REAL-TIME PARALLEL PROCESSING FOR POWER APPLICATIONS

Yan Guo

B.Eng. and M.Eng.
Tsinghua University, Beijing, China

A Thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfilment of the requirements for
the degree of Doctor of Philosophy

Department of Electrical Engineering
McGill University
Montréal, Québec, Canada

November 1993

ISBN 0-315-94627-X

ABSTRACT

The thesis describes the design, implementation and applications of two multiprocessor systems. A Multiprocessor Controller and an Extensible Modular Multiprocessor System have been built and have been used to solve problems of real-time digital control and real-time digital simulation in the power electronics and power systems areas.

The Multiprocessor Controller, built around three fixed-point digital signal processors(DSPs), has been used in real-time parallel processing to control a voltage-source type pulse-width-modulated power converter. In a pole-placement control strategy with a state observer, the converter has been stabilized with its dc link capacitance reduced by a factor of as much as 120, thus making the converter a potentially practical device for High Voltage direct current transmission.

The Extensible Modular Multiprocessor System consists of modules which can be easily added in a mesh architecture to provide more computing power. Each module consists of one or two autonomous processing units (PUs) and the supporting control/interface circuits. A prototype of three modules (five floating-point DSPs) has been built and used in parallel processing to simulate a small power system with two turbo-generators operating in real time as a Transient Network Analyzer(TNA).

The power system equations are partitioned by using a new method in which the system is modeled as an interconnection of functional blocks. The power system is simulated by an interconnection of DSP modules, with one module simulating one block. The results of elaborate tests demonstrate the correctness of :(a) the new partitioning method, and (b) the design and operation of the Extensible Modular Multiprocessor System. The results further show that the new partitioning method together with the Extensible Modular Multiprocessor System form a promising approach to digitize the Transient Network Analyzer.

RÉSUMÉ

Cette thèse décrit les étapes de la conception, de l'implantation et de la mise en service de deux systèmes multiprocesseurs. Un contrôleur multiprocesseur et un système modulaire extensible ont été conçus, et ont servis pour résoudre des problèmes de commande numérique et de simulation numérique en temps réel, dans les domaines de l'électronique de puissance et des réseaux électriques.

Le contrôleur multiprocesseur à 3 processeurs numériques (DSP) à points fixes a permi, par un traitement parallèle en temps réel, de contrôler un convertisseur de puissance de type source-de-tension à modulation par largeur d'implusions. L'implantation d'un observateur à placement de pôles a permis de maintenir la stabilité du convertisseur malgré une diminution de la capacitance de jonction CC allant jusqu'à un facteur de 120. Cela semble indiquer que cette application est prometteuse et pratique pour le transport du courant continu à haute tension.

Le système modulaire extensible est constitué de modules façilement assemblés en grille, constituant ainsi un puissant ordinateur parallèle. Chaque module est constitué d'un ou de deux processeurs autonomes et de circuits connexes, dont les circuits de commande et d'interfaces. Un prototype composé de 3 modules (avec 5 DSP à points flottants) a été construit, puis a servis dans l'implantation parallèle d'un simulateur de transitoires de réseaux (TNA) numérique en temps réel pour étudier le cas particulier d'un petit réseau comportant deux groupes turbo-alternateurs.

Une nouvelle méthodologie est utilisée pour partitionner les équations du modèle mathématique, décomposant le réseau électrique en un ensemble de blocs fonctionnels. Chaque bloc du réseau est simulé par un module, et le réseau au complet est constitué d'une interconnexion de modules. Les resultats obtenus à partir des nombreux tests démontrent la validité de: a) cette nouvelle méthode de partitionnement de réseau, et b) la conception et le fonctionnement du système modulaire extensible. Nous pouvons désormais affirmer avec assurance que la nouvelle méthodologie de partitionnement et le système modulaire extensible forment ensemble une approche prometteuse pour numériser les simulateurs de transitoires de réseaux.

ACKNOWLEDGEMENTS

I wish to express my most sincere gratitude to Professor B. T. Ooi and Professor H. C. Lee, my thesis supervisors, for their advice, guidance, encouragement, and help in the preparation of the thesis. I would also like to thank Professor Ooi for arranging financial support for me throughout my graduate study in McGill from a Research Grant of the NSERC of Canada.

I am deeply indebted to Professors F. D. Galiana, L. Daneshmend, A. Malowany and Z. Wolanski for their valuable advice and generous help in the research work.

Special thanks to the friends and colleagues, M. Huneault, M. Tremblay, L. Loud, K. Mehrdad, H. Javidi, J. Liu, L. Fan, B.M.M. Mwinyiwiwa, N. Lujara, S. Dimassi, B. Picard and visiting scholars S. Z. Dai, Z. C. Zhang and R. Asgharian. M. Huneault and D. Leong-Kon have made an expert French translation of the Abstract. Many thanks to J. Mui and his colleagues in the workshop and the department secretaries for their kindness in helping me.

Finally, I would like to thank my husband, Xiao, for his unlimited understanding, support and inspiration, and many sacrifices he made on my behalf, which made this thesis possible.

CONTENTS

LIST OF FIGURES

CHAPTER 4

CHAPTER 5

CHAPTER 6

CHAPTER 7

LIST OF TABLES

ACRONYMS AND ABBREVIATIONS

AB	Address Bus
ACK	Acknowledgement
A/D	Analogue to Digital converter
CB	Control Bus
CPU	Central Processing Unit
D/A	Digital to Analogue converter
DB	Data Bus
DSP	Digital Signal Processor
EMTP	ElectroMagnetic Transients Program
EPROM	Erasable Program Read Only Memory
GM	Global Memory
GTO	Gate Turn Off
HVdc	High Voltage Direct Current
INT	Interrupt
MPP	Massively Parallel Processors
OP-AMP	Operational Amplifier
PC	Personal Computer
P-I	Proportional-Integral
PSS	Power System Stabilizer
PU	Processor Unit
PWM	Pulse Width Modulation
RAM	Random Access Memory
Req.	Request
SSR	Subsynchronous Resonance Phenomena
SVC	Static VAR Compensator
TNA	Transient Network Analyzer
UART	Universal Asynchronous Receiver/Transmitter
VCO	Voltage Controlled Oscillator

LIST OF SYMBOLS

PART I (CHAPTERS 3 AND 4)

A, G, $\hat{G}$	transition matrices
B, H, $\hat{H}$	input matrices
C	output matrix
C_{dc}	dc link capacitance
d	disturbance
$E. F$	disturbance matrices
i_a , i_b , i_c	line currents in a-b-c frame
i_d , i_q	line currents in d-q frame
I_{dc}	dc link current
I_{out}	output current of dc link
K	number of carrier triangles per modulating signal cycle
K_{mod}	modulation index
$\hat{K}$	state feedback gain of entire control system
K_e	state feedback gain of observer
K_p	proportional feedback gain of angle
K_i	integral gain
K_f	proportional feedback gain of frequency
K_1	gain of state integrator
K_2	gain of state feedback
L, X_t	inductance and inductive reactance of transmission line
N	number of points in discrete representation of sinusoidal waveform
N_δ	pointer to sinusoidal waveform corresponding angle δ
N_s	pointer to sinusoidal waveform synchronized with ac source
P	real power of receiving-end
Q	reactive power of receiving-end

R, R_t	resistance of transmission line
$r(k)$	reference input vector
S_a, S_b, S_c	trigger signals of valves
t	real time
T_s	sampling period
u	input vector
$u_e(k)$	error input vector
$v(k)$	state vector of integrator
v_s, v_{sa}, v_{sb}, v_{sc}	sending-end voltage of a transmission line
V_s, V_{ac}	rms value of sending-end voltage of a transmission line
v_r, v_{ra}, v_{rb}, v_{rc}	receiving-end voltage of a transmission line
V_r	rms value of receiving-end voltage of a transmission line
V_{dc}	dc link voltage
$V_{dc\ ref}$	reference setting of dc link voltage
V_m	amplitude of modulating sinusoidal waveform
v_m	modulating sinusoidal waveform
v_t	carrier triangular waveform
$w(k)$	feedback input of entire system
$x(n)$	sinusoidal waveform template
$\tilde{x}(k)$, $\tilde{\eta}(k)$	estimated state vectors
$x_e(k)$	error state vector
x	state vector
y	output vector
Z	reactance of transmission line
Δt_a, Δt_b, Δt_c	time loaded to *a, b, c*-phase timers
ΔI_d, ΔI_q, ΔV_{dc}, $\Delta\delta$, $\Delta\omega_r$	small perturbation variables
δ	angle between voltage phasors at sending-end and receiving-end
δ_o, V_{dco}, I_{do}, I_{qo}, I_{outo}, T_{so}	equilibrium operating point of system variables
ϵ	error between voltage and reference setting of dc link
θ	angle between source voltage and current

$\xi(k)$	state of entire system
$\varphi_c(.)$, $\varphi_o(.)$	polynomials of control system and observer
ω_s	angular frequency of ac source or setting
ω_r, f_r	angular frequency and frequency of receiving-end
ω_o	angular frequency

PART II (CHAPTERS 6 AND 7)

a_{kj}, b_{kj}, d_i, q_i	constant coefficients from circuit parameters
e_t, e_d, e_q, e_f	voltage sources of stator and field windings
e_f	voltage of field exciter
e_3, e_r, e_{fd}, K_f, τ_f, τ_e, K_e, K_a, τ_a	states and parameters of excitation system
e_i	electromotive force of branch i
g_i, k_g, τ_{gi}	states and parameters of governor system
h	step size of numerical integration
H	rotor inertia constant
H_g, H_t, D_g, D_t, D_{gt}	moment of inertias and damping factors of generator and turbine
i_d, i_q, i_{md}, i_{mq}	d, q current of generator and transformer
i_i	current of branch i
i_{id}, i_{iq}, i_{ird}, i_{irq}, i_t, i_{td}, i_{tq}	currents in power system
i_{41d}, i_{41q}	transformed current from frame $d2$-$q2$ to $d1$-$q1$
K_{gt}	torsional stiffness constant of shaft
k_d, k_q, k_{fd}, k_{kd}, k_{kq}	coefficients from generator parameters
L_i, R_i, M_i, L_T, R_T, C_T	circuit parameters
l_d, l_q, l_F, l_D, l_Q, l_{MD}, l_{MQ}	inductances of generator
P_m	output power of governor system

P_a	accelerate power of system
P_e	electrical power of generator
pm_i	power demand reference
R_{r1}, X_{r1}, R_{r2}, X_{r2}, X_c, X_T, R_T, M_1, M_2	circuit parameters of simulated power system
r_a, x_l, x_{ad}, x_{aq}, r_f, x_{fd}, r_{kd}, x_{kd}, r_{kq}, x_{kq}	circuit parameters of stator, field and rotor winding
r, r_F, r_D, r_Q	resistances of generator
s_i, K_{qv}, τ_{qv}, K_{qs}, τ_q, τ_{qi}, τ'_{qi}	states and parameters of power system stabilizer
T_e	electromagnetic torque of generator
T_{gt}	mechanical torque between turbine and generator
T_{in}	turbine driving torque
t_n	time in modified Euler's method
v_{oj}, v_{dj}, v_{qj}	voltages of three phase in *o-d-q* frame
v_{dt}, v_{qt}	*d, q* voltage across transformer
v_o, v_{od}, v_{oq}	voltage of infinite bus
V_{ref}	reference voltages of generator output
V_s	output of power system stabilizer
v_{cd}, v_{cq}	voltage across load bus
v_{2cd}, v_{2cq}	transformed voltage from frame *d1-q1* to *d2-q2*
v_i	voltage of node *i*
y_n	state variables in modified Euler's method
α	difference between torque angles of two generators
$\delta(t_k)$	torque angle
λ_d, λ_q, λ_f, λ_D, λ_Q, λ_{AD}, λ_{AQ}	flux linkages of generator
λ_d, λ_q, λ_{fd}, λ_{kd}, λ_{kq}, λ_{md}, λ_{mq}	flux linkages of stator, field and rotor windings
ω_g, ω_t, δ_g, δ_t	angular velocities and angles of generator and turbine

CHAPTER ONE

INTRODUCTION

1.1 HISTORICAL PERSPECTIVE AND MOTIVATION

As soon as the microprocessors were commercially available in the early 1970s, engineers in power electronics were quick to exploit them to control variable speed motor drives [1]. Unfortunately, because of their slow computational speeds with respect to the time constants of the converters and drives, the success had been modestly limited to algorithms involving simple computation and implementation by using the "look-up table" approach. In the "look-up table" me:hod, the pre-calculated control strategy is stored in EPROMs and the control command is fetched from memory in response to the status of the feedback signals.

Over the past two decades, the computing hardware has become cheaper and faster [2]. Presently, as a new generation of fast microprocessors with the facility for parallel processing appears in the market, another computer revolution is on the verge of taking off. The clocking rates of the microprocessors for 1993 are in the 50 MHz range, but fast as they are, there will always be demands for greater computational power [3]. This can be made available by architectures which put the microprocessors in parallel operation for concurrent computation of the tasks. The research of this thesis explores how the benefits of multi-microprocessor systems can be brought to the application area of power electronics and power systems.

The challenges of research in the power electronics and power system area are related to:

(1) the high dimensionality of the power system equations.

(2) the stiffness of the equations which arises from the wide spread of the system eigenvalues. The large magnitude eigenvalues require short numerical integration step-sizes.

(3) the broad frequency bandwidth requirements of the information channels.

This thesis addresses two possible uses of the multi-microprocessor systems: real-time control and real-time simulation.

1.1.1 Real-Time Control

Among the earliest pioneers to apply microprocessors in power electronics and drives were Professors B. K. Bose [1] and S.R Bowes [4-7]. Professor B. K. Bose, in particular, had not only made original contributions but also gave surveys as to trends and directions through invited papers and editorship of collected papers published by the IEEE Press. He wrote a chapter on "Microcomputers" in his book "Power Electronics and AC Drives" [8].

Over the years, the single microprocessor has evolved in architecture from 4 bits to 32 bits, and in processing speeds. As the number of bits increases, the resolution of the digital signal improves. Likewise, the sampling frequency increases with the clock rates so that the bandwidth of the control channel broadens, thus giving promise of greater stabilizing and fast response capability.

Some control applications to which the microprocessors have been put are:

(1) Converter control, including the triggering pulses of pulse width modulation (PWM) [4, 5, 9-11].

(2) Feedback control [12-20].

(3) Sequencing of control strategies [15].

Over a time span of two decades, the opportunities of application have expanded; for example, photovoltaic power conditioning system [12], adaptive control of electrical drives [13], vector control of induction motor drives [14]. As the literature is too voluminous to list here, the interested reader is referred to the volume of collected papers edited by B. K. Bose [1], published by the IEEE Press, 1987. It is significant to note that the bulk of the application examples are taken from dc and ac motor drives. This is because the time constants of the motor drives are relatively long. As exemplified by the vector control [14], the current control part of the program is executed every *500 μs* and the speed and the flux control part, is executed every *1 ms*.

To date, the successful advance of real-time digital control has been limited to processes with long time constants. For the sophisticated modern and adaptive control methods to have any meaningful impact on converters and drives, parallel computation is needed to implement computationally demanding algorithms in real time.

1.1.2 Real-Time Simulation

The stiffness and the high dimensionality of the modelling equations have resulted in the past in using analog computers or hybrid computers as simulation tools for designing and planning. For example, the Corporate Research and Development Department of General Electric [21], Schenectady, U. S. A., the Departments of Electrical Engineering of the University of Wisconsin (Madison), and of Purdue University (Lafayette) [22] had in the past depended on analog computer real-time simulations in their research and development on power electronic drive systems.

On a larger scale for power systems, the specialized analog computers take the name of Transient Network Analyzers (TNAs). The TNAs use passive R-L-C ladder networks to simulate the transmission lines and electronic op-amp based integrators [23] to solve the equations of generators, governors and field exciters. These TNAs are used by the electric power utilities as real-time simulation tools for system studies, equipment specifications, contingency studies, trouble shooting, protection studies, etc. [24]. The real-time feature allows relays and other components to be tested with the simulations. Presently, the research centres of the following organizations have projects in digitizing the Transient Network Analyzers: Manitoba HVdc Research Centre [25-33] and the Institute of Research of Hydro-Quebec (IREQ) [34, 35] in Canada, EDF (Electricité de France), Toshiba Corporation [36] and Mitsubishi Electric Corporation of Japan [37].

The alternative of the TNA is to use numerical integration by digital computers for off-line simulations using programs such as the EMTP (ElectroMagnetic Transient Program) [38, 39]. For complex systems, this may take many hours for each run, even with supercomputers.

For engineers engaged in analog simulations, the advent of the microprocessor has been welcomed as a possible numerical integrator which can replace the op-amp based analog integrator [21]. However, this idea had to wait for the microprocessors to attain the necessary computational speeds and to have the features for real-time multi-processing. Beginning in about 1988, more and more high-speed integrated-circuit digital signal processors (DSPs) such as the TMS320C25 became available to fill this need. Given the importance and the interests in real-time digital simulation and real-time digital control described previously, it was felt at the beginning of this research that the time was right to investigate how multiple DSPs could be harnessed for the above purposes.

1.2 RESEARCH OBJECTIVES

Hitherto, the parallel-microprocessor systems originate mostly from the experts in computer engineering and computer science and their architectures are attempts to anticipate the needs of the users. The research work of this thesis begins with the needs of the application area and proceeds in the reverse direction to find the architectures which can fulfil the tasks with a high factor of utilization of the microprocessors.

The specific needs have their origins in an on-going research program in the Power Electronics Laboratory of McGill University. The research program has been evaluating the feasibility of using pulse width modulated (PWM), voltage-source type converters as the candidates for the next-generation converter stations for High Voltage direct current transmission [41, 42]. The proposed converters have encountered stability problems which cannot be resolved by analog proportional integral feedback [43]. In order to be able to implement more sophisticated modern control techniques, it is necessary to have a digital controller with sufficient frequency bandwidth.

In fulfilling the objective of building a digital controller for the specific purpose of stabilizing the converter, the broader goals discussed in the previous section have also been incorporated in the design.

The thesis research is basically hardware and experimental in orientation. Thus after the design stage, the prototypes were built, programmed, tested and finally proven in the application areas. The hardware realizations bring into focus the physical dimensions of the components and thus the compromises which have to be made between the number of processors and the associated circuits which can be placed on a single board. The boards inter-communicate through the back-planes, the serial ports and/or the parallel ports, each with different communication delays and different levels of difficulties in networking the communication cables. Thus, design decisions have to be made as to the best method for communication. This preoccupation with physical layout and dimensions has the salutary result that many architectures are easily dismissed as unrealizable, thus reducing the sub-set which needs to be investigated. In the research of this thesis, there has been time to investigate the two-dimensional mesh architecture only in two iterations: the Mark I, which is a single board containing three parallel digital signal processors (DSPs), and the Mark II, which is based on the experience of Mark I and which is intended to be an extensible system based on networking add-on boards which are built in the modular fashion.

1.2.1 Mark I - Multiprocessor Controller

The Mark I has been developed as a real-time digital controller for power electronics application in general. In particular, it has the more stringent demands of the McGill University power electronics laboratory in mind. This is because the focus of the research of the power electronics laboratory will be the application of modern control theory, nonlinear control theory and adaptive control, etc. on pulse width modulated (PWM) converters. The duties of the converters include regulation, tracking, active filtering, etc. The sampling rate of the pulse width modulation (PWM) strategy is as high as *2520 Hz* and it is desirable that the sophisticated control methods can be implemented digitally in real time. As an application of Mark I, the stability problems of the PWM converters have been investigated.

1.2.2 Mark II - Extensible Modular Multiprocessor System

One important conclusion from the experiences in applying Mark I is that the computational power of the three processors on board is adequate for implementing a pole-placement control with observer in the C language mixed with Assembly language. But for the extended multi-converter system or for other control methods, the computational power will be insufficient. Thus there is a need for an architecture by which, as many boards as are required, can be added on to augment the real-time computational power by parallel computation. Mark II is the proposed solution.

More importantly, Mark II has been designed also for real-time digital simulation. As an application, some problems of digitizing the Transient Network Analyzer (TNA) are investigated. The architecture is based on the two-dimensional mesh as distinct from the hypercube and fat-tree architectures [44].

1.3 REVIEW OF PREVIOUS WORK

The literature survey has been conducted with the objective of obtaining technical information which can contribute to the success of the research work. In this respect, the Transactions on the Power Electronics, the Transactions on Industrial Electronics and the Transactions on Industrial Application of the Institute of Electrical and Electronics Engineers (IEEE) have been most pertinent as they have kept alive the tradition of microprocessor hardware research. Most of the publications to date are based on the single-microprocessor [1]. Multi-microprocessor control has been used in several papers, [45-47] for example.

Although there is little guidance from the power electronics and drives area on parallel microprocessor architecture, there is no lack of ideas from the computer science and computer engineering area. The difficulty from the literature of computer science and engineering is in getting a true evaluation of the worth of the ideas and in obtaining guidance on how the ideas can be realized in hardware using available and affordable components.

The literature survey is organized into the following parts:

1. Multiprocessors in power electronics and drives.

2. Multiprocessor and parallel processing in computer engineering.

3. Parallel computation in power system.

4. Transient Network Analyzer.

1.3.1 Multiprocessors in Power Electronics and Drives

An increasing number of papers reached towards the multi-processing concept in the late 1980s. For example in Ref. [45], the quick response induction motor drive made use of three microprocessors. Two single-chip microprocessors i-8031 (*12 MHz*) were assigned to the 3 phase/2 phase and 2 phase/3 phase signal conversions. These signal conversions were initiated by the interrupt signal at every *1 ms*. The main program of the vector control and speed control was executed by one 16-bit microprocessor i-8086 (*5 MHz*) with a period of *2.2 ms*. The two i-8031 reduced the duties of the main processor i-8086 thus permitting it to fulfil more sophisticated control.

Other applications were less dedicated towards achieving the fast computational speed of Ref. [45]. For example, in Ref. [15], multi-microprocessors were used to execute programs which in total consisted of 50,000 lines of code. In addition to controlling a GTO power conditioning system in real time, the duties included: fault detection, sequence control (start, stop, interrupt, shutdown), maintenance management, data, acquisition, display, etc. The co-ordination of the omnibus duties does not require the hardware architecture dedicated to achieving broad bandwidth through parallelism.

Other researchers such as Professor S. R. Bowes of the University of Bristol [40, 48] have taken advantage of the INMOS transputer [49] which has its own parallel programming language OCCAM. The transputers, which were available in 1989 when the work of this thesis began, did not have the required computing power and architecture which were thought to be desirable.

In McGill University itself, under the leadership of Professor Ian Hunter of the Department of Biomedical Engineering, the researchers of the Biorobotics Laboratory [50, 51] have pioneered parallel architectures for robotic applications.

The paper, which proved to be most relevant from the literature search, is Ref. [4]. Although the paper is not on parallel processing, it describes how the 3-phase pulse width modulation (PWM) technique can be implemented by the microprocessor in conjunction with 4 timers. The 4-Timer Method has been adopted in the Multiprocessor Controller (the Mark I) described in **Chapter 3, 4.**

Because most of the literature pre-date the availability of the new generation of microprocessors with the features for real-time multi-processing, they have not been helpful. The resource which had been most helpful was the Texas Instrument User's Guides of the TMS320C25 [52] and the TMS320C30 [53]. The papers [83, 84, 109, 111, 112] arisen from the work of this thesis were the first in the field when they were presented in the IEEE Conferences of the Industry Application Society and the Industrial Electronics Society.

1.3.2 Multiprocessor and Parallel Processing in Computer Engineering

Computer engineering has been to a large extent concerned with parallel vector supercomputers and massively parallel processors (MPPs) [44]. The computational complexity of certain technical problems (in Chemistry, Aerodynamics, Magnetohydrodynamics, space structures) are measured in 10^{10} to 10^{24} operations per second. A workstation running in the range of 10^{7} to 10^{8} operations per second, would take a few hours to a few days to solve these problems [54]. Thus parallelism is sought as the means of achieving order-of-magnitudes improvement in speed. As the price range of the Supercomputers and MPPs are of the order of $1 million to $30 million or more, one easily sees that the research of this thesis is not in the same big league.

However, the ideas on the many kinds of parallel architecture and parallel processing and their optimization with respect to: (1) degree of parallelism, (2) uniformity of parallelism, (3) granularity of synchronization, and (4) communications, have been helpful. Some of the successful concepts are now in standard undergraduate/graduate textbooks such as in Ref. [55].

