# A mathematical model of a ship's electrical power system

# A MATHEMATICAL MODEL

# OF

# A SHIP'S POWER SYSTEM

by

K.K.NG, B.Sc. (Hons)

A Master's Thesis

submitted in partial fulfilment of the requirements

for the award of the degree of

Master of Philosophy in Engineering

of Loughborough University of Technology

June, 1989

Supervisors: Mr. J.G. Kettleborough

Professor I.R. Smith

Department of Electronic and

Electrical Engineering

## Acknowledgements

I wish to express my deepest gratitude to my supervisor, Mr J. G. Kettleborough and Professor I. R. Smith (Director of Research) for their invaluable guidance, advice, encouragement and their patience during the period of research and preparation of this thesis.

Many thanks to Mr K. Gregory and Dr. C. R. Fitton for their advice during the periods of research.

Also, I wish to thanks my parents for their endurance and precious help throughout the period of study.

# Synopsis

The work described in this thesis concerns the time-domain simulation of various items of plant for a limited-size electrical power system. Initially an isolated 3-phase synchronous generator is considered, with the electrical equations being expressed in the phase reference frame, since this copes easily with both unbalanced fault and load switching conditions. The study looks at theoretical results for a 3-phase short circuit test on a generator as provided by the computer model and by a conventional *dqo* approach. In addition, the generator model is used in investigations of various unbalanced load conditions.

The single generator study is then extended to a multi-generator power system, and models for the following items of plant are developed:

a) A 3-phase synchronous generator driven by a diesel engine. The engine is governor speed controlled, and the generator has an automatic voltage regulator (AVR) to maintain a constant generator output voltage.

b) A motor/generator set, comprising a 3-phase synchronous machine, mechanically coupled to a separately-excited dc machine.

c) Section switches and a bus-coupler, which may be switched both in or out during the simulation.

d) A fully controlled 3-phase bridge converter with back-to-back thyristors, which is capable of both rectification and pulse-width-modulated (PWM) inversion.

A method of numerical analysis based on Kron's diakoptic approach is used to investigate the behaviour of the complete system. For the purpose of calculation the system is torn into 5 sub-networks and for each seperate sub-network a set of differential equations is solved. Using numerical data derived from each sub-network, the currents and voltages of the complete system are then obtained using inverse transformations.

Finally, the performance of the system is illustrated by considerations for a variety of balanced and unbalanced switching conditions.

# LIST OF SYMBOLS

In the following list, subscripts i and j equal r, y, b, f, d and q, referring respectively to the red, yellow and blue armature phase windings, the field winding, and the effective direct- and quadrature- axis damper windings of the synchronous generator. Subscript L refers to the load. An additional subscript 0 used with the inductance coefficients implies the value of the coefficient with zero field current.

| | | |
|---|---|---|
| $C, C^t$ | - | branch/mesh current transformation for a synchronous machine and its transpose. |
| $C^{L}_{\cdot m}, C^{\cdot L}_{m}$ | - | link/mesh transformation and its transpose. |
| $E_o$ | - | Synchronous machine open - circuit phase voltage. |
| $G_{ij}$ | - | time rate-of-change of inductances. |
| h | - | integration step length. |
| $\bar{i}_{TO}$ | - | thyristor current at the beginning of an integration step. |
| $i_T$ | - | thyristor current at the end of an integration step. |
| J | - | combined interia of the synchronous machine and dc machine. |
| k | - | DC machine voltage constant. |
| $k_a$ | - | AVR amplifier gain. |
| $k_e$ | - | exciter gain. |
| $k_f$ | - | exciter feedback circuit gain. |
| $k_{ff}$ | - | frictional constant for DC machine. |
| $k_R$ | - | feedback transformer/rectifier gain. |
| $L_{ii}$ | - | self inductance of winding i. |
| $L_i$ | - | $L_{ii} + L_{Li}$. |
| $L_{ad}, L_{aq}$ | - | direct- and quadrature- axis coefficients of armature phase/phase self inductances. |

| | | |
|---|---|---|
| $M_{sf}, M_{fs}$ | - | mutual inductance between series and shunt fields of DC machine and vice versa. |
| $M_{ha}, M_{ah}$ | - | mutual inductance between interpole and armature and vice versa. |
| $M_{ij}$ | - | mutual inductance between windings i and j. |
| $M_{ad}, M_{aq}$ | - | direct- and quadrature-axis coefficients of armature phase/phase mutual inductances. |
| $M_f$ | - | direct-axis coefficient of armature phase/field mutual inductance. |
| $M_d, M_q$ | - | direct- and quadrature- axis coefficients of armature phase/damper mutual inductances. |
| $N_1/N_4$ | - | effective armature phase/field turns ratio. |
| $N_5/N_4$ | - | effective damper/field turns ratio. |
| $N_6/N_1$ | - | q-axis damper/q-axis armature turns ratio. |
| P | - | synchronous machine output power. |
| $p_o$ | - | number of poles on synchronous machine. |
| $R_{ii}$ | - | resistance of winding i. |
| $R_i$ | - | $R_{ii} + R_{Li}$ |
| $R_a, L_a$ | - | DC machine armature resistance and inductance. |
| $R_h, L_h$ | - | DC machine interpole resistance and inductance. |
| $R_y, L_y$ | - | DC machine series field resistance and inductance. |
| $R_f, L_f$ | - | DC machine shunt field resistance and inductance. |
| SF | - | synchronous machine saturation factor. |
| $T_a$ | - | AVR amplifier time constant. |
| $T_b$ | - | exciter time constant. |
| $T_e$ | - | electrical torque produced by synchronous machine. |
| $T_{do}'$ | - | direct-axis transient open-circuit time constant. |

$T_{do}''$ - direct-axis sub-transient open-circuit time constant

$T_d'$ - direct-axis transient short-circuit time constant.

$T_d''$ - direct-axis sub-transient short-circuit time constant.

$T_{kd}$ - direct-axis damper leakage time constant.

$T_q''$ - quadrature-axis sub-transient short-circuit time constant.

$T_{qo}''$ - quadrature-axis sub-transient open-circuit time constant.

$T_{f1}$, $T_{f2}$ - exciter feedback circuit time constant.

$t_d$ - time to current discontinuity in the converter from the beginning of an integration step.

$t_{poi}$ - time to a point of intersection from the start of an integration step.

$V_T$ - AVR feedback voltage.

$\omega$ - angular speed of a dc machine.

$\omega_o$ - synchronous speed.

$X_d$ - direct-axis synchronous reactance.

$X_d'$ - direct-axis transient reactance.

$X_d''$ - direct-axis sub-transient reactance.

$X_{md}$ - direct-axis magnetizing reactance

$X_{kd}$ - direct-axis damper leakage reactance.

$X_q$ - quadrature-axis synchronous reactance.

$X_q'$ - quadrature-axis transient reactance.

$X_q''$ - quadrature-axis sub-transient reactance.

$X_{mq}$ - quadrature-axis magnetizing reactance.

$X_{kq}$ - quadrature-axis leakage reactance.

$X_a$ - armature leakage reactance.

$X_2$ - negative sequence reactance.

$X_z$, $L_z$ - zero-sequence reactance and inductance.

# Table of Contents

## Chapter 1

# INTRODUCTION

Due to the continual expansion of power systems, a more accurate and time-saving means of studying their behaviour is required. In this context, mathematical modelling provides a very effective technique of considerable value to the designer of electrical power systems. It enables the designer to carry out a detailed investigation of the system for both transient and steady state operation and, in addition, it provides a theoretical basis from which the system parameters may be optimised.

In this thesis, a mathematical model is developed for the typical ship's electrical power system shown in Fig 1.1. The modelling is based on Kron's [1] diakoptic approach, in which the system is torn into several sub-networks which are solved as if they existed separately. It enables an efficient solution to be obtained for the numerical integration of the system equations and, although a time-varying inductance matrix has to be inverted at every step of the solution, this presents few problems to a high speed digital computer. The disadvantage of diakoptics is that the numerical solution obtained may be unstable if many torn sub-network are used, but this is not a problem in the present study.

The thesis describes the simulation of the various item of plants of the power system shown in Fig 1.1. Individual models are given for a 3-phase synchronous generator, a motor/generator set, a bus-coupler and a 3-phase thyristor bridge.

## 1.1 Three-phase synchronous generator model

Modelling of a synchronous generator in either $dqo$ or $\alpha\beta o$ co-ordinates involves considerable approximation. In the *dqo* approach, the behaviour of the machine is considered along both its direct and quadrature axes, and the employment of various tensor transformations enables the time-varying coefficients present in the basic equations to be eliminated, so allowing an analytical solution. However, the analytical solution is limited to only a certain range of problems and if the resulting model is used to investigate an unbalanced loading conditions, inaccurate results may be obtained [2]. In general, it is preferrable to use a model based upon the phase reference frame, which is both more flexible and allows saturation of the generator to be included. In this case, the only approximation involved is that saturation arises principally from the effective direct-axis current. In chapter 2, a synchronous generator model is developed, with the equation expressed in the phase reference frame. The performance of the generator following a sudden short-circuit is investigated, and a comparison is made with theoretical predictions obtained using a *dqo* approach. Various unbalanced fault and load switching conditions are also simulated.

## 1.2 Motor/generator set model

In chapter 3, a mathematical model is presented for a motor/generator set comprising a 3-phase synchronous machine directly coupled to a separately-excited DC machine. Using electrical and mechanical equations developed for the DC machine, together with the model of the synchronous generator from chapter 2, the computer simulation is tested for various load rejection conditions.

## 1.3 Bus-coupler model

The section switches in Fig 1.1 (SW1 to SW5) have to be switched in and out during the simulation and this may be modelled using constant flux-linkage considerations.

However, the process involves a considerable number of computing statements and a simpler method is to change instantaneously the switch inductances of the bus-coupler. This reduces the number of computer statements and is also closer to the practical situation.

Tensor analysis is introduced to facilitate modelling of the bus-coupler and the technique is described in chapter 4.

## 1.4 The 3-phase thyristor bridge converter model

Chapter 5 describes a mathematical model for a 3-phase bridge converter which is capable of both rectification and inversion in a PWM manner.

Tensor analysis is again used for the converter model, with the program being developed to handle the changing thyristor conduction pattern. Results of simulations of various voltages and currents waveforms are presented for different trigger angles in the rectification mode. In the inversion mode, the converter response to various modulating frequencies is simulated and typical results obtained are presented.

## 1.5 The complete ship's power system model

In chapter 6, the complete model for the ships power system of Fig 1.1 is developed using a diakoptic approach, and in chapter 7 simulated results of the voltage and current waveforms at various points of the system are used to illustrate the system behaviour for a number of different switching conditions. The computer program is written in Fortran 77 and runs on a Honeywell Multics computer.

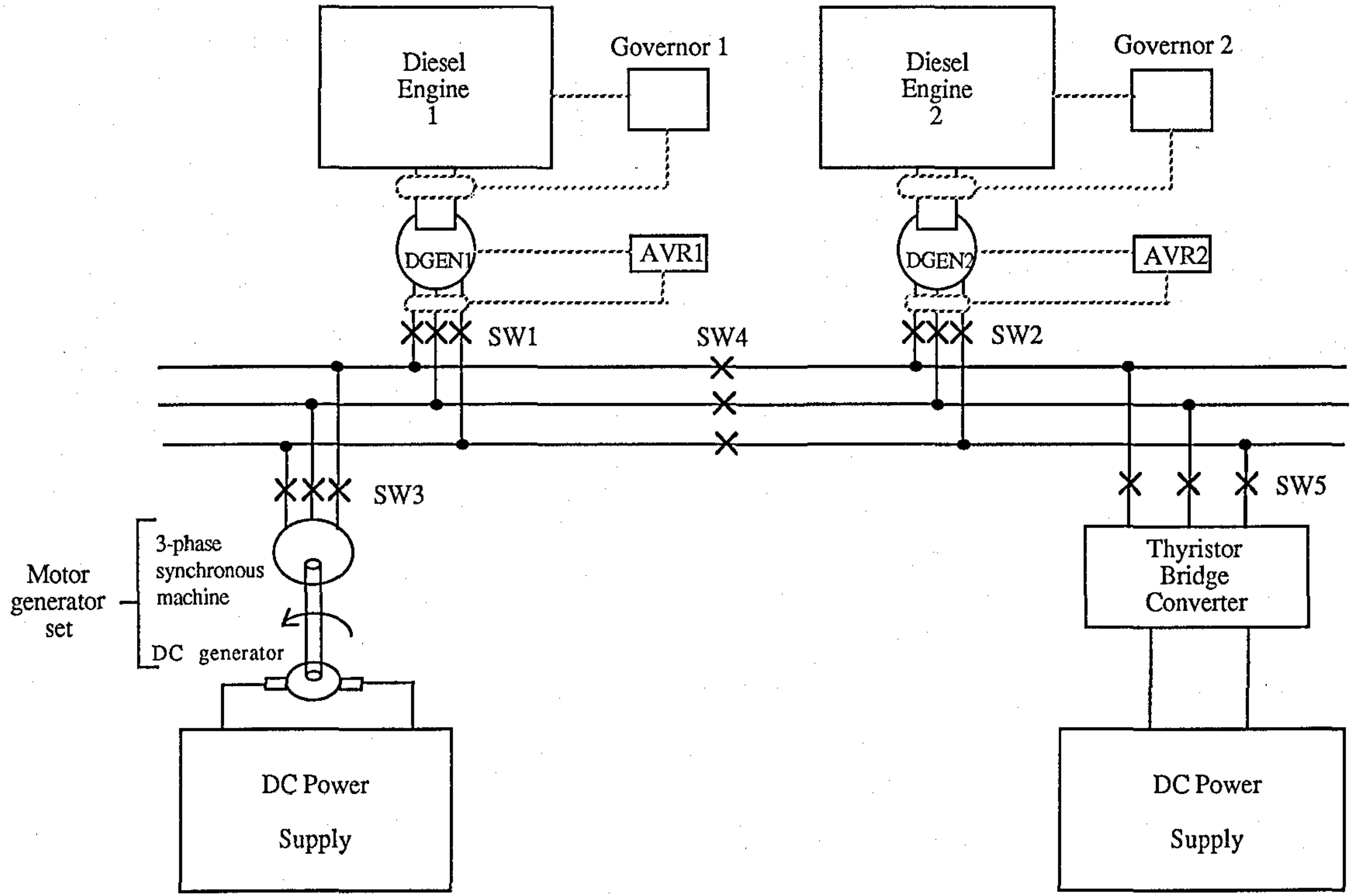



**Fig 1.1 A typical ship's electrical power system**

# Chapter 2

## SIMULATION OF A 3-PHASE SYNCHRONOUS GENERATOR

This chapter presents a mathematical model for an isolated diesel driven 3-phase synchronous generator, with the output voltage controlled by an automatic voltage regulator *(AVR)*. The model is based on the phase reference frame for the machine [3], and a set of linear differential equations with variable coefficients is presented which describes the machine behaviour under both steady state and transient conditions.

## 2.1 The generator model

The phase reference frame is used to define the generator model, since it copes easily with both unbalanced fault and switching conditions.In addition, any higher order harmonics present in the airgap mmf of the machine may easily be included [3]. The early disadvantage of the phase reference frame representation was that the inversion of a time-varying inductance matrix with a rank of five, which is needed at each step of the numerical solution, could introduce long computer run-times. However, following the development of modern high-speed computer, this does not now present any significant problems.

In this thesis, a saturation factor *(SF)* accounts for magnetic saturation of the inductance coefficients associated with the short direct-axis airgap. The quadrature-axis coefficients are associated with the long quadrature-axis airgap, which is naturally much less sensitive to saturation.

### 2.1.1 Electrical Equations

Fig 2.1 is a circuit representation of a 3-phase synchronous generator with damping on both the direct and quadrature axes. The armature windings r, y, b are carried on the rotor of the machine, with the field winding f and the effective direct and quadrature axes damper winding (d and q respectively) being on the stator. The corresponding circuit equations are [3]

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ V_f \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} R_r & 0 & 0 & 0 & 0 & 0 \\ 0 & R_y & 0 & 0 & 0 & 0 \\ 0 & 0 & R_b & 0 & 0 & 0 \\ 0 & 0 & 0 & R_{ff} & 0 & 0 \\ 0 & 0 & 0 & 0 & R_{dd} & 0 \\ 0 & 0 & 0 & 0 & 0 & R_{qq} \end{bmatrix} \begin{bmatrix} i_r \\ i_y \\ i_b \\ i_f \\ i_d \\ i_q \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} L_r & M_{ry} & M_{rb} & M_{rf} & M_{rd} & M_{rq} \\ M_{yr} & L_y & M_{yb} & M_{yf} & M_{yd} & M_{yq} \\ M_{br} & M_{by} & L_b & M_{bf} & M_{bd} & M_{bq} \\ M_{fr} & M_{fy} & M_{fb} & L_{ff} & M_{fd} & M_{fq} \\ M_{dr} & M_{dy} & M_{db} & M_{df} & L_{dd} & M_{dq} \\ M_{qr} & M_{qy} & M_{qb} & M_{qf} & M_{qd} & L_{qq} \end{bmatrix} \begin{bmatrix} i_r \\ i_y \\ i_b \\ i_f \\ i_d \\ i_q \end{bmatrix} \qquad (2.1)$$

or, in abbreviated form,

$$\left[V_b\right] = \left[R_{bb}\right]\left[I_b\right] + \frac{d}{dt}\left[L_{bb}\, I_b\right] \qquad (2.2)$$

For a 3-phase 3-wire connection to the armature, it follows that

$$i_r + i_y + i_b = 0 \qquad (2.3)$$

which allows the rank of the matrices in equation (2.1) to be reduced by one, by use of a transformation matrix defined by,

$$\begin{bmatrix} i_r \\ i_y \\ i_b \\ i_f \\ i_d \\ i_q \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_r \\ i_y \\ i_f \\ i_d \\ i_q \end{bmatrix} \tag{2.4}$$

or , in abbreviated form,

$$I_b = C\, I_m \tag{2.5}$$

where C is the branch/mesh transformation of the synchronous machine,

$I_b = \begin{bmatrix} i_r & i_y & i_b & i_f & i_d & i_q \end{bmatrix}^t$

and $I_m = \begin{bmatrix} i_r & i_y & i_f & i_d & i_q \end{bmatrix}^t$

Assuming power invariance between the branch and mesh reference frames [4], the mesh and branch voltages of the generator are related by

$$V_m = C^t\, V_b \tag{2.6}$$

where $C^t$ is the transpose of C,

$V_b$ is the vector of generator branch voltages,

and $V_m$ is the vector of generator mesh voltages.

Substituting equation (2.5) into equation (2.2) yields

$$\left[V_b\right] = \left[R_{bb}\right]\left[C\,I_m\right] + \frac{d}{dt}\left[L_{bb}\,[C\,I_m]\right] \qquad (2.7)$$

and combining equations (2.6) and (2.7)

$$V_m = \left[[C^t]\,[R_{bb}]\,[C]\,[I_m]\right] + \frac{d}{dt}\left[[C^t]\,[L_{bb}]\,[C]\,[I_m]\right] \qquad (2.8)$$

Equation (2.8) may be expressed in the abbreviated form,

$$V_m = R_{mm} I_m + \frac{d}{dt}\left[L_{mm} I_m\right] \qquad (2.9)$$

where $R_{mm} = C^t R_{bb} C$

and $L_{mm} = C^t L_{bb} C$

Equation (2.9) may be defined in full as,

$$\begin{bmatrix} 0 \\ 0 \\ V_f \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} R_r + R_b & R_b & 0 & 0 & 0 \\ & R_y + R_b. & 0 & 0 & 0 \\ \text{SYMMETRICAL} & & R_{ff} & 0 & 0 \\ \text{ABOUT} & & & R_{dd} & 0 \\ \text{LEADING DIAGONAL} & & & & 0 \end{bmatrix} \begin{bmatrix} i_r \\ i_y \\ i_f \\ i_d \\ i_q \end{bmatrix} +$$

$$\frac{d}{dt}\begin{bmatrix} L_r + L_b - 2M_{br} & L_b - (M_{br} + M_{yb} - M_{ry}) & M_{rf} - M_{bf} & M_{rd} - M_{bd} & M_{rq} - M_{bq} \\ & L_y + L_b - 2M_{yb} & M_{yf} - M_{bf} & M_{yd} - M_{bd} & M_{yq} - M_{bq} \\ & & L_{ff} & M_{fd} & 0 \\ \text{SYMMETRICAL ABOUT} & \text{LEADING DIAGONAL} & & L_{dd} & 0 \\ & & & & L_{qq} \end{bmatrix} \begin{bmatrix} i_r \\ i_y \\ i_f \\ i_d \\ i_q \end{bmatrix}$$

(2.10)

*If equation (2.10) is presented in the abbreviated form*

$$\left[V_m\right] = \left[R_{mm}\right]\left[I_m\right] + \left[\frac{dL_{mm}}{dt}\right]\left[I_m\right] + \left[L_{mm}\right]\left[\frac{dI_m}{dt}\right] \qquad (2.11)$$

it may be re-arranged in the form suitable for numerical integration as

$$\left[\frac{dI_m}{dt}\right] = \left[L_{mm}\right]^{-1}\left[V_m - \left(R_{mm} + \frac{dL_{mm}}{dt}\right) I_m\right] \qquad (2.12)$$

From equation (2.5), it is seen that,

$$I_b = C\,I_m$$

or

$$\frac{dI_b}{dt} = C\frac{dI_m}{dt}$$

The time rate-of-change of the branch current vector is therefore

$$\begin{bmatrix} \frac{di_r}{dt} \\ \frac{di_y}{dt} \\ \frac{di_b}{dt} \\ \frac{di_f}{dt} \\ \frac{di_d}{dt} \\ \frac{di_q}{dt} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{di_r}{dt} \\ \frac{di_y}{dt} \\ \frac{di_f}{dt} \\ \frac{di_d}{dt} \\ \frac{di_q}{dt} \end{bmatrix} \qquad (2.13)$$

and the terminal voltage is [5]

$$\begin{bmatrix} V_r \\ V_y \\ V_b \\ V_f \end{bmatrix} = \begin{bmatrix} R_{rr}+G_{rr} & G_{rb} & G_{ry} & G_{rf} & G_{rd} & G_{rq} \\ G_{yr} & R_{yy}+G_{yy} & G_{yb} & G_{yf} & G_{yd} & G_{yq} \\ G_{br} & G_{by} & R_{bb}+G_{bb} & G_{bf} & G_{bd} & G_{bq} \\ G_{fr} & G_{fy} & G_{fy} & R_{ff}+G_{ff} & G_{fd} & 0 \end{bmatrix} \begin{bmatrix} i_r \\ i_y \\ i_b \\ i_f \\ i_d \\ i_q \end{bmatrix} +$$

$$\begin{bmatrix} L_{rr} & M_{ry} & M_{rb} & M_{rf} & M_{rd} & M_{rq} \\ M_{yr} & L_{yy} & M_{yb} & M_{yf} & M_{yd} & M_{yq} \\ M_{br} & M_{by} & L_{bb} & M_{bf} & M_{bd} & M_{bq} \\ M_{fr} & M_{fy} & M_{fb} & L_{ff} & M_{fd} & 0 \end{bmatrix} \begin{bmatrix} \frac{di_r}{dt} \\ \frac{di_y}{dt} \\ \frac{di_b}{dt} \\ \frac{di_f}{dt} \\ \frac{di_d}{dt} \\ \frac{di_q}{dt} \end{bmatrix} \qquad (2.14)$$

where $M_{ry} = M_{yr}$, $G_{ry} = G_{yr}$, etc, and the G terms are time rate-of-change of inductances.

The power produced by the synchronous generator per phase is

$$P = V_m I_m \qquad (2.15)$$

or, after substitution from equation (2.11)

$$P = R I_m^2 + L_{mm} I_m \frac{dI_m}{dt} + I_m^2 \frac{dL_{mm}}{dt} \tag{2.16}$$

In the above equation, the first term is the Ohmic copper loss and the second is the rate-of-change of stored magnetic energy within the machine. The third term alone is the actual mechanical power associated with the generator itself. The torque at the machine shaft is given by the equation

$$T_e = \frac{P}{\omega}$$

from which it follows that,

$$\begin{aligned} T_e &= \frac{1}{\omega} I_m^2 \frac{dL_{mm}}{dt} \\ &= p_0 I_m^2 \frac{dL_{mm}}{d\theta} \end{aligned} \tag{2.17}$$

and therefore, that

$$T_e = p_0 \begin{bmatrix} i_r \\ i_y \\ i_f \\ i_d \\ i_q \end{bmatrix}^t \begin{bmatrix} G_{rr}+G_{bb}-2G_{br} & G_{bb}-(G_{br}+G_{yb}-G_{ry}) & G_{rf}-G_{bf} & G_{rd}-G_{bd} & G_{rq}-G_{bq} \\ & G_{yy}+G_{bb}-2G_{yb} & G_{yf}-G_{bf} & G_{yd}-G_{bd} & G_{yq}-G_{bq} \\ & & 0 & 0 & 0 \\ & \text{SYMMETICAL ABOUT LEADING DIAGONAL} & & 0 & 0 \\ & & & & 0 \end{bmatrix} \begin{bmatrix} i_r \\ i_y \\ i_f \\ i_d \\ i_q \end{bmatrix} \tag{2.18}$$

### 2.1.2 <u>Saturation *(SF)*</u>

In this thesis, the armature-phase mmf of the generator is assumed to be sinusoidally distributed in space and saturation to be solely produced by the resultant direct-axis mmf. This mmf may be defined in terms of an effective direct-axis current $I_d$ [3], referred to the generator field winding and expressed as

$$i_d = i_f + \frac{N_5}{N_4} i_5 + \frac{N_1}{N_4}\left(i_r \cos\theta_r + i_y \cos\theta_y + i_b \cos\theta_b\right) \qquad (2.19)$$

The generator open-circuit characteristic shown in Fig 2.2(a) relates the open-circuit phase voltage to the field current. The saturation function *SF* shown in Fig 2.2(b), is obtained from the piecewise linearisation of this characteristic and is defined between the break points as [5]

$$\begin{aligned}
\text{If} \quad & 0 \le i_d \le i_{f1} && SF = SF_1 \\
& i_{f1} < i_d < i_{f2} && SF = Grad\,(i_d - i_{f1}) + SF_1 \\
& i_d \ge i_{f2} && SF = SF_2
\end{aligned} \qquad (2.20)$$

$$\text{where} \quad SF_1 = \frac{E_1}{i_{f1}\ Norm} = 1$$

$$SF_2 = \frac{E_2}{i_{f2}\ Norm}$$

$$Norm = \frac{E_1}{i_{f1}}$$

$$\text{and} \quad Grad = \frac{SF_1 - SF_2}{i_{f1} - i_{f2}}$$

In a numerical solution of the generator equations, the value of $i_d$ is obtained at each integration step and *SF* thus calculated from Fig 2.2(b) to gives a realistic representation of saturation according to the currents in the various machine windings.

### 2.1.3 Machine inductance

The angles $\theta_r$, $\theta_y$ and $\theta_b$ refer respectively to the displacement of the centres of the red, yellow and blue armature phase windings from the direct-axis. The angle, $\theta_r$, is defined in Fig 2.1 and

$$\theta_y = \theta_r + 120^\circ$$
$$\theta_b = \theta_r - 120^\circ \tag{2.21}$$

#### a) Self inductances

The self inductance of the r-phase is [3]

$$L_{rr} = L_{ad} SF \cos^2\theta_r + L_{aq} \sin^2\theta_r \tag{2.22}$$

For $L_{yy}$ substitute $\theta_y$ for $\theta_r$ and
and for $L_{bb}$ substitute $\theta_b$ for $\theta_r$.

The self inductances of the field, d-axis damper and q-axis damper windings, are all independent of the rotor position and saturation is included in the d-axis winding inductances. Hence,

$$L_f = L_{fo} SF \tag{2.23}$$
$$L_d = L_{do} SF \tag{2.24}$$
$$L_q = L_{qo} \tag{2.25}$$

#### b) Mutual inductances

The mutual inductance between phases r and y is [3]

$$M_{ry} = M_{ad} SF \cos\theta_r \cos\theta_y + M_{aq} \sin\theta_r \sin\theta_y \qquad (2.26)$$

For $M_{rb}$, retain $\theta_r$ and substitute $\theta_b$ for $\theta_y$

and for $M_{yb}$, substitute $\theta_y$ for $\theta_r$ and $\theta_b$ for $\theta_y$.

The mutual inductances between the r-phase and the field, d-axis and q-axis dampers winding are respectively,

$$M_{rf} = M_f SF \cos\theta_r \qquad (2.27)$$

$$M_{rd} = M_d SF \cos\theta_r \qquad (2.28)$$

$$M_{rq} = M_q \sin\theta_r \qquad (2.29)$$

For $M_{yf}$, $M_{yd}$, and $M_{yq}$, substitute $\theta_y$ for $\theta_r$ and

and for $M_{bf}$, $M_{bd}$, and $M_{bq}$, substitute $\theta_b$ for $\theta_r$.

### 2.1.4 Rate-of-change of inductances

The time rate-of-change of the inductance terms are defined as

$$G = \frac{dL}{dt} = \frac{dL}{d\theta}\frac{d\theta}{dt} \qquad (2.30)$$

a) Rate-of-change of self inductances

It follows using equation (2.22) that

$$G_{rr} = -\omega\left(2 L_{ad} SF \cos\theta_r \sin\theta_r - 2 L_{aq} \cos\theta_r \sin\theta_r\right) \qquad (2.31)$$

For $G_{yy}$ substitute $\theta_y$ for $\theta_r$

and for $G_{bb}$ substitute $\theta_b$ for $\theta_r$.

$$G_{ff} = 0 \qquad (2.32)$$

$$G_{dd} = 0 \qquad (2.33)$$

and $$G_{qq} = 0 \qquad (2.34)$$

b) <u>Rate-of-change of mutual inductances</u>

The time rate-of-change of mutual inductances between the r- and y-phase armature windings follow from equation (2.26) as

$$G_{ry} = -\omega\left[M_{ad}SF\left(\sin\theta_r\cos\theta_y + \cos\theta_r\sin\theta_y\right) - M_{aq}\left(\cos\theta_r\sin\theta_y + \sin\theta_r\cos\theta_y\right)\right] \qquad (2.35)$$

For $G_{rb}$, retain $\theta_r$ and substitute $\theta_b$ for $\theta_y$
and for $G_{ry}$, substitute $\theta_y$ for $\theta_r$ and $\theta_b$ for $\theta_y$.

The time rate-of-change of the mutual inductances between the r-phase and the field, d-axis and q-axis damper winding, follow respectively from equations (2.27) to (2.29) as,

$$G_{rf} = -\omega M_f SF \sin\theta_r \qquad (2.36)$$

$$G_{rd} = -\omega M_d SF \sin\theta_r \qquad (2.37)$$

$$G_{rq} = -\omega M_q \cos\theta_r \qquad (2.38)$$

For $G_{yf}$, $G_{yd}$ and $G_{yq}$, substitute $\theta_y$ for $\theta_r$ and
and for $G_{bf}$, $G_{bd}$ and $G_{bq}$, substitute $\theta_b$ for $\theta_r$.

The dq/phase transformation and the expressions for $L_{ad}$, $L_{aq}$, $L_{fo}$, $L_{qo}$, $M_{ad}$, $M_{aq}$, $M_f$, $M_d$ and $M_q$ are derived in Appendix A.

## 2.2 Diesel Engine/Speed Governor

The diesel engine/governor model is based on a overall block diagram obtained from RAE (West Drayton) [5], and shown in Fig 2.3. The state-variable form of the equation describing the overall system in Fig 2.3 is

$$\begin{bmatrix} \frac{dS_3}{dt} \\ \frac{dS_5}{dt} \\ \frac{d\omega}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{k_2} & 0 & -\frac{k_0}{k_2} \\ (1-\frac{k_1}{k_2})\frac{1}{k_3} & -\frac{1}{k_3} & -\frac{k_1 k_0}{k_2 k_3} \\ 0 & \frac{1}{J} & -\frac{k_f}{J} \end{bmatrix} \begin{bmatrix} S_3 \\ S_5 \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{k_0}{k_2} & 0 \\ \frac{k_1 k_0}{k_2 k_3} & 0 \\ 0 & -\frac{1}{J} \end{bmatrix} \begin{bmatrix} S_1 \\ T_e \end{bmatrix}$$

(2.39)

Equation (2.39) may be solved numerically on a step-by-step basis, using a suitable numerical integration process such as the 4th-order Runge Kutta method. The electrical torque ($T_e$) produced by the synchronous generator is defined in equation (2.18).

## 2.3 AVR/exciter model

The 3-phase synchronous generator model has an automatic voltage regulator *(AVR)* to ensure a constant output voltage. The model adopted for the AVR is based on the type 2 representation described in the IEEE report on excitation system [6] and given in the block diagram of Fig 2.4. The state-variable equation relating to this representation is,

$$
\begin{bmatrix} \frac{dV_4}{dt} \\ \frac{dV_7}{dt} \\ \frac{dV_8}{dt} \\ \frac{dV_f}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{T_a} & -\frac{k_a}{T_a} & \frac{k_a}{T_a} & 0 \\ \frac{k_f'}{T_{f1}} & -\frac{1}{T_{f1}} & 0 & 0 \\ \frac{k_f'}{T_{f2}} & 0 & -\frac{1}{T_{f2}} & 0 \\ \frac{k_e}{T_b} & 0 & 0 & -\frac{1}{T_b} \end{bmatrix} \begin{bmatrix} V_4 \\ V_7 \\ V_8 \\ V_f \end{bmatrix} + \begin{bmatrix} \frac{k_a}{T_a} & -\frac{k_a k_R}{T_a} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{ref} \\ V_T \end{bmatrix} \tag{2.40}
$$

where $k_f' = \frac{k_f}{T_{f2} - T_{f1}}$

Equation (2.40) may be solved by to give the generator field voltage $V_f$ defined in equation (2.10).

### 2.3.1 <u>AVR Feedback Voltage</u> ($V_T$)

In the practical scheme the 3-phase output voltage is transformed and rectified, and the resulting direct voltage is filtered to provide a smooth dc feedback voltage $V_T$ proportional to the output voltage. However, a simplified method of determining numerically the average phase voltage is given below.

The positive and negative peak of each generator phases voltage is recorded during each cycle. At the end of the cycle, the average of the six valued stored represents the AVR feedback voltage $V_T$. The method eliminates the need to model accurately the rectifier/filter arrangement and thereby reduces the program run-times.

## 2.4 Comparison of the theoretical results for a sudden 3-phase short circuit

The following sections will consider the theoretical results for a sudden 3-phase short circuit test on an open circuit synchronous generator using the conventional *dqo* approach.

### 2.4.1 Synchronous generator sudden short circuit

When a sudden armature short circuit is applied to the armature of a synchronous generator, it takes time for the flux to penetrate into the iron of the machine. The corresponding changes which occur in the armature currents can be subdivided into three states. These states are illustrated in Fig 2.5 and they are normally categorised as;

(i) Sub-transient state

(ii) Transient state

(iii) Steady state

These states will be discussed individually in the following sections.

#### 2.4.1.1 Sub-transient state

Fig 2.5 respresents the current waveform of a synchronous generator following a sudden short circuit. It is clear that there is a large momentary increase in current which occurs at the instant the short circuit is applied. Any change in flux linkage is immediately opposed by eddy currents in the iron, although these penetrate only as far as the damper winding. The *sub-transient reactances* associated with this period of change in the direct and quadrature axes are $X_d''$ and $X_q''$ respectively, both of which are small when compared with the steady state *synchronous reactances*, $X_d$ and $X_q$

respectively. The generator sub-transient current ($I''$) given by

$$I'' = \frac{E_0}{X_d''} \tag{2.41}$$

is much greater than the steady state value of the short-circuit current.

#### 2.4.1.2 Transient state

The sub-transient state of a synchronous generator normally lasts for a very short period, until the change in flux linkages have penetrated to the field winding. The flux path then becomes more in iron than in air, and the *transient reactances* for this state $X_d'$ and $X_q'$ are larger than those for the sub-transient state. Hence, the transient state current ($I'$) given by

$$I' = \frac{E_0}{X_d'} \tag{2.42}$$

is smaller than that in the sub-transient state, as is evident in Fig 2.5.

#### 2.4.1.3 Steady state

In the steady state, the change in flux linkage in the magnetic circuit have ceased and the corresponding *synchronous reactances*, $X_d$ and $X_q$, are larger than for the transient state. The steady state armature current is shown in Fig 2.5.

#### 2.4.1.4 The DC offset

A DC offset is initially present in the short-circuit current, as shown in Fig 2.5. The magnitude of this component depends on the instant at which the short circuit is applied, with the maximum offset occurring when the short circuit is applied when the corresponding phase voltage passes through zero.

### 2.4.2 Simulation of a synchronous generator

Simulation results for sudden symmetrical and unsymmetrical short circuit on a 60kVA, 400Hz generator are presented and described in this section. The generator parameters are given in section 2.5.

#### 2.4.2.1 The 3-phase short circuit test

Fig 2.6(a), (b) and (c) shows waveforms of the phase currents of the synchronous generator following a sudden armature short circuit from open circuit. The large current which occurs in the sub-transient state, persists for only about one cycle. The DC offset decays at the same rate as the armature current in the transient state and the steady state is reached after about 0.04s. The field current of the synchronous generator following the short circuit is shown in Fig 2.6(d).

Fig 2.7 (a) and (b) show respectively the transient current in the red armature phase when the sudden short circuit is applied at the instant the red phase voltage passes through zero and at a voltage maximum. Fig 2.7(a) confirms that the maximum DC offset is obtained when switching is at a voltage zero and Fig 2.7(b) that no DC offset occurs when switching is at a voltage maximum.

#### 2.4.2.2 Unbalanced fault situation

Figs 2.8 and 2.9 show respectively the results obtained when two phase to earth, and single phase to earth faults are simulated. It will be seen that, the field currents now contain an additional fundamental frequency component, due to the unbalanced currents in the armature.

The unbalanced stator mmf may be resolved into two components, each rotating at synchronous speed but in opposite directions. The backwards-rotating component induces second-harmonic currents in the rotor, which in turn give rise to higher-order harmonic currents in the rotor windings. The effect of these is responsible for the non sinusoidal armature current evident in both figures.

#### 2.4.2.3 Load application and rejection

Fig 2.10 and 2.11 show the armature currents and the field current of the generator following the sudden application to the unloaded generator of rated load of 0.8pf and 0.2pf lag respectively. It is clear that, for the load of 0.8pf, the highly resistive load impedance causes the armature currents to rise rapidly to the new steady state. However, in the case of the 0.2pf load, the time taken to achieve the steady state is much longer due to the load impedance now being highly reactive. In addition, oscillatory currents at fundamental frequency are seen prominently in the field winding for a load application of 0.2pf. However, in the case of the 0.8pf load, no oscillatory current is evident in the field winding due to the more resistive nature of the load.

Figs 2.12 and 2.13 show the armature currents of the generator following the sudden rejection of rated load at 0.8pf and 0.2pf lag respectively. It is clear that, in both cases, the armature current falls instantaneously to zero.

From all these computer simulation tests, it is clear that the phase reference frame is highly flexible for the modelling of a synchronous generator. It can cope with both balanced and unbalanced load conditions and also allows a saturation factor to be included in the model.

## 2.5 Dqo and phase parameters for the synchronous generator

A 60kVA synchronous generator with the following parameters were used to provide the data for the simulation. Base per unit values were taken as 60kVA and 160V/phase. Parameters with a *bar* denote per unit values.

$$Z = 0.6930\Omega$$
$$\overline{X_d} = 1.7753$$
$$\overline{X_{md}} = 1.6664$$
$$\overline{X_q} = 0.9251$$
$$\overline{X_{mq}} = 0.8162$$
$$\overline{X_d'} = 0.2506$$
$$\overline{X_d''} = 0.1998$$
$$\overline{X_q''} = 0.1735$$
$$\overline{R_a} = 0.0186$$
$$R_{ff} = 0.6119\Omega$$

The generator phase parameters as obtained from the dqo/phase transformations (See Appendix A) are:

$$L_{ad} = 0.3280\text{mH}$$
$$L_{aq} = 0.1724\text{mH}$$
$$L_f = 103.96\text{mH}$$

$L_d = 0.0392 mH$

$L_q = 0.0181 mH$

$M_{ad} = 0.3216 mH$

$M_{aq} = 0.1652 mH$

$M_{fd} = 1.7993 mH$

$M_f = 5.3979 mH$

$M_d = 0.1021 mH$

$M_q = 0.0500 mH$

$R_{rr} = 0.0128 \Omega$

$R_{yy} = 0.0128 \Omega$

$R_{bb} = 0.0128 \Omega$

$R_{ff} = 0.6119 \Omega$

$R_{dd} = 0.9262 m\Omega$

$R_{qq} = 0.01672 m\Omega$

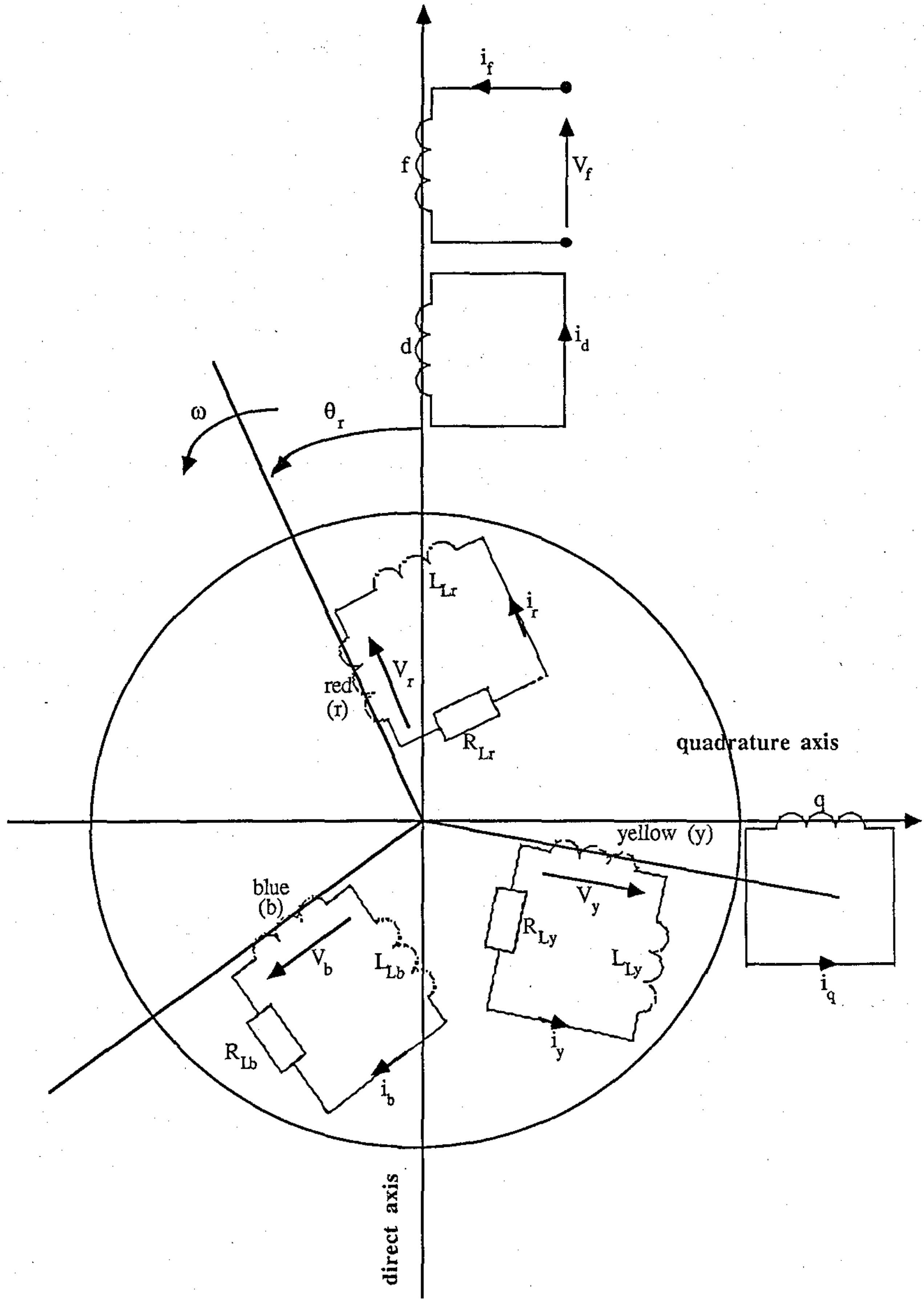
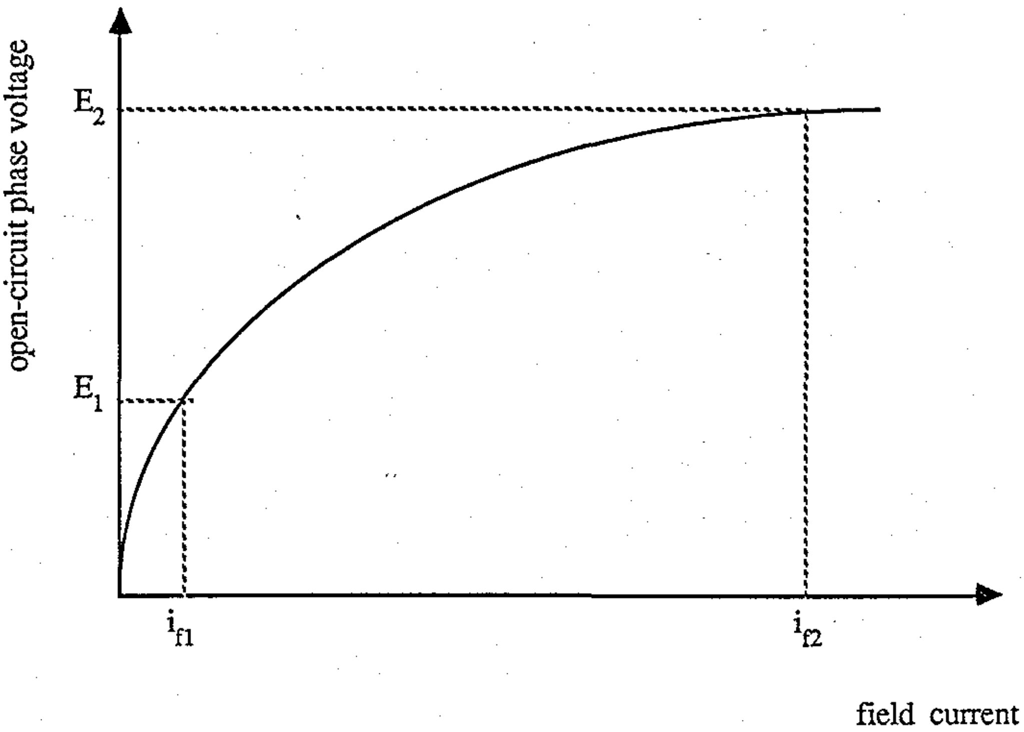
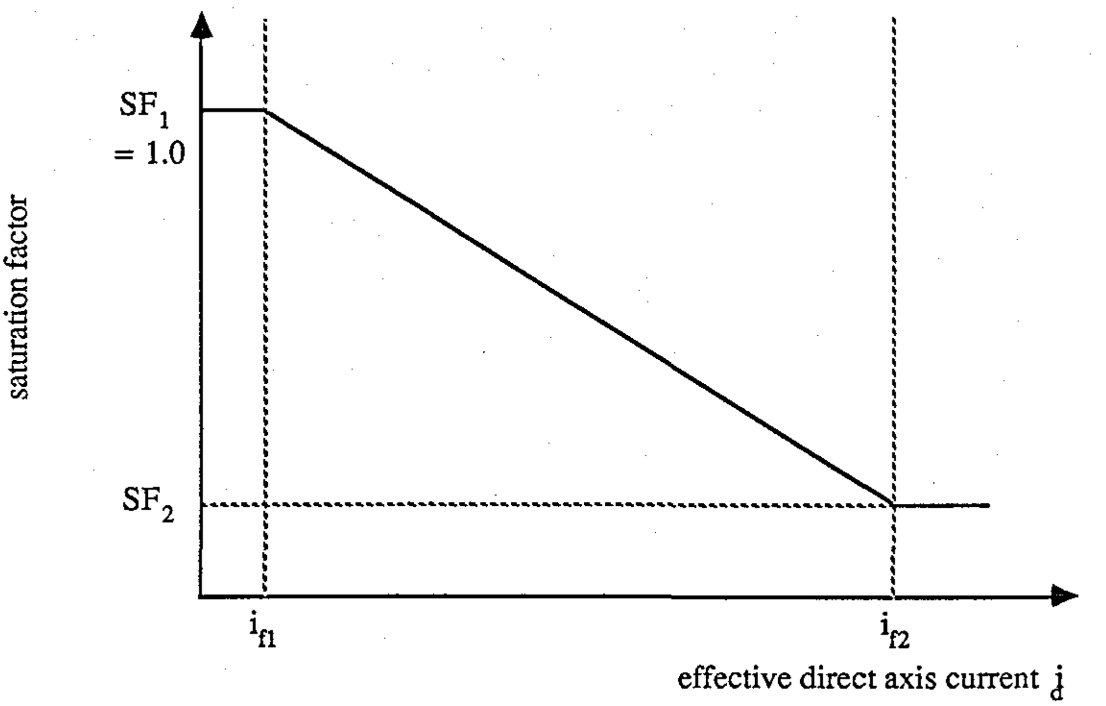



**Fig 2.1 An ideal 3-phase synchronous generator**




(a)




(b)

**Fig 2.2 Determination of saturation factor**

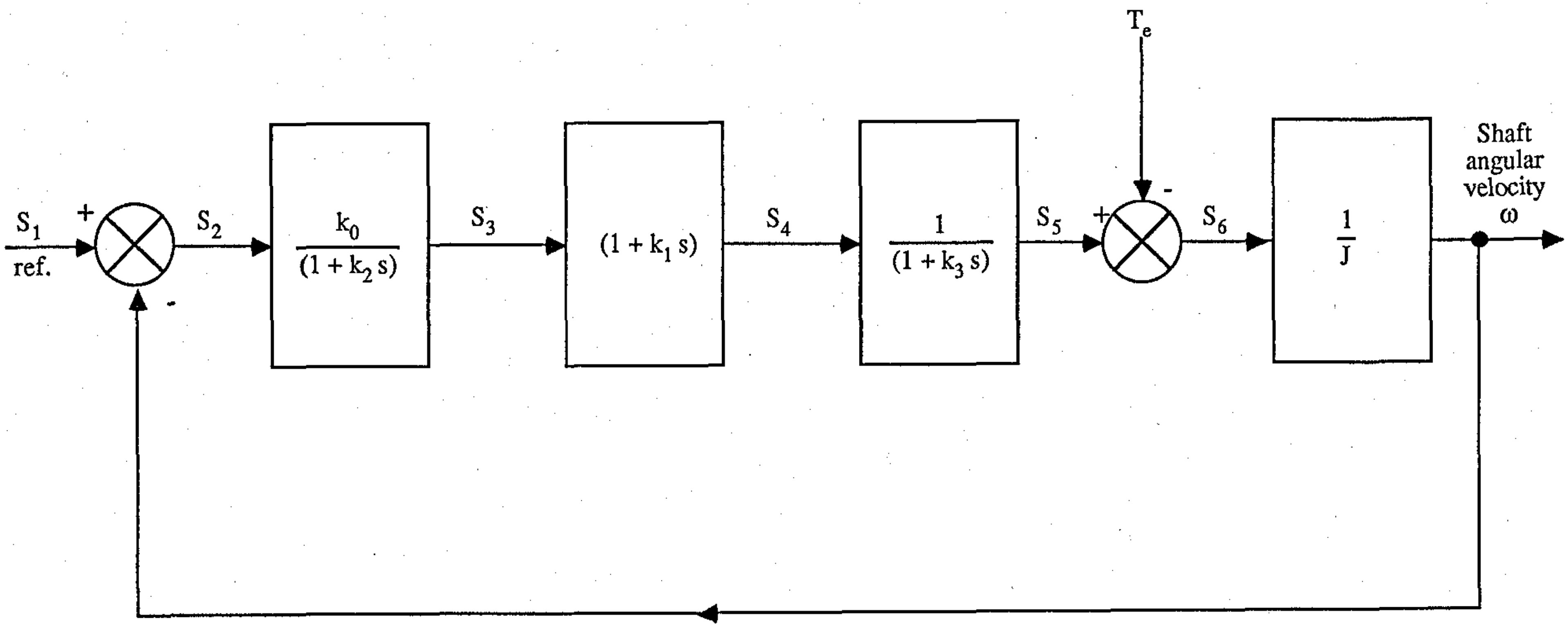
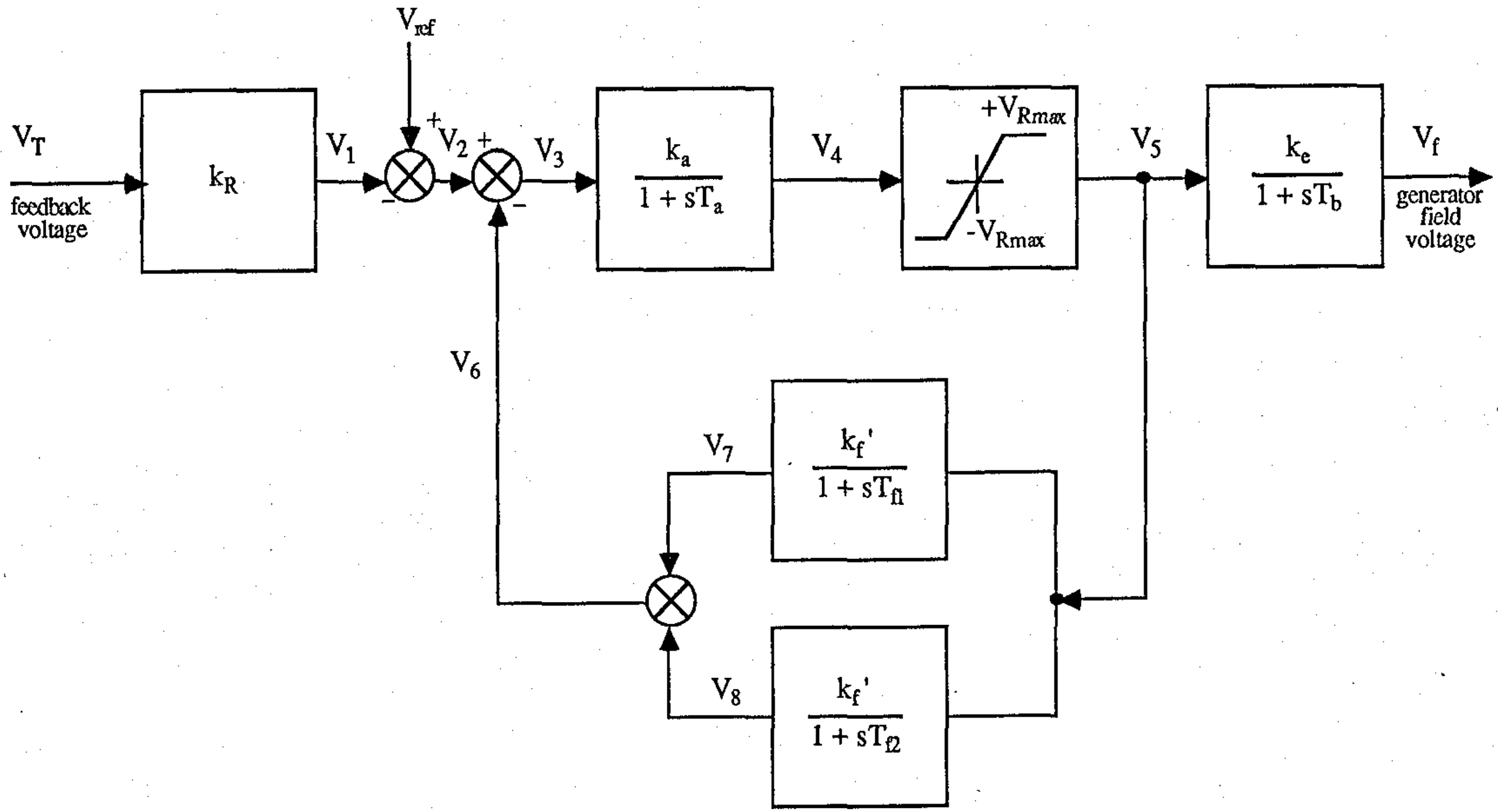



**Fig 2.3 Diesel Engine/Governor Block Diagram**




# Fig 2.4 AVR Block Diagram

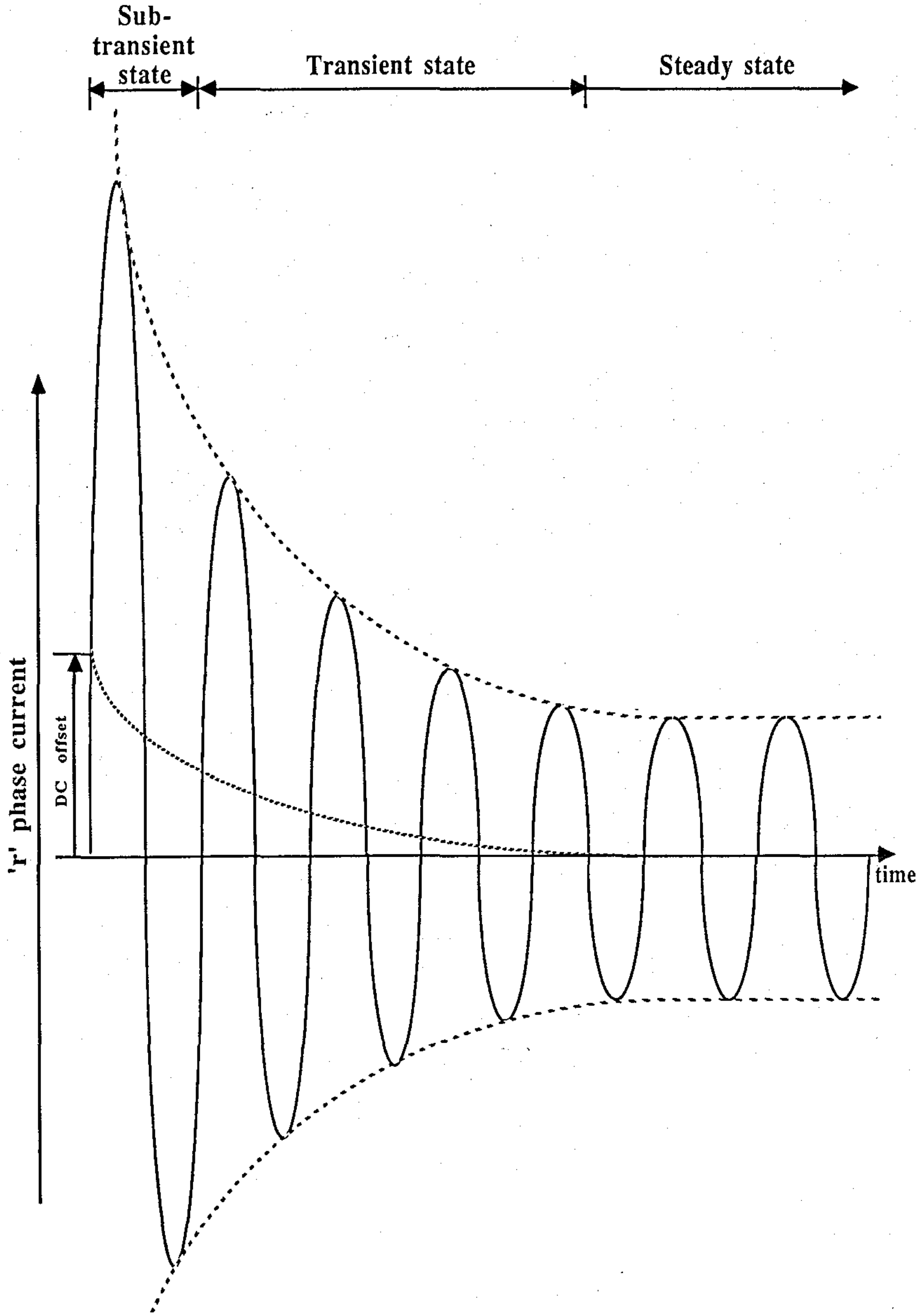



Fig 2.5 Synchronous Current: Sudden Short Circuit appied from open circuit

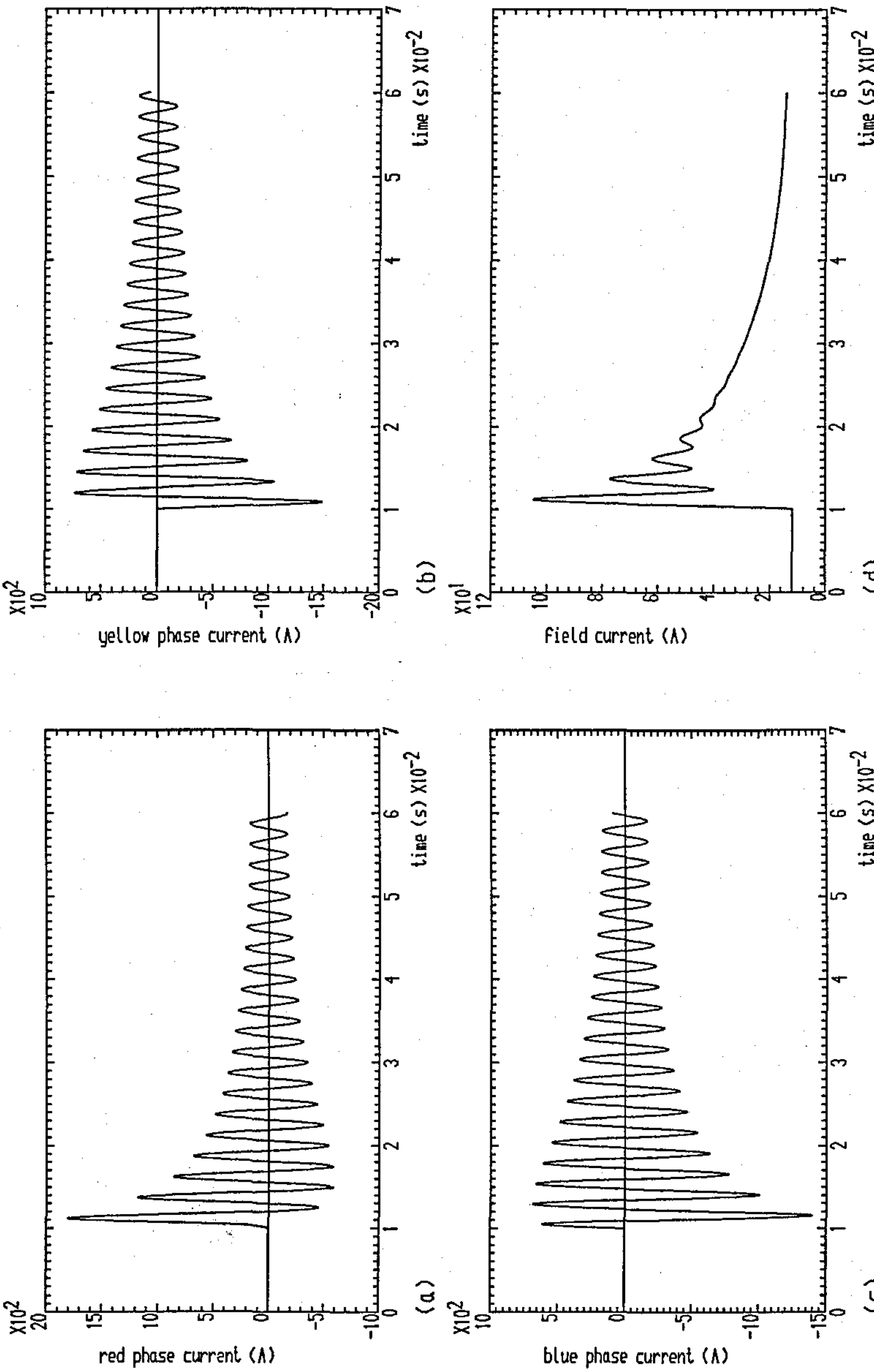



Fig 2.6 Synchronous generator: sudden armature short circuit from open circuit

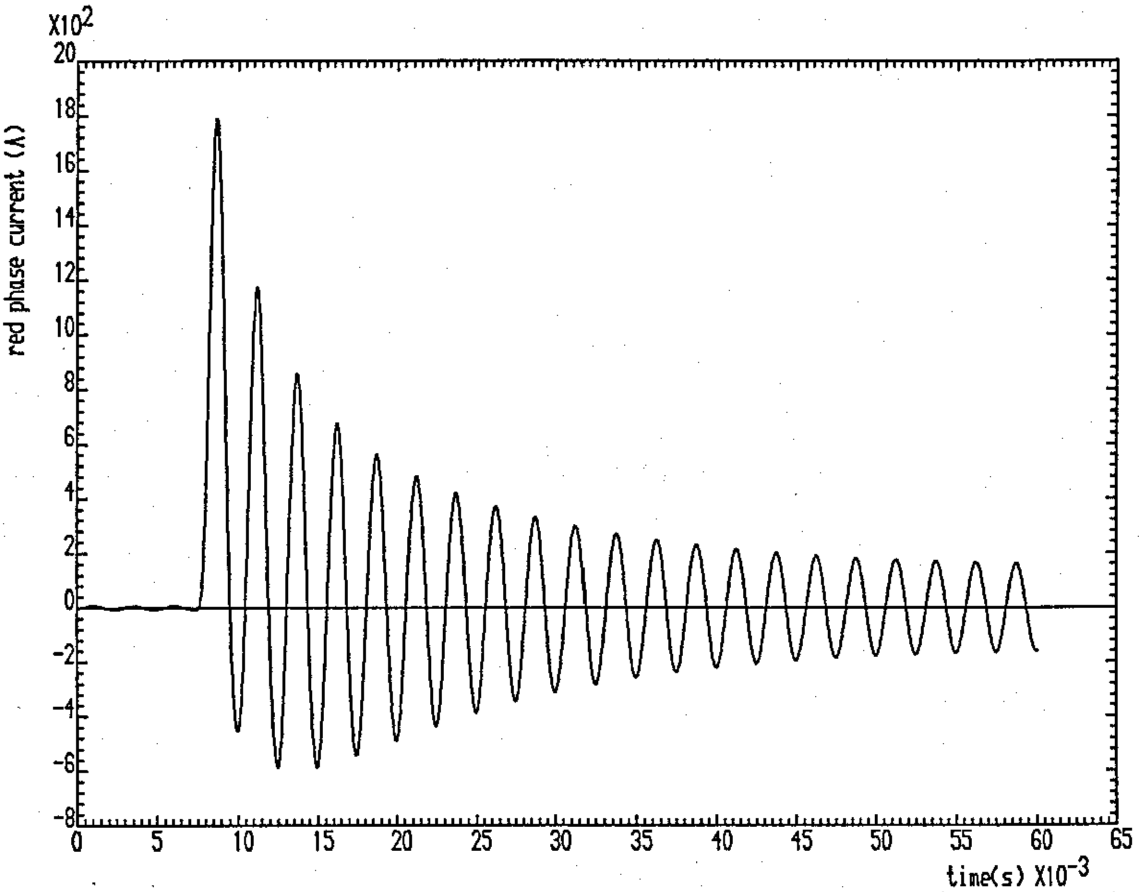



Fig2.7(a) Synchronous generator red phase current when short circuit applied at red phase voltage zero from open circuit

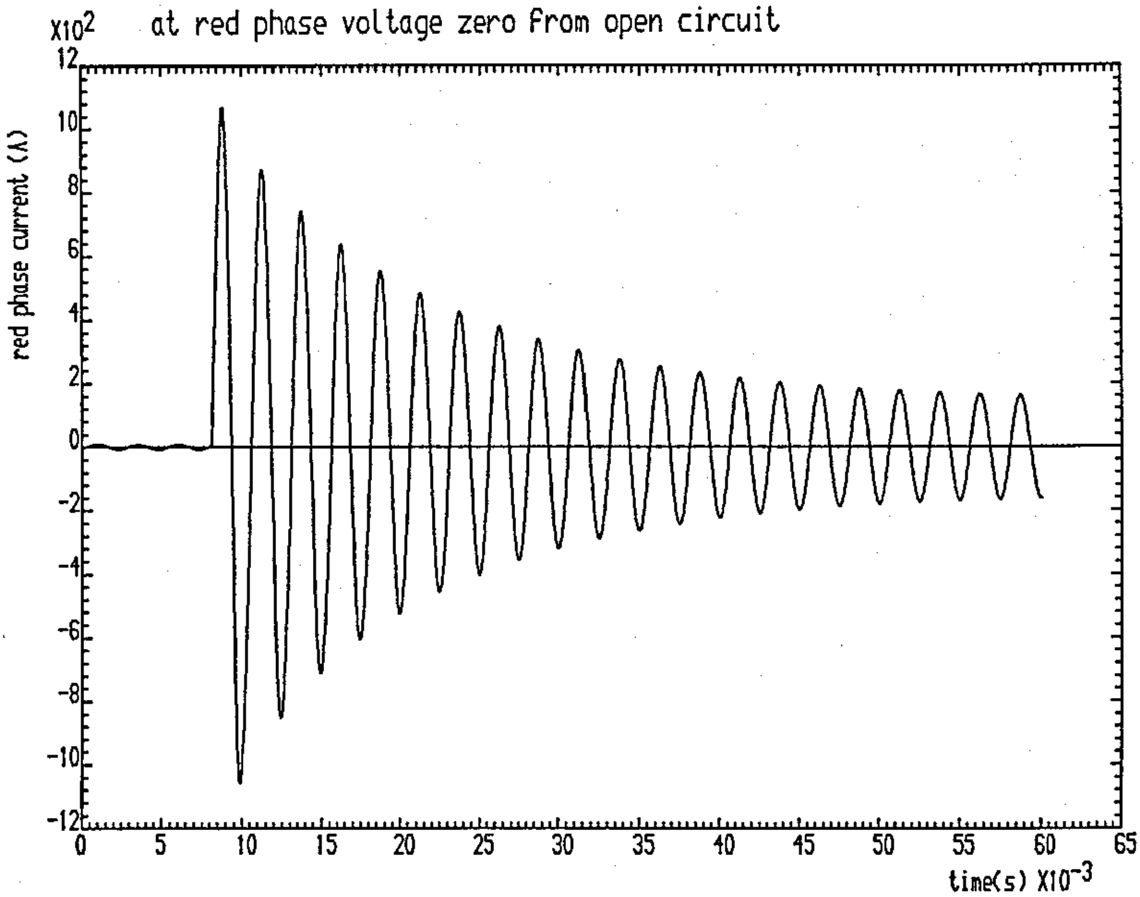



Fig2.7(b) Synchronous generator red phase current when short circuit applied at red phase voltage maximum from open circuit

