        table[i+3].widthAMPL=tba(i/3, AMPL);
        table[i+3].widthZero=tba(i/3, 0);

        table[i+4].widthAMPL=tca(i/3, AMPL);
        table[i+4].widthZero=tca(i/3, 0);

        table[i+5].widthAMPL=tcb(i/3, AMPL);
        table[i+5].widthZero=tcb(i/3, 0);



  }
}

/************************** End of 2PH_2.C ******************************/
```

BIBLIOGRAPHY

[1] A. Alesina and M.Venturini, "Solid-State Power Conversion: A Fourier Analysis Approach to Transformer Synthesis," IEEE Transactions on Circuits and Systems, vol. CAS-28, pp 319-330, April 1981.

[2] G. Roy, L. Duguay, S. Manias and G. E. April, "Asynchronous Operation of a Cycloconverter with Improved Voltage Gain by Employing a Scalar Algorithm," Proc. 87CH2499-2, IEEE - IAS 22nd Annual Meeting, Atlanta, October 18-23, 1987, pp. 891-898.

[3] Y. Kim and M. Eshani, "Control of Force-Commutated Direct Frequency Changers," Proc. 90/CH 2935-5, IEEE pp 1163-1170, May 1990.

[4] R. R. Beasant W. C. Beattie and A. Refsum, "An Approach to the Realization of a High-Power Venturini Converter," Proc. CH2873-8/90/0000-0291 IEEE, pp 291-297.

[5] C. T. Pan, T. C. Chen and J. J. Sheih, "A Zero Loss Matrix Converter," Proc. 0-7803-1243-0/93 IEEE pp 545-550.

[6] P. D. Ziogas, S. I. Khan, M. H. Rashid, "Some Improved Forced Commutated Cycloconverter Structures," IEEE Transactions on Industry Applications, Vol. 1A-21, No. 5, September 1985, pp 1242-1251.

[7] N. Mohan, T. M. Undeland, W. P. Robbins, "Power Electronics," John Wiley and Sons, Inc. , Copyright 1989, 1995.

[8] Muhammad H. Rashid, "SPICE for Power Electronics and Electric Power," Englewood Cliffs, N.J. Prentice Hall, 1993.