Although the literature on parallel software development is extensive, and Ref. [56] is a sample, once the architecture is chosen, the subject narrows down to the specifics of how to achieve maximum computational efficiency with respect to the hardware. As each processor has been designed to have its own portion of memory, the memory is loaded with the program to run the processor as an autonomous unit. The programmer strives to observe the rule of "load balancing" [56], which is to keep all the processors as busy as possible.

"Load balancing" assumes a prior knowledge that efficient ways of partitioning the problem among the processors have been found. Partitioning is non-trivial. In fact, **Chapter 6** is devoted to partitioning the power system for parallel numerical integration.

The present status reported by G. Cybenko and D. J. Kuck in their article "Revolution or Evolution " [57] is that "... software is unexplored ..." and "... the true performance story behind massive parallelism is hard to come by and not entirely flattering".

1.3.3 Parallel Computation in Power System

The electric power utility industry has used the digital computers for planning and design for the past three decades. As a back as in 1977 , there were suggestions of using parallel microcomputer structures for parallel processing of power system problems [58, 59]. However, the hardware was not available then.

After the long wait, the recent availability of parallel vector supercomputers and the massively parallel processors (MPPs) have stimulated research interests in parallel processing. The emphasis has been mainly in parallel software because it is tacitly assumed that the parallel vector supercomputer or the massively parallel processors will be affordable. One approach is to convert existing production codes written for the single-processor for solution by multiprocessors as in Ref. [60].

The research challenge is in partitioning the mathematical models of the large interconnected power system so as to make them amenable for parallel processing. The literature survey has narrowed the focus to "transient stability analysis", because it comes nearest to the application for which the architecture of the Extensible Modular Multiprocessor System of **Chapter 5** has been designed.

Some of the methods [61-66], which have been proposed, rely on mathematical insights. In the main, they are concerned with using multiprocessors to find faster ways to solve the algebraic equations through the L and U factors of the Y-matrix. The numerical integration of the swing equations of the turbo-generators constitutes only a small fraction of the research effort. The researchers, who have backgrounds in system engineering, normally adopt without questioning, the mathematical models which have been framed for the single-processor era.

The Capacitor Break-Point Partitioning method, which is proposed in **Chapter 6**, takes the viewpoint that the mathematical models are themselves dependent on the tools for the solutions of the mathematical equations. The partitioning difficulties, presently being encountered, have arisen because the capacitors which are present in the loads and the lines for voltage support and power factor correction are usually neglected so as to lighten the computational load of the single processor. In the multiprocessor area, one needs a re-examination of the mathematical models. One objective of the research of this thesis is to show that these capacitors should be restored in the mathematical models because they are nature's break-points for partitioning.

1.3.4 Digital Transient Network Analyzer

Certain electric utilities, such as Hydro-Quebec, have preferred to use the Transient Network Analyzer (TNA) in their system study [24]. The TNA originally consisted of analog electronic simulators of the turbo-generators which are interconnected by transmission lines which are made up of *pi*-sections of *L-C* passive components. The simulations are frequently performed in real time for several reasons. The large number of repeated runs which are needed to carry out the programs of tests require real-time simulations. Real-time simulations also allow the relays from the manufacturers to be tested with the TNA as components in the system network. The value of the TNA lies in the comprehensiveness of the tests so that transmission line surges and reflections are simulated together with intersystem swings and hunting.

In contrast, the digital computer approach consists of breaking the problem into load-flow studies, short-circuit studies, transient stability studies which are handled by separate programs. The digital equivalent of the TNA is Dommel's ElectroMagnetic Transient Program (EMTP) [38] which has the same comprehensiveness as the TNA, but which may take hours to simulate one operating point of a large network, even using a Cray computer.

It should be mentioned at this point, that Professor Dommel's EMTP program [38] includes the delay-line method for partitioning the power system. It is based on using the travelling wave theory of transmission lines [39]. The delay-line method has been the only partitioning method until the proposal of the Capacitor Break-Point Partitioning method (Chapter 6) described in this thesis.

Over the years, the analog content in the TNA has gradually given way to microprocessors. With the advent of parallelable and fast microprocessors, it is now possible to replace analog electronic simulator of turbo-generator units by multiprocessor equivalents [34, 68].

At this point in time, the challenge is one of finding a method of integrating the multiprocessor based turbo-generators into a power system. One approach is to follow the hybrid-route. This consists of converting the digital outputs of the turbo-generator modules by a D/A and after suitable amplification, the analog voltages are used to drive the pi-sections of *L-C* passive components which simulate the transmission lines [34, 68].

The alternate route, taken by the Manitoba HVdc Research Centre, is to digitize the entire TNA by loading the EMTP programs into their multiprocessor system [25]. The integration of the digital turbo-generator modules depends on the delay-line method of modelling the transmission lines [39]. The Manitoba HVdc Research Centre [25-33] has been the first to work on digitizing the TNA. After developing the hardware multiprocessors, they are presently working on incorporating user friendliness in the software by making programming by graphics possible [32].

In the light of the work which has been accomplished, one objective of the thesis work is to investigate alternative approaches, one of which is the Capacitor Break-Point Partitioning method (**Chapter 6**).

1.4 OUTLINE OF THESIS

Chapters 1 and **8** are the Introduction and the Conclusions of the thesis. The remainder of the thesis is divided into two parts:

Mark I - Multiprocessor Controller - **Chapters 2, 3, 4.**

Mark II - Extensible Modular Multiprocessor System - **Chapters 5, 6, 7.**

Chapters 2 and **5** describe the architectures, the implementations and the functions of the two systems and how they relate to the intended applications:

Mark I - a single board consisting of three processors intended as a controller of power electronic converter.

Mark II - an extensible system based on connecting modular boards. One intended application is as a Transient Network Analyzer.

The chapters following **Chapter 2** and **5** describe the tests and applications of the systems. The PWM converter in the applications of Mark I has been researched as a potential, next-generation converter station for a High Voltage direct current transmission (HVdc) system using forced-commutation technology [41, 42]. This research has been carried out in the Power Electronics Laboratory at McGill University for many years. One recent finding [69] is that when operated as a dc voltage regulator, the converter has limited stability unless the dc link capacitor is very large. The Mark I is intended to implement the sophisticated methods of modern control theory to extend the stability region and to reduce the capacitor size.

Chapter 3 introduces the power electronic converter and the pulse width modulation controls. The chapter also describes the implementation of proportional and integral (P-I) feedbacks. The intent is to use the relatively simple feedback control as the bases for learning parallel computing and testing the operational integrity of Mark I.

In **Chapter 4**, a digital control based on pole-placement with a state observer is designed to improve the stability of the PWM converter and reduce the dc link capacitor size. In the implementation, Mark I was programmed to match the control tasks with the real-time computational power of the three processors. It is found that the Assembly language programming yields sufficient computational speed. However, C language programming is likely to require more parallel processors. This conclusion leads to the design and the building of the Mark II of **Chapter 5**, which enables more computational power to be added by connecting more processing modules to the extensible architecture.

The application of Mark II is in the simulation of a small electric power network. Because the circuit components traditionally used in modelling the electric power network are entirely R and L elements, fundamental theoretical difficulties exist in partitioning the coupled system equations so that the different partitions can be assigned to the processors to be numerically integrated in parallel. It is recognized in this thesis that stray capacitors and power factor correction capacitors abound in the network and although they are presently neglected in the approximations, they are break-points for partitioning for parallel computation. **Chapter 6** describes the method of Capacitor Break-point Partitioning.

Chapter 7 applies the methodology of **Chapter 6** to show how the Mark II can be used as a digital Transient Network Analyzer. This chapter includes the extensive tests performed on the digital TNA.

1.5 TECHNICAL MANUALS

Because of the bulk of the thesis, the detailed circuit diagrams and programs are documented in two separate laboratory manuals [70, 71] rather than in the appendices.

CHAPTER
TWO

MULTIPROCESSOR CONTROLLER

2.1 INTRODUCTION

This chapter describes the design and the implementation of a Multiprocessor Controller for 3-phase bridge converters under pulse width modulation (PWM) mode of operation. This controller will be needed in the research programs at McGill University to evaluate modern [72], robust [73] and adaptive control [74, 75] theories for power converter applications. The immediate use was in stabilizing a PWM-HVdc station which will be described in **Chapters 3** and 4.

The controller, which is the first iteration of the multiprocessor systems, consists of three digital signal processors (DSPs). The selection of DSPs is based on the fact that they have been designed to support real-time digital signal processing and computation-intensive applications in areas such as telecommunications and digital controls.

In this chapter, **Section 2.2** briefly introduces the characteristics of the processor used in the Multiprocessor Controller. **Section 2.3** describes the hardware of the controller which includes the architecture of processor units, human interface and external equipment interface. In **Section 2.4** the system software, such as the system initialization and communication is described. A conclusion is given in **Section 2.5**. The detailed circuit diagrams and the software codes of the Multiprocessor Controller are documented in the technical manual [70].

2.2 BRIEF CHARACTERISTICS OF PROCESSOR

The prototype of the controller was built by using the TMS320C25 [52] which is the second generation of the TMS320 digital signal processor family. The TMS320C25 operates with a clock frequency up to *40 MHz* and an instruction execution time of *100 ns* for most of its instructions. In addition to the arithmetic and logic operations, the memory organization and the interrupt support of ordinary microprocessors, this 16-bit chip has some attractive features for multiprocessing such as the bus request signal for the data space of global memory and the clock synchronization control for two or more processors. The on-chip timer and serial communication ports make the processor flexible to use. This processor was one of the most advanced and economical type available in 1989, when the Multiprocessor Controller was planned.

2.3 HARDWARE DESCRIPTION

2.3.1 Overview of Multiprocessor Controller

Fig. 2-1 shows the functional block diagram of the Multiprocessor Controller. The controller receives the feedback signals of the dc link voltage and current from the bridge converter. Another important input is the voltage waveform of the ac source for synchronization. The controller exercises control

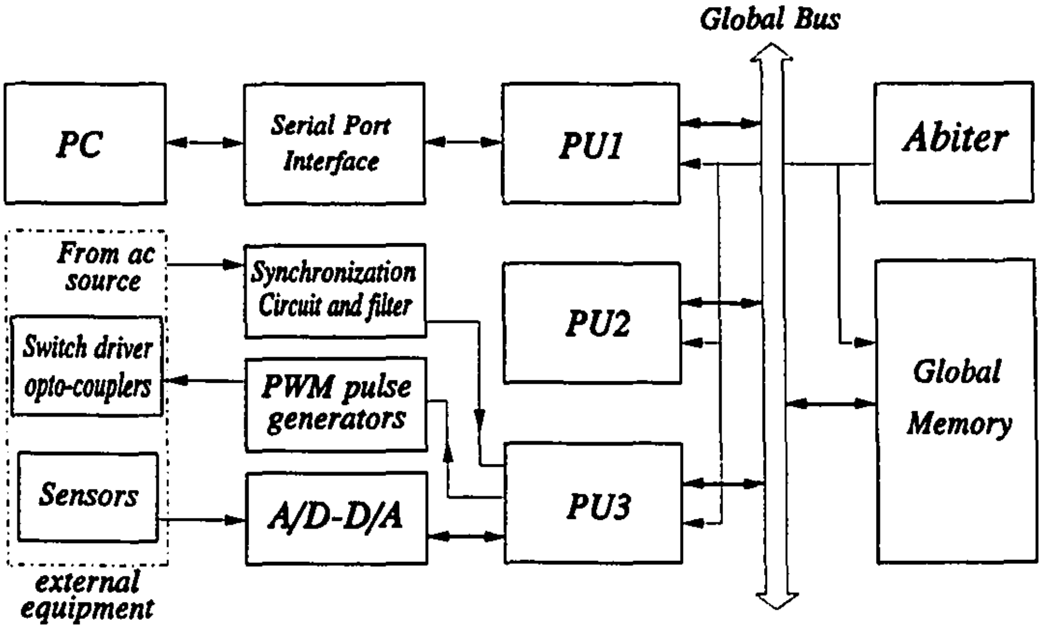


Figure 2-1 Functional block diagram of multiprocessor controller.

over the converter by using the width of the logic pulses to the trigger circuits of the valves. As the four-timer [4] method is used to implement the sinusoidal pulse width modulation (SPWM) strategy, the Multiprocessor Controller includes three electronic timers in addition to the timer on the processor chip. Each timer is set by the digital system and the time duration to the interrupt is determined by the data inputted to the timer.

In the system, there are three independent Processing Units: *PU1*, *PU2*, and *PU3*. Each of the PUs incorporates a high speed digital signal processor (DSP) and associated circuitry. Data between processors are interchanged through the global memory, the access to which is controlled by the system bus arbiter circuit. When the Multiprocessor Controller is in operation, *PU1*, *PU2* and *PU3* can compute autonomously in parallel for data processing.

PU1 provides the human interface to the system via a serial port to a personal computer (PC). It enables the downloading of programs, commands and data to any specified processor; and similarly, data from any specified processor or memory location can be acquired and displayed on the screen of the PC. The design is such that the user can change interactively the controller parameters in the multiprocessor system during real-time operation.

PU3 controls the external interface circuits of the Multiprocessor Controller, such as external timers, drivers and ac source synchronization circuit for power converters. In addition, it samples and outputs the required data of the outside world through A/D-D/A converters. *PU2* in the system just provides additional computing power in parallel processing with the other two processors. The prototype of the system operates at a clock frequency of *40 MHz.*

2.3.2 Processing Units

A typical block diagram of the PU's is shown in Fig. 2-2. Each PU contains a digital signal processor (TMS320C25), EPROM (2K words), local RAM (8K words), address decoders, a wait-state generator, and a group of switches of the global memory.

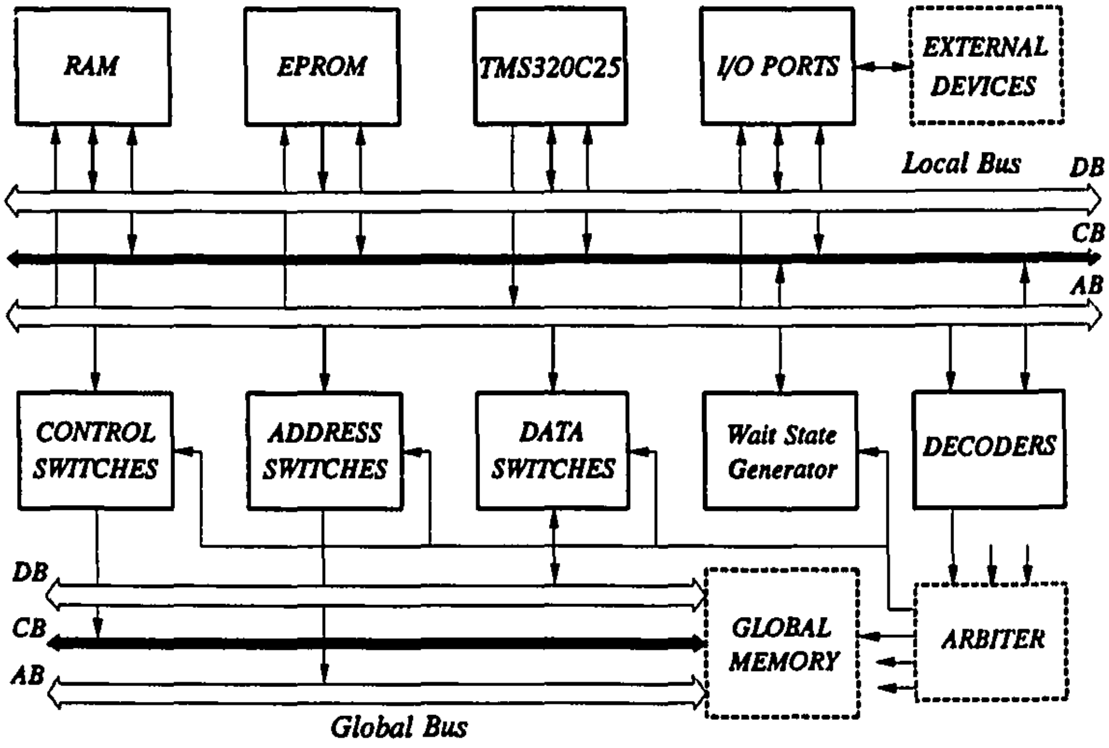


Figure 2-2 Typical block diagram of processing unit.

Access to the local RAM requires no wait state, but one wait state is needed for the EPROM and the global memory. Depending on the external devices, one or more wait states are necessary for the I/O port operations. In the 16-bit address bus, the 13 least significant bits are used to select memory locations of the RAMs and EPROMs on these component chips, while the 3 most significant bits are used for chip select. I/O select is achieved similarly. Depending on the chip or the port selected, the corresponding address decoder will signal the wait-state generator to send the ready signal to the processor with the required number of wait states. The processor gains access to the global memory through the bus switches controlled by the bus arbiter circuit.

2.3.3 Bus Arbiter Circuit

Fig. 2-3 shows a simple arbiter circuit used in the prototype system. The

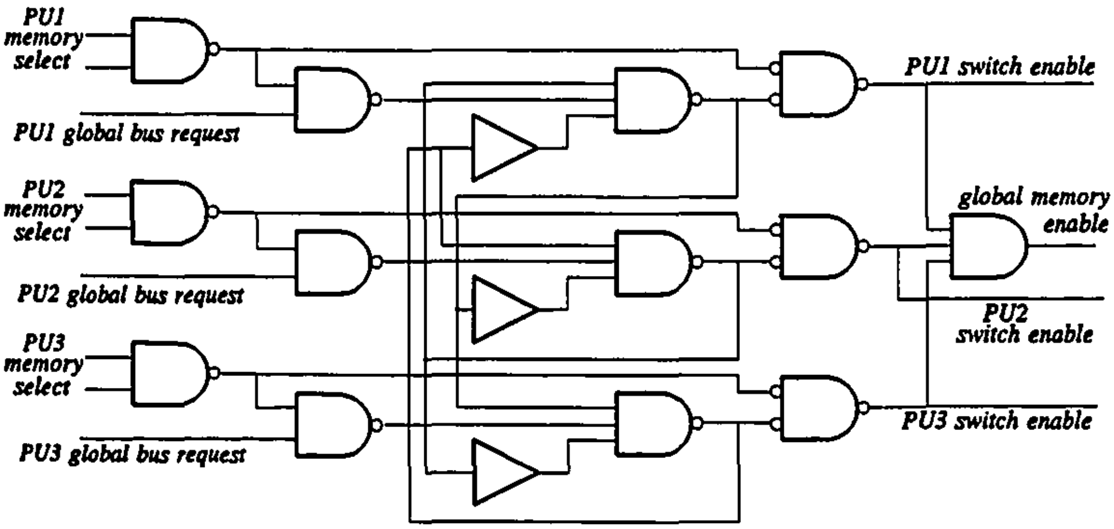


Figure 2-3 Arbiter circuit of global memory.

simple circuit has been chosen for its speed. The clock signals for the processors are derived from the system clock, but a small fixed delay is introduced between each pair so that the access to the global memory is granted on the first-come-first-served basis without conflicts.

2.3.4 Power Converter Interface

The interface between the Multiprocessor Controller and the power converter consists of power switch drivers, opto-couplers, a circuit for synchronization with the *60 Hz* ac source, analogue filters, A/D-D/A converters, and the PWM pulse generators for the timing sequences of a pulse width modulation (PWM) strategy [4]. In the following, the interface circuits on the side of the multiprocessor system are described.

2.3.4.1 A/D-D/A converters

The A/D-D/A converter interface has been implemented simply by using the TLC32040 chip via the serial port of the processor in *PU3*, as shown in Fig. 2-4. A control port has been added to control a multiplexer so that the A/D can be used to convert four different analog signals by time multiplexing.

2.3.4.2 Synchronization with ac source

The synchronization circuit is implemented as shown in Fig. 2-5. The voltage of phase A from the 3-phase ac source is taken by a transformer and

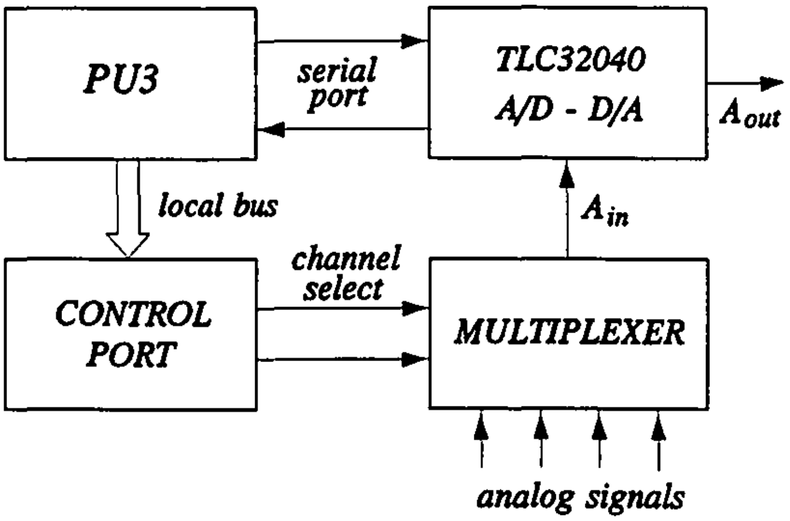


Figure 2-4 Circuit of A/D and D/A converters. A_{in} and A_{out} are analog input and output signals respectively.

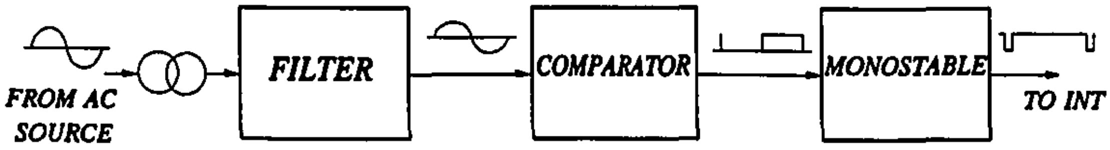


Figure 2-5 Ac source synchronization circuit.

applied to a filter which can adjust the phase of the sinusoidal waveform. A voltage comparator is used to change the sinusoidal waveform into a square waveform. The monostable circuit changes the square waveform into a sequence of pulses. The pulses are generated at the instants when the sinusoidal waveform of the ac source crosses zero from the negative to the positive level. An interrupt signal is generated whenever a new cycle of the ac voltage begins in order to enable the control program to synchronize with the ac source.

2.3.4.3 PWM pulse generator

The PWM pulse generator is specially designed for the control of the PWM converter. Fig. 2-6 shows the schematic of one of the three PWM pulse generators in the system. The 16-bit counter can be loaded with a number corresponding to the pulse width by the processor in *PU3*. A control signal from the control port of the processor starts the decrement of the content in the counter with each pulse from the clock CLK. When the content of the counter becomes zero, an active low is outputted from the counter to stop the operation of the circuit. At the same time, it triggers the D flip-flop to change its state. After this, another number is loaded in the counter and the whole process is

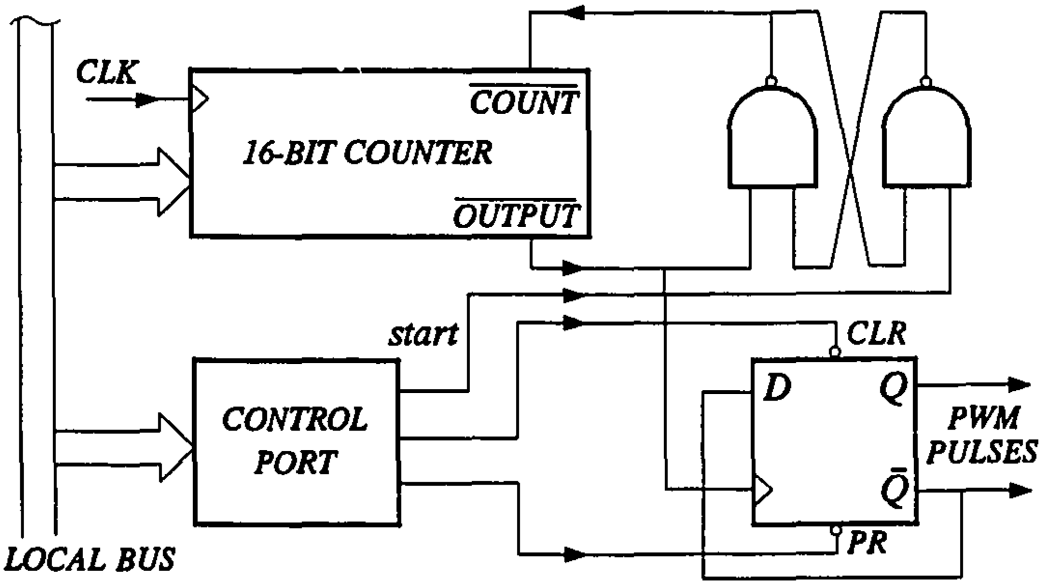


Figure 2-6 Circuit of PWM pulse generator.

repeated to generate the PWM pulses. At the beginning of the operation, the D flip-flop is initialized by the control port so that it has correct initial state (0 or 1).