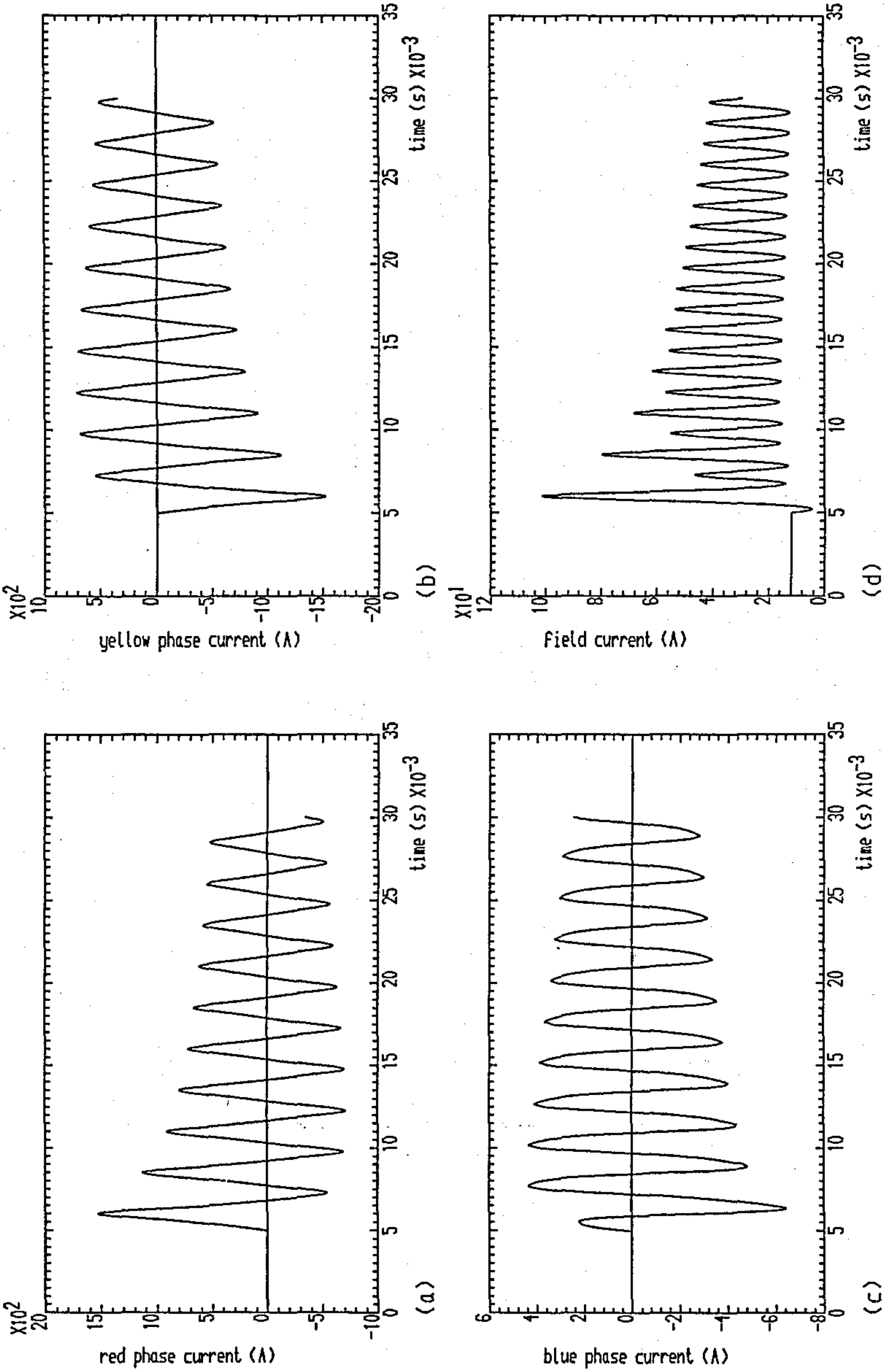



Fig 2.8 Synchronous generator: sudden armature short circuit applied between red and yellow phases from open circuit

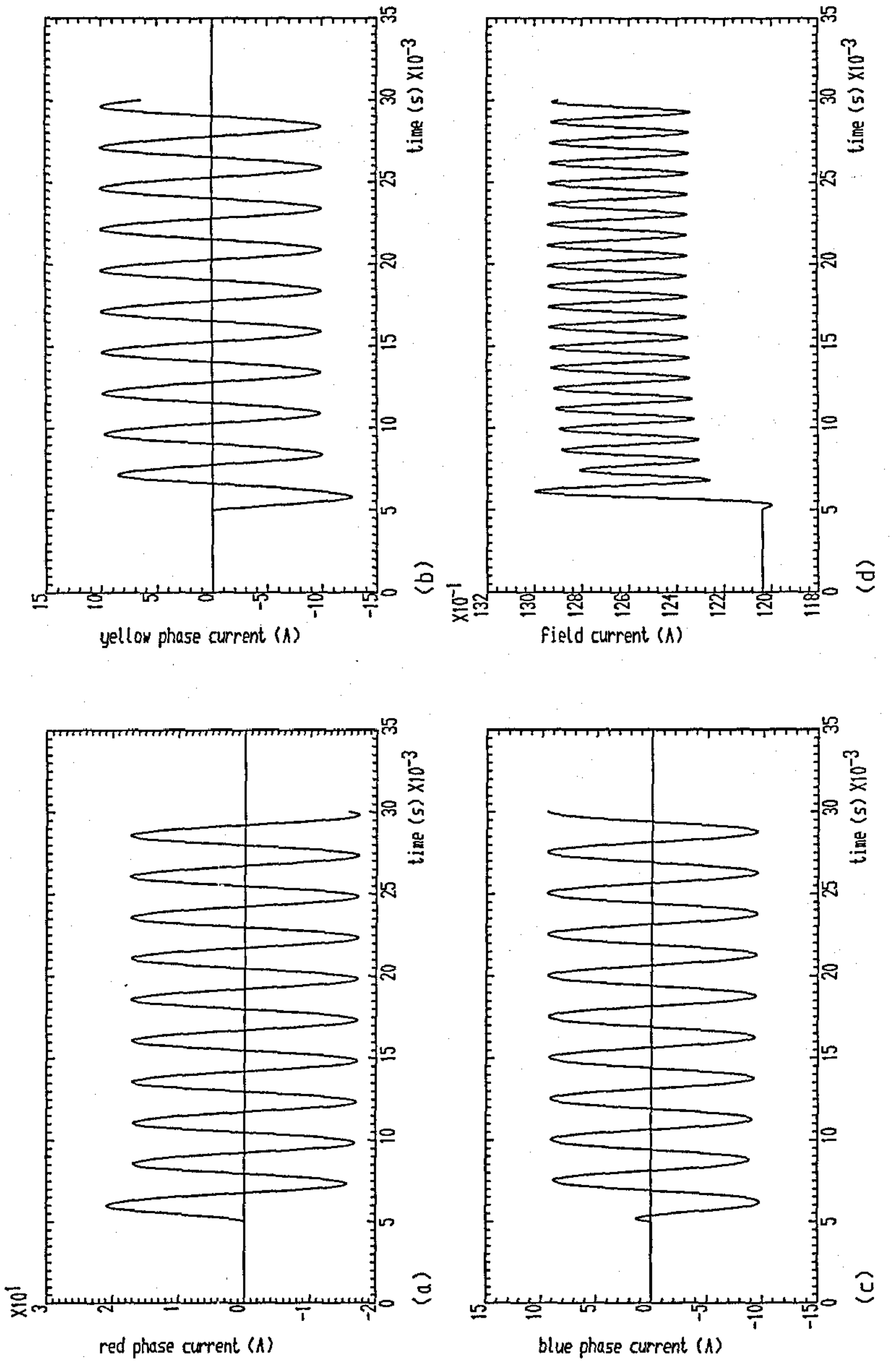



Fig 2.9 Synchronous generator: sudden armature short circuit applied to red phase to neutral from open circuit

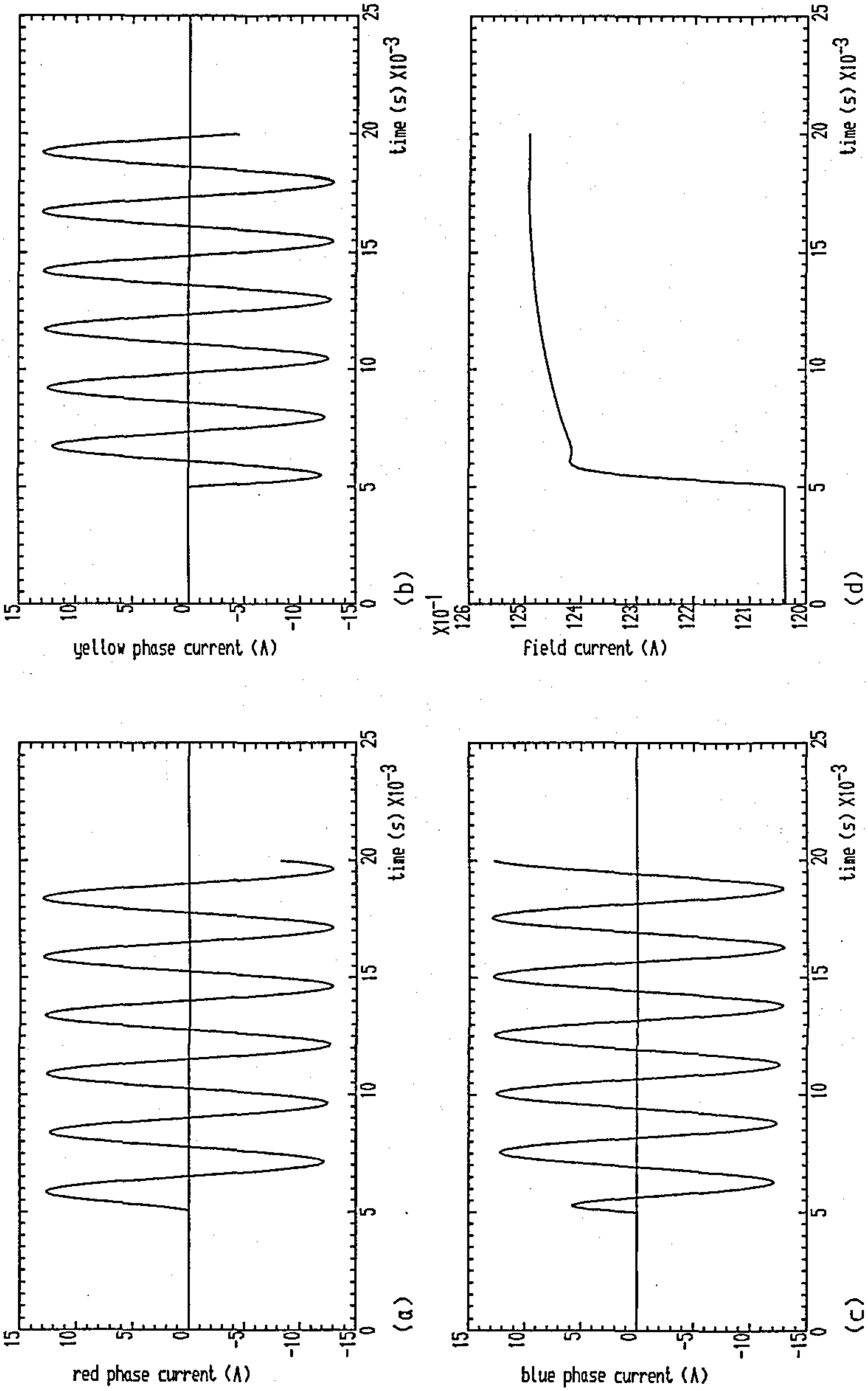



Fig 2.10 Synchronous generator currents following sudden application of rated load at 0.8pf lag

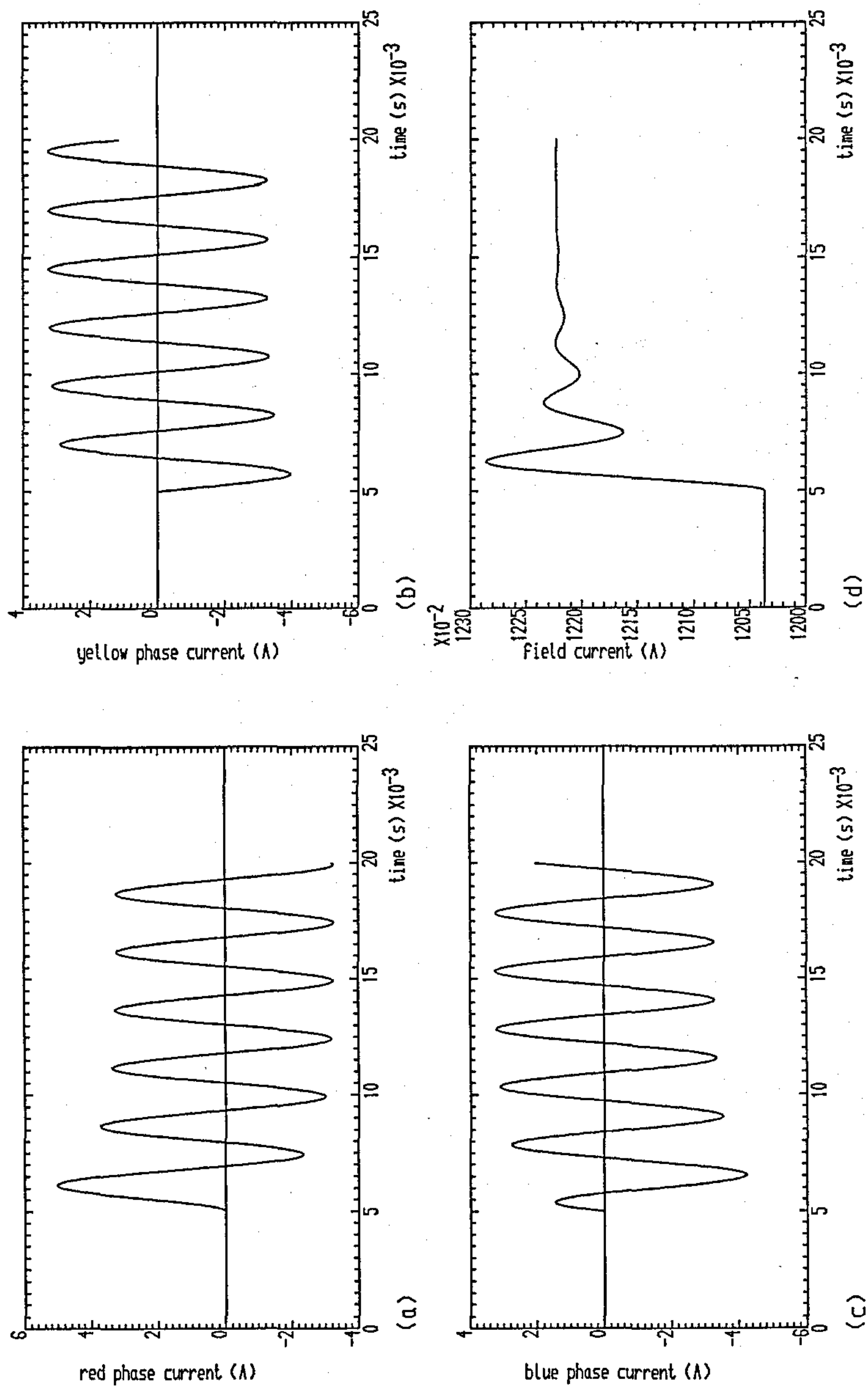



Fig 2.11 Synchronous generator currents following sudden application of rated load at 0.2pf lag

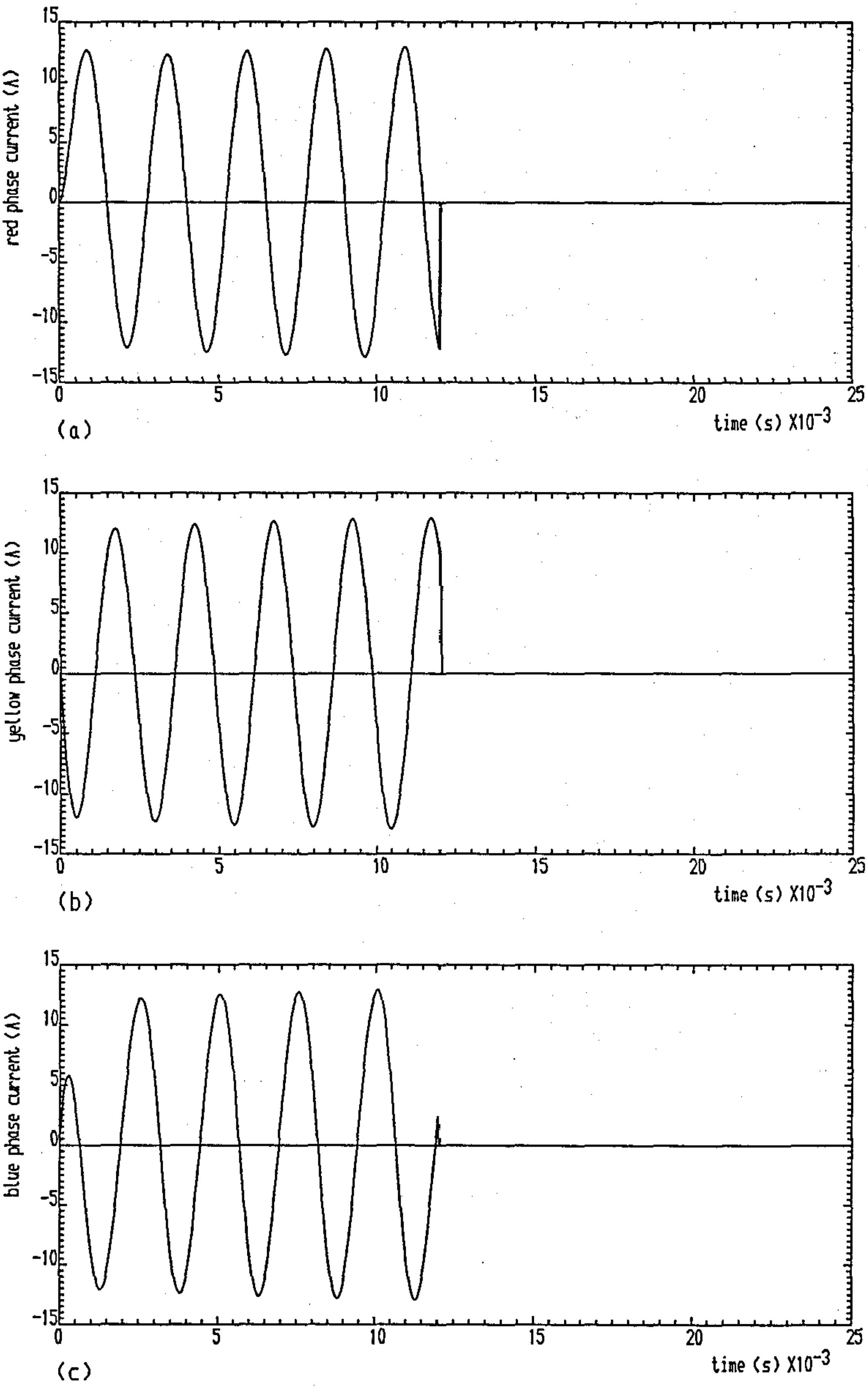



Fig 2.12 Synchronous generator currents following sudden rejection of rated load at 0.8pF lag

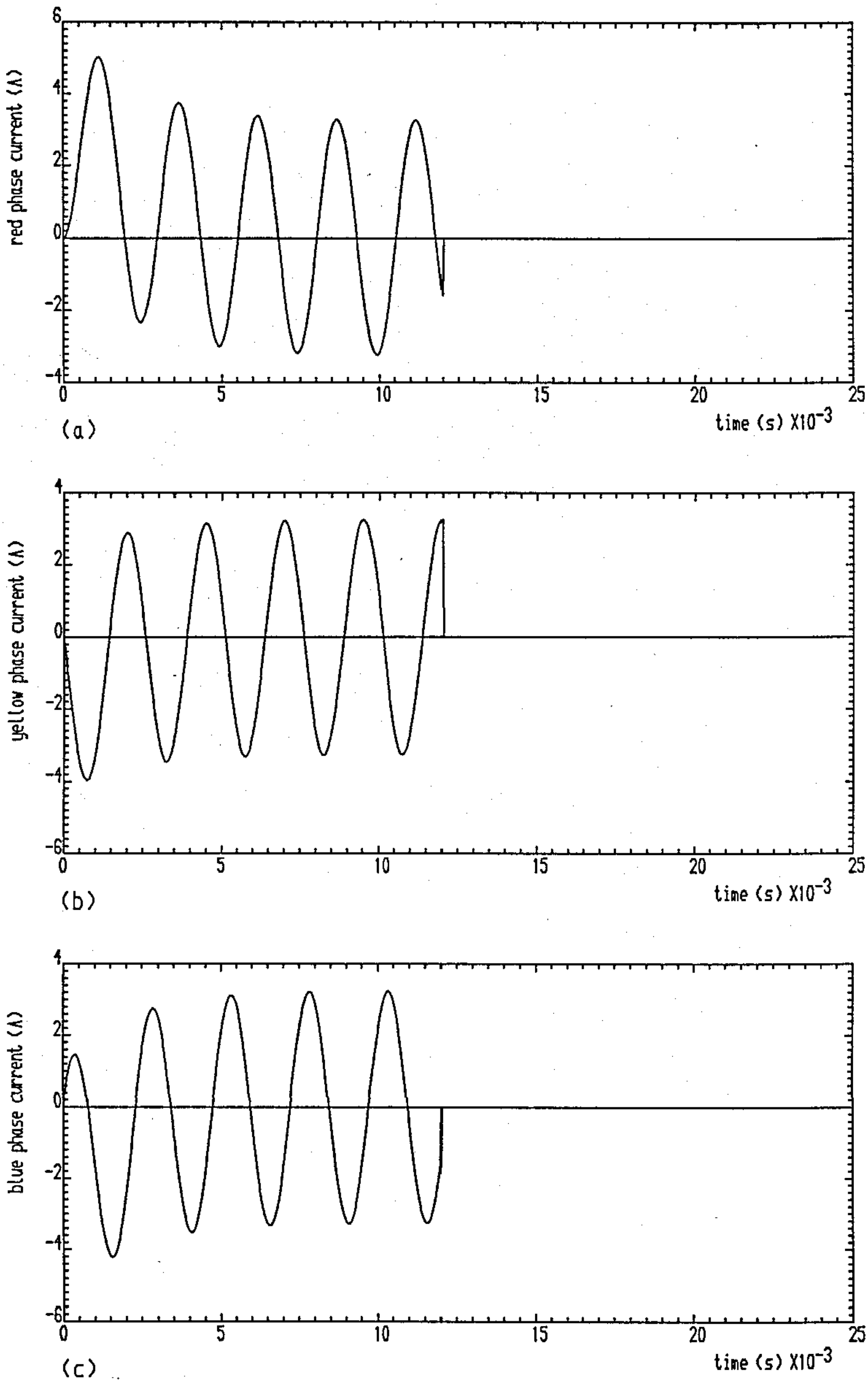



Fig 2.13 Synchronous generator currents following sudden rejection of rated load at 0.2pf lag

## Chapter 3

# SIMULATION OF THE MOTOR/GENERATOR SET

This chapter describes a mathematical model for a motor/generator set comprising a 3-phase synchronous machine directly coupled to a separately-excited DC machine. A block diagram of the arrangement considered is shown in Fig 3.1.

The model used for the synchronous generator is as described in the previous chapter, while that for the DC machine is based on the equivalent circuit of Fig 3.2. The electrical and mechanical equations for the MG set are presented below, together with results illustrating the transient performance of the set.

## 3.1 Synchronous machine

The synchronous machine of Fig 3.1 is capable of both generator and motor operation and the following sections present the electrical equations for both cases.

### 3.1.1 Generator operation

The electrical behaviour of this machine is defined by equation (2.12) and its output torque ($T_e$) may be calculated using equation (2.18).

### 3.1.2 Motor operation

Fig 3.3 is a circuit representation of a 3-phase synchronous motor with a damper winding on both the direct and quadrature axes. The armature currents on the rotor now flow in the opposite direction to those described in the generator model of section 2.1.

As a result, the direction of the torque produced is reversed. If the same analysis is followed as for the generator in section 2.1, the resulting voltage equation is

$$\begin{bmatrix} V_r - V_b \\ V_y - V_b \\ V_f \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} R_{rr} + R_{bb} & R_{bb} & 0 & 0 & 0 \\ & R_{yy} + R_{bb} & 0 & 0 & 0 \\ \text{SYMMETRICAL} & & R_{ff} & 0 & 0 \\ \text{ABOUT LEADING} & & & R_{dd} & 0 \\ \text{DIAGONAL} & & & & 0 \end{bmatrix} \begin{bmatrix} i_r \\ i_y \\ i_f \\ i_d \\ i_q \end{bmatrix} +$$

$$\frac{d}{dt} \begin{bmatrix} L_{rr} + L_{bb} - 2M_{br} & L_{bb} - (M_{br} + M_{yb} - M_{ry}) & M_{rf} - M_{bf} & M_{rd} - M_{bd} & M_{rq} - M_{bq} \\ & L_{yy} + L_{bb} - 2M_{yb} & M_{yf} - M_{bf} & M_{yd} - M_{bd} & M_{yq} - M_{bq} \\ & & L_{ff} & M_{fd} & 0 \\ \text{SYMMETRICAL ABOUT} & \text{LEADING DIAGONAL} & & L_{dd} & 0 \\ & & & & L_{qq} \end{bmatrix} \begin{bmatrix} i_r \\ i_y \\ i_f \\ i_d \\ i_q \end{bmatrix}$$

(3.1)

As in case of the generator model in section 2.1, the motor model defined by equation (3.1) may be re-stated in the form suitable for numerical integration

$$\left[\frac{dI_m}{dt}\right] = \left[L_{mm}\right]^{-1} \left[V_m - \left(R_{mm} + \frac{dL_{mm}}{dt}\right) I_m\right] \qquad (3.2)$$

and as in section 2.1, the output torque $T_e$ may be calculated using equation (2.18).

## 3.2 DC machine

The following sections present the electrical and mechanical equations for the DC machine of Fig 3.2 during both motor and generator operation.

### 3.2.1 Motor operation

When the synchronous machine is generating and the DC machine motoring, the direction of the armature current in the latter machine is as given in Fig 3.2 and the following differential equations may be deduced.

a) Armature circuit

$$E_a = \left[R_a + R_h + R_y\right] i_a + \left[L_a + L_h + L_y + M_{ha} + M_{ha}\right] \frac{di_a}{dt} + M_{sf}\frac{di_f}{dt} + V_a \tag{3.3}$$

Re-arranged in a form suitable for numerical integration

$$\frac{di_a}{dt} = \frac{E_a - \left[R_a + R_h + R_y\right] i_a - M_{sf}\frac{di_f}{dt} - V_a}{\left[L_a + L_h + L_y + M_{ha} + M_{ah}\right]} \tag{3.4}$$

b) Field circuit

$$V_f = R_f i_f + L_f \frac{di_f}{dt} + M_{fs}\frac{di_a}{dt}$$

or

$$\frac{di_f}{dt} = \frac{V_f - R_f i_f - M_{fs}\frac{di_a}{dt}}{L_f} \tag{3.5}$$

c) Torque

The torque-balance equation for the MG set is

$$T_e = T_{eg} + k_{ff}\omega + J\frac{d\omega}{dt}$$

which may be rearranged as

$$T_{eg} = k\, i_f\, i_a$$

$$\text{where} \quad \frac{d\omega}{dt} = \frac{1}{J}\left[T_e - k\, i_f\, i_a - k_{ff}\,\omega\right] \tag{3.6}$$

## 3.2.2 Generator operation

When the synchronous machine is motoring and the DC machine generating, the direction of the armature current flow is reversed and the differential equations become:

a) Armature circuit

$$V_a = \left[R_a + R_h + R_y\right] i_a + \left[L_a + L_h + L_y + M_{ha} + M_{ah}\right]\frac{di_a}{dt} +$$
$$M_{sf}\frac{di_f}{dt} + E_a$$

or

$$\frac{di_a}{dt} = \frac{V_a - \left[R_a + R_h + R_y\right] i_a - M_{sf}\frac{di_f}{dt} - E_a}{\left[L_a + L_h + L_y + M_{ha} + M_{ah}\right]} \tag{3.7}$$

b) Field circuit

The field circuit is independent of the operating mode of the DC machine. and $i_f$ can always be calculated using equation (3.5).

c) Torque

The torque-balance equation is now

$$T_{eg} = T_e + k_{ff}\omega + J\frac{d\omega}{dt}$$

or

$$\frac{d\omega}{dt} = \frac{1}{J}\left[k\, i_f\, i_a - T_e - k_{ff}\,\omega\right] \qquad (3.8)$$

## 3.3 MG set performance during load switching

Two load application tests were performed to illustrate the MG set performance with the DC machine motoring. Firstly, the application of rated load impedance at 0.8 pf lag from open circuit gave the results shown in Fig 3.4. The load impedance is substantially resistive, causing the armature currents to rise rapidly to the new steady state. The application of rated load impedance at 0.4 pf lag gave the results shown in Fig 3.5, in which the armature currents again rise to the steady state but in a longer time, due to more reactive nature of the load.

Two load rejection tests were performed to illustrate the MG set performance with the DC machine motoring. Firstly, the rejection of rated load at 0.8 pf lag gave the results shown in Fig 3.6, where the armature currents falls instantaneously to zero. A second load rejection using rated load at 0.4 pf lag produced a similar result as shown in Fig 3.7.

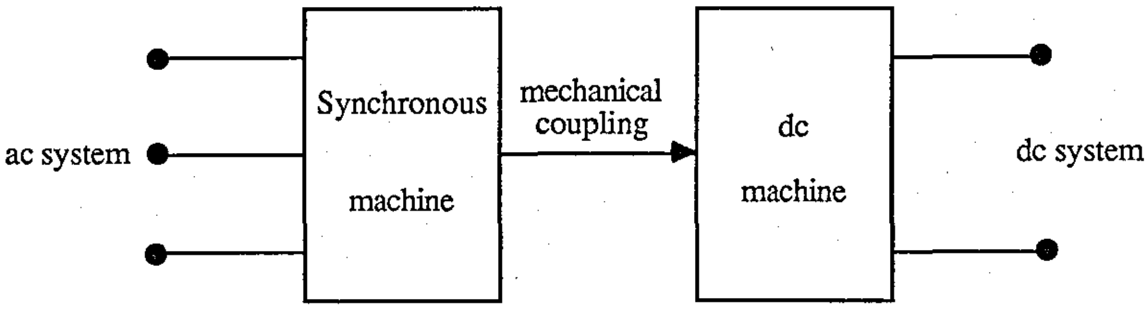



**Fig 3.1 Block Diagram of motor/generator set**

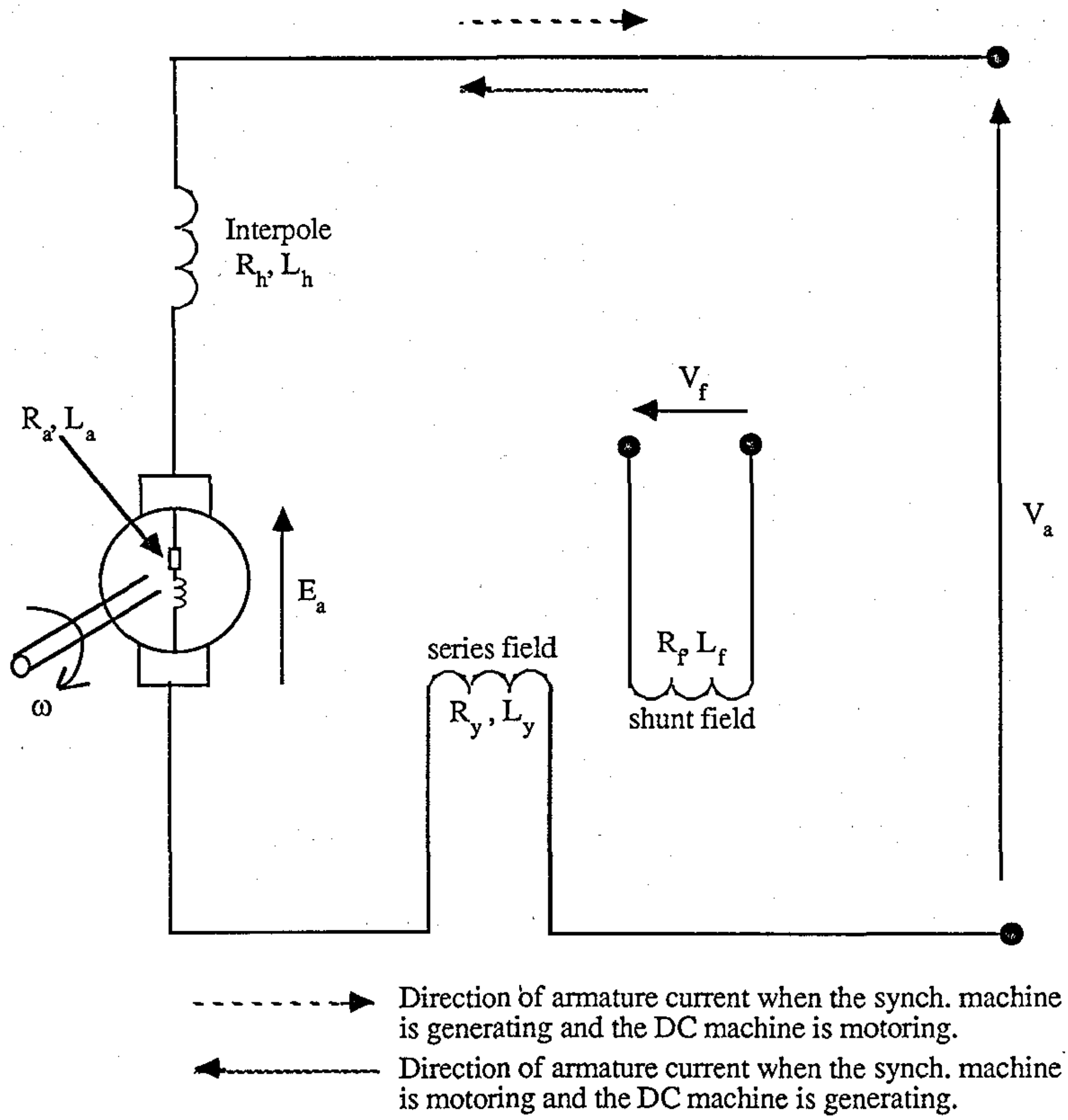



**Fig 3.2 Equivalent circuit for dc machine**

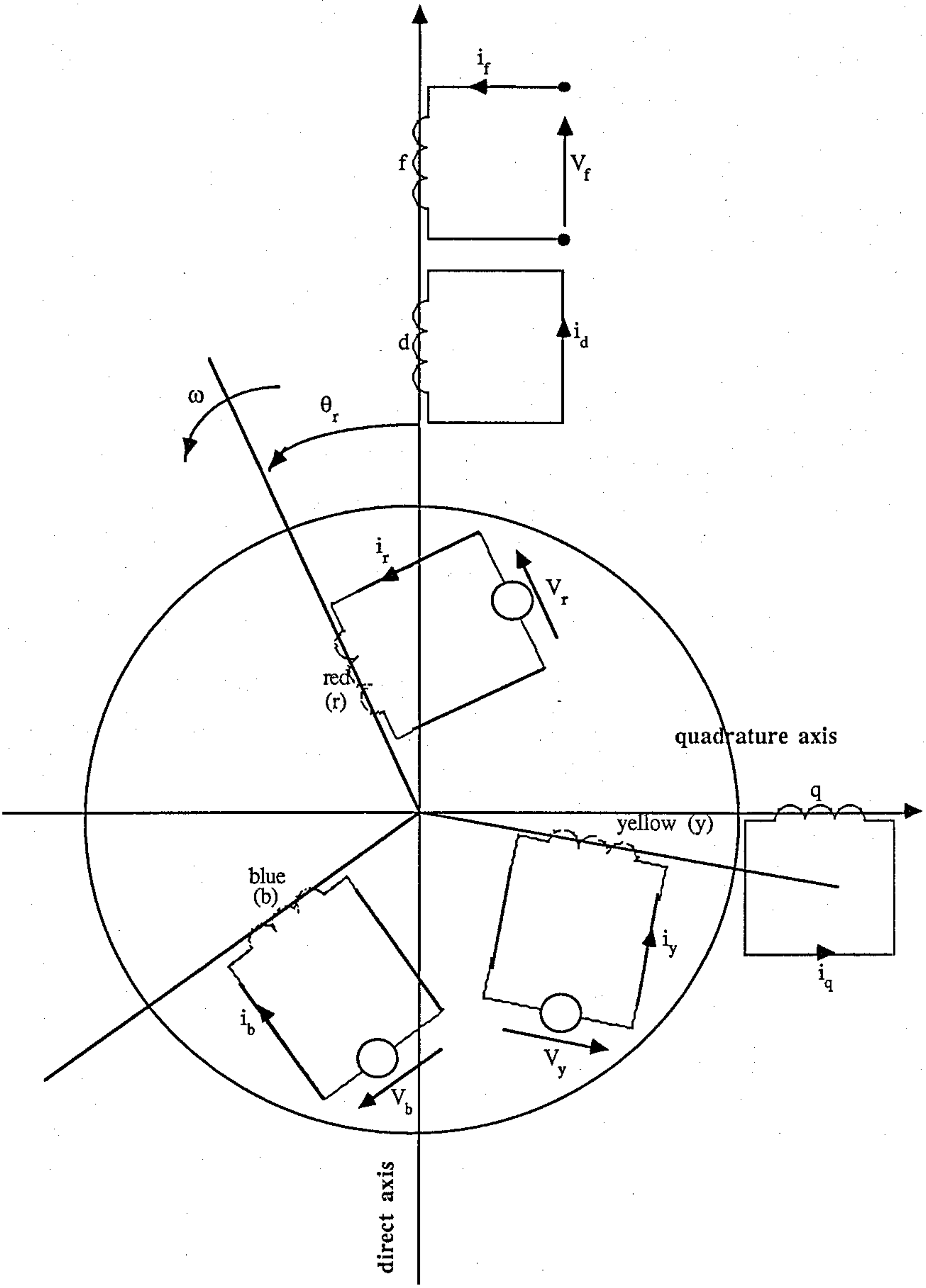



Fig 3.3 An ideal 3-phase synchronous motor

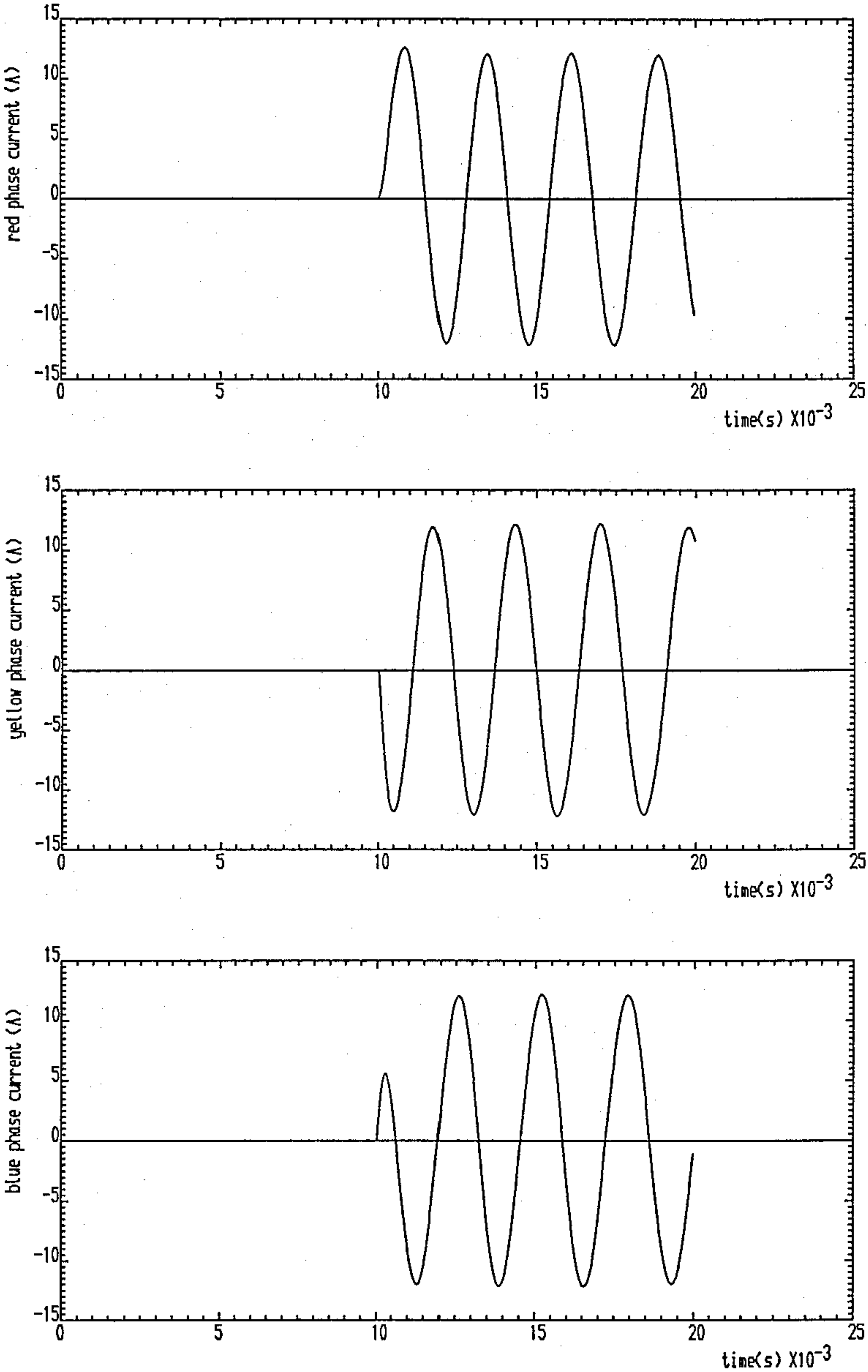



Fig 3.4 AC line currents of MG set following application of rated load at 0.8pf

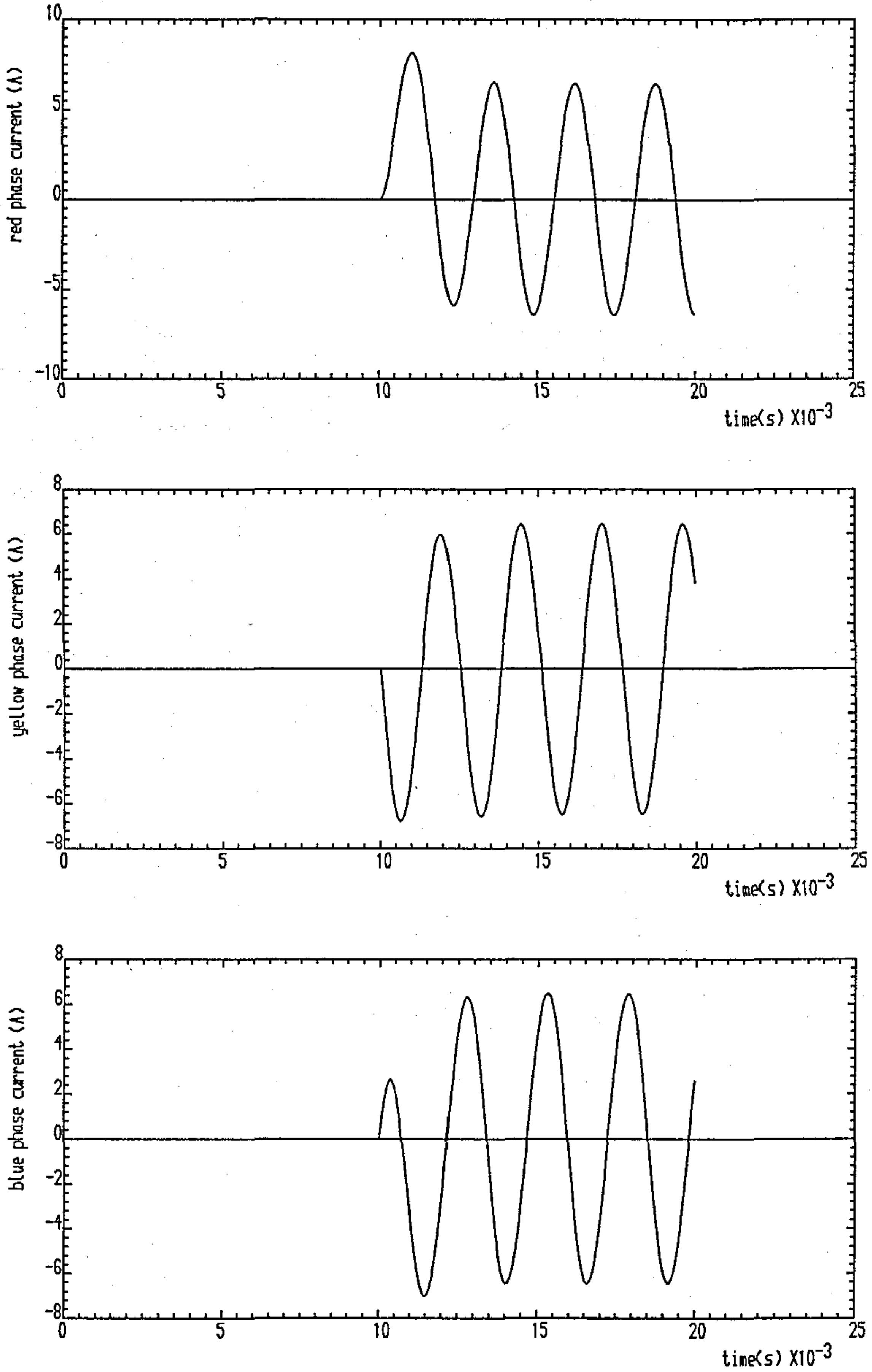



Fig 3.5 AC line currents of MG set following application of rated load at 0.4pf

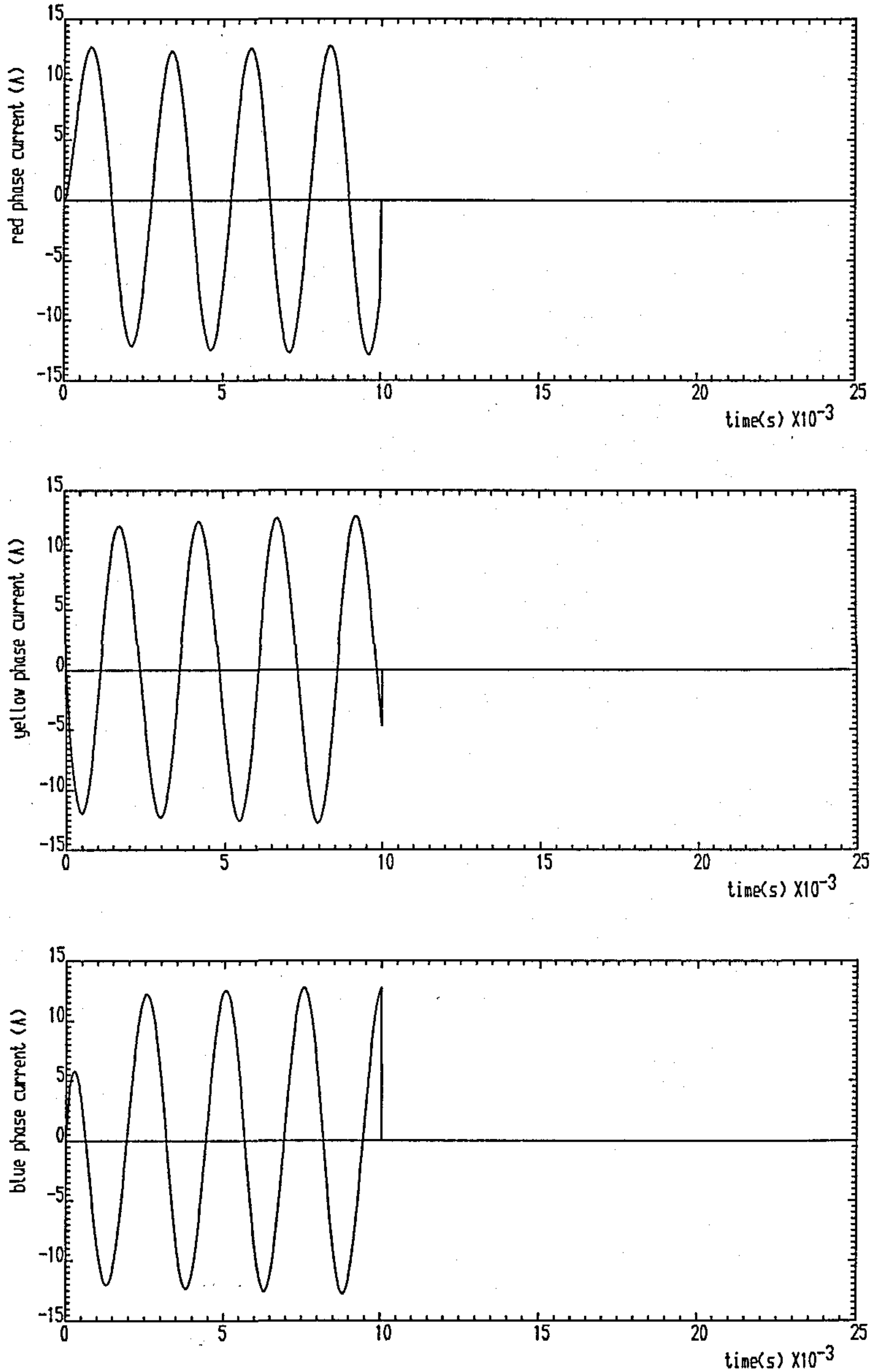



Fig 3.6 AC line currents of MG set following rejection of rated load at 0.8pf

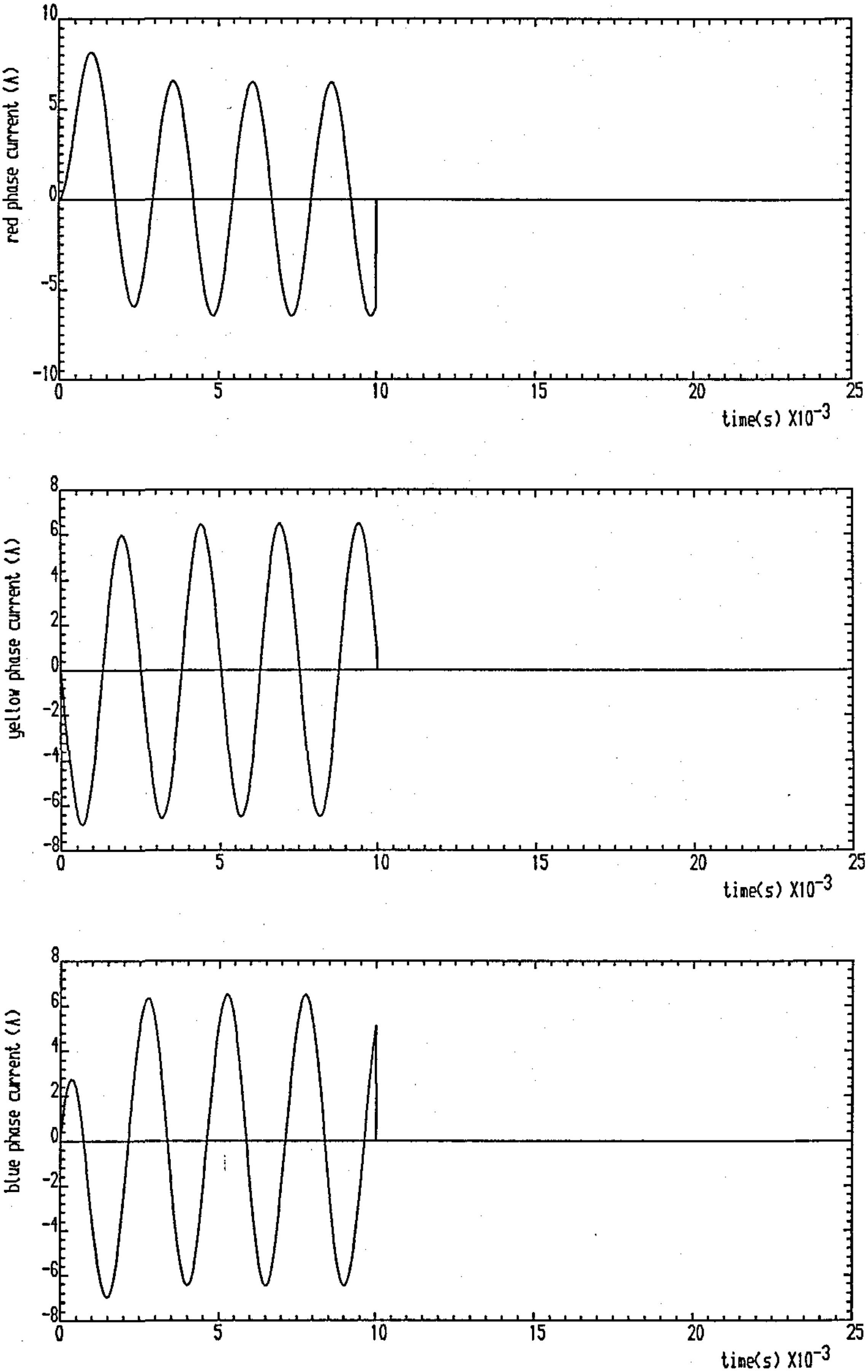



Fig 3.7 AC line currents of MG set following rejection of rated load at 0.4pf

## Chapter 4

# BUS-COUPLER REPRESENTATION

A bus coupler is a section-switch which can be opened or closed in order to direct the current flow in a circuit. As used in Fig 1.1, it will allow current to be directed to any junction, so that, for example, it is possible to direct power flow from synchronous generator 1 to the converter simply by closing SW1, SW4 and SW5 and opening SW2 and SW3.

The section switches (SW1 to SW5) in Fig 1.1 have to be effectively switched in and out during the simulation, a process which is regarded as happening instantaneously and which may be analyzed using the constant flux-linkage theorem. However, in a computer solution this will consume considerable computing time and the alternative and simpler approach outlined below was therefore adopted.

In the computer program, switching from an open to a closed state is achieved by an instantaneous change of the switch inductance from a very high to a low value. The more complex process of switching from a closed to an open state is best implemented by an initial change in the inductance of one pole of the switch from a low to a very high value at the instant of a current zero when the 'a' phase in Fig 4.1 is opened. This process eliminates the need for constant flux-linkage considerations, and is close to the practical situation where current extinction usually occurs at a current zero. The remaining two poles of the switch are opened subsequently as the corresponding 'b' and 'c' phase currents fall to zero, as indicated in Fig 4.2, when their inductances are increased to the high value. Fig 4.3 shows the circuit representation when the switch is closed, with the values of L and R being simply the cable inductance and resistance respectively.

Tensor mesh analysis is used to convert a branch reference frame the bus coupler (see Fig 4.4(a)) to a mesh reference frame (see Fig 4.4(b)). The reason for this conversion becomes apparent from consideration of the associated resistive/inductive matrix, which in the branch reference frame is both large and complicated, but by means of such a conversion becomes smaller and simplified. By using tensor mesh analysis the rank of the matrix is reduced, which considerably reduces the computing time is required in a numerical solution.

## 4.1 Tensor mesh analysis

The equations which relate the voltages and currents of the bus coupler in the branch reference frame follow from Fig 4.4(a) as

$$\begin{bmatrix} E_1 \\ E_2 \\ E_3 \end{bmatrix} + \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} R_1 & & \\ & R_2 & \\ & & R_3 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} + \begin{bmatrix} L_1 & & \\ & L_2 & \\ & & L_3 \end{bmatrix} \begin{bmatrix} \frac{di_1}{dt} \\ \frac{di_2}{dt} \\ \frac{di_3}{dt} \end{bmatrix} \tag{4.1}$$

or, in abbreviated form,

$$E_b + V_b = R_{bb} I_b + L_{bb} \frac{dI_b}{dt} \tag{4.2}$$

The abbreviated form of the mesh reference frame equations, which is concerned with the mesh equations relating to the closed meshes of Fig 4.4(b) is

$$E_m + V_m = R_{mm} I_m + L_{mm} \frac{dI_m}{dt} \tag{4.3}$$

with the mesh currents $I_m$ being relation to the individual branches $I_b$ by a transformation matrix, such that

$$\begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} \tag{4.4}$$

or

$$I_b = C^{b}_{.m} \, I_m \tag{4.5}$$

If power invariance is assumed between the branch and mesh reference frame [4], the impressed mesh voltage vector $E_m$ is

$$E_m = C^{.b}_{m} \, V_b \tag{4.6}$$

It was shown previously in section 2.1 that the mesh inductance and resistance matrices $L_{mm}$ and $R_{mm}$ may be determined from the corresponding branch matrices by

$$L_{mm} = C^{.b}_{m} \, L_{bb} \, C^{b}_{.m} \tag{4.7}$$

and

$$R_{mm} = C^{.b}_{m} \, R_{bb} \, C^{b}_{.m} \tag{4.8}$$

respectively, and using these results they may be calculated as

$$R_{mm} = \begin{bmatrix} (R_1 + R_3) & R_3 \\ R_3 & (R_2 + R_3) \end{bmatrix}$$

$$L_{mm} = \begin{bmatrix} (L_1 + L_3) & L_3 \\ L_3 & (L_2 + L_3) \end{bmatrix} \qquad (4.9)$$

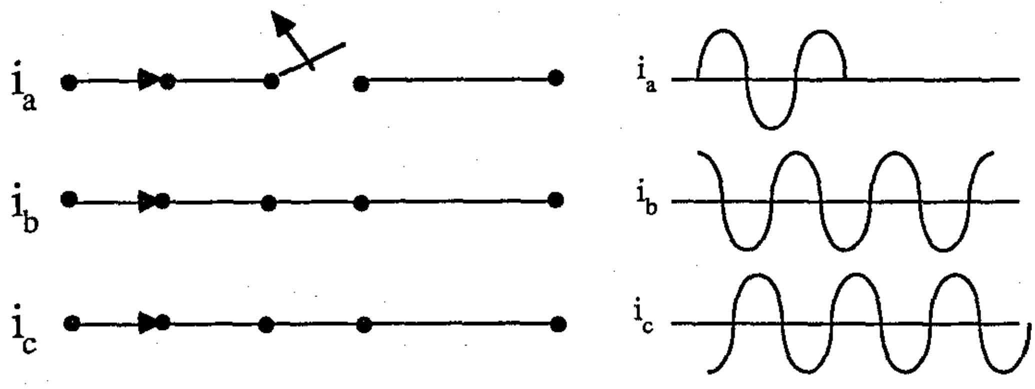
Kirchhoff's voltage law states that the sum of the voltage in a closed mesh is zero, or

$$V_m = 0 \qquad (4.10)$$

so that equation (4.3) becomes,

$$\frac{dI_m}{dt} = L_{mm}^{-1} \left[ E_m - R_{mm} I_m \right] \qquad (4.11)$$

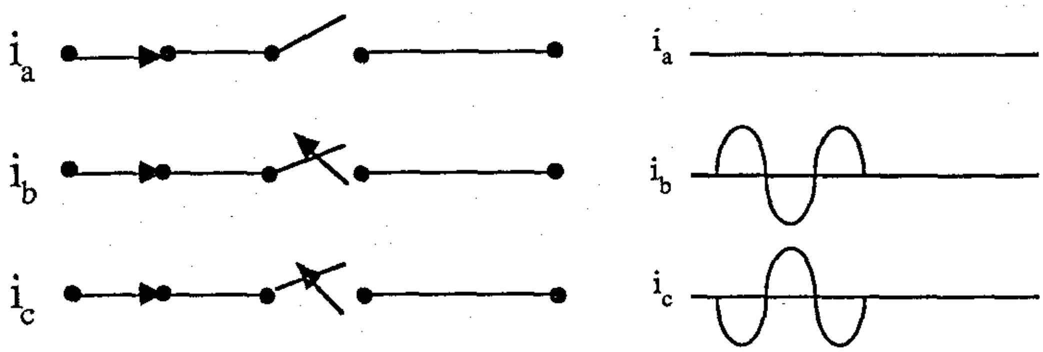
The elements of equation (4.11) may be assembled using equations (4.5) to (4.10) and the resulting equation may be solved using numerical integration.




Phase 'a' current zero

indicate the current in the three phases, with $i_a$ falling to zero

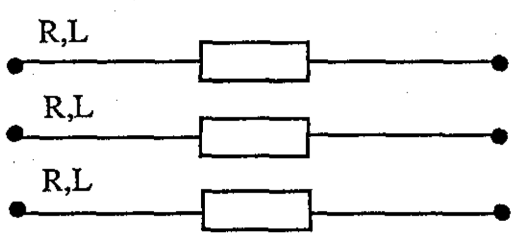
**Fig 4.1 Phase 'a' open circuit**




Phase 'b' and 'c' current zero

indicate $i_b$ and $i_c$ falling to zero

**Fig 4.2 Phase 'b' and 'c' open circuit**




Switch

**Fig 4.3 Switch closed representation**

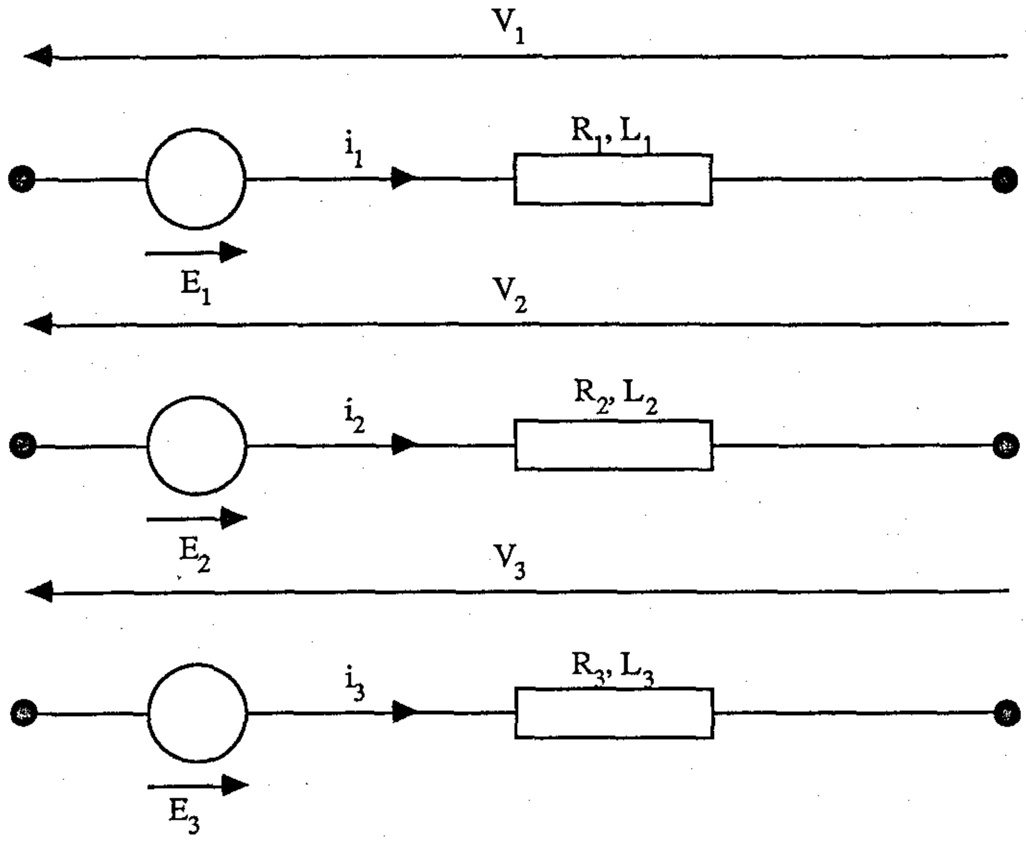



**Fig 4.4(a) Branch reference frame for the bus bar**

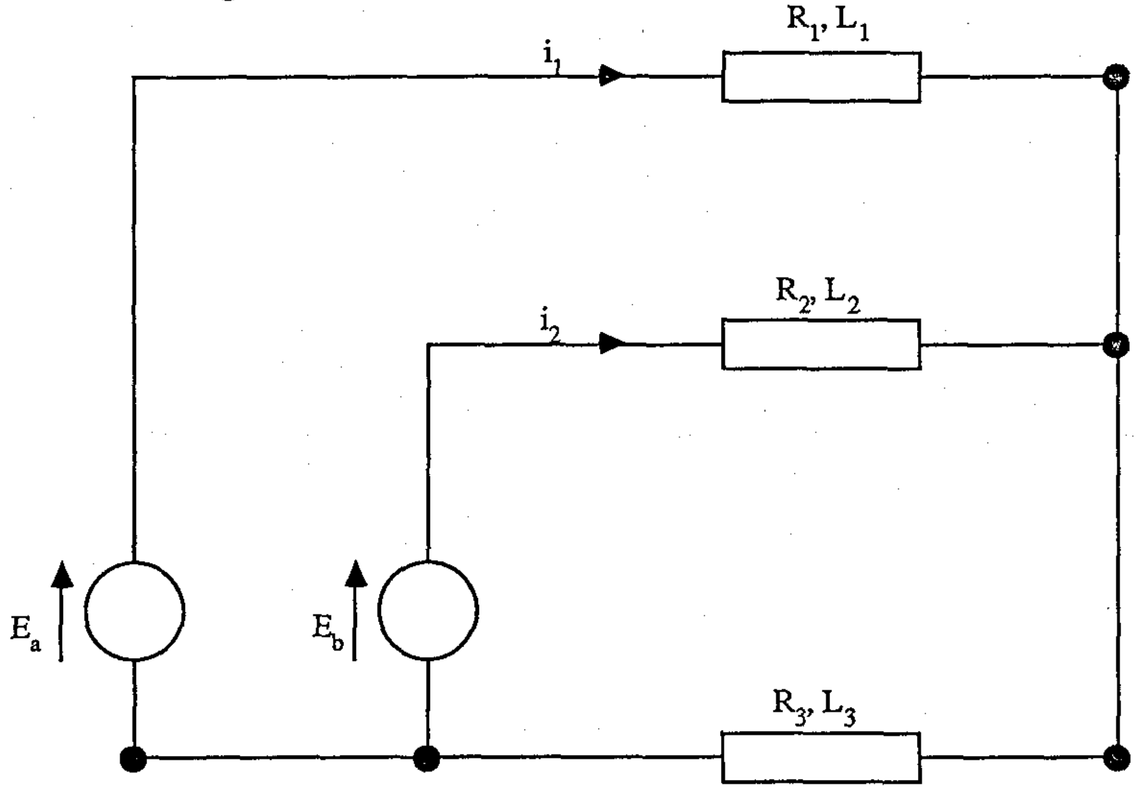



**Fig 4.4(b) Mesh reference frame for the bus bar**

Chapter 5

# SIMULATION OF A 3-PHASE FULL-WAVE THYRISTOR BRIDGE CONVERTER

Power conversion from AC to DC is often achieved by means of a static converter in the form of a thyristor bridge. This chapter describes a mathematical model for the analysis of the 3-phase bridge converter shown in Fig 5.1, in which each arm contains two back-to-back thyristor pairs. For example, the arm to phase 'r' contains thyristor pairs (1,7) and (4,10) and the arm to phase 'y' contains thyristor pairs (3,9) and (6,12). During rectification the power circuit becomes effectively as shown in Fig 5.2, with thyristors 7 to 12 switching sequentially to convert the 3-phase AC input to a DC output. During inversion, the power flow is reversed and the effective power circuit becomes as shown in Fig 5.3. Thyristors 1 to 6 are now fired in a PWM manner and thyristors 7 to 12 provide freewheeling paths. Sections 5.1 and 5.2 describe respectively mathematical models for both rectification and inversion modes of the thyristor bridge, and numercial results obtained from these models are presented at the end of each section.

The conduction pattern in the converter changes continually with time and for this reason tensor analysis is used to assemble and to solve automatically the system equations. The computer program which solves the converter equations is able to handle these change automatically, and also to assemble the required transformation matrix ($C_{\cdot m}^{b}$ ) relating the currents in the mesh reference frame to those in the branch reference frame. The resulting differential equation is solved using the 4th-order Runge Kutta method outlined in Appendix B.

## 5.1 Rectification

The circuit model for the converter operating as a rectifier is shown in Fig 5.2.

### 5.1.1 Rectifier meshes

The converter circuit may be specified by the 10 branch currents defined in Fig 5.2, with the various currents conveniently being categorised into three groups;

(i) $i_1$ in the converter DC side mesh,

(ii) $i_2$ to $i_4$ in the converter AC side meshes,

(iii) $i_5$ to $i_{10}$ in the meshes including the thyristors 7 to 12.

The branch/mesh current transformation tensor $C_{.m}^{b}$ for the rectifier is assembled from two master matrices. The first of these, Cmast, is shown in Table 5.1 and defines the six meshes which each contain one thyristor from the top row and one from the bottom row of the bridge. As the various thyristors become both sequentially forward biased and triggered, the relevant column from this matrix is selected and loaded into $C_{.m}^{b}$ .

During the transfer of current from one thyristor to another, the commutation mesh which is established is defined by the second master matrix Cmin in Table 5.2. In the situation when only thyristors 7 and 12 are conducting, $C_{.m}^{b}$ contains the first column of Cmast (mesh 1). When thyristor 8 is fired, the commutation mesh shown in Fig 5.4 is formed and the second column of Cmin (mesh 8) is added to $C_{.m}^{b}$ . At the end of the commutation interval, the current in thyristor 12 has reduced to zero, this device turns off and a new transformation matrix is formed from the second column of Cmast. In summary, two meshes are required during commutation but only one during normal conduction periods.

| Mesh / Branch | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | -1 | 0 | -1 | 0 |
| 3 | -1 | 0 | 1 | 1 | 0 | -1 |
| 4 | 0 | -1 | 0 | -1 | 1 | 1 |
| TH7 | 1 | 1 | 0 | 0 | 0 | 0 |
| TH9 | 0 | 0 | 1 | 1 | 0 | 0 |
| TH11 | 0 | 0 | 0 | 0 | 1 | 1 |
| TH10 | 0 | 0 | 1 | 0 | 1 | 0 |
| TH12 | 1 | 0 | 0 | 0 | 0 | 1 |
| TH8 | 0 | 1 | 0 | 1 | 0 | 0 |

Table 5.1 Normal conduction matrix Cmast for rectification

| Mesh / Branch | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | -1 | -1 | 0 | 1 |
| 3 | 0 | 1 | 1 | 0 | -1 | -1 |
| 4 | -1 | -1 | 0 | 1 | 1 | 0 |
| TH7 | 1 | 0 | -1 | 0 | 0 | 0 |
| TH9 | 0 | 0 | 1 | 0 | -1 | 0 |
| TH11 | -1 | 0 | 0 | 0 | 1 | 0 |
| TH10 | 0 | 0 | 0 | 1 | 0 | -1 |
| TH12 | 0 | -1 | 0 | 0 | 0 | 1 |
| TH8 | 0 | 1 | 0 | -1 | 0 | 0 |
| incoming thy | TH7 | TH8 | TH9 | TH10 | TH11 | TH12 |
| outoging thy | TH11 | TH12 | TH7 | TH8 | TH9 | TH10 |

Table 5.2 Commutation conduction matrix Cmin for rectification

## 5.1.2 Thyristor switching

a) Turn-on

The two conditions necessary to turn-on a thyristor are that the anode voltage is positive with respect to the cathode and that an adequate trigger pulse is applied to the gate. Thyristor turn-on occurs in the mathematical model at the start of an integration step following the one in which both of these conditions have been achieved.

b) Turn-off

Thyristor turn off occurs when the anode current falls below the holding value, which is normally sufficiently small to be regarded as zero. The time to a turn-off discontinuity from the start of an integration step is then determined by linear interpolation as shown in Fig 5.5, such that

$$t_d = \frac{i_{TO}}{i_{TO} - i_T} h \qquad (5.1)$$

where $i_{TO}$ is the current at the beginning of the integration step,

$i_T$ is the predicted current at the end of the integration step,

and $h$ is the integration step length.

## 5.1.3 Computer algorithm for the rectifier model

### 5.1.3.1 Conduction states

The conduction state of each arm of the bridge is recorded in a single dimensional array having three elements, icirc, with the three possible conduction states of each arm being considered below.

(i) Normal conduction period

The array icirc records the identification number of the conducting thyristors in each arm. If there is no conducting thyristor, a zero is recorded in the corresponding location of the array. As an example, if thyristors 7 and 12 are conducting the three entries in icirc are (7,12,0).

(ii) Start of commutation

During commutation, the location in icirc which corresponds to the incoming thyristor records a number which is larger by 6 than the identification number of the incoming thyristor. Thus when thyristors 7 and 12 are conducting, and thyristor 8 is triggered causing thyristor 12 to commutate off, the entries in icirc are (7,12,14).

(iii) End of commutation

At the end of commutation, the entry in icirc that corresponds to the incoming thyristor is reduced by 6, to identify the newly fired device. The entry in icirc which corresponds to the outgoing thyristor is set to zero, so that the entries in icirc, at the end of the commutation conditions specified in (ii) are (7,0,8).

By repeating steps (i) to (iii), the computer records the conduction states of all three arms of the converter bridge.

### 5.1.3.2 Conduction meshes

In the computer program, a composite condition matrix Crec which contains the twelve possible conduction meshes in the rectifier is defined as in Table 5.3. Each mesh contains two conducting thyristors, so that, for example, if thyristors 7 and 12 are conducting column 1 of Crec is chosen. The twelve possible conduction meshes for the rectifier are represented in Table 5.4; with each column in the table giving the conditions required for choosing a particular column from Crec.

| Branch \ Mesh | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 1 | -1 | 0 | -1 | 0 | 1 | 0 | -1 | -1 | 0 | 1 |
| 3 | -1 | 0 | 1 | 1 | 0 | -1 | 0 | 1 | 1 | 0 | -1 | -1 |
| 4 | 0 | -1 | 0 | -1 | 1 | 1 | -1 | -1 | 0 | 1 | 1 | 0 |
| TH7 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | -1 | 0 | 0 | 0 |
| TH9 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | -1 | 0 |
| TH11 | 0 | 0 | 0 | 0 | 1 | 1 | -1 | 0 | 0 | 0 | 1 | 0 |
| TH10 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | -1 |
| TH12 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | -1 | 0 | 0 | 0 | 1 |
| TH8 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | -1 | 0 | 0 |
| Converter mesh | 1 | 3 | 4 | 2 | 6 | 5 | 3 | 2 | 4 | 6 | 5 | 1 |

Table 5.3 Complete master matrix Crec for rectification

The computer compares the values stored in icirc with the conditions given in each column in Table 5.4, and a column in Crec is chosen only if the stored elements correspond with the conditions of that column. The selected columns of Crec are used to form the required transformation tensor $C_{\cdot m}^{b}$ as indicated below.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| First entry in icirc | 7 | 7 | 10 | X | 10 | X | 13 | X | 7 | 16 | X | 10 |
| Second entry in icirc | 12 | X | 9 | 9 | X | 12 | X | 12 | 15 | X | 9 | 18 |
| Third entry in icirc | X | 8 | X | 8 | 11 | 11 | 11 | 14 | X | 8 | 17 | X |
| Column chosen in Crec | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

where X denotes *"don't care"* .

Table 5.4 The twelve possible conditions for Crec

(i) Normal conducting period

During this period, the entries in icirc are (7,12,0). The computer compares these figures with the condition in each column of the table and chooses column 1 from Crec to form $C^{b}_{\cdot m}$ .

(ii) Start of commutation

The elements in icirc are now (7,12,14) and the comparison process again selects column 1. The eighth column of the table is also satisfied, since the second and third entries of icirc are 12 and 14, and column 8 of Crec is chosen. Columns 1 and 8 of Crec together form $C^{b}_{\cdot m}$ .

(iii) End of commutation

The entries in icirc are now (7,0,8) and column 2 in Crec is accordingly chosen to form $C^{b}_{\cdot m}$ .

In the program, a subroutine called recmesh holds the above table and determines the required columns in Crec during each period.

After $C_{\cdot m}^{b}$ has been formed, the tensor methods described in section 4.1 are used to produce the relevant meshes for the rectifier.

### 5.1.4 Converter performance in rectification mode

Simulated results for load voltage and current, AC side line currents, and line voltages are all shown in Fig 5.6 to 5.9 for various trigger angles between $0^{o}$ to $120^{o}$.

For trigger angles between $0^{o}$ and $60^{o}$ (See Fig 5.6 to 5.8) the load voltage is continuous. When the trigger angle is greater than $60^{0}$, the load voltage becomes discontinuous (See Fig 5.9), with the mean output voltage and current decreasing as the trigger angles increases from $0^{o}$ to $120^{o}$. At, the maximum trigger angle of $120^{o}$, no voltage and current are supplied to the load, since the thyristors are now reverse biased when the firing pulse is applied.

## 5.2 Inversion

During inversion, the current flow in Fig 5.3 may be defined in terms of the sixteen branches defined by the branch currents $i_1$ to $i_{16}$. Thyristors 1 to 6 are fired in a PWM manner, while thyristors 7 to 12 are fired to provide freewheeling paths as the current commutates between thyristors 1 to 6.

During inversion the synchronous generators 1 and 2 in Fig 1.1, are represented by suitable phase impedances inserted to act as inductive/resistive loads on the 3-phase system.

### 5.2.1 Pulse-Width Modulation

The pulse-width modulation technique is illustrated in Fig 5.10, whereby a comparison between three reference sine-waves and a high-frequency triangular carrier wave determines the firing instants for each thyristor and results in three trains of output pulses shifted $120^0$ with respect to one another. The reference waveform has a variable frequency which determines the frequency of the PWM waveform [7].

With the control as in Fig 5.10 one or other of the thyristors in each arm is conducting at all time, so connecting an output line to either the positive or negative side of the DC source. For example, consideration of Fig 5.3 shows that, if $i_2$ is positive, thyristor 1 is conducting and the corresponding output line is connected to the positive DC input. However, when thyristor 4 is fired, thyristor 1 turns off and the load current transfers to thyristor 10, which provides a freewheeling path. Similarly, the load current will transfer to thyristor 7 when thyristor 1 is fired.

### 5.2.2 Intersection of reference and carrier waveforms

The difference between the instantaneous values of the modulating and carrier waveforms is calculated and recorded at the beginning (zb) and the end (za) of each integration step. If a change in sign occurs, ie: $zb \times za < 0$, an intersection point exists, as shown in Fig 5.11(a) at a position determined by linear interpolation as

$$t_{poi} = \frac{\text{mod (zb)}}{\text{mod (zb)} + \text{mod (za)}} \, h \qquad (5.2)$$

It is possible that several points of intersection between the reference and carrier waveforms occur in a single step, as shown in Fig 5.11(b), in which case the smallest value of $t_{poi}$ chosen. Thus, in Fig 5.11(b), point A is chosen and $t_1$ used as the integration step length. The next step then ends at point B and $t_3$ is used as the step length.

### 5.2.3 Conduction patterns

The branch/mesh current transformation tensor $C_{\cdot m}^{b}$ for the inverter is assembled from three master matrices,

(i) Cm1 which contains the normal conduction meshes,

(ii) Cm2 which contains the commutation meshes with one freewheeling thyristor,

(iii) Cm3 which contains the commutation meshes with two freewheeling thyristors.

These three master matrix are described in the following sections.

#### (a) Normal Conduction Meshes

Fig 5.12 shows the mesh formed when thyristors 1 and 6 are conducting. Five further meshes are formed by other thyristor pairs, as given in Table 5.5 below,

| **Mesh** | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **Thyristor no. (top row)** | 1 | 1 | 3 | 3 | 5 | 5 |
| **Thyristor no. (bottom row)** | 6 | 2 | 2 | 4 | 4 | 6 |

Table 5.5 The six normal conduction meshes

The master matrix Cm1 relating to the above six normal conduction meshes is shown in Table 5.6.

| Branch \ Mesh | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | -1 | 0 | -1 | 0 |
| 3 | -1 | 0 | 1 | 1 | 0 | -1 |
| 4 | 0 | -1 | 0 | -1 | 1 | 1 |
| TH1 | 1 | 1 | 0 | 0 | 0 | 0 |
| TH3 | 0 | 0 | 1 | 1 | 0 | 0 |
| TH5 | 0 | 0 | 0 | 0 | 1 | 1 |
| TH4 | 0 | 0 | 1 | 0 | 1 | 0 |
| TH6 | 1 | 0 | 0 | 0 | 0 | 1 |
| TH2 | 0 | 1 | 0 | 1 | 0 | 0 |
| TH7 | 0 | 0 | 0 | 0 | 0 | 0 |
| TH8 | 0 | 0 | 0 | 0 | 0 | 0 |
| TH9 | 0 | 0 | 0 | 0 | 0 | 0 |
| TH10 | 0 | 0 | 0 | 0 | 0 | 0 |
| TH11 | 0 | 0 | 0 | 0 | 0 | 0 |
| TH12 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 5.6 The master matrix Cm1 for the inverter

(b) Freewheeling paths

Freewheeling paths are provided by the rectifier thyristors, as the current commutates between the inverter thyristors. Fig 5.13(a) shows the situation when thyristors 1, 6 and 2 are conducting. When thyristor 6 is turned off, freewheeling thyristor 9 is triggered to provide a path for current flow, as shown in Fig 5.13(b). After a predetermined delay, during which time device 6 has turned off, thyristor 3 is turned on.

There are twelve possible freewheeling meshes, formed by freewheeling thyristors, as defined in Table 5.7 and the master matrix Cm2 relating to the above twelve freewheeling meshes is shown in Table 5.8.

(c) Freewheeling paths with two thyristors

Consider the situation when thyristor 9 is conducting, thyristor 1 is off and the freewheeling thyristor 10 is triggered before the current in device 9 has reduced to zero. Two freewheeling thyristors are conducting simultaneously and two freewheeling meshes are consequently formed as shown in Fig 5.14.

There are now six possible freewheeling meshes, formed by other pairs of freewheeling thyristors, as defined in Table 5.9 and the master matrix Cm3 relating to the above six freewheeling possibilites is shown in Table 5.10.

In the computer program Cm1, Cm2 and Cm3 are combined in the composite condition matrix Cmod, which contains all the 24 possible conduction meshes of the inverter and is shown in Table 5.11.

| Mesh | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Freewheeling Thyristor** | 7 | 7 | 8 | 8 | 9 | 9 | 10 | 10 | 11 | 11 | 12 | 12 |
| **Inverter Thyristor** | 3 | 5 | 4 | 6 | 1 | 5 | 2 | 6 | 1 | 3 | 2 | 4 |

Table 5.7 The twelve commutation meshes with one freewheeling thyristor

| Branch \ Mesh | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | -1 | 0 | 1 | 1 | 0 | -1 | -1 | -1 | 0 | 1 | 1 | 0 |
| 3 | 0 | -1 | -1 | 0 | 1 | 1 | 1 | 0 | -1 | -1 | 0 | 1 |
| 4 | 1 | 1 | 0 | -1 | -1 | 0 | 0 | 1 | 1 | 0 | -1 | -1 |
| TH1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| TH3 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| TH5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| TH4 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| TH6 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| TH2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| TH7 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| TH8 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| TH9 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| TH10 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| TH11 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| TH12 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |

Table 5.8 The master matrix Cm2 for the inverter

| Mesh | 19 | 20 | 21 | 22 | 23 | 24 |
|---|---|---|---|---|---|---|
| **Freewheeling Thyristor (top row)** | 7 | 7 | 9 | 9 | 11 | 11 |
| **Freewheeling Thyristor (bottom row)** | 12 | 8 | 10 | 8 | 10 | 12 |

Table 5.9 The six commutation meshes with two freewheeling thyristors

| Branch \ Mesh | 19 | 20 | 21 | 22 | 23 | 24 |
|---|---|---|---|---|---|---|
| 1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 2 | -1 | -1 | 1 | 0 | 1 | 0 |
| 3 | 1 | 0 | -1 | -1 | 0 | 1 |
| 4 | 0 | 1 | 0 | 1 | -1 | -1 |
| TH1 | 0 | 0 | 0 | 0 | 0 | 0 |
| TH3 | 0 | 0 | 0 | 0 | 0 | 0 |
| TH5 | 0 | 0 | 0 | 0 | 0 | 0 |
| TH4 | 0 | 0 | 0 | 0 | 0 | 0 |
| TH6 | 0 | 0 | 0 | 0 | 0 | 0 |
| TH2 | 0 | 0 | 0 | 0 | 0 | 0 |
| TH7 | 1 | 1 | 0 | 0 | 0 | 0 |
| TH8 | 0 | 1 | 0 | 1 | 0 | 0 |
| TH9 | 0 | 0 | 1 | 1 | 0 | 0 |
| TH10 | 0 | 0 | 1 | 0 | 1 | 0 |
| TH11 | 0 | 0 | 0 | 0 | 1 | 1 |
| TH12 | 1 | 0 | 0 | 0 | 0 | 1 |

Table 5.10 The master matrix Cm3 for the inverter

| Mesh / Branch | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | -1 | -1 | -1 | -1 | -1 |
| 2 | 1 | 1 | -1 | 0 | -1 | 0 | -1 | 0 | 1 | 1 | 0 | -1 | -1 | -1 | 0 | 1 | 1 | 0 | -1 | -1 | 1 | 0 | 1 | 0 |
| 3 | -1 | 0 | 1 | 1 | 0 | -1 | 0 | -1 | -1 | 0 | 1 | 1 | 1 | 0 | -1 | -1 | 0 | 1 | 1 | 0 | -1 | -1 | 0 | 1 |
| 4 | 0 | -1 | 0 | -1 | 1 | 1 | 1 | 1 | 0 | -1 | -1 | 0 | 0 | 1 | 1 | 0 | -1 | -1 | 0 | 1 | 0 | 1 | -1 | -1 |
| TH1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| TH3 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| TH5 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| TH4 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| TH6 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| TH2 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| TH7 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| TH8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| TH9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| TH10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| TH11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| TH12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| Converter mesh | 1 | 3 | 4 | 2 | 6 | 5 | 4 | 6 | 6 | 5 | 1 | 5 | 3 | 1 | 3 | 2 | 2 | 4 | 4 | 6 | 1 | 5 | 3 | 2 |

Table 5.11 The complete master matrix Cmod

### 5.2.4 Computer implementation of the inverter model

The algorithm is basically the same as that for the rectifier program and the computer implementation is illustrated below by an example.

#### 5.2.4.1 Thyristor numbering

The freewheeling thyristor labels are six higher than their corresponding inverter thyristor labels. For example, in Fig 5.3 the freewheeling thyristor relating to inverter thyristor 1 is labelled 7.

#### 5.2.4.2 Conduction state

The conduction state in each arm of the bridge is recorded in a single dimension array, icirc, having three elements with the three possible condition states of each arm being illustrated below.

(i) Normal conduction

The array icirc records respectively the conducting thyristor numbers in each arm of the bridge. Thus, if thyristors 1, 6 and 2 are conducting the entries in icirc are (1,6,2).

(ii) Start of freewheeling

The freewheeling thyristor number is recorded in its corresponding location in icirc. When thyristor 6 in the second arm is turned off, and its freewheeling thyristor 9 is triggered, the second column of icirc records the value of the freewheeling thyristor 9 and the entries in icirc are (1,9,2).

(iii) End of freewheeling

The location in icirc corresponding to the arm which contains the freewheeling thyristor is reduced by 6, to give the number of the newly fired inverter thyristor. Thus, at the end of freewheeling, the second column of icirc is reduced to 3 and the entries in icirc become (1,3,2).

### 5.2.4.3 Conduction meshes

The 24 possible conduction meshes for the inverter may be represented as in Table 5.12, with each row in the table giving the conditions required for choosing a column from Cmod. Two inverter meshes are formed at all times and the comparison between icirc and the table stops when two meshes in Cmod have been found. The operation is illustrated below, using the example of the previous section.

(i) Normal conduction states

The entries in icirc are (1,6,2) and for this condition columns 1 and 2 of Cmod are used to form $C_{\cdot m}^{b}$ .

(ii) Start of freewheeling

The entries in icirc are (1,9,2) and columns 2 and 11 of Cmod are used to form $C_{\cdot m}^{b}$ .

(iii) End of freewheeling

The entries in icirc are (1,3,2) and columns 2 and 4 of Cmod are used to form $C_{\cdot m}^{b}$ .

In the program a subroutine, called choose, holds the above table and determines the required columns of Cmod which form $C_{\cdot m}^{b}$ , after which tensor methods are used to produce the relevant inverter meshes.

| | First entry in icirc | Second entry in icirc | Third entry in icirc | Column chosen in Cmod |
|---|---|---|---|---|
| Normal conduction patterns | 1 | 6 | X | 1 |
| | 1 | X | 2 | 2 |
| | 4 | 3 | X | 3 |
| | X | 3 | 2 | 4 |
| | 4 | X | 5 | 5 |
| | X | 6 | 5 | 6 |
| One freewheeling thyristor conduction patterns | 7 | 3 | X | 7 |
| | 7 | X | 5 | 8 |
| | 4 | X | 8 | 9 |
| | X | 6 | 8 | 10 |
| | 1 | 9 | X | 11 |
| | X | 9 | 5 | 12 |
| | 10 | X | 2 | 13 |
| | 10 | 6 | X | 14 |
| | 1 | X | 11 | 15 |
| | X | 3 | 11 | 16 |
| | X | 12 | 2 | 17 |
| | 4 | 12 | X | 18 |
| Two thyristor conduction patterns | 7 | 12 | X | 19 |
| | 7 | X | 8 | 20 |
| | 10 | 9 | X | 21 |
| | X | 9 | 8 | 22 |
| | 10 | X | 11 | 23 |
| | X | 12 | 11 | 24 |

where X denotes *"don't care"* .

Table 5.12 The 24 possible conditions for Cmod

### 5.2.5 Converter performance in inversion mode

To test the performance of the converter during the inversion mode, rated load at 0.9pf lag was applied suddenly to the AC side output. The converter parameters are:

DC input voltage = 400V

Frequency of reference wave = 40Hz

Amplitude of reference wave = 5V

Amplitude of carrier wave = 10V

The results of simulations for carrier frequencies of 800Hz and 2kHz respectively are shown in Fig 5.15 and 5.16. In both case, three PWM voltage waveforms with an amplitude of 400V and a mutual phase shift of $120^0$ are obtained. For a fixed reference frequency, the number of pulses per half cycle increases and the pulse width reduces as the carrier frequencies increases (see Figs 5.15 and 5.16). The process reduces the harmonic content in the output voltage [8], showing that a better-quality output with less harmonics is obtained by increasing the carrier frequency.

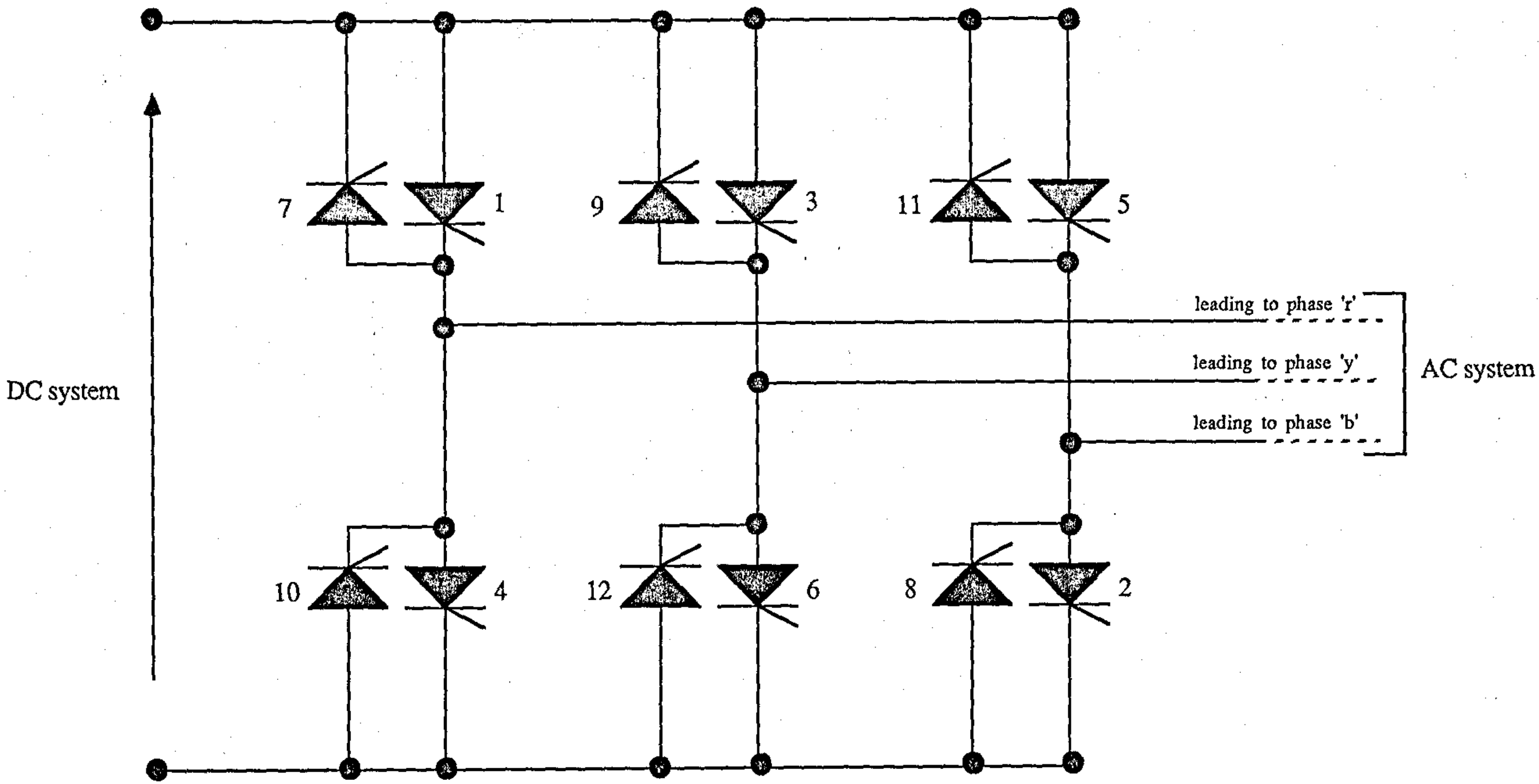



**Fig 5.1 Converter with back-to-back thyristor pairs**

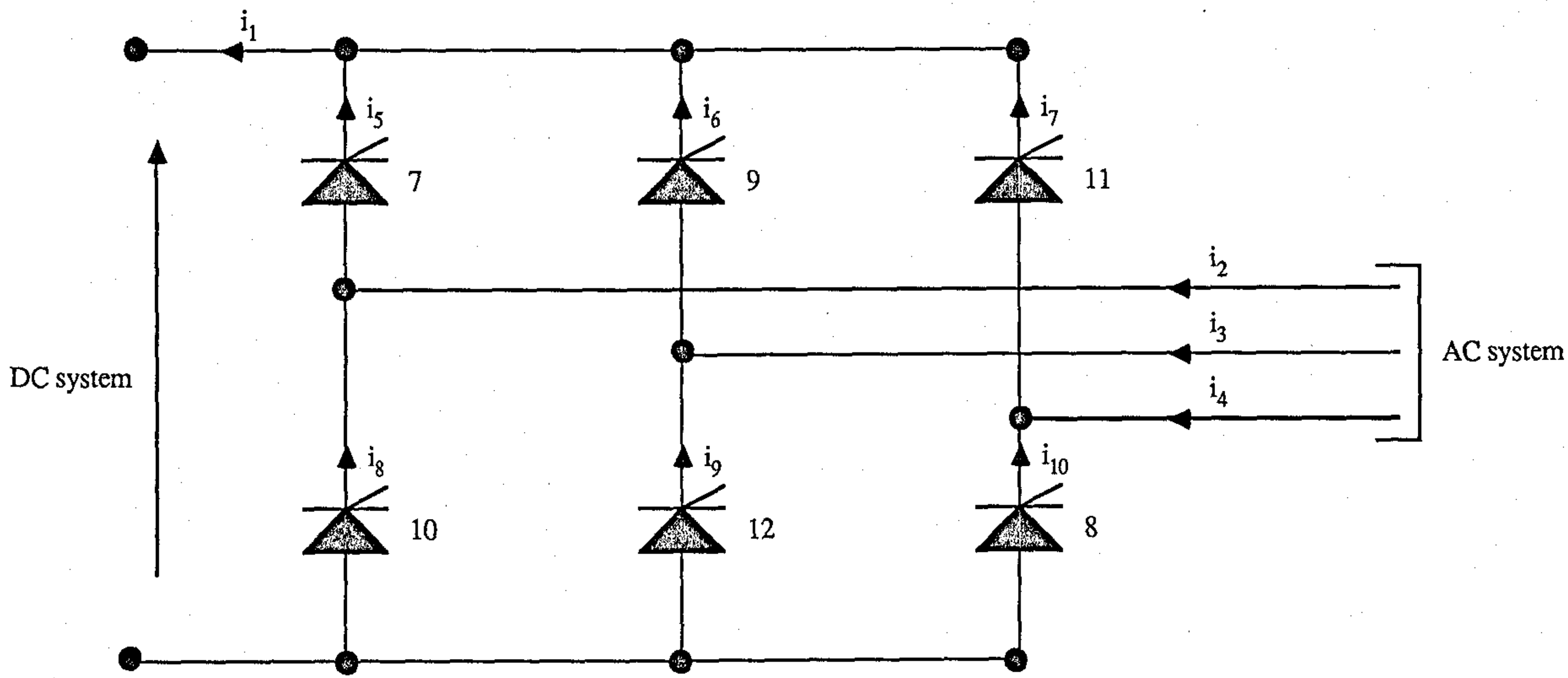



**Fig 5.2 The converter circuit, showing the current flow during rectification**

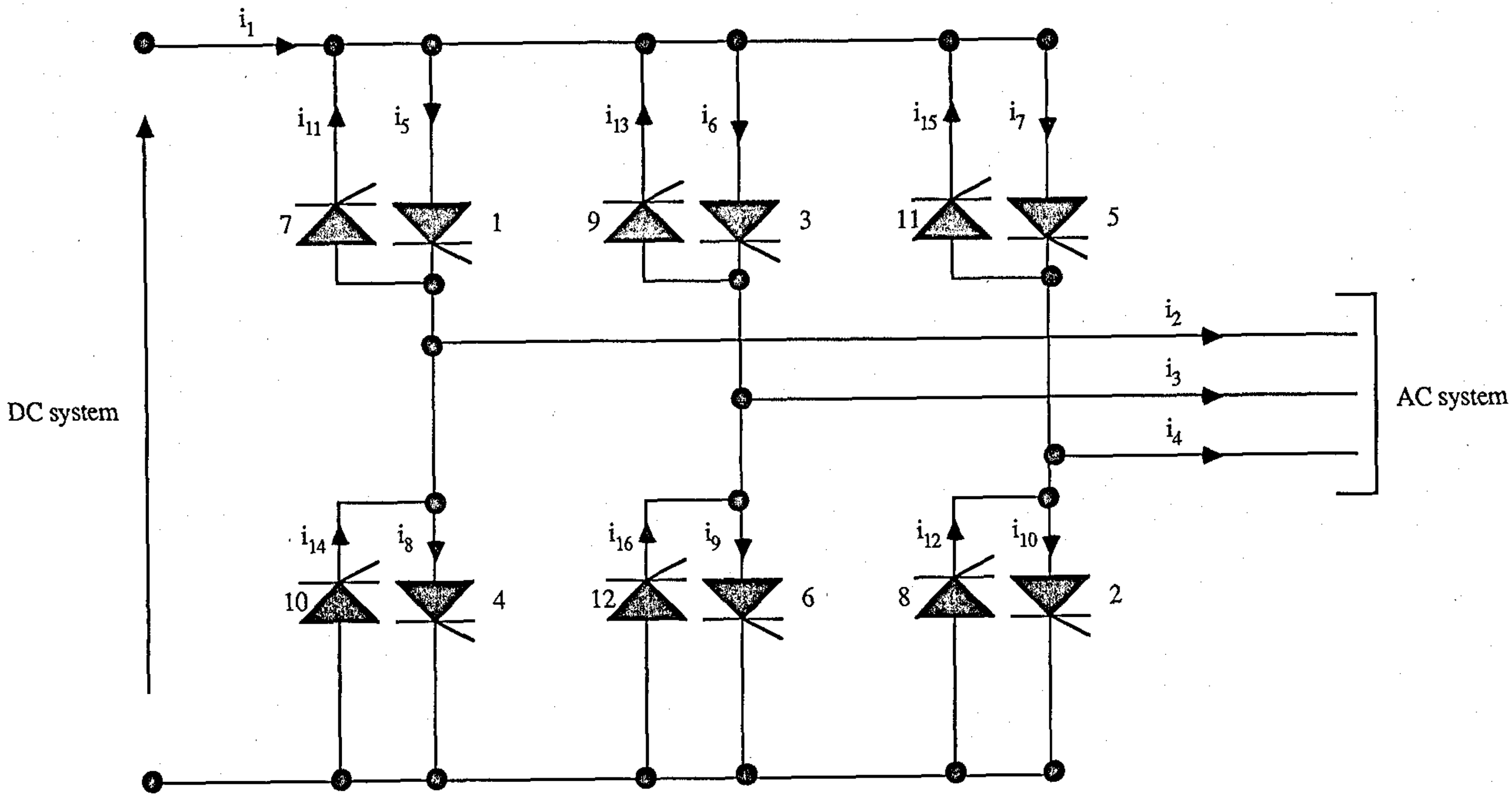



**Fig 5.3 The converter circuit, showing the current flow during inversion**

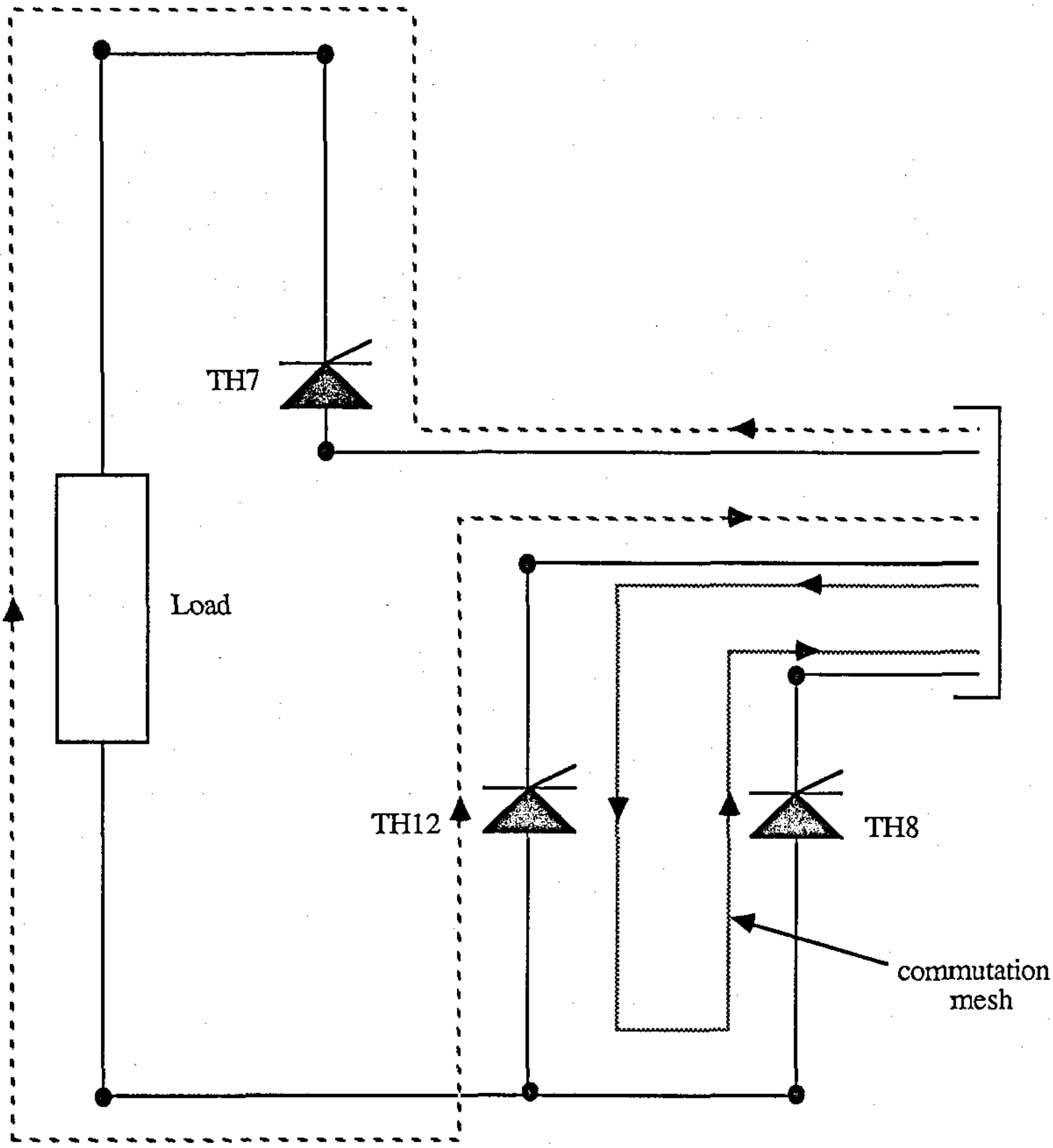



**Fig 5.4 Commutation mesh formed on firing thyristor 2**

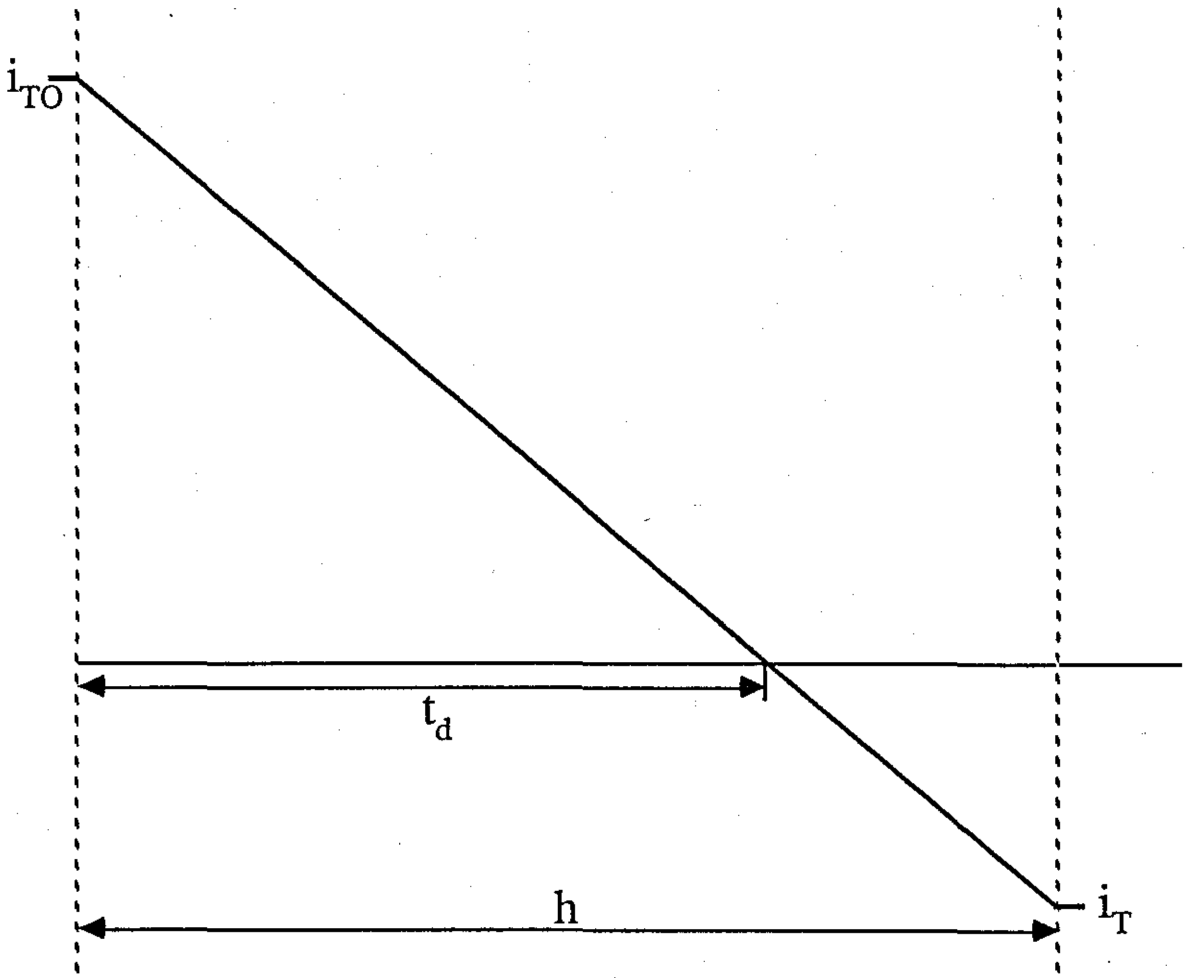



**Fig 5.5 The time to a turn-off discontinuity**

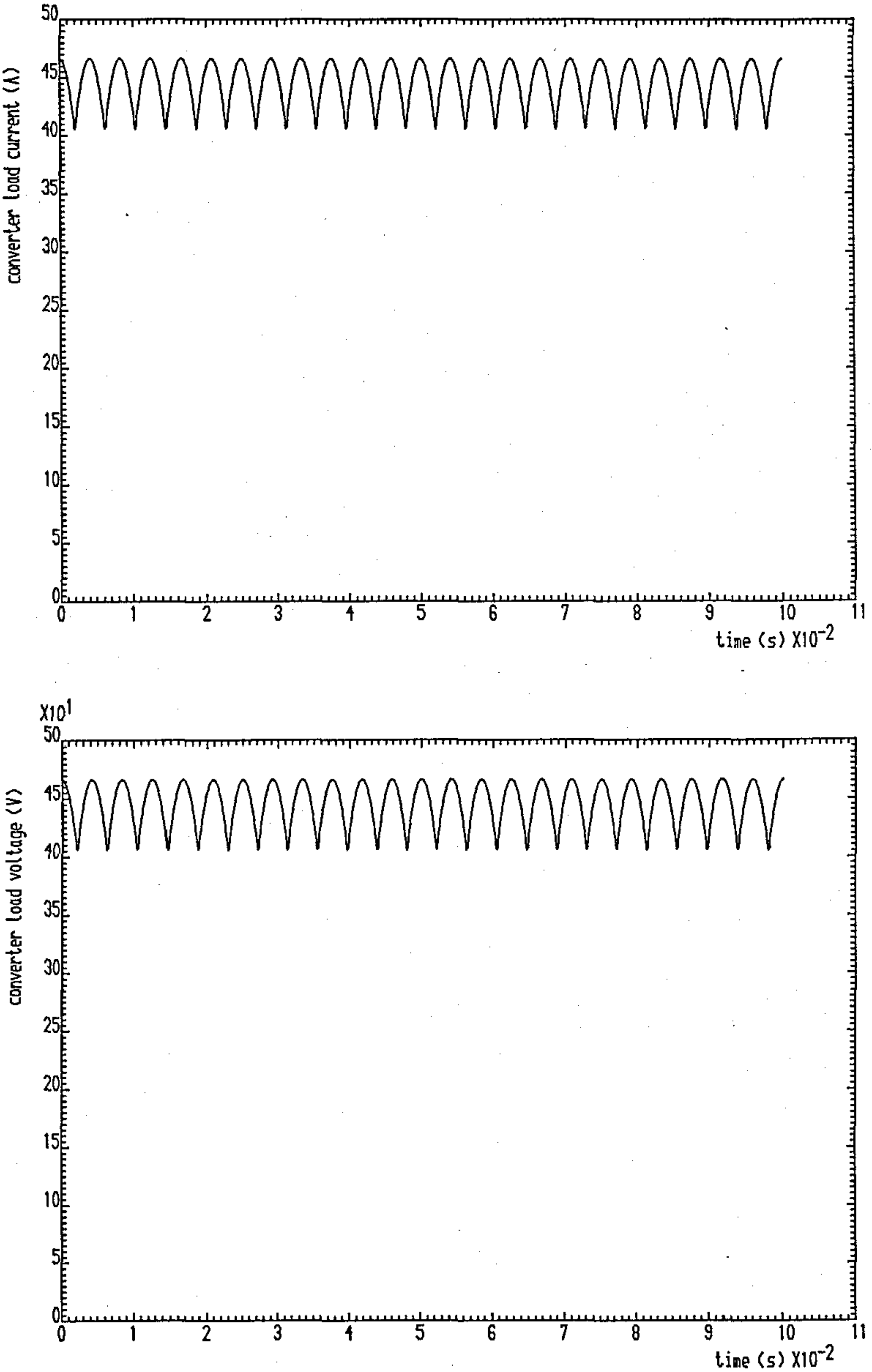



Fig 5.6(a) Converter load current and voltage with zero degree trigger angle

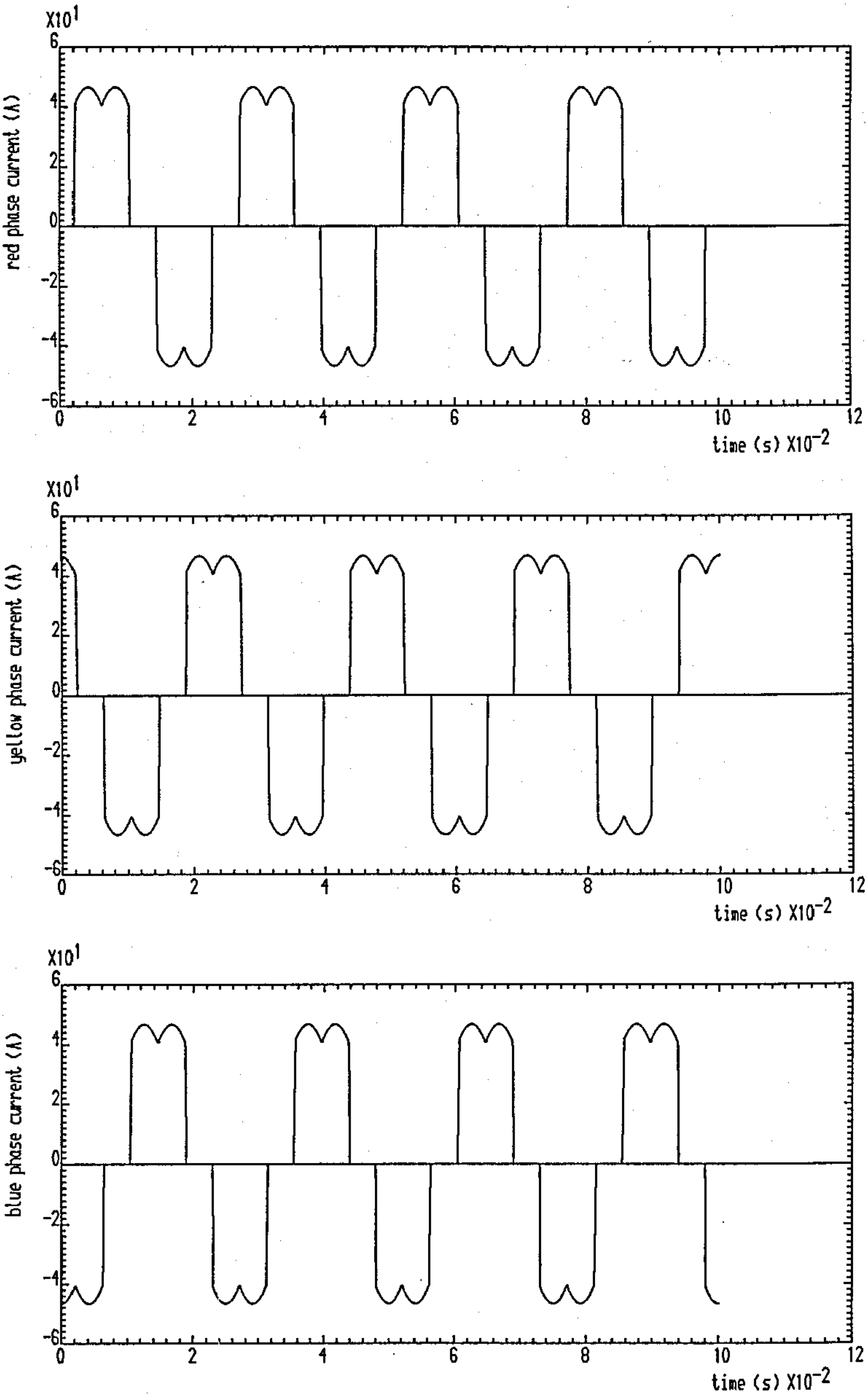



Fig 5.6(b) Converter AC side line currents with zero degree trigger angle

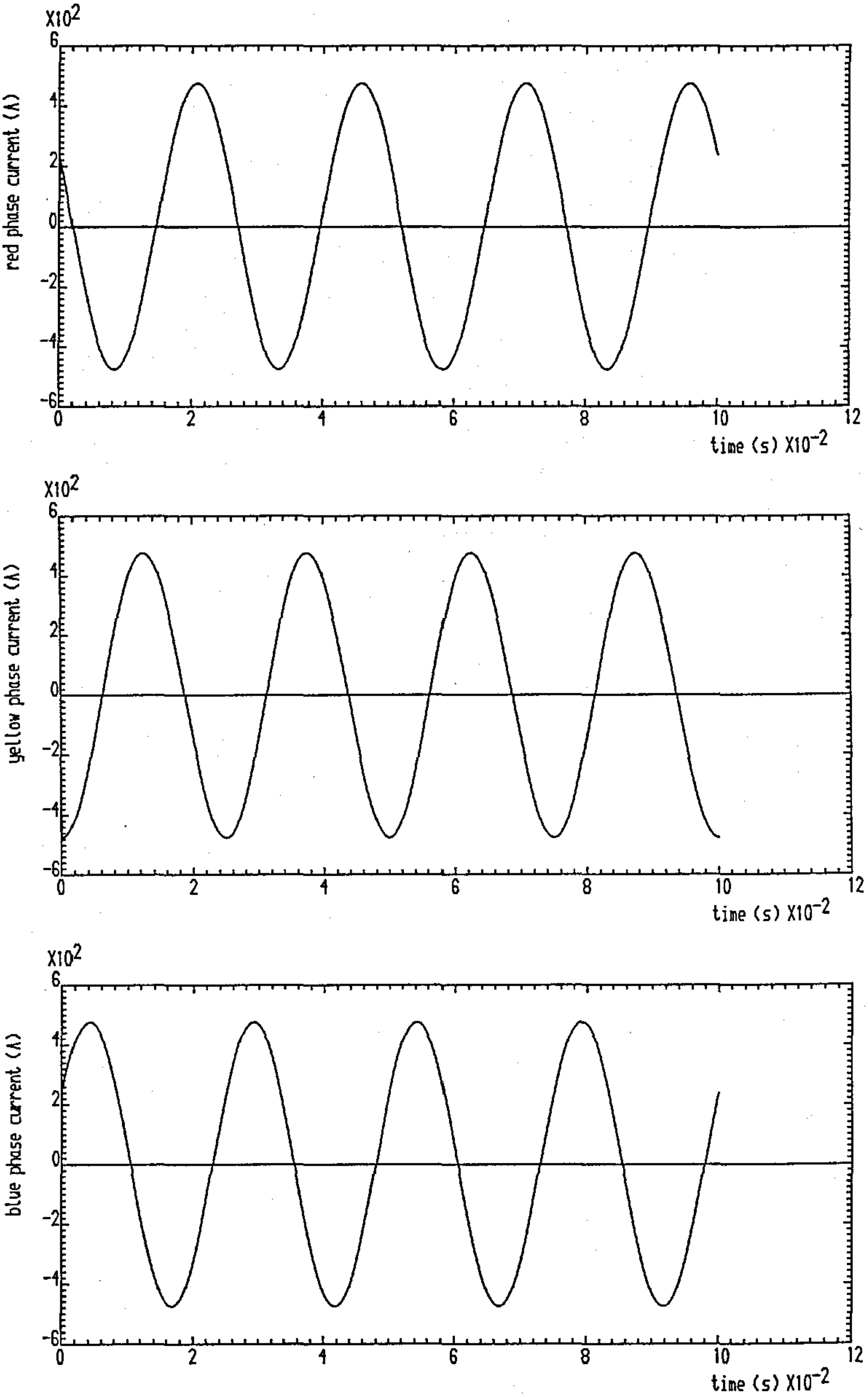



Fig 5.6(c) Converter AC side line voltages with zero degree trigger angle

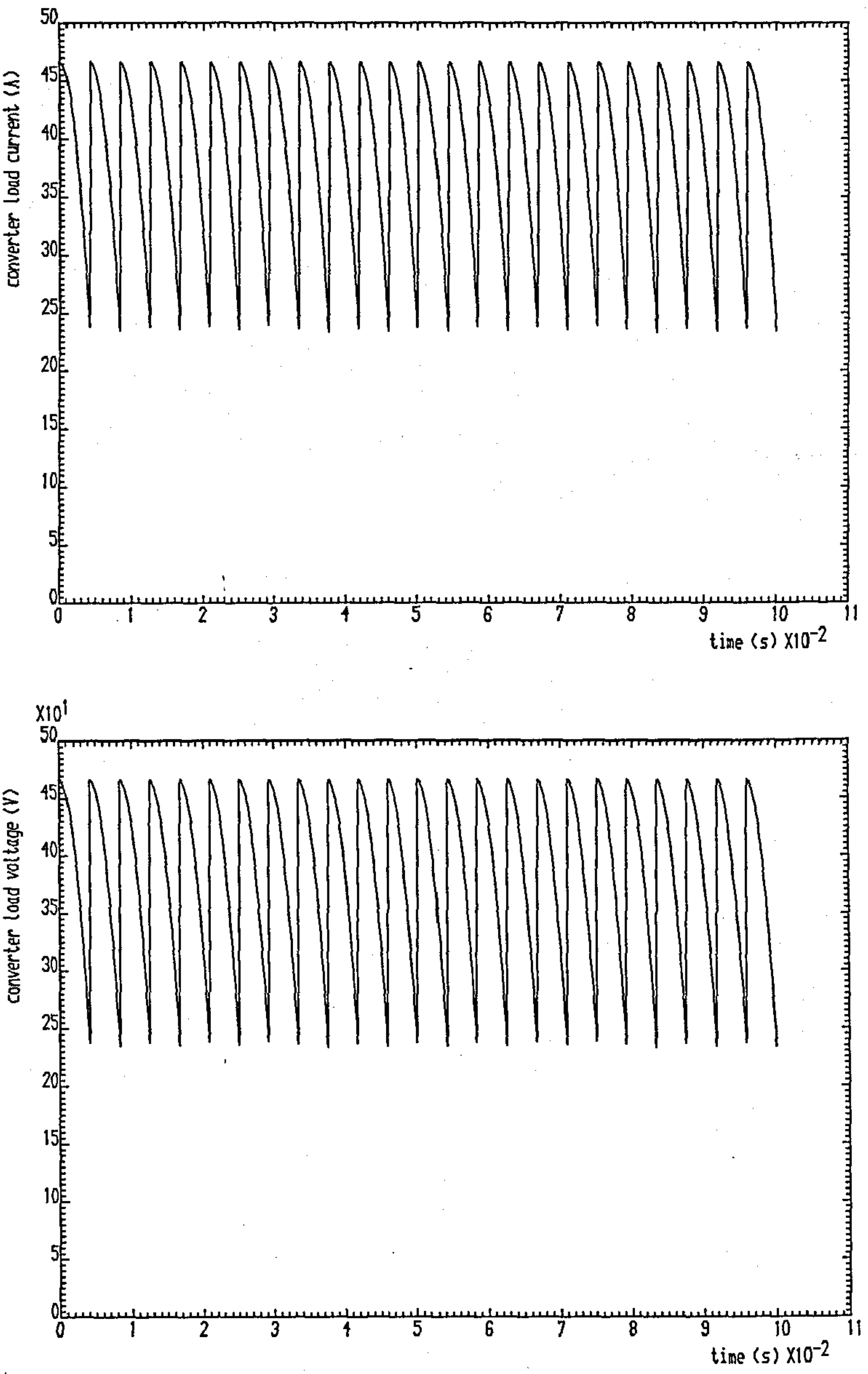



Fig 5.7(a) Converter load current and voltage with 30 degrees trigger angle

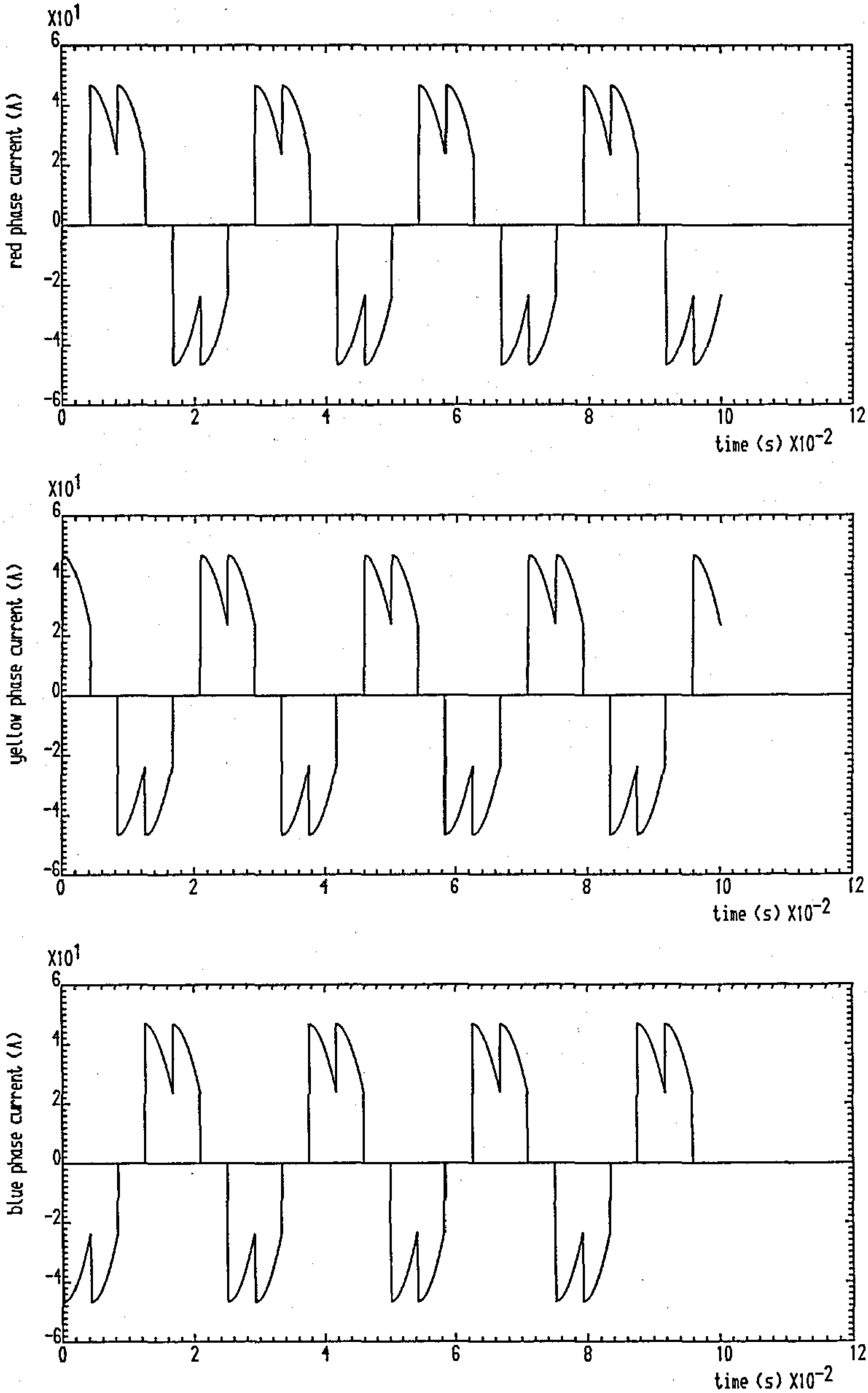



Fig 5.7(b) Converter AC side line currents with 30 degrees trigger angle

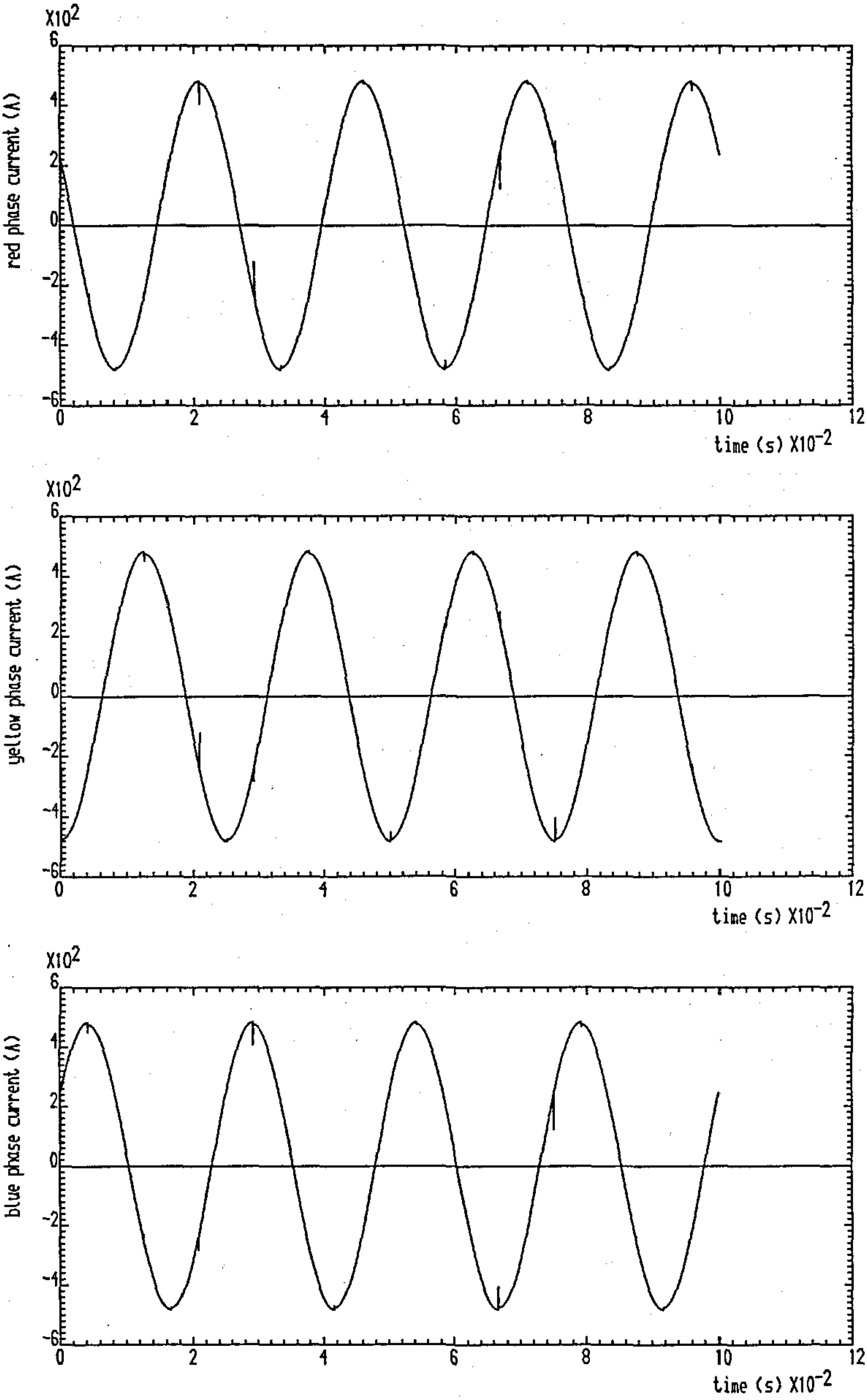
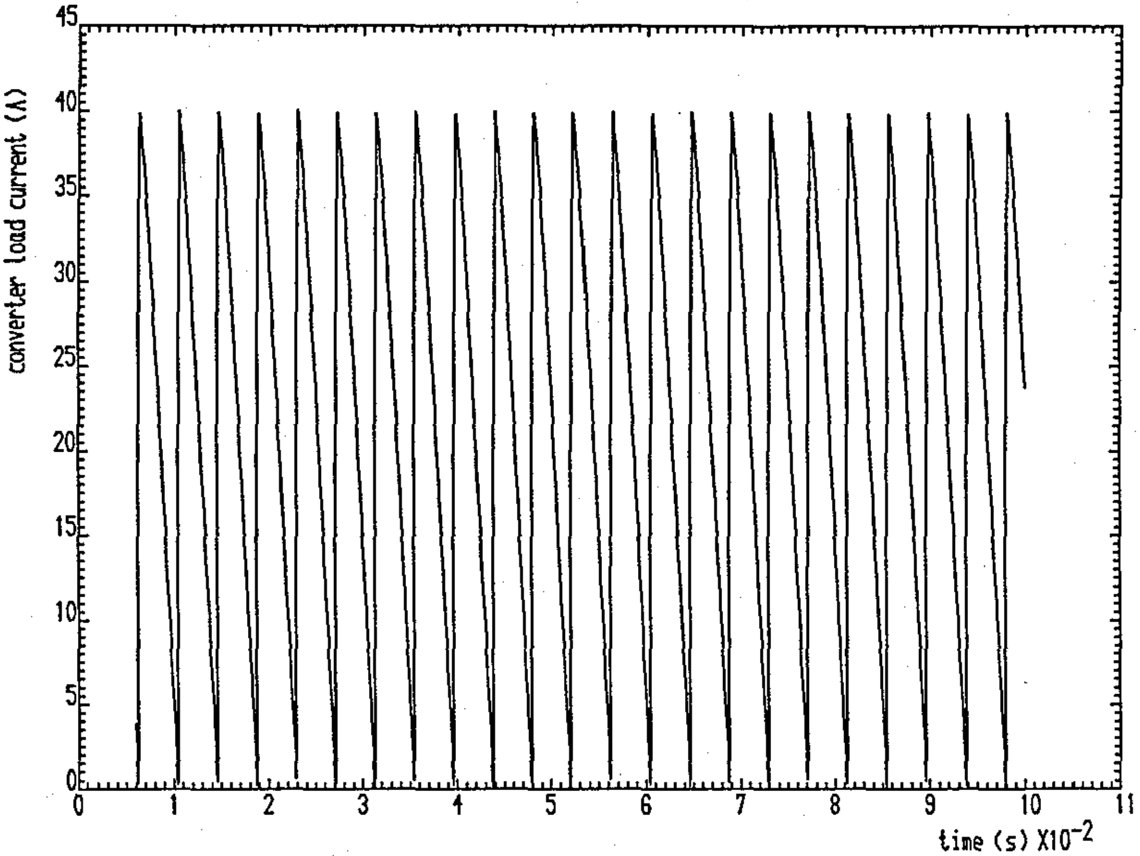