The PWM timing sequences are used to control the upper and lower valves in one of the three "legs" of the PWM converter, as will be described in the next chapter.

2.3.5 Human Interface

The human interface is provided through a personal computer via the serial port of the processor in *PU1*. A hardware circuit is needed between the 8-bit asynchronous serial port of the PC and the on-chip 16-bit synchronous serial port of the processor. Although a commercially available UART could be used, the arrangement shown in Fig. 2-7 is used instead in order to conserve the external interrupt inputs since they are used extensively for interprocessor

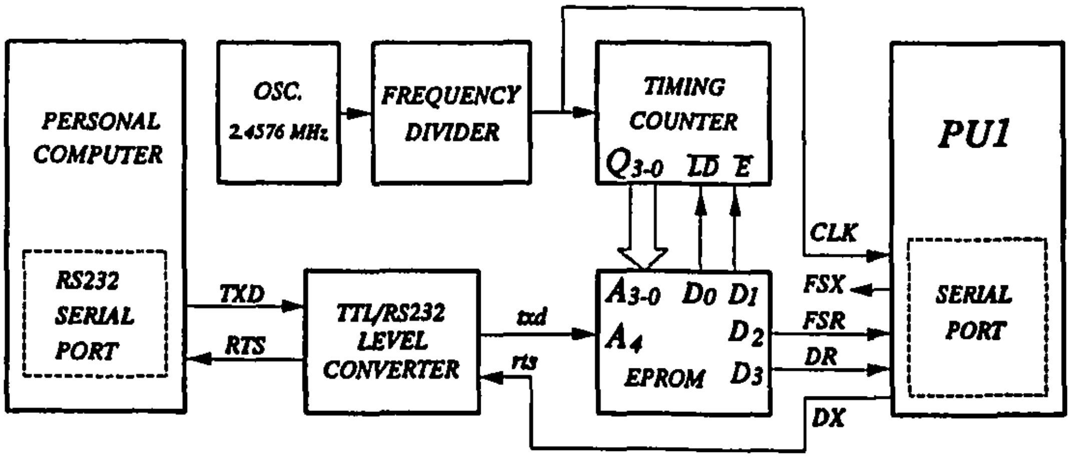


Figure 2-7 Serial port interface between PC and Processor 1.

communications.

To transmit data from the processor in *PU1* to the PC, each word is divided into two bytes and each byte is formatted by adding the start bit before loading it to the serial port. The level converter changes the voltage level to suit the requirement of the serial ports involved. The procedure to receive data from the PC is implemented by using a timing counter and an EPROM as shown in Fig. 2-7. The output of the timing counter, together with the data line txd, select the memory location $A_4A_3A_2A_1A_0$ of the EPROM whose contents $D_3D_2D_1D_0$ control the data transfer to the serial port of the processor in *PU1*: D_3 conveys the transmitted data, D_2 generates the frame synchronization, D_1 and D_0 control the counter. The principle of operation is illustrated by the example given in Table 2-

Table 2-I Example for reception of byte 01010101

clock	txd	$A_4A_3A_2A_1A_0$	in EPROM	DR	FSR	E	ID
wait	1	11111	0010	0	0	1	0
start	0	11111	0101	0	1	0	1
1	0	01110	0001	0	0	0	1
2	1	11101	1001	1	0	0	1
3	0	01100	0001	0	0	0	1
4	1	11011	1001	1	0	0	1
5	0	01010	0001	0	0	0	1
6	1	11001	1001	1	0	0	1
7	0	01000	0001	0	0	0	1
8	1	10111	1001	1	0	0	1
9	0	00110	0010	0	0	1	0

I for the reception of the byte 01010101. Before reception of the start bit, the counter is always set to 1111. Here the EPROM is working like a combination logic circuit. Its address lines work as inputs and the data lines as outputs.

The communication between the PC and other processors is implemented through the processor in *PU1* via the global memory.

2.3.6 Communication between Processors

The processing units communicate with each other by four basic methods: data interchange through the global memory; testing and transferring of control and status information by using the I/O channels; interlocking the start of parallel computation within the control interval, and signalling by using external hardwired interrupts.

2.3.6.1 Interrupter bus

There are three external interrupt pins on each processor. Two of them are used to receive interrupt signals from the other two processors. The third one is used for other purposes, such as the synchronization with the ac source described in **Chapter 3**. Some I/O channels of each processor are used to send interrupt request signals to other processors.

Interrupts are used extensively to coordinate the system operation. In particular, they are used to control program execution and to interlock *PU1*, *PU2* and *PU3* as well as to initiate the interchange of data between the processors and the global memory. In the control of the power converter, the interrupts have been used to synchronize the controller operation with the *60 Hz* waveform of the ac source, to acquire data from the A/D converter, to initiate each control time interval by using the on-chip timer, and to transmit the computed timing data to the PWM pulse generators for the pulse width modulation. In the human interface, the serial port interrupts are used to control the communication between the processors and the PC through the serial port. All these signals form an interrupt bus for the Multiprocessor Controller.

2.3.6.2 Interlock circuit

The interlock circuit makes the three processors work synchronously at the task level. When the processors are working in parallel, they need to interchange information so as to share the previous processing results. But the instant of the interchange cannot be determined well beforehand because of the interrupt service routines and the condition branches existing in programs. Depending on the status ports or the registers in the global memory, many test loops are needed, especially in the case that many processors are working together. Depending on the interrupt control, the interrupt service program will become more complex and time consuming in identifying the sources of the interrupts, especially if much data need to be interchanged. In both cases, it will take the processors a long time to communicate, thus lowering the processing speed. The interlock circuit has been designed to enhance the communication between processors. In each stage of parallel execution, a "load balancing" rule is used to segment the program so that the processors can work in parallel as much as possible. The intermediate results, which will be used by the other processors in the next stage, are put into the global memory before the processor asks for the interlock of the next stage of parallel computation. The results can be fetched by other processors after the interlock operation. In this case, the procedure of the data interchanges become simple and secure.

Fig. 2-8 shows the diagram of the interlock circuit. One bit from the

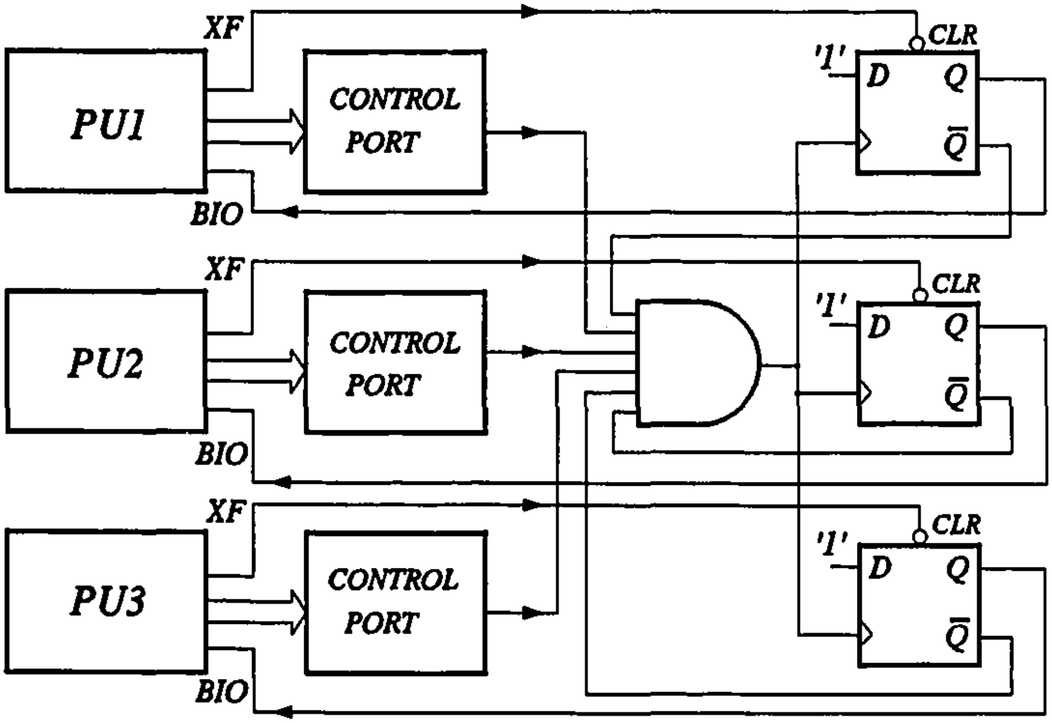


Figure 2-8 Interlock circuit of system.

control port is used to signal a request for interlock. BIOs and XFs are respectively the Branch Control Input and the External Flag Output pins which are directly controlled by the processors. Three flip-flops are used to register the interlock signals. At the beginning, they are reset by the processors. The AND gate determines if the condition for interlock has been satisfied. When the three processors are all asking for synchronization, the AND gate gives out an up-edge pulse to trigger the three flip-flops to change state. The negative outputs of the flip-flops are fedback to disable the AND gate in case that other ask signals will enter before this interlock processing is completed. When the processors get the interlock signals from the BIO pins, they reset the flip-flops by the signal from the XF pins to prepare for the next interlock operation. The program for the interlock circuit is described in the software description in the next section.

2.4 SOFTWARE DESCRIPTION

The system software has been written for the architecture of the Multiprocessor Controller hardware [70] which has been designed. It consists of monitors and service routines which will be described in the following sections.

2.4.1 Monitors and Service Routines

There are four levels of system software: (1) the Host monitor, (2) the Manager monitor, (3) the Processor monitor, and (4) the Service Routines.

Host monitor

This is at the highest level of the operating system hierarchy. It resides in the personal computer (PC) and provides user interface, allowing the user to download programs and data to the global memory, the local off-chip memories or the on- chip memories of any specified processor. Similarly, it enables the us̊er to gain access to data from any specified memory locations and have them displayed on the screen of the PC. A command from the PC can start a processor to execute a program in its program memory. In addition, it enables the user to change the parameter of the controller while the system is in operation. The Host monitor was written in a high level language (Quick Basic).

Fig. 2-9 shows the function block diagram of the Host monitor. When the

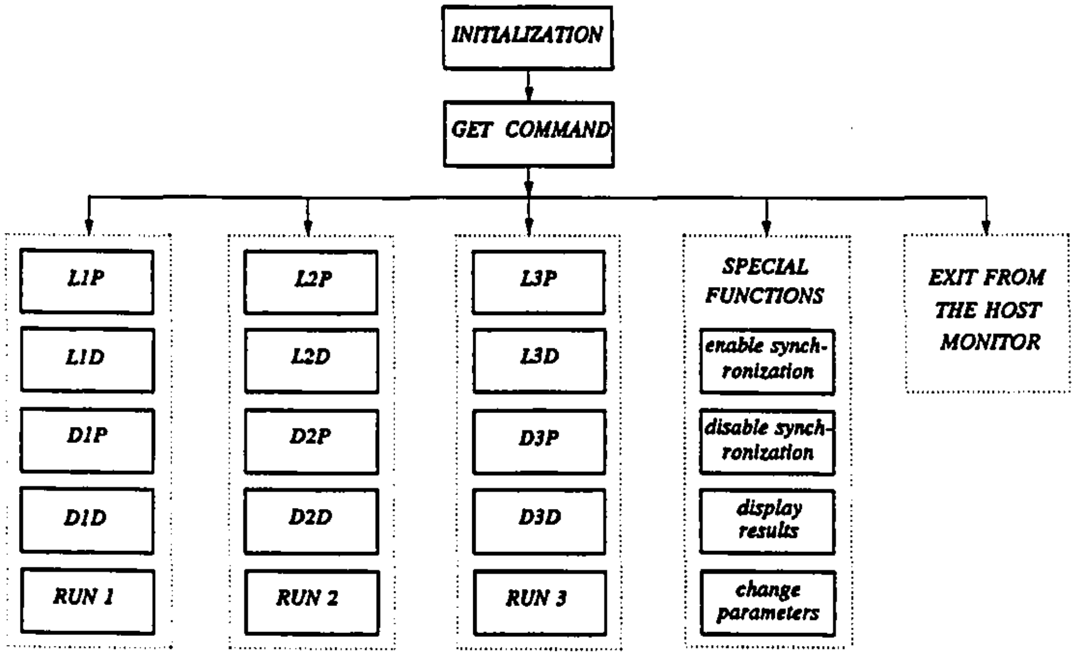


Figure 2-9 Function blocks of Host monitor.

program is initiated, the communication control parameters are initialized and the help menu is displayed on the PC screen. Operations can be selected and sent to the processors. There are three groups of command shown in the four dotted-line blocks. The left thıee blocks contain the basic functions of the monitor. They are: LkP and LkD, to load a program or a data file to the kth processor, k=1, 2, 3; DkP and DkD, to display the program code or data for the kth processor; and RUNk, to start the kth processor to run the program. The special functions are designed to display results of the executions in a specified memory, to change the parameters of the control system in memory, and to control the ac source synchronization (on or off). These functions are useful when the Multiprocessor Controller is in real-time control of the power converter. Finally, the quit function provides exit from the Host monitor.

Manager monitor

This part of the program resides in the EPROM of *PU1*. Its main function is to coordinate the information transfer between the PC and the processing units. Through the on-chip serial port of the processor in *PU1*, it communicates with the PC. In the communication waiting state, the Manager monitor is ready to receive commands from the Host monitor or from the Processor monitor. After decoding the command, which is always the first received byte, it calls the Processor monitor to implement the command. Fig. 2-10 shows the block diagram of the Manager monitor, the Processor monitors and the service routines. The serial port communication represents the information transfer between *PU1* and the PC. The manager determines the destination of the in-coming command or data.

Processor monitor and Service Routines

Each processor has its own monitor and a set of Service Routines. The Processor monitor decodes the commands from the Manager monitor and directs program execution to the appropriate service routines to implement the commands. The communication is via the global memory with the use of the interrupt. The functions of the service routines include: downloading and uploading of program and data, initialization of program execution, data

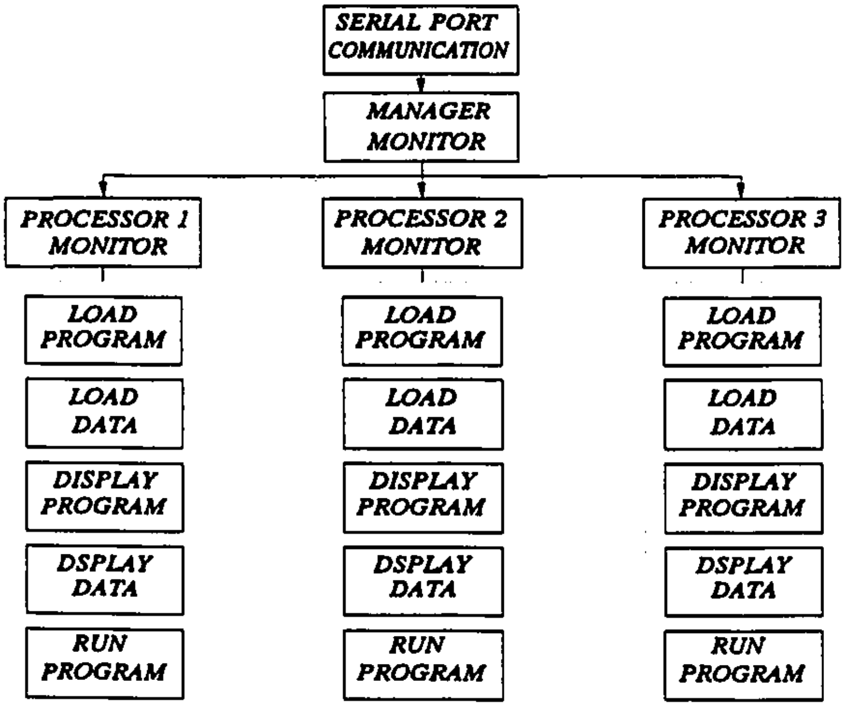


Figure 2-10 Block diagram of monitors and service routine.

acquisition from A/D converter, and other data transfer. They implement the corresponding commands of the Host monitor.

2.4.2 System Initialization

A bootstrap program resides in the EPROM of each processing unit. At power up or system reset, all control and status registers and flags on the chips of the processors are reset and the processors start to run program from address 0. Fig. 2-11 shows the block diagram of the initialization program. At the beginning, the processors initialize the internal registers such as those for interrupt control, status, and serial port communication. They load interrupt vectors in the pre-specified memory locations and clear the communication status registers, the

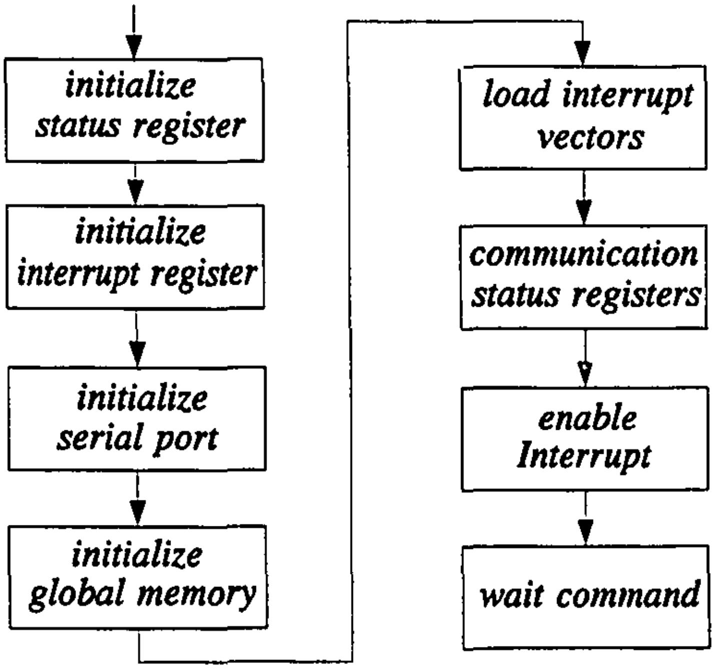


Fig. 2-11 Block diagram of processors' initialization programs.

contents of which are used to indicate the type of the data transfer. Thus the serial port of the processor in *PU1* is set to receive commands from the PC; and *PU2* and *PU3* are set to receive commands from *PU1* through the global memory, so that they can communicate indirectly with the PC.

2.4.3 Interrupt Management

In order to provide flexibility, the interrupts are managed as follows. Each hardware interrupt has an address stored in the EPROM together with branch instruction to transfer program execution to a pre-specified location in the program RAM. During system initialization, additional branch instructions are loaded in the program RAM to point to the address of the desired interrupt service routine. Thus whenever an interrupt occurs, the program execution always jumps first to the address on the EPROM, then to the address in RAM, and finally to the address of the interrupt service routine. During system operation for various purposes, the branch instruction in RAM can be changed so that other interrupt service routines can be executed.

2.4.4 Interlock Operation

The interlock operation enables the three processors to execute certain parts of the program at the same starting time. When the interlock is necessary, the processor sends an ask signal from its I/O port to the interlock circuit (Fig. 2-8) by the I/O port output instruction (OUT). Then the processor tests its BIO pin by its BIO test and branch instruction (BIOZ). If no active signal is set on the BIO pin, the processor will return to do the same test again. Otherwise, the processor will branch out of the test loop and reset the interlocked signal by using the XF control instructions (RXF and SXF). Then the processor can resume its computation.

2.5 CONCLUSION

The design and implementation of the hardware and software of the Multiprocessor Controller have been described in this chapter. The application to control the PWM converter in **Chapters 3** and **4** has proved that the controller has been built properly. It is well suited for real-time digital signal processing, particularly, where multiprocessing can be utilized to overcome the constraints of real-time control of dynamic wide-band system.

CHAPTER
THREE

PROPORTIONAL-INTEGRAL FEEDBACK CONTROL OF PWM CONVERTER

3.1 INTRODUCTION

The research on pulse width modulated (PWM) converters has a long history. This includes the use of microprocessors for their control [1, 5, 9, 10]. But because of their slow computational speed, the first generation of microprocessor-based controllers had to rely on the interpolations of pre-calculated data in look-up tables to cope with the broad-bandwidth requirements [11]. The currently available multiprocessors have many-fold increase in computational power. This advance makes it possible to calculate the control variables of the PWM converter in real time. It also brings the sophisticated digital control theories [77-79] one step nearer to real-time implementation.

This chapter describes the application of the Multiprocessor Controller of **Chapter 2** in the control of the PWM converter in real time. The existing laboratory PWM converter is a voltage-source type **[76, 80, 81]**. The experiments, which are described here, have been part of a research program to prove the conceptual feasibility of using the voltage-source topology and the PWM technique in converter station for High Voltage direct current (HVdc) transmission **[82]**. It has been demonstrated **[41, 42, 69]** that the PWM-HVdc requires two types of stations: (1) the dc voltage regulator and (2) the power dispatcher. The controls described in this chapter and the next are suitable for both types of PWM-HVdc stations. The experimental tests have been conducted for its service as the dc voltage regulator. This is because it is the weaker of the two as it is prone to instability at heavy loads **[43]**. Using analog controller, stabilization has been achieved by increasing the dc link capacitor. Besides the high cost from the capacitor size, the response time is slow. The Multiprocessor Controller has been conceived with the objectives of using more sophisticated control techniques to reduce the dc link capacitor size and to improve the response time. The analog controller essentially implements a voltage angle lock loop.

This chapter describes the implementation of a digital proportional-integral (P-I) control **[83]** to the PWM converter using the multiprocessor system of **Chapter 2.** The computational power available is excessive for the duties required. However, the *P-I* control has been deliberately chosen because the simplicity facilitates the learning experience in multiprocessing, and in the testing of the hardware and the software. The simplicity also helps in explaining in **Section** 3.4, how the three processors are organized to perform concurrent computations. An important objective is to gain experience in exchanging information through the global memory so that the three processors can share the computational burden evenly. After the learning experience of the simple *P-I* control, the more complex pole-placement control has been attempted and this is described in the next chapter.

In this chapter, **Section 3.2** introduces the voltage source-type PWM converter **[80, 81]** and the method of controlling the real power **[41, 42]**, *P*, and the reactive power, *Q* **[67, 84]**. **Section 3.3** describes the control of the phase angle, the frequency and the amplitude of the fundamental harmonic component of the ac terminal voltage of the converter. **Section 3.4** reports on the implementation of the digital *P-I* control of the PWM converter in real time. **Section 3.5** presents experimental results and their analysis. And finally, **Section 3.6** states the conclusions of the research work of this chapter.

3.2 PWM CONVERTER

3.2.1 Principles of *P* and *Q* Control

Prior to discussing the PWM-HVdc converter, a brief review is given here

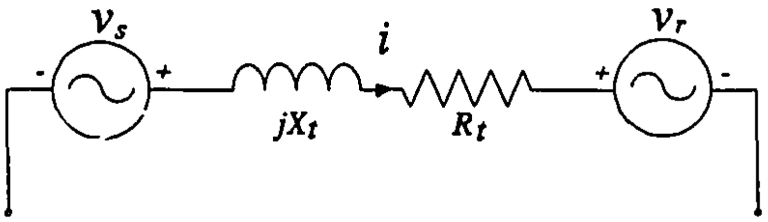


Figure 3-1 Single-line diagram of radial line between sending-end and receiving-end voltages.

of the principles of the control of the real power, P, and the reactive power, Q, in a 3-phase radial transmission line [67, 85]. This will lead quickly to the control requirements of the HVdc converter station. Fig. 3-1 shows the single line diagram in which v_s and v_r are respectively the sending-end and the receiving-end voltages of a transmission line whose resistance and inductive reactance are represented by R_t and X_t respectively.

The instantaneous voltages are:

$$v_s = \sqrt{2}\, V_s \sin \omega_s t \tag{3-1}$$

$$v_r = \sqrt{2}\, V_r \sin(\omega_r t - \delta) \tag{3-2}$$

For steady state operation, the angular frequencies are synchronized,

$$\omega_r = \omega_s = \omega_o \tag{3-3}$$

and δ is the angle between the voltage phasors.

In this study, the receiving-end is the location of the HVdc station. The receiving-end real power, P, and the reactive power, Q, are given by the formulae:

$$P = (3R_t V_s^2 - 3R_t V_s V_r \cos\delta + 3X_t V_s V_r \sin\delta)/Z^2 \tag{3-4}$$

$$Q = (3X_t V_s^2 - 3R_t V_s V_r \sin\delta - 3X_t V_s V_r \cos\delta)/Z^2 \tag{3-5}$$

where $Z^2 = R_t^2 + X_t^2$.