Fig 5.7(c) Converter AC side line voltages with 30 degrees trigger angle



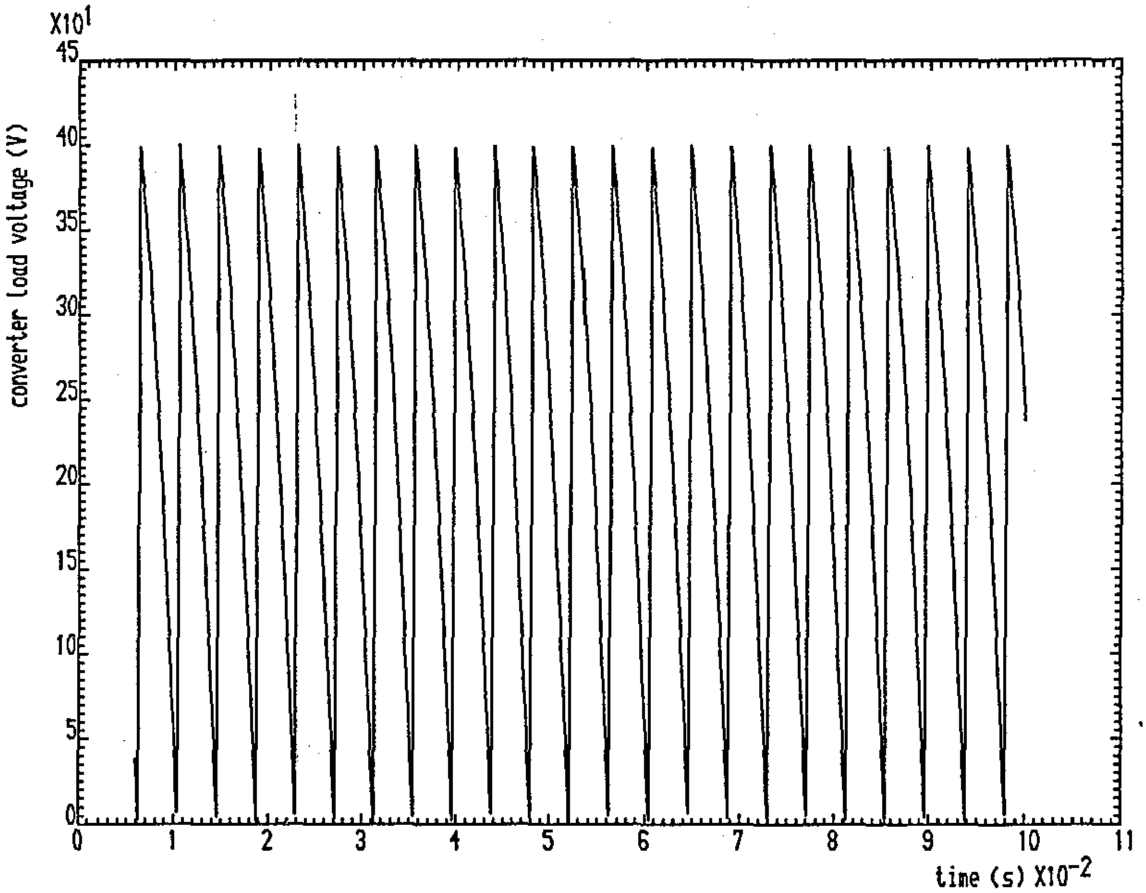




Fig 5.8(a) Converter load current and voltage with 60 degrees trigger angle

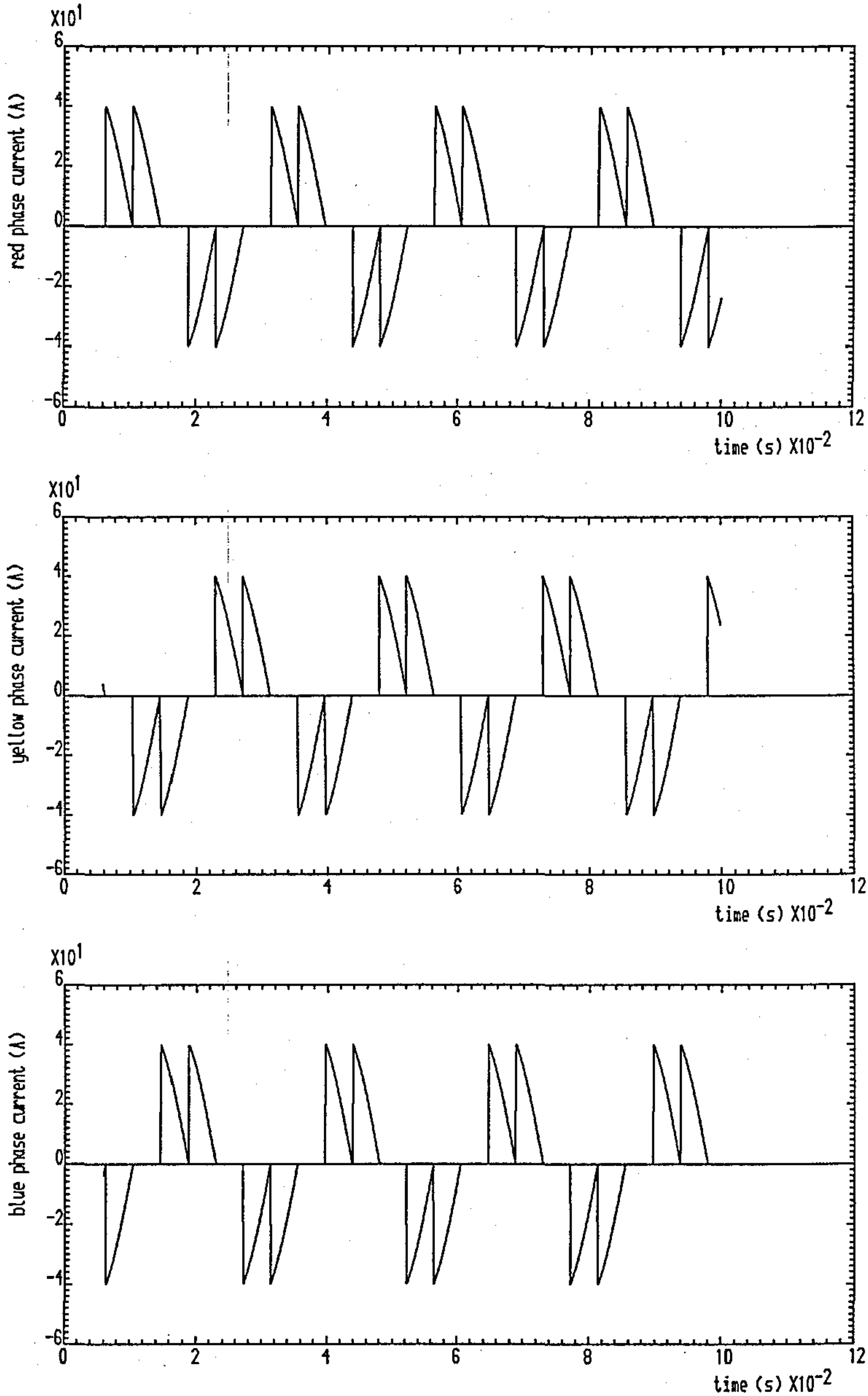



Fig 5.8(b) Converter AC side line currents with 60 degrees trigger angle

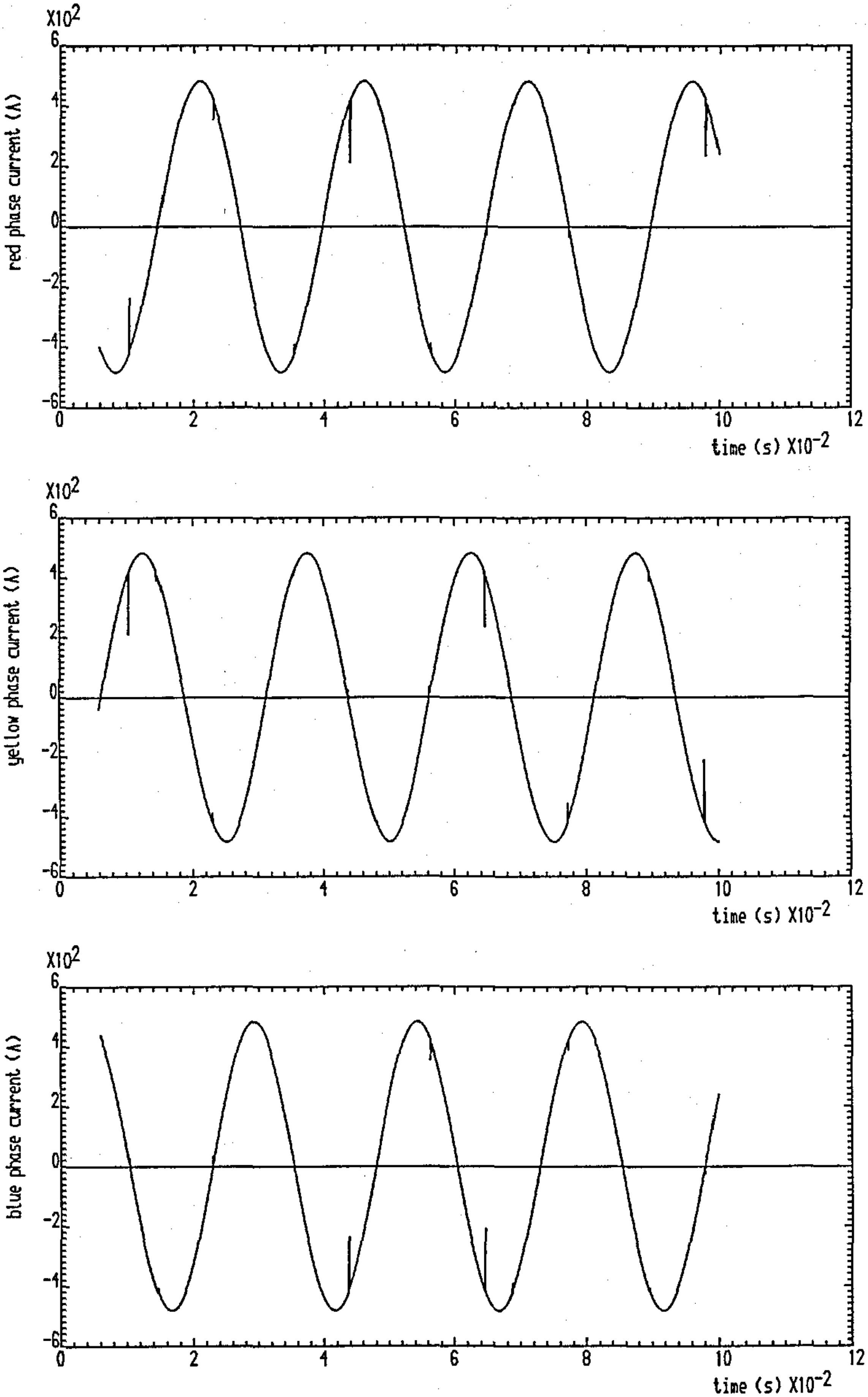



Fig 5.8(c) Converter AC side line voltages with 60 degrees trigger angle

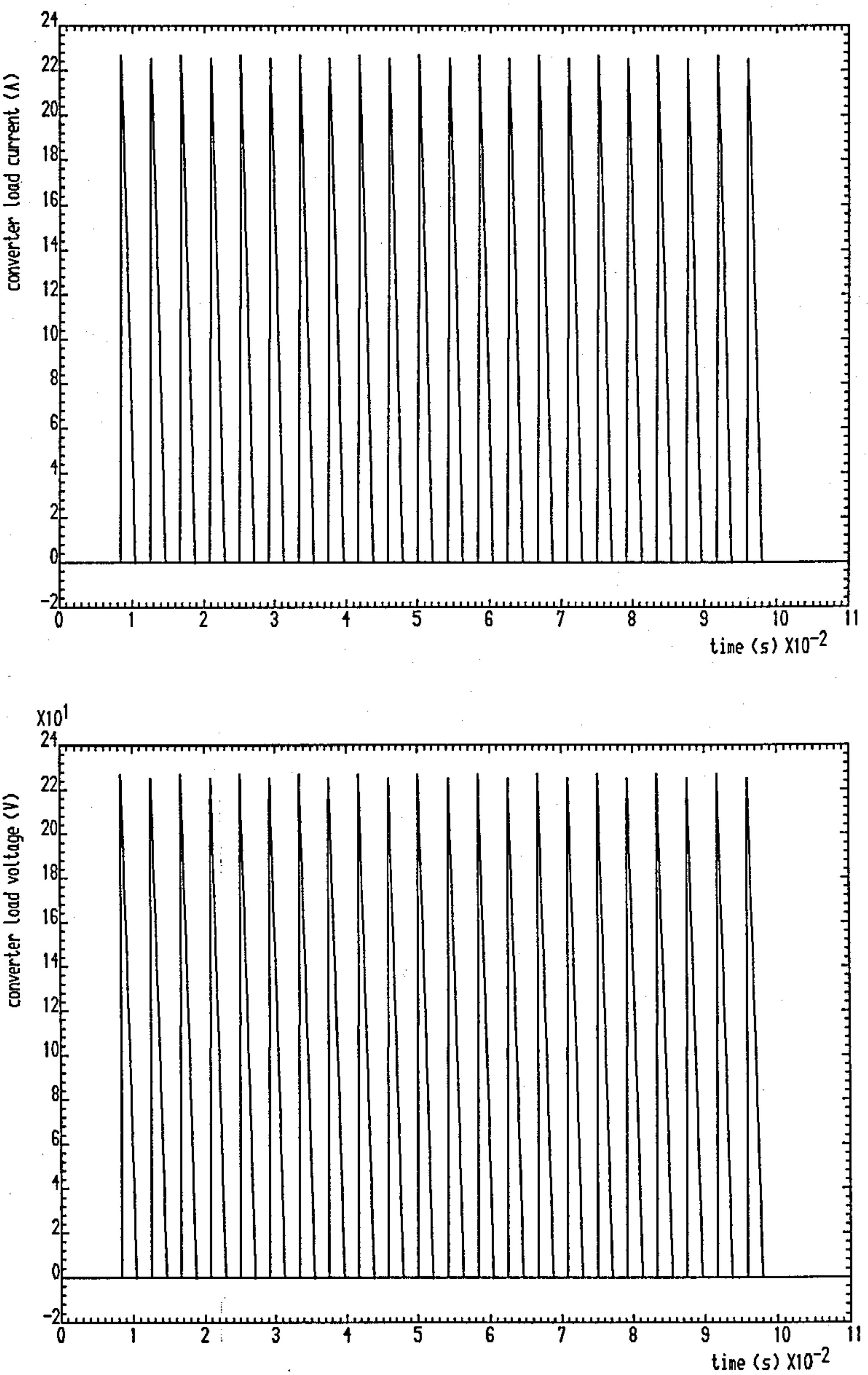



Fig 5.9(a) Converter load current and voltage with 90 degrees trigger angle

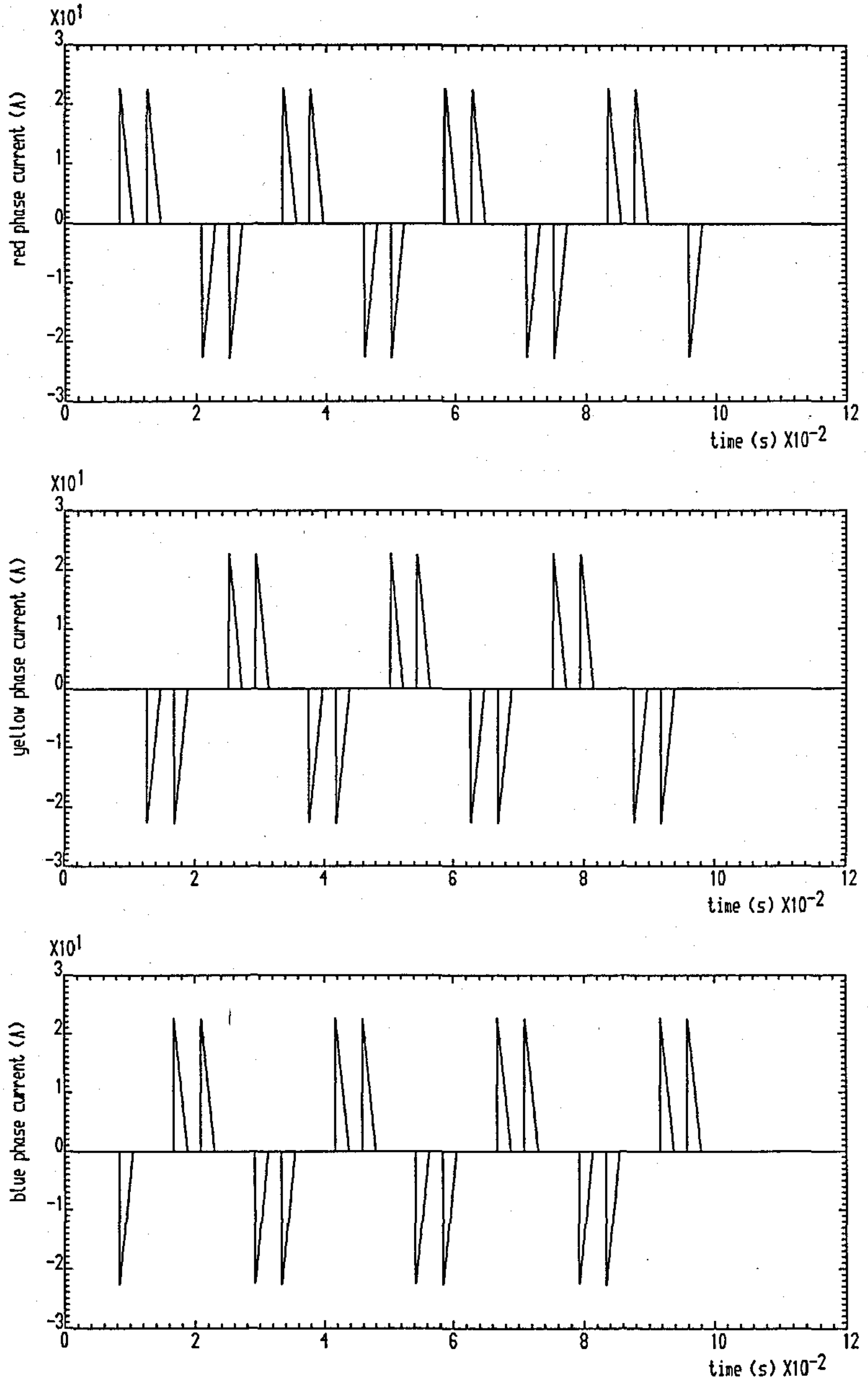



Fig 5.9(b) Converter AC side line currents with 90 degrees trigger angle

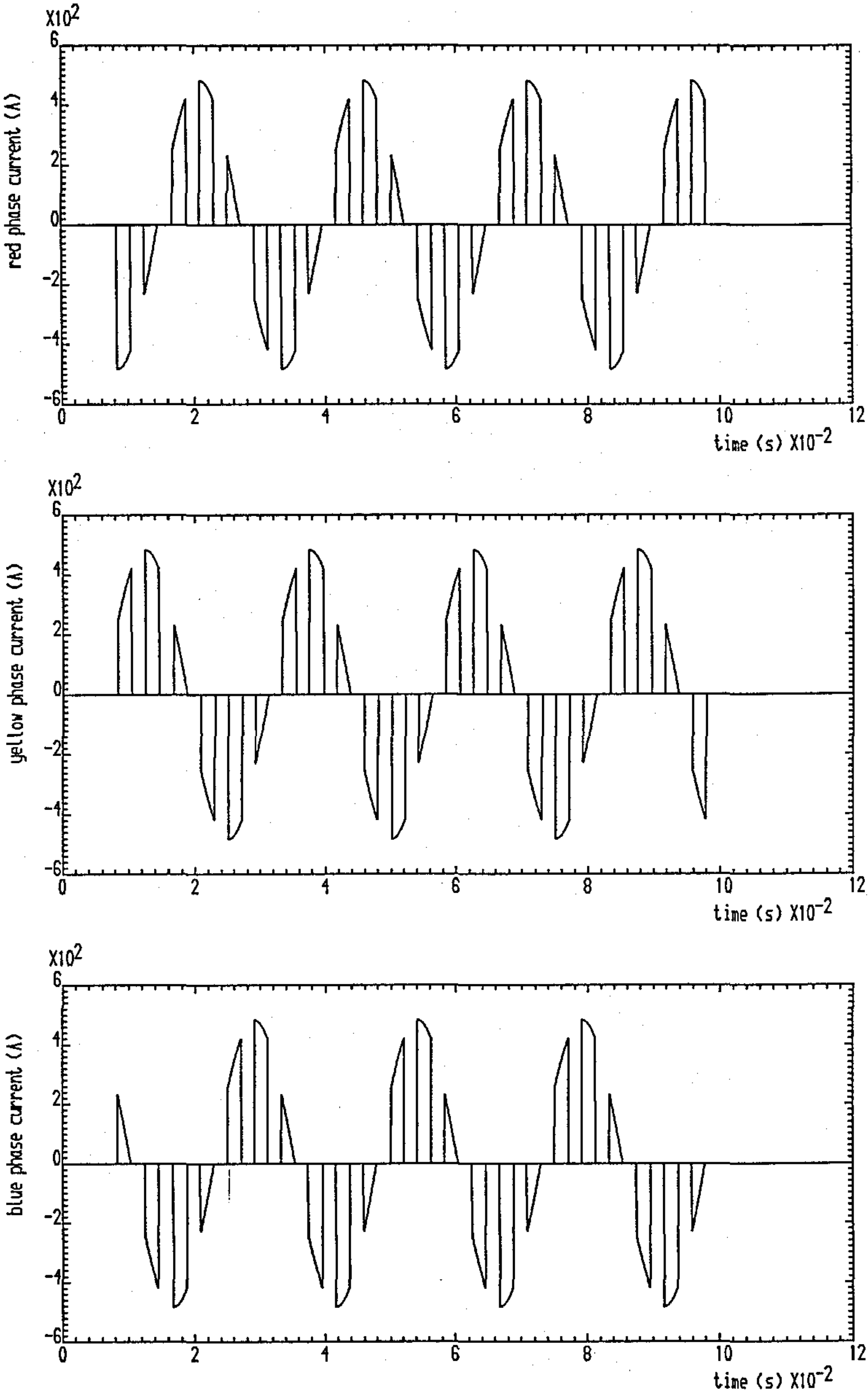



Fig 5.9(c) Converter AC side line voltages with 90 degrees trigger angle

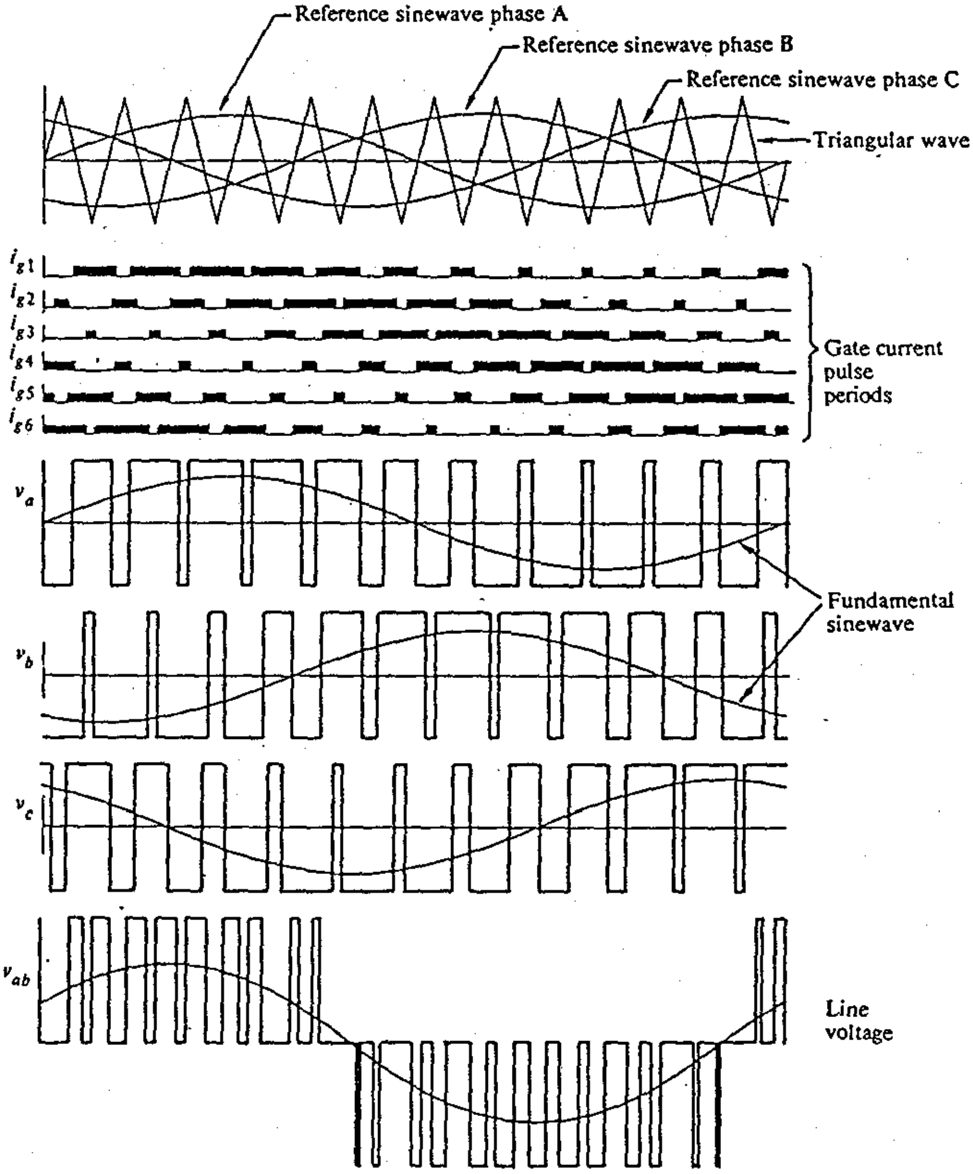



**Fig 5.10 PWM waveform for 3-phase inverter**

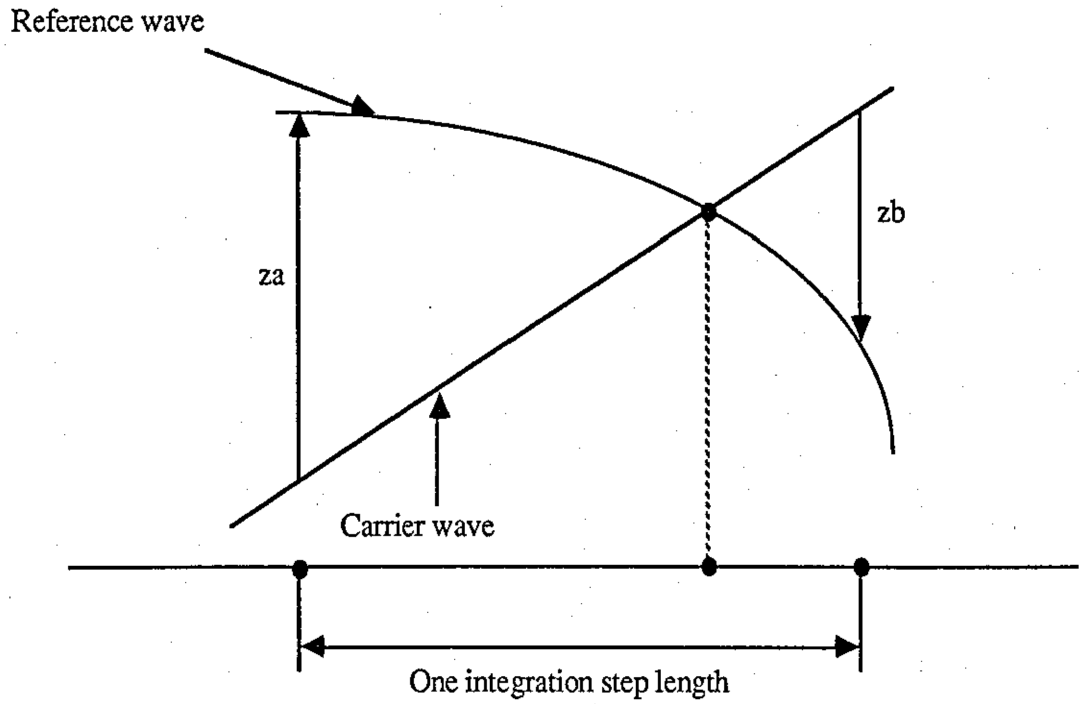



**Fig 5.11(a) Intersection between modulating and carrier waves**

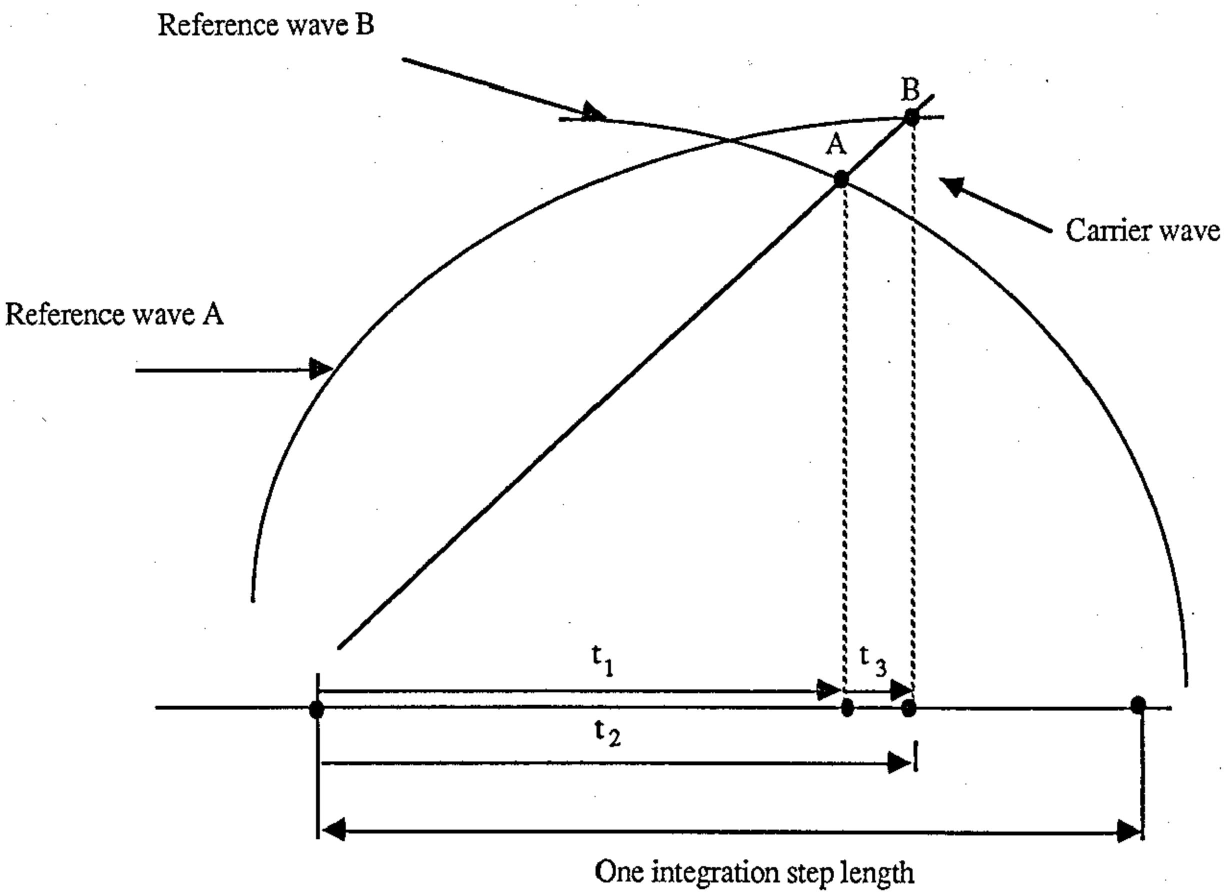



**Fig 5.11(b) Points of intersection between modulating and carrier waveforms**

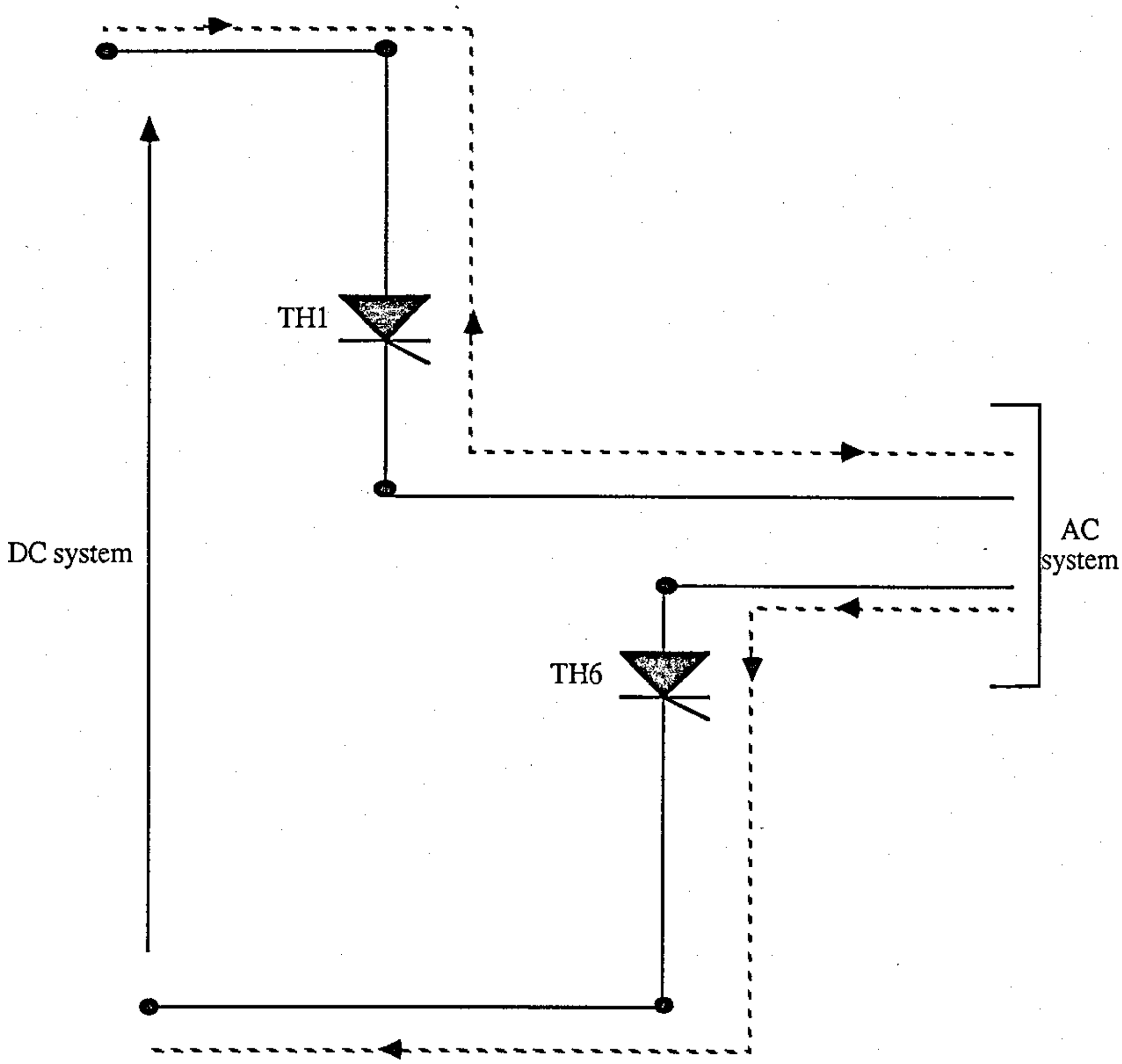



Fig 5.12 Conduction mesh formed by thyristors 1 and 6

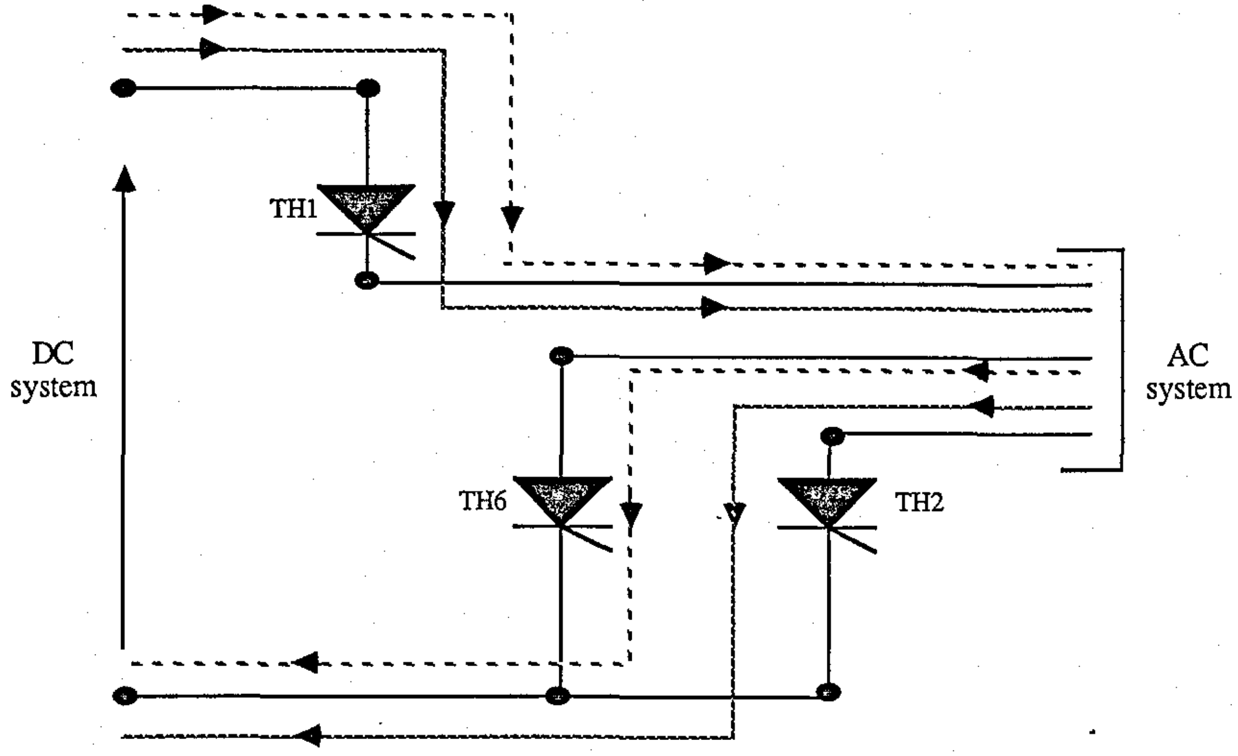



**Fig 5.13(a) Conduction meshes formed by thyristors 1, 6 and 2**

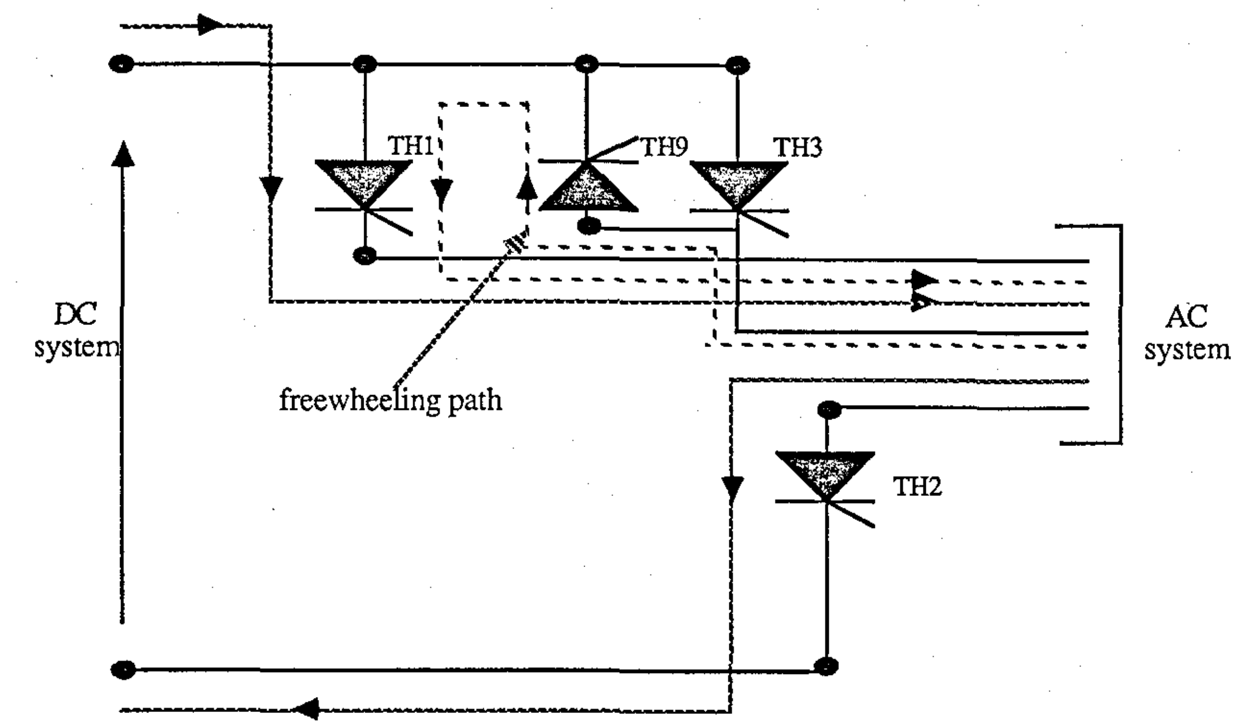



**Fig 5.13(b) Freewheeling path formed by thyristors 1 and 9**

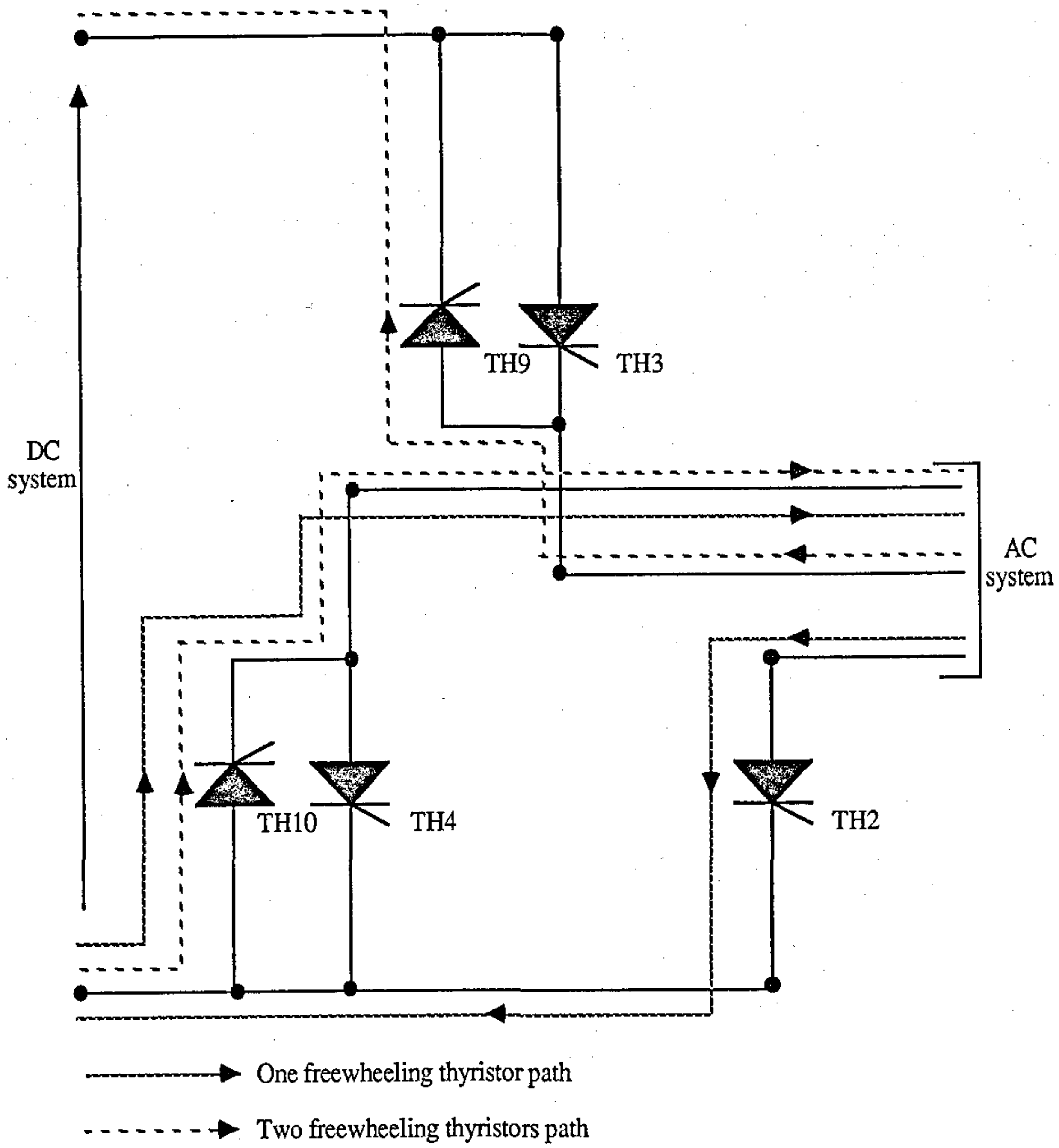



**Fig 5.14 Two freewheeling thyristors current meshes**

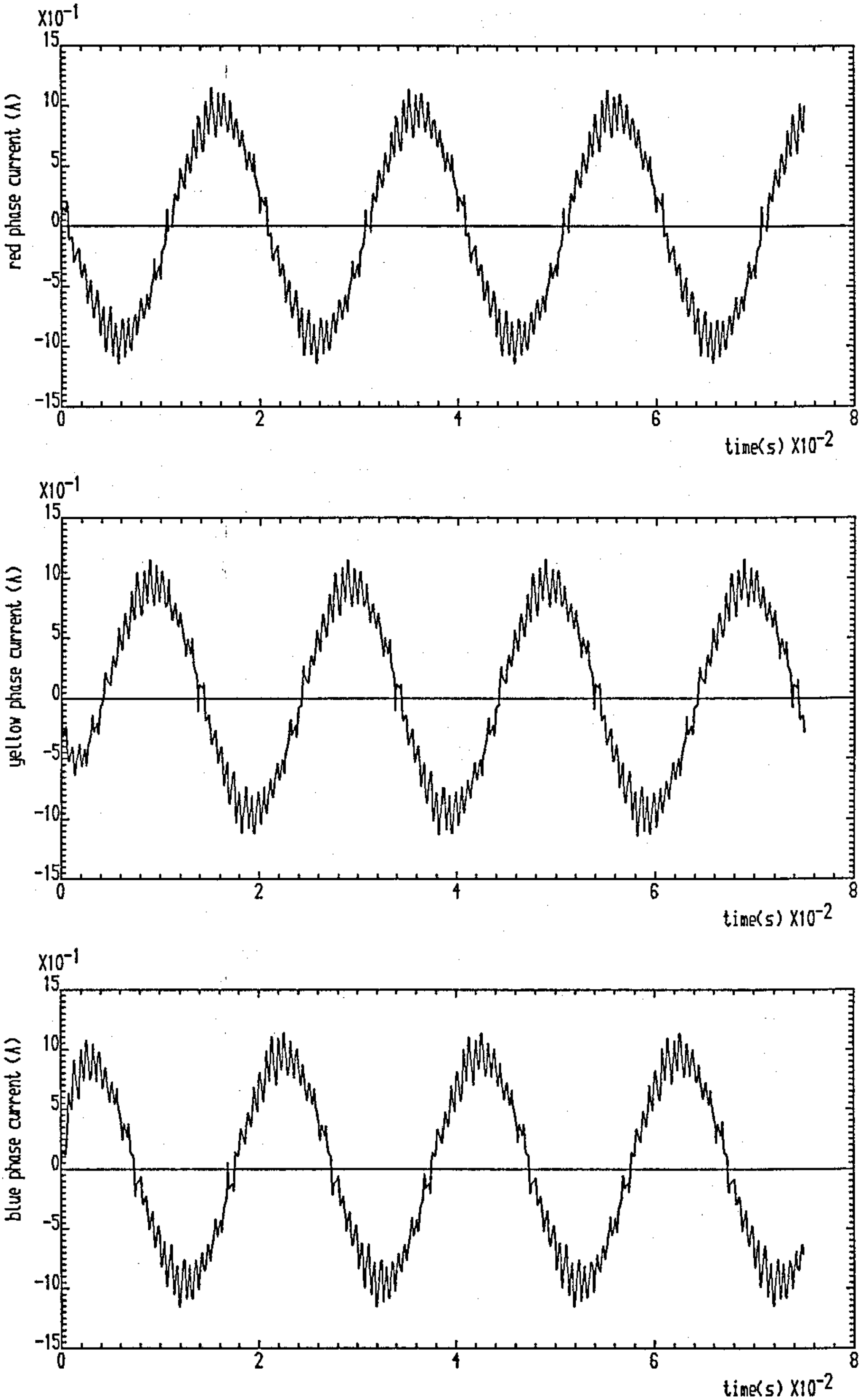



Fig 5.15(a) Converter AC side line currents with a modulating frequency of 800 Hz

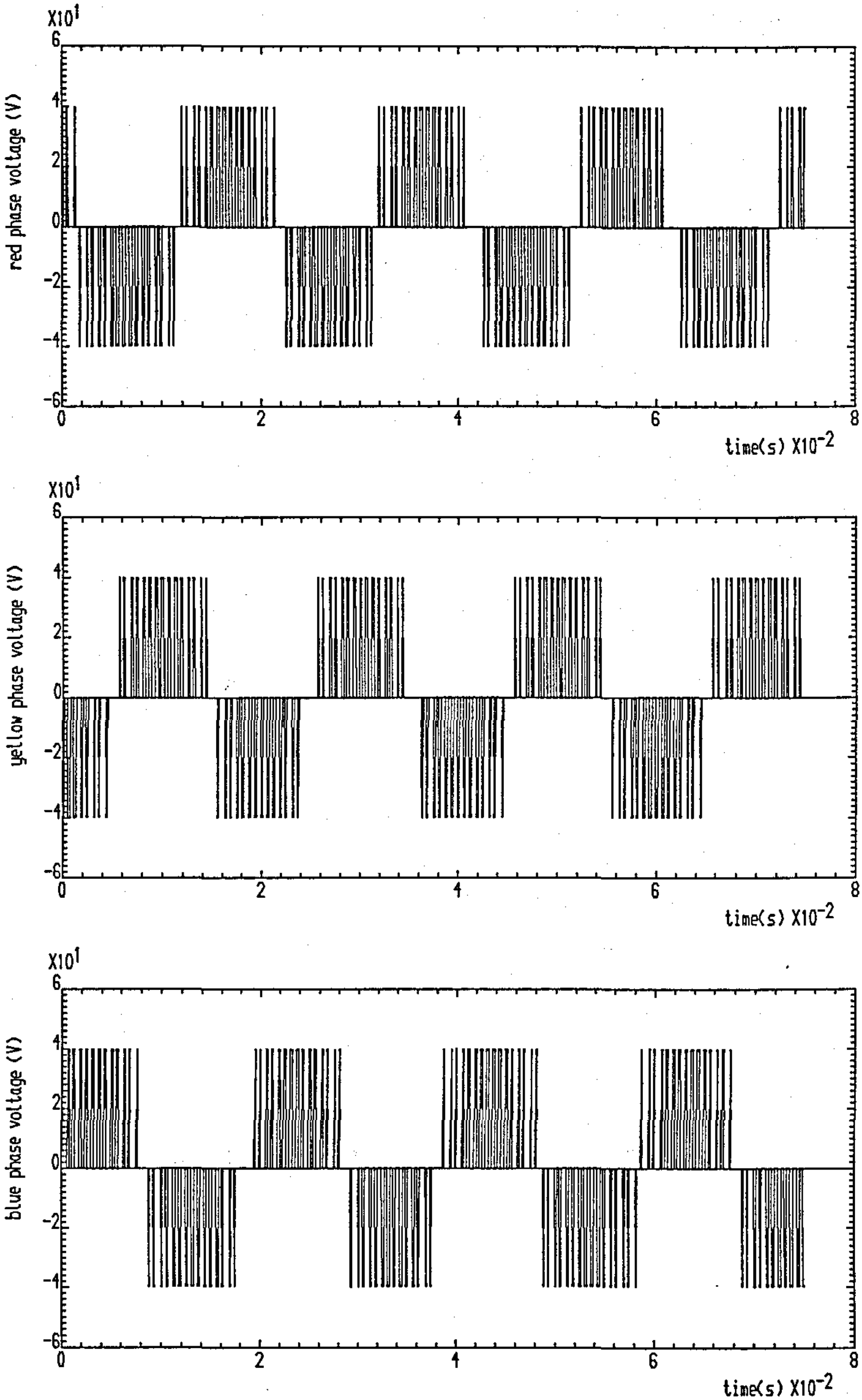



Fig 5.15(b) Converter AC side line voltages with a modulating frequency of 800 Hz

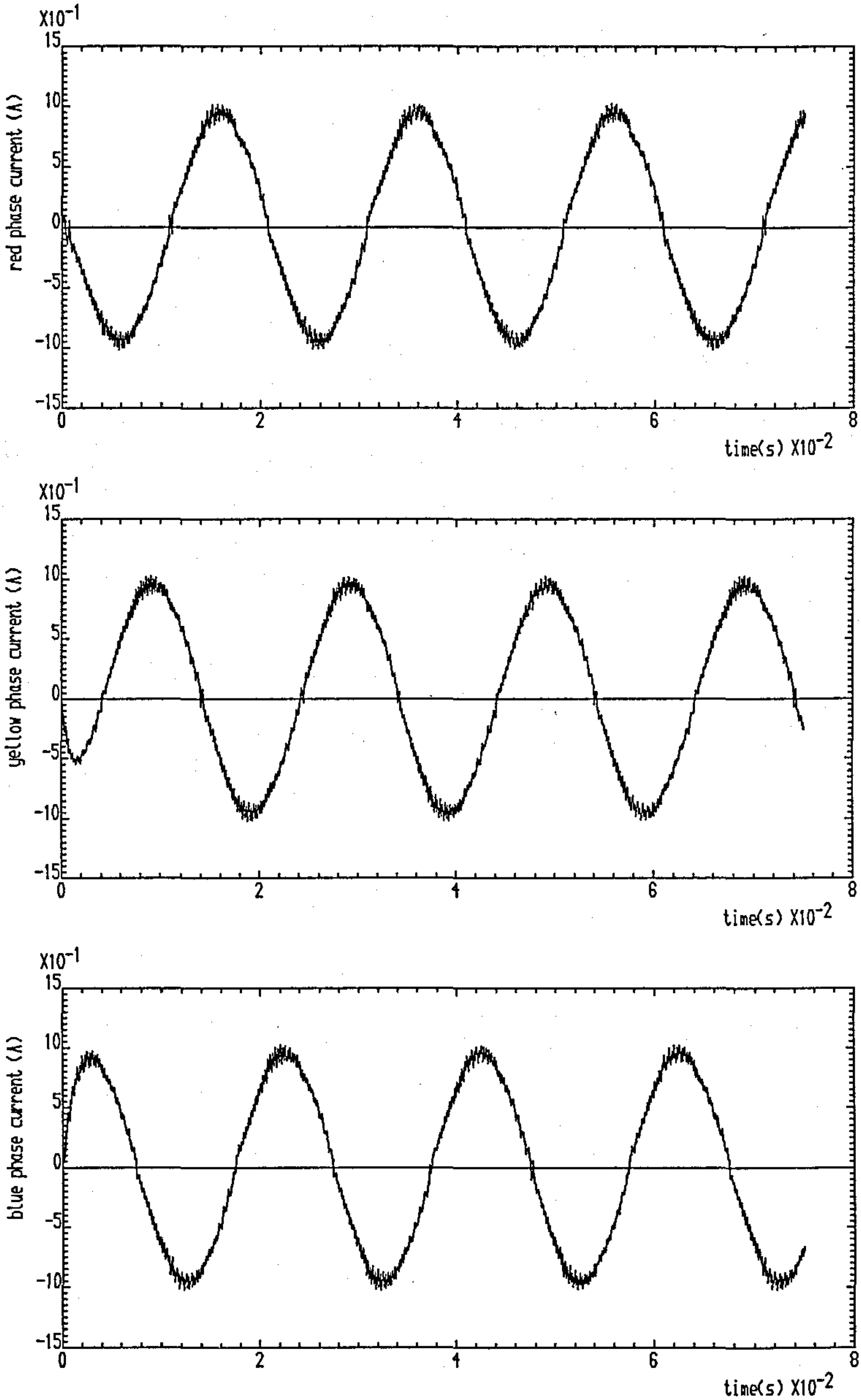



Fig 5.16(a) Converter AC side line currents with a modulating frequency of 2kHz

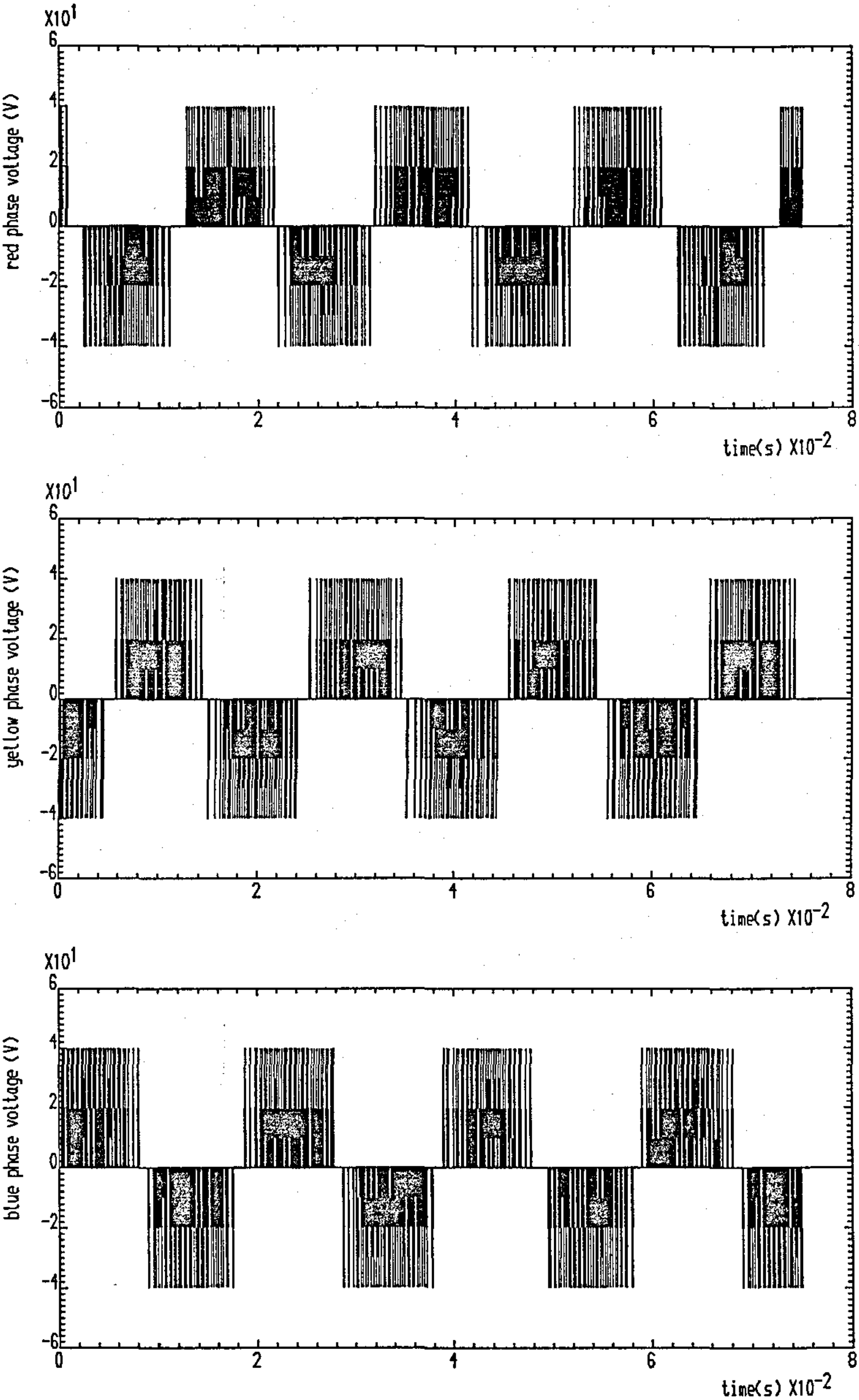



Fig 5.16(b) Converter AC side line voltage with a modulating frequency of 2kHz

# Chapter 6

## SIMULATION OF THE COMPLETE SHIP'S POWER SYSTEM

The solution of an electrical network containing synchronous generators yields inductance matrices with time-varying coefficients, which require to be inverted at every stage of a numerical solution [5]. A complete power system of the form of Fig 1.1 contains many meshes and the order of the resultant inductances matrix is consequently large. The solution time required is approximately proportional to the cube of the matrix order and in a conventional mesh analysis a very long computer run-time is required[9]. In this thesis, an alternative approach based on Kron's diakoptic method [1] is used to solve the time-varying differential equations. Mesh analysis is used to produce the network equation which can subsequently be partitioned to enable a relatively rapid numerical solution to be obtained, since the order of the corresponding system matrices is much reduced [9]. The conduction pattern of the thyristor bridge in Fig 1.1 changes continually with time. The computer program is therefore developed so as to handle automatically the changing conduction pattern and to assemble the corresponding link/mesh transformation ($C_m^L$) described in section 6.2.

### 6.1 Modelling of the Complete Ship's Power System

#### 6.1.1 A diakoptic approach

It is convenient to consider the sub-networks of the complete system of Fig 6.1 as the various items of plant, and these are defined as: synchronous generator 1, synchronous generator 2, the synchronous machine of the MG set, the bus coupler (SW4) and the converter bridge. Only the AC interconnected parts of the system are considered, so reducing the rank of the resulting resistive/inductive matrix, which in turn reduces the computing time for the numerical solution. In addition, it may also avoid the probability of numerical instability in the resulting solution if too many sub-networks are formed[10]. Using the numerical data obtained from the resulting solution, the DC part of the system (ie. the DC machine in the MG set) are solved separately from the set of differential equations derived in section 2.2 and 3.2. To facilitate the diakoptic tearing process, fictitious infinite inductances [11] are inserted at points A and B where, for the purpose of analysis, they are be replaced by the fictitious voltage sources $V_1$ to $V_4$ of Fig 6.2. Each voltage source is common to more than one sub-network. (eg $V_1$ and $V_2$ are common to synchronous generator 1, the synchronous motor and the bus coupler, and $V_3$ and $V_4$ to generator 2, the converter bridge and the bus coupler) The networks may now be separated at tear points A and B, without affecting the system currents, to give five sub-networks, together with two extra networks called the link networks comprising the infinite inductance and the current sources $i_1$ to $i_4$. The torn and link networks are all shown in Fig 6.3. The mesh current $i_{11}$, $i_{12}$, etc of Fig 6.1 are related to the current sources $i_1$ to $i_4$ in the link network of Fig 6.3 by a link/torn-mesh transformation $C^{L}_{m}$ which may be shown to be

$$
\begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \end{bmatrix}
=
\begin{bmatrix}
\underbrace{\begin{matrix} 1 & & & & \\ & 1 & & & \\ & & & & \\ & & & & \end{matrix}}_{\text{gen 1}} &\vdots&
\underbrace{\begin{matrix} & & & & \\ & & & & \\ 1 & & & & \\ & 1 & & & \end{matrix}}_{\text{gen 2}} &\vdots&
\underbrace{\begin{matrix} -1 & & & & \\ & -1 & & & \\ & & & & \\ & & & & \end{matrix}}_{\text{synch MG set}} &\vdots&
\underbrace{\begin{matrix} -1 & \\ & -1 \\ 1 & \\ & 1 \end{matrix}}_{\text{bus coupler}} &\vdots&
\underbrace{\begin{matrix} \text{changes} \\ \text{with} \\ \text{conuction} \\ \text{pattern} \end{matrix}}_{\text{converter bridge}}
\end{bmatrix}
\begin{bmatrix} i_{11} \\ i_{12} \\ i_{f1} \\ i_{d1} \\ i_{q1} \\ \hline i_{21} \\ i_{22} \\ i_{f2} \\ i_{d2} \\ i_{q2} \\ \hline i_{31} \\ i_{32} \\ i_{f3} \\ i_{d3} \\ i_{q3} \\ \hline i_{41} \\ i_{42} \\ \hline i_{51} \\ i_{52} \end{bmatrix}
\quad (6.1)
$$

Column labels (in order): 11 12 f1 f2 f3 | 21 22 f2 d2 q2 | 31 32 f3 d3 q3 | 41 42 | 51 52 or, in abbreviated form

$$I^{L} = C^{L}_{.m}\, I^{m} \tag{6.2}$$

In a similar manner, the fictitious voltage sources in the torn networks of Fig 6.3 may be shown to be related to the voltages across the infinite inductances of the link network of Fig 6.3

$$e_{m} = C^{.L}_{m}\, V_{L} \tag{6.3}$$

where $e_{m} = [e_{11}\ e_{12}\ e_{f1}\ e_{d1}\ e_{q1}\ e_{21}\ e_{22}\ e_{f2}\ e_{d2}\ e_{q2}\ e_{31}\ e_{32}\ e_{f3}\ e_{d3}\ e_{q3}\ e_{41}\ e_{42}\ e_{51}\ e_{52}]^{t}$

and $V_{L} = [V_{1}\ V_{2}\ V_{3}\ V_{4}]^{t}$

Using the mesh currents defined in Fig 6.1, the mesh voltage equation for the torn sub-networks of Fig 6.3 is,

$$\begin{bmatrix} E_{m1} \\ E_{m2} \\ E_{m3} \\ E_{m4} \\ E_{m5} \end{bmatrix} - \begin{bmatrix} e_{m1} \\ e_{m2} \\ e_{m3} \\ e_{m4} \\ e_{m5} \end{bmatrix} = \begin{bmatrix} R_{m1} & & & & \\ & R_{m2} & & & \\ & & R_{m3} & & \\ & & & R_{m4} & \\ & & & & R_{m5} \end{bmatrix} \begin{bmatrix} I^{m1} \\ I^{m2} \\ I^{m3} \\ I^{m4} \\ I^{m5} \end{bmatrix} + \begin{bmatrix} L_{m1} & & & & \\ & L_{m2} & & & \\ & & L_{m3} & & \\ & & & L_{m4} & \\ & & & & L_{m5} \end{bmatrix} \begin{bmatrix} \frac{dI^{m1}}{dt} \\ \frac{dI^{m2}}{dt} \\ \frac{dI^{m3}}{dt} \\ \frac{dI^{m4}}{dt} \\ \frac{dI^{m5}}{dt} \end{bmatrix} \tag{6.4}$$

where $L_{m1}$, $L_{m2}$ and $L_{m3}$ are the mesh inductance matrices for generators 1 and 2 and the synchronous machine, as defined in equation 2.10.

$R_{m1}$, $R_{m2}$ and $R_{m3}$ are the sum of both the R and G matrices for generators 1 and 2 and the synchronous machine, as defined in equation 2.10.

$E_{m1}$, $E_{m2}$ and $E_{m3}$ are the mesh voltage source vectors for generators 1 and 2 and the synchronous machine, as defined in equation 2.10.

$E_{m4}$, $R_{m4}$ and $L_{m4}$ are the impressed voltage vector, the mesh resistance and inductance matrices for the bus coupler, as defined in equations 4.1 to 4.11.

$E_{m5}$, $R_{m5}$ and $L_{m5}$ are the source voltage vector, the mesh resistance and inductance matrices of the converter, as described in chapter 5.

In abbreviated form, equation (6.4) may be written

$$E_m - e_m = R_{mm} I^m + L_{mm} \frac{dI^m}{dt} \tag{6.5}$$

The matrix voltage equation for the link networks is

$$V^L = L_{LL} \frac{dI^L}{dt} \tag{6.6}$$

where $L^{LL}$ is a diagonal matrix containing the infinite inductances.

Combining equation (6.3) and (6.5)

$$E_m - C_m^{\cdot L} V_L = R_{mm} I^m + L_{mm} \frac{dI^m}{dt} \tag{6.7}$$

from which

$$\frac{dI^m}{dt} = L_{mm}^{-1} \left(E_m - C_m^{\cdot L} V_L - R_{mm} I^m\right) \tag{6.8}$$

From equation (6.6)

$$\frac{dI^L}{dt} = L_{LL}^{-1} \; V_L \tag{6.9}$$

Differentiating both sides of equation (6.2) and substituting the result into equation (6.6) gives

$$C^{L}_{.m} \frac{dI^{m}}{dt} = L^{-1}_{LL} \quad V_{L} \qquad (6.10)$$

Substituting equation (6.7) into equation (6.10) gives

$$C^{L}_{.m} L_{mm}^{-1} \left(E_{m} - C^{.L}_{m} V_{L} - R_{mm} I^{m}\right) = L^{-1}_{LL} \quad V_{L} \qquad (6.11)$$

and on re-arranging

$$C^{L}_{.m} L_{mm}^{-1} \left(E_{m} - R_{mm} I^{m}\right) = \left(L^{-1}_{LL} \quad + C^{L}_{.m} L_{mm}^{-1} C^{.L}_{m}\right) V_{L} \qquad (6.12)$$

If a simplifying substitution is defined by

$$A = \left(L^{-1}_{LL} \quad + C^{L}_{.m} L_{mm}^{-1} C^{.L}_{m}\right) \qquad (6.13)$$

then, on substituting equation (6.12) and (6.13) in equation (6.8)

$$\frac{dI^{m}}{dt} = L_{mm}^{-1} \left(E_{m} - R_{mm} I^{m} - C^{.L}_{m} A^{-1} C^{L}_{.m} L_{mm}^{-1} \left(E_{m} - R_{mm} I^{m}\right)\right)$$

$$= L_{mm}^{-1} \left(U_{mm} - C^{.L}_{m} A^{-1} C^{L}_{.m} L_{mm}^{-1}\right) \left(E_{m} - R_{mm} I^{m}\right) \qquad (6.14)$$

where $U_{mm}$ is a unit matrix.

Since the elements of the matrix $L_{LL}$ have infinite value, the matrix $L_{LL}^{-1}$ is null, i.e. $L_{LL}^{-1} = 0$, so that,

$$A = C^{L}_{.m} L_{mm}^{-1} C^{.L}_{m} \qquad (6.15)$$

Equation (6.14) may be solved using numerical integration to give a step-by-step solution for the vector $I^{m}$, which is a solution for the mesh currents in the original networks.

### 6.1.2 Partitioning of the Network Equation

Equation (6.14) may be re-arranged in the abbreviated form,

$$\frac{dI^m}{dt} = A^{mm}\left(E_m - R_{mm} I_m\right) \qquad (6.16)$$

$$\text{where } A^{mm} = L_{mm}^{-1}\left(U_{mm} - C_m^{.L} A^{-1} C_{.m}^{L} L_{mm}^{-1}\right)$$

$$\text{in which } A = C_{.m}^{L} L_{mm}^{-1} C_m^{.L}$$

$A^{mm}$ is a large matrix which may require considerable computer storage space. A technique of partitioning [9] can be used to simplified the network equation and to reduce the program run-times.

The link/mesh transformation of equation (6.1) may be partitioned to give

$$C_{.m}^{L} = [\; \underbrace{C_1}_{\text{gen 1}} \quad \underbrace{C_2}_{\text{gen 2}} \quad \underbrace{C_3}_{\text{synch. MG set}} \quad \underbrace{C_4}_{\text{bus coupler}} \quad \underbrace{C_5}_{\text{converter bridge}} \;]$$

(6.17)

and it can be shown [9], that $A^{mm}$ can be partitioned as follows

$$A^{mm} = \begin{bmatrix} A^{11} & A^{12} & A^{13} & A^{14} & A^{15} \\ A^{21} & A^{22} & A^{23} & A^{24} & A^{25} \\ A^{31} & A^{32} & A^{33} & A^{34} & A^{35} \\ A^{41} & A^{42} & A^{43} & A^{44} & A^{45} \\ A^{51} & A^{52} & A^{53} & A^{54} & A^{55} \end{bmatrix}$$

(6.18)

where

$$A^{ab} = L_{ma}^{-1} \left(- C_a^{\,t} A^{-1} C_b L_{mb}^{-1}\right)$$

with a = 1 to 5
b = 1 to 5
and $a \neq b$

$$A^{aa} = L_{ma}^{-1} \left(U_{aa} - C_a^{\,t} A^{-1} C_a L_{ma}^{-1}\right)$$

with a = 1 to 5

$$\text{and} \quad A^{-1} = \left(\sum_{a=1}^{5} C_a L_{ma}^{-1} C_a^{\,t}\right)^{-1}$$

Combining equations (6.16) and (6.18) gives

$$\begin{bmatrix} \dfrac{dI^{m1}}{dt} \\ \dfrac{dI^{m2}}{dt} \\ \dfrac{dI^{m3}}{dt} \\ \dfrac{dI^{m4}}{dt} \\ \dfrac{dI^{m5}}{dt} \end{bmatrix} = \begin{bmatrix} A^{11} & A^{12} & A^{13} & A^{14} & A^{15} \\ A^{21} & A^{22} & A^{23} & A^{24} & A^{25} \\ A^{31} & A^{32} & A^{33} & A^{34} & A^{35} \\ A^{41} & A^{42} & A^{43} & A^{44} & A^{45} \\ A^{51} & A^{52} & A^{53} & A^{54} & A^{55} \end{bmatrix} \left[ \begin{bmatrix} E_{m1} \\ E_{m2} \\ E_{m3} \\ E_{m4} \\ E_{m5} \end{bmatrix} - \begin{bmatrix} R_{m1} & & & & \\ & R_{m2} & & & \\ & & R_{m3} & & \\ & & & R_{m4} & \\ & & & & R_{m5} \end{bmatrix} \begin{bmatrix} I^{m1} \\ I^{m2} \\ I^{m3} \\ I^{m4} \\ I^{m5} \end{bmatrix} \right] \tag{6.19}$$

The first sub-vector $\left(\dfrac{dI^{m1}}{dt}\right)$ of equation (6.19) is

$$\frac{dI^{m1}}{dt} = A^{11}\left(E_{m1} - R_{m1} I^{m1}\right) + A^{12}\left(E_{m2} - R_{m2} I^{m2}\right) + A^{13}\left(E_{m3} - R_{m3} I^{m3}\right) + A^{14}\left(E_{m4} - R_{m4} I^{m4}\right) + A^{15}\left(E_{m5} - R_{m5} I^{m5}\right) \quad (6.20)$$

from which a numercial solution for $I^{m1}$ may be obtained using a numerical integration. Since similar solutions may be obtained for $I^{m2}$, $I^{m3}$, $I^{m4}$ and $I^{m5}$, each sub-vector of equation (6.19) may be solved separately.

### 6.1.3 Summary of the complete model solution algorithm

After the system of Fig 6.1 is torn into the 5 sub-networks shown in Fig 6.3 the following steps are performed [9] .

(a) Assemble and invert the subdivision inductance matrices

$$L_{ma}^{-1} \quad \text{with} \quad a = 1 \text{ to } 5$$

where n is the number of sub-networks.

(b) Determine the link inductance matrix, $A^{-1}$, from equation (6.16).

(c) Determine the component of the torn mesh rate-of-change of current vectors.

$$\frac{dI^{ma}}{dt} = \sum_{b=1}^{n} A^{ab}\left[ V_{mb} - R_{mb} I_{mb} \right] \quad (6.21)$$

where a = 1 to 5

## 6.2 Formation of the link/mesh transformation

The converter bridge tensor, $C_5$ in equation (6.17), changes as the conduction pattern in the bridge changes continually with time. The following section describes the way in which the program assembles automatically the required tensor, according to the changes in the thyristor bridge.

### 6.2.1 Possible current transformations for the thyristor converter

Depending on which thyristors are conducting, the six possible mesh currents on the AC side of the converter shown in Fig 6.4(a) to (f) illustrate the current flow into the tear points A and B for each of the possible converter meshes. Each of the converter meshes has a corresponding connection to the torn sub-network of the converter bridge represented in Fig 6.5(a) to (f). These enables the converter bridge tensor, $C_5$, to be derived for each individual converter mesh.

The converter link/torn mesh sub-tensor $C_5$ of equation (6.17), for the situation when converter mesh 1 is conducting as in Fig 6.4(a), is obtained as follows. The mesh currents $i^{51}$ at the point of tear are related to the currents $i_1$ to $i_4$ in the link network of Fig 6.3 such that

$$C_5 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}$$

$$\text{and} \quad I^{m5} \doteq \begin{bmatrix} i^{51} \end{bmatrix}$$

Similarly, when converter mesh 2 conducts as shown in Fig 6.4(b),

$$C_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\text{and} \quad I^{m5} = \begin{bmatrix} i^{51} \end{bmatrix}$$

The same procedure applies to all the other converter meshes, and a master link/torn mesh transformation matrix Cpwm, which contains the six individual link/torn mesh transformations, is given in Table 6.1.

| Converter Meshes | | | | | |
|---|---|---|---|---|---|
| **1** | **2** | **3** | **4** | **5** | **6** |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | -1 | 0 | -1 |
| -1 | 1 | 0 | 1 | -1 | 1 |

Table 6.1 The master link/torn mesh transformation matrix Cpwm

### 6.2.2 Computer algorithm for determining $C_5$

The algorithm for determining the transformation matrix $C_5$ is conveniently illustrated by considering the inversion mode of the converter. The converter mesh labels (1 to 6) are defined in the last row of Cmod in Table 5.11. When a column in Cmod is chosen, its converter mesh is also recorded in an array ipass. According to the elements of ipass, the corresponding columns of Cpwm are chosen to form $C_5$. The technique is described below using the example of section 5.2.4.

(i) Normal conduction period

The mesh currents flowing into the tear points A and B, when thyristors 1, 6 and 2 are conducting, are as shown in Fig 6.6(a). Columns 1 and 2 of Cmod corresponding to mesh labels 1 and 3 are chosen and the elements of ipass are (1,3). According to ipass, columns 1 and 3 in Cpwm are chosen to form $C_5$. Thus

$$C_5 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \\ -1 & 0 \end{bmatrix}$$

$$\text{and} \quad I^{m5} = \begin{bmatrix} i^{51} \\ i^{52} \end{bmatrix}$$

In this manner, $C_5$ may be formed by determining the converter meshes and choosing their corresponding columns in Cpwm.

(ii) Start of freewheeling

When thyristor 6 turns off, and its freewheeling thyristor 9 is triggered, the mesh currents flowing into the tear points A and B are as shown in Fig 6.6(b). Columns 2 and

11 of Cmod are chosen and their corresponding converter mesh labels are 3 and 1. The corresponding elements of ipass are therefore (3,1) and columns 3 and 1 of Cpwm are therefore chosen to form $C_5$. Thus

$$C_5 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0 & -1 \end{bmatrix}$$

(iii) **End of freewheeling**

At the end of the freewheeling period, thyristor 3 is turned on and current flows into the tear points as shown in Fig 6.6(c). Columns 2 and 4 of Cmod are chosen and their corresponding mesh labels are 3 and 2. The elements of ipass are therefore (3,2) and columns 3 and 2 in Cpwm are chosen to form $C_5$.

$$C_5 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The same technique applies to the rectification mode and matrix $C_5$ may be determined during each conduction period.

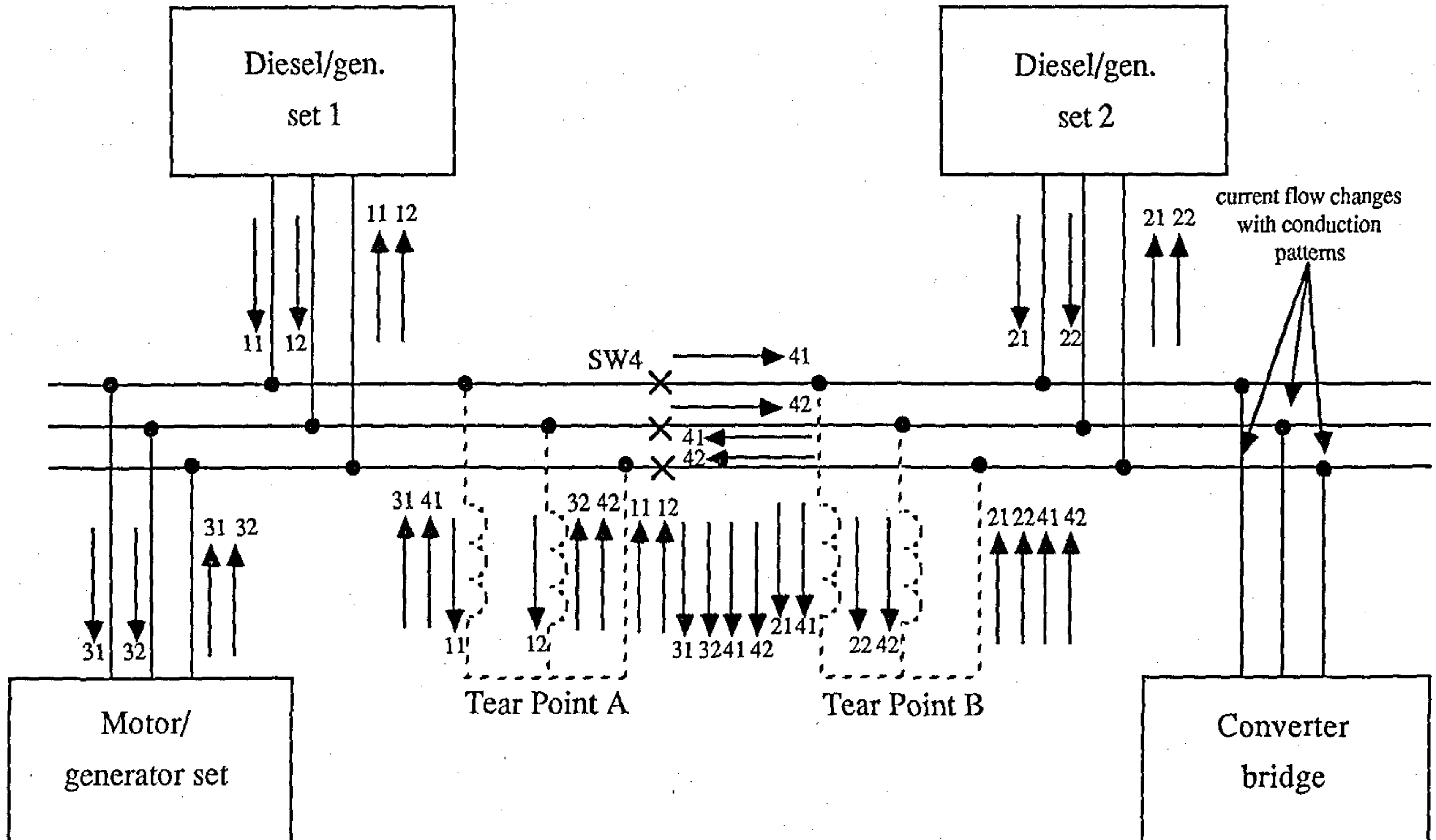



**Fig 6.1 Points of tear showing link currents**

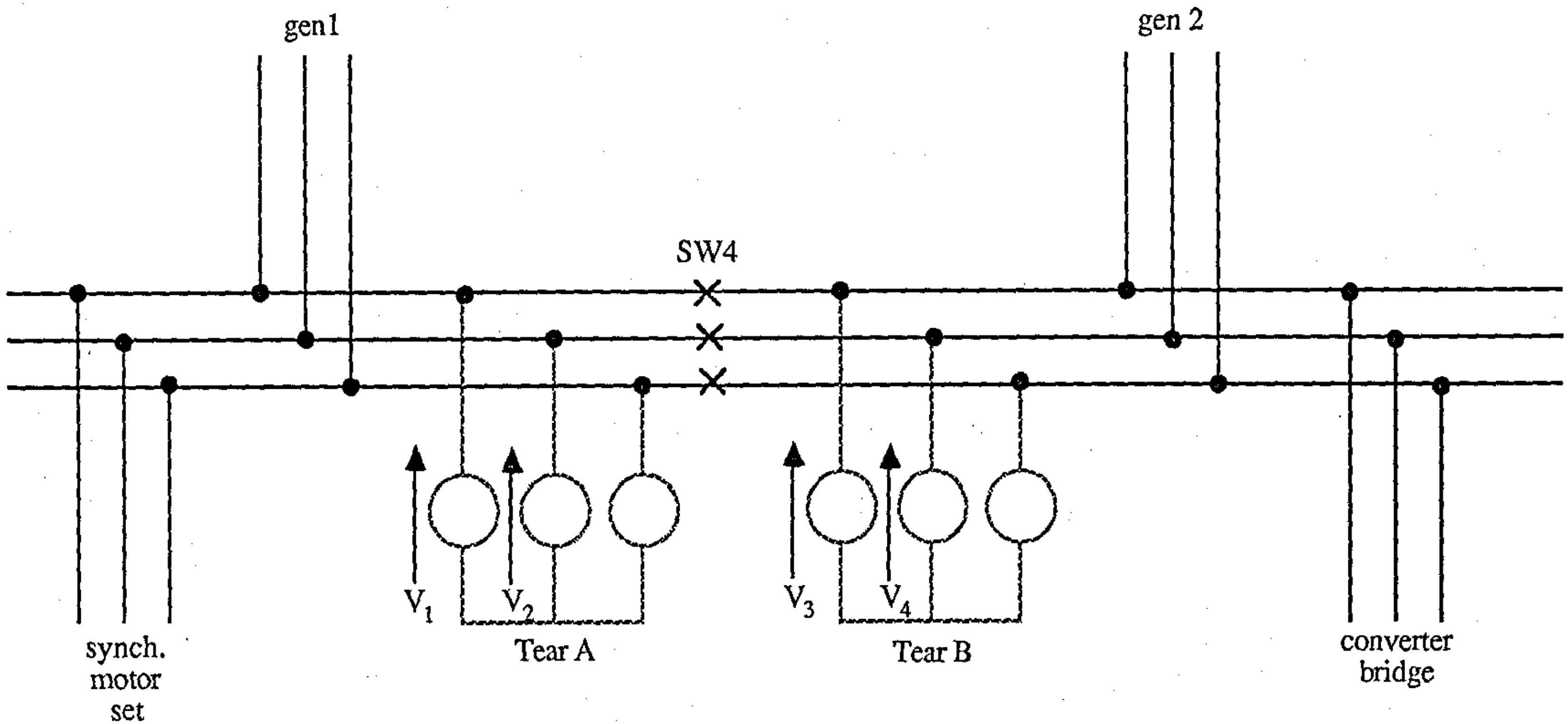
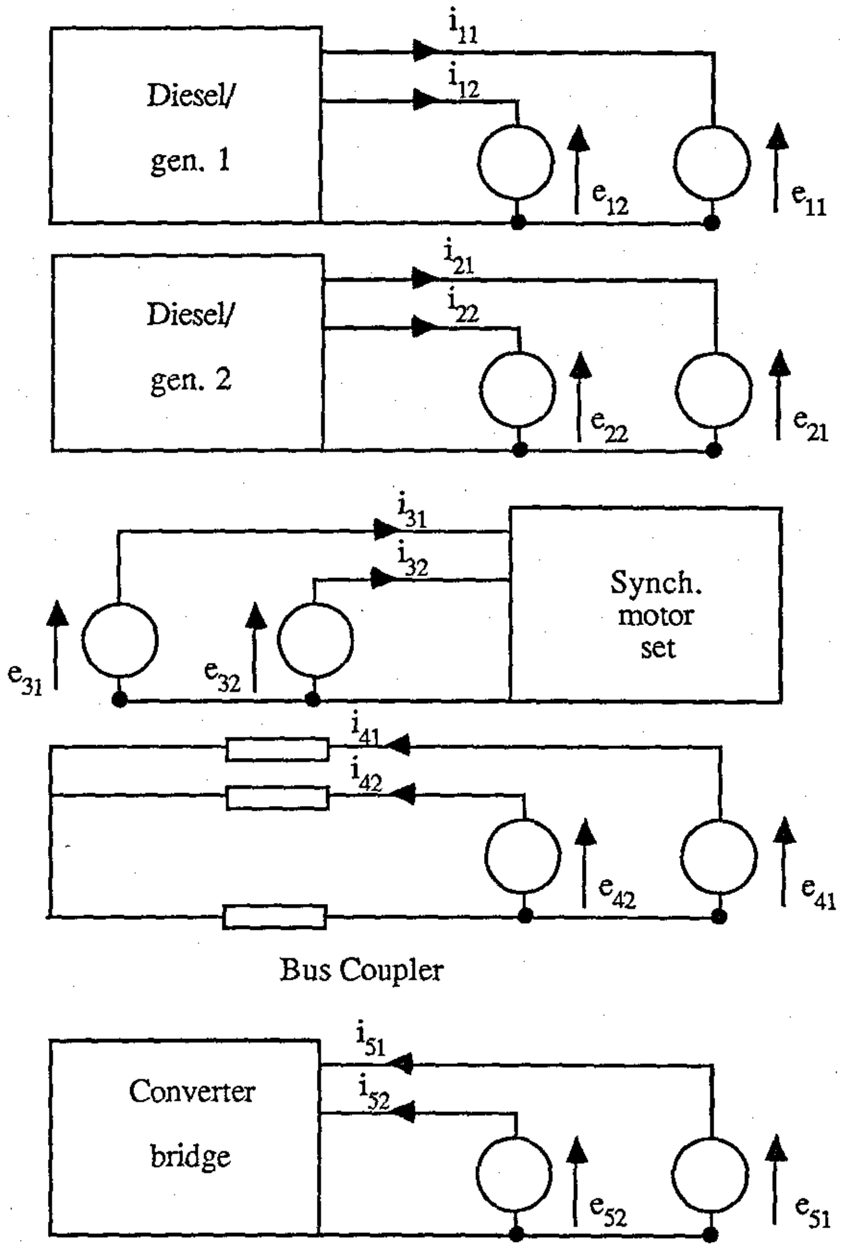
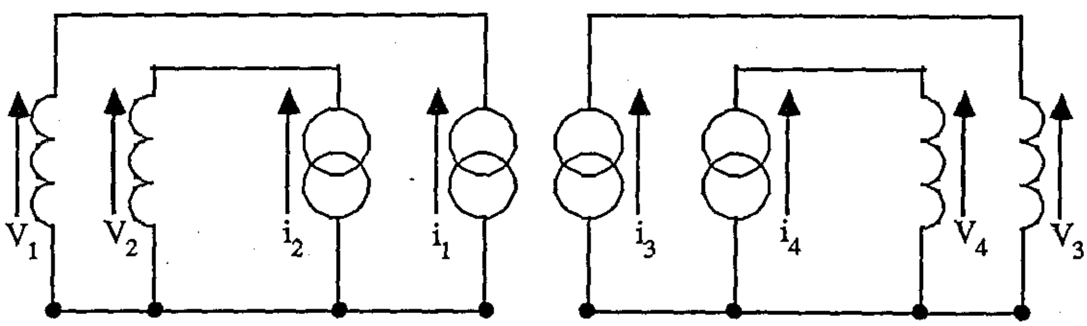



**Fig 6.2 Replacing infinite inductances by voltage sources**




Torn sub-networks




Link networks

**Fig 6.3 Torn and link networks**

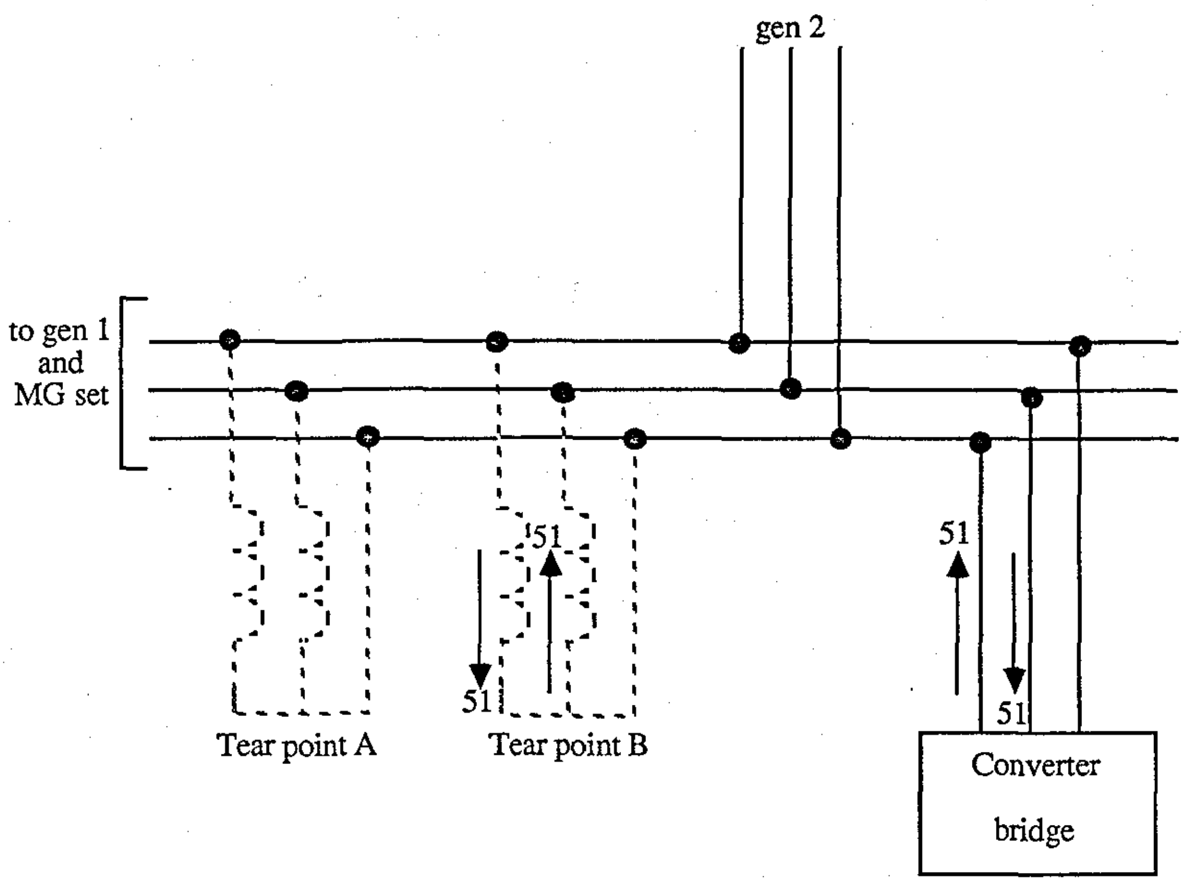



Fig 6.4(a) Current flow in converter for mesh 1

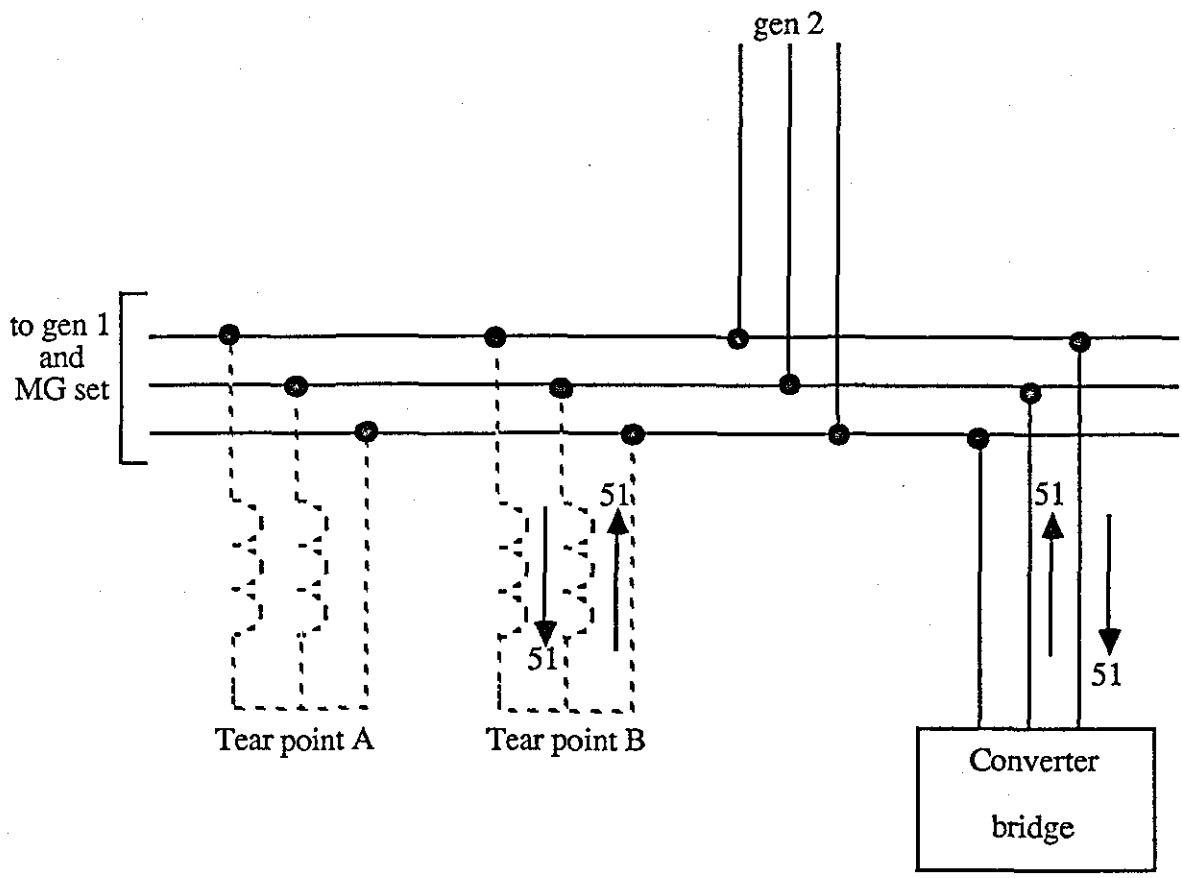



Fig 6.4(b) Current flow in converter for mesh 2

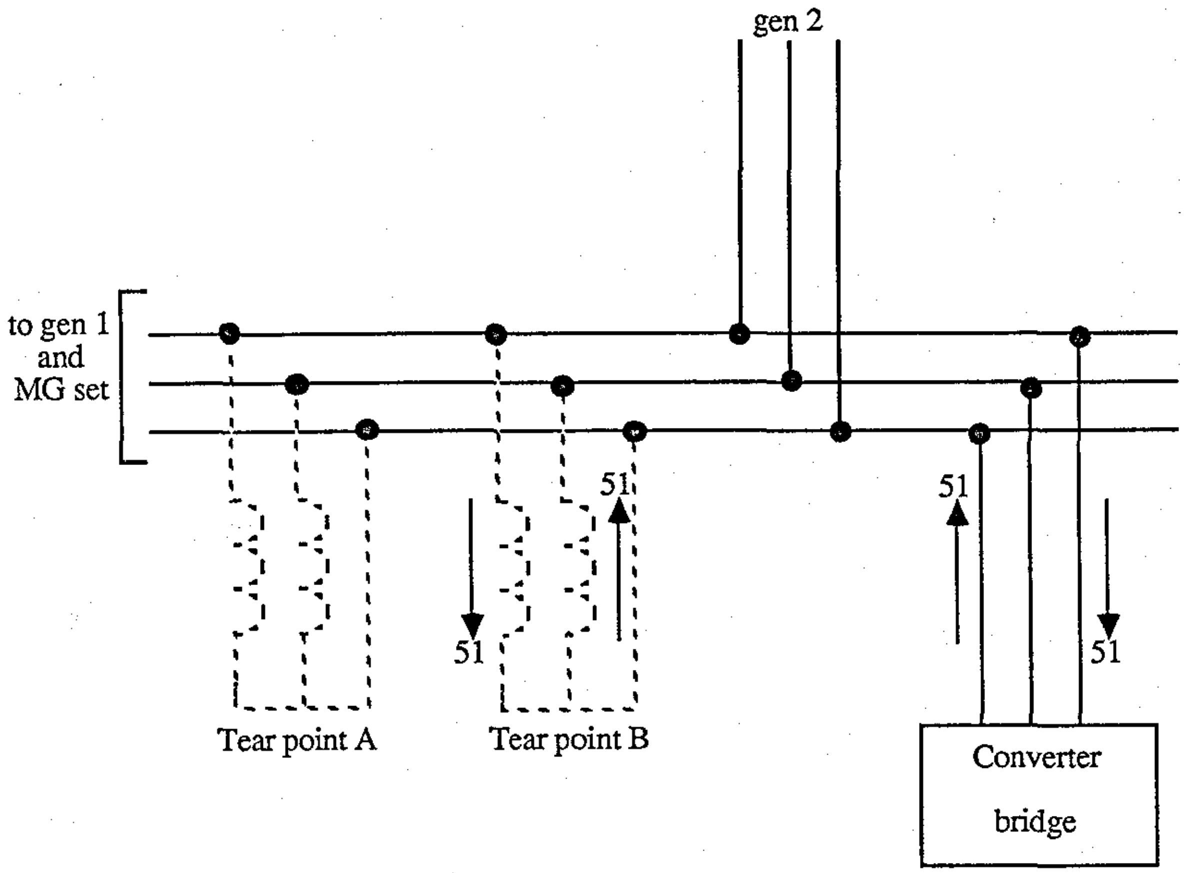



**Fig 6.4(c) Current flow in converter for mesh 3**

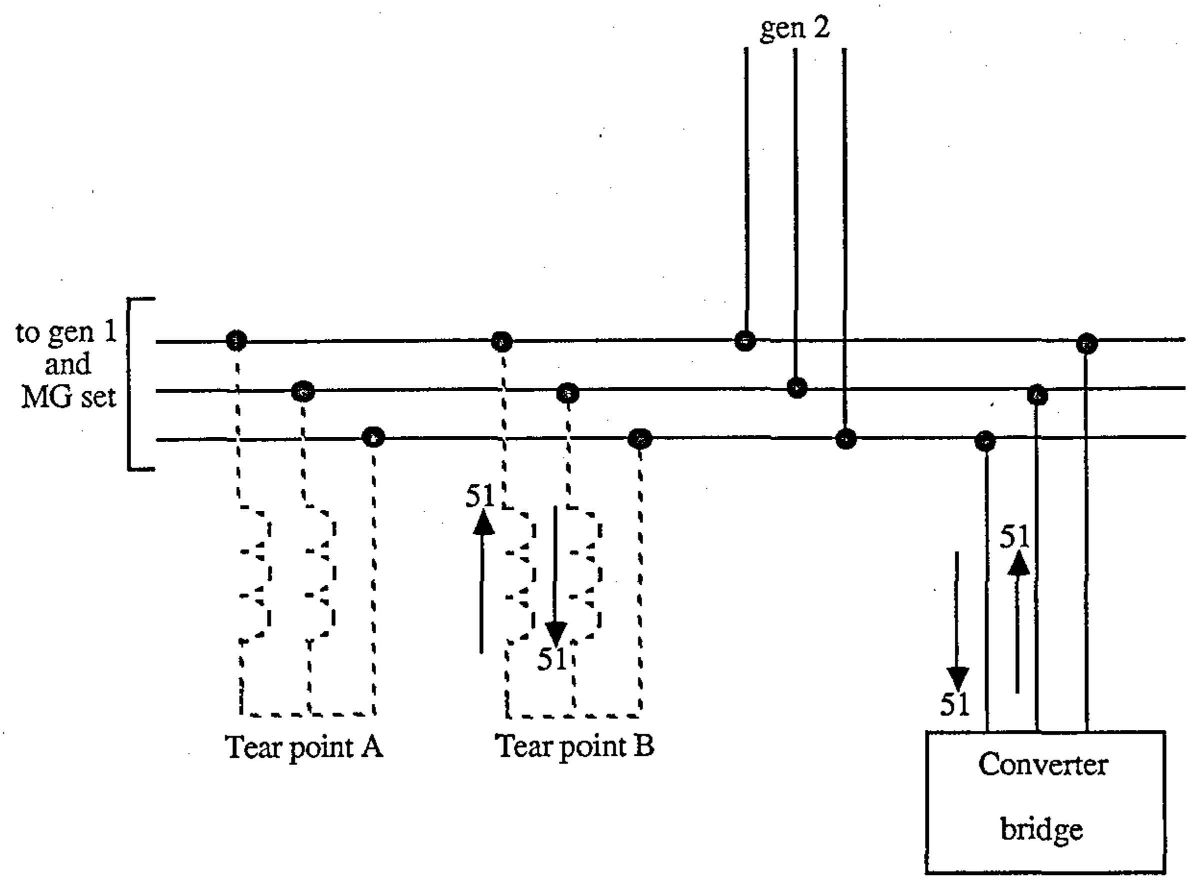



**Fig 6.4(d) Current flow in converter for mesh 4**

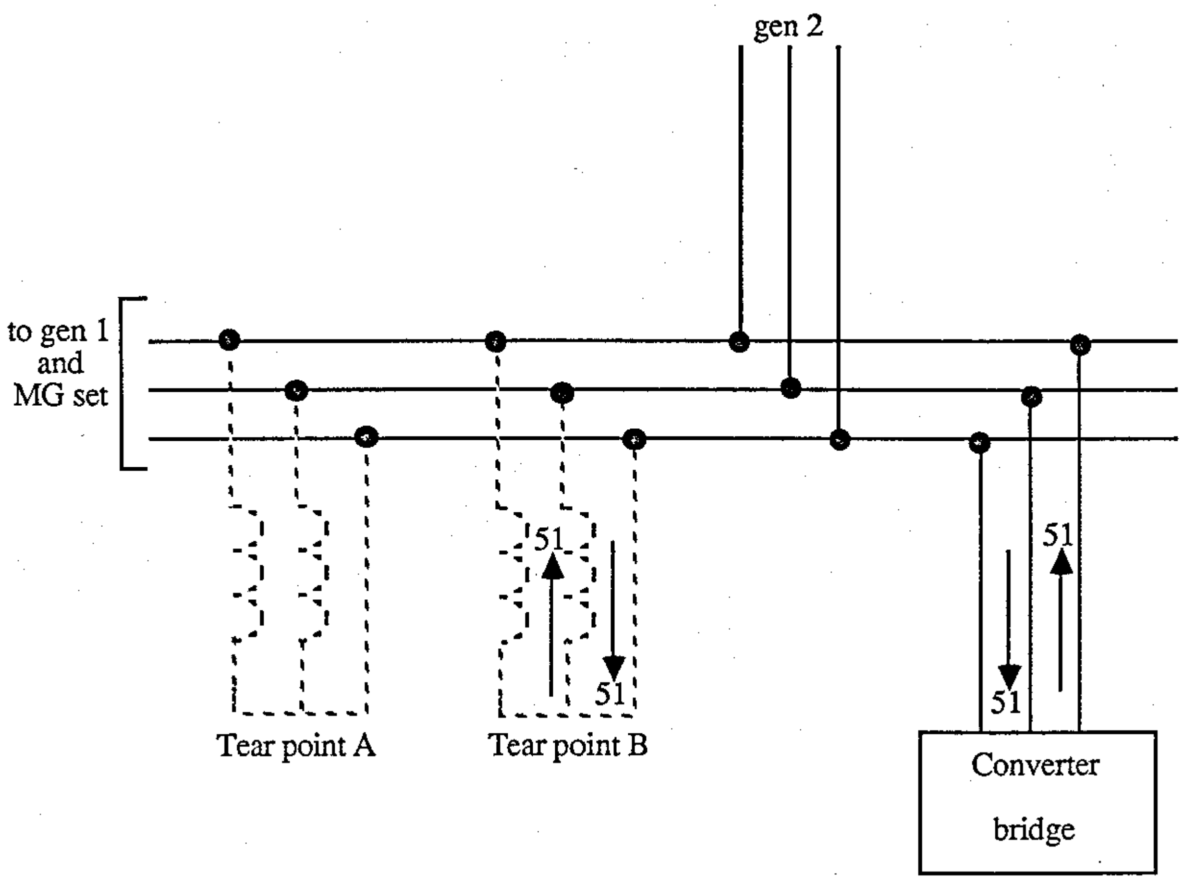


**Fig 6.4(e) Current flow in converter for mesh 5**

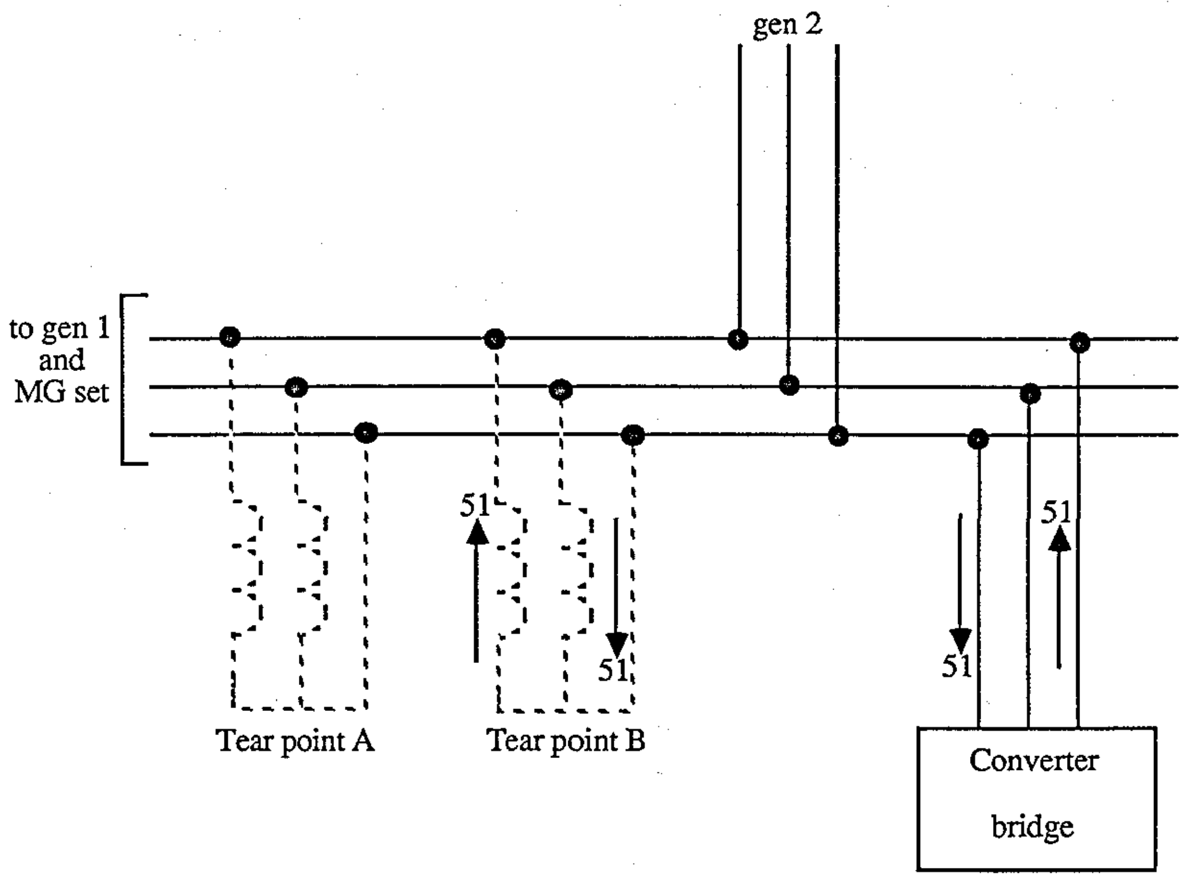


**Fig 6.4(f) Current flow in converter for mesh 6**

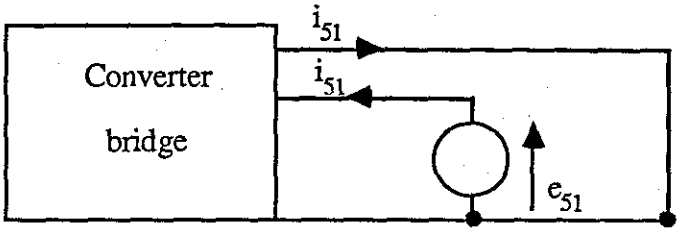



Fig 6.5(a) Converter torn sub-network for mesh 1

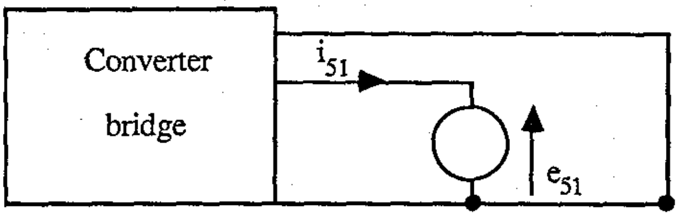



Fig 6.5(b) Converter torn sub-network for mesh 2

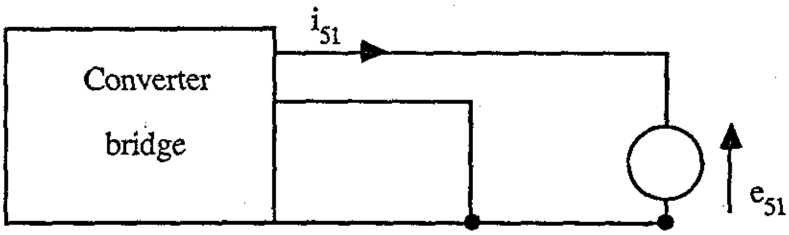



Fig 6.5(c) Converter torn sub-network for mesh 3

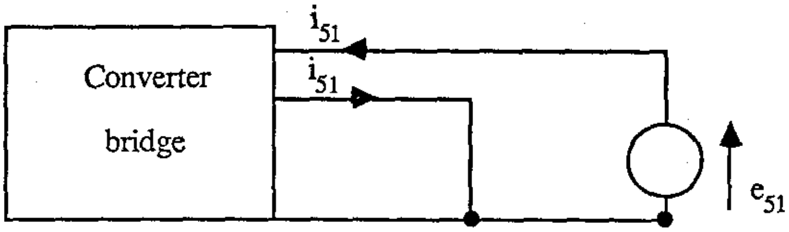



Fig 6.5(d) Converter torn sub-network for mesh 4

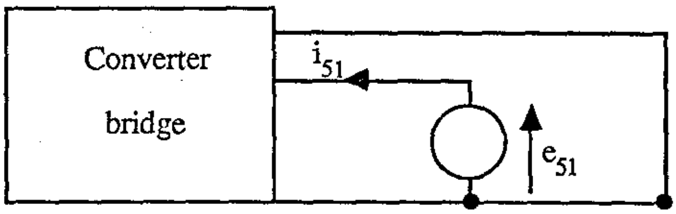



Fig 6.5(e) Converter torn sub-network for mesh 5

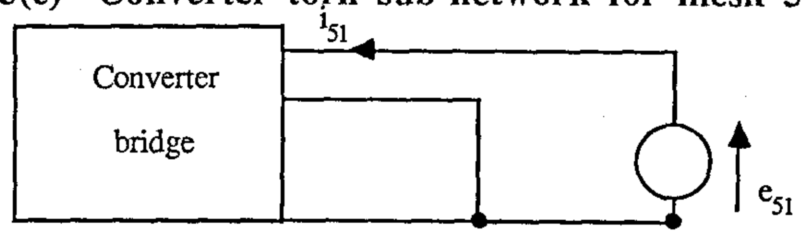



Fig 6.5(f) Converter torn sub-network for mesh 6

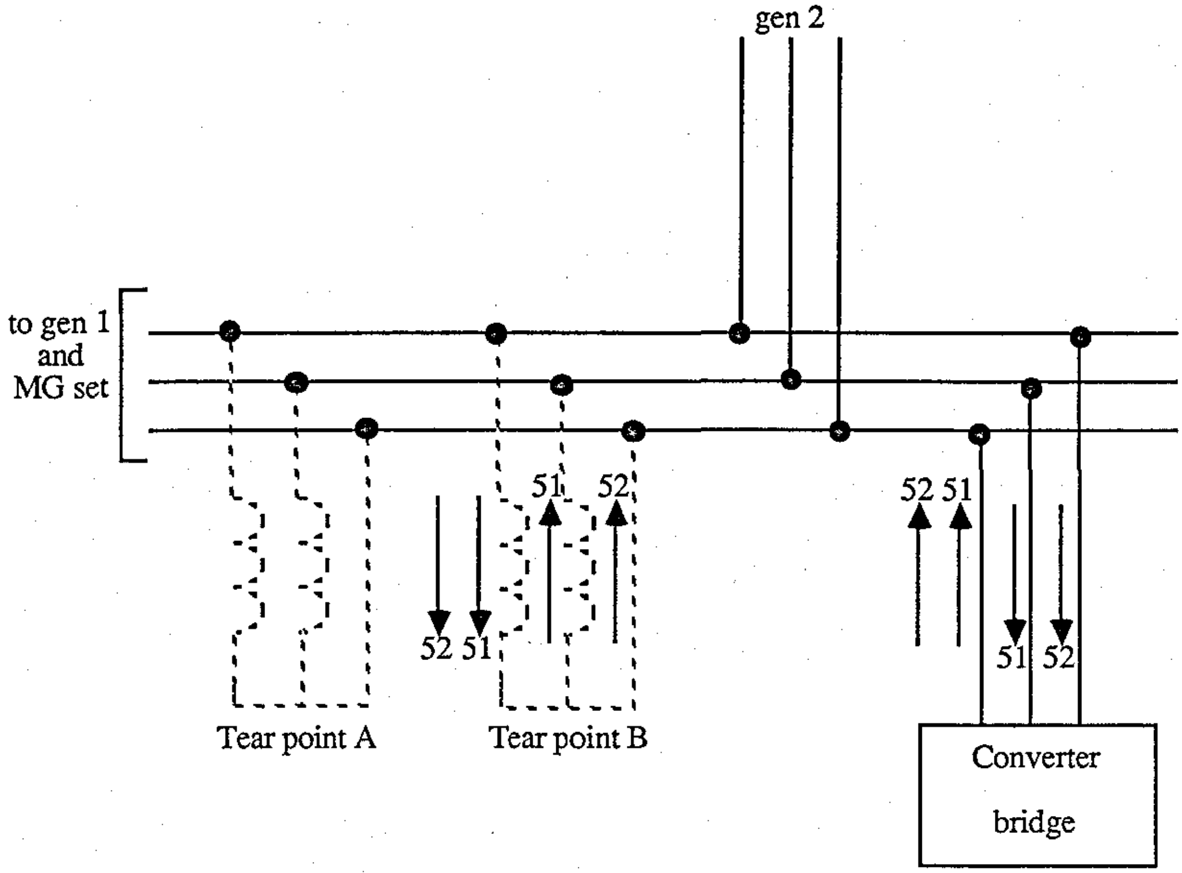



**Fig 6.6(a) Current flow in converter when thyristor 1,6 and 2 are conducting**

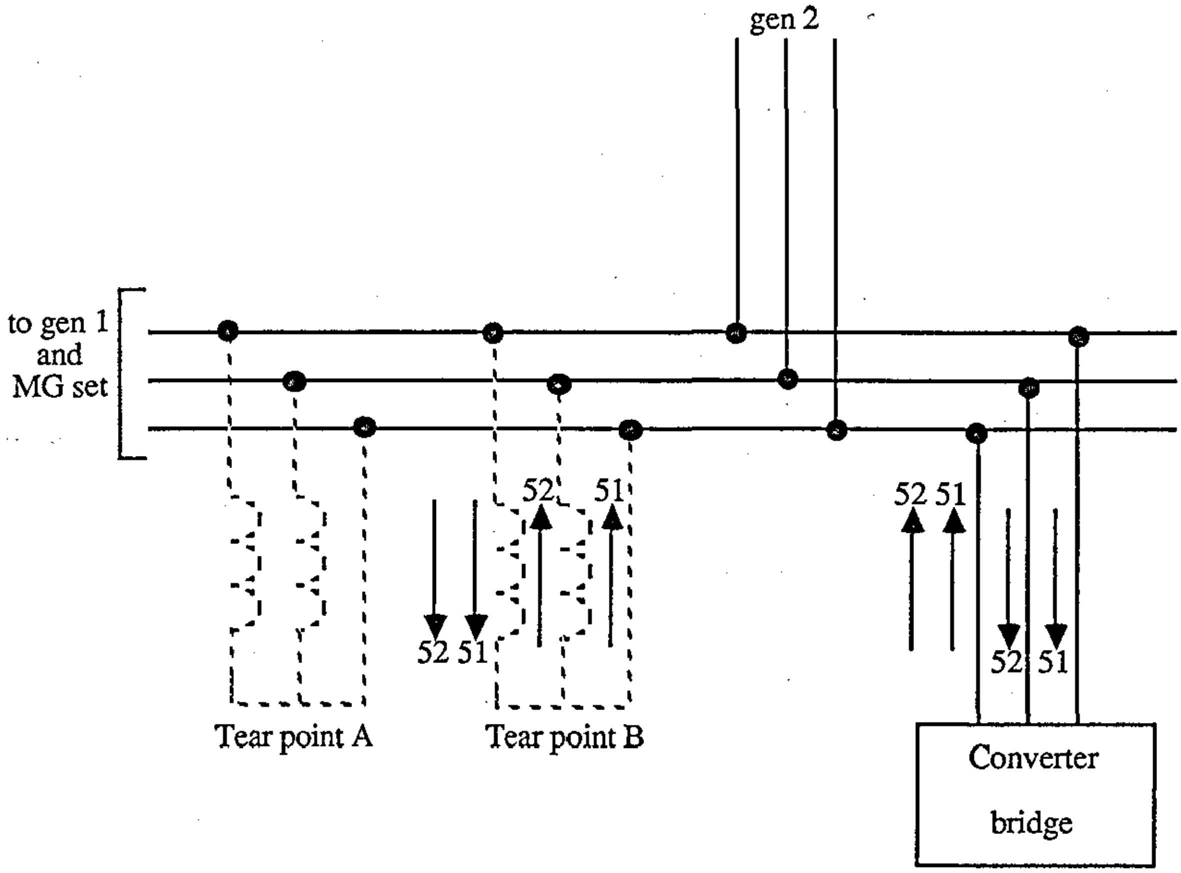



**Fig 6.6(b) Current flow in converter when thyristor 1,9 and 2 are conducting**

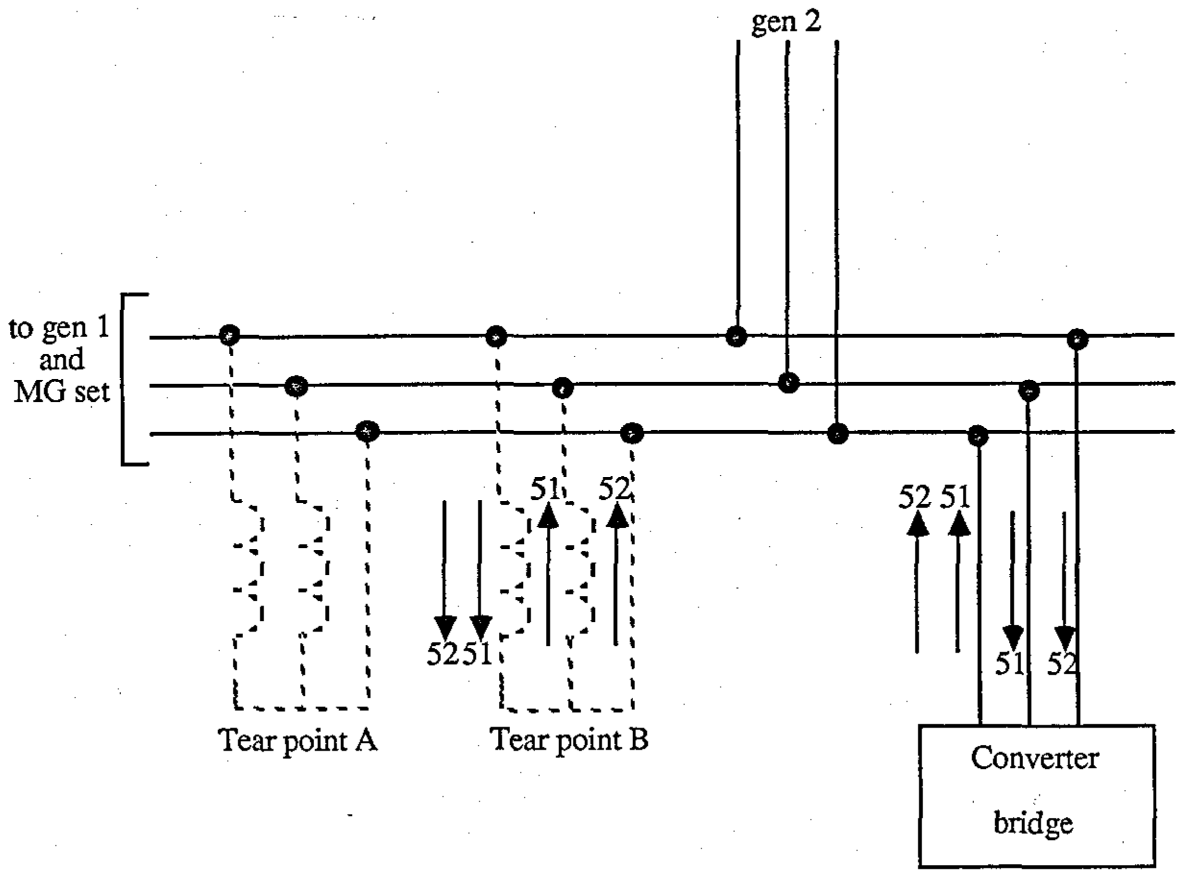



**Fig 6.6(c) Current flow in converter when thyristors 1, 3 and 2 are conducting**

# Chapter 7

## SIMULATION RESULTS FOR THE SHIP'S POWER SYSTEM

In this chapter, the overall performance of the ship's power supply system is examined. To illustrate the various switching conditions which are considered, a simplified representation of the system is given in Fig 7.1. Simulation results are presented as waveforms of voltage and current at different points of the system.

## 7.1 The system performance on generation

### 7.1.1 Converter Rectification

Fig 7.2 shows the result of a simulation in which the converter has a constant trigger angle of $30^0$ and rated load at 0.9pf lag is applied suddenly to the DC side. The supply is provided by synchronous generator 1, with switches 1, 4 and 5 of Fig 7.1 being closed and switches 2 and 3 open.

It follows from the defined switching conditions that power is fed from the synchronous generator to the converter through the bus bar. Fig 7.2(a) shows the waveforms obtained for the load current and voltage of the converter. As expected, the load voltage has a a six-pulse characteristic and a mean voltage of 220V. The load impedance is mainly resistive and the load current is therefore almost cophasal with the load voltage waveform. The converter AC side line currents and voltages are shown in Fig 7.2(b) and (c) respectively. Due to the sequential switching of thyristors 1 to 6, the line voltage is somewhat distorted with voltage spikes occurring in the voltage waveform.

Fig 7.2(d) and (e) show the generator line current and voltage waveforms. The defined switching conditions cause the converter line currents to be identical with those of the source (generator 1), so that identical current waveforms are obtained. However, voltage spikes occur in the line voltages shown in Fig 7.2(e). Fig 7.2(f) and (g) show respectively the bus bar line currents and voltages.

### 7.1.2 The motor/generator set

Fig 7.3 shows various circuit waveforms obtained when the motor/generator set is supplied by synchronous generator 2, with switches 2, 3 and 4 closed and switches 1 and 5 open.

Power from the generator is fed to the MG set via the bus bar. Figs 7.3(a) and (b) show the sinusoidal waveforms obtained for the line currents and voltages of the generator. Fig 7.3(c) and (d) show waveforms obtained for the line currents and voltages of the MG set, which are identical with those of the generator.

## 7.2 System performance on re-generation

### 7.2.1 Converter Inversion

Two inversion mode tests were made to illustrate the system performance with the converter operating in the PWM manner. Generator 1 of Fig 7.1 is held stationary to form a static load of rated kVA at 0.8pf lag on the 3-phase system, with SW1, SW4 and SW5 of Fig 7.1, closed and SW2 and SW3 open. The relevant data is;

DC input voltage = 400V

Carrier wave of amplitude = 10V

Reference wave of amplitude = 5V

Carrier wave frequency = 800Hz

Reference wave frequency = 40Hz

The resulting circuit waveforms are presented in Fig 7.4.

From the waveforms of the converter line currents and voltages shown in Fig 7.4(a) and (b), it will be seen that the line voltage contains approximately 20 pulses per half cycle, while Fig 7.4(b) makes clear that three PWM voltage waveforms, each with an amplitude of 400V and mutual phase shift of $120^0$, are obtained at the converter output. Fig 7.4(c) and (d) show waveforms obtained for the line currents and voltages in generator 1, with the direct connection between the generator and the converter causing the generator and converter currents to be identical.

A second test with the same parameters, but with the carrier frequency increased to 2kHz, gave the circuit waveforms shown in Fig 7.5. The number of pulses per half cycle of the converter line voltages and currents, shown in Figs 7.5(a) and (b) are greater than in the first test, with a narrower pulse width due to the increase in carrier frequency. Fig 7.5(c) and (d) show the generator 1 line currents and voltages, which are almost cophasal since the load is substantially resistive.

## 7.3 System performance on load application

A test was performed in which switches SW1, SW3 and SW4 were initially open and switches SW2 and SW5 were closed and power was directed to the converter from generator 2. After steady state had been achieved, switches SW3 and SW4 were closed to direct power to both the converter and the MG set from generator 2.

When current is supplied only to the converter, generator 2 and the converter current waveforms are identical (See Fig 7.6), and when the MG set is connected after 0.1 sec, generator 2 line current increases as shown in Fig 7.6(a). The mean converter DC voltage and current both decrease following application of the MG set and the converter AC side line currents, shown in Fig 7.6(d), are reduced. The line currents and voltages of the MG set shown in Fig 7.7(e) and (f) both rises rapidly until a new steady state is reached.

## 7.4 Summary

In this chapter, various load conditions and switching permutations were investigated, in order to demonstrate the flexibility of the computer programme.

Tests performed during generation show that

(i) During rectification, a load voltage with a six-pulse characteristic was obtained.

(ii) When the MG set was fed from a generator, sinusoidal waveforms were obtained for both the line currents and voltages.

(iii) In the load application test, changing the switch conditions in the network enables the transient performance to be studied.

During re-generation; when power was directed to the generator from the converter:

(iv) A better-quality of PWM waveforms with a reduced harmonic content was obtained by increasing the carrier frequencies [8].

Overall, the system was shown to be highly flexible. Results from the computer model produced an entirely consistent performance in both generation and re-generation operation.