As $R_t << X_t$, the first two terms on the right hand side of Eq. (3-4) are negligible and P is directly proportional to $\sin\delta$. By using δ as the control variable in a negative feedback loop, it is possible to regulate P in spite of changes in Z, V_s and V_r. As V_r is controllable by the converter, one can use it to control Q through Eq. (3-5).

Eqs. (3-1) to (3-5) describe the operation under ideal sinusoidal voltage conditions. When the receiving-end consists of the ac terminals of a PWM-HVdc station, the voltage of the Eq. (3-2), and P and Q of Eqs. (3-4) and (3-5) are mathematical expressions based on the fundamental component of the Fourier series of the switched voltages of the PWM converter. The pattern of the switched voltages are determined by the sinusoidal PWM (SPWM) strategy described in **Section 3.3**.

3.2.2 Voltage Source PWM Converter

Fig. 3-2 shows the schematic of the voltage source-type PWM converter

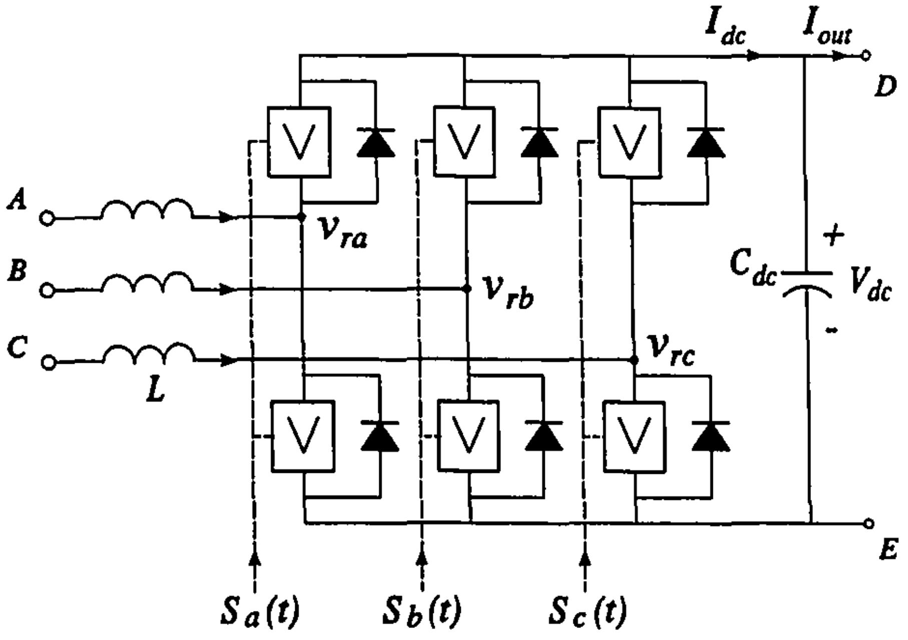


Figure 3-2 Voltage-source-type bridge converter.

whose ac terminals *A*, *B*, *C* and dc terminals *D*, *E* are connected to the 3-phase ac system and the dc system respectively. The dc link capacitor C_{dc} is charged to a voltage V_{dc} . This is the capacitor whose increase in size can stabilize the converter when used as a dc voltage regulator.

The sinusoidal PWM (SPWM) technique [86] consists of generating a modulating sinusoidal waveform $v_m(t)$ and a carrier triangular waveform $v_t(t)$ as shown in Fig. 3-3(a) and using the intersection points of the two waveforms as the instants for sending the ON-OFF gating pulses to the upper and lower valves of the branches of the 3-phase bridge. The valves are represented by the "V"s in the

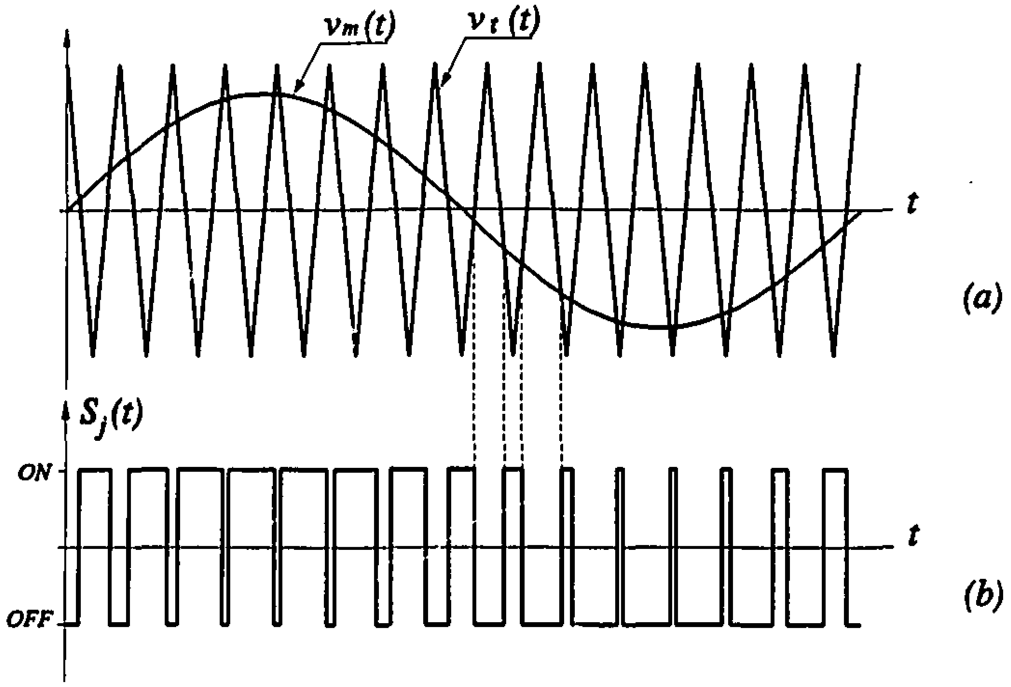


Figure 3-3 Sinusoidal PWM switching strategy. (a) Modulating signal $v_m(t)$, carrier $v_t(t)$, (b) Logic signal to base drive of jth phase.

square boxes. These pulsed signals S_j, ($j=a, b, c$) as illustrated in Fig. 3-3(b), are channelled to the valves of the 3 phases as shown in Fig. 3-2. It is assumed that the dc link voltage V_{dc} across the capacitor is kept constant by the regulation feedback loop of the converter configured as a dc voltage regulator. When the valve pairs are switched according to the sinusoidal PWM strategy, the voltage waveforms across the ac terminals of the converter are switched pulses of the dc output voltage. Fourier analysis shows that the resulting waveform contains a predominant fundamental component and high frequency harmonics at the carrier and sideband frequencies. If the high frequency harmonics can be neglected, each of the valve pairs can be represented by an equivalent sinusoidal ac voltage source v_r equal to the fundamental component as it is in Eq. (3-2).

The root mean squared (*rms*) value of the equivalent source voltage v_r is related to the dc output V_{dc} by [86]

$$V_r = K_{mod} V_{dc} \tag{3-6}$$

where

$$K_{mod} = \frac{0.5}{\sqrt{2}} \frac{V_m}{V_t} \tag{3-7}$$

V_m is the amplitude of the modulating sinusoidal waveform , and V_t is the peak value of the triangle carrier.

By varying the amplitude of the modulating sinusoidal waveform, the amplitude of the fundamental Fourier series component of the periodic sequence of the switched voltage at the ac terminal is also varied proportionally. The phase angle of the fundamental Fourier series component can be changed by varying the phase angle of the modulating sinusoidal waveform $v_m(t)$ in Fig. 3-3. The frequency of the equivalent voltage source $v_r(t)$ of Eq. (3-2) can be controlled by changing both the frequencies of the modulating waveform $v_m(t)$ and the triangle carrier waveform $v_t(t)$ of Fig. 3-3.

3.3 CONTROL STRATEGY FOR PWM CONVERTER

When the PWM converter takes the place of the 3-phase ac voltage source v_r in Fig. 3-1, the control of the real power, P, in Eq. (3-4) and the reactive power,

Q_r in Eq. (3-5) of the converter can be implemented by adjusting the modulating signal $v_m(t)$ in Fig. 3-3. V_m influences K_{mod} in Eq. (3-7) and hence the voltage *rms* value V_r. Likewise, the phase angle and the frequency of $v_m(t)$ control the phase angle δ and the frequency ω_r of the equivalent voltage source v_r in Eq. (3-2).

Fig. 3-4 shows the schematic of the voltage-source type bridge converter

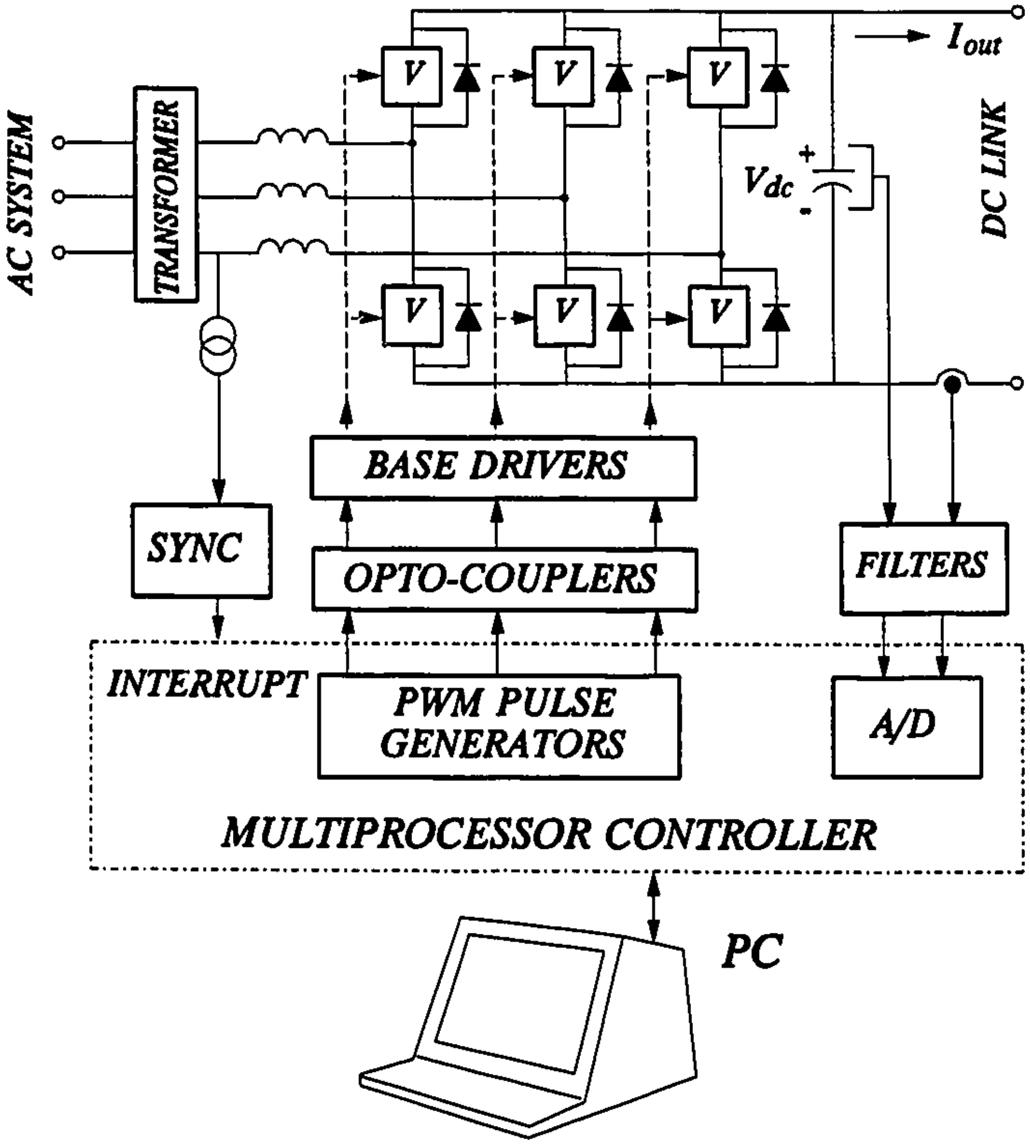


Figure 3-4 Multiprocessor controlled PWM converter.

controlled by the Multiprocessor Controller. The valves are activated by the multiprocessor system through its PWM pulse generators which produce the logic signals S_j, $j=a$, b, c. The multiprocessor system samples the values of the voltage V_{dc} and the current I_{dc} from the dc link, and receives a synchronization signal from the ac side.

3.3.1 Voltage Source Regulator [41, 69]

The converter under study will act as a voltage regulator [42, 69] to maintain the voltage V_{dc}, across the capacitor C_{dc}, close to the voltage reference setting $V_{dc\ ref}$. The voltage regulation is implemented by comparing V_{dc} with the reference voltage setting $V_{dc\ ref}$ to generate an error signal:

$$\epsilon = V_{dc\ ref} - V_{dc} \tag{3-8}$$

and using this error signal in a negative feedback control loop. In the feedback control, the real ac power P of Eq. (3-4) is converted to dc power to charge or discharge the capacitor C_{dc} to null the error.

3.3.2 Phase Angle Control [42, 69]

The voltage phase angle δ in Eq. (3-2) is made proportional to the error signal, that is:

$$\delta = K_p \epsilon \tag{3-9}$$

where K_p is a constant.

From Eq. (3-4), one sees that if any error exists, the real power P will try to decrease it. As has already been mentioned R_t is usually very small so that the real power control is dominated by the last term of Eq. (3-4).

3.3.3 Frequency Control [42, 69]

This is introduced through the angular frequency ω_r in Eq. (3-2) using the equation:

$$\omega_r = \omega_s - \Delta\omega_r \tag{3-10}$$

$$\Delta\omega_r = K_f \epsilon \tag{3-11}$$

where K_f is a constant, and ω_s is the reference angular frequency setting which is adjusted during the synchronizing procedure to be made equal to the angular frequency of ac network voltage.

As the phase angle is the integration of the frequency:

$$\delta_\omega = \int \Delta\omega_r dt \tag{3-12}$$

i.e. $$\delta_\omega = K_f \int \epsilon\, dt \tag{3-13}$$

the proportional control for the angular frequency is equivalent to the integral control for the phase angle. Thus through the frequency control, the lock loop control of the voltage phase angle is achieved.

The feedback gains K_p and K_f are the P-I controller parameters which can be adjusted to allow the transient response to be modified.

3.3.4 Amplitude Control [42, 69]

As the control of reactive power Q is thought to be not as critical to the major functions such as fast response in the dc voltage regulation, open loop control of Q is considered adequate in the research scope of this thesis. Solving Eqs. (3-4) and (3-5), one has the expression for the voltage rms value:

$$V_r = \frac{\sqrt{(\frac{3R_t V_s^2}{Z^2}-P)^2+(\frac{3X_t V_s^2}{Z^2}-Q)^2}}{(\frac{3V_s}{Z})} \tag{3-14}$$

which is required to deliver the specified Q for an assumed sending end voltage V_s, and for fixed transmission line parameters $Z=R_t+jX_t$. The real power P is calculated from transducer measurements of V_{dc} and I_{dc}.

3.4 IMPLEMENTATION OF REAL-TIME CONTROL

3.4.1 Implementation of Sinusoidal PWM (SPWM) Strategy

The sinusoidal PWM (SPWM) strategy [86] is implemented by using the four timer method described in Ref. [4]. The block "PWM Pulse Generators" in Fig. 3-4 contain the 4-timers which interrupt at time spans set by the digital numbers loaded by the multiprocessors. The timers interpret the computed outputs of the multiprocessors as the widths of the pulses illustrated in Fig. 3-3(b). One timer on chip is responsible for the half-period of the triangle carrier signal.

The other three timers are responsible for implementing the widths of the pulses of the three phases.

In the EPROM of one processor, a digital sinusoidal waveform template

$$x(n) = \sin 2\pi n/N\,, \qquad (n = 1, 2, \cdots, N) \tag{3-15}$$

is stored in N sequential addresses as a ring buffer to represent the sinusoidal waveform of the modulating waveform $v_m(t)$ in Fig. 3-3(a). In the discretization, $N=1260$. The number of carrier triangles $v_t(t)$ in one period of the sine waveform is an integer K. Fig. 3-5 is an enlarged portion of Fig. 3-3 to illustrate the digital implementation of the sinusoidal PWM strategy.

It is assumed that the address counter of the EPROMs containing sine

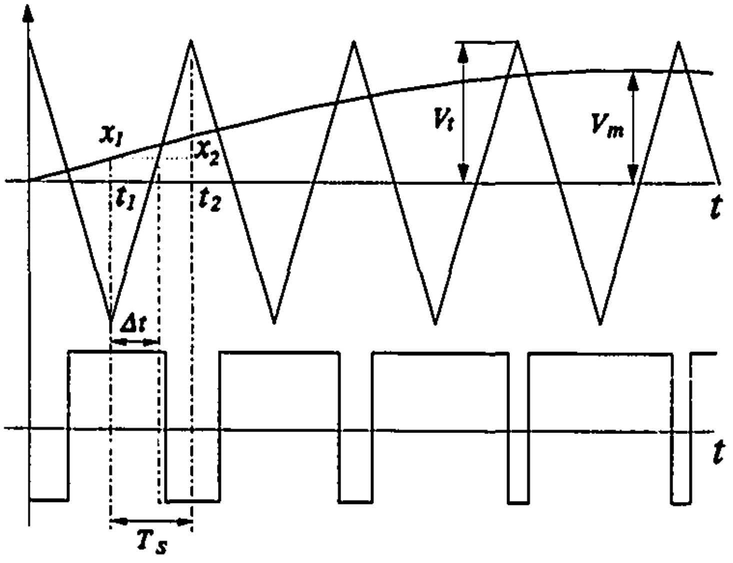


Figure 3-5 Calculation of width of switching pu

waveform template in the controller is counted cyclically in synchronism with the external ac voltage source which is sensed by the SYNC line in Fig. 3-4. The control strategy depends on using the counts n at the extreme points of the carrier triangles to sample the sinusoidal waveform template $x(n)$, $n=1, 2, ..., N$. As illustrated in Fig. 3-5, the counts n corresponds to the times $t_1, t_2, ...,$ and $x(n)$, $n=1, 2, ...$ correspond to the sampled values x_1 and $x_2, ...$. If the frequency of the ac network is f_r, the average sampling period is:

$$T_s=\frac{1}{2K f_r} \tag{3-16}$$

From Fig. 3-5, the width of Δt as determined by "uniform sampling" is based on samples at the apexes of the triangles. Suppose that the value of the sine wave template is x_1 at t_1 and x_2 at t_2. Then the Δt is estimated by using the following linear interpolation formula:

$$\Delta t=0.5T_s(1\pm x(n)\frac{V_m}{V_t}) \tag{3-17}$$

The positive sign in Eq. (3-17) is used for the ascending side of the triangular carrier and the negative sign for the descending side. The formula can be simplified by making V_t equal to 1. Then the formula becomes:

$$\Delta t=0.5T_s(1\pm x(n)V_m) \tag{3-18}$$

After Δt is computed, it is converted into its corresponding timing integer. The integer equivalent of T_s is loaded into the on-chip timer to frame the step of each calculation and control action. The integer equivalent of Δt is loaded into the PWM pulse generator to control the instants that the valves are ON or OFF, referring to the PWM pulse generator circuit in **Section 2.3.**

3.4.2 Implementation of Phase Angle Control

Prior to synchronization, the voltage of the ac system of Fig. 3-4 is sensed by the SYNC block. The instant of zero cross-over of this voltage is converted into an address N_s and stored in a register. The controlled phase angle δ computed with Eq. (3-9) is also converted into an integer number N_δ . The required angular adjustment is then implemented by using the following relation for the address index of the sine waveform template:

$$n = N_s - N_\delta \tag{3-19}$$

Thus, the template values *x(n)*, *x(n-N/3)* and *x(n-2N/3)* for the *a*, *b* and *c* phases can be accessed respectively from the memory contents of the EPROMs. The instantaneous voltages v_r of the three phases with the requisite phase angle δ are generated by the valves with pulse durations Δt controlled by Eq. (3-18).

3.4.3 Implementation of Frequency Control

The frequency control variable computed with Eqs. (3-10) and (3-11) are used to calculate the sampling time intervals of Eq. (3-16):

$$T_s = \frac{\pi}{K\,\omega_r} \tag{3-20}$$

An integer equivalent of this step length is loaded into the main timer (on-chip timer) of the controller. Whenever the main timer counts to zero, an interrupt is generated to cause the Multiprocessor Controller to iterate the cycle of operation. At the beginning of each iteration, the number N_s in the register which address $x(n)$ of the sinusoidal template is increased by $N/(2K)$. Thus the frequency control is achieved by adjusting the control step size in Eq. (3-20).

3.4.4 Implementation of Amplitude Control

Amplitude control enters through the term V_r, which is computed with Eq. (3-14). Then the magnitude of the modulating signal V_m is varied using the formula:

$$V_m = \frac{\sqrt{2}}{0.5}\frac{V_r}{V_{dc}} \tag{3-21}$$

from Eqs. (3-6) and (3-7). Whenever new timing data for PWM pulse generators are calculated, the current V_m is used in the Eq. (3-18) so that the amplitude control is taken care of.

3.4.5 Processor Co-ordination

The co-ordination of the parallel processing for the three processors (*PU1*, *PU2* and *PU3* as shown in Fig. 2-1 is through interrupts and their service routines. The interrupts have been used for the following functions:

(1) The communication between the Multiprocessor Controller and the PC is

under the control of the interrupt logic of the serial port. After the application programs have been prepared, they are loaded into the memories of each of the three processors. They are initiated to run by commands from the PC. The completion of the loading and the results from the real-time execution can be displayed on the screen of the PC. Some control parameters can be changed during the real-time execution of the program. All these functions are implemented through the serial communication port of the PC and *PU1*.

(2) T_s of Eq. (3-20), the sampling period is controlled by the on-chip timer in *PU3*. When this main timer of *PU3* counts to zero, it interrupts the processor to start a new control and computation cycle.

(3) When the processor in *PU3* starts a new cycle, it also issues interrupt signals to the other two processors through their external interrupt pins to inform them that a new iteration has begun. Then the three processors begin to run their programs in parallel.

(4) The sampling operation of the multiprocessor system is controlled by the serial port interrupt logic. The analog signals from the PWM converter of Fig. 3-4 are acquired through the A/D converter via the on-chip serial port of the processor in *PU3*.

(5) The synchronization of the control program with the ac source is controlled

by the external interrupt logic. When a new cycle of the ac source voltage begins, a signal is generated to cause the Multiprocessor Controller to align its control step.

(6) The data interchange and the processor interlock execution are controlled by the external interrupt logic. During the execution of the application program, the data from different PUs are interchanged through the global memory controlled by the external interrupt logic so that the three processors can work in co-ordination.

3.4.6 Processor Duties

Fig. 3-6 shows the function block diagram of the control programs which are executed by the 3 processors. The following duties have been assigned to the 3 processors in controlling the PWM-HVdc system to accomplish the goals of Proportional and Integral feedback described in **Section 3.3**.

Processor 1, *PU1*, interfaces with the PC and computes the voltage amplitude V_r and the magnitude of the modulating signal, V_m, using Eq. (3-21), which are needed to deliver the reactive power Q which has been specified.

Processor 2, *PU2*, is assigned the duty of feedback loop control. Essentially, it uses Eqs. (3-9), (3-10) and (3-11) to compute the voltage angle δ and the angular frequency ω_r based on the value of the dc link voltage V_{dc}, which is measured by the feedback transducer.