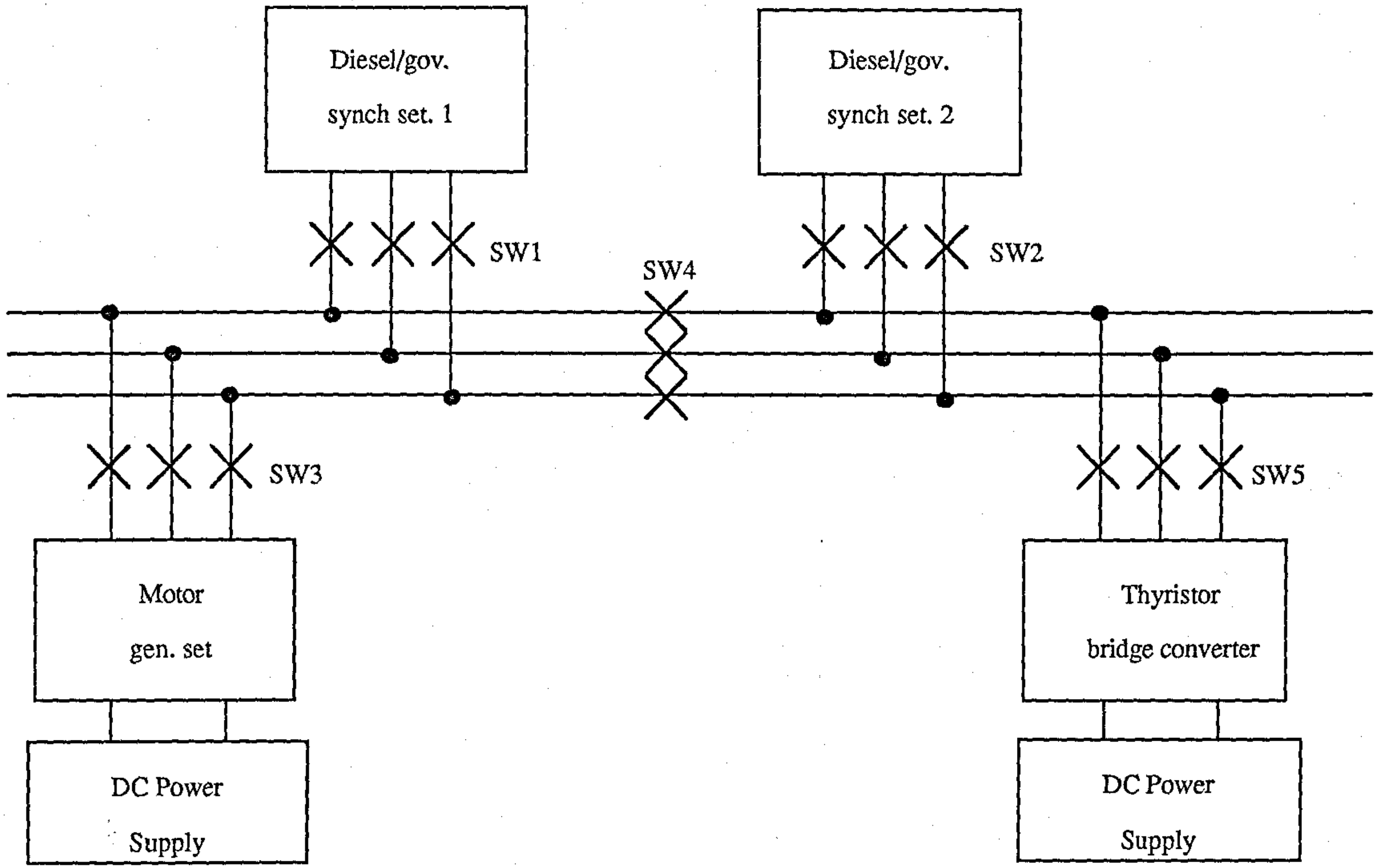



**Fig 7.1 Simplified representation of ship's power system**

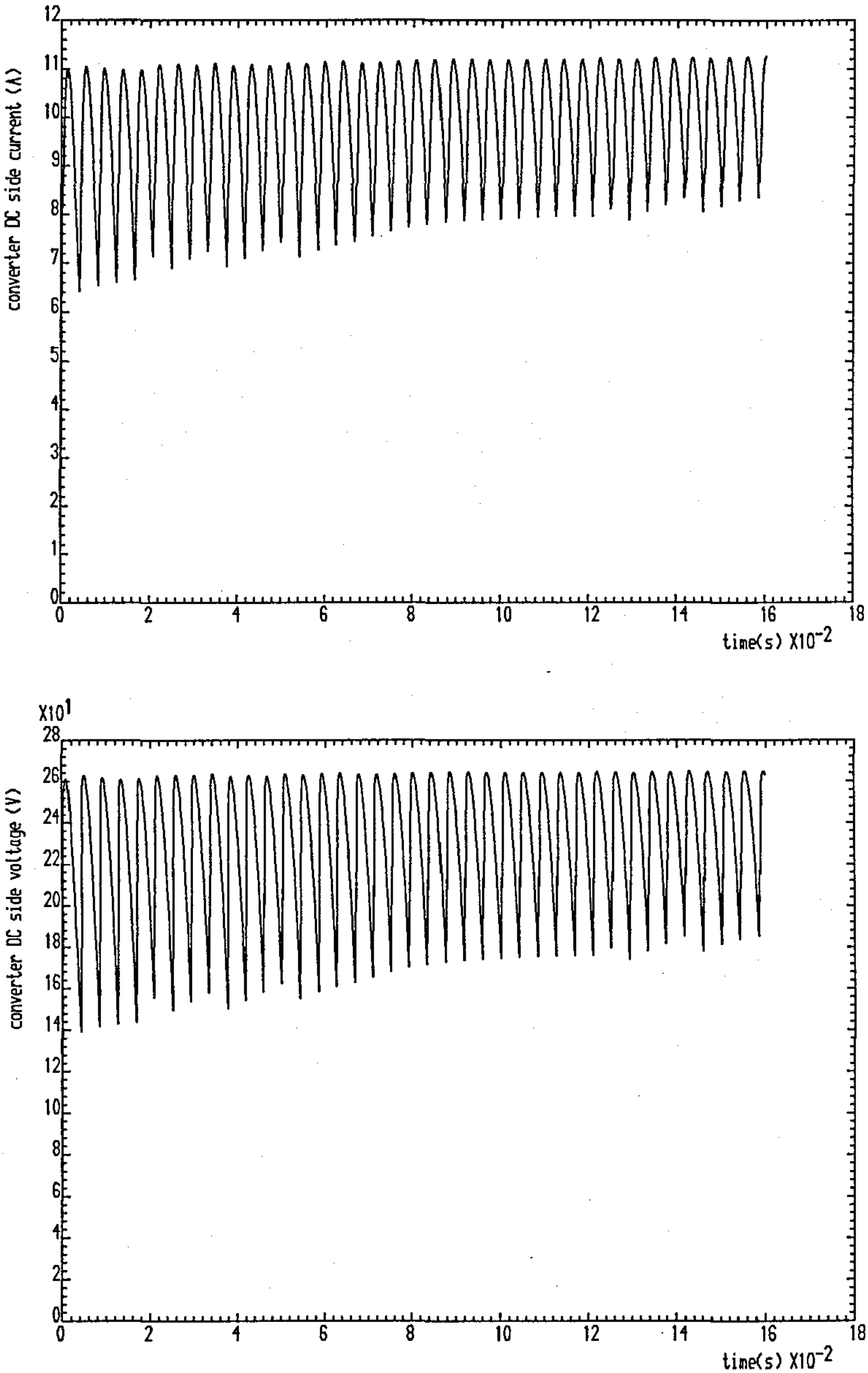



Fig 7.2(a) Converter response on generation

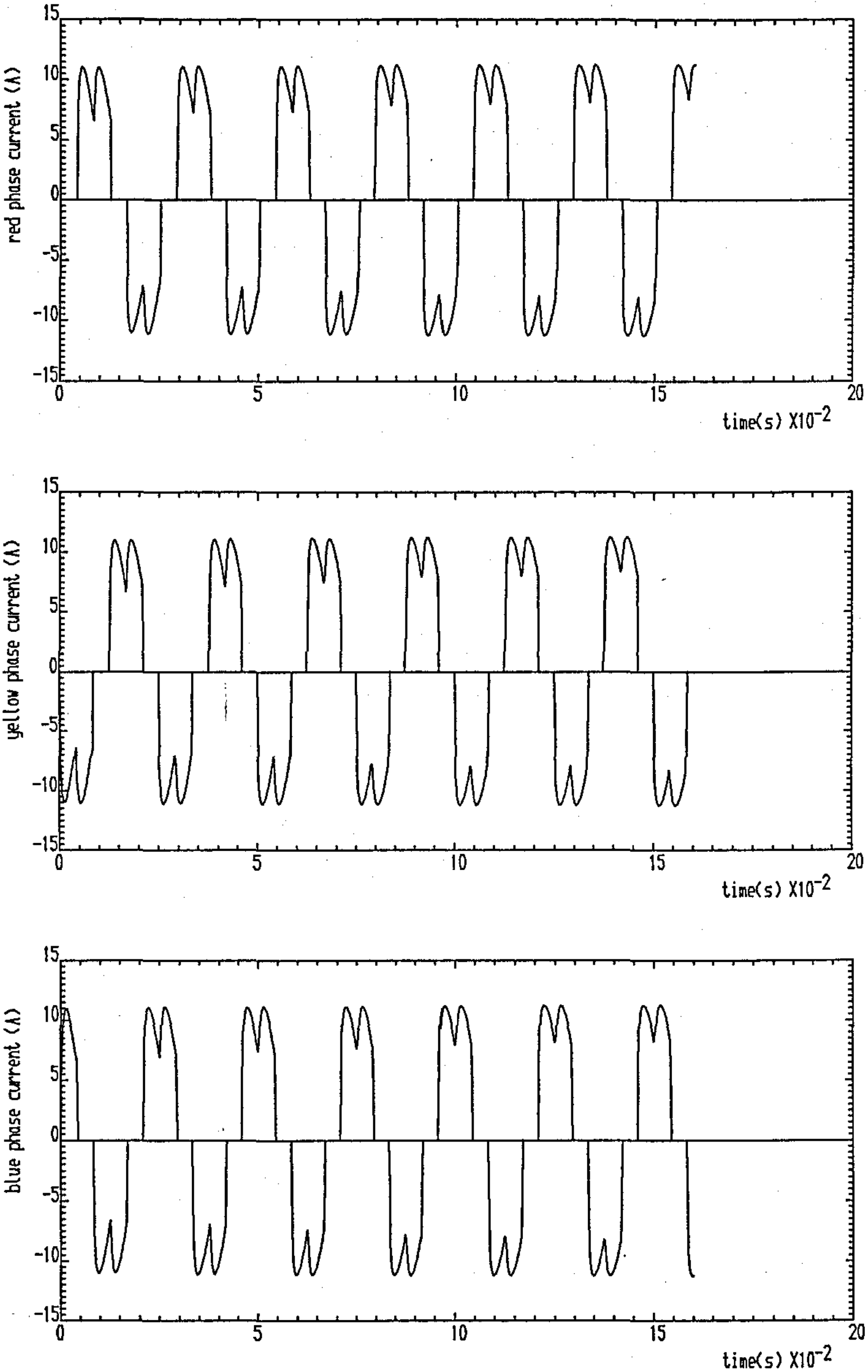



Fig 7.2(b) Converter AC side currents

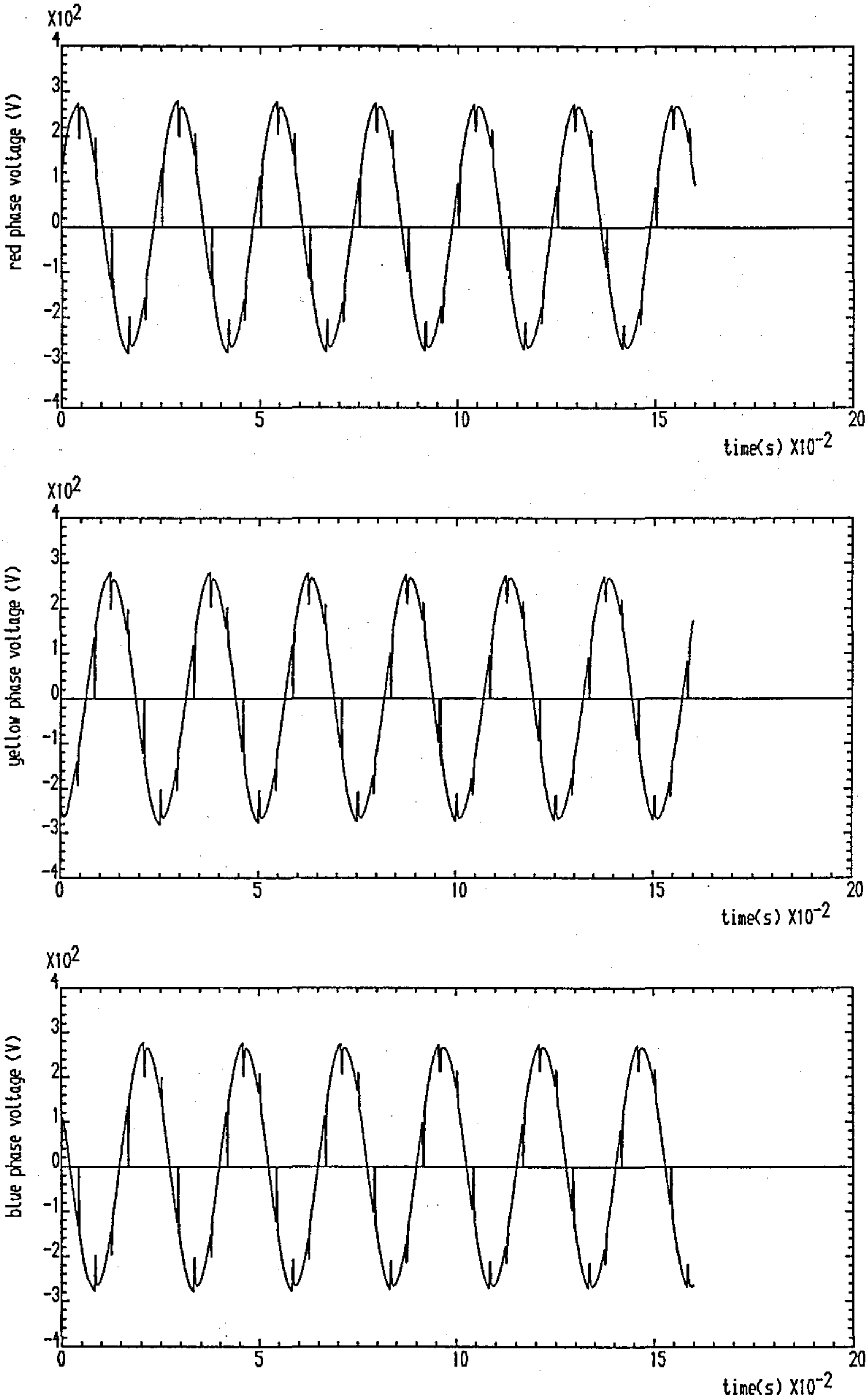



Fig 7.2(c) Converter AC side line voltages

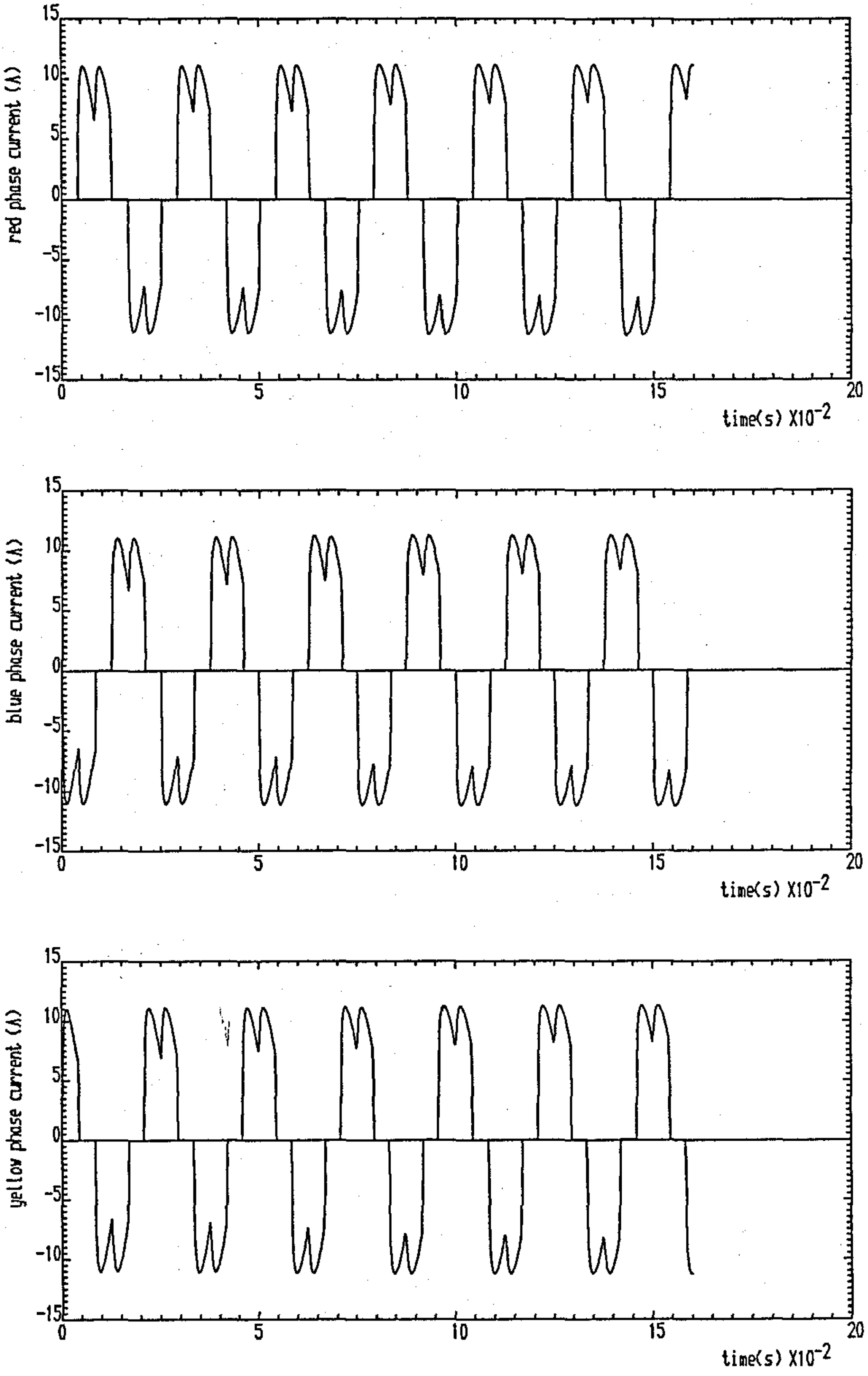



Fig 7.2(d) Generator 1 line currents

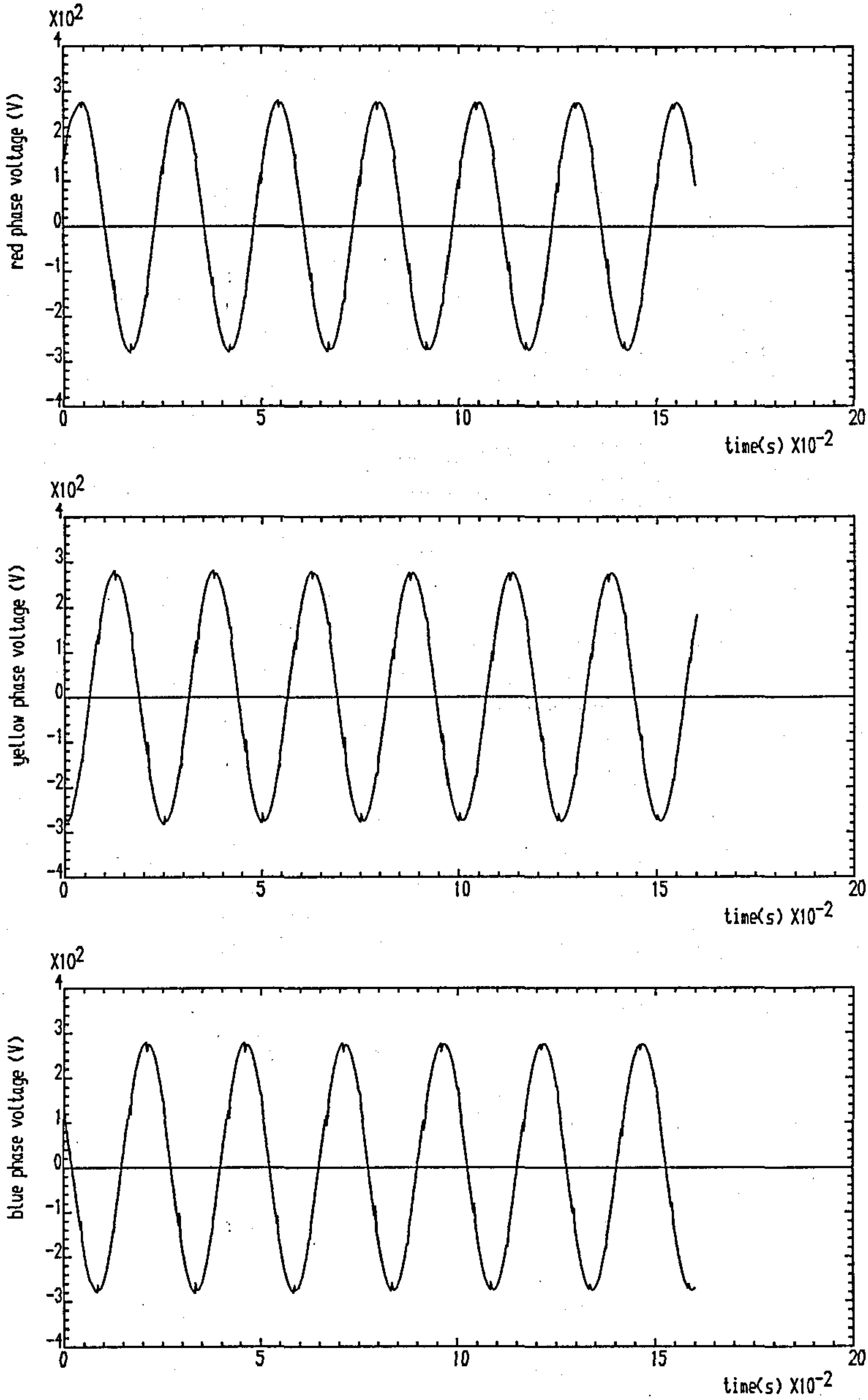



Fig 7.2(e) Generator 1 line voltages

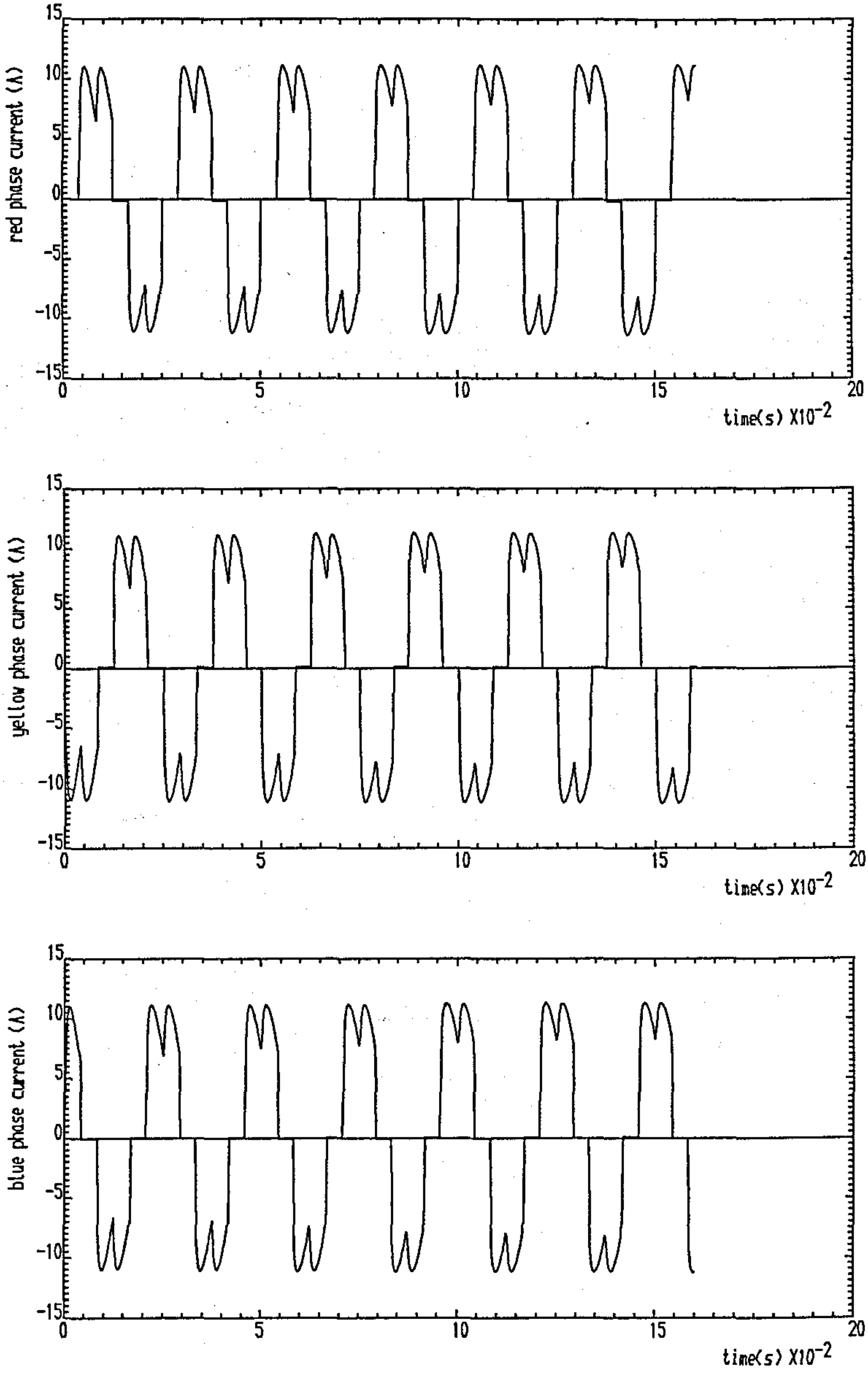



Fig 7.2(f) Bus coupler line currents

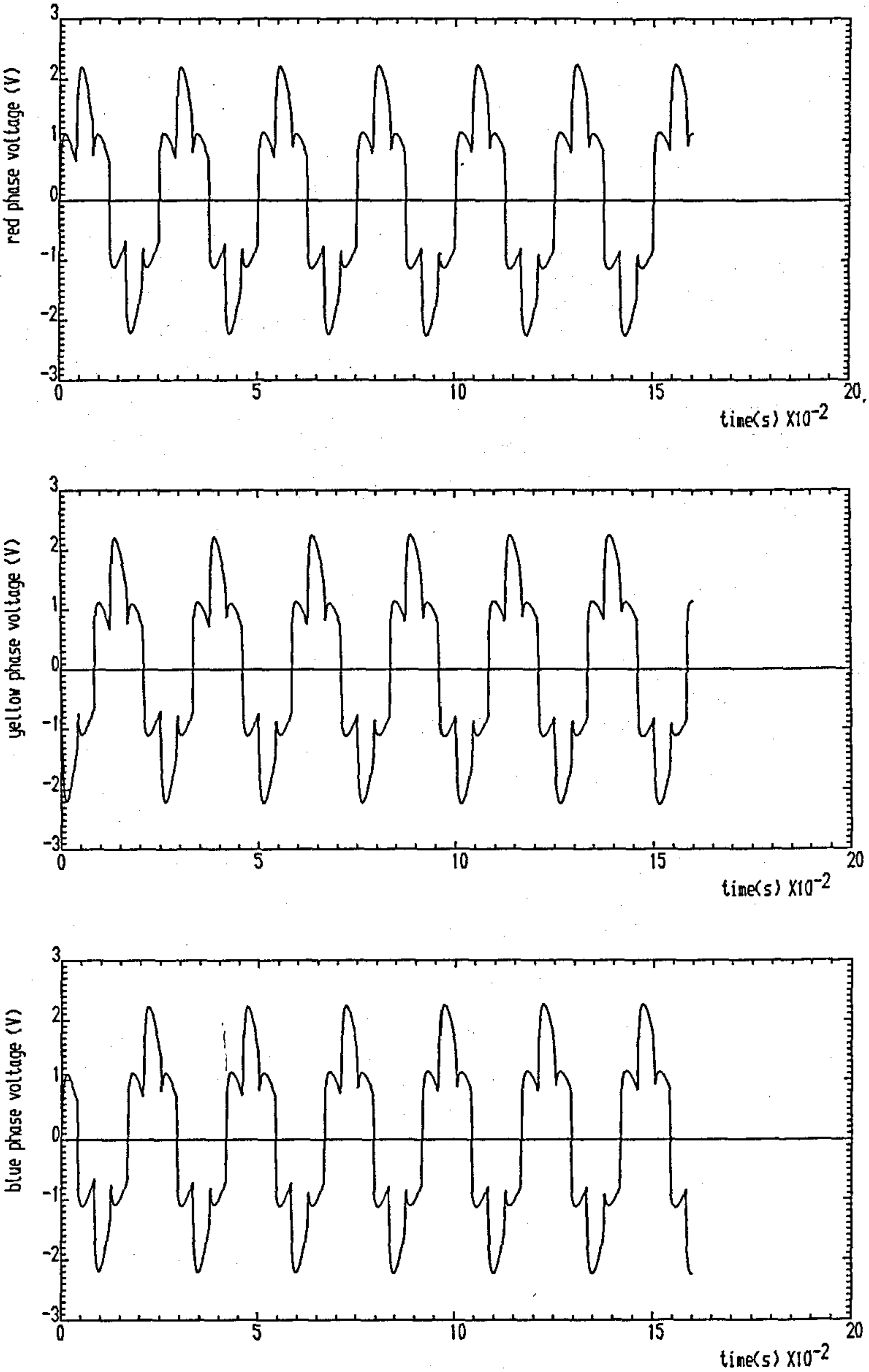



Fig 7.2(g) Bus coupler line voltages

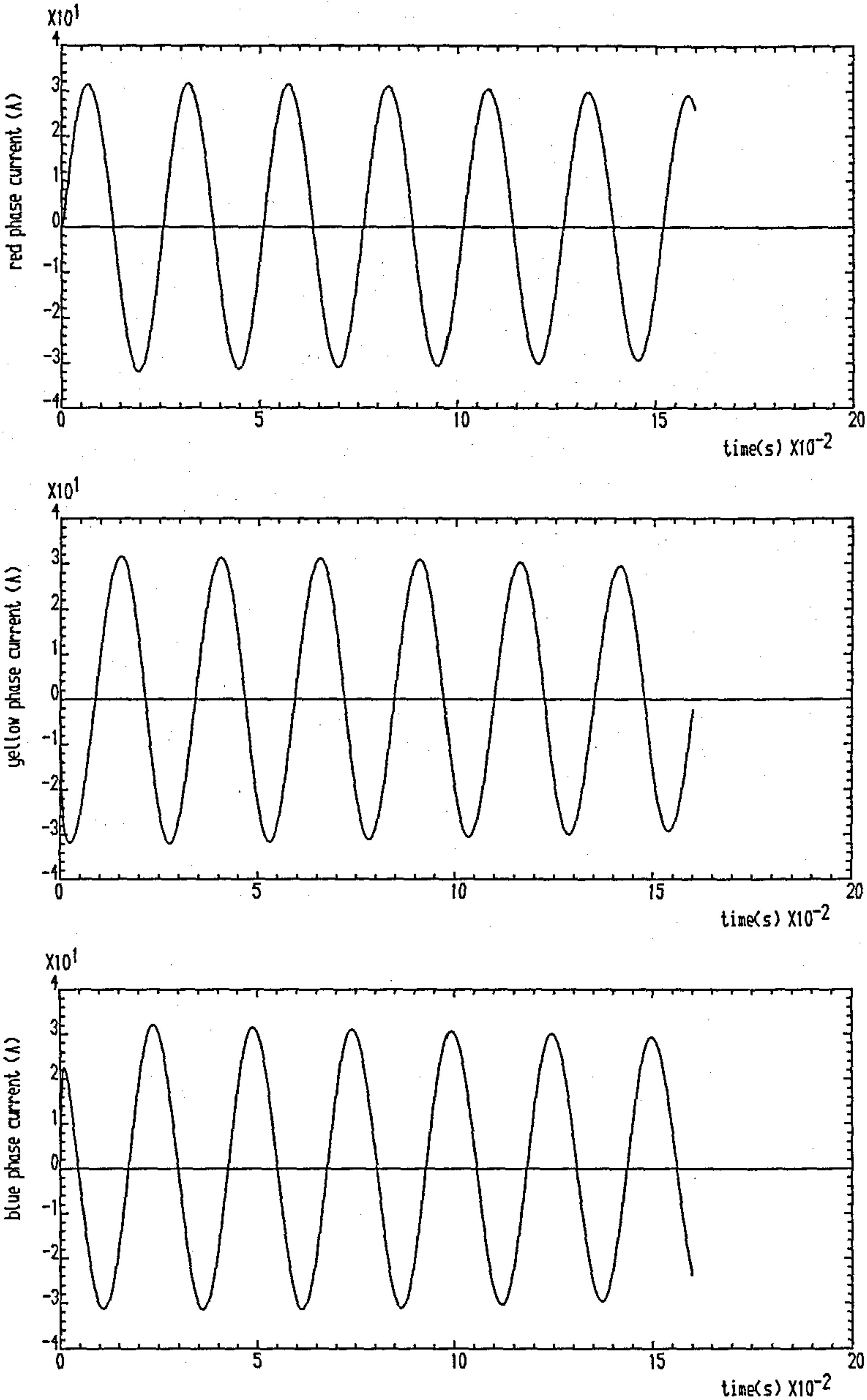



Fig 7.3(a) Generator 2 line currents

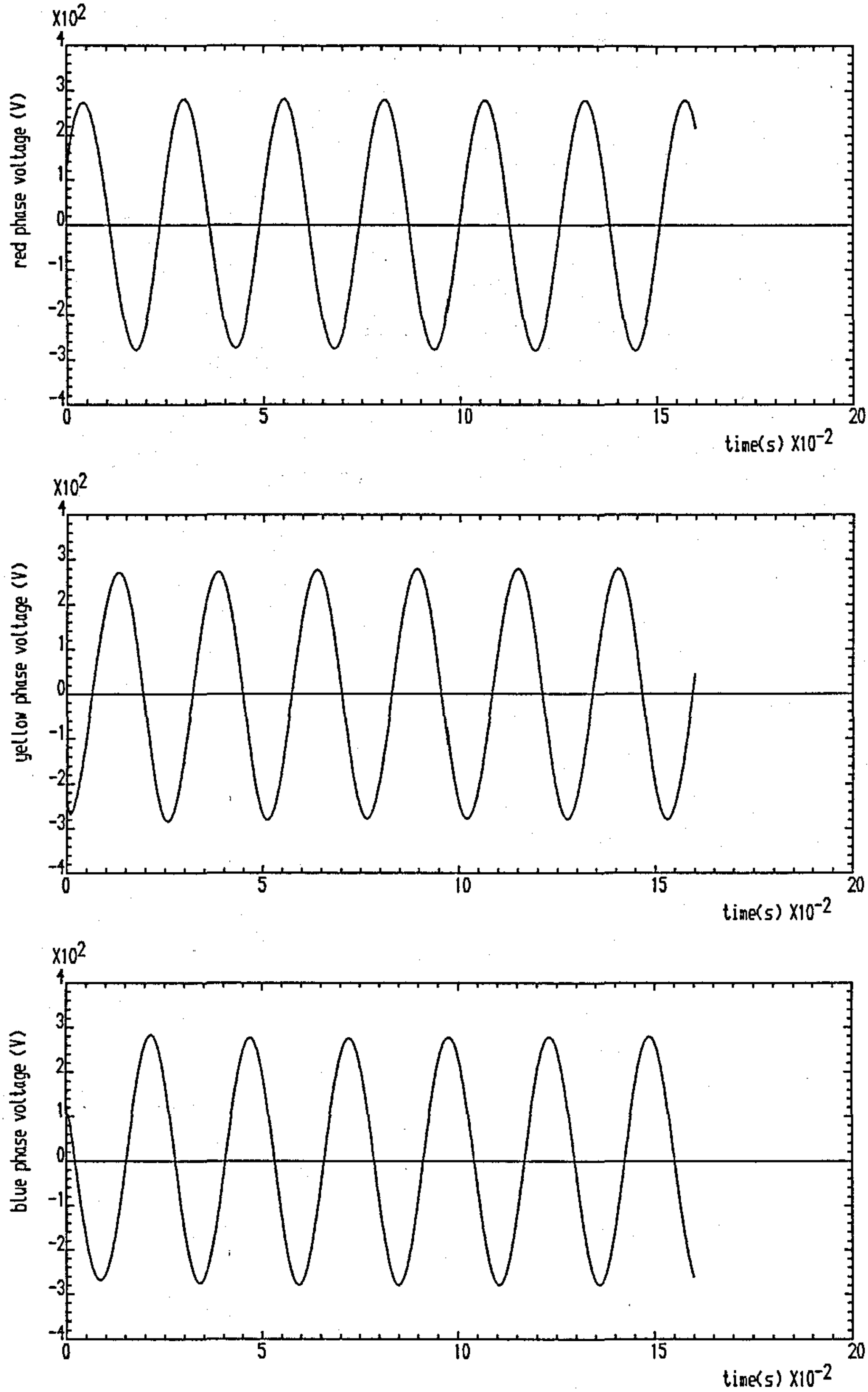



Fig 7.3(b) Generator 2 line voltages

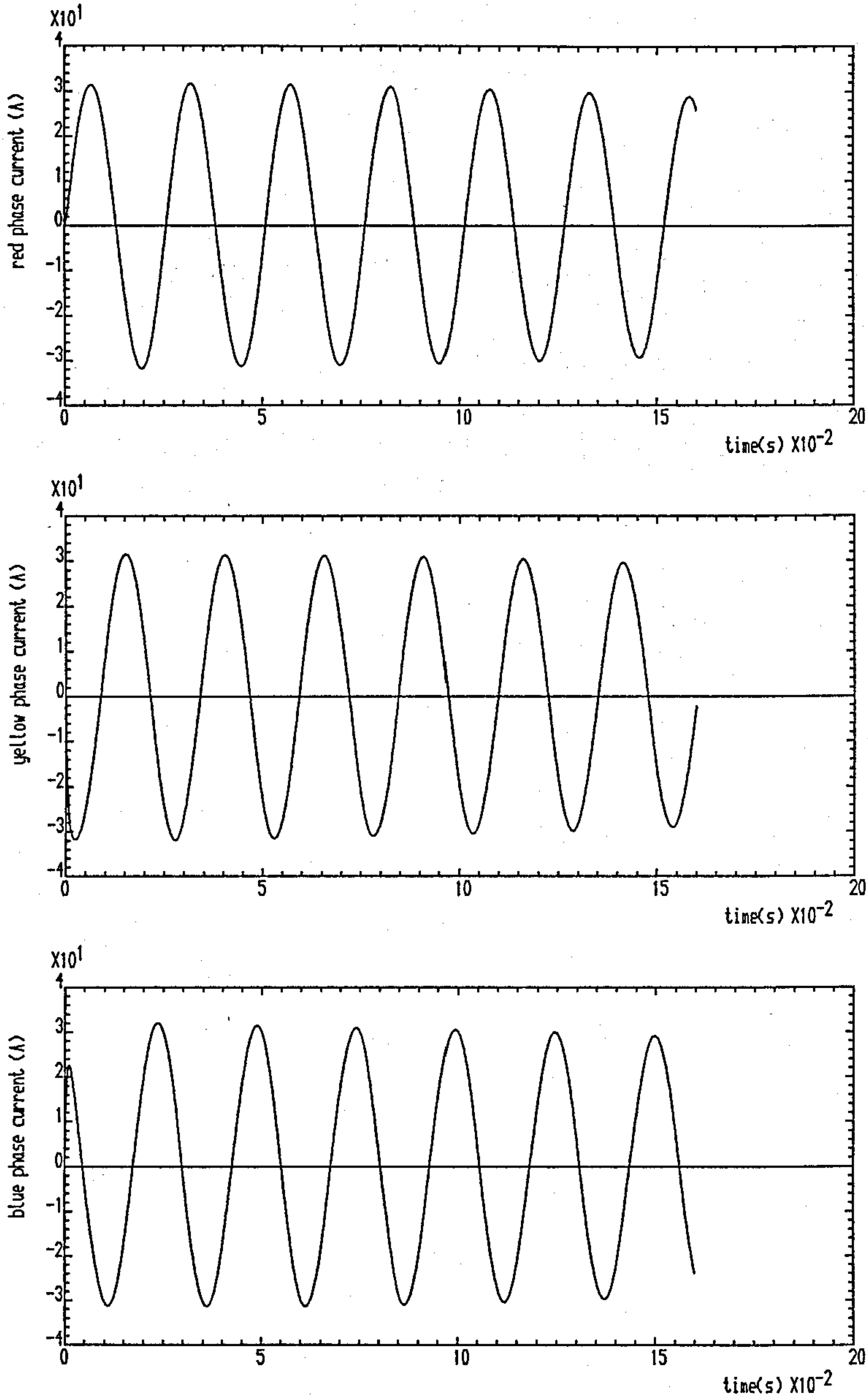



Fig 7.3(c) The MG set line currents

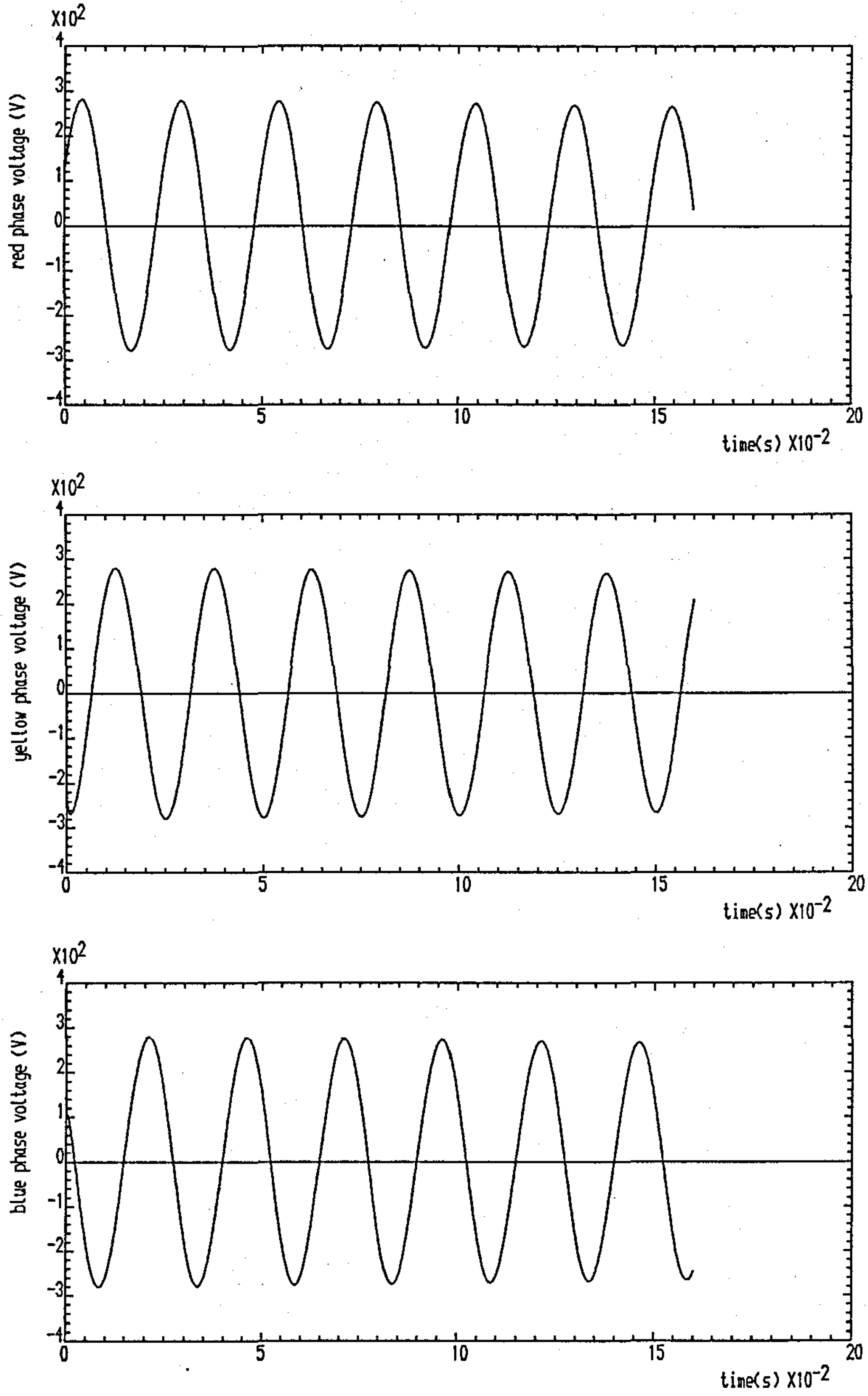



Fig 7.3(d) The MG set line voltages

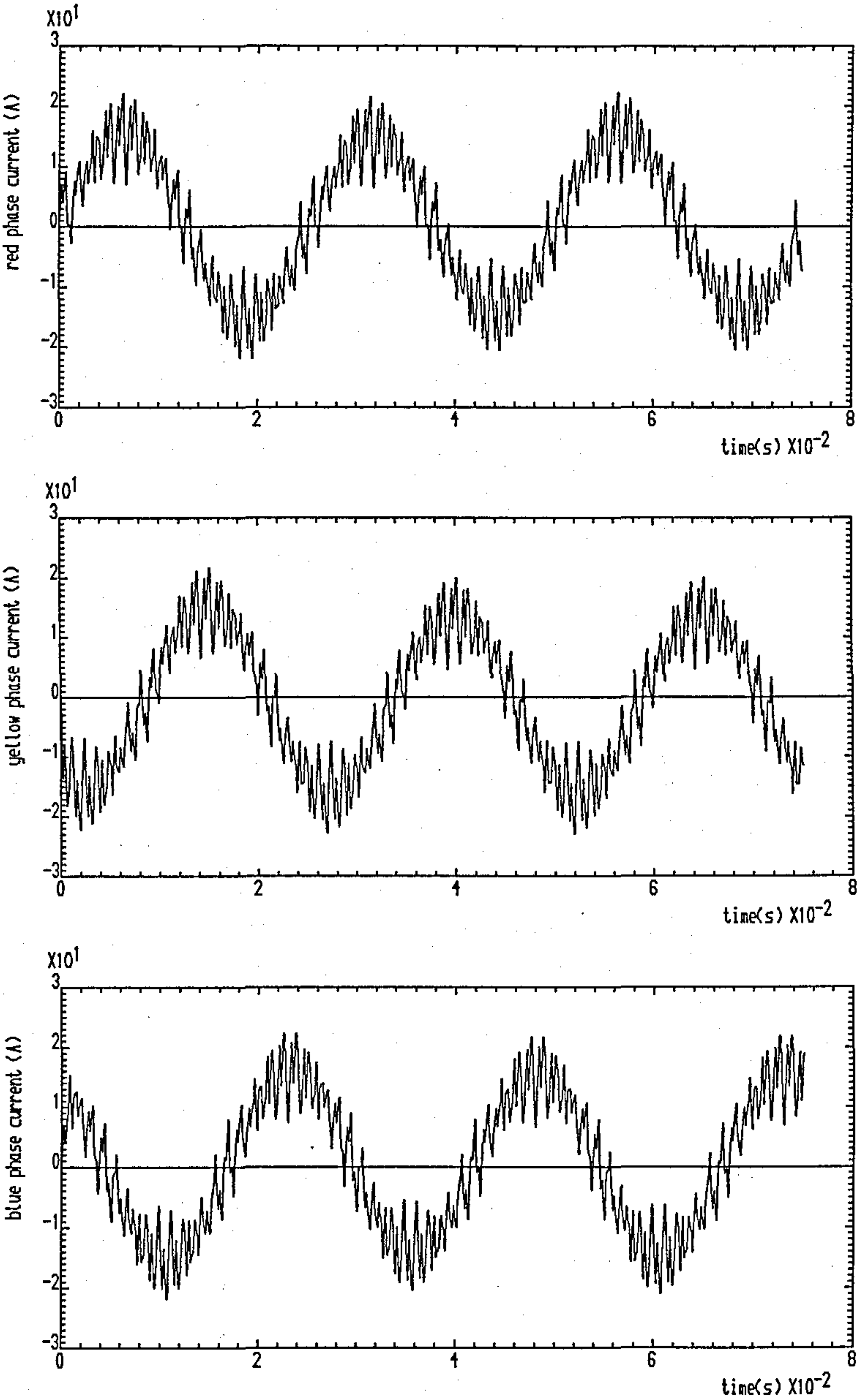



Fig 7.4(a) Converter line currents with carrier frequency of 800Hz

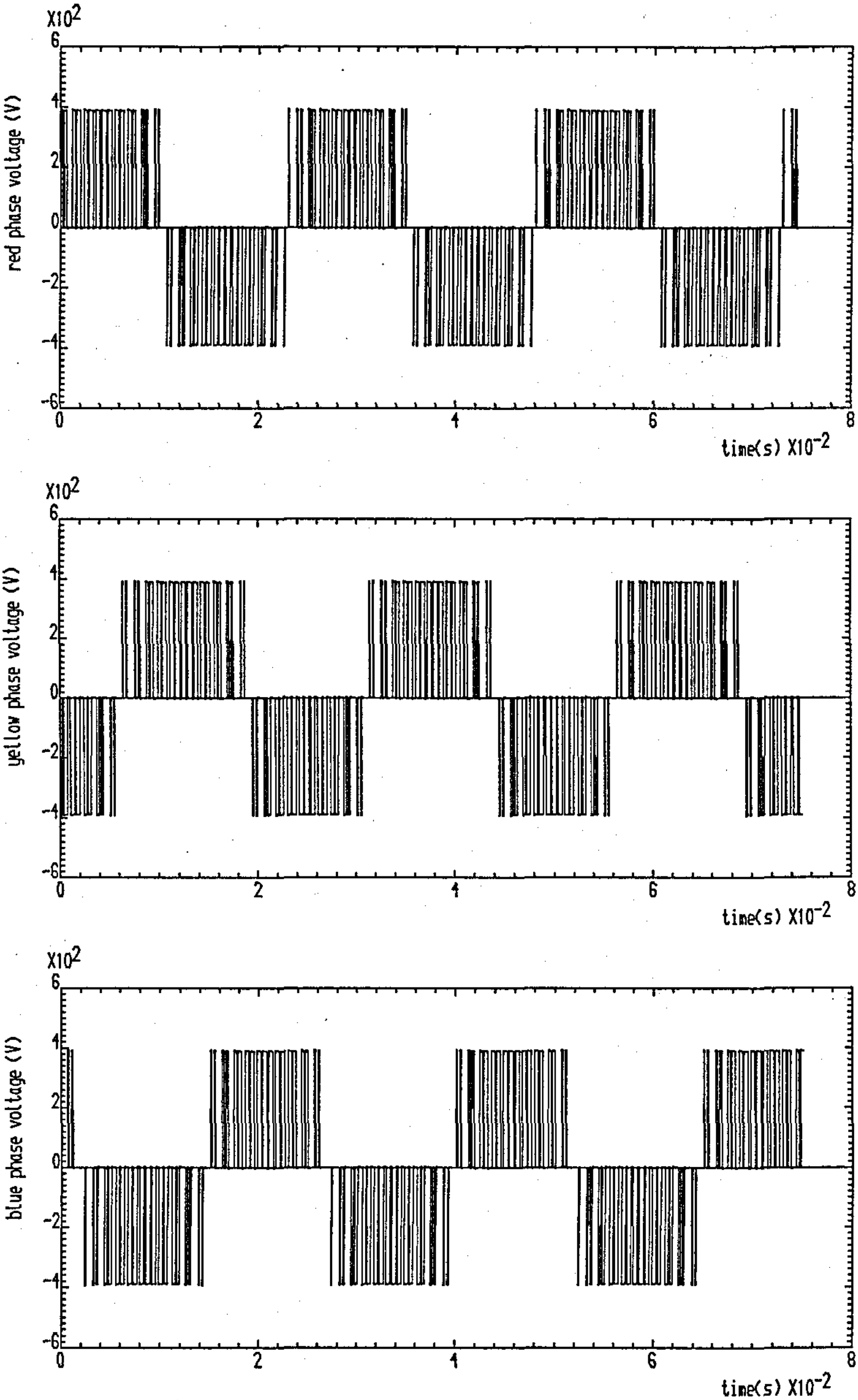



Fig 7.4(b) Converter line voltages with carrier frequency of 800Hz

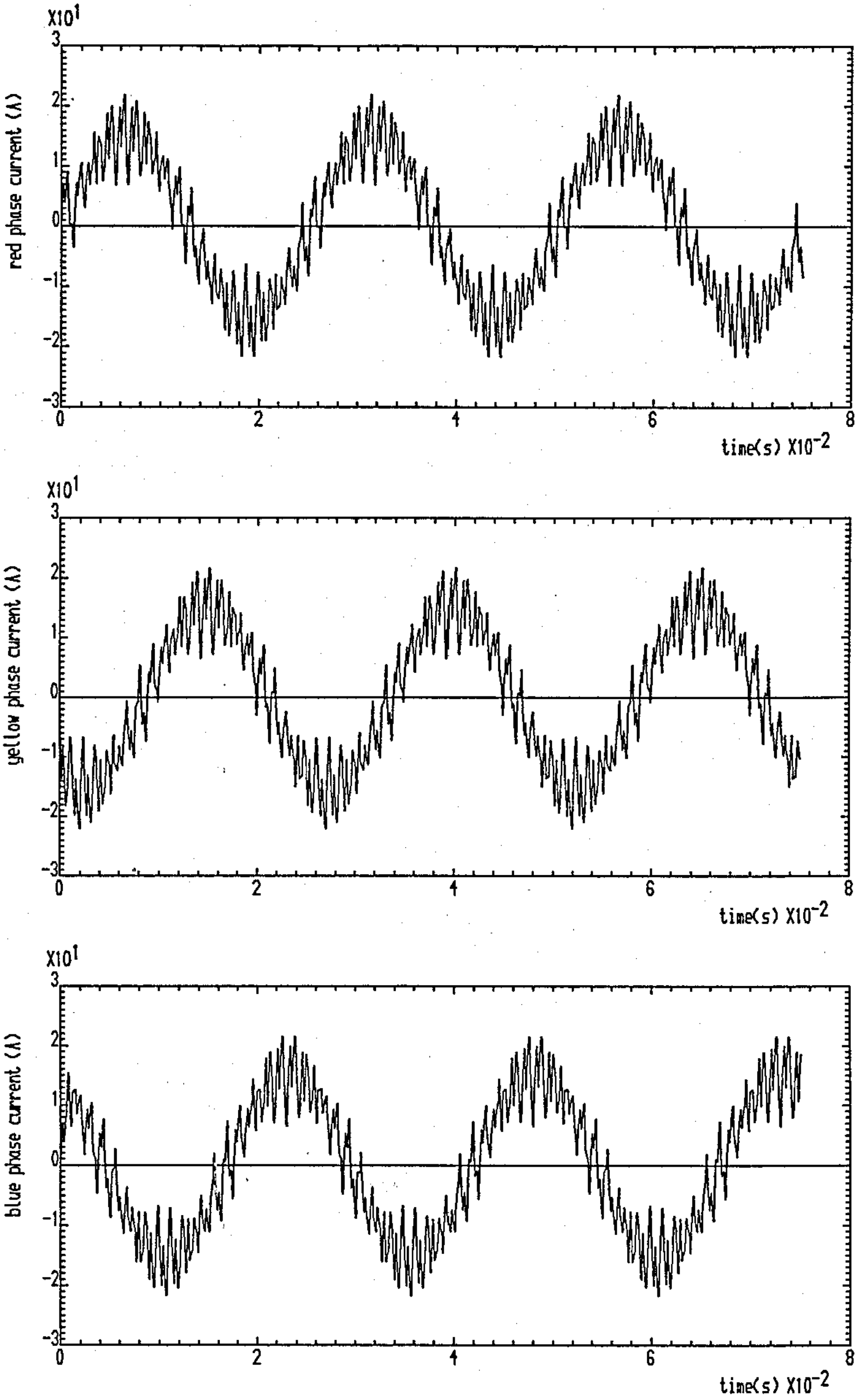



Fig 7.4(c) Generator 1 line currents with carrier frequency of 800Hz

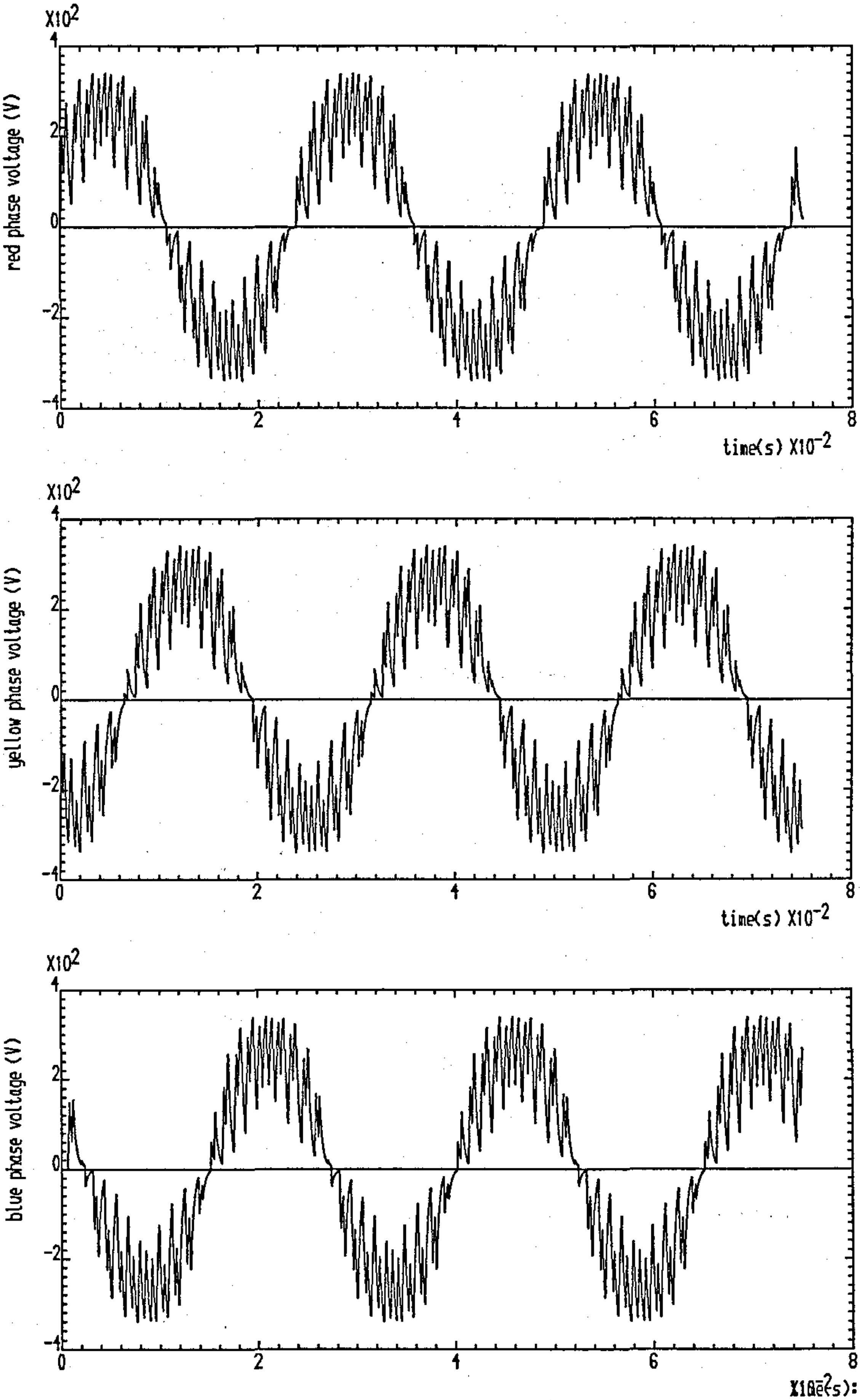



Fig 7.4(d) Generator 1 line voltages with carrier frequency of 800Hz

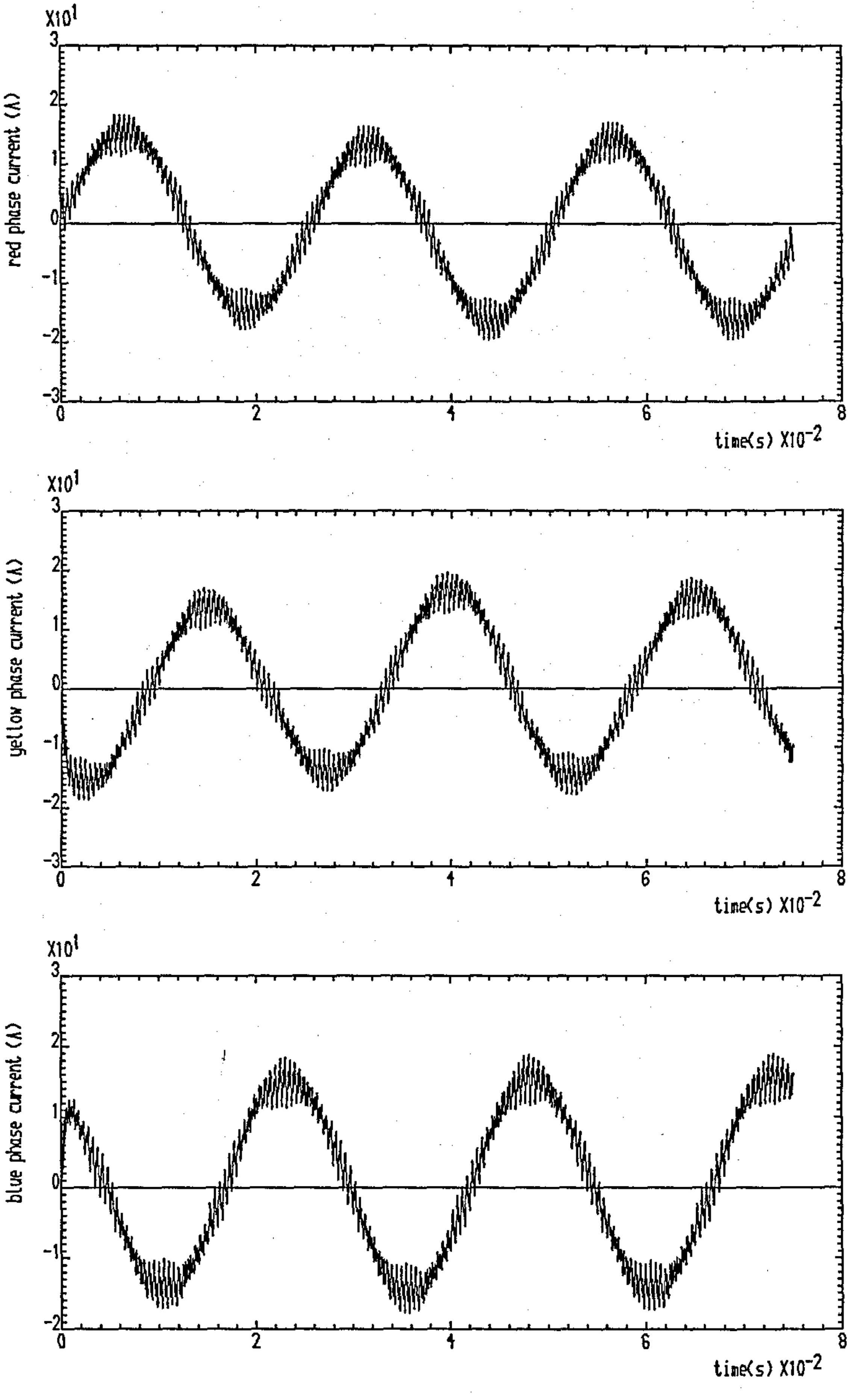



Fig 7.5(a) Converter line currents with carrier frequency of 2kHz

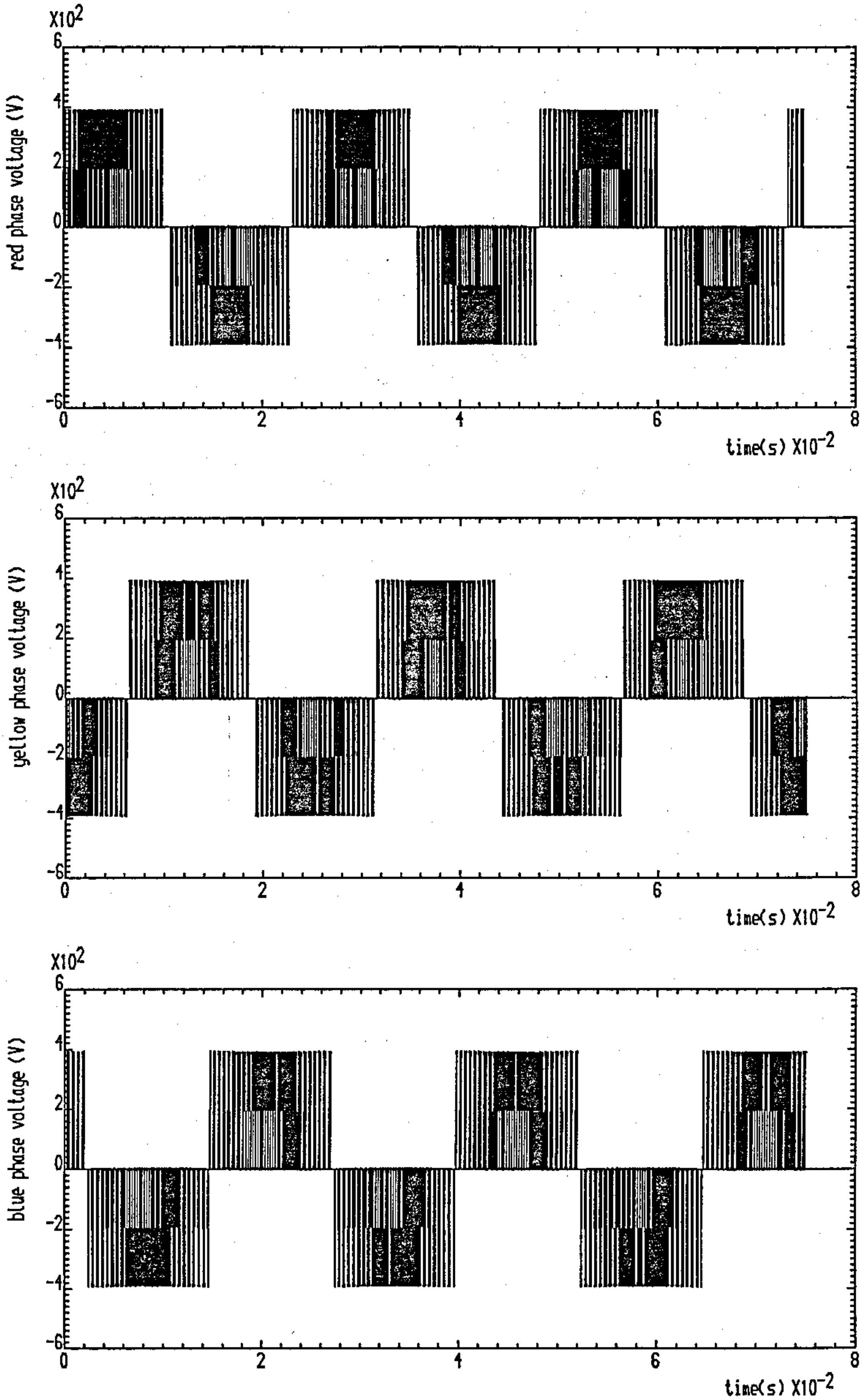


Fig 7.5(b) Converter line voltages with carrier frequency of 2kHz

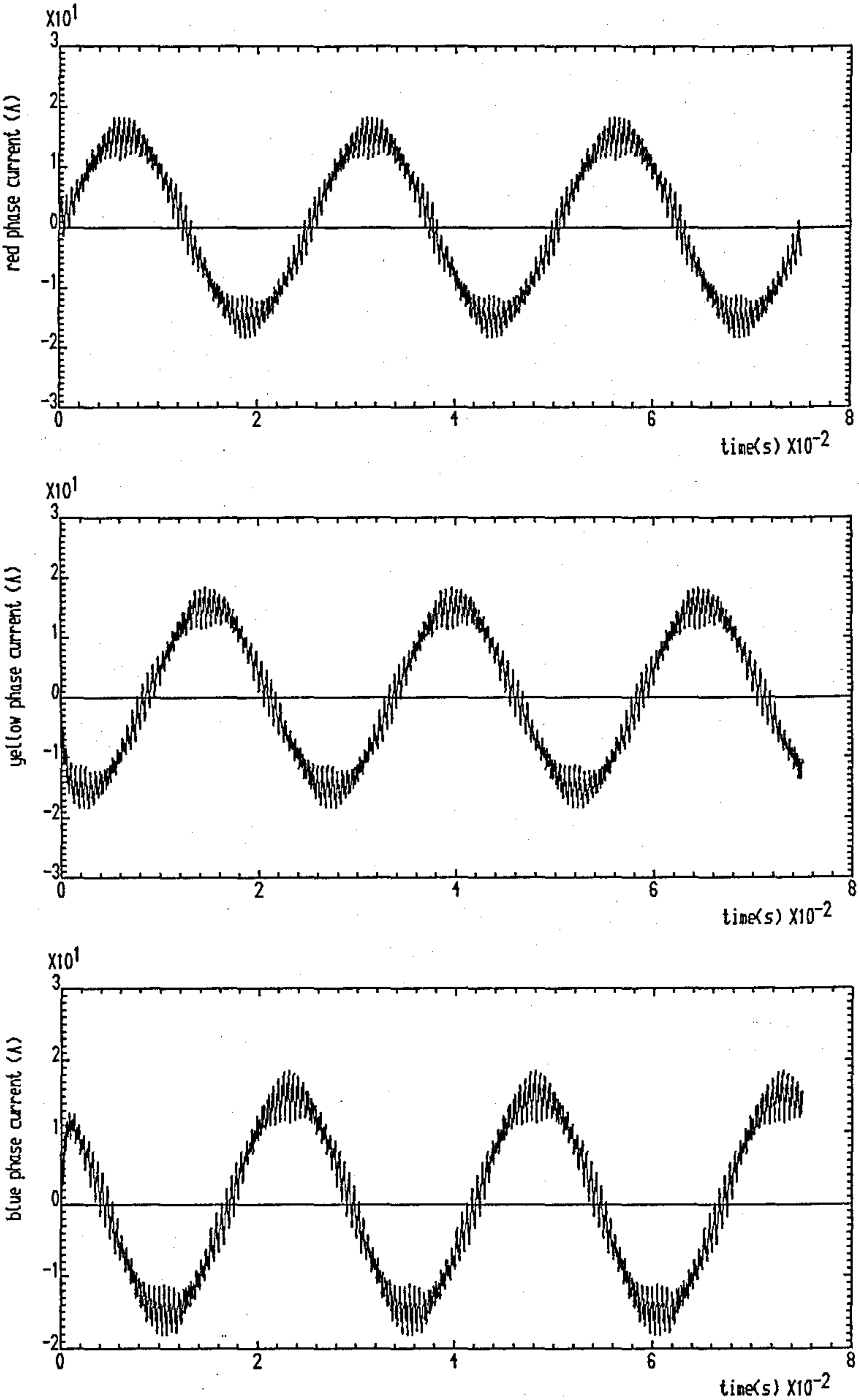



Fig 7.5(c) Generator 1 line currents with carrier frequency of 2kHz

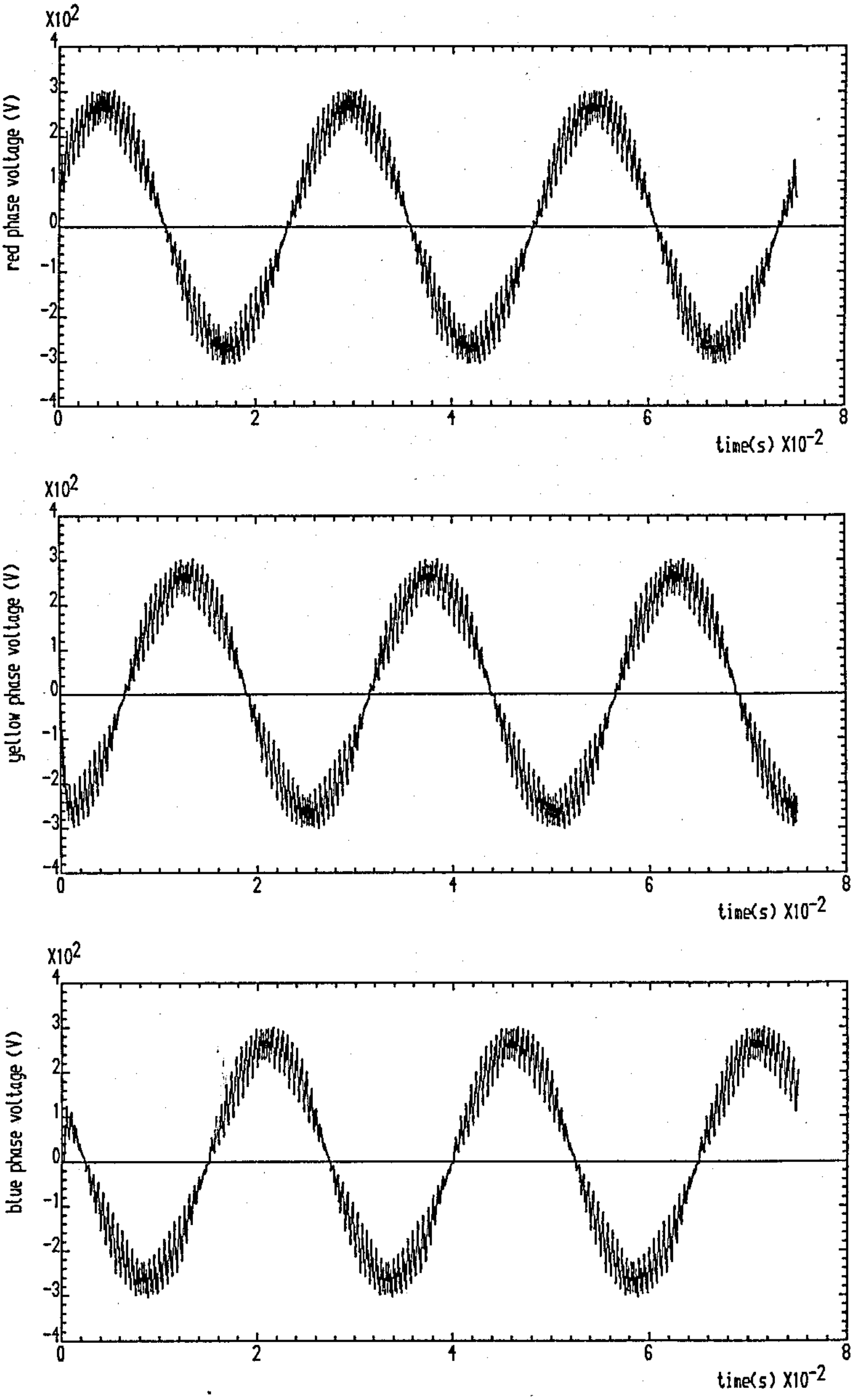



Fig 7.5(d) Generator 1 line voltages with carrier frequency of 2kHz

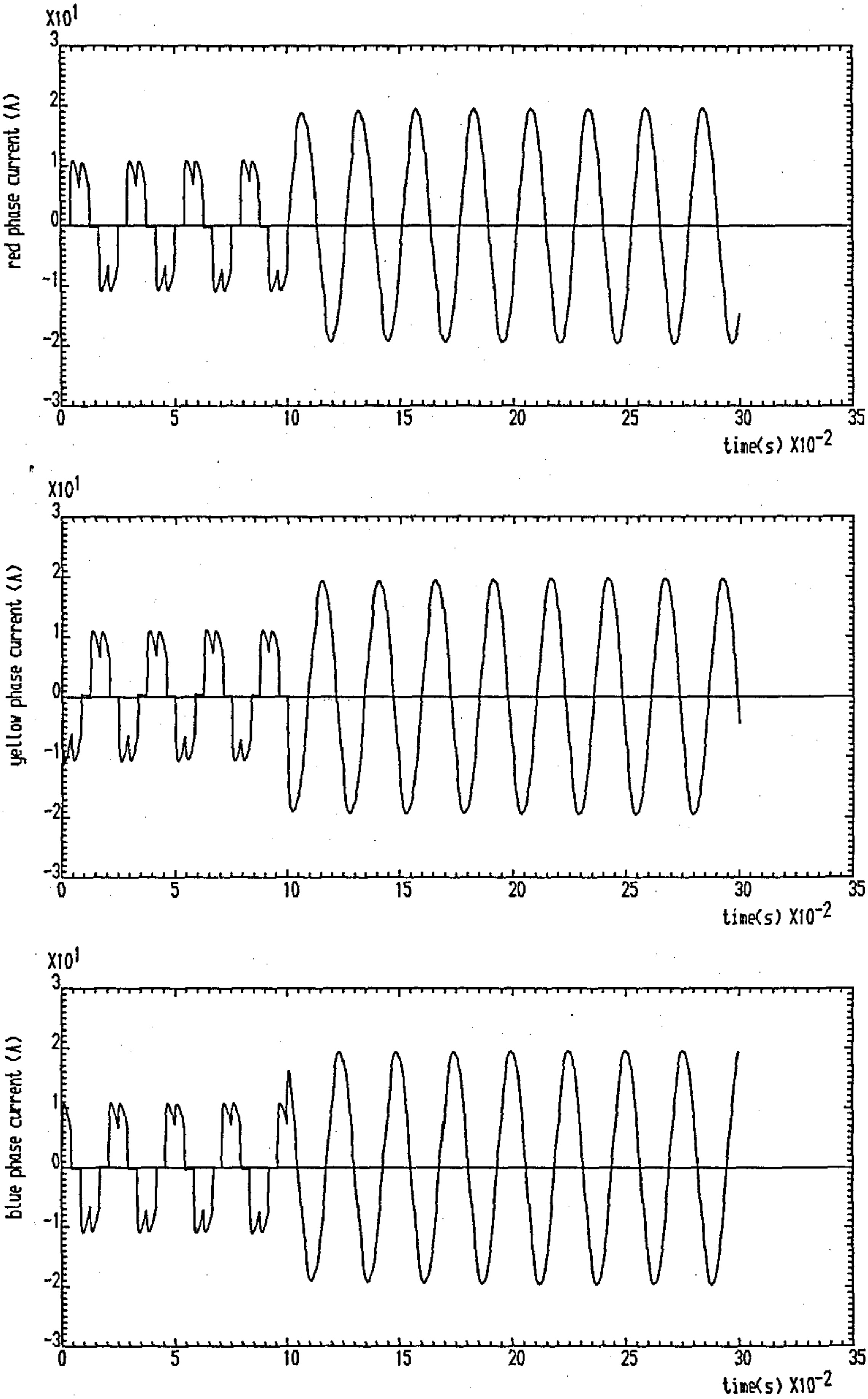



Fig 7.6(a) Generator 2 line currents

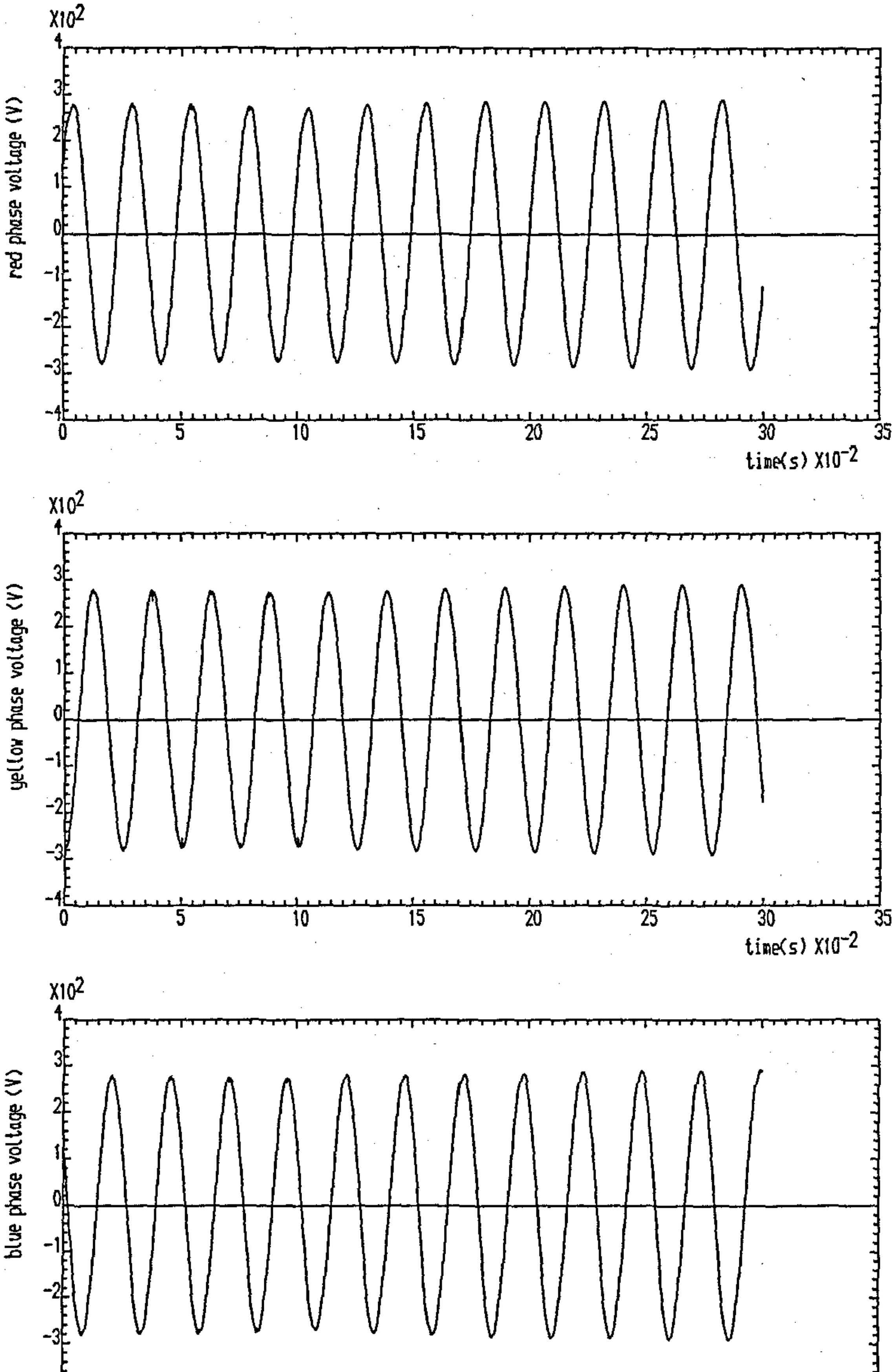



Fig 7.6(b) Generator 2 line voltages

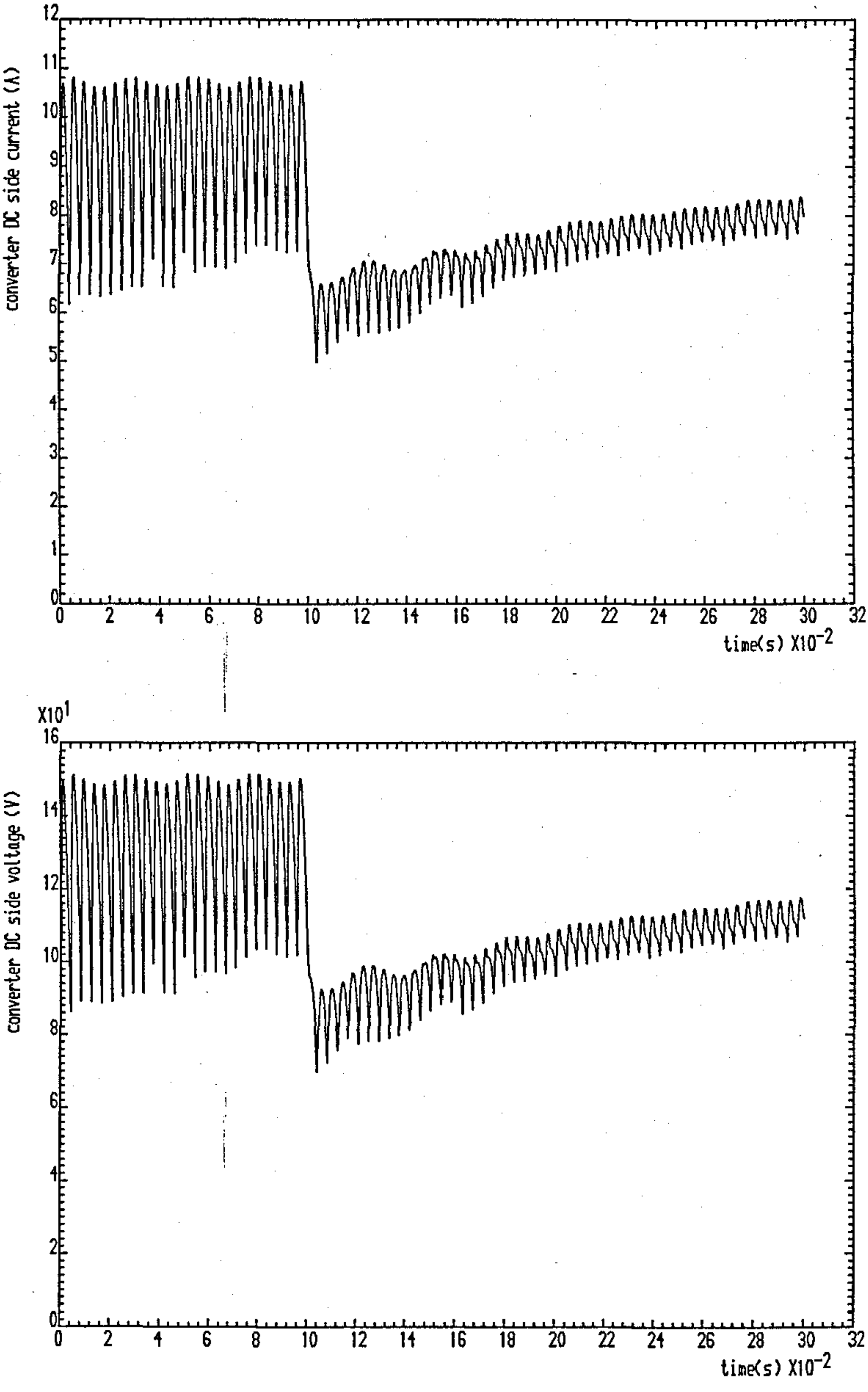



Fig 7.6(c) The converter DC side current and voltage

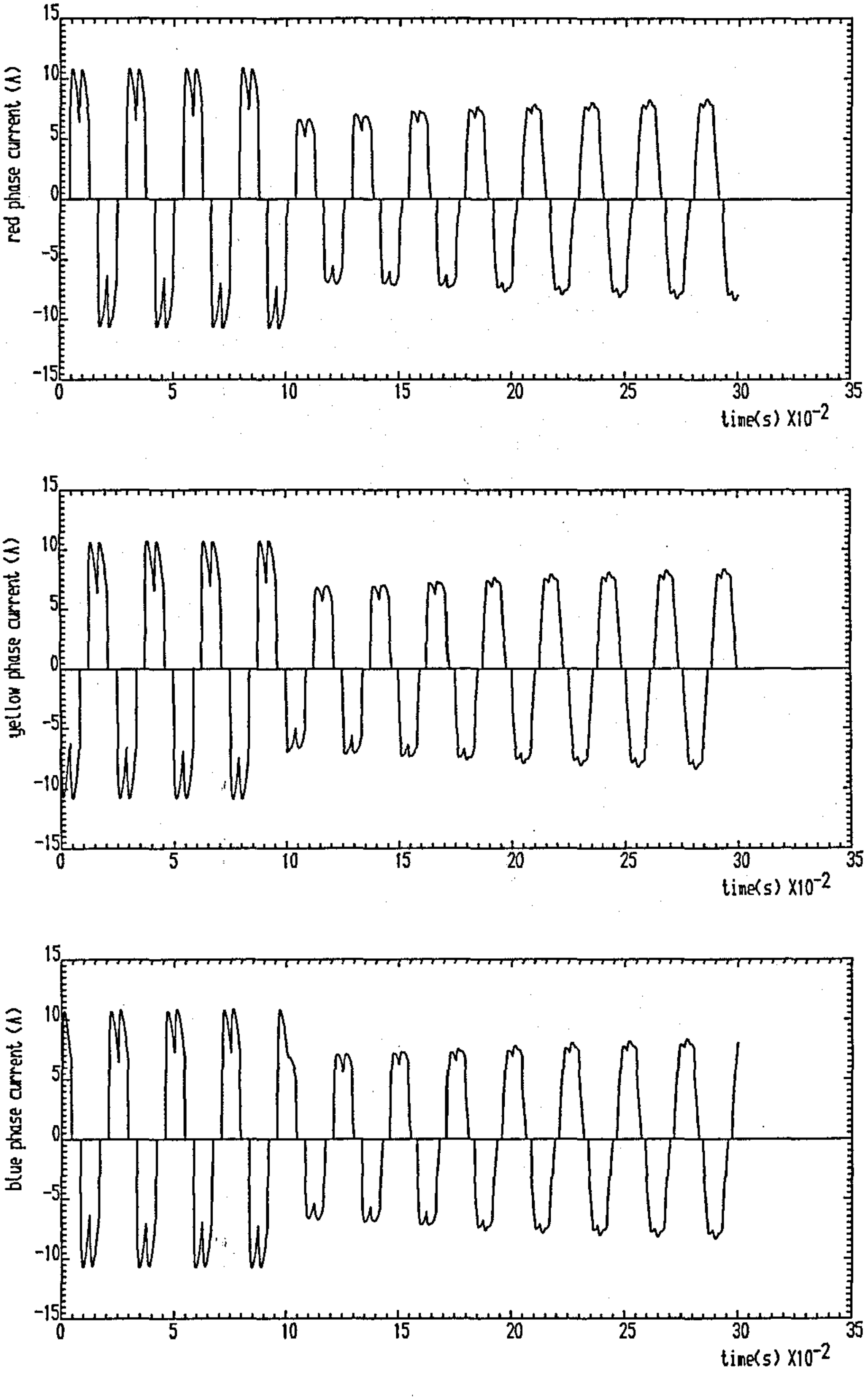



Fig 7.6(d) Converter AC side currents

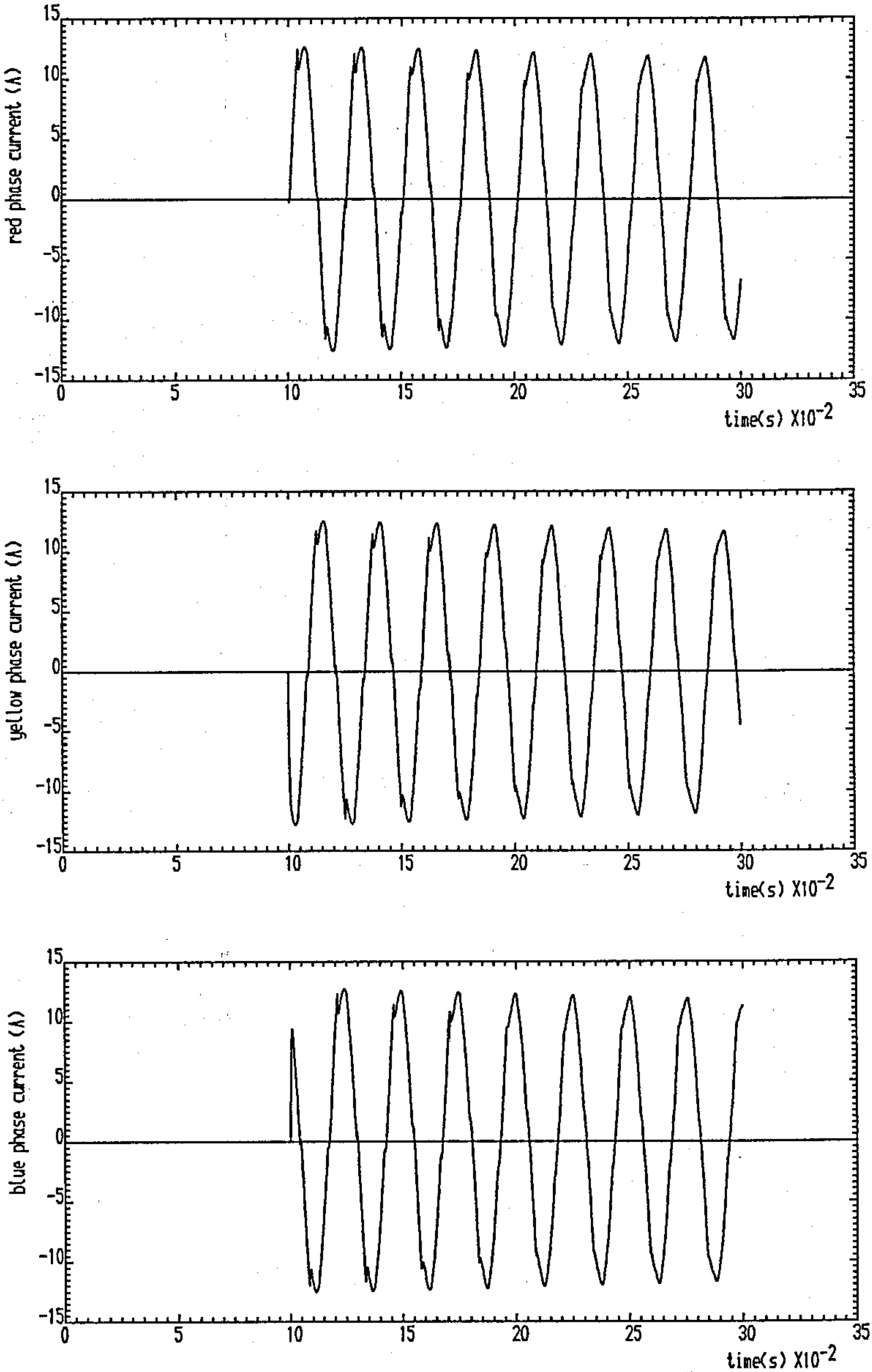



Fig 7.6(e) The MG set line currents

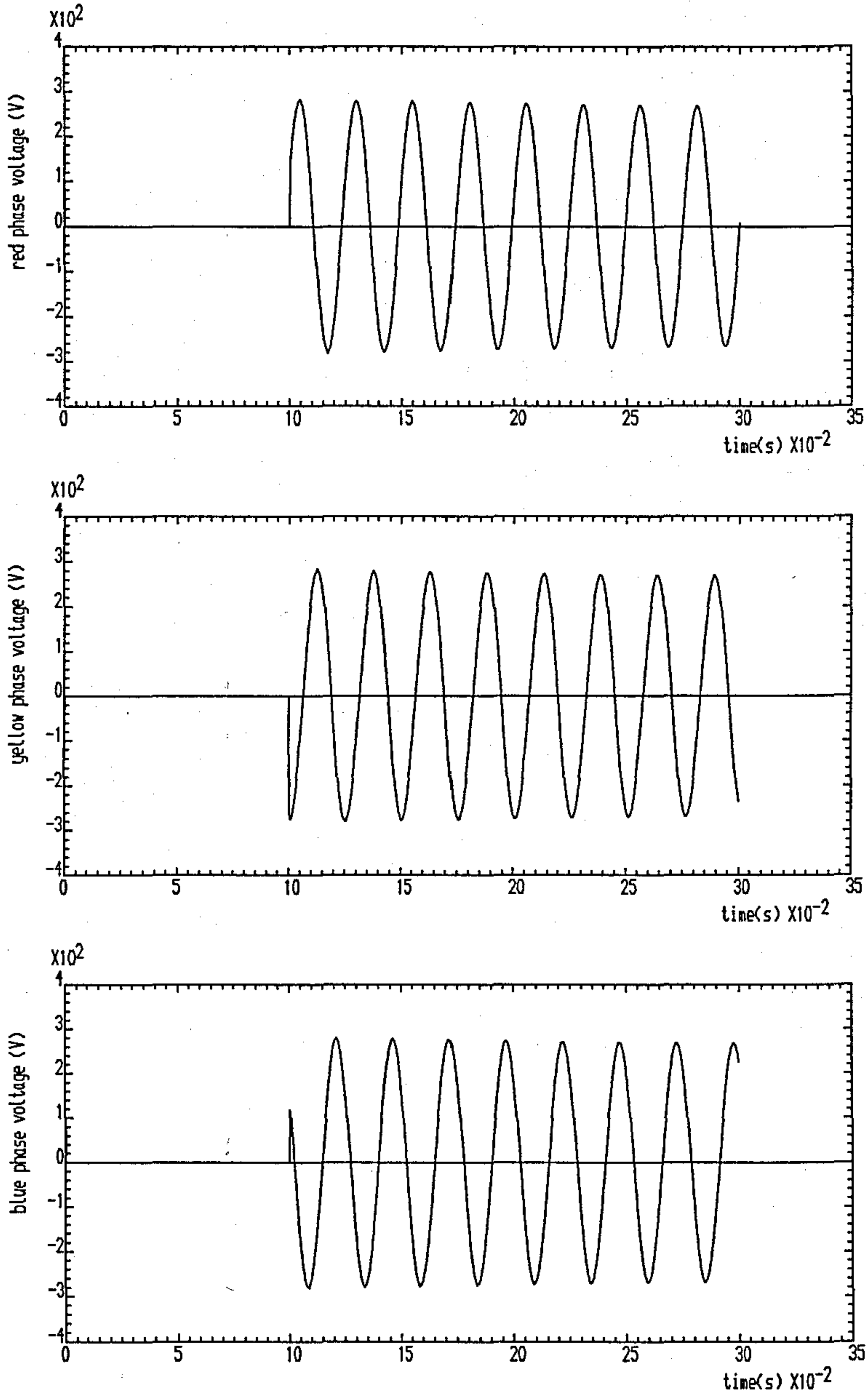



Fig 7.6(f) The MG set line voltages

Chapter 8

# CONCLUSIONS

This thesis has presented a mathematical model for a ship's electrical power system. Initially, an isolated 3-phase synchronous generator was considered, with the electrical equation being expressed in the phase reference frame which was shown to be highly flexible for the modelling of a synchronous generator. It could cope with both balanced and unbalanced load conditions and also allowed for saturation in the model. In addition, the parameters used in the program were physical values, and the approach eliminates the need for complex current and voltage transformation such as in the *dqo* model. Accurate predictions of the behaviour of the synchronous generator were shown to be obtained for a variety of sudden symmetrical and unsymmetrical short circuit tests as well as various balanced and unbalanced loading conditions. The model thus forms the basis of an accurate technique for the modelling of a synchronous generators of any rating.