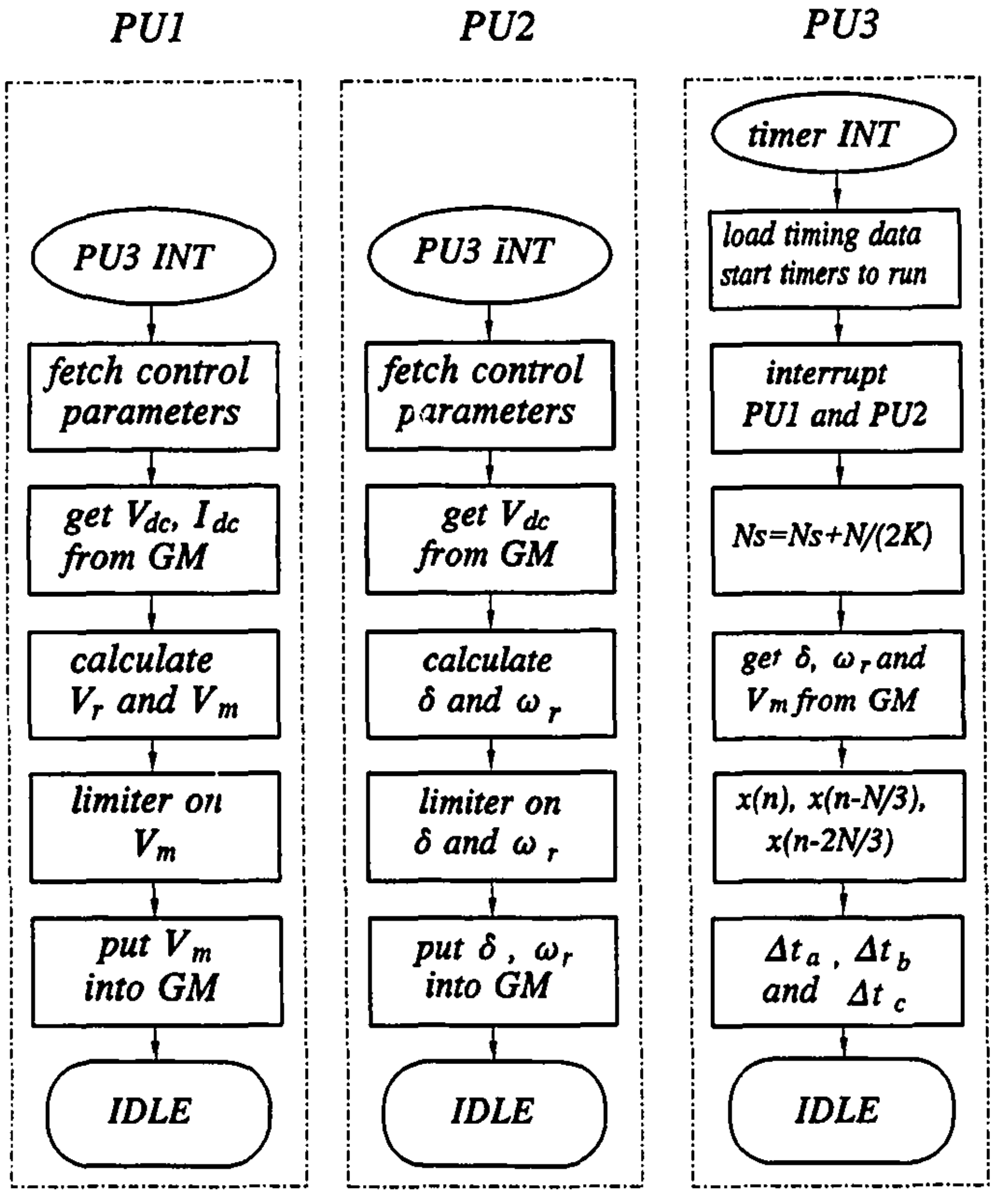

PU1
PU2
PU3
timer INT
load timing data start timers to run
PU3 INT
fetch control parameters
PU3 INT
fetch control parameters
interrupt PU1 and PU2
get V_{dc}, I_{dc} from GM
get V_{dc} from GM
$Ns=Ns+N/(2K)$
calculate V_r and V_m
calculate δ and ω_r
get δ, ω_r and V_m from GM
limiter on V_m
limiter on δ and ω_r
$x(n)$, $x(n-N/3)$, $x(n-2N/3)$
put V_m into GM
put δ, ω_r into GM
Δt_a, Δt_b and Δt_c
IDLE
IDLE
IDLE

Figure 3-6 Block diagram of control program.

Processor 3, *PU3*, manages the interface with the PWM converter. It controls the sampling of the dc link voltage V_{dc} and the current I_{dc} which are measured by transducers located as shown in Fig. 3-4. In addition, it computes the pulse width data for the 4 timers of the PWM pulse generators by using the outputs δ and ω_r from *PU2* and the output V_m from *PU1*.

3.4.7 Data Flow in System

Fig. 3-7 depicts the data communication routes in the multiprocessor

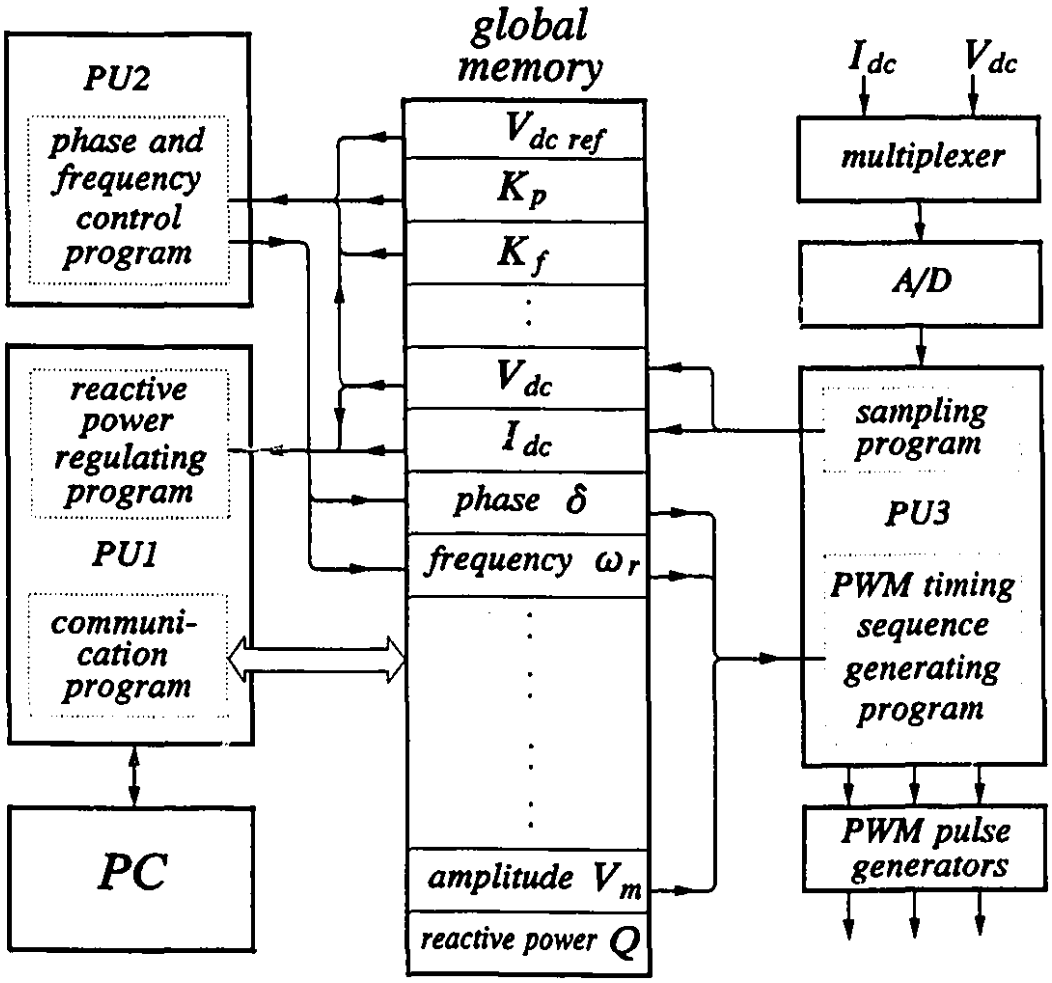


Figure 3-7 Data exchange through global memory.

system. The sampled values of V_{dc} and I_{dc} are put in the global memory by *PU3* so that they are accessible to *PU1* and *PU2*. The computed voltage phase angle δ and frequency ω_r from *PU2*, and the voltage amplitude V_m from *PU1* are passed through the global memory to *PU3* for the computation of the pulse widths.

The settings of the control system such as $V_{dc\,ref}$, Q and the control gains K_p and K_f are stored in the global memory. They can be changed through the PC at any time when the PWM-HVdc system is in operation.

The computational tasks of the processors are arranged as a pipeline [55] with two stages. The sampling of I_{dc} and V_{dc} , and the computation of δ, ω_r and V_m form one stage, while the computation of the pulse widths form the other. The tasks of the stages are performed in parallel. The pipelining principle has been implemented as an exercise and there is no real need for it since there is abundant computational power.

3.5 EXPERIMENTAL RESULTS

The laboratory model of the PWM-HVdc station of Fig. 3-4 consisted of a *1 KVA* size bipolar transistor bridge. Under test condition the parameter settings were: the dc link reference voltage $V_{dc\,ref}=110\ V$, $V_{ac}=31.1\ V$. The sinusoidal PWM modulation index before synchronization was $M=0.8$ and the number of carrier triangles per modulation signal period was $K=21$. The circuit parameters were: $X_l = 5$ *ohms*, $C_{dc} = 24000$ μF.

A special program, which "locked" the frequency and the voltage angle of the converter to those of the ac system, was used for synchronization. The ac system was sensed through the SYNC line in Fig. 3-4. After synchronization, the SYNC line was disabled.

The experimental tests have been conducted to demonstrate that different parts of the program have been implemented correctly.

3.5.1 PWM Waveforms

Figs. 3-8 (a) and (b) show the experimental waveforms of the gating signals

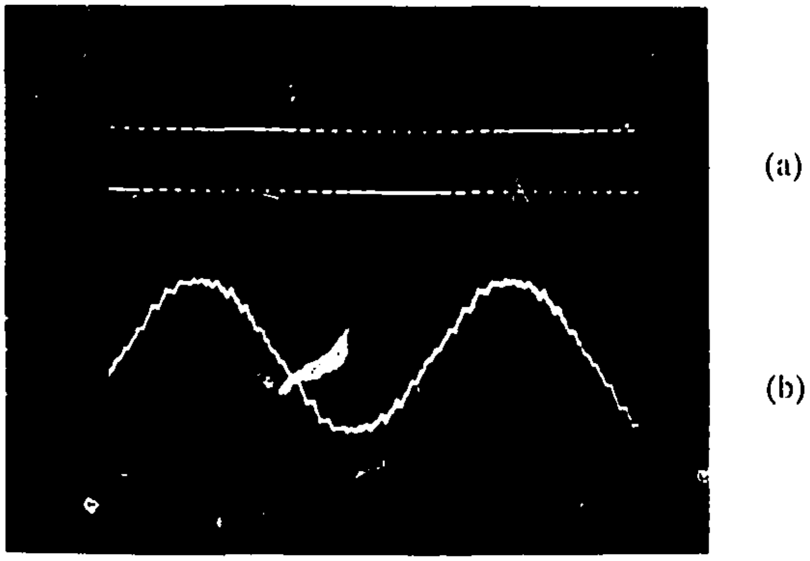


Figure 3-8 (a) Typical Sinusoidal PWM gating signal and
(b) corresponding ac current of one phase.

of the valves and the corresponding ac current of one phase under sinusoidal PWM control. The nearly sinusoidal current waveform is evidence that the sinusoidal PWM strategy using the four timer method has been successfully implemented by the processor in *PU3.*

3.5.2 Voltage Regulation

Fig. 3-9 is a plot of the measurements of the dc link voltage as the real power varies. The lines are for the different settings of the dc voltage references $V_{dc\ ref}$ *=100, 105, 110, 115* and *120 V.* The results prove that *PU3* has successfully measured V_{dc} and transferred the information to *PU2* for the calculation of ω_r and

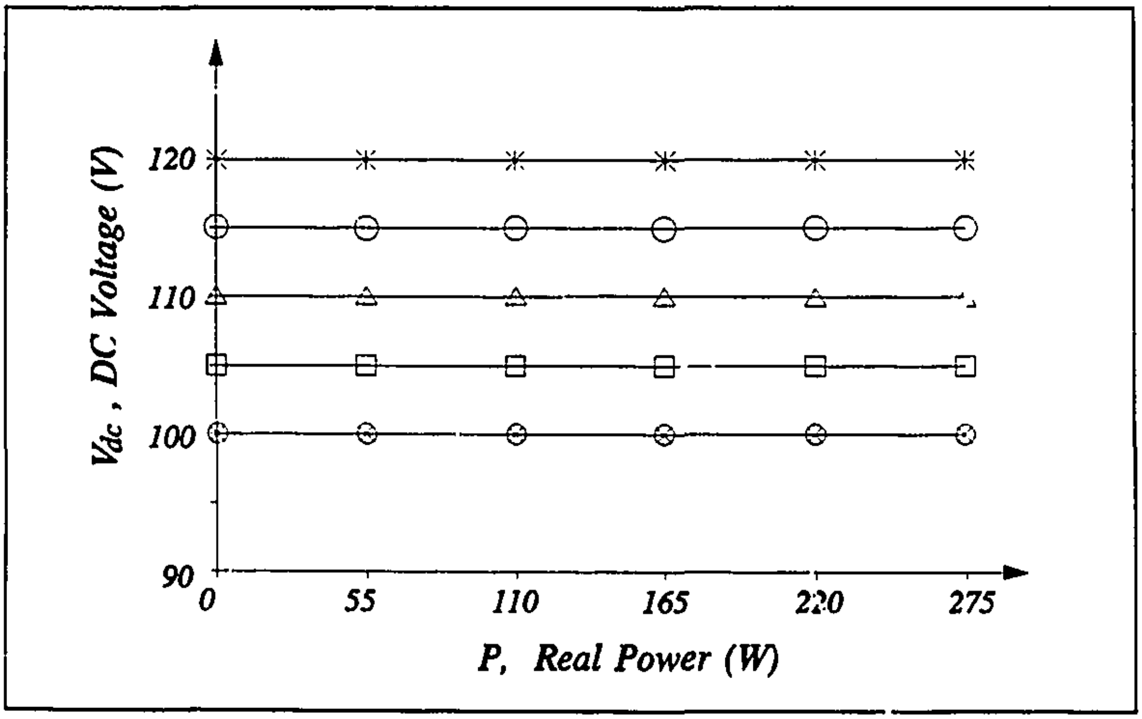


Figure 3-9 Dc voltage regulation curves : V_{dc} vs P. For voltage reference $V_{dc\ ref}$ = 100 (⊛), 105 (□), 110 (△), 115 (○), 120 (*) Volts.

6. The lack of voltage droop in Fig. 3-8 indicates that the integral feedback gain K_f acting through the frequency ω_r has been operative through Eqs. (3-10), (3-11) and (3-20).

3.5.3 Transient Response

Figs. 3-10 and 3-11 are oscillograms of transients associated with a step change in the dc load current demand. The waveforms are for: (a) the ac current, (b) the step change of the dc link current from *0.5* to *3 A*, (c) the dc link voltage. In both cases the operating conditions have been identical: $K_f=0.25$ unit and the modulation index $M=0.8$ unit. The difference is that the proportional feedback

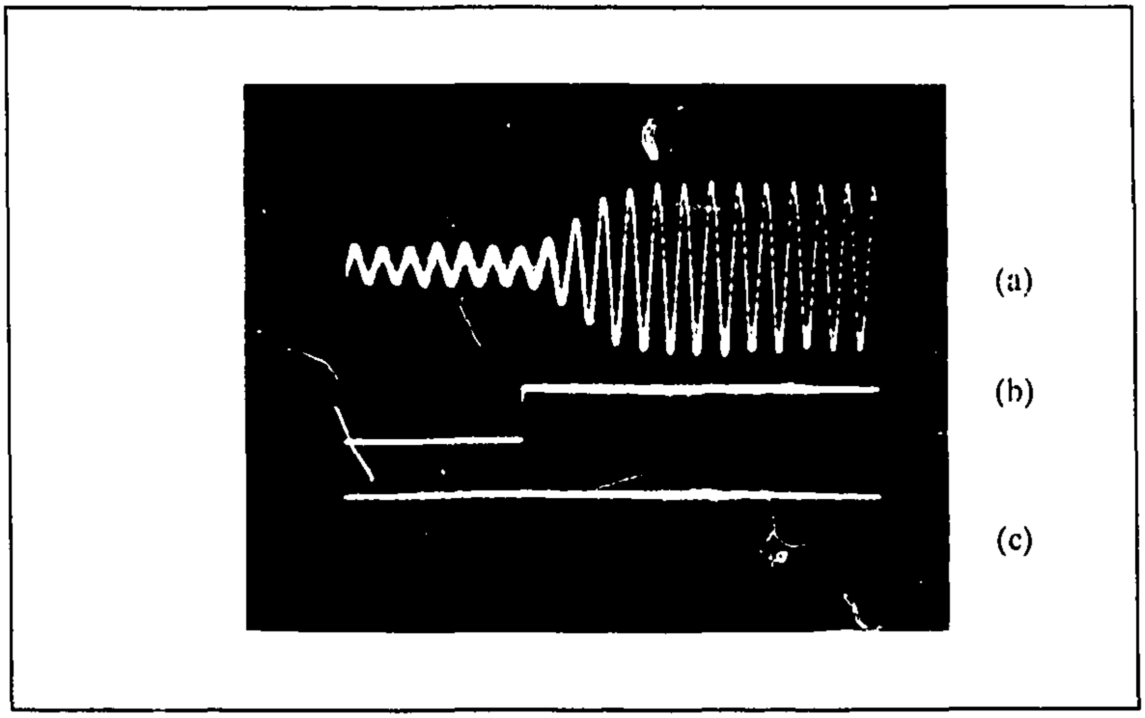


Figure 3-10 Step response with $K_f=0.25$, $K_p=0.125$. (a) ac current, (b) dc current and (c) dc voltage.

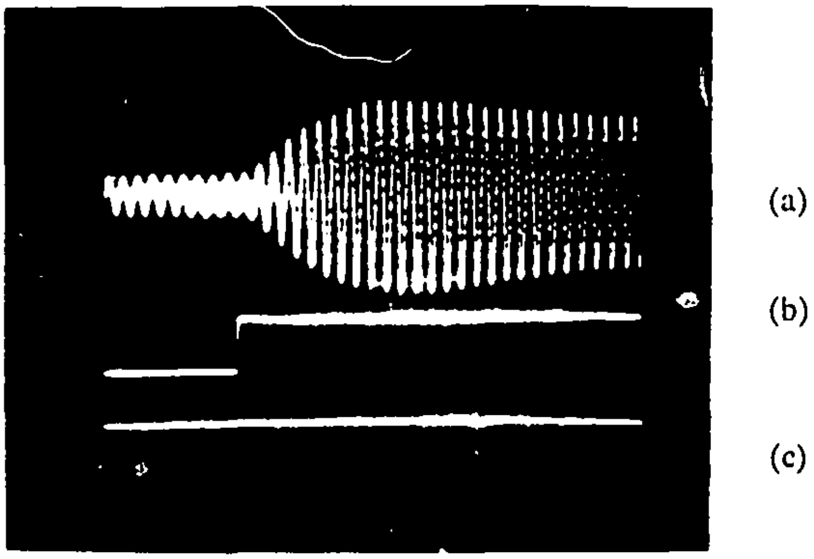


Figure 3-11 Step transient response with $K_f=0.25$, $K_p=0.0625$. (a) ac current, (b) dc current and (c) dc voltage.

gain $K_p=0.125$ in Fig. 3-9, and $K_p=0.0625$ in Fig. 3-10. As it is expected, the decrease in K_p reduces the damping of the transients as a comparison between Figs. 3-9 and 3-10 reveals. These results are evidence that Eqs. (3-9), (3-10), (3-11) and (3-19) have been implemented in real time by the processors in *PU2* and *PU3*.

3.5.4 Reactive Power Control

Although the processor in *PU1* had no difficulty in computing Eq. (3-14) within the allotted time, experimental results of constant Q delivery were poor. It is believed that this is due to the inability to model converter switching losses accurately. For this reason, an empirically based algorithm was developed in which the voltage amplitude V_m was computed based on measurements of V_{dc}, I_{dc} and the request for Q.

Fig. 3-12 shows the plots of the measurements of Q as the real power was varied, for different constant settings of Q requests. These test results demonstrate that while the voltage angle is used to control the real power, the voltage amplitude V_m can be used to control the reactive power Q. It also shows that *PU1* is implementing the empirically based algorithm correctly in real time.

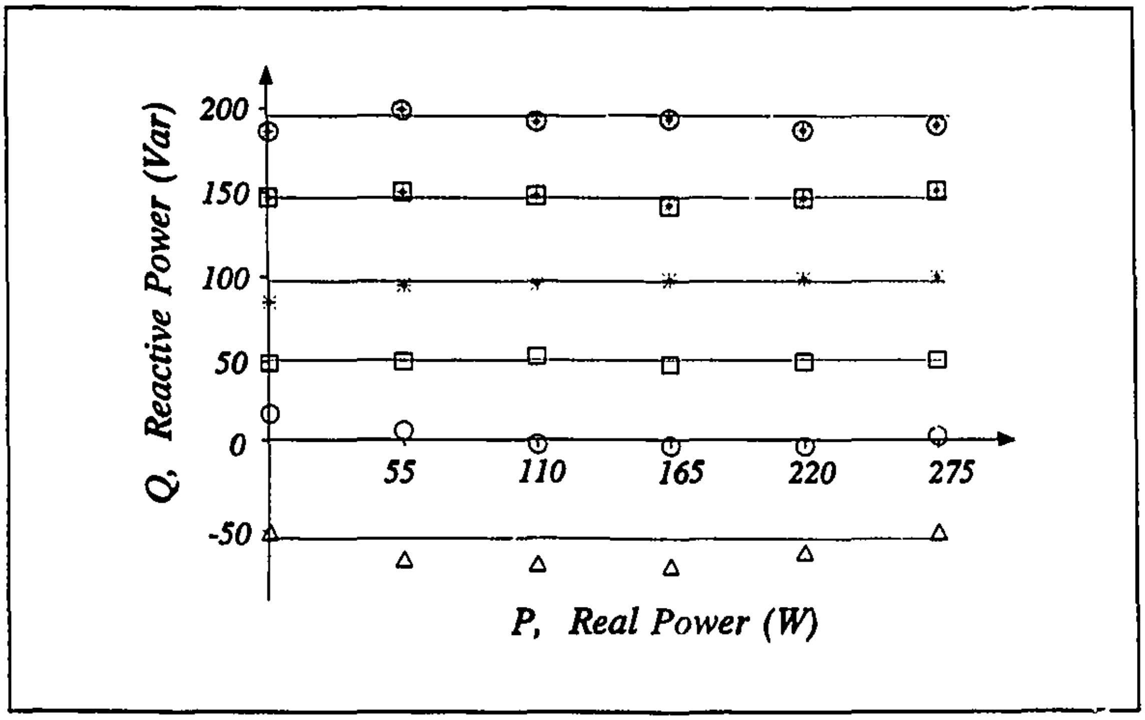


Figure 3-12 Q vs P for different constant settings of Q requests.
(Q requests: -50 (△), 0 (○), 50 (□), 100 (*), 150 (⊞), 200 (⊕) Vars)

3.6 CONCLUSIONS

At the level of real-time multiprocessing research, the experimental results have demonstrated that the proposed architecture and the method of organizing the program for concurrent computation are effective.

At the level of power electronics research, the experimental results demonstrate that the full potential of PWM control is realized when the 3 attributes of the sine wave - magnitude, frequency and phase angle are used independently as control levers. In particular, the voltage angle and the voltage magnitude may be used tc control the real power and the reactive power in the power utility environment.

CHAPTER
FOUR

POLE-PLACEMENT CONTROL OF PWM CONVERTER

4.1 INTRODUCTION

In **Chapter 3**, the computational power of the Multiprocessor Controller far exceeds the requirements of the simple proportional and integral feedbacks which have been implemented. The relative simplicity of the tasks has been deliberately chosen so that the Multiprocessor Controller can be thoroughly tested. Furthermore, it allows time to learn the characteristics of the hardware architecture of the multiprocessor and how it should be programmed for concurrent operation. This chapter is concerned with reaching the full potential of the three processors on board in the Multiprocessor Controller. This is by testing it against an application which in real time utilizes a good fraction of the computation time of the three processors operating in parallel.

In **Chapter 3**, a scheme has been proposed and implemented for the control of the voltage-source type converter for application as a dc voltage regulator in a PWM-HVdc transmission system [41, 42, 69]. It involves 3 input reference settings: the ampliıude, the phase angle, and the frequency of the fundamental Fourier component of the ac terminal voltage of the converter. When the converter operates as a dc voltage-regulated rectifier, with the phase angle and frequency control (*P-I* control), it rectifies the right amount of ac power to supply the demands of the dc system and to keep the dc link capacitor charged to the reference voltage $V_{dc\,ref}$. However, the system is stable only if [43]

$$C_{dc} > \frac{K_p I_{dc}}{\omega_s}$$

where, C_{dc} = dc link capacitance,

K_p = proportional feedback gain,

I_{dc} = output dc link current, and

ω_s = angular synchronous frequency.