The single generator study was extended to a multi-generator power system which contained a 3-phase synchronous generator, a motor/generator set, a bus coupler and a 3-phase thyristor bridge converter. A mathematical model for the 3-phase bridge converter which was capable of both rectification and inversion, was developed. In the rectification mode, it was shown that the mean output voltage and current decreased as the trigger angle increased, with the load voltage being continuous for trigger angles between $0^0$ and $60^0$ and discontinuous when the trigger angles exceeded $60^0$. Although the converter can, in general, be used as a variable DC power supply, it suffers from the drawback of producing a high harmonic content in the supply voltage. In the inversion mode, the converter was operated in a PWM manner at a fixed reference frequency and was shown that the higher the frequency of the carrier wave, the greater the number of pulses per half cycle produced in the PWM output voltage waveforms. For a fixed reference wave amplitude, the pulses in the output waveform are reduced in width as the frequency of the carrier wave is increased, which makes the harmonic content smaller [8]. It is preferable, therefore, to work at a high carrier frequency, in order to produce a high-quality output voltage with a reduced harmonic content.

Using the component models developed earlier, the final stage of the thesis explains how diakoptic analysis was employed to model the complete power system, with matrix partitioning being used to reduce the program run-time. The performance of the ship's power system was assessed, by means of simulated results for various switching conditions. Diakoptic analysis was thereby shown to be an effective method for modelling interconnected items and the results obtained using these techniques shows a considerable saving in computing time. In addition, the approach readily allows for a detailed investigation of the system's performance under a variety of switching conditions. The techniques developed in this thesis are sufficiently flexible to be used for the modelling of any complicated parallel-connected network. They enable the designer to investigate economically both the transient and steady state response of a proposed system using various parameters, ultimately enable the performance of the system to be optimised.

# REFERENCES

1. Kron, G: "Diakoptics - the piecewise solution for large-scale systems", MacDonald, 1963.

2. Adkin, B and Harley, R: "The General Theory of Alternating Current Machines", Chapman and Hall, London, 1975.

3. Snider, L.A and Smith, I.R: "Measurement of inductance coefficients of saturated synchronous machines", IEE Proceedings, 1972, Vol. 119, Page 597-602.

4. Kron, G: "Tensor analysis of systems", MacDonald, 1955.

5. Kettleborough, J.G: "Mathematical model for a ship's electrical power systems", Loughborough University of Technology, 1987.

6. "Computer representation of excitation systems", IEEE Committee Report, Pas-87, 1968, page 1460-1464.

7. Lander, C.W: "Power Electronics", MacGraw-Hill (UK) Limited, 1981.

8. Ohno, E.C: "Introduction to Power Electronics", Clarendon Press, Oxford, 1988

9. Gregory, K, Kettleborough, J.G and Smith, I.R: "Diakoptic mesh analysis of limited size power supply systems", IEE Proceedings, Vol. 135, Pt. C, No. 2, March, 1988.

10. Fernando, L. T. M: "Modelling of electrical power system", MPhil Thesis, Loughborough University of Technology, 1984.

11. Kettleborough, J.G., Smith, I.R., Fernando, L.T.M: "Numerical solution of electrical power systems using diakoptics", Fourth International Conference on Mathematical Modelling, Zurich, Switzerland, 15-17 August, 1983.

12. Kettleborough, J.G: "Mathematical model of an aircraft generator/radar load system", RBX Contract Report, 1980.

13. Dubey, G.K., Doradla, S.R., Joshi, A., Sinba, R.M.K: "Thyristorised power controllers", John Wiley and Sons, 1986.

14. Mergen, A.F: "Minimisation of Inverter-fed induction-motor losses by optimisation of PWM voltage waveforms", Ph.D. Thesis, Loughborough University of Loughborough, 1977.

# APPENDICES

# Appendix A

## Dqo/phase Transformation

In deriving a conversion between the two sets of parameters, the following assumptions are made,

a) The 2nd-harmonic components of phase self and phase/phase mutual inductances in the phase co-ordinate reference frame are equal. This assumption is a fundamental requirement of the transformation, and non-compliance with it would yield time varying dq parameters.

b) The d-axis damper/d-axis armature turns ratio $N_5/N_1$ is assumed to be 0.33. This typical value was obtained from static AC tests on several generators. Its actual value is not however critical since, although the damper parameters may be incorrect, their referred values, the mmf contribution by the damper winding and the loss dissipation are all correct.

c) The q-axis damper/q-axis armature turns ratio $N_6/N_1$ is assumed to be 0.33. The arguments of (b) apply again here.

d) $T_d'' = 0.0025s$, and is a typical value [12] based on experimental values obtained from several 400 Hz different machines.

e) $T_q'' = 1.5\ T_d''$, and as above, is an assumption based on experimental values obtained from several 400 Hz machines.

The conversion equations developed in the following sections are all derived from the basic dq parameter relationships. Parameters with a *bar* denote per-unit values and Z is the base impedance given by

$$Z = \frac{\text{rated phase voltage}}{\text{rated phase current}} \tag{A.1}$$

## A.1 The dqo parameters relationships

### A.1.1 Time Constant

All reactances are in per unit and time constant in seconds. [5]

$$T_d{}' = \frac{1}{\omega_0 \overline{R}_f}\left[\overline{X}_f + \frac{\overline{X}_{md}\,\overline{X}_a}{\overline{X}_{md} + \overline{X}_a}\right] \tag{A.2}$$

$$T_d{}'' = \frac{1}{\omega_0 \overline{R}_{kd}}\left[\overline{X}_{kd} + \frac{\overline{X}_{md}\overline{X}_a\overline{X}_f}{\overline{X}_{md}\overline{X}_a + \overline{X}_{md}\overline{X}_f + \overline{X}_a\overline{X}_f}\right] \tag{A.3}$$

$$T_q{}'' = \frac{1}{\omega_0 \overline{R}_{kq}}\left[\overline{X}_{kq} + \frac{\overline{X}_{mq}\overline{X}_a}{\overline{X}_{mq} + \overline{X}_a}\right] \tag{A.4}$$

$$T_{qo}{}'' = \frac{1}{\omega_0 \overline{R}_{kq}}\left[\overline{X}_{kq} + \overline{X}_{mq}\right] \tag{A.5}$$

$$T_{do}{}' = \frac{1}{\omega_0 \overline{R}_f}\left[\overline{X}_f + \overline{X}_{md}\right] \tag{A.6}$$

$$T_{do}'' = \frac{1}{\omega_0 \overline{R}_{kd}} \left[ \overline{X}_{kd} + \frac{\overline{X}_{md}\overline{X}_f}{\overline{X}_{md} + \overline{X}_f} \right] \tag{A.7}$$

## A.1.2 Dqo reactances

$$\overline{X}_d = \overline{X}_{md} + \overline{X}_a \tag{A.8}$$

$$\overline{X}_d' = \overline{X}_d \frac{T_d'}{T_{do}'} = \overline{X}_a + \frac{\overline{X}_{md}\overline{X}_f}{\overline{X}_{md} + \overline{X}_f} \tag{A.9}$$

$$\overline{X}_d'' = \overline{X}_d \frac{T_d' \, T_d''}{T_{do}' \, T_{do}''}$$

$$= \overline{X}_a + \frac{\overline{X}_{md}\overline{X}_f\overline{X}_{kd}}{\overline{X}_{md}\overline{X}_{mf} + \overline{X}_{md}\overline{X}_{kd} + \overline{X}_f\overline{X}_{kd}} \tag{A.10}$$

$$\overline{X}_q = \overline{X}_a + \overline{X}_{mq} \tag{A.11}$$

$$\overline{X}_q'' = \overline{X}_q \frac{T_q''}{T_{qo}''} = \overline{X}_a + \frac{\overline{X}_{mq}\overline{X}_{kq}}{\overline{X}_{mq} + \overline{X}_{kq}} \tag{A.12}$$

$$\overline{X}_2 = \frac{\overline{X}_d'' + \overline{X}_q''}{2} \tag{A.13}$$

### A.1.3 Dqo/phase parameter relationship

$$\bar{L}_d = \frac{\bar{X}_d}{\omega_0} = \bar{L}_{ao} + \bar{M}_{abo} + \frac{3}{2}\bar{L}_{a2} \qquad (A.14)$$

$$\bar{L}_q = \frac{\bar{X}_q}{\omega_0} = \bar{L}_{ao} + \bar{M}_{abo} - \frac{3}{2}\bar{L}_{a2} \qquad (A.15)$$

$$\bar{L}_{md} = \frac{\bar{X}_{md}}{\omega_0} = \bar{M}_F \qquad (A.16)$$

$$\bar{L}_2 = \frac{\bar{X}_2}{\omega_0} = \bar{L}_{ao} - 2\bar{M}_{abo} \qquad (A.17)$$

## A.2 Dqo/phase conversion

### A.2.1 D-axis armature/field turns ratio

The per-unit field self-reactances [5] is

$$\bar{X}_{ff} = \frac{3}{2}\left[\frac{N_d}{N_f}\right]^2 \frac{X_4}{Z} \qquad (A.18)$$

where $\frac{N_d}{N_f}$ is the d-axis armature/field turns ratio

$$\therefore \frac{N_d}{N_f} = \sqrt{\frac{2}{3}\frac{\bar{X}_{ff}}{X_4}Z} \qquad (A.19)$$

and from equation A.8 and A.9,

$$\overline{X}_{ff} = \frac{\overline{X}_{md}^{\,2}}{\overline{X}_d - \overline{X}_d'} \qquad \text{(A.20)}$$

From equation A.6, it follows that

$$X_4 = T_{do}'\,\omega_0\,R_4 \qquad \text{(A.21)}$$

Hence

$$\frac{N_d}{N_f} = \sqrt{\frac{2}{3}\,\frac{Z\,\overline{X}_{md}^{\,2}}{T_{do}'\,\omega_0\,R_4\,(\overline{X}_d - \overline{X}_d')}} \qquad \text{(A.22)}$$

## A.2.2 Phase parameters

From equations A.14 and A.15, the second-harmonic component of armature phase self inductance is

$$L_{a2} = Z\,\overline{L}_2 = \frac{\overline{Z}}{3\,\omega_0}\left[\overline{X}_d - \overline{X}_q\right] \qquad \text{(A.23)}$$

The second harmonic component of phase/phase mutual inductance is

$$M_{ab2} = L_{a2} \quad \text{(according to assumption (a))} \qquad \text{(A.24)}$$

and from equations A.14 and A.17, the constant component of armature phase/phase mutual inductance is

$$M_{abo} = \frac{Z}{3\,\omega_0}\left[\overline{X}_d - \overline{X}_Z\right] - \frac{1}{2}L_{a2} \qquad (A.25)$$

From equation A.17, the constant component of the armature phase self inductance is

$$L_{ao} = \frac{Z}{\omega_0}\overline{X}_Z + 2\,M_{abo} \qquad (A.26)$$

From equation A.6, the field self inductance is

$$L_{fo} = T_{do}'\,R_{ff} \qquad (A.27)$$

$$\overline{X}_{md} = \frac{3}{2}\left[\frac{N_d}{N_f}\right]\frac{\omega_0}{Z}M_f \qquad (A.27)$$

hence,

$$M_f = \frac{2}{3}\left[\frac{N_f}{N_d}\right]\frac{Z}{\omega_0}\overline{X}_{md} \qquad (A.28)$$

and

$$R_{rr} = Z\,\overline{R}_a \qquad (A.29)$$

### A.2.3 D-axis damper winding parameters

From equation A.10, the d-axis damper leakage reactances is

$$\overline{X}_{kd} = \frac{\overline{X}_{md}\overline{X}_f(\overline{X}_d'' - \overline{X}_a)}{\overline{X}_{md}\overline{X}_f - \overline{X}_{ff}(\overline{X}_d - \overline{X}_a)} \tag{A.30}$$

and the d-axis damper leakage reactance is

$$\overline{X}_{kkd} = \overline{X}_{kd} + \overline{X}_{md} \tag{A.31}$$

but

$$\overline{X}_{kkd} = \frac{3}{2}\left[\frac{N_1}{N_5}\right]^2 \frac{\omega_0 L_{do}}{Z} \tag{A.32}$$

therefore

$$L_{do} = \frac{2}{3}\left[\frac{N_5}{N_1}\right]^2 \frac{Z\overline{X}_{kkd}}{\omega_0} \tag{A.33}$$

Assuming that all of the mutual reactances on the d-axis are equal

$$\overline{X}_{md} = \frac{3}{2}\left[\frac{N_1}{N_5}\right] \frac{\omega_0 M_d}{Z} \tag{A.34}$$

therefore

$$M_d = \frac{2}{3}\left[\frac{N_5}{N_1}\right] \frac{Z\overline{X}_{md}}{\omega_0} \tag{A.35}$$

From equation A.10, the d-axis open-circuit sub-transient time constant is

$$T_{do}'' = \frac{T_d' T_d''}{T_{do}'} \frac{\overline{X}_d}{\overline{X}_d''} \qquad (A.36)$$

and from equation A.7, the per-unit d-axis damper resistance is

$$\overline{R}_{kd} = \frac{1}{\omega_0 T_{do}''} \left[ \overline{X}_{kd} + \frac{\overline{X}_{md} \overline{X}_f}{\overline{X}_{md} + \overline{X}_f} \right] \qquad (A.37)$$

Therefore,

$$R_{dd} = \frac{2}{3} \left[ \frac{N_5}{N_1} \right]^2 Z \, \overline{R}_{kd} \qquad (A.38)$$

and

$$M_{fd} = \left[ \frac{N_5}{N_1} \right] M_f \qquad (A.39)$$

## A.2.4 Q-qxis damper winding parameters

From equation A.11, the q-axis damper leakage reactance is

$$\overline{X}_{kkq} = \overline{X}_{kq} + \overline{X}_{mq} \qquad (A.40)$$

$$\overline{X}_{kkq} = \frac{3}{2} \left[ \frac{N_1}{N_6} \right]^2 \frac{\omega_0 L_{qo}}{Z} \qquad (A.41)$$

and,

$$L_{\varphi} = \frac{2}{3}\left[\frac{N_6}{N_1}\right]\frac{Z\overline{X}_{kkq}}{\omega_0} \tag{A.42}$$

$$\overline{X}_{mq} = \frac{3}{2}\left[\frac{N_1}{N_6}\right]\frac{\omega_0 M_q}{Z} \tag{A.43}$$

therefore,

$$M_q = \frac{2}{3}\left[\frac{N_6}{N_1}\right]\frac{Z\overline{X}_{mq}}{\omega_0} \tag{A.44}$$

From assumption (e), $T_q'' = 1.5\, T_d''$

Also from equation A.12, the q-axis open-circuit sub-transient time constant is

$$T_{\varphi}'' = \frac{\overline{X}_q}{\overline{X}_q''}\, T_q'' \tag{A.45}$$

$$T_{\varphi}'' = \frac{L_{qq}}{R_{qq}} \tag{A.46}$$

therefore,

$$R_{qq} = \frac{L_{qq}}{T_{\varphi}''} \tag{A.47}$$

The d-axis and q-axis component of phase self and mutual inductances are

$$L_{ad} = L_{ao} + L_{a2} \tag{A.48}$$

$$L_{aq} = L_{ao} - L_{a2} \tag{A.49}$$

$$M_{ad} = 2\,M_{abo} + M_{ab2} \tag{A.50}$$

$$M_{aq} = 2\,M_{abo} - M_{ab2} \tag{A.51}$$

## Appendix B

### 4th ORDER RUNGE KUTTA EQUATIONS

A first-order differential equation may usually be arranged in the form

$$\frac{dx}{dt} = ax + bu \qquad \text{(B.1)}$$

where x is the state variable and u the system input.

A step-by-step solution for this equation may be obtained using the 4th order Runge Kutta integration procedure, defined as

$$x_n = x_{n-1} + \frac{h}{6}\left[G_0 + 2\,G_1 + 2\,G_3 + G_3\right] \qquad \text{(B.2)}$$

where

$x_n$ is the value of the state variable at the end of the integration step,

$x_{n-1}$ is the value of the state variable at the beginning of the integration step,

h is the duration of the integration step,

$$G_0 = \frac{dx}{dt}(t_{n-1}, x_{n-1}) \qquad \text{(B.3)}$$

$$G_1 = \frac{dx}{dt}\left(t_{n-1} + \frac{h}{2},\ x_{n-1} + G_0\frac{h}{2}\right) \qquad \text{(B.4)}$$

$$G_2 = \frac{dx}{dt}\left(t_{n-1} + \frac{h}{2},\ x_{n-1} + G_1\frac{h}{2}\right) \qquad \text{(B.5)}$$

$$G_3 = \frac{dx}{dt}\left(t_{n-1} + h,\ x_{n-1} + G_2 h\right) \qquad \text{(B.6)}$$

where $t_{n-1}$ is the time at the start of the integration step.

# Appendix C

## PROGRAM LISTING

## C.1 Typical input data for the ship's power system model

```
400.0, 40.00, 10.00e-06, 10.00e-05, 0.075, 1
98.00, 7.35,
163.3, 14.70,
212.00, 29.4
1.7753, 1.6664, 0.9251, 0.8162, 0.2506, 0.1998, 0.1735, 0.0259
0.0186, 0.6119, 10.00, 40.00, 0.6930
0.1699, 0.0239, 0.0025
1.000, 0.000, 0.000, 0.000, 0.000,
0.000, 1.000, 0.000, 0.000, 0.000,
-1.000, -1.000, 0.000, 0.000, 0.000,
0.000, 0.000, 1.000, 0.000, 0.000,
0.000, 0.000, 0.000, 1.000, 0.000,
0.000, 0.000, 0.000, 0.000, 1.000
200.00, 5.0, -5.0
1.20, 5.000e-04, 2.00, 4.04e-02
9.00e-03, 7.517e-03, 19.38e-03, 12.00e-03
0.65, 5.00, 0.05, 0.05, 0.001
0.03, 250
240.00, 35.00
0.148, 0.300, 0.300, 90.9094
67.4e-03, 30.00e-03, 30.00e-03, 24.00e-03
0.000, 2.419, 0.000, 38.6-03, 38.6e-03, 38.6e-03, 38.6e-03
0.100,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.100,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.100,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.100,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.100,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.100,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.100,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.100,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.100,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.100,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.100,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.100,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.100,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.100,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.100,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.100
10.0e-03,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,10.0e-03,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,10.0e-03,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,10.0e-03,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,10.0e-06,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,10.0e-06,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,10.0e-06,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,10.0e-06,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,10.0e-06,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,10.0e-06,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,10.0e-07,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,10.0e-07,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,10.0e-07,0.0,0.0,0.0,
```

```
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,10.0e-07,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,10.0e-07,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,10.0e-07
1.0,1.0,1.0,1.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,0.0,
1.0,1.0,-1.0,0.0,-1.0,0.0,-1.0,-1.0,-1.0,0.0,1.0,0.0,1.0,1.0,1.0,0.0,0.0,-1.0,-1.0,-1.0,1.0,0.0,1.0,0.0,0.0,
-1.0,0.0,1.0,1.0,0.0,-1.0,1.0,0.0,0.0,-1.0,-1.0,-1.0,0.0,-1.0,0.0,1.0,1.0,1.0,1.0,0.0,-1.0,-1.0,0.0,1.0,0.0,
0.0,-1.0,0.0,-1.0,1.0,1.0,0.0,1.0,1.0,1.0,0.0,1.0,-1.0,0.0,-1.0,-1.0,-1.0,0.0,0.0,1.0,0.0,1.0,-1.0,-1.0,0.0,
1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,1.0,1.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,1.0,1.0,0.0,1.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,1.0,0.0,1.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,
1.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,1.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,1.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0,1.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,1.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0,1.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,1.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,1.0,1.0,0.0,0.0,0.0,0.0,1.0,0.0
800.00, 10.00, 8.00, 20.00e-06
1, 6, 5, 0, 3, 2
1.0,0.0,
0.0,1.0,
-1.0,-1.0
0.100,0.0,0.0,
0.0,0.100,0.0,
0.0,0.0,0.100
1.0,0.0,0.0,0.0,0.0,
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,
1.0,0.0,0.0,0.0,0.0,
0.0,1.0,0.0,0.0,0.0,
-1.0,0.0,0.0,0.0,0.0,
0.0,-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,
-1.0,0.0,0.0,0.0,0.0,
0.0,-1.0,0.0,0.0,0.0,
1.0,0.0,0.0,0.0,0.0,
0.0,1.0,0.0,0.0,0.0,
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,
1.0,0.0,1.0,-1.0,0.0,-1.0,0.0,
-1.0,1.0,0.0,1.0,-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,
1.0,0.0,1.0,-1.0,0.0,-1.0,0.0,
-1.0,1.0,0.0,1.0,-1.0,0.0,0.0
1.0,1.0,1.0,1.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
1.0,1.0,-1.0,0.0,-1.0,0.0,1.0,0.0,-1.0,-1.0,0.0,1.0,0.0,
-1.0,0.0,1.0,1.0,0.0,-1.0,0.0,1.0,1.0,0.0,-1.0,-1.0,0.0,
0.0,-1.0,0.0,-1.0,1.0,1.0,-1.0,-1.0,0.0,1.0,1.0,0.0,0.0,
1.0,1.0,0.0,0.0,0.0,0.0,1.0,0.0,-1.0,0.0,0.0,0.0,0.0,
```

```
0.0,0.0,1.0,1.0,0.0,0.0,0.0,0.0,1.0,0.0,-1.0,0.0,0.0,
0.0,0.0,0.0,0.0,1.0,1.0,-1.0,0.0,0.0,0.0,1.0,0.0,0.0,
0.0,0.0,1.0,0.0,1.0,0.0,0.0,0.0,0.0,1.0,0.0,-1.0,0.0,
1.0,0.0,0.0,0.0,0.0,1.0,0.0,-1.0,0.0,0.0,0.0,1.0,0.0,
0.0,1.0,0.0,1.0,0.0,0.0,0.0,1.0,0.0,-1.0,0.0,0.0,0.0
50.00, 1.00e-03, 10.0e-06, 10.0e-06, 10.0e-07
10.00, 10.00, 10.00
0,0,0,0,0
0.0,0.0,0.100,0.0,0.0,0.0
1,0,0,1,1
2.00
1,2
```