In order to keep C_{dc} small so as to reduce cost, the proportional feedback gain K_p must be small. Small K_p implies poor dc voltage regulation. By using both proportional and integral feedback control, it is possible to lower K_p without serious dc voltage droop. However, the transient response is slow. The limitations of the *P-I* control can be overcome by using modern control theory - the pole-placement technique [72, 77-79].

This chapter describes a study of digital control to improve the system dynamic response by the pole-placement technique through state feedback. The pole-placement technique is now taught as part of the standard undergraduate/graduate control curriculum and there are many excellent text books on the subject [78, 79]. In addition, there are computer aided packages such as MATLAB supporting the design work. In the research work of this chapter, the control algorithm is implemented in real time by using the Multiprocessor Controller described in **Chapter 2.** Results from experiments with the laboratory model of the PWM rectifier show that the dynamic responses have been significantly improved even when the dc link capacitor is substantially reduced.

In this chapter, **Section 4.2** presents the mathematical model of the PWM converter based on small perturbation linearization. **Section 4.3** describes the design of the digital controller based on the model. This includes the design of the state feedback gain, the minimum-order observer and the control strategy for the non-linear part of the model. **Section 4.4** presents the implementation of the digital control in real time by using the multiprocessor system. **Section 4.5** shows the experimental results of the implementation and **Section 4.6** gives the conclusions of the research work of this chapter.

4.2 PWM CONVERTER - MATHEMATICAL MODEL

The first step in designing by modern control theory is to start with the mathematical model of the plant. The model of the PWM converter is well known and has been verified experimentally as reliable [76] and is presented here without proof. For completeness, this section briefly reviews the essential elements of the mathematical development to the discrete-time model.

4.2.1 Equivalent Circuit

The equivalent circuit of the voltage-source type PWM converter of Fig. 3-2 is represented by the ideal voltage and current sources within the rectangular box in Fig. 4-1. When the high-frequency switching harmonics are neglected, each of the valve pairs is represented by an equivalent sinusoidal ac voltage source v_{ra}, v_{rb} and v_{rc} on the ac terminal of the PWM converter. On the dc side, the converter acts as an ideal dc current source I_{dc}. Representing the distant power plant as a three-phase ac voltage source v_{sa}, v_{sb} and v_{sc} and the transmission line as a resistance R and an inductance L in series, the equivalent circuit of the radial line connected to the PWM-HVdc converter is shown in Fig. 4-1. On the dc side, the ideal dc current source, I_{dc}, feeds a load current I_{out} and the charging current of the capacitor C_{dc}.

4.2.2 Mathematical Model

By using *Kirchhoff* 's Voltage Law and Current Law and by using the conservation of power, a set of nonlinear equations are derived to describe the system of Fig. 4-1. In this study, the magnitude and frequency of the equivalent source v_{ra}, v_{rb} and v_{rc} are controlled to be equal to those of the ac source v_{sa}, v_{sb} and v_{sc}. Only the phase angle δ is to be adjusted to keep the dc voltage V_{dc} fixed as a

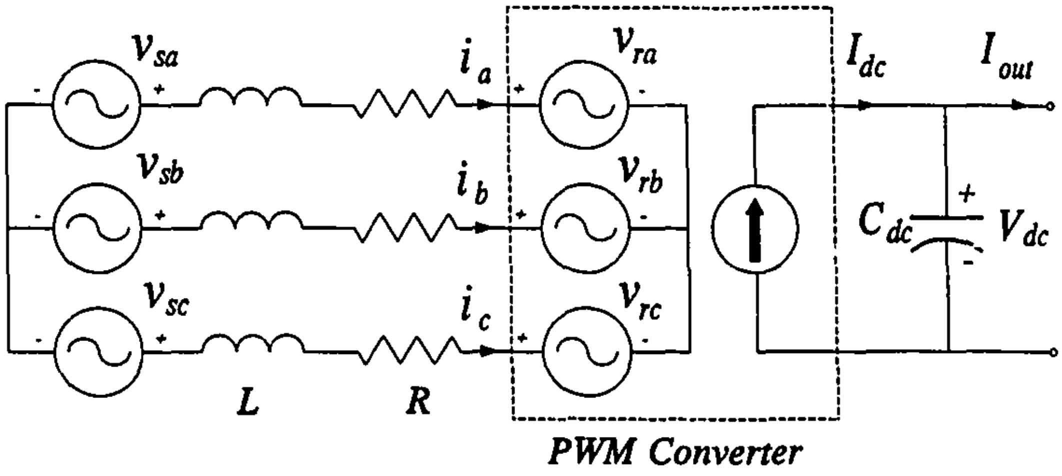


Figure 4-1 Equivalent circuit of PWM converter.

constant at the reference setting $V_{dc\,ref}$. This is because the converter is operated as a dc voltage regulator. Implicitly the converter varies δ to rectifier or invert power so that I_{dc} keeps C_{dc} charged in the face of changes in the load current I_{out}.

The voltages and the current of the sending and the receiving ends of the phase-a are defined as the following:

$$v_{sa}=\sqrt{2}\,V_s\sin(\omega_s t) \tag{4-1}$$

$$v_{ra}=\sqrt{2}\,V_s\sin(\omega_s t-\delta) \tag{4-2}$$

$$i_a=\sqrt{2}\,I\sin(\omega_s t-\theta) \tag{4-3}$$

From *Kirchhoff*'s Voltage Law, the equation on the ac side for a-phase is:

$$L\frac{d}{dt}i_a+R\,i_a=v_{sa}-v_{ra} \tag{4-4}$$

The equations for phase *b* and *c* are similar to (4-4).

The equation on the dc side is:

$$C_{dc}\frac{d}{dt}V_{dc}=I_{dc}-I_{out} \tag{4-5}$$

Neglecting the switching losses, the power balance equation is:

$$V_{dc}\,I_{dc}=v_{ra}\,i_a+v_{rb}\,i_b+v_{rc}\,i_c \tag{4-6}$$

Thus the Eqs. (4-4), (4-5) and (4-6) represent the dynamics of the equivalent system in time domain. The nonlinearity appears in Eq. (4-6), which essentially is the mathematical model of the converter.

In order to eliminate the trigonometric time varying function in the voltages of Eqs. (4-1) to (4-3) and in the currents, the power invariant transformations [87] from the *a-b-c* to the *o-d-q* are applied to the three phase equivalent circuit. Two stages of power invariant transformations have been used: firstly the Park's Transformation from *a-b-c* to *o-α-β* frames and then from *o-α-β* to the o-d-q frames. It is assumed that the zero-sequence is not present. The system dynamics in the d-q coordinate frame become:

$$L\frac{d}{dt}i_d+R\,i_d-\omega\,L\,i_q=\sqrt{3}\,K_{mod}\,V_{dc}\,sin\delta \tag{4-7}$$

$$L\frac{d}{dt}i_q+\omega\,L\,i_d+R\,i_q=\sqrt{3}\,K_{mod}\,V_{dc}\,cos\delta-\sqrt{3}\,V_s \tag{4-8}$$

$$C_{dc}\frac{d}{dt}V_{dc}=-\sqrt{3}\,K_{mod}(sin\delta\,i_d+cos\delta\,i_q)-I_{out} \tag{4-9}$$

where the K_{mod} is from Eq. (3-7).

The design of the control system is based on the equations of the plant in the *d-q* coordinates. As will be apparent later, some modifications have to be added to Eqs. (4-7) to (4-9). This is because the *o-α-β* to *o-d-q* transformation which removes the trigonometric time varying functions existing in the voltages and currents of Eqs. (4-1) to (4-3) assume that the frequency ω_s is a constant. In actual fact, the frequency ω_r in the converter ac voltages v_{ra}, v_{rb} and v_{rc} can be varied by the PWM control. In Eqs. (4-7) to (4-9), the PWM control is restricted to the voltage angle δ. The modification to take into account of the controllability of the frequency ω_r is deferred to **Sections 4.3.3** and **4.4.2** where it is introduced as a digital integrator in the block diagram of Fig. 4-4.

4.2.3 Small Perturbation Linearization

Eqs. (4-7) to (4-9) are nonlinear ordinary first order differential equations. As the pole-placement method can only be applied to linear systems, the development for the linear model, by small perturbation around an operating point for the converter, is necessary. Linearizing Eqs. (4-7), (4-8) and (4-9) about an equilibrium operating point (δ_o, V_{dco}, I_{do}, I_{qo}, I_{outo}), the state space representation of the linearized system can be expressed as:

$$\dot{x} = A\,x + B\,u + E\,d \tag{4-10}$$

$$y = C\,x \tag{4-11}$$

where

$$A = \begin{bmatrix} -\dfrac{R}{L} & \omega & \dfrac{\sqrt{3}\, K_{mod}\, sin(\delta_o)}{L} \\ - \omega & -\dfrac{R}{L} & \dfrac{\sqrt{3}\, K_{mod}\, cos(\delta_o)}{L} \\ \dfrac{-\sqrt{3}\, K_{mod}\, sin(\delta_o)}{C_{dc}} & \dfrac{-\sqrt{3}\, K_{mod}\, cos(\delta_o)}{C_{dc}} & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} \dfrac{\sqrt{3}\, K_{mod}\, V_{dco}\, cos(\delta_o)}{L} \\ \dfrac{-\sqrt{3}\, K_{mod}\, V_{dco}\, sin(\delta_o)}{L} \\ \dfrac{\sqrt{3}\, K_{mod}\, [\, sin(\delta_o)\, I_{qo} - cos(\delta_o)\, I_{do}\,]}{C_{dc}} \end{bmatrix}$$

$$E = \begin{bmatrix} 0 & 0 & -\dfrac{1}{C_{dc}} \end{bmatrix}^T$$

$$C = [\, 0 \quad 0 \quad 1 \,]$$

$$x = [\, \Delta I_d \quad \Delta I_q \quad \Delta V_{dc} \,]^T$$

$$d = \Delta I_{out} \ .$$

I_{do} and I_{qo} are the operating point currents of the converter in the *d-q* coordinate system; ΔI_d and ΔI_q are their small perturbations. V_{dco} is the operating point voltage of the dc link and ΔV_{dc} is its perturbation. δ_o is the phase angle between the source voltage v_s and the equivalent source v_r at the specified operating point and $\Delta\delta$ is the controlled component of the phase angle for the linearized system.

The input and output of the system are respectively:

$$u = \Delta\delta$$

and $y = \Delta V_{dc}$.

The external load current variation $d = \Delta I_{out}$ is considered as a disturbance variable.

4.2.4 Discrete Model of PWM Converter

For the application of digital control theory, the next step is to pass from the continuous time formulation of Eqs. (4-10) and (4-11) to the discrete-time formulation. It is assumed that the input signal $\Delta\delta$ and disturbance ΔI_{out} are held constant during the sampling period T_s. The discrete-time model of Eq. (4-10) and (4-11) becomes [78]:

$$x(k+1) = G\, x(k) + H\, u(k) + F\, d(k) \tag{4-12}$$

$$y(k) = C\, x(k), \qquad k = 0,\ 1,\ 2,\ .... \tag{4-13}$$

where

$$x(k) = [\ \Delta I_d(kT_s) \quad \Delta I_q(kT_s) \quad \Delta V_{dc}(kT_s)\ ]^T$$

$$G = exp(AT_s)$$

$$H = \int_0^{T_s} exp(A\,\tau)\ d\tau\ B$$

$$F = \int_0^{T_s} exp(A\,\tau)\ d\tau\ E$$

.

The input, output, and disturbance variable respectively are:

$$u(k) = \Delta\delta(kT_s) \equiv \Delta\delta(k)$$

$$y(k) = \Delta V_{dc}(kT_s) \equiv \Delta V_{dc}(k)$$

$$d(k) = \Delta I_{out}(kT_s) \equiv \Delta I_{out}(k)$$

Fig. 4-2 shows the block diagram of the discrete model.

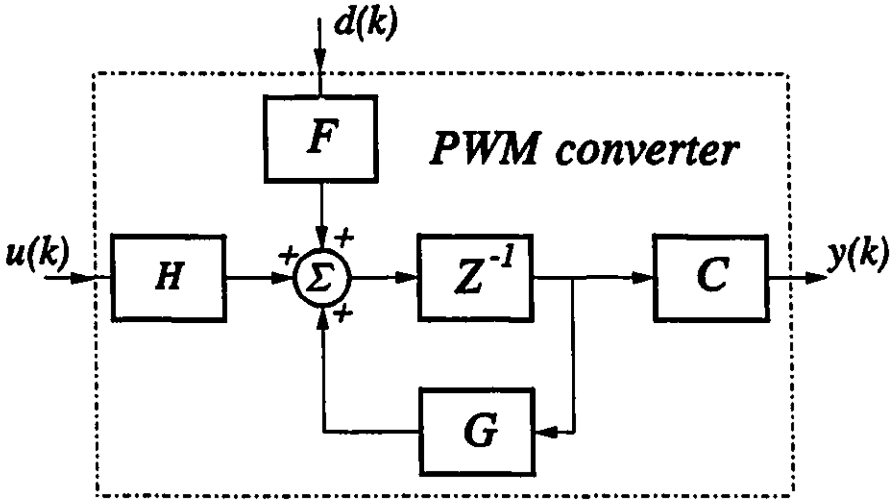


Figure 4-2 Discrete model of PWM converter.

4.3 DESIGN OF DIGITAL CONTROLLER

4.3.1 Design Philosophy

As already mentioned, one sees from Eqs. (4-7) to (4-9) that the converter is nonlinear. When the power converter is working under the normal operating condition, the power variation is relatively small. Thus it can be considered as a linear system around an operating point. The linear theory method can be used for the small perturbation control. But if by some reason a large change occurs to the system, the control must include the nonlinearity of the system. In the research work of this thesis, the nonlinearity of the PWM converter is dealt with in **Section 4.4.1** by considering 20 steady-state operating points and storing their G, H and C matrices in the memories.

For each of the linear models of the converter stored in the memories, the pole-placement method in modern control theory is chosen to stabilize the system and to improve its transient response for the steady-state operating point. Apart from the state feedback, an integrator is added to the controller in **Section 4.3.3** in order to eliminate the error of the steady states. As the state variables of the system are not all measurable, an state observer needs to be built for the state feedback controller. Fig. 4-3 shows the block diagram of the linear control system under design.

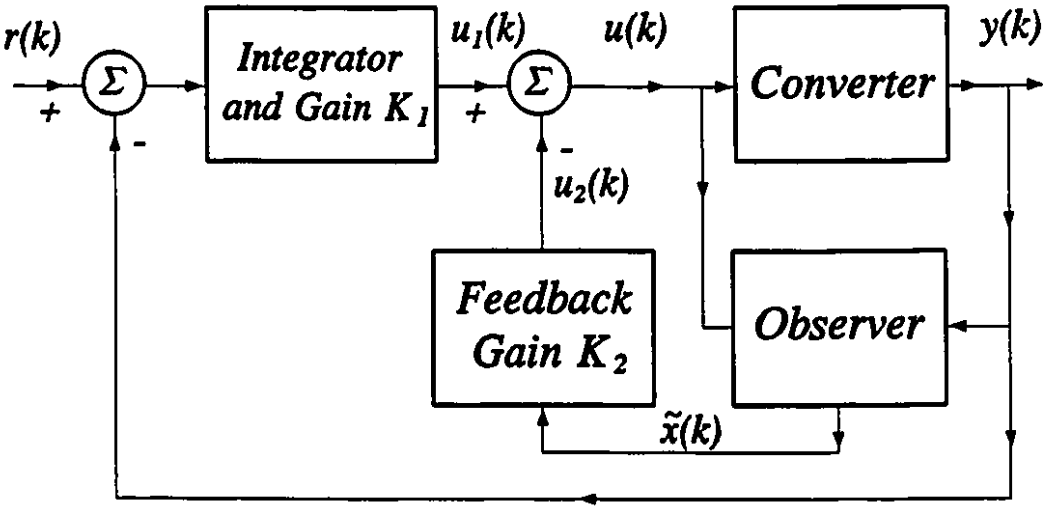


Figure 4-3 Control system of linearized converter.

4.3.2 Controllability and Observability

Prior to the pole-placement design of the linear controller, the controllability matrix and the observability matrix have been investigated under different operating points by off-line programming. The results have proved that the modelled plant is both completely controllable and observable based on standard methods described in [78].

4.3.3 Digital Integrator

When the entire control system is designed, an integrator must be added so as include the capability of the PWM converter to control the frequency ω_r of its ac output voltages. In Eq. (4-10), the control input is limited to $u = \Delta\delta$. In the P-I control of **Chapter 3**, the feedbacks channelled to the phase angle and the frequency levers of control have the attributes of proportional and integral feedbacks. For this reason, a digital integrator serves the same function as a frequency controller. This digital integrator ensures zero error in the dc voltage regulation feedback loop. Fig. 4-4 shows a block diagram of the digital integrator [79]. Its input-output relation is:

$$\begin{aligned} v(k+1) &= v(k) + (r(k+1) - y(k+1)) \\ &= v(k) + r(k+1) - C(Gx(k) + Hu(k)) \end{aligned} \tag{4-14}$$

4.3.4 Pole-Placement Design

As the converter is completely state controllable, all the poles of the closed loop system can be placed at any desired locations through the appropriate integrator gain K_I and a feedback gain K_2 which is introduced in Eq. (4-15). For the present development, without loss of generality, it is assumed that $d(k)=0$ and all the states $x(k)$ are accessible for feedback. Since only ΔV_{dc} is in the output, $x(k)$ is estimated from the state estimate $\bar{x}(k)$ using an observer described in **Section 4.3.5**. As shown in Fig. 4-3 and Fig. 4-4, the feedback control input $u(k)$ is given by

$$u(k+1) = K_I v(k+1) - K_2 x(k+1) \tag{4-15}$$

Substituting $v(k+1)$ by Eq. (4-14) and $x(k+1)$ by Eq. (4-12) for $u(k+1)$:

$$\begin{aligned} u(k+1) &= K_I v(k) + K_I(r(k+1) - C(Gx(k)+Hu(k))) - K_2(Gx(k)+Hu(k)) \\ &= u(k) + K_2 x(k) + K_I r(k+1) - (K_2+K_I C)(Gx(k)+Hu(k)) \end{aligned} \tag{4-16}$$

Together with (4-12), the closed loop system is described by:

$$\begin{bmatrix} x(k+1) \\ u(k+1) \end{bmatrix} = \begin{bmatrix} G & H \\ K_2-K_2G-K_ICG & 1-K_2H-K_ICH \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} + \begin{bmatrix} 0 \\ K_I \end{bmatrix} r(k+1) \tag{4-17}$$

$$y(k) = [C \;\; 0] \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \tag{4-18}$$

Since the reference voltage is constant, $r(k) = r =$ constant. Assuming a

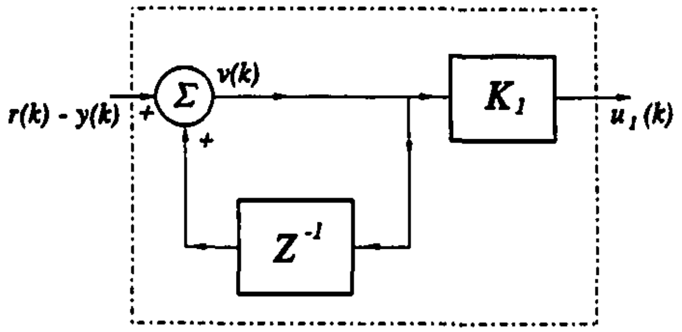


Figure 4-4 Block diagram of digital integrator.

stable system, in the steady state Eq. (4-17) becomes:

$$\begin{bmatrix} x(\infty) \\ u(\infty) \end{bmatrix} = \begin{bmatrix} G & H \\ K_2-K_2G-K_1CG & 1-K_2H-K_1CH \end{bmatrix} \begin{bmatrix} x(\infty) \\ u(\infty) \end{bmatrix} + \begin{bmatrix} 0 \\ K_1 r \end{bmatrix} \tag{4-19}$$

Defining

$$\begin{aligned} x_e(k) &= x(k) - x(\infty) \\ u_e(k) &= u(k) - u(\infty) \end{aligned} \tag{4-20}$$

Subtracting Eq. (4-19) from (4-17)

$$\begin{bmatrix} x_e(k+1) \\ u_e(k+1) \end{bmatrix} = \begin{bmatrix} G & H \\ K_2-K_2G-K_1CG & 1-K_2H-K_1CH \end{bmatrix} \begin{bmatrix} x_e(k) \\ u_e(k) \end{bmatrix} \tag{4-21}$$

This equation can be re-written as:

$$\xi(k+1) = \hat{G}\ \xi(k) + \hat{H}\ w(k) \tag{4-22}$$

where

$$w(k) = -\ \hat{K}\ \xi(k)$$

$$\xi(k) = \begin{bmatrix} x_e(k) \\ u_e(k) \end{bmatrix}$$

$$\hat{G} = \begin{bmatrix} G & H \\ 0 & 0 \end{bmatrix}$$

$$\hat{H} = [0\ \ 0\ \ 0\ \ 1]^T$$

$$\hat{K} = -\big[K_2-K_2G-K_1CG \quad 1-K_2H-K_1CH\big] \tag{4-23}$$

It can be shown that the system described by (4-22) is also completely controllable. Thus, the pole-placement method [79] can be applied to find the gains

K_1 and K_2. By specifying the desired pole locations or the dynamics for the closed loop control system, the characteristic polynomial $\varphi_c(\cdot)$ is defined. Since the matrix $\hat{G}$ is known, $\hat{K}$ can be calculated by using Ackermann's formula [78]:

$$\hat{K} = [\ 0\ \ 0\ \ 0\ \ 1\ ]\,[\ \hat{H}\ \ \hat{G}\hat{H}\ \ \hat{G}^2\hat{H}\ \ \hat{G}^3\hat{H}\ ]^{-1}\ \varphi_c(\hat{G}) \tag{4-24}$$

But from (4-23):

$$\hat{K} = [\ K_2\ \ K_1\ ]\begin{bmatrix} G-I & H \\ CG & CH \end{bmatrix} + [\ 0\ \ 0\ \ 0\ \ -1\ ] \tag{4-25}$$

Therefore,

$$[\ K_2\ \ K_1\ ] = (\ \hat{K} + [\ 0\ \ 0\ \ 0\ \ 1\ ]\ )\begin{bmatrix} G-I & H \\ CG & CH \end{bmatrix}^{-1} \tag{4-26}$$

4.3.5 Design of the Minimum-Order Observer

The output equation Eq. (4-13) contains only $\Delta V_{dc}(k)$. Since observability has been proved, it is possible to estimate the state variables $\Delta I_d(k)$ and $\Delta I_q(k)$ which can not be measured directly.

To reduce computing time, a minimum-order observer is designed to estimate the unmeasurable state variables. Let the state vector be partitioned into the unmeasurable part $x_1(k)$ and measurable part $x_2(k)$:

$$x(k) = \begin{bmatrix} x_1(k) \\ \hdashline x_2(k) \end{bmatrix} = \begin{bmatrix} \Delta I_d(k) \\ \Delta I_q(k) \\ \hdashline \Delta V_{dc}(k) \end{bmatrix}$$

Then, (4-12) and (4-13) can be re-written as:

$$\begin{bmatrix} x_1(k+1) \\ \cdots\cdots \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} x_1(k) \\ \cdots\cdots \\ x_2(k) \end{bmatrix} + \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} u(k) \tag{4-27}$$

$$y(k) = [\ 0 \ \vdots \ I\ ] \begin{bmatrix} x_1(k) \\ \cdots\cdots \\ x_2(k) \end{bmatrix} \tag{4-28}$$

where $$G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$$

$$H = \begin{bmatrix} H_1 \\ H_2 \end{bmatrix}$$

From (4-27),

$$x_1(k+1) = G_{11}\ x_1(k) + [\ G_{12}\ x_2(k) + H_1\ u(k)\ ] \tag{4-29}$$

and $$x_2(k+1) - G_{22}\ x_2(k) - H_2\ u(k) = G_{21}\ x_1(k) \tag{4-30}$$

By the well known procedures of the state reconstruction [78], the minimum-order observer can be determined by:

$$\begin{aligned} \tilde{x}_1(k+1) &= (G_{11} - K_e G_{21})\tilde{x}_1(k) + G_{12}x_2(k) + H_1 u(k) \\ &\quad + K_e[x_2(k+1) - G_{22}x_2(k) - H_2 u(k)] \end{aligned} \tag{4-31}$$

where $\tilde{x}_1(k)$ is the estimate of $x_1(k)$, and K_e is the observer feedback gain matrix.

Substituting

$$y(k) = x_2(k) \tag{4-32}$$

and simplifying it,

$$\begin{aligned}\tilde{\eta}(k+1) &= (G_{11}-K_eG_{21})\,\tilde{\eta}(k) + [(G_{11}-K_eG_{21})K_e \\ &\quad +G_{12}-K_eG_{22}]y(k) + (H_1-K_eH_2)u(k)\end{aligned} \tag{4-33}$$

where $\tilde{\eta}(k) = \tilde{x}_1(k) - K_e\, x_2(k)$ (4-34)

Eqs. (4-33) and (4-34) define the minimum-order observer. The observer feedback gain K_e can be calculated by *Ackermann*'s formula [78, 79]. Thus

$$Ke = \varphi_o(G_{11})\begin{bmatrix} G_{21} \\ G_{21}\,G_{11} \end{bmatrix}^{-1}\begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{4-35}$$

where $\varphi_o(\cdot)$ is the characteristic polynomial of the observer whose poles are

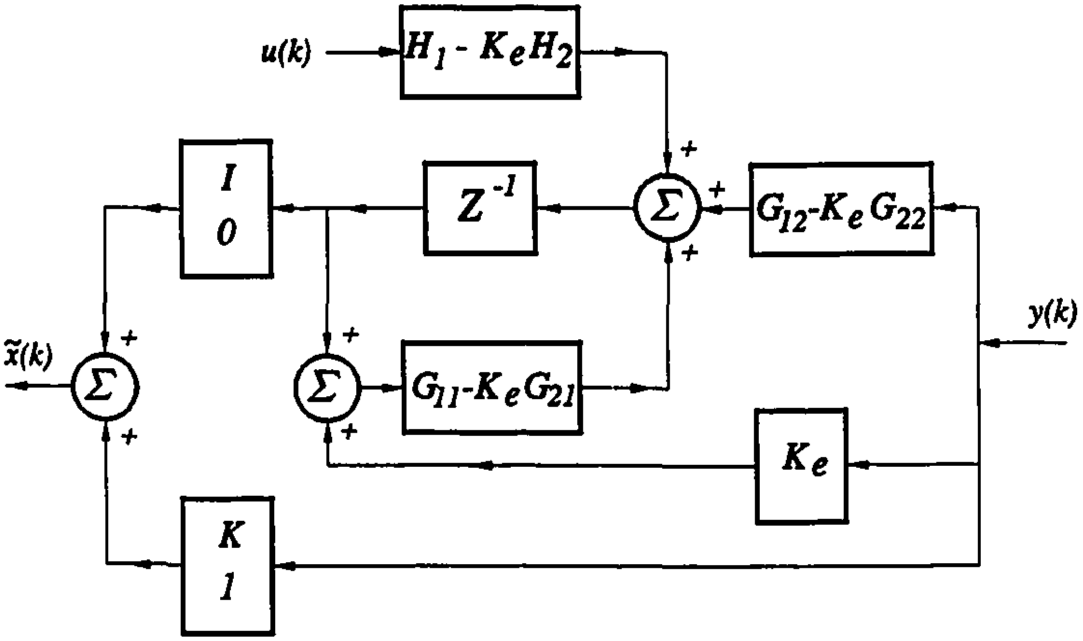


Figure 4-5 Block diagram of minimum-order observer.

arbitrarily chosen to give a rise time which is *2* to *3* times faster than that of the closed loop system. Fig. 4-5 shows the block diagram of the observer.

4.4 IMPLEMENTATION OF REAL-TIME DIGITAL CONTROL

The design of the digital controller described in **Section 4.3** has been applied to control the laboratory voltage-source type PWM converter shown in Fig. 3-2. As in the P-I controller in **Chapter 3**, the valves are activated by the PWM pulse generators of the multiprocessor system through the opto-couplers and base drives. The *'four timer'* method [4] is used again to implement the multiprocessor based sinusoidal pulse width modulation strategy.

4.4.1 Treatment of Nonlinearity

The nonlinear converter has been represented by linearized models at *20* operating points. The models have been pre-determined using the equations developed in **Chapter 3** and Eqs. (4-10) to (4-13) in this chapter. The matrices G, H, C and their corresponding gains K_1, K_2, K_e have been stored in the memories of the multiprocessor system. When the converter is under the digital control, the output power of the converter is computed by the product of the periodic samplings of the dc link current I_{dc} and voltage V_{dc}. From the real power of the converter, the operating point is estimated. The corresponding G, H, C matrices and the key parameters are read from the memories for the computation of the control input $u(k)$

in real time. This approach is equivalent to the "Gain Schedule" method in the adaptive control theory [78].

4.4.2 Implementation of Frequency Control

In the *d-q* frame formulation of the converter model in **Section 4.2.3**, only the phase angle is considered as the control input, that is $u = \Delta\delta$. The formulation omits the capability of the PWM converter to control the frequency ω_r. In fact, ω_r has to be adjusted to be equal to ω_s, the frequency of ac source voltages before synchronization. Furthermore, both ω_s and δ drift with time and the feedback controls of the PWM converter must ensure that it remains locked to the ac system.

In **Section 3.4**, it has been described that the frequency control is implemented using the "Four Timer Method" by adjusting the sampling period T_s of the main-timer through the equation

$$T_s = \frac{\pi}{K\,\omega_r} \tag{4-36}$$

where the converter frequency ω_r is related to the system frequency ω_s by

$$\omega_r(k) = \omega_s(k) + \Delta\,\omega_r(k) \tag{4-37}$$

In applying the pole-placement method, the omission of the frequency control has been remedied in **Section 4.3.3** by adding the digital integrator of Fig. 4-4. At this point, it is necessary to clarify the part played by the digital integrator so that the main timer can be loaded with the sampling period T_s which includes the $\Delta\omega_r$

adjustments in Eq. (4-37).

For linear feedback, one assumes that the perturbation frequency in Eq. (4-37) has the relation:

$$\Delta\omega_r(k) = K_f y(k) \tag{4-38}$$

where the constant K_f is to be determined.

Returning to **Section 4.3.3**, one has from Eq. (4-14)

$$v(k) = v(k-1) + r(k) - y(k) \tag{4-39}$$

and $$u_1(k) = K_1 v(k) \tag{4-40}$$

Substituting Eq. (4-39) into (4-40), then

$$u_1(k) = K_1 [v(k-1) + r(k) - y(k)] \tag{4-41}$$

As $r(k)$ can be set to zero for the dc voltage regulator,

$$u_1(k) = u_1(k-1) - K_1 y(k) \tag{4-42}$$

The term $-K_1 y(k)$ in Eq. (4-42) is the added phase angle generated by the integration of the frequency in the sampling period T_s. Since "sample and hold" is assumed,

$$-K_1 y(k) = \Delta\omega_r(k) T_s \tag{4-43}$$

Substituting Eq. (4-38)

$$-K_1 y(k) = K_f y(k) T_s \tag{4-44}$$

Thus the unknown constant K_f is determined from

$$K_f = -\frac{K_I}{T_{so}} \tag{4-45}$$

where K_I has previously been determined from Eq. (4-26) and T_{so} is the equilibrium sampling period of T_s when $\Delta\omega_r(k)=0$ in Eq. (4-37). The main-timer is loaded with T_s from Eq. (4-36), where $\omega_r(k)=\omega_s+K_f\,y(k)$.

4.4.3 Pole Locations of the Control System

From theory, the poles of the closed-loop system can be located at any place inside the unit circle (digital system) or on the left hand side (continuous system) of the complex plane by the pole-placement technique. Then the dynamic response of the system is defined by the pole locations. In practice, the pole locations cannot be placed anywhere because of the saturation limits of the physical sub-systems.

For the design of the control of the PWM converter, the pole locations have been chosen initially to be close to those of open loop system. Then the poles are moved in a direction to make the system to become more stable and to have faster damping.

4.4.4 Algorithm in Each Control Step

During the real-time operation, the control signals must be computed within the time interval T_s between the sampling instants. Based on the design described above, the algorithm in each control step is as follows:

At the beginning of each step, the dc voltage V_{dc} and current I_{dc} are sampled and the output dc power is computed as their product. From the dc current and the output dc power, the nearest operating point, $(\delta_o\ ,\ I_{do}\ ,\ I_{qo}\ ,\ V_{cdo})$, and its corresponding linear model G, H, C and the precalculated gains K_1, K_2 and K_e are accessed from the memory. From the dc voltage, ΔV_{dc} is calculated and the state $\bar{\eta}(k)$ is estimated from (4-33). By using $\bar{\eta}(k)$ and (4-34), the converter state variables ΔI_d and ΔI_q are computed. After this, the control input $\delta(k)=\delta_o+u(k)$ is determined by Eq. (4-15), where δ_o is the phase angle at the operating point. As shown in Fig. 4-3, $u(k)$ has two components $u_2(k)$ and $u_1(k)$ which are the contributions of the state feedback and integral controls respectively. By using the phase angle δ_o and $u_2(k)$, the timing data for the PWM pulse generators are calculated and loaded to the timers to switch the valves of the converter at the beginning of the next control step. The integral control component, $u_1(k)$, is implemented by the frequency control channel using the main-timer as described in **Section 4.4.2.**

4.4.5 Parallel Processing

The control algorithm has been implemented by the Multiprocessor Controller which has been described in **Chapter 2.** As already mentioned, in order to reduce the computing time, all the known matrices and parameters at every operating point have been calculated and stored in the memory beforehand. The computing tasks have been allocated to the 3 processors with the following guidelines.

1. The three processors must perform concurrent computation as much as

possible.

2. The access to the global memory for data communication must be efficient so as to decrease the waiting time for global memory.

3. The waiting time for synchronization should be minimized.

Fig. 4-6 shows the flow chart of program which has been written for parallel processing by the multiprocessor system. Each column represents the work each processor does during the real-time computation. Each row of the blocks represents

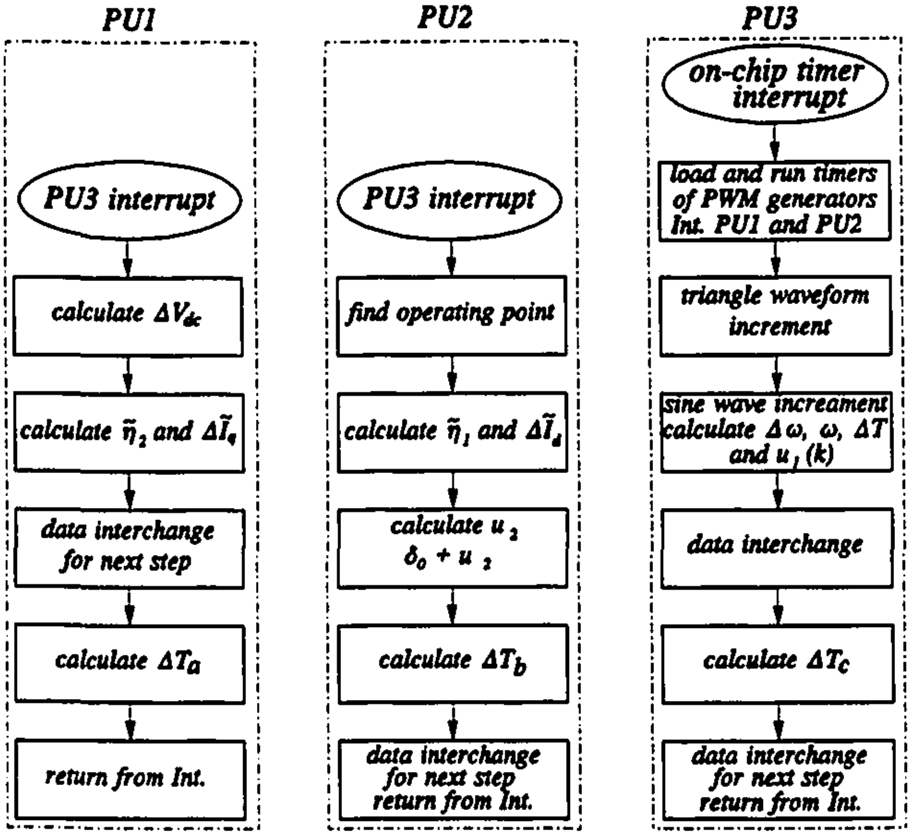

PU1
PU2
PU3
on-chip timer interrupt
PU3 interrupt
PU3 interrupt
load and run timers of PWM generators Int. PU1 and PU2
calculate ΔV_{dc}
find operating point
triangle waveform increment
calculate $\tilde{\eta}_2$ and $\Delta\tilde{I}_q$
calculate $\tilde{\eta}_1$ and $\Delta\tilde{I}_d$
sine wave increament calculate $\Delta\omega$, ω, ΔT and $u_1(k)$
data interchange for next step
calculate u_2 $\delta_0 + u_2$
data interchange
calculate ΔT_a
calculate ΔT_b
calculate ΔT_c
return from Int.
data interchange for next step return from Int.
data interchange for next step return from Int.

Figure 4-6 Flow chart of parallel processing program.

the work all the processors perform in one intermediate interlocked stage. The data interchanges occur before the interlocking instances.

The main computing tasks were distributed as follows:

Processor 1: Estimation of the output error $\Delta V_{dc}(k)$ of the control system; part of the calculations of the state observer and state feedback; the calculation of the timing data for the sinusoidal pulse width modulation (SPWM) strategy for phase *a*; user interface through the personal computer.

Processor 2: Determination of the operating point (δ_o, V_{dco}, I_{do}, I_{qo}, I_{outo}) and accessing its corresponding model *G*, *H*, *C* from the memory; part of the calculations of the state observer and the state feedback; finding the phase angle $\delta(k)=\delta_o+u_2(k)$; the calculation of the timing data for phase *b*.

Processor 3: Control of timing, status, and coordination of the multiprocessor system; implementation of the frequency control through $u_1(k)$; the calculation of the timing data for phase *c*; interface services for the converter.

The control procedure in each step is implemented in *5* stages. Fig. 4-7 shows the active computing times of the processors during each step computation. When the signal is 'high', the processor is active. Table 4-I shows the durations of the time for which the processors are active. All computing tasks have been completed in *260* μs of the *397* μs control period. The time taken by A/D sampling in *PU3* and by the communication of *PU1* with PC is not included.

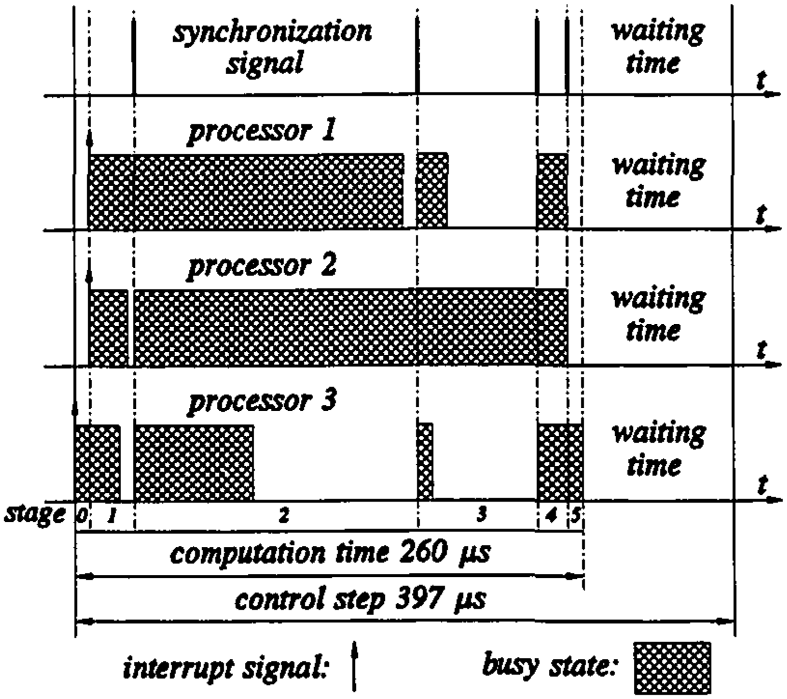


Figure 4-7 Timing diagram of parallel processing by processors.

Table 4-I Length of active computing time for processors.

stage	Processor			stage time
	#1	*#2*	*#3*	
0	*0.0 µs*	*0.0 µs*	*4.0 µs*	*4.0 µs*
1	*14.0 µs*	*8.0 µs*	*3.5 µs*	*14.0 µs*
2	*142.0 µs*	*166.0 µs*	*32.0 µs*	*166.0 µs*
3	*5.0 µs*	*58.0 µs*	*1.5 µs*	*58.0 µs*
4	*8.5 µs*	*7.0 µs*	*7.0 µs*	*8.5 µs*
5	*0.5 µs*	*1.0 µs*	*2.0 µs*	*2.0 µs*
Total	*170.0 µs*	*240.0 µs*	*50.0 µs*	*260.0 µs*

4.5 EXPERIMENTAL RESULTS

The experimental tests were designed with the research goals of proving the feasibility of the PWM-HVdc transmission system. As already mentioned in the Introduction in **Chapter 1**, and again in **Section 4.1**, the dc link capacitor, C_{dc}, had to be very large in order to stabilize the dc voltage regulator converter station under P-I control using analog control. Apart from the large cost, the large C_{dc} slowed the response time. This was reconfirmed in the experimental results of the digital P-I controller in **Chapter 3**. The experimental tests have been repeated with pole-placement control and they show that the PWM converter, configured as a dc voltage regulator, remain stable even for small C_{dc} and the speed of response improves. The success of the tests has proven: (1) the feasibility of PWM-HVdc transmission because the stability can be secured at economic costs [88], (2) the implementability of multiprocessor controlled pole-placement technique.

The laboratory PWM converter was identical to the one in **Chapter 3**.

4.5.1 Transients of Step Change of Load

The voltage-regulated rectifier system was tested with a dc link reference voltage of *110 V* and a 3-phase ac input of *31.1 V*. The modulation index for the sinusoidal PWM was set at *0.8* and held constant, and the number of carrier triangles per modulating signal period was *21*.

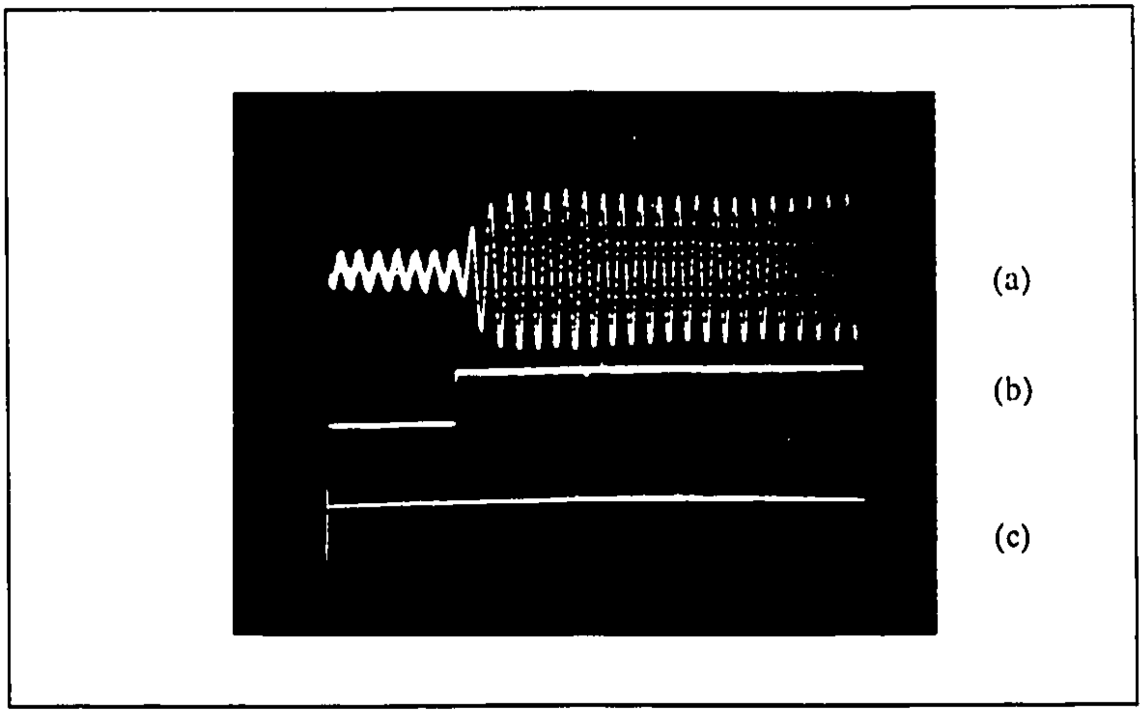


Figure 4-8 Transient response to step change in dc current demand in pole-placement controller. (a) ac line current, (b) dc link current, and (c) dc link voltage. $C=24000\ \mu F$.

Fig. 4-8 shows an oscillogram of the transient response for a step change in the dc load current demand when the system has the poles placed at $-162\pm j377$, -32, and -0.60. The traces are: (a) the response of the ac line current from $1.1\ A_{rms}$ to $5.0\ A_{rms}$ in the steady state; (b) the step change in dc link current demand from $0.5\ A$ to $3.0\ A$; and (c) the dc link voltage, $110\ V$.

For comparison, Fig. 4-9 shows the response of the system under the best proportional-plus-integral feedback control described in **Chapter 3**. The dc link capacitance used in both cases was $24000\ \mu F$. Clearly, pole-placement with state feedback gives a faster and less oscillatory response in the ac line current.

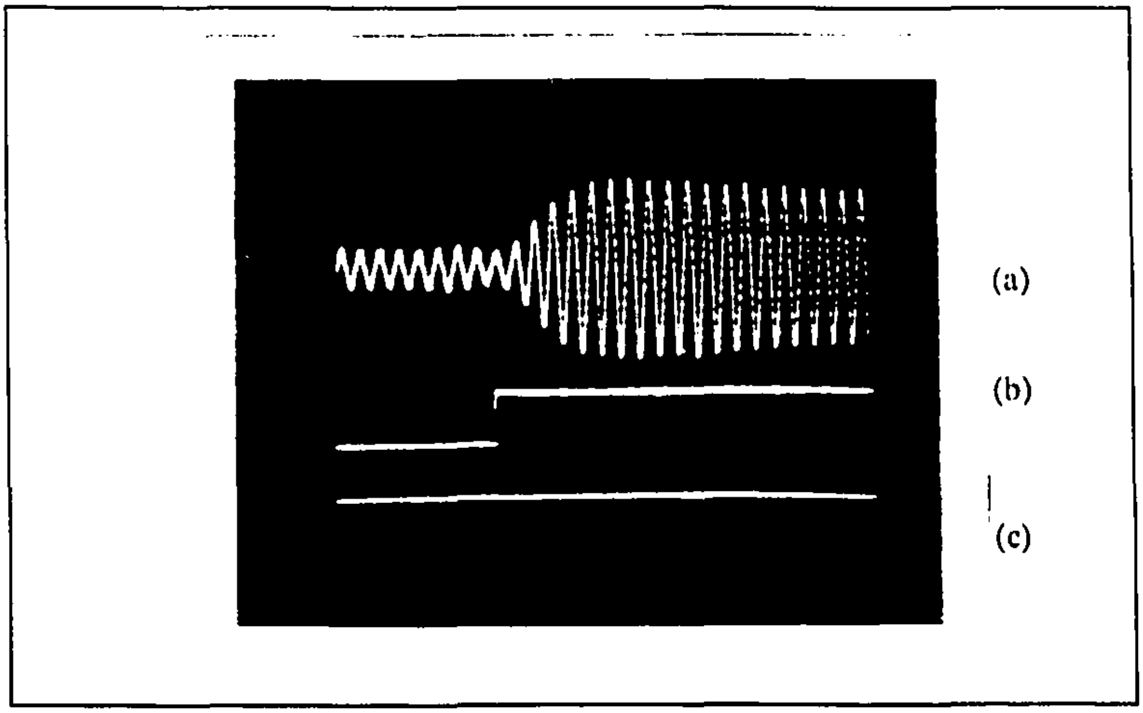


Figure 4-9 Transient responses of digital P-I controller of **Chapter 3**. (a) ac line current, (b) dc link current, and (c) dc link voltage. *C=24000 μF*.

4.5.2 Transient under Reduced Capacitor Size

Fig. 4-10 shows the same response of the system with pole-placement control when the dc link capacitance was substantially reduced to *200 μF*. The poles were placed at *-200±j470*, *-200*, *-0.65*. The step change in dc load current demand is the same as the previous cases. Because of the small capacitance, the dc link voltage (Trace (c)) exhibits a noticeable transient lasting for about *35 ms*. However, it is important to note that the system is stable, whereas the P-I controlled system would be unstable with such a small capacitance at the dc link.