## C.2 Program listing of the ship's electrical power system

```
c     This program models the complete system. Diakoptics is used to
c     reduces the program complexity.
c     Open the input and output files required.
      Open(unit = 50, file = "system_data", status = "old",
     +      form = 'formatted')
      Open(unit = 60, file = '>site>plot_dir>see>gen1',
     +      form = 'unformatted', binary stream = .true., mode = 'out')
      Open(unit = 70, file = '>site>plot_dir>see>gen2',
     +      form = 'unformatted', binary stream = .true., mode = 'out')
      Open(unit = 80, file = '>site>plot_dir>see>gen3',
     +      form = 'unformatted', binary stream = .true., mode = 'out')
      Open(unit = 90, file = '>site>plot_dir>see>bus',
     +      form = 'unformatted', binary stream = .true., mode = 'out')
      Open(unit = 92, file = '>site>plot_dir>see>diode',
     +      form = 'unformatted', binary stream = .true., mode = 'out')
      Open(unit = 95, file = '>site>plot_dir>see>PWM',
     +      form = 'unformatted', binary stream = .true., mode = 'out')
      Open(unit = 96, file = '>site>plot_dir>see>exc',
     +      form = 'unformatted', binary stream = .true., mode = 'out')
      Open(unit = 97, file = '>site>plot_dir>see>dies',
     +      form = 'unformatted', binary stream = .true., mode = 'out')
      Open(unit = 98, file = '>site>plot_dir>see>motor',
     +      form = 'unformatted', binary stream = .true., mode = 'out')
      Open(unit = 99, file = '>site>plot_dir>see>line',
     +      form = 'unformatted', binary stream = .true., mode = 'out')
c     Set up the common block.
      Common/b1/Vb(5,16),e(16),h(4),gg(4),t
      Common/b2/Cd(16),Rb(16),Xb(16),sl,w,V1,ire, slr, sli
      Common/b3/Cbt(2,16), Vdrop(5), Rba(3,3)
      Common/b4/Cd1(5), Cd2(5), Rbb(16,16), Xbb(16,16)
      Common/b5/rr(2,16), xx(2,16), idone
      Common/b6/t1, amp, am, ti, ti1, ti2, ti3
      Common/b7/twen, vsa, vsb, vsc, samp, ff
      Common/b8/za1, za2, za3, zb1, zb2, zb3, aa, bb
      Common/b9/icio(3), icirc(3), int
      Common/b10/nire, inter, ipass(2), iooo, ibbb
      Common/b11/tr1, tr2, nn1, nn2
      Common/b12/wo,pi,fo,xd,xmd,xq,xd1,xd2,xq2,xo,xmq
      Common/b13/ra,r4,p,z,tdo,td1,td2,freq
      Common/b14/xlad,xlaq,xmad,xmaq,xmf,xxmd,xxmq,xlf,xmfd
      Common/b15/xld,xlq,r1,r2,r3,r5,r6,rl1,rl2,rl3
      Common/b16/Con1(5,4,5),Clm(5,4,5),Cml(5,5,4),xLl(5,3),xopen,xclose
      Common/b17/cos1(3),cos2(3),cos3(3),sin1(3),sin2(3),sin3(3)
      Common/b18/Cm1(3,4,6),Cm2(3,4,6)
      Common/b19/cm(5,5), cma(5,5), cmb(5,5),Cmast(6,5),dd(5,5)
      Common/b20/cf1,cf2,sf(3),Oc(3,2),coo,const,ca1,ca2
      Common/b21/Vh, Vv, Vp, three, ppi, V5, Vmin, Vmax
      Common/b22/ava(4), avb(4), avc(4), avd(4)
      Common/b23/aref(2,2), d1(4), Vft(3), iexc
      Common/b24/time(3), cycle, dummy(3,3,2), Tot
      Common/b26/S(3,3), Si(2,2), d2(3), Xji, S1(3), S2(3), zx
      Common/b27/ca,cf,Va,Vf,rtt,rt,zxy,xt,xf,ck,rf,ckf,Ji
      Common/b28/dumm,ion,Vfr(6),max,trig,itrg,ioni
      Common/b29/idis,cbi(16),tsis,icom
      Common/b30/dd1,dd2,dd3,Vdc,sign,db1,db2,zfy,zyh
      Common/b31/aug(2,2),Ar(4,4),Ar1(2,4,4)
      Common/b32/isw,iti(5),cb(5,16),Xba(3,3),Rload,Xload,srec,spwm
      Common/b33/tyr,diod,tdc
c     Define the real and integer variables.
```

```
      Real Lmm(5,5,5), A(1,4,4), ta1(3), pi, twen, Vm(5,5),
     +  tad(3), ww(5), ta2(3), ta3(3),G(3,5,5),ttt(4), Cpwm(4,7),
     +  Amm(16,5,5), Lii(5,5,5), Ai(1,4,4), Rmm(5,5,5),
     +  av1(4,4), av2(4,2), av(4,4), xmat1(3,3), xmat2(3,2),
     +  Cbar(3,2),tc(6),thr,diod,shift,sfi(3),Cbm(16,2),
     +  Con(5,4,5),An(5,5),Amt(16,5,5),Te(5),Vrt(4),
     +  ka,kf,ke,kr,Cmod(16,25),rl1,rl2,rl3,
     +  Rit(5,5,5),Crec(10,13),Crpm(4,7),Vli(5,3),opf(4)
      Integer mesh1, mesh2, num(3), ino, itwo, iooo, ibbb, zero,k3,
     +  ire, ioy, change(3), Total, mm(4), open, close, switch,
     +  ch(5),SW(6),State(5),run,ipold(2),power,in,ini,ic,icalc,
     +  icin(6), imot, ichh, ifree(3), ifn(3), iq
      Data (Rmm(i,1,3),Rmm(i,1,4),Rmm(i,1,5),Rmm(i,2,3),Rmm(i,2,4),
     +  Rmm(i,2,5),Rmm(i,3,1),Rmm(i,3,2),Rmm(i,3,4),Rmm(i,3,5),
     +  Rmm(i,4,1),Rmm(i,4,2),Rmm(i,4,3),Rmm(i,4,5),Rmm(i,5,1),
     +  Rmm(i,5,2),Rmm(i,5,3),Rmm(i,5,4), i = 1,3)/54*0.000/
      Data (Vm(i,1), Vm(i,2), Vm(i,4), Vm(i,5), i = 1,3)/12*0.000/
      Data (G(i,3,3),G(i,4,4),G(i,5,5),G(i,3,4),G(i,3,5),G(i,4,3),
     +  G(i,5,3),G(i,4,5),G(i,5,4),Lmm(i,3,5),Lmm(i,5,3),Lmm(i,4,5),
     +  Lmm(i,5,4),i = 1,3)/39*0.000/
      Data av1(1,4), av1(2,3), av1(2,4), av1(3,2), av1(3,4), av1(4,2),
     +  xmat1(1,2), xmat2(3,1), ((av2(i,j), j = 1,2), i = 2,4)
     +  /14*0.000/
      Read(50,*) V1, freq, spwm, srec, Tstop, jcc
      Read(50,*)((Oc(I,J), J = 1,2), I = 1,3)
      Read(50,*)xd, xmd, xq, xmq, xd1, xd2, xq2, xo
      Read(50,*)ra, r4, p, fo,z
      Read(50,*)tdo, td1, td2
      Read(50,*)((Cmast(I,J), J = 1,5), I=1,6)
      Read(50,*)Vref, Vmax, Vmin
      Read(50,*)ka, kf, ke, kr
      Read(50,*)Ta, Tf1, Tf2, Te1
      Read(50,*) ck0, ck1, ck2, ck3, ckf
      Read(50,*) Xji, sref
      Read(50,*)  Vf, Ji
      Read(50,*) rma, rh, ry, rf
      Read(50,*) xa, xh, xy, xf
      Read(50,*) ckf, ck, cky, zha, zah, zfy, zyh
      Read(50,*) ((Rbb(I,J), J = 1,16), I = 1,16)
      Read(50,*) ((Xbb(I,J), J = 1,16), I = 1,16)
      Do 551 I = 1,16
  551 Read(50,*) (Cmod(I,J), J = 1,25)
      Do 552 i = 1,25
  552 print*, "mesh ", i, " = ", (Cmod(j,i), j=1,16)
      Read(50,*) carry, amp, samp, safe
      Read(50,*) (icin(I), I = 1,6)
      Read(50,*) ((Cbar(i,j), j = 1,2), i= 1,3)
      Read(50,*) ((Rba(i,j), j = 1,3), i = 1,3)
      Read(50,*) (((Clm(i,j,k), k = 1,5), j = 1,4), i = 1,5)
      Read(50,*) ((Cpwm(i,j), j = 1,7), i = 1,4)
      Read(50,*) ((Crpm(i,j), j = 1,7), i = 1,4)
      Read(50,*) ((Crec(i,j), j = 1,13), i= 1,10)
      Read(50,*) xopen, xclose, tdc, tyr, diod
      Read(50,*) rl1,rl2,rl3
      Read(50,*) (SW(i), i=1,5)
      Read(50,*) (tc(i), i=1,5)
      Read(50,*) (State(i), i=1,5)
      Read(50,*) shift
      Read(50,*) in,ini
  133 Print*, "Run as power forward(1), or power reverse(2)?"
      Read*, power
      If(power .EQ. 1) then
```

```
        print*, "Input the load resistance? "
        Read*, Rbb(1,1)
        print*, "Input the load Inductance? "
        Read*, Xbb(1,1)
        print*, "Input the trigger interval in degrees"
        Read*, deg
        print*, "It is now carry out rectification!"
      elseif(power .EQ. 2) then
        print*, "Input the modulating frequency in Hz"
        Read*, carry
        print*, "It is now carry out convertion!"
      else
        go to 133
      endif
c     Setting up the load resistance and inductance.
      Rload = Rbb(1,1)
      Xload = Xbb(1,1)
c     Setting the binary file.
      Write(60)0,11,3
      Write(70)0,11,3
      Write(80)0,11,3
      Write(90)0,7,3
      Write(92)0,13,3
      Write(95)0,9,3
      Write(96)0,9,3
      Write(97)0,7,3
      Write(98)0,9,3
      Write(99)0,7,3
c     Defining the values pi and calculates the value of w
      pi = 4.0 * atan(1.0)
      ww(1) = 2 * pi * freq
      Do 1 i = 2,5
 1    ww(i) = ww(1)
      twen = 120.00*pi/180.00
c     Calculate the trigger angle for the rectification.
      trig = deg*pi/ww(1)/180.00
c     Reset the trigger angle to zero.
      tsi = 0.000
      tsis = 1.00/freq/6.00
      itrg = 1
      icond = 1
      idone = 1
      ion = 1
      ioni = 0
      jint = 0
      icom = 0
      max = 1
      sl = sli
      Do 3 i= 1,2
      ipold(i) = 0
 3    ipass(i) = 0
      Do 2 i = 1,3
 2    sf(i) = 0.000
c     Setting the inductance values for the swithch. All the switches
c     are assumed to be closed except the bus-bar.
      Do 56 j = 1,5
      If(State(j) .EQ. 1) then
        Do 55 i = 1,3
 55     xLl(j,i) = xclose
        go to 56
      endif
      If(State(j) .EQ. 0) then
```

```
               Do 57 i = 1,3
 57         xLl(j,i) = xopen
             endif
 56          Continue
             If(State(5) .EQ. 0) then
               Do 58 i = 1,16
 58       Xbb(i,i) = xopen
               Do 465 i = 1,4
               Do 465 j = 1,4
               Lii(5,i,j) = 0.000
               Lmm(5,i,j) = 0.000
 465       Rmm(5,i,j) = 0.000
             elseif(State(5) .EQ. 1) then
               Do 59 i = 1,16
              If(power .EQ. 2) then
               If(i .EQ. 1) Xbb(i,i) = tdc
              elseif(power .EQ. 1) then
               If(i .LE. 4 .AND. i .NE. 1) Rbb(i,i) = rl1
              endif
               If(i .LE. 4 .AND. i .NE. 1) Xbb(i,i) = xclose
               If(i .GT. 4 .AND. i .LE. 10) Xbb(i,i) = tyr
               If(i .GT. 10) Xbb(i,i) = diod
 59          Continue
             endif
             If(power .EQ. 2 .AND. State(4) .EQ. 0)then
               iforw = 1
               iback = 2
               ww(1) = 0.000
               ww(2) = 0.000
             elseif(power .EQ. 2 .AND. State(4) .EQ. 1) then
               iback = 1
               iforw = 2
               ww(1) = 0.000
               ww(2) = 0.000
             elseif(power .EQ. 1 .AND. State(4) .EQ. 0) then
                 iforw = 1
                 iback = 2
             elseif(power .EQ. 1 .AND. State(4) .EQ. 1) then
                If(State(1) .EQ. 1) then
                  iback = 1
                  iforw = 2
                elseif(State(2) .EQ. 1) then
                  iback = 2
                  iforw = 1
                else
                  print*, "Error in switching pattern"
                  go to 3000
                endif
             else
               print*, "Error in switching pattern"
               go to 3000
             endif
c             Determine if the motor is on during power forward generation.
             If(State(3) .EQ. 0) then
               ww(3) = 0.000
               imot = 1
             else
               imot = 0
             endif
             If(power .EQ. 2) then
               Do 61 ix = 1,2
               cos1(ix) = cos(ta1(1))
```

```
          cos2(ix) = cos(ta2(1))
          cos3(ix) = cos(ta3(1))
          sin1(ix) = sin(ta1(1))
          sin2(ix) = sin(ta2(1))
          sin3(ix) = sin(ta3(1))
 61    Continue
        endif
c       Setting up the initial step length fot eh system.
        If(power .EQ. 2 .AND. State(5) .EQ. 1) then

          sli = spwm
          Do 88 i = 1,3
 88    icirc(i) = icin(i)
        elseif(power .EQ. 1 .OR. State(5) .EQ. 0) then
          sli = srec
          Do 89 i = 4,6
 89    icirc(i-3) = icin(i)
        else
          print*, "Error in step lengh"
          go to 3000
        endif
c       Updating the step length.
        sl = sli
c       Setting up all the initial conditions for the current
c        and voltage.
        Do 5 ii = 1,5
        Do 5 i = 1, 5
 5      cm(ii,i) = 0.000
        Do 9 ii = 1,4
        Do 8 i = 1,6
 8      cb(ii,i) = 0.000
        Do 9 i = 1,4
 9      Vb(ii,i) = 0.000
        Do 10 i = 1,16
        Vb(5,i) = 0.000
 10     cb(5,i) = 0.000
        Do 233 i = 1,5
        Do 233 j = 1,3
 233    Vli(i,j) = 0.000
        Do 234 i = 1,3
 234    Te(i) = 0.000
c       Setting the time to zero to start the program.
        t = 0.000
c       Setting the initial phase difference for the synchronous
c       machine.
        ta1(1) = 0.000
        ta2(1) = ta1(1) - twen
        ta3(1) = ta1(1) + twen
        Do 11 i = 2,3
        ta1(i) = ta1(1)
        ta2(i) = ta2(1)
 11     ta3(i) = ta3(1)
c       Set up the constant values for Runge-Kutta Method.
        h(1) = 0.5000
        h(2) = 0.5000
        h(3) = 1.0000
        h(4) = 0.0000
        gg(1) = 1.00/6.00
        gg(2) = 1.00/3.00
        gg(3) = 1.00/3.00
        gg(4) = 1.00/6.00
        ttt(1) = 0.0000
```

```
      ttt(2) = 0.5000
      ttt(3) = 0.5000
      ttt(4) = 1.0000
c     Updating the number of inversion in the generators.
      ic = 1
      icalc = ini+1
      ich = 0
      Do 12 i = 1,3
 12   tad(i) = 0.000
      cycle = 1.00/freq
c     Calculate the required matrix for the exciter.
      kf1 = kf/(Tf2 - Tf1)
      av1(1,1) = -1.00/Ta
      av1(1,2) = -ka/Ta
      av1(1,3) = -av1(1,2)
      av1(2,1) = kf1/Tf1
      av1(2,2) = -1.00/Tf1
      av1(3,1) = kf1/Tf2
      av1(3,3) = -1.00/Tf2
      av1(4,1) = ke/Te1
      av1(4,4) = -1.00/Te1
      av2(1,1) = ka/Ta
      av2(1,2) = -ka*kr/Ta
      Do 16 ii = 1,2
      Do 15 i = 1, 4
 15   av(ii,i) = 0.000
      Do 77 i = 1,3
      Do 77 j = 1,2
      dummy(ii,i,j) = 0.000
 77   Continue
 16   Continue
      Do 18 i = 1,2
      aref(i,1) = Vref*kr
 18   aref(i,2) = aref(i,1)/kr
      iexc = 0
c      Set up the initial condition for the diesel engine.
      S(1,3) = ww(1)
      S(2,3) = ww(2)
      Do 31 ii = 1,2
      Do 31 i = 1, 2
      S(ii,i) = 0.000
 31   Continue
c     Set up the required matrix.
      Do 166 i = 1,2
 166  Si(i,1) = sref
      xmat1(1,1) = -1.00/ck2
      xmat1(1,3) = -ck0/ck2
      xmat1(2,1) = (1.00 - ck1/ck2)/ck3
      xmat1(2,2) = -1.00/ck3
      xmat1(2,3) = -ck1*ck0/ck2/ck3
      xmat1(3,2) = 1.00/Xji
      xmat1(3,3) = -ckf/Xji
      xmat2(1,1) = ck0/ck2
      xmat2(2,1) = ck1*ck0/ck2/ck3
      xmat2(3,2) = -1.00/Xji
c      Calculate the time for the conduction pattern.
      tim = 1/carry
      ff = 2*pi*freq
      iii = 0
c     Set up some initial values for the PWM modulating and
c     carrier waves.
      vt = 0.000
```

```
      tr = 0.000
      slr = 0.000
      zb1 = 0.000
      zb2 = samp*sin(twen/shift)
      zb3 = samp*sin(-twen/shift)
      vsa = 0.000
      vsb = zb2
      vsc = zb3
c     Calculate the parameter for the dc machine.
      dd1 = 0.000
      dd2 = 0.000
      dd3 = 0.000
      db1 = 0.000
      db2 = 0.000
      ca = 0.000
      cf = Vf/rf
c      Determine if it is motoring or generating in the dc machine.
      If(power .EQ. 1) then
        sign = 1.000
        Vdc = 0.000
        rtt = rma + rh + ry + Rbb(1,1)
        xt = xa + xh + xy + zha + zah + Xbb(1,1)
      elseif(power .EQ. 2) then
        sign = -1.000
        Vdc = V1
        rtt = rma + rh + ry
        xt = xa + xh + xy + zha + zah
      endif
c      Calculate some required constant.
      ti = 0.500*tim
      am = 2.0*amp/ti
      ti1 = 0.500*ti
      ti2 = ti + ti1
      ti3 = tim
      yy = 0
c     Set up all the initial values of the program.
      t1 = 0.000
      t2 = 0.000
      dumm = 0
      jjj = 0
      mesh1 = 0
      mesh2 = 0
      Do 99 i = 1,3
      ifree(i) = 0
      ifn(i) = 0
 99   num(i) = 0
      ino = 0
      itwo = 0
      iooo = 0
      ibbb = 0
      zero = 0
      int = 1
      ire = 0
      nire = 0
      ioy = 0
      iq = 0
      ichh = 0
      open = 0
      close = 0
      jclose = 0
      switch = 0
```

```
      k3 = 0
      Do 43 i = 1,5
 43   iti(i) = 0.000
      Do 13 i = 1,3
      time(i) = 0.000
 13   change(i) = 0
c     Test if it is a inverter or a rectifier and
c     set up the required impressed branch voltage vectors.
      If(power .EQ. 2) then
        e(1) = V1
      elseif(power .EQ. 1) then
        e(1) = 0.000
      endif
      Do 40 i = 2,16
 40   e(i) = 0.000
      Do 41 i = 1,2
 41   Vm(5,i) = 0.000
      Do 42 i = 1,4
      Do 42 j = 1,4
      Ar1(1,i,j) = 0.000
      Ar1(2,i,j) = 0.000
 42   Ar(i,j) = 0.000
c      Produce Cml from Clm.
      Do 19 ii = 1,5
      Do 19 i = 1,5
      Do 19 j = 1,4
 19   Cml(ii,i,j) = Clm(ii,j,i)
c      Set up the constant for the matrix.
      ch(1) = 5
      ch(2) = 5
      ch(3) = 5
      ch(4) = 2
      ch(5) = 2
c     Setting up the initial values for the bus-bar.
      Xba(1,1) = xLl(4,1)
      Xba(2,2) = xLl(4,2)
      Xba(3,3) = xLl(4,3)
      Do 144 i = 1,3
      Do 144 j= 1,3
      If(i .EQ. j) Xba(i,j) = 0.000
 144  Continue
      Lmm(4,1,1) = xLl(4,1)+xLl(4,3)
      Lmm(4,1,2) = xLl(4,3)
      Lmm(4,2,1) = xLl(4,3)
      Lmm(4,2,2) = xLl(4,2)+xLl(4,3)
      Rmm(4,1,1) = Rba(1,1)+Rba(3,3)
      Rmm(4,1,2) = Rba(3,3)
      Rmm(4,2,1) = Rba(3,3)
      Rmm(4,2,2) = Rba(2,2)+Rba(3,3)
      Call twin(Lmm,Lii,4,aug,icond)
      Call recalc(Lii,4,ich)
c     Call the subroutine to calculate the required for the
c     synchnous generator.
      Call para(Vm,cm,vb,Rmm)
c     Calculate the open circuit voltage for the three generators.
c     Since saturation factor = 1.
      If(power .EQ. 1) then
        opf(1) = -ww(1)*xmf*sin(ta1(1))
        opf(2) = -ww(1)*xmf*sin(ta2(1))
        opf(3) = -ww(1)*xmf*sin(ta3(1))
        opf(4) = r4
        Do 455 i = 1,4
```

```
            Vb(1,i) = 0.000
  455   Vb(1,i) = opf(i)*cm(1,3)
            Do 466 i = 2,3
            Do 466 j = 1,4
  466   Vb(i,j) = Vb(1,j)
            Do 477 i = 1,3
  477   e(i+1) =  Vb(2,i)
         endif
         Do 4 i = 1,3
c        Define some values for the matrix.
         Cm1(i,4,6) = 0.000
         Cm1(i,4,5) = 0.000
         Cm2(i,4,6) = 0.000
  4      Lmm(i,5,5) = xlq
c        Calculate the initial line voltage.
         Do 787 i = 1,5
         If(i .EQ. 5) then
            Vli(i,1) = Vb(i,2) - Vb(i,3)
            Vli(i,2) = Vb(i,3) - Vb(i,4)
            Vli(i,3) = Vb(i,4) - Vb(i,2)
         else
            Vli(i,1) = Vb(i,1) - Vb(i,2)
            Vli(i,2) = Vb(i,2) - Vb(i,3)
            Vli(i,3) = Vb(i,3) - Vb(i,1)
         endif
  787    Continue
c        Output the initial values.
         Write(60) t, (cb(1,i), i=1,6), (Vli(1,i), i =1,3), Vb(1,4)
         Write(70) t, (cb(2,i), i=1,6), (Vli(2,i), i =1,3), Vb(2,4)
         Write(80) t, (cb(3,i), i=1,6), (Vli(3,i), i =1,3), Vb(3,4)
         Write(90) t, (cb(4,i), i=1,3), (Vli(4,i), i =1,3)
         Write(92) t, (cb(5,i), i=5,16)
         Write(95) t, (cb(5,i), i=1,4), (Vb(5,i), i =1,4)
         Write(96) t, ((av(i,j), j=1,4), i=1,2)
         Write(97) t, ((S(i,j), j=1,3), i=1,2)
         Write(98) t, (ww(i), i=1,3), ca, cf, (Te(i), i=1,3)
         Write(99) t, (Vli(iback,i), i=1,3), (Vli(5,i), i=1,3)
         jjjj = 1
         jc = 0
c        Determine if the convertor is turn off.
c        Determine if it is doing rectification.
  5005   If(power .EQ. 1) then
c        Call the subroutine to test if there is any thyristor
c        turn on in rectification.
         Call rect(Vm,Rmm,Lmm,Lii,mesh1,mesh2,mm,ipold,icond,tsi,iback,
     +   jclose)
           If(max .EQ. 1) then
             max = 0
             go to 6000
           elseif(max .EQ. 0) then
             go to 2000
           else
             print*, "Error in rectifier"
             print*, max
             go to 3000
           endif
         endif
         If(t .EQ. 0.000) then
           Do 311 i = 1,3
  311      icio(i) = icirc(i)
           Call test(change, num, tr, t2, vt, ioy, shift)
           Call choose(mesh1, mesh2, ino, ifree, ioy, ifn, icio, ipass)
```

```
           If(ire .EQ. 0) then
             sl = sli
           elseif(ire .EQ. 1) then
             ire = 0
             sl = slr
           else
             print*, "Error in step lengh"
           endif
           mm(1) = mesh1
           mm(2) = mesh2
           mm(3) = ipass(1)
           mm(4) = ipass(2)
         print*, "111", (mm(i), i=1,4)
           go to 6000
         endif
 5001    ift = ifree(1) + ifree(2) + ifree(3)
         If(ift .GT. 0) Call check(ifree,ifn,cb,icirc,cm,ichh,mm,iq)
         If(ire .EQ. 0) then
           sl = sli
         elseif(ire .EQ. 1) then
           ire = 0
           sl = slr
         else
           print*, "Error in step lengh"
         endif
c         Determine the required matrix for the operation.
         Total = change(1) + change(2) + change(3)
         Do 710 i = 1,3
 710     icio(i) = icirc(i)
         Call test(change, num, tr, t2, vt, ioy, shift)
         If(Total .GE. 1 .OR. ichh .EQ. 1) then
           ichh = 0
           Do 121 i = 1,2
 121    ipold(i) = ipass(i)
           Call choose(mesh1, mesh2, ino, ifree, ioy, ifn, icio, ipass)
           If(ino .EQ. 1) then
             mm(1) = 25
             mm(2) = 25
             mm(3) = 7
             mm(4) = 7
           else
             Call rearr(ipold, ipass, mm, mesh1, mesh2)
           endif
           go to 6000
         elseif(ichh .EQ. 2) then
           ichh = 1
           go to 6000
         else
           go to 2002
         endif
 6000    If(power .EQ. 2) then
         Do 45 i = 1,16
         Cbm(i,1) = Cmod(i,mm(1))
         Cbt(1,i) = Cmod(i,mm(1))
         Cbm(i,2) = Cmod(i,mm(2))
 45      Cbt(2,i) = Cmod(i,mm(2))
         Do 46 i = 1,4
         Clm(5,i,1) = Cpwm(i,mm(3))
         Clm(5,i,2) = Cpwm(i,mm(4))
         Cml(5,1,i) = Cpwm(i,mm(3))
 46      Cml(5,2,i) = Cpwm(i,mm(4))
         elseif(power .EQ. 1) then
```

```
        Do 66 i= 1,10
        Cbm(i,1) = Crec(i,mm(1))
        Cbt(1,i) = Crec(i,mm(1))
        Cbm(i,2) = Crec(i,mm(2))
66      Cbt(2,i) = Crec(i,mm(2))
        Do 67 i = 1,4
        Clm(5,i,1) = Crpm(i,mm(3))
        Clm(5,i,2) = Crpm(i,mm(4))
        Cml(5,1,i) = Crpm(i,mm(3))
67      Cml(5,2,i) = Crpm(i,mm(4))
        idone = 1
        endif
        iii = 1
2000    If(iiii .EQ. 0 .AND. power .EQ. 2) go to 2002
        iii = 0
c       Determine if it is doing inversion or rectification, then
c       call the subroutine to calculate Rm, Xm for the converter.
        If(power .EQ. 2) then
          Call calc(Vm, Rmm, Lmm,Cbm,2,power)
        elseif(power .EQ. 1 .AND. icond .EQ. 1) then
          Call calc(Vm,Rmm,Lmm,Cbm,2,power)
        elseif(power .EQ. 1 .AND. icond .EQ. 2) then
          Call calc(Vm,Rmm,Lmm,Cbm,1,power)
        endif
        If(idone .EQ. 1 .AND. power .EQ. 1 .OR. power .EQ. 2)then
c       Call subroutine to calculate the inverse of Lii, and
c       Clm*Lii*Cml for the converter.
        If(ichh .EQ. 1) then
          ichh = 0
          Lii(5,1,2) = 0.000
          Lii(5,2,1) = 0.000
          If(iq .EQ. 2) then
            Lii(5,1,1) = 1.0/Lmm(5,1,1)
            Lii(5,2,2) = 0.000
          elseif(iq .EQ. 1) then
            Lii(5,1,1) = 0.000
            Lii(5,2,2) = 1.0/Lmm(5,2,2)
          endif
        elseif(ino .EQ. 1) then
            Do 3111 i = 1,2
            Do 3111 j = 1,2
3111     Lii(5,i,j) = 0.000
        else
            Call twin(Lmm, Lii, 5, aug, icond)
        endif
          Call recalc(Lii,5,ich)
          idone = 0
        endif
c        Determine if there is any change in the switching pattern
c        for the switches.
2002    Do 48 i = 1,5
        If(SW(i) .EQ. 1 .AND. t .GE. tc(i)) then
          isw = i
          close = 1
          open = 0
          Call opcl(Rmm,Lmm,Lii,power,close,open,State,
     +      rl1,icond,ich,Vm,Cbm,SW,ww,imot,iback)
          If(isw .EQ. 5) then
            jclose = 1
          else
            jclose = 0
          endif
```

```
        elseif(SW(i) .EQ. 2 .AND. t .GE. tc(i)) then
          isw = i
          open = 1
          close = 0
          Call opcl(Rmm,Lmm,Lii,power,close,open,State,
     +         rl1,icond,ich,Vm,Cbm,SW,ww,imot,iback)
        endif
 48     Continue
        If (jclose .EQ. 1 .AND. power .EQ. 1) then
          go to 5005
        elseif(jclose .EQ. 1.AND. power .EQ. 2) then
          go to 5001
        endif
c       Produce constant for Runge-Kutta.
 2100   Do 25 i = 1,5
        Do 25 j = 1,5
        cma(i,j) = cm(i,j)
 25     cmb(i,j) = cm(i,j)
c       Calculate the saturation function for each synch.machine.
        Do 44 i = 1,3
        sfi(i) = sf(i)
 44     Call sat(cb(i,5), i)
c       Determine if it is necessary to recalculate the stationary
c       generators.
        Do 667 iii = 1,2
        If(power .EQ. 2 .AND. sf(iii) .NE. sfi(iii))then
c       Calculate the inductance mesh for the three synchuous gen.
        Call synch(iii, ww(iii), Lmm,G)
c       Call the inverse of the inductanch matrix.
        Call inv(Lmm, Lii, 5, 5, iii, is)
c       Calculating Rmm+G.
        Do 533 i = 1,5
        Do 533 j = 1,5
 533    Rit(iii,i,j) = Rmm(iii,i,j) + G(iii,i,j)
        endif
 667    Continue
c       Calculate the values of the ta in the three phase
c       circuit.
 5004   Do 21 ij = 1, 4
        If(ij .NE. 1) ich = 0
        Do 98 i = 1,5
        Do 98 j = 1,5
 98     dd(i,j) = 0.000
c       Determine if it is necessary to calculate the inversions.
        If(ic .LE. in .OR. icalc .EQ. ini .OR. ich .EQ. 1) then
        Do 22 iii = 1,3
        If(power .EQ. 2 .AND. iii .NE. 3) go to 22
        ta1(iii) = tad(iii) + ww(iii)*sl*ttt(ij)
        If(ta1(iii) .GE. 3*twen) ta1(iii) = ta1(iii) - 3*twen
        ta2(iii) = ta1(iii) - twen
        ta3(iii) = ta1(iii) + twen
c       Setting up the constant for the trig functions.
        cos1(iii) = cos(ta1(iii))
        cos2(iii) = cos(ta2(iii))
        cos3(iii) = cos(ta3(iii))
        sin1(iii) = sin(ta1(iii))
        sin2(iii) = sin(ta2(iii))
        sin3(iii) = sin(ta3(iii))
c       Calculate the inductance mesh for the three synchuous gen.
        Call synch(iii, ww(iii), Lmm, G)
c       Call the inverse of the inductanch matrix.
        Call inv(Lmm, Lii, 5, 5, iii, is)
```

```
c       Calculating Rmm+G.
        Do 553 i = 1,5
        Do 553 j = 1,5
553     Rit(iii,i,j) = Rmm(iii,i,j) + G(iii,i,j)
22      Continue
c       Find out A and its inverse in diakoptics.
555     Do 23 i = 1,4
        Do 23 j = 1,4
23      A(1,i,j) = Ar(i,j)
        Do 24 iii = 1,3
        Do 26 i = 1,4
        Do 27 j = 1,5
        Con1(iii,i,j) = 0.000
        Do 27 k = 1,5
27      Con1(iii,i,j) = Con1(iii,i,j) + Clm(iii,i,k)*Lii(iii,k,j)
26      Continue
        Do 28 i = 1,4
        Do 28 j = 1,4
        Do 28 k = 1,5
28      A(1,i,j) = A(1,i,j) + Con1(iii,i,k)*Cml(iii,k,j)
24      Continue
        Call inv(A, Ai, 4, 1, 1,is)
c       Calculate Ai*Cim*Lii
        Do 49 ii = 1,5
        Do 49 i = 1,4
        Do 32 j = 1,ch(ii)
        Con(ii,i,j) = 0.000
        Do 32 k = 1,4
32      Con(ii,i,j) = Con(ii,i,j) + Ai(1,i,k)*Con1(ii,k,j)
49      Continue
        endif
c       Carry out diakoptic to solve the rate of change of current.
        it = 0
        Do 33 ih = 1,5
c       Determine if it is a diagonal and off-diagonal elements, if
c       jz = 1, it is diagonal and vica versa.
        jz = 0
        Do 33 iv = ih,5
        jz = jz+1
        it = it + 1
c       Determine if it is necessary to carry out the operation.
        If(ic .LE. in .OR. icalc .EQ. ini .OR. ich .EQ. 1) then
        Do 34 i = 1,ch(iv)
        Do 34 j = 1,ch(ih)
        If(jz .EQ. 1 .AND. i .EQ. j) then
          An(i,j) = 1.000
        else
          An(i,j) = 0.000
        endif
        Do 35 k = 1,4
35      An(i,j) = An(i,j) - Cml(iv,i,k)*Con(ih,k,j)
34      Continue
c       Working out Lii*Am
        Do 36 i = 1,ch(iv)
        Do 36 j = 1,ch(ih)
        Amm(it,i,j) = 0.000
        Do 37 k = 1,ch(iv)
37      Amm(it,i,j) = Amm(it,i,j) + Lii(iv,i,k)*An(k,j)
36      Continue
        endif
c       Call the subroutine to calculate the rate of change of current.
        ic1 = ch(iv)
```

```
      ic2 = ch(ih)
      If(ih .EQ. 1 .OR. ih .EQ. 2 .OR. ih .EQ. 3) then
        Call rate(Amm,ic1,ic2,iv,ih,ij,Vm,Rit,Lmm,power,it)
      elseif(ih .EQ. 4 .OR. ih .EQ. 5) then
        Call rate(Amm,ic1,ic2,iv,ih,ij,Vm,Rmm,Lmm,power,it)
      else
        print*, "Error in diakoptic"
        go to 3000
      endif
      If(jz .NE. 1) then
c     Determine if it is necessary to carry out the operations to
c     calculate Lii for the synchronous machine.
      If(ic .LE. in .OR. icalc .EQ. ini .OR. ich .EQ. 1) then
c     Work out the transpose of Amm.
        Do 38 i = 1,ch(ih)
        Do 38 j = 1,ch(iv)
 38   Amt(it,i,j) = Amm(it,j,i)
      endif
c     Call the subroutine to calculate the rate of change of current.
      If(iv .EQ. 1 .OR. iv .EQ. 2 .OR. iv .EQ. 3) then
        Call rate(Amt,ic2,ic1,ih,iv,ij,Vm,Rit,Lmm,power,it)
      elseif(iv .EQ. 4 .OR. iv .EQ. 5) then
        Call rate(Amt,ic2,ic1,ih,iv,ij,Vm,Rmm,Lmm,power,it)
      else
        print*, "Error in diakoptic"
        go to 3000
      endif
      endif
      If(icalc .EQ. ini) then
        icalc = 1
      elseif(icalc .LT. ini) then
        icalc = icalc + 1
      endif
 33   Continue
c     Updating the new cm.
      Do 199 i =1,5
      Do 199 j =1,ch(i)
      cm(i,j) = cm(i,j) + dd(i,j)*sl*gg(ij)
 199  cmb(i,j) = cma(i,j) + dd(i,j)*sl*h(ij)
 21   Continue
c     Updating the ic and icalc.
      ic = ic+1
      If(ic .GT. in .AND. ic .NE. 100)then
        ic = 99
        icalc = 1
      endif
c      Determine if it is rectification or inversion.
      If(power .EQ. 1) then
c      Calculate the derivate of dcm/dt.
      Do 331 i = 1,5
      Do 331 j = 1,5
      cmb(i,j) = cm(i,j)
 331  dd(i,j) = 0.000
      it = 0
      Do 333 ih = 1,5
c      Determine if it is a diagonal and off-diagonal elemnets.
      jz = 0
      Do 333 iv = ih,5
      jz = jz+1
      it = it + 1
c      Call the subroutine to calculate the rate of change of current.
      ic1 = ch(iv)
```

```
      ic2 = ch(ih)
      If(ih .EQ. 1 .OR. ih .EQ. 2 .OR. ih .EQ. 3) then
        Call rate(Amm,ic1,ic2,iv,ih,ij,Vm,Rit,Lmm,power,it)
      elseif(ih .EQ. 4 .OR. ih .EQ. 5) then
        Call rate(Amm,ic1,ic2,iv,ih,ij,Vm,Rmm,Lmm,power,it)
      else
        print*, "Error in diakoptic"
        go to 3000
      endif
      If(jz .NE. 1) then
c     Call the subroutine to calculate the rate of change of current.
      If(iv .EQ. 1 .OR. iv .EQ. 2 .OR. iv .EQ. 3) then
        Call rate(Amt,ic2,ic1,ih,iv,ij,Vm,Rit,Lmm,power,it)
      elseif(iv .EQ. 4 .OR. iv .EQ. 5) then
        Call rate(Amt,ic2,ic1,ih,iv,ij,Vm,Rmm,Lmm,power,it)
      else
        print*, "Error in diakoptic"
        go to 3000
      endif
      endif
 333  Continue
      endif
c      Determine if the convertor is turned off.
      If(State(5) .EQ. 0) then
        t = t+sl
        go to 7001
      endif
c      Calculate the branch voltage and current for converter.
      Do 777 i = 1,16
 777  cbi(i) = cb(5,i)
      If(power .EQ. 2) then
       If(ioy .EQ. 0) Call branch(Rmm,Lii,Vm,Cbm,5,16,16,power)
      elseif(power .EQ. 1) then
       Call branch(Rmm,Lii,Vm,Cbm,5,16,10,power)
      endif
c     Determine if there is any current discontinuity in rectification.
      If(power .EQ. 1) then
c     Call the subroutine to test if there is current discontinuity
        Call dis(jint,tsi)
        If(jint .EQ. 1) then
          jint = 2
          max = 2
          Do 111 i = 1,5
          Do 111 j = 1,5
          cm(i,j) = cma(i,j)
 111  cmb(i,j) = cma(i,j)
          go to 5004
        endif
      endif
c     Updating the new time and ta.
 7001 Do 51 i = 1,3
 51   tad(i) = tal(i)


c     Calculate the branch current, voltage and its torque for
c     three synchronous machine.
      Call torque(Rit,Lii,Vm,Te,G,ww,power,iback,imot)


c     Calculate the branch voltage and current for the bus-bar.
      Call branch(Rmm,Lii,Vm,Cbar,4,3,3,power)
c     Calculate exciter and diesel output for the generators.
```

```
      If(power .EQ. 1) then
        Do 177 i = 1,2
        Call excite(av, av1, av2, Vrt, Vm, i)
        Si(i,2) = Te(i)
        Call approx(xmat1, xmat2, ckf, ww, i)
177     Continue
      endif
c     Updating the output for the dc motor.
      If (imot .EQ. 0) Call motor (3,ww,Te)
      If(power .EQ. 1) then
        e(2) = Vb(iback,1)
        e(3) = Vb(iback,2)
        e(4) = Vb(iback,3)
      endif
c     Calculate the line voltage.
      Do 87 i = 1,5
      If(i .EQ. 5) then
        Vli(i,1) = Vb(i,2) - Vb(i,3)
        Vli(i,2) = Vb(i,3) - Vb(i,4)
        Vli(i,3) = Vb(i,4) - Vb(i,2)
      else
        Vli(i,1) = Vb(i,1) - Vb(i,2)
        Vli(i,2) = Vb(i,2) - Vb(i,3)
        Vli(i,3) = Vb(i,3) - Vb(i,1)
      endif
87    Continue
c     Print out the results.
      If(jc .EQ. jcc) then
      Write(60) t, (cb(1,i), i=1,6), (Vli(1,i), i =1,3), Vb(1,4)
      Write(70) t, (cb(2,i), i=1,6), (Vli(2,i), i =1,3), Vb(2,4)
      Write(80) t, (cb(3,i), i=1,6), (Vli(3,i), i =1,3), Vb(3,4)
      Write(90) t, (cb(4,i), i=1,3), (Vli(4,i), i =1,3)
      Write(92) t, (cb(5,i), i=5,16)
      Write(95) t, (cb(5,i), i=1,4), (Vb(5,i), i =1,4)
      Write(96) t, ((av(i,j), j=1,4), i=1,2)
      Write(97) t, ((S(i,j), j=1,3), i=1,2)
      Write(98) t, (ww(i), i=1,3), ca, cf, (Te(i), i=1,3)
      Write(99) t, (Vli(iback,i), i=1,3), (Vli(5,i), i=1,3)
      jiij = jiij + 1
      jc = 0
      else
      jc = jc + 1
      endif
      If(t .GT. Tstop) go to 3000
c      Determine if the converter is turned off.
      If(State(5) .EQ. 0) go to 2002
c      Determine if it is a rectifier.
      If(power .EQ. 1) go to 5005
      If(power .EQ. 2) go to 5001
c     Rewind the output binary files.
3000  Rewind(60)
      Rewind(70)
      Rewind(80)
      Rewind(90)
      Rewind(92)
      Rewind(95)
      Rewind(96)
      Rewind(97)
      Rewind(98)
      Rewind(99)
      Write(60) jiij
      Write(70) jiij
```

```
        Write(80) jjjj
        Write(90) jjjj
        Write(92) jjjj
        Write(95) jjjj
        Write(96) jjjj
        Write(97) jjjj
        Write(98) jjjj
        Write(99) jjjj
        Stop
        End


c       This subroutine is used to calculate the required values
c       of Rm, Xm, em in the converter set.
        Subroutine calc(Vm, Rmm, Lmm,Cbm,kk,power)
        Common/b1/Vb(5,16),e(16),h(4),gg(4),t
        Common/b3/Cbt(2,16),  Vdrop(5), Rba(3,3)
        Common/b4/Cd1(5), Cd2(5), Rbb(16,16), Xbb(16,16)
        Common/b5/rr(2,16), xx(2,16), idone
        Real Vm(5,5), Rmm(5,5,5), Lmm(5,5,5), Cbm(16,2)
        Integer i,j,k, kk, power
c       Determine if it is necessary to recalculate Lmm and Rmm.
        If(power .EQ. 1 .AND. idone .EQ. 0) go to 1
c       Calculate the value of Rm = Cbt*Rbb*Cbm
        Do 10 i = 1,kk
        Do 10 j = 1,16
        rr(i,j) = 0.000
        Do 20 k = 1,16
20      rr(i,j) = rr(i,j)+Cbt(i,k)*Rbb(k,j)
10      Continue
        Do 30 i = 1,kk
        Do 30 j = 1,kk
        Rmm(5,i,j) = 0.000
        Do 40 k = 1,16
40      Rmm(5,i,j) = Rmm(5,i,j)+rr(i,k)*Cbm(k,j)
30      Continue
c       Calculate the value of Xm = Cbt*Xbb*Cbm.
        Do 50 i = 1,kk
        Do 50 j = 1,16
        xx(i,j) = 0.000
        Do 60 k = 1,16
60      xx(i,j) = xx(i,j)+Cbt(i,k)*Xbb(k,j)
50      Continue
        Do 70 i = 1,kk
        Do 70 j = 1,kk
        Lmm(5,i,j) = 0.000
        Do 80 k = 1,16
80      Lmm(5,i,j) = Lmm(5,i,j)+xx(i,k)*Cbm(k,j)
70      Continue
c       Calculate the value of em = Cbt*e
1       Do 90 i = 1,kk
        Vm(5,i) = 0.000
        Do 90 j = 1,16
90      Vm(5,i) = Vm(5,i) + Cbt(i,j)*e(j)
        Return
        End


c       This subroutine calculates the inverse of a 2 by 2 matrix.
        Subroutine twin (x, xout, ni, aug, icon)
        Real x(5,5,5), xout(5,5,5),aug(2,2)
        Integer i,n
```

```
      If(icon .EQ. 2) then
        xout(5,1,1) = 1/x(5,1,1)
        go to 200
      endif
      aug(1,1) = x(ni,2,2)
      aug(2,2) = x(ni,1,1)
      aug(1,2) = -1 * x(ni,1,2)
      aug(2,1) = -1 * x(ni,2,1)
      det = x(ni,1,1)*x(ni,2,2) - x(ni,1,2)*x(ni,2,1)
      Do 10 i = 1,2
      Do 10 j = 1,2
      aug(i,j) = aug(i,j)/det
      xout(ni,i,j) = aug(i,j)
 10   continue
 200  Return
      End

c     This subroutine is used to determine if there is any
c     point of contact between the reference waves.
      Subroutine test (change, num, tr, t2, vt, ioy, shift)
      Common/b1/Vb(5,16),e(16),h(4),g(4),t
      Common/b2/Cd(16),Rb(16),Xb(16),sl,w,V1,ire, slr, sli
      Common/b6/t1, amp, am, ti, ti1, ti2, ti3
      Common/b7/twen, vsa, vsb, vsc, samp, ff
      Common/b8/za1, za2, za3, zb1, zb2, zb3, aa, bb
      Common/b9/icio(3), icirc(3), int
      Common/b10/nire, inter, ipass(2), iooo, ibbb
      Common/b11/tr1, tr2, nn1, nn2
      Integer change(3), num(3)
      Real tr, shift
      Do 3 i = 1,3
 3    change(i) = 0
      zt1 = 0.000
      zt2 = 0.000
      zt3 = 0.000
c     Calculate the amplitude in the triangle wave.
      t1 = t1 + sli
      t2 = t1
      t = t + sli
 1111 Continue
      If (t1 .LE. ti1) then
        vt = t1*am
      elseif (t1 .GT. ti1 .AND. t1 .LE. ti2) then
        vt = amp - (t1 - ti1)*am
      elseif (t1 .GT. ti2 .AND. t1 .LE. ti3) then
        vt = -amp + (t1 - ti2)*am
      else
        t1 = t1 - ti3
        go to 1111
      endif
c     Calculate the values of the three reference carrier waves
c     Updating the new time.
      w = ff*t
      vsa = -samp*sin(w)
      vsb = samp*sin(w + twen/shift)
      vsc = samp*sin(w - twen/shift)
c     Determine if there is any point of intersection.
      za1 = vsa - vt
      za2 = vsb - vt
      za3 = vsc - vt
      If(ire .EQ. 1) go to 2222
      inter = 0
```

```
      nn1 = 0
      nn2 = 0
      tr2 = 0.000
      tr1 = 0.000
      If((za1*zb1) .LE. 1.00e-07 .AND. zb1 .NE. 0.000) then
        change(1) = change(1) + 1
        num(1) = num(1) + 1
        aa = abs(zb1)
        bb = abs(za1)
        nn2 = nn1
        nn1 = 1
        tr2 = tr1
        tr1 = (t - sli) + sli*aa/(aa + bb)
        inter = inter + 1
      endif
      If((za2*zb2) .LE. 1.00e-07 .AND. zb2 .NE. 0.000)then
        change(2) = change(2) + 1
        num(2) = num(2) + 1
        aa = abs(zb2)
        bb = abs(za2)
        nn2 = nn1
        nn1 = 2
        tr2 = tr1
        tr1 = (t - sli) + sli*aa/(aa + bb)
        inter = inter + 1
      endif
      If((za3*zb3) .LE. 1.00e-07 .AND. zb3 .NE. 0.000)then
        change(3) = change(3) + 1
        num(3) = num(3) + 1
        aa = abs(zb3)
        bb = abs(za3)
        nn2 = nn1
        nn1 = 3
        tr2 = tr1
        tr1 = (t - sli) + sli*aa/(aa + bb)
        inter = inter + 1
      endif
      If(inter .EQ. 0) go to 1113
c     If point of intersection occurs, determine the smallest point and
c     use it as the step length for the next integrating step.
      If(tr2 .NE. 0.000) then
        If(tr2 .GT. tr1) then
          tr = tr1
          num(nn2) = 0
          change(nn2) = 0
        elseif(tr2 .LT. tr1) then
          tr = tr2
          num(nn1) = 0
          change(nn1) = 0
        endif
      else
        tr = tr1
      endif
      slr = tr - t + sli
      t2 = t2 - sli + slr
      t1 = t2
      ire = 1
      t = tr
      go to 1111
c     Determine which new thyristor is triggering if change in
c     conduction pattern occurs.
 2222 If(num(1) .NE. 0 .AND. ioy .EQ. 0) then
```

```
      If(icirc(1) .EQ. 1) then
        icirc(1) = 10
      elseif(icirc(1) .EQ. 10) then
        icirc(1) = 1
      elseif(icirc(1) .EQ. 4)then
        icirc(1) = 7
      elseif(icirc(1) .EQ. 7) then
        icirc(1) = 4
      endif
    endif
    If(num(2) .NE. 0 .AND. ioy .EQ. 0) then
      If(icirc(2) .EQ. 3) then
        icirc(2) = 12
      elseif(icirc(2) .EQ. 12) then
        icirc(2) = 3
      elseif(icirc(2) .EQ. 6) then
        icirc(2) = 9
      elseif(icirc(2) .EQ. 9) then
        icirc(2) = 6
      endif
    endif
    If(num(3) .NE. 0 .AND. ioy .EQ. 0) then
      If(icirc(3) .EQ. 5) then
        icirc(3) = 8
      elseif(icirc(3) .EQ. 8) then
        icirc(3) = 5
      elseif(icirc(3) .EQ. 2) then
        icirc(3) = 11
      elseif(icirc(3) .EQ. 11) then
        icirc(3) = 2
      endif
    endif
    If(num(1) .NE. 0 .AND. ioy .EQ. 1) then
      ioy = 0
      If(icirc(1) .EQ. 1) then
        icirc(1) = 4
      elseif(icirc(1) .EQ. 4) then
        icirc(1) = 1
      endif
    endif
    If(num(2) .NE. 0 .AND. ioy .EQ. 1) then
      ioy = 0
      If(icirc(2) .EQ. 3) then
        icirc(2) = 6
      elseif(icirc(2) .EQ. 6) then
        icirc(2) = 3
      endif
    endif
    If(num(3) .NE. 0 .AND. ioy .EQ. 1) then
      ioy = 0
      If(icirc(3) .EQ. 5) then
        icirc(3) = 2
      elseif(icirc(3) .EQ. 2) then
        icirc(3) = 5
      endif
    endif
    If(num(1) .NE. 0) za1 = 0.000
    If(num(2) .NE. 0) za2 = 0.000
    If(num(3) .NE. 0) za3 = 0.000
    num(1) = 0
    num(2) = 0
    num(3) = 0
```

```
c       Resetting the difference between two waves.
 1113   zb1 = za1
        zb2 = za2
        zb3 = za3
        Return
        End


c       This subroutine calculates the inverse of a m by m matrix x
c       and stores it into y.
        Subroutine inv(x, y, m, m1, kk, is)
        Dimension x(m1,m,m), y(m1,m,m), a(8,16)
        Integer is
        n = 2*m
        Do 3 i = 1, m
        Do 3 j = 1, m
        If(abs(x(kk,i,j)) .LT. 1.0e-08) x(kk,i,j) = 0.000
        a(i,j) = x(kk,i,j)
        a(i,j+m) = 0.000
        If(i .EQ. j) a(i,j+m) = 1.000
 3      Continue
        Do 14 k = 1, m
        Do 12 i = 1, m
        If(a(i,k) .EQ. 0.000 .AND. i .EQ. k) go to 17
        If(a(i,k) .EQ. 0.000) go to 12
        d = a(i,k)
        Do 11 j = 1, n
 11     a(i,j) = a(i,j)/d
 12     Continue
        Do 14 i = 1, m
        If(a(i,k) .EQ. 0.000 .OR. i .EQ. k) go to 14
        Do 13 j = 1, n
 13     a(i,j) = a(i,j) - a(k,j)
 14     Continue
        Do 33 i = 1, m
        d = a(i,i)
        Do 33 j = 1, n
        If(a(i,j) .EQ. 0.000) go to 33
        a(i,j) = a(i,j)/d
 33     Continue
        Do 16 i = 1, m
        Do 16 j = 1, m
 16     y(kk,i,j) = a(i,j+m)
        Return
 17     is = 1
        End


c       This subroutine is written to calculate the values of
c       inductance and rate of change of inductance of the
c       synchronous machine.
        Subroutine synch(k,ww,Lmm,G)
        Common/b13/ra,r4,p,z,tdo,td1,td2,freq
        Common/b14/xlad,xlaq,xmad,xmaq,xmf,xxmd,xxmq,xlf,xmfd
        Common/b15/xld,xlq,r1,r2,r3,r5,r6,rl1,rl2,rl3
        Common/b16/Con1(5,4,5),Clm(5,4,5),Cml(5,5,4),xLl(5,3),xopen,xclose
        Common/b17/cos1(3),cos2(3),cos3(3),sin1(3),sin2(3),sin3(3)
        Common/b18/Cm1(3,4,6),Cm2(3,4,6)
        Common/b20/cf1,cf2,sf(3),Oc(3,2),coo,const,ca1,ca2
        Common/b32/isw,iti(5),cb(5,16),Xba(3,3),Rload,Xload,srec,spwm
        Real ww,Lmm(5,5,5),G(3,5,5)
c       Updating the second matrix.
```

```
      Cm2(k,1,1) = xlad*sf(k)*cos1(k)**2.0 + xlaq*sin1(k)**2.0+xLl(k,1)
      Cm2(k,1,2) = xmad*sf(k)*cos1(k)*cos2(k) + xmaq*sin1(k)*sin2(k)
      Cm2(k,1,3) = xmad*sf(k)*cos1(k)*cos3(k) + xmaq*sin1(k)*sin3(k)

      Cm2(k,1,4) = xmf*sf(k)*cos1(k)
      Cm2(k,1,5) = xxmd*cos1(k)*sf(k)
      Cm2(k,1,6) = -xxmq*sin1(k)
      Cm2(k,2,1) = Cm2(k,1,2)
      Cm2(k,2,2) = xlad*sf(k)*cos2(k)**2.0 + xlaq*sin2(k)**2.0+xLl(k,2)
      Cm2(k,2,3) = xmad*sf(k)*cos2(k)*cos3(k) + xmaq*sin2(k)*sin3(k)
      Cm2(k,2,4) = xmf*sf(k)*cos2(k)
      Cm2(k,2,5) = xxmd*cos2(k)*sf(k)
      Cm2(k,2,6) = -xxmq*sin2(k)
      Cm2(k,3,1) = Cm2(k,1,3)
      Cm2(k,3,2) = Cm2(k,2,3)
      Cm2(k,3,3) = xlad*sf(k)*cos3(k)**2.0 + xlaq*sin3(k)**2+xLl(k,3)
      Cm2(k,3,4) = xmf*sf(k)*cos3(k)
      Cm2(k,3,5) = xxmd*cos3(k)*sf(k)
      Cm2(k,3,6) = -xxmq*sin3(k)
      Cm2(k,4,1) = Cm2(k,1,4)
      Cm2(k,4,2) = Cm2(k,2,4)
      Cm2(k,4,3) = Cm2(k,3,4)
      Cm2(k,4,4) = xlf*sf(k)
      Cm2(k,4,5) = xmfd*sf(k)
c     Updating the L matrix.
      Lmm(k,1,1) = Cm2(k,1,1) + Cm2(k,3,3) - 2*Cm2(k,1,3)
      Lmm(k,1,2) = Cm2(k,3,3) - (Cm2(k,1,3) + Cm2(k,2,3) - Cm2(k,1,2))
      Lmm(k,1,3) = Cm2(k,1,4) - Cm2(k,3,4)
      Lmm(k,1,4) = Cm2(k,1,5) - Cm2(k,3,5)
      Lmm(k,1,5) = Cm2(k,1,6) - Cm2(k,3,6)
      Lmm(k,2,2) = Cm2(k,2,2) + Cm2(k,3,3) - 2*Cm2(k,2,3)
      Lmm(k,2,3) = Cm2(k,2,4) - Cm2(k,3,4)
      Lmm(k,2,4) = Cm2(k,2,5) - Cm2(k,3,5)
      Lmm(k,2,5) = Cm2(k,2,6) - Cm2(k,3,6)
      Lmm(k,3,3) = Cm2(k,4,4)
      Lmm(k,3,4) = Cm2(k,4,5)
      Lmm(k,2,1) = Lmm(k,1,2)
      Lmm(k,3,1) = Lmm(k,1,3)
      Lmm(k,3,2) = Lmm(k,2,3)
      Lmm(k,4,1) = Lmm(k,1,4)
      Lmm(k,4,2) = Lmm(k,2,4)
      Lmm(k,4,4) = xld*sf(k)
      Lmm(k,5,1) = Lmm(k,1,5)
      Lmm(k,5,2) = Lmm(k,2,5)
      Lmm(k,4,3) = Lmm(k,3,4)
      Lmm(k,5,5) = xlq
c     Setting up some common values.
      G11 = -ww*(2.0*xlad*sf(k)*cos1(k)*sin1(k)
     +      -2.0*xlaq*cos1(k)*sin1(k))
      G22 = -ww*(2.0*xlad*sf(k)*cos2(k)*sin2(k)
     +      -2.0*xlaq*cos2(k)*sin2(k))
      G33 = -ww*(2.0*xlad*sf(k)*cos3(k)*sin3(k)
     +      -2.0*xlaq*cos3(k)*sin3(k))
c      Updating the two matrix for calculating the branch voltage.
      Cm1(k,1,1) = r1 + G11
      Cm1(k,1,2) = -ww*(xmad*sf(k)*(sin1(k)*cos2(k)+cos1(k)*sin2(k))-
     +            xmaq*(cos1(k)*sin2(k)+sin1(k)*cos2(k)))
      Cm1(k,1,3) = -ww*(xmad*sf(k)*(sin1(k)*cos3(k)+cos1(k)*sin3(k))-
     +            xmaq*(cos1(k)*sin3(k)+sin1(k)*cos3(k)))
      Cm1(k,1,4) = -ww*xmf*sf(k)*sin1(k)
      Cm1(k,1,5) = -ww*xxmd*sf(k)*sin1(k)
      Cm1(k,1,6) = -ww*xxmq*cos1(k)
```

```
      Cm1(k,2,1) = Cm1(k,1,2)
      Cm1(k,2,2) = r2+G22
      Cm1(k,2,3) = -ww*(xmad*sf(k)*(sin2(k)*cos3(k)+cos2(k)*sin3(k))-
     +            xmaq*(cos2(k)*sin3(k)+sin2(k)*cos3(k)))
      Cm1(k,2,4) = -ww*xmf*sf(k)*sin2(k)
      Cm1(k,2,5) = -ww*xxmd*sf(k)*sin2(k)
      Cm1(k,2,6) = -ww*xxmq*cos2(k)
      Cm1(k,3,1) = Cm1(k,1,3)
      Cm1(k,3,2) = Cm1(k,2,3)
      Cm1(k,3,3) = r3+ G33
      Cm1(k,3,4) = -ww*xmf*sf(k)*sin3(k)
      Cm1(k,3,5) = -ww*xxmd*sf(k)*sin3(k)
      Cm1(k,3,6) = -ww*xxmq*cos3(k)
      Cm1(k,4,1) = Cm1(k,1,4)
      Cm1(k,4,2) = Cm1(k,2,4)
      Cm1(k,4,3) = Cm1(k,3,4)
      Cm1(k,4,4) = r4
c     Updating the new G matrix.
      G(k,1,1) = G11 + G33 - 2*Cm1(k,1,3)
      G(k,1,2) = G33 - (Cm1(k,1,3) + Cm1(k,2,3) - Cm1(k,1,2))
      G(k,1,3) = Cm1(k,1,4) - Cm1(k,3,4)
      G(k,1,4) = Cm1(k,1,5) - Cm1(k,3,5)
      G(k,1,5) = Cm1(k,1,6) - Cm1(k,3,6)
      G(k,2,2) = G22 + G33 - 2*Cm1(k,2,3)
      G(k,2,3) = Cm1(k,2,4) - Cm1(k,3,4)
      G(k,2,4) = Cm1(k,2,5) - Cm1(k,3,5)
      G(k,2,5) = Cm1(k,2,6) - Cm1(k,3,6)
      G(k,2,1) = G(k,1,2)
      G(k,3,1) = G(k,1,3)
      G(k,4,1) = G(k,1,4)
      G(k,5,1) = G(k,1,5)
      G(k,3,2) = G(k,2,3)
      G(k,4,2) = G(k,2,4)
      G(k,5,2) = G(k,2,5)
      Return
      End


c     This subroutine is written to calculate the rate of change
c     of current. It solves the equation dIa/dt=A(Vm-Rtm*Ia)
      Subroutine rate(A,k1,k2,n1,n2,nn,Vm,Rtm,Lmm,power,it)
      Common/b1/Vb(5,16),e(16),h(4),gg(4),t
      Common/b2/Cd(16),Rb(16),Xb(16),sl,w,V1,ire, slr, sli
      Common/b3/Cbt(2,16), Vdrop(5), Rba(3,3)
      Common/b19/cm(5,5), cma(5,5), cmb(5,5), Cmast(6,5), dd(5,5)
      Real A(16,5,5), Vm(5,5), Rtm(5,5,5), Lmm(5,5,5), Vmm(5),
     +     d1(5)
      Integer power
      Integer n1,n2,nn
      Do 10 i = 1,k2
      Vdrop(i) = 0.000
      Do 10 j = 1,k2
 10   Vdrop(i) = Vdrop(i) + Rtm(n2,i,j)*cmb(n2,j)
c     Work out Vm - Rtm*Im
      Do 11 i = 1,k2
 11   Vdrop(i) = Vm(n2,i) - Vdrop(i)
c     Work out A(Vm-Rtm*Im)
      Do 12 i = 1,k1
      d1(i) = 0.000
      Do 12 j = 1,k2
 12   d1(i) = d1(i) + A(it,i,j)*Vdrop(j)
c     Work out the constant in the runge-kutta.
```

```
      Do 13 i = 1,k1
13    dd(n1,i) = dd(n1,i) + d1(i)
      Return
      End


c     This subroutine is written to calculate the dq parameters
c     for the synchnous machine.
      Subroutine para(Vm,cm,vb,Rmm)
      Common/b12/wo,pi,fo,xd,xmd,xq,xd1,xd2,xq2,xo,xmq
      Common/b13/ra,r4,p,z,tdo,td1,td2,freq
      Common/b14/xlad,xlaq,xmad,xmaq,xmf,xxmd,xxmq,xlf,xmfd
      Common/b15/xld,xlq,r1,r2,r3,r5,r6,rl1,rl2,rl3
      Common/b18/Cm1(3,4,6),Cm2(3,4,6)
      Common/b20/cf1,cf2,sf(3),Oc(3,2),coo,const,ca1,ca2
      Common/b32/isw,iti(5),cb(5,16),Xba(3,3),Rload,Xload,srec,spwm
      Real Vm(5,5), cm(5,5), vb(5,16), Rmm(5,5,5)
c     Calculating the required from the given parameter.
      wo = 2*pi*fo
      r1 = ra * z
      xa1 = xd - xmd
      xff = xmd*xmd/(xd -xd1)
      xf1 = xff - xmd
      x1 = (xd2 - xa1)*xmd*xf1/(xmd*xf1 - xff*(xd2 - xa1))
      xq1 = xmq*(xq2 - xa1)/(xq -xq2)
      tdodd = xd*td1*td2/tdo/xd2
      rd = xmd*xf1/(xmd + xf1)/wo/tdodd
      xffo = tdo*wo*r4
      dfn = sqrt(2.0*xff*z/3.0/xffo)
      xa2 = z*(xd - xq)/3.0/wo
      xabo = z*(xd - xo)/3.0/wo - 0.5*xa2
      xao = z*xo/wo + 2.0*xabo
      xlad = xao + xa2
      xlaq = xao - xa2
      xmad = 2.0*xabo + xa2
      xmaq = 2.0*xabo - xa2
      xlf = 2.0*xff*z/(3.0*wo*dfn*dfn)
      xmf = 2.0*z*xmd/(3.0*wo*dfn)
      xdd = xmd + x1
      xld = 2.0*z*xdd/27.0/wo
      xxmd = 2.0*z*xmd/9.0/wo
      xxmq = 2.0*z*xmq/9.0/wo
      r5 = 2.0*rd*z/27.00
      xmfd = xmf/3.0
      xkq = xmq + xq1
      xlq = 2.0*z*xkq/27.0/wo
      tddd = (xmd*xa1*xf1/(xmd*xa1 + xmd*xf1 + xa1*xf1) + x1)/wo/r5
      tqdd = 1.5*tddd
      rq = (xmq*xa1/(xmq + xa1) +xq1)/wo/tqdd
      r6 = 2.0*z*rq/27.0
      r2 = r1
      r3 = r1
c     Calculate the open circuit characteristics for the
c     saturation function.
      cf1 = Oc(1,2)
      cf2 = Oc(3,2)
      coo = Oc(1,1)/cf1
      const = Oc(3,1)/(cf2*coo)
c     Updating the Rm matrix.
      Rmm(1,1,1) = r1+r3+rl1+rl3
      Rmm(1,1,2) = r3+rl3
      Rmm(1,2,1) = Rmm(1,1,2)
```

```
      Rmm(1,2,2) = r2+r3+rl2+rl3
      Rmm(1,3,3) = r4
      Rmm(1,4,4) = r5
      Rmm(1,5,5) = r6
      Do 1 i = 2,3
      Rmm(i,1,1) = Rmm(1,1,1)
      Rmm(i,1,2) = Rmm(1,1,2)
      Rmm(i,2,1) = Rmm(1,2,1)
      Rmm(i,2,2) = Rmm(1,2,2)
      Rmm(i,3,3) = Rmm(1,3,3)
      Rmm(i,4,4) = Rmm(1,4,4)
 1    Rmm(i,5,5) = Rmm(1,5,5)
c     Updating the new field current.
      cm(1,3) = (2**0.5)*200.00/((3**0.5)*xmf*2.00*pi*freq)
      Vm(1,3) = cm(1,3)*r4


      vb(1,4) = Vm(1,3)
      cb(1,4) = cm(1,3)
      Do 10 ii = 2,3
      cm(ii,3) = cm(1,3)
      Vm(ii,3) = Vm(1,3)
      cb(ii,4) = cb(1,4)
 10   vb(ii,4) = Vm(1,4)
      Return
      End

c     This subroutine is used to calculated the saturation function
c     for the synchronous machine.
      Subroutine sat(a,i)
      Common/b20/cf1,cf2,sf(3),Oc(3,2),coo,const,ca1,ca2
      Real a
c     Calculate the saturation function.
      If(abs(a) .LE. cf1) then
        sf(i) = 1.000
      elseif(abs(a) .GT. cf2) then
        sf(i) = const
      else
        sf(i) = -(1.000 - const)*(abs(a) - cf1)/(cf2 - cf1) + 1.000
      endif
      Return
      End


c     This subroutine is written to deal with the open and close
c     circuits for the system.
      Subroutine opcl(Rmm,Lmm,Lii,power,close,open,State,
     +               rl1,icond,ich,Vm,Cbm,SW,ww,imot,iback)
      Common/b1/Vb(5,16),e(16),h(4),gg(4),t
      Common/b2/Cd(16),Rb(16),Xb(16),sl,w,V1,ire, slr, sli
      Common/b3/Cbt(2,16),  Vdrop(5), Rba(3,3)
      Common/b4/Cd1(5), Cd2(5), Rbb(16,16), Xbb(16,16)
      Common/b5/rr(2,16), xx(2,16), idone
      Common/b19/cm(5,5), cma(5,5), cmb(5,5),Cmast(6,5),dd(5,5)
      Common/b16/Con1(5,4,5),Clm(5,4,5),Cml(5,5,4),xLl(5,3),xopen,xclose
      Common/b31/aug(2,2),Ar(4,4),Ar1(2,4,4)
      Common/b32/isw,iti(5),cb(5,16),Xba(3,3),Rload,Xload,srec,spwm
      Common/b33/tyr,diod,tdc
      Real Rmm(5,5,5), Lmm(5,5,5), Lii(5,5,5), rl1, Vm(5,5),Cbm(16,2),
     +  ww(5)
      Integer power, close, open, isw, State(5), icond, ich, SW(6),
     +      imot
```

```
c        Test if it is a closed circuit.
         kk = 0
         If(close .EQ. 1) then
           Do 10 i = 1,3
 10      xLl(isw,i) = xclose
           close = 0
           iti(isw) = 0
           SW(isw) = 0
           State(isw) = 1
           If(isw .EQ. 5) then
             If(State(1) .EQ. 1 .AND. State(4) .EQ. 1) then
               iback = 1
             elseif(State(2) .EQ. 1) then
               iback = 2
             else
               print*, "Error : Incomplete path for the converter"
             endif
           endif
           kk = 1
           go to 300
         endif
c        Test if it is an opened circuit.
         If(open .EQ. 1) then
           Do 20 ii = 1,3
           If (isw .EQ. 5) then
             jj = ii+1
           else
             jj = ii
           endif
           vvv =  abs(cb(isw,jj))
           If(vvv .LE. 0.200 .AND. xLl(isw,ii) .NE. xopen) then
             xLl(isw,ii) = xopen
             iti(isw) = iti(isw) + 1
             kk = 1
           endif
 20      Continue
           If(kk .EQ. 1) then
             If(iti(isw) .EQ. 2 .OR. iti(isw) .EQ. 3) then
              Do 35 i = 1,3
              xLl(isw,i) = xopen
              If(isw .EQ. 5) Xbb(isw,i+1) = xLl(isw,i)
 35           Continue
              Do 36 i = 1,16
             If(isw .EQ. 5) Vb(isw,i) = 0.000
 36           cb(isw,i) = 0.000
              Do 37 i = 1,2
 37           cm(isw,i) = 0.000


              open = 0
              iti(isw) = 0
              SW(isw) = 0
              State(isw) = 0
              go to 300
             else
              go to 300
             endif
           elseif(kk .EQ. 0) then
              go to 400
           endif
         endif
c        Resetting the speed of the synch. mot.
```

```
300   If(State(3) .EQ. 0 .AND. kk .EQ. 1 .AND. isw .EQ. 3) then
        ww(3) = 0.000
        imot = 1
        go to 400
      elseif(State(3) .EQ. 1 .AND. kk .EQ. 1 .AND. isw .EQ. 3) then
        ww(3) = ww(1)
        imot = 0
        go to 400
      endif
c     Resetting the bus bar inductances matrix.
      If(isw .EQ. 4) then
        Xba(1,1) = xLl(4,1)
        Xba(2,2) = xLl(4,2)
        Xba(3,3) = xLl(4,3)
        Lmm(4,1,1) = xLl(4,1)+xLl(4,3)
        Lmm(4,1,2) = xLl(4,3)
        Lmm(4,2,1) = xLl(4,3)
        Lmm(4,2,2) = xLl(4,2)+xLl(4,3)
        Call twin(Lmm,Lii,4,aug,0)
        Call recalc(Lii,4,ich)
        go to 400
      endif
c     Resetting the converter inductances matrix.
      If(isw .EQ. 5 .AND. State(5) .EQ. 1) then
        If(iti(5) .GT. 0) then
          Do 30 i = 2,4
30    Xbb(i,i) = xLl(5,i-1)
        else
          Do 40 i=1,16
          If(i .LE. 4 .AND. i .NE. 1) Xbb(i,i) = xclose
          If(i .GT. 4 .AND. i .LE. 10 ) Xbb(i,i) = tyr
          If(i .GT. 10) Xbb(i,i) = diod
40     Continue
        If(power .EQ. 1) then
          Rbb(1,1) = Rload
          Xbb(1,1) = Xload
          sli = srec
        elseif(power .EQ. 2) then
          Rbb(1,1) = rl1
          Xbb(1,1) = tdc
          sli = spwm
        endif
        endif
       If(icond .EQ. 2) then
         Call calc(Vm, Rmm, Lmm,Cbm,2,power)
       elseif(icond .EQ. 1) then
         Call calc(Vm, Rmm, Lmm,Cbm,2,power)
       else
         print*, "Error in icond"
       endif
       Call twin(Lmm, Lii, 5,aug,icond)
       Call recalc(Lii,5,ich)
      elseif(isw .EQ. 5 .AND. State(5) .EQ. 0) then
        Do 50 i = 1,16
50    Xbb(i,i) = xopen
        Do 60 i = 1,4
        Do 60 j = 1,4
        Lii(5,i,j) = 0.000
        Lmm(5,i,j) = 0.000
60    Rmm(5,i,j) = 0.000
        sli = srec
      endif
```

```
400   Return
      End



c     This subroutine is written to recalculate the value of
c     Ar where Ar = Clm*Lii*Cml and Con1 where Con1 = Clm*Lii
c     for the diakoptics.
      Subroutine recalc(Lii,kk,ich)
      Common/b16/Con1(5,4,5),Clm(5,4,5),Cml(5,5,4),xLl(5,3),xopen,xclose
      Common/b31/aug(2,2),Ar(4,4),Ar1(2,4,4)
      Real Lii(5,5,5)
      ich = 1
c     Firstly, calculate the value of Ar.
      Do 10 i = 1,4
      Do 10 j = 1,2
      Con1(kk,i,j) = 0.000
      Do 11 k = 1,2
11    Con1(kk,i,j) = Con1(kk,i,j) + Clm(kk,i,k)*Lii(kk,k,j)
10    Continue
      Do 13 i = 1,4
      Do 13 j = 1,4
      Ar1(kk-3,i,j) = 0.000
      Do 14 k = 1,2
14    Ar1(kk-3,i,j) = Ar1(kk-3,i,j) + Con1(kk,i,k)*Cml(kk,k,j)
13    Continue
      Do 15 i = 1,4
      Do 15 j = 1,4
15    Ar(i,j) = Ar1(1,i,j) + Ar1(2,i,j)
      Return
      End


c     This subroutine is written to calculate the output of
c     the exciter.
      Subroutine excite(av, av1, av2, Vrt, Vm, ni)
      Common/b1/Vb(5,16),e(16),h(4),gg(4),t
      Common/b2/Cd(16),Rb(16),Xb(16),sl,w,V1,ire, slr, sli
      Common/b21/Vh, Vv, Vp, three, ppi, V5, Vmin, Vmax
      Common/b22/ava(4), avb(4), avc(4), avd(4)
      Common/b23/aref(2,2), d1(4), Vft(3), iexc
      Common/b24/time(3), cycle, dummy(3,3,2), Tot
      Real av(4,4), av1(4,4), av2(4,4), Vrt(4), Vm(5,5)
      Integer ni
c     Calculate the input voltage (Vt) for the exciter.
      time(ni) = time(ni)+sl
c     Records positive and negative voltage for each phase
c     and calculates the average voltage at the end of each
c     cycle.
      If(time(ni) .LT. cycle) then
        Do 50 i = 1,3
        If(Vb(ni,i) .GT. dummy(ni,i,1)) dummy(ni,i,1) = Vb(ni,i)
        If(Vb(ni,i) .LT. dummy(ni,i,2)) dummy(ni,i,2) = Vb(ni,i)
50    Continue
      else
        Do 51 i = 1,3
        If(Vb(ni,i) .GT. dummy(ni,i,1)) dummy(ni,i,1) = Vb(ni,i)
        If(Vb(ni,i) .LT. dummy(ni,i,2)) dummy(ni,i,2) = Vb(ni,i)
51    Continue
        Tot = 0.000
        Do 52 i = 1,3
        Do 52 j = 1,2
```

```
          Tot = Tot + abs(dummy(ni,i,j))
 52     Continue
          aref(ni,2) = Tot/6.00
          time(ni) = 0.000
          Do 53 i = 1,3
          Do 53 j = 1,2
 53     dummy(ni,i,j) = 0.000
          iexc = 1
        endif
        If(iexc .NE. 1) go to 1
        Do 8 i = 1,4
        ava(i) = av(ni,i)
        avb(i) = av(ni,i)
        avc(i) = av(ni,i)
 8      avd(i) = av(ni,i)
c       In order to calculate the result actually, the voltage
c       output is calculate first. However, reintegeration is
c       carried out if V4 is larger than the practical limit
c       of the exciter.
c       Test if the value of V4 is greater than the practical
c       limit.
        If(av(ni,1) .GT. Vmax) then
          V5 = Vmax
          go to 2000
        elseif(av(ni,1) .LT. Vmin) then
          V5 = Vmin
          go to 2000
        endif
c       Put this into for dav/dt = Vrt + av1*av
        Vrt(1) = 0.000
        Do 5 i = 1,2
 5      Vrt(1) = Vrt(1) + av2(1,i)*aref(ni,i)
        Do 10 ii = 1,4
        Do 20 i = 1,4
        d1(i) = 0.000
        Do 25 j = 1,4
 25     d1(i) = d1(i) + av1(i,j)*avb(j)
 20     Continue
        Do 30 i = 1,4
 30     d1(i) = d1(i) + Vrt(i)
        Do 40 i = 1,4
        av(ni,i) = av(ni,i) + d1(i)*sl*gg(ii)
 40   avb(i) = ava(i) + d1(i)*sl*h(ii)
 10   Continue

c       Test if the value of V4 is larger than the practical limit.
        If(av(ni,1) .GE. Vmin .AND. av(ni,1) .LE. Vmax) then
          Vm(ni,3) = av(ni,4)
          go to 1000
        elseif(av(ni,1) .GT. Vmax) then
          V5 = Vmax
          go to 2500
        elseif(av(ni,1) .LT. Vmin) then
          V5 = Vmin
          go to 2500
        else
          Print*, "Error in the comparator."
          go to 1000
        endif
c       Calculate the value of V4.
 2000 Vrt(1) = 0.000
        Do 15 i = 1,2
```

```
15    Vrt(1) = Vrt(1) + av2(1,i)*aref(ni,i) + av1(1,i+1)*ava(i+1)
      Do 16 ii = 1,4
      d1(1) = 0.000
      d1(1) = av1(1,1)*avb(1) + Vrt(1)
      av(ni,1) = av(ni,1) + sl*d1(1)*gg(ii)
      avb(1) = ava(1) + sl*d1(1)*h(ii)
16    Continue
c     Reintegeration is carried out.
c     Get this into form dav/dt = Vft + av1*av

2500  Do 49 i = 2,4
49    av(ni,i) = avc(i)
      Do 57 i = 1,3
      Vft(i) = 0.000
57    Vft(i) = av1(i+1,1)*V5
      Do 55 ii = 1,4
      Do 60 i = 1,3
      d1(i) = 0.000
      Do 65 j = 1,3
65    d1(i) = d1(i) + av1(i+1,j+1)*avd(j+1)
60    Continue
      Do 70 i = 1,3
70    d1(i) = d1(i) + Vft(i)


      Do 80 i = 1,3
      av(ni,i+1) = av(ni,i+1) + d1(i)*sl*gg(ii)
80    avd(i+1) = avc(i+1) + d1(i)*sl*h(ii)
55    Continue
c     Making sure the field voltage never goes negative.
1000  If(av(ni,4) .LT. 0.000) av(ni,4) = 0.000
      Vm(ni,3) = av(ni,4)
1     Return
      End


c     This is a simple model of a diesel engine according
c     to the diesel engine block diagram given.
      Subroutine approx(xmat1, xmat2, ckf, ww, ni)
      Common/b1/Vb(5,16),e(16),h(4),gg(4),t
      Common/b2/Cd(16),Rb(16),Xb(16),sl,w,V1,ire, slr, sli
      Common/b23/aref(2,2), d1(4), Vft(3), iexc
      Common/b26/S(3,3), Si(2,2), d2(3), Xji, S1(3), S2(3), zx
      Real xmat1(3,3), xmat2(3,2), ww(5)
c     Carry out runge-kutta method.
      Do 15 i = 1,3
      S1(i) = S(ni,i)
15    S2(i) = S(ni,i)
      Do 20 ii = 1,4
      Do 30 i = 1,3
      d1(i) = 0.000
      Do 30 j = 1,3
30    d1(i) = d1(i) + xmat1(i,j)*S1(j)
      Do 40 i = 1,3
      d2(i) = 0.000
      Do 40 j = 1,2
40    d2(i) = d2(i) + xmat2(i,j)*Si(ni,j)
      Do 45 i = 1,3
45    d1(i) = d1(i) + d2(i)
      Do 50 i = 1,3
      S(ni,i) = S(ni,i) + sl*d1(i)*gg(ii)
50    S1(i) = S2(i) + sl*d1(i)*h(ii)
```

```
        ww(ni) = S(ni,3)
 20     Continue
        Return
        End


c       This subroutine is written to calculate motor response,
c       the state variables output are the armature current ca,
c       the field current cf and the rotational speed of the
c       motor ww.
        Subroutine motor(ni,ww,Te)
        Common/b1/Vb(5,16),e(16),h(4),gg(4),t
        Common/b2/Cd(16),Rb(16),Xb(16),sl,w,V1,ire, slr, sli
        Common/b27/ca,cf,Va,Vf,rtt,rt,zxy,xt,xf,ck,rf,ckf,Ji
        Common/b30/dd1,dd2,dd3,Vdc,sign,db1,db2,zfy,zyh
        Real ww(5),Te(3)
c       Carry out Runge-Kutta for the mechanical equations.
        ca1 = ca
        ca2 = ca
        cfa = cf
        cfb = cf
        ww1 = ww(ni)
        ww2 = ww(ni)
        Do 601 i = 1,4
        db1 = dd1
        db2 = dd2
        dd1 = (Vdc + sign*ck*ww1 - rtt*ca1 -zfy*db2)/xt
        dd2 = (Vf - rf*cfa - zyh*db1)/xf
        dd3 = (sign*(Te(ni) - ck*ca1*cfa) - ckf*ww1)/Ji
        ca = ca + dd1*sl*gg(i)
        cf = cf + dd2*sl*gg(i)
        ww(ni) = ww(ni) + dd3*sl*gg(i)
        ca1 = ca2 + dd1*sl*h(i)
        cfa = cfb + dd2*sl*h(i)
        ww1 = ww2 + dd3*sl*h(i)
 601    Continue
        Return
        End



c       This subroutine is used to calculate the branch current, voltage
c       and torque for the generators.
        Subroutine torque(Rit,Lii,Vm,Te,G,ww,power,iback,imot)
        Common/b1/Vb(5,16),e(16),h(4),gg(4),t
        Common/b2/Cd(16),Rb(16),Xb(16),sl,w,V1,ire, slr, sli
        Common/b3/Cbt(2,16), Vdrop(5), Rba(3,3)
        Common/b4/Cd1(5), Cd2(5), Rbb(16,16), Xbb(16,16)
        Common/b18/Cm1(3,4,6),Cm2(3,4,6)
        Common/b16/Con1(5,4,5),Clm(5,4,5),Cml(5,5,4),xLl(5,3),xopen,xclose
        Common/b19/cm(5,5), cma(5,5), cmb(5,5),Cmast(6,5), dd(5,5)
        Common/b32/isw,iti(5),cb(5,16),Xba(3,3),Rload,Xload,srec,spwm
        Real Rit(5,5,5), Lii(5,5,5), Vm(5,5), Te(5), zz(5), Ct(1,5),
     +   G(3,5,5),v(5),ww(5)
        Integer power,imot
        Do 1 iy = 1,3
c       Determine if it is rectification or convertion.
        If(power .EQ. 2) then
c       Updating the new dIm/dt
        Do 91 i = 1,5
        vdrop(i) = 0.000
        Do 91 j = 1,5
```

```
        vdrop(i) = vdrop(i) + Rit(iy,i,j)*cm(iy,j)
91      Continue
        Do 110 i = 1, 5
110     v(i) = Vm(iy,i) - vdrop(i)
        Do 111 i = 1, 5
        Cd2(i) = 0.000
        Do 111 j = 1, 5
111     Cd2(i) = Cd2(i) + Lii(iy,i,j)*v(j)
        endif
c       Updating the new branch current and dIb/dt
        Do 115 i = 1, 6
        cb(iy,i) = 0.000
        Cd(i) = 0.000
        Do 115 j = 1,5
        cb(iy,i) = cb(iy,i) + Cmast(i,j)*cm(iy,j)
        If(power .EQ. 1) then
          Cd(i) = Cd(i) + Cmast(i,j)*dd(iy,j)
        elseif(power .EQ. 2) then
          Cd(i) = Cd(i) + Cmast(i,j)*Cd2(j)
        endif
115     Continue
c       Calculate the branch voltage for the synchronous machine.
        If(power .EQ. 1) then
c       Resetting the matrix.
          Do 199 i = 1,3
199     Cm2(iy,i,i) = Cm2(iy,i,i) - xLl(iy,i)
        endif
c       Calculate Cmast*Ib.
        Do 201 i = 1,4
        Rb(i) = 0.000
        Do 211 k = 1,6
211     Rb(i) = Rb(i) + Cm1(iy,i,k)*cb(iy,k)
201     Continue
        Do 221 i = 1,4
        Xb(i) = 0.000
        Do 241 k = 1,6
241     Xb(i) = Xb(i) + Cm2(iy,i,k)*Cd(k)
221     Continue
        Do 251 i = 1,4
251     Vb(iy,i) = Rb(i) + Xb(i)
c       If it is in re-generation , generator 1 and 2 is stationary,
c       so their output torque is zero, hence, the torque is
c       unnecessary to calculate.
        If(power .EQ. 2 .AND. iy .NE. 3 .OR. imot .EQ. 1) go to 1
        If(power .EQ. 1 .AND. imot .EQ. 1 .AND. iy .EQ. 3) go to 1
c       Find the transpose of the mesh current.
        Do 301 i = 1,5
301     Ct(1,i) = cm(iy,i)/ww(iy)
c       Find the electric torque.
        Do 351 i = 1,5
        zz(i) = 0.000
        Do 351 j = 1,5
351     zz(i) = zz(i) + G(iy,i,j)*cm(iy,j)
        Te(iy) = 0.000
        Do 401 j = 1,5
401     Te(iy) = Te(iy) + Ct(1,j)*zz(j)
        Te(iy) = abs(Te(iy))
1       Continue
        Return
        End
```

```
c     This subroutine is written to calculate the branch current and
c     voltage for both the bus bar and the converter.
      Subroutine branch(a,b,c,d,n1,n2,n3,power)
      Common/b1/Vb(5,16),e(16),h(4),gg(4),t
      Common/b2/Cd(16),Rb(16),Xb(16),sl,w,V1,ire, slr, sli
      Common/b3/Cbt(2,16), Vdrop(5), Rba(3,3)
      Common/b4/Cd1(5), Cd2(5), Rbb(16,16), Xbb(16,16)
      Common/b18/Cm1(3,4,6),Cm2(3,4,6)
      Common/b19/cm(5,5), cma(5,5), cmb(5,5),Cmast(6,5), dd(5,5)
      Common/b32/isw,iti(5),cb(5,16),Xba(3,3),Rload,Xload,srec,spwm
      Real a(5,5,5), b(5,5,5), c(5,5), d(n2,2), v(2)
      Integer n1, n2, n3,power
c     Updating the new dIm/dt.
      v(1) = 0.000
      v(2) = 0.000
      Do 90 i = 1,2
      Do 90 j = 1,2
      v(i) = v(i) + a(n1,i,j)*cm(n1,j)
 90   Continue
      Do 100 i = 1,2
 100  v(i) = c(n1,i) - v(i)
      Cd2(1) = 0.000
      Cd2(2) = 0.000
      Do 110 i = 1,2
      Do 110 j = 1,2
 110  Cd2(i) = Cd2(i) + b(n1,i,j)*v(j)
c     Updating the new branch current and dIb/dt
      Do 115 i = 1,n3
      cb(n1,i) = 0.000
      Cd(i) = 0.000
      Do 115 j = 1,2
      cb(n1,i) = cb(n1,i) + d(i,j)*cm(n1,j)
      Cd(i) = Cd(i) + d(i,j)*Cd2(j)
 115  Continue
c     If it is a bus bar, calculate its branch voltage using its
c     branch resistance Rba and inductance Xba.
      If(n1 .EQ. 4) then
c     Calculate Rba*Ib
        Do 200 i = 1,n3
        Rb(i) = 0.000
        Do 210 k = 1,n3
 210  Rb(i) = Rb(i) + Rba(i,k)*cb(n1,k)
 200  Continue
        Do 220 i = 1,n3
        Xb(i) = 0.000
        Do 240 k = 1,n3
 240  Xb(i) = Xb(i) + Xba(i,k)*Cd(k)
 220  Continue
        Do 250 i = 1,n3
 250  Vb(n1,i) = Rb(i) + Xb(i)
c     If it is a converter, calculate its branch voltage using its
c     branch resistance Rbb and inductance Xbb.
      elseif(n1 .EQ. 5) then
c     Calculate Rbb*Ib
        Do 201 i = 1,n3
        Rb(i) = 0.000
        Do 211 k = 1,n3
 211  Rb(i) = Rb(i) + Rbb(i,k)*cb(n1,k)
 201  Continue
        Do 221 i = 1,n3
        Xb(i) = 0.000
```

```
            Do 241 k = 1,n3
241     Xb(i) = Xb(i) + Xbb(i,k)*Cd(k)
221     Continue
            If(power .EQ. 1) then
              Do 1 i =1,n3
1             Vb(n1,i) = Rb(i) + Xb(i) - e(i)
            elseif(power .EQ. 2) then
              Do 2 i =1,n3
2             Vb(n1,i) = Rb(i) + Xb(i)
            endif
          endif
          Return
          End


c         This subroutine program is written to determine which
c         thyristors is turned on for rectification. Also, change
c         the meshes accordingly.
          Subroutine rect(Vm,Rmm,Lmm,Lii,mesh1,mesh2,mm,ipold,icond,tsi,
     +                  iback,jclose)
          Common/b1/Vb(5,16),e(16),h(4),gg(4),t
          Common/b2/Cd(16),Rb(16),Xb(16),sl,w,V1,ire, slr, sli
          Common/b9/icio(3), icirc(3), int
          Common/b10/nire, inter, ipass(2), iooo, ibbb
          Common/b19/cm(5,5), cma(5,5), cmb(5,5),Cmast(6,5),dd(5,5)
          Common/b28/dumm,ion,Vfr(6),max,trig,itrg,ioni
          Common/b29/idis,cbi(16),tsis,icom
          Real Vm(5,5), Rmm(5,5,5), Lmm(5,5,5), Lii(5,5,5)
          Integer mesh1, mesh2, mm(4), ipold(2), icond, jclose, iox
c         Determine if it is the commutation interval, if it is,
c         go to line 500.
          If(icom .EQ. 1) go to 500
c         Determine which thyristor is forward biased.
          Vfr(1) = Vb(iback,1) - Vb(iback,2)
          Vfr(2) = Vb(iback,1) - Vb(iback,3)
          Vfr(3) = Vb(iback,2) - Vb(iback,3)
          Vfr(4) = -Vfr(1)
          Vfr(5) = -Vfr(2)
          Vfr(6) = -Vfr(3)
          Do 10 i = 1,6
          If(ion .EQ. i) go to 10
          If(Vfr(i) .GE. Vfr(ion)) ion = i
10        Continue
c         Setting up the initial condition if the converter is suddenly closed
c         circuit.
          If(jclose .EQ. 1) then
            If(ion .EQ. 1) then
              icirc(1) = 1
              icirc(3) = 0
              icirc(2) = 6
            elseif(ion .EQ. 2) then
              icirc(3) = 2
              icirc(2) = 0
              icirc(1) = 1
            elseif(ion .EQ. 3) then
              icirc(2) = 3
              icirc(1) = 0
              icirc(3) = 2
            elseif(ion .EQ. 4) then
              icirc(1) = 4
              icirc(3) = 0
              icirc(2) = 3
```

```
         elseif(ion .EQ. 5) then
           icirc(3) = 5
           icirc(2) = 0
           icirc(1) = 4
         elseif(ion .EQ. 6) then
           icirc(2) = 6
           icirc(1) = 0
           icirc(3) = 5
         endif
         ioni = ion
         go to 500
       endif
       If(t .EQ. 0.000) then
         ioni = ion
         Do 1 i = 1,3
1        icio(i) = icirc(i)
         Call recmesh(mesh1,mesh2,icond)
         If(icond .EQ. 1) then
           mm(1) = mesh1
           mm(2) = mesh2
           mm(3) = ipass(1)
           mm(4) = ipass(2)
         elseif(icond .EQ. 2) then
           mm(1) = mesh1
           mm(2) = 13
           mm(3) = ipass(1)
           mm(4) = 7
         endif
           Vm(5,2) = 0.000
           cm(5,2) = 0.000
           Do 31 i = 1,2
           Do 31 j = 1,2
           If(i .EQ. 1 .AND. j .EQ. 1) go to 31
           Rmm(5,i,j) = 0.000
           Lmm(5,i,j) = 0.000
           Lii(5,i,j) = 0.000
31         Continue
           go to 500
       elseif(ion .EQ. ioni) then
           go to 1000
       elseif(ion .NE. ioni .AND. t .GT. sli .AND. itrg .EQ. 1)
     +   then
           tsi = 0.000
           itrg = 0
       endif
c      Test if the thyristor is both forward biased and trigged.
       If(tsi .GE. trig .AND. itrg .EQ. 0) then
         If(ion .EQ. 1) icirc(1) = 7
         If(ion .EQ. 2) icirc(3) = 8
         If(ion .EQ. 3) icirc(2) = 9
         If(ion .EQ. 4) icirc(1) = 10
         If(ion .EQ. 5) icirc(3) = 11
         If(ion .EQ. 6) icirc(2) = 12
         ioni = ion
         max = 1
         icom = 1
         itrg = 1
       endif
500    If(max .EQ. 2) then
         icom = 0
         max = 1
       elseif(t .EQ. 0.000 .OR. jclose .EQ. 1) then
```

```
            max = 1
          elseif(max .NE. 1) then
            go to 1000
          endif
          icond = 1
          Do 20 i = 1,2
20        ipold(i) = ipass(i)
          Do 21 i = 1,3
21        icio(i) = icirc(i)
c         If new thyristor is firing, call the subroutine recmesh to
c         determine the column in the master matrix used.
          Call recmesh(mesh1,mesh2,icond)
c         If two meshes is formed, commutation occurs.
c         Two meshes are conducting.
          If(icond .EQ. 1) then
            If(t .EQ. 0.000 .OR. jclose .EQ. 1)then
              mm(1) = mesh1
              mm(2) = mesh2
              mm(4) = ipass(2)
              jclose = 0
            else
             mm(1) = mesh2
             mm(2) = mesh1
             mm(3) = ipass(2)
             mm(4) = ipass(1)
             jclose = 0
            endif
c         If one mesh is formed, normal conduction pattern.
c         Only one mesh is conducting.
          elseif(icond .EQ. 2) then
              mm(1) = mesh1
              mm(2) = 13
              mm(3) = ipass(1)
              mm(4) = 7
              jclose = 0
              Vm(5,2) = 0.000
              cm(5,1) = cm(5,2)
              cm(5,2) = 0.000
              Do 30 i = 1,2
              Do 30 j = 1,2
              If(i .EQ. 1 .AND. j .EQ. 1) go to 30
              Rmm(5,i,j) = 0.000
              Lmm(5,i,j) = 0.000
              Lii(5,i,j) = 0.000
30            Continue
          endif
1000      Return
          End

c         This subroutine is written to calculate the discontinuous
c         of the current waveform.
          Subroutine dis(jint,tsi)
          Common/b1/Vb(5,16),e(16),h(4),gg(4),t
          Common/b2/Cd(16),Rb(16),Xb(16),sl,w,V1,ire, slr, sli
          Common/b9/icio(3), icirc(3), int
          Common/b28/dumm,ion,Vfr(6),max,trig,itrg,ioni
          Common/b29/idis,cbi(16),tsis,icom
          Common/b32/isw,iti(5),cb(5,16),Xba(3,3),Rload,Xload,srec,spwm
          Real tsi
          Integer jint
          If(jint .EQ. 2)then
            jint = 0
```

```
              go to 40
            endif
c           Determine if commutation happening.
            If(icom .NE. 1) then
              idis = 0
              sl = sli
              go to 50
            endif
c           Test if the current in the conducting diode flows below
c           holding value.
            If(icirc(1) .GE. 0) then
              If(icirc(1) .EQ. 1 .AND. cb(5,5) .LE. 0.000)
     +      idis = 5
              If(icirc(1) .EQ. 4 .AND. cb(5,8) .LE. 0.000)
     +      idis = 8
            endif
            If(icirc(2) .GE. 0) then
              If(icirc(2) .EQ. 3 .AND. cb(5,6) .LE. 0.000)
     +      idis = 6
              If(icirc(2) .EQ. 6 .AND. cb(5,9) .LE. 0.000)
     +      idis = 9
            endif
            If(icirc(3) .GE. 0) then
              If(icirc(3) .EQ. 5 .AND. cb(5,7) .LE. 0.000)
     +      idis = 7
              If(icirc(3) .EQ. 2 .AND. cb(5,10) .LE. 0.000)
     +      idis = 10
            endif
c           If current discontinuity occurs, calculate the time when
c           the current drops to zero.

            If(idis .NE. 0) then
              slr = abs(cbi(idis))*sl/(abs(cbi(idis))+abs(cb(5,idis)))
              sl = slr
              jint = 1
              go to 80
            else
              sl = sli
              go to 50
            endif
c           Setting the pole which contains the outgoing thyristor to
c           zero.
      40    If(idis .EQ. 5) then
              icirc(1) = 0
            elseif(idis .EQ. 6) then
              icirc(2) = 0
            elseif(idis .EQ. 7) then
              icirc(3) = 0
            elseif(idis .EQ. 8) then
              icirc(1) = 0
            elseif(idis .EQ. 9) then
              icirc(2) = 0
            elseif(idis .EQ. 10) then
              icirc(3) = 0
            endif
            Do 200 i = 1,3
            If(icirc(i) .GE. 7) icirc(i) = icirc(i) - 6
      200   Continue
            max = 2
      50    t = t+sl
            tsi = tsi + sl
            If(tsi .GE. tsis) then
```

```
          itrg = 0
          tsi = tsi - tsis
        endif
        sl = sli
 80     Return
        End


c       This subroutine is written to choose the correct pattern
c       of conduction for the rectifier.
        Subroutine recmesh (mesh1, mesh2, jjj)
        Common/b8/za1, za2, za3, zb1, zb2, zb3, aa, bb
        Common/b9/icio(3), icirc(3), int
        Common/b10/nire, inter, ipass(2), iooo, ibbb
c       Find out the required meshes according to the
c       values stored in icio(3). The values stored in icio(3) records
c       respectively the conduction pattern in each poles in the
c       converter.
        mesh1 = 0
        mesh2 = mesh1
        ipass(1) = 0
        ipass(2) = ipass(1)
        If(icio(1) .EQ. 1 .AND. icio(2) .EQ. 6) then
          mesh2 = mesh1
          mesh1 = 1
          ipass(2) = ipass(1)
          ipass(1) = 1
        endif
        If(icio(1) .EQ. 1 .AND. icio(3) .EQ. 2) then
          mesh2 = mesh1
          mesh1 = 2
          ipass(2) = ipass(1)
          ipass(1) = 3
        endif
        If(mesh2 .NE. 0) go to 2222
        If(icio(2) .EQ. 3 .AND. icio(1) .EQ. 4) then
          mesh2 = mesh1
          mesh1 = 3
          ipass(2) = ipass(1)
          ipass(1) = 4
        endif
        If(mesh2 .NE. 0) go to 2222
        If(icio(2) .EQ. 3 .AND. icio(3) .EQ. 2) then
          mesh2 = mesh1
          mesh1 = 4
          ipass(2) = ipass(1)
          ipass(1) = 2
        endif
        If(mesh2 .NE. 0) go to 2222
        If(icio(3) .EQ. 5 .AND. icio(1) .EQ. 4) then
          mesh2 = mesh1
          mesh1 = 5
          ipass(2) = ipass(1)
          ipass(1) = 6
        endif
        If(mesh2 .NE. 0) go to 2222
        If(icio(3) .EQ. 5 .AND. icio(2) .EQ. 6) then
          mesh2 = mesh1
          mesh1 = 6
          ipass(2) = ipass(1)
          ipass(1) = 5
        endif
```

```
      If(mesh2 .NE. 0) go to 2222
      If(icio(1) .EQ. 7) then
        mesh2 = mesh1
        mesh1 = 7
        ipass(2) = ipass(1)
        ipass(1) = 3
      elseif(icio(3) .EQ. 8) then
        mesh2 = mesh1
        mesh1 = 8
        ipass(2) = ipass(1)
        ipass(1) = 2
      elseif(icio(2) .EQ. 9) then
        mesh2 = mesh1
        mesh1 = 9
        ipass(2) = ipass(1)
        ipass(1) = 4
      elseif(icio(1) .EQ. 10) then
        mesh2 = mesh1
        mesh1 = 10
        ipass(2) = ipass(1)
        ipass(1) = 6
      elseif(icio(3) .EQ. 11) then
        mesh2 = mesh1
        mesh1 = 11
        ipass(2) = ipass(1)
        ipass(1) = 5
      elseif(icio(2) .EQ. 12) then
        mesh2 = mesh1
        mesh1 = 12
        ipass(2) = ipass(1)
        ipass(1) = 1
      endif
c     Determine how many meshes are consider. If mesh2 = 0
c     which means 1 mesh is considered.
      If(jjj .EQ. 1 .AND. mesh2 .EQ. 0) then
        jjj = 2
      endif
 2222 Return
      End

c     This subroutine is written to choose the correct pattern
c     of conduction.
      Subroutine choose(mesh1, mesh2, ino, ifree, ioy, ifn, icio,
     +                  ipass)
      Integer ifree(3), ifn(3), mesh1, mesh2, ioy, ipass(2), icio(3)
c     Find out the required meshes.
      mesh1 = 0
      mesh2 = mesh1
      ipass(1) = 0
      ipass(2) = ipass(1)
      If(icio(1) .EQ. 1 .AND. icio(2) .EQ. 3 .AND. icio(3) .EQ. 5)then
        ino = 1
        ioy = 1
        go to 2222
      elseif(icio(1) .EQ. 4 .AND. icio(2) .EQ. 6 .AND.
     +  icio(3) .EQ. 2) then
        ino = 1
        ioy = 1
        go to 2222
      else
        ino = 0
      endif
```

```
      If(icio(1) .EQ. 1 .AND. icio(2) .EQ. 6) then
        mesh2 = mesh1
        mesh1 = 1
        ipass(2) = ipass(1)
        ipass(1) = 1
      endif
      If(icio(1) .EQ. 1 .AND. icio(3) .EQ. 2) then
        mesh2 = mesh1
        mesh1 = 2
        ipass(2) = ipass(1)
        ipass(1) = 3
      endif
      If(mesh2 .NE. 0) go to 2222
      If(icio(2) .EQ. 3 .AND. icio(1) .EQ. 4) then
        mesh2 = mesh1
        mesh1 = 3
        ipass(2) = ipass(1)
        ipass(1) = 4
      endif
      If(mesh2 .NE. 0) go to 2222
      If(icio(2) .EQ. 3 .AND. icio(3) .EQ. 2) then
        mesh2 = mesh1
        mesh1 = 4
        ipass(2) = ipass(1)
        ipass(1) = 2
      endif
      If(mesh2 .NE. 0) go to 2222
      If(icio(3) .EQ. 5 .AND. icio(1) .EQ. 4) then
        mesh2 = mesh1
        mesh1 = 5
        ipass(2) = ipass(1)
        ipass(1) = 6
      endif
      If(mesh2 .NE. 0) go to 2222
      If(icio(3) .EQ. 5 .AND. icio(2) .EQ. 6) then
        mesh2 = mesh1
        mesh1 = 6
        ipass(2) = ipass(1)
        ipass(1) = 5
      endif
      If(mesh2 .NE. 0) go to 2222
 1111 Continue
c     Calculate the freewheeling diode conducting.
      If(icio(1) .EQ. 7) then
       If(icio(2) .EQ. 3) then

          mesh2 = mesh1
          mesh1 = 7
          ipass(2) = ipass(1)
          ipass(1) = 4
       endif
       If(icio(3) .EQ. 5)then
          mesh2 = mesh1
          mesh1 = 8
          ipass(2) = ipass(1)
          ipass(1) = 6
       endif
      endif
      If(mesh2 .NE. 0) go to 2222
      If(icio(3) .EQ. 8) then
       If(icio(1) .EQ. 4) then
```

```
        mesh2 = mesh1
        mesh1 = 9
        ipass(2) = ipass(1)
        ipass(1) = 6
      endif
      If(icio(2) .EQ. 6)then
        mesh2 = mesh1
        mesh1 = 10
        ipass(2) = ipass(1)
        ipass(1) = 5
      endif
    endif
    If(mesh2 .NE. 0) go to 2222
    If(icio(2) .EQ. 9) then
      If(icio(1) .EQ. 1) then
        mesh2 = mesh1
        mesh1 = 11
        ipass(2) = ipass(1)
        ipass(1) = 1
      endif
      If(icio(3) .EQ. 5)then
        mesh2 = mesh1
        mesh1 = 12
        ipass(2) = ipass(1)
        ipass(1) = 5
      endif
    endif
    If(mesh2 .NE. 0) go to 2222
    If(icio(1) .EQ. 10) then
      If(icio(3) .EQ. 2) then
        mesh2 = mesh1
        mesh1 = 13
        ipass(2) = ipass(1)
        ipass(1) = 3
      endif
      If(icio(2) .EQ. 6)then
        mesh2 = mesh1
        mesh1 = 14
        ipass(2) = ipass(1)
        ipass(1) = 1
      endif
    endif
    If(mesh2 .NE. 0) go to 2222
    If(icio(3) .EQ. 11) then
      If(icio(1) .EQ. 1) then
        mesh2 = mesh1
        mesh1 = 15
        ipass(2) = ipass(1)
        ipass(1) = 3
      endif
      If(icio(2) .EQ. 3)then
        mesh2 = mesh1
        mesh1 = 16
        ipass(2) = ipass(1)
        ipass(1) = 2
      endif
    endif
    If(mesh2 .NE. 0) go to 2222
    If(icio(2) .EQ. 12) then
      If(icio(3) .EQ. 2) then
        mesh2 = mesh1
        mesh1 = 17
```

```
        ipass(2) = ipass(1)
        ipass(1) = 2
      endif
      If(icio(1) .EQ. 4)then
        mesh2 = mesh1
        mesh1 = 18
        ipass(2) = ipass(1)
        ipass(1) = 4
      endif
    endif
    If(mesh2 .NE. 0) go to 2222
    If(icio(1) .EQ. 7) then
      If(icio(2) .EQ. 12) then
        mesh2 = mesh1
        mesh1 = 19
        ipass(2) = ipass(1)
        ipass(1) = 4
      endif
      If(icio(3) .EQ. 8) then
        mesh2 = mesh1
        mesh1 = 20
        ipass(2) = ipass(1)
        ipass(1) = 6
      endif
    endif
    If(mesh2 .NE. 0) go to 2222
    If(icio(2) .EQ. 9) then
      If(icio(1) .EQ. 10) then
        mesh2 = mesh1
        mesh1 = 21
        ipass(2) = ipass(1)
        ipass(1) = 1
      endif
      If(icio(3) .EQ. 8) then
        mesh2 = mesh1
        mesh1 = 22
        ipass(2) = ipass(1)
        ipass(1) = 5
      endif
    endif
    If(mesh2 .NE. 0) go to 2222
    If(icio(3) .EQ. 11) then
      If(icio(1) .EQ. 10) then
        mesh2 = mesh1
        mesh1 = 23
        ipass(2) = ipass(1)
        ipass(1) = 3
      endif
      If(icio(2) .EQ. 12) then
        mesh2 = mesh1
        mesh1 = 24
        ipass(2) = ipass(1)
        ipass(1) = 2
      endif
    endif
2222 Do 1 i = 1,3
    If(icio(i) .GT. 6) then
      ifree(i) = 1
      ifn(i) = icio(i)
    else
      ifree(i) = 0
      ifn(i) = 0
```

```
      endif
 1    Continue
      Return
      End

      Subroutine rearr(ipold, ipass, mm, mesh1, mesh2)
      Integer ipass(2), ipold(2), mm(4)
      If(ipass(1) .EQ. ipold(2) .OR. ipass(2) .EQ. ipold(1)) then
        mm(1) = mesh2
        mm(2) = mesh1
        dummy = ipass(1)
        ipass(1) = ipass(2)
        ipass(2) = dummy
        mm(3) = ipass(1)
        mm(4) = ipass(2)
      else
        mm(1) = mesh1
        mm(2) = mesh2
        mm(3) = ipass(1)
        mm(4) = ipass(2)
      endif
      Return
      End

      Subroutine check(ifree, ifn, cb, icirc, cm, ichh, mm, iq)
      Integer ifree(3), ifn(3), icirc(3), ichh, mm(4)
      Real cb(5,16), cm(5,2)
      kk = 0
      Do 1 i = 1,3
      If(ifree(i) .EQ. 1) then
        kk = 1
        ix = 4 + ifn(i)
      If(cb(5,ix) .LE. 0.05 .AND. icirc(i) .GT. 6) then
        icirc(i) = icirc(i) - 6
        ifree(i) = 0
        ifn(i) = 0
        kk = 0
        ichh = 1
      endif
      endif
 1    Continue
      If(kk .EQ. 1) then
      If(cm(5,1) .GT. 0.05 .AND. cm(5,2) .GT. 0.05) then
        go to 100
      else
        Do 2 i = 1,2
        If(cm(5,i) .LE. 0.05) then
          iq = i
          mm(i) = 25
          ichh = 2
        endif
 2      Continue
      endif
      endif
 100  Return
      End
```