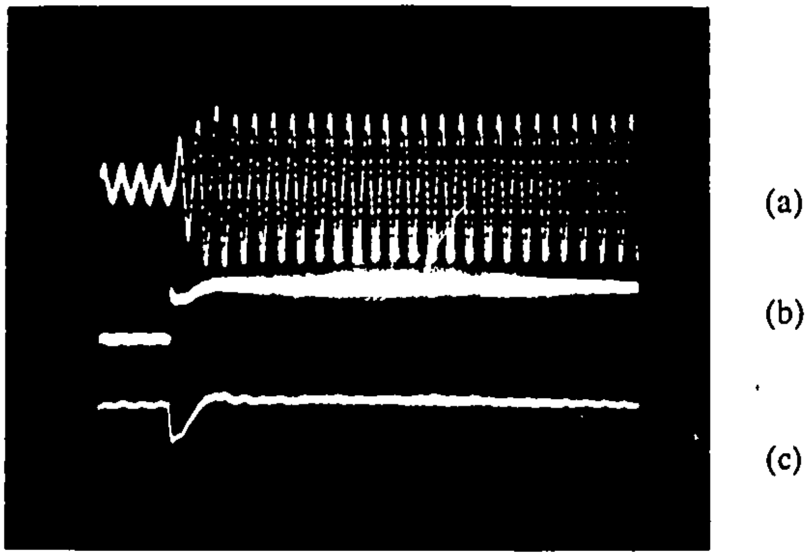


Figure 4-10 Transient response of a step change in pole-placement controller. (a) ac line current, (b) dc link current, and (c) dc link voltage. *C=200 μF.*

4.6 CONCLUSIONS

The pole-placement control through state feedback has been shown to provide better transient response and stability to the voltage-source type PWM converter operating as a dc voltage-regulator. In implementing the pole-placement control, the real-time digital controller has been based on the multiprocessor system with the three processors performing concurrent computation. The experimental results show that the dc link capacitance of the PWM converter can be substantially reduced while ensuring not only stability, but also fast transient response. The reduction from

24000 μF to *200 μF*, by a factor of *120*, brings the per unit size of the capacitor within the practical range. This finding makes the PWM-HVdc transmission system a viable concept as all the components sizes [88] are now comparable to the existing HVdc transmission system based on thyristor technology.

CHAPTER
FIVE

EXTENSIBLE MODULAR MULTIPROCESSOR SYSTEM

5.1 INTRODUCTION

For real-time digital control of dynamic systems, the computation of the control algorithm must be completed within the control time increment. As the time increment is inversely proportional to the system bandwidth, the demand on the computing system can be very severe, especially for the dynamic systems of wide bandwidth. Thus, given a computing system, the design of the control algorithm is limited by what can be implemented in the time increment. In the pole-placement control of the PWM converter presented in **Chapter** 4, the time increment used was 397 μs and it was necessary to use the "table-look-up" method partly to implement the control algorithm. It is desirable to have flexible availability of computing power so that the design of control algorithm is not unreasonably constrained. Further, such availability of computing power is also needed in real-time digital simulation, especially of large systems such as the power systems. In order to provide this flexible computing power, the Extensible Modular Multiprocessor System (the Mark II) is designed.

Compared with the Multiprocessor Controller described in **Chapter 2**, this system has several improvements. The **first** and most important improvement is that the system can be easily extended to a more powerful one. The Computational Module and the I/O Module have been designed as building blocks. When more computing power is needed in a problem, more building blocks can be added. The **second** improvement is that a more advanced and faster processor is chosen. In particular, the processor can perform floating point operations and it has multiprocessing support. The **third** improvement is that the human interface of the system is enhanced. Each processing module has its own interface circuit to the bus of a personal computer (PC). One PC can connect with several modules and more PCs can work together through the connection of the multiprocessor system. The communications between the PC and the processors are implemented by using parallel port operation and direct memory access.

In the next section, the essential characteristics of the processor in the system will be described. **Section 5.3** presents the design of the hardware modules of the Extensible Modular Multiprocessor System, which includes the architecture of the processing unit (PU), the interface to the PC, and the interface to external equipment. In **Section 5.4** the system software is described which includes the system initialization and the communication. **Section 5.5** gives test results for the arbiter circuit and for the synchronization circuit. **Section 5.6** concludes the chapter. The detailed circuits and the software codes are documented in the technical manual [71].

5.2 ESSENTIAL CHARACTERISTICS OF PROCESSOR

The processor chosen for the Extensible Modular Multiprocessor System is the TMS320C30 [53], which belongs to the third generation of the TMS320 digital signal processor family. It is a high performance CMOS 32-bit floating point device, which is specially designed to support parallel processing and other real-time embedded applications. It can operate at a speed up to 40 million floating-point operations per second (MFLOPS) at 20 million instructions per second (MIPS). Other special features include two external buses (the primary bus and the expansion bus), the interlocked instructions for multiprocessing support, the parallel instructions, the pipeline operation, and the DMA controller freeing its CPU from data transfers. This chip was selected because it was readily available, relatively powerful and economical when the system was planned.

5.3 SYSTEM HARDWARE DESCRIPTION

5.3.1 Overview

The Extensible Modular Multiprocessor System is composed of two types of system modules: the Computational Module and the I/O Module. The Computational Module is designed mainly to provide the computation power while the I/O Module has the interface circuit with the external devices. They are the basic " building blocks" of the system. For example, they can be connected together in a network to operate in real-time, as a transient network analyzer (TNA) by using as many modules as needed as described in **Chapters 7**.

Each module has 1 or 2 basically autonomous processing units (PUs), each of which has its own processor, clock, EPROM, local RAM, control and communication circuits, and the interface circuit with a personal computer (PC). The PUs signal each other by interrupt, interlock and I/O operations. The data interchanges are implemented through global memories [71].

Fig. 5-1 is a photograph of the prototype of the Extensible Modular Multiprocessor System, which includes two Computational Modules and one I/O Module. The circuit is built on the speed-wire boards. The clock frequency is *32 MHz*.

The processing unit (PU) is the main constituent part of the system

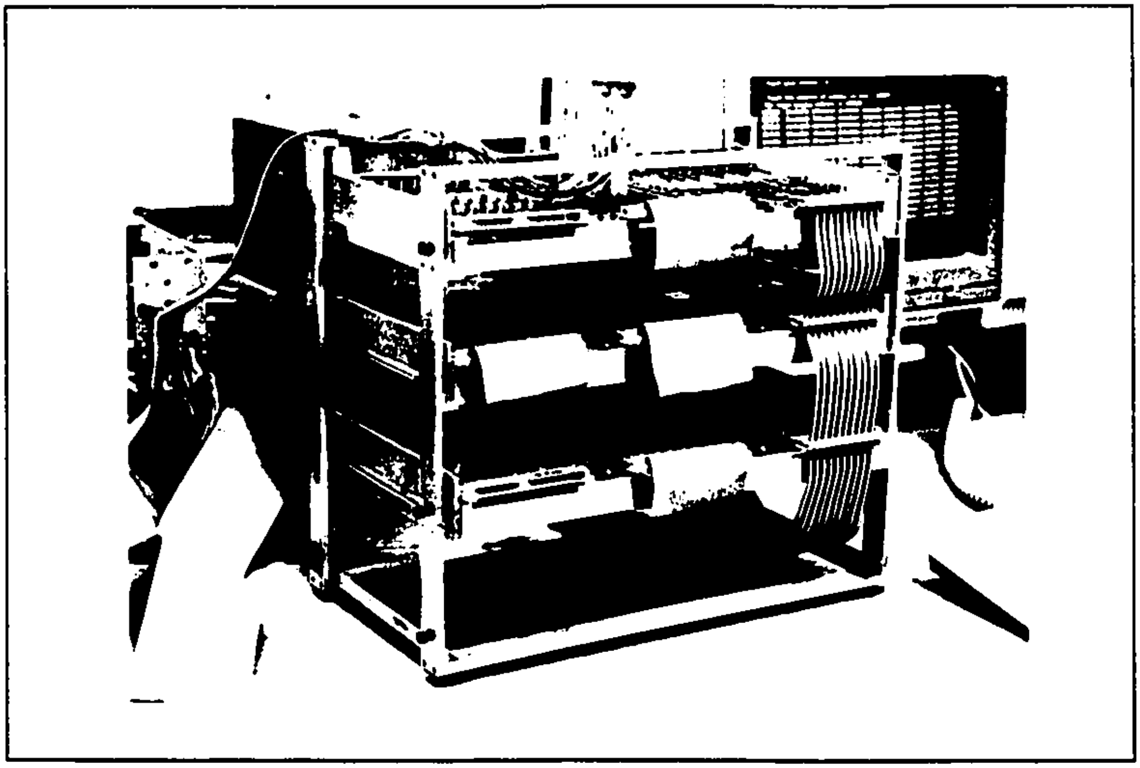


Figure 5-1 Prototype of Extensible Modular Multiprocessor System.

modules. Its design is described next.

5.3.2 Processing Unit (PU)

The processing unit is basically autonomous, comprising one digital signal processor with its own clock, EPROM, static RAM, and circuits of control and communications as shown in Fig. 5-2. The EPROM (32K words), connected to the primary bus, is needed to store the bootstrap program and system routines. The RAM (16K words) on the primary bus provides additional memory, in which the application program and data are usually stored. This RAM is also mapped to the memory space of the PC so that the personal computer can have direct

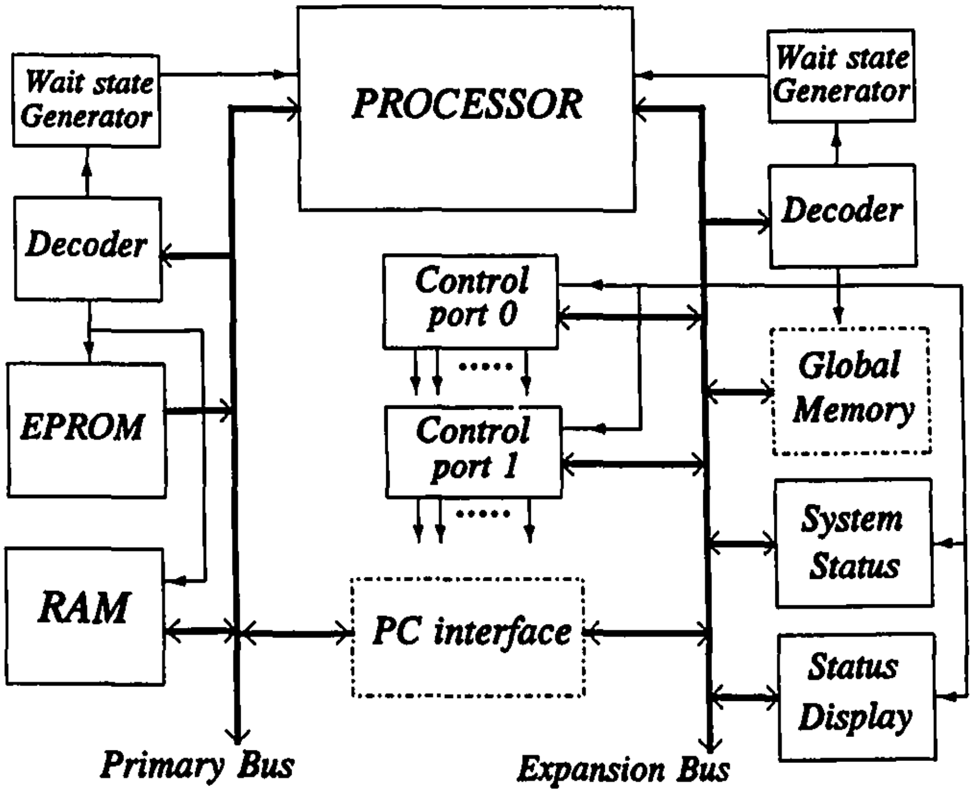


Figure 5-2 Block diagram of processing unit (PU).

access to it, as will be described in **Section 5.3.3**.

Connected to the processor's expansion bus are the global memory (GM) and some I/O ports. The output of the control port 1 defines the system configuration and the interrupt structure. The system configuration includes the specification of: i) the global memory as the multi-module or the single-module configuration, detailed in **Section 5.3.2**, ii) the PC-DSP data interchange registers as one group of 32-bit I/O ports or two groups of 16-bit ports, detailed in **Section 5.3.3**, and iii) the interrupt structure of enabling/disabling interrupts from the requests for reading/writing the data interchange register, and from the PC. Control port 0 can send out signals such as the interrupt requests, the interrupt acknowledgements, and the DMA requests to the PC and other PUs. The status display port is designed to monitor the status of the processor execution. Eight LEDs are connected to the output of the port. The flashing pattern of the LEDs can be programmed so that the different flashing patterns from the different parts of a test program will help in debugging the system. The system status port is used to get the system status from the PC and other PUs. The status signals include the interrupt requests from the PC and other PUs, the status of the requests for reading/writing the PC-PU data interchange register, and the status of the system interlock.

Access to the primary-bus RAM requires no wait state, but one wait state is needed for the access to the EPROM. One wait state is needed for the processor to access the global memory on the same module while two wait states are needed to access the global memory on another module. All the control and status ports are designed with proper wait states. Depending on the chip or the port selected, the corresponding decoder will signal the wait-state generator to send the ready signal to the processor with the required number of wait states.

5.3.3 System Modules

There are two types of modules in the system. One is configured for computations only, while the other includes input/output functions. The modules have been designed for easy compatible connection so that they can be used as building blocks to form an operational multiprocessor system by including as many modules as needed to within practical limits.

5.3.3.1 Computational Module

Fig. 5-3 shows a block diagram of the Computational Module. It consists of 2 basically autonomous processing units, a global memory and other circuits for control and communication purpose. The data interchanges between PUs are through the global memory.

Each Computational Module can work alone as a multiprocessor system or as a building block of a larger multiprocessor system, depending the configuration of the global memory. When it is configured as a one-module system, all the 8K-

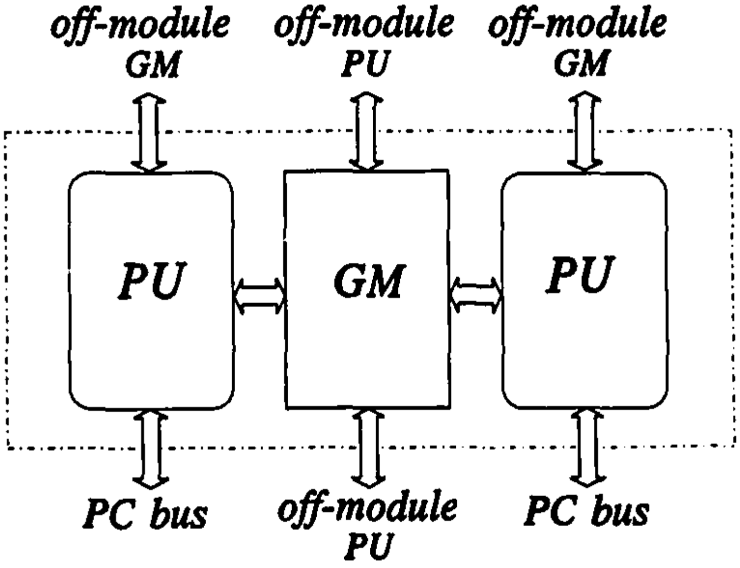


Figure 5-3 Computational Module.

words global memory is on the same module. When the global memory is configured for multi-module operation, the top 4K-word global memory is on the same module, but the remaining 4K-word memory is the map of the top 4K-word global memory of the neighbouring module to which it communicates.

Four ports have been designed for the global memory (GM), with 2 for the PUs in the same module and the remaining 2 for the PUs off the module. Each processor in the module has access to 2 global memories, with one in the same module and the other off the module. With this arrangement, each PU can transfer data directly to any of its 6 adjacent PUs through one GM, and the system becomes an extendible network. Fig. 5-4 shows an example of an 18-PU system with 9 Computational Modules.

In addition to the data interchanges through the GMs, control and status signals are transferred through a communication network. The communication network includes the connections of the requests and the acknowledgements cf the interrupts between processors, the connections of the requests and the acknowledgements of the interlock operation through the synchronization circuit, and the connections of the control and status ports between PUs. When the system is extended, the communication network needs to be extended so as to meet the requirement of the communication between the multiple processors detailed in **Section 5.3.5.**

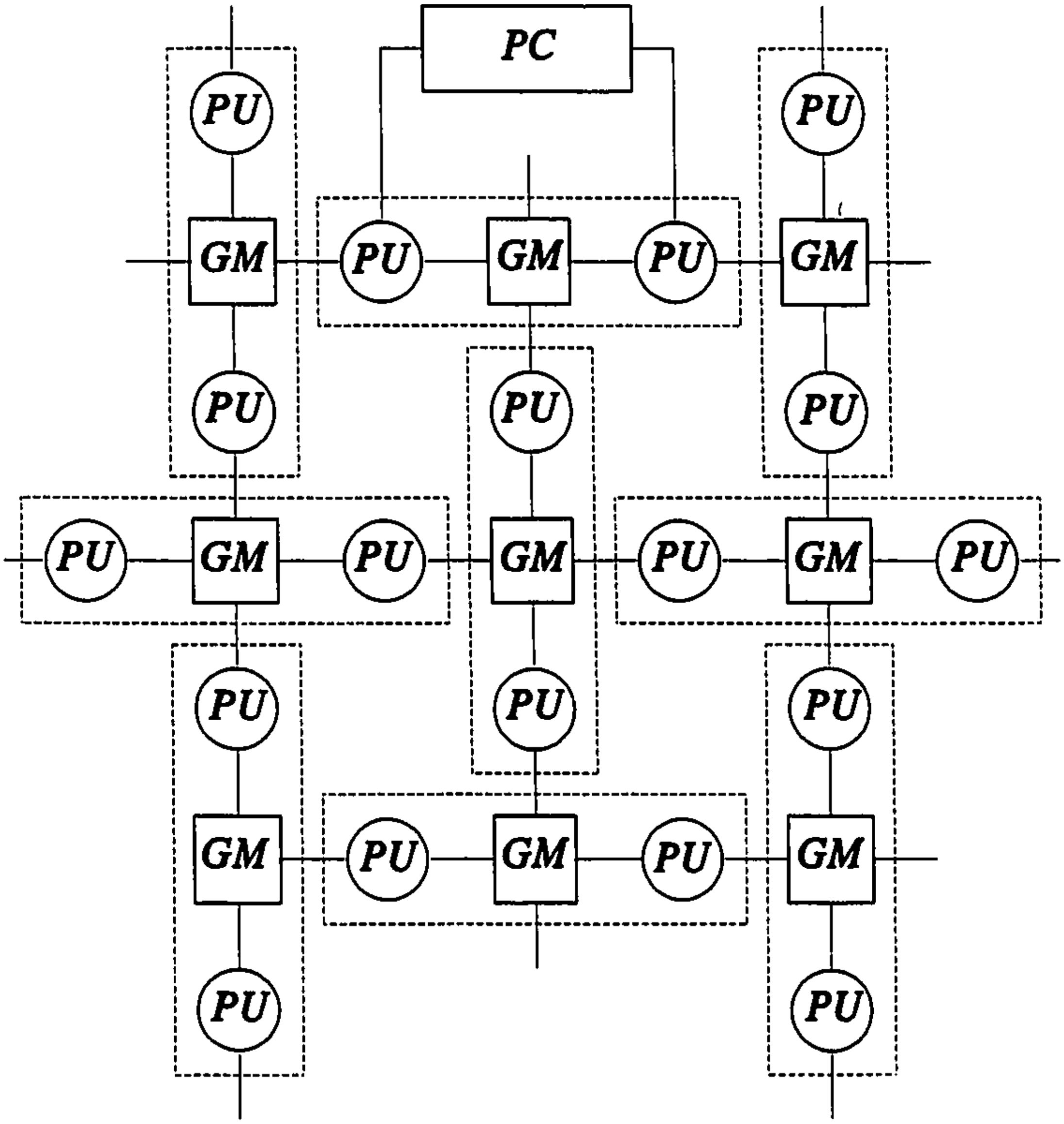


Figure 5-4 Example of extensible modular architecture.

5.3.3.2 I/O Module

The architecture of the I/O Module is shown in Fig. 5-5. It consists of one PU, one GM, A/D and D/A converters, programmable timers and other signal

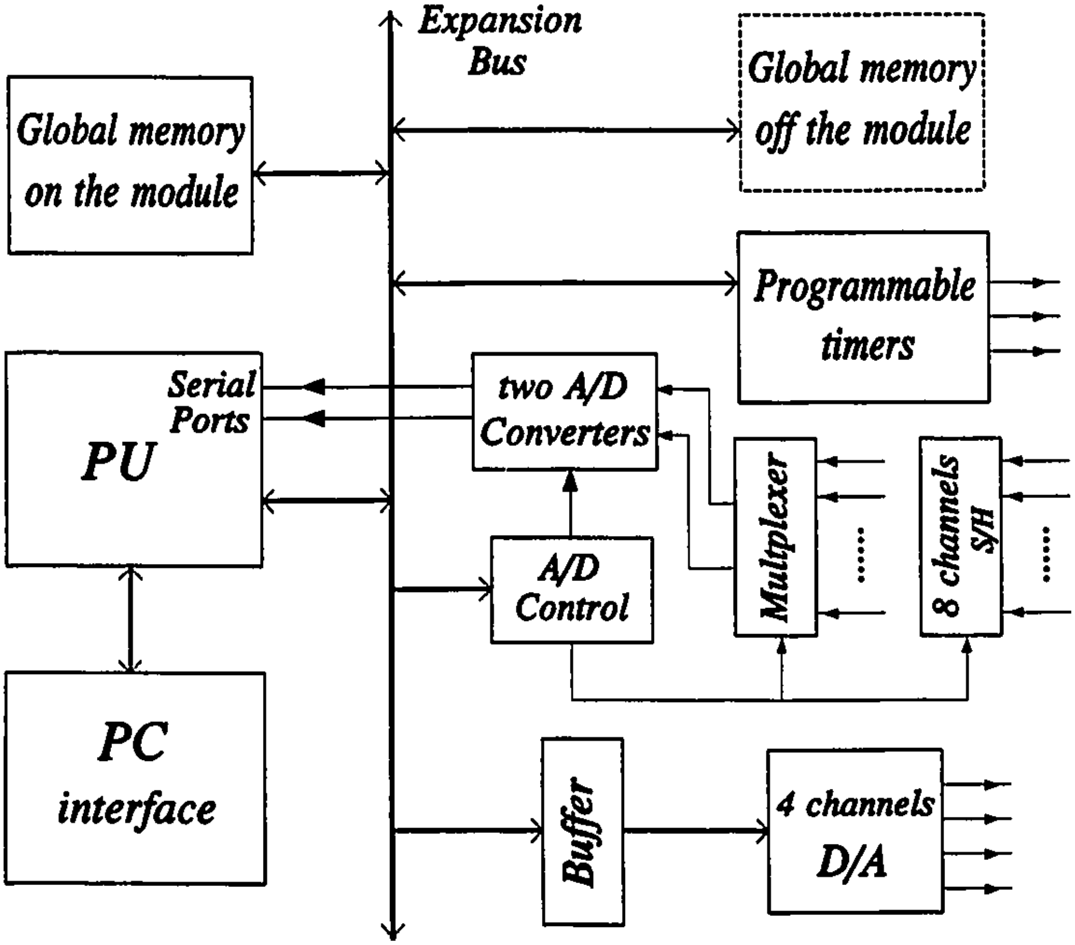


Figure 5-5 I/O Module.

conditioning circuits.

The PU on the I/O Module has two channels for the access to two GMs, one on the same module and the other on another module. Similarly, the GM has two ports for connections to two PUs. All the circuits are essentially the same as those on the Computational Module except the interface circuits for the external equipment.

There are two 16-bit A/D converters with a conversion time of *5 µs* connected to the two serial receiving ports of the processor. Eight channels of analogue signals can be converted through two multiplexers. The processor on the I/O Module can select the channels of the multiplexers and signal to start the conversion. A status register indicates the active channel.

There are four 16-bit D/A converters with *3 µs* conversion time in the I/O Module. The D/A converters receive data through data buffers on the expansion bus of the processor, which work as if they are the output ports.

There are six 16-bit timers in the I/O Module, which can be programmed to generate PWM timing sequences, for example. The signal conditioning circuits include amplifiers for system inputs, the sampler-and-holders for analog signals and buffers for control outputs.

5.3.4 Interface between PC and PUs

The PC is the host of the whole system, and the PUs appear as a block of memory and a group of I/O ports of the PC. Up to 32 PUs or 16 modules can be connected to the host. Fig. 5-6 shows the basic functional blocks of the interface circuits. Most of the interface circuit are built in the module, and the remainder is on a plug-in board for insertion in a 16-bit data slot of the PC (AT bus). The plug-in board is connected to the module through a 50-line ribbon cable, giving access on the module to the AT bus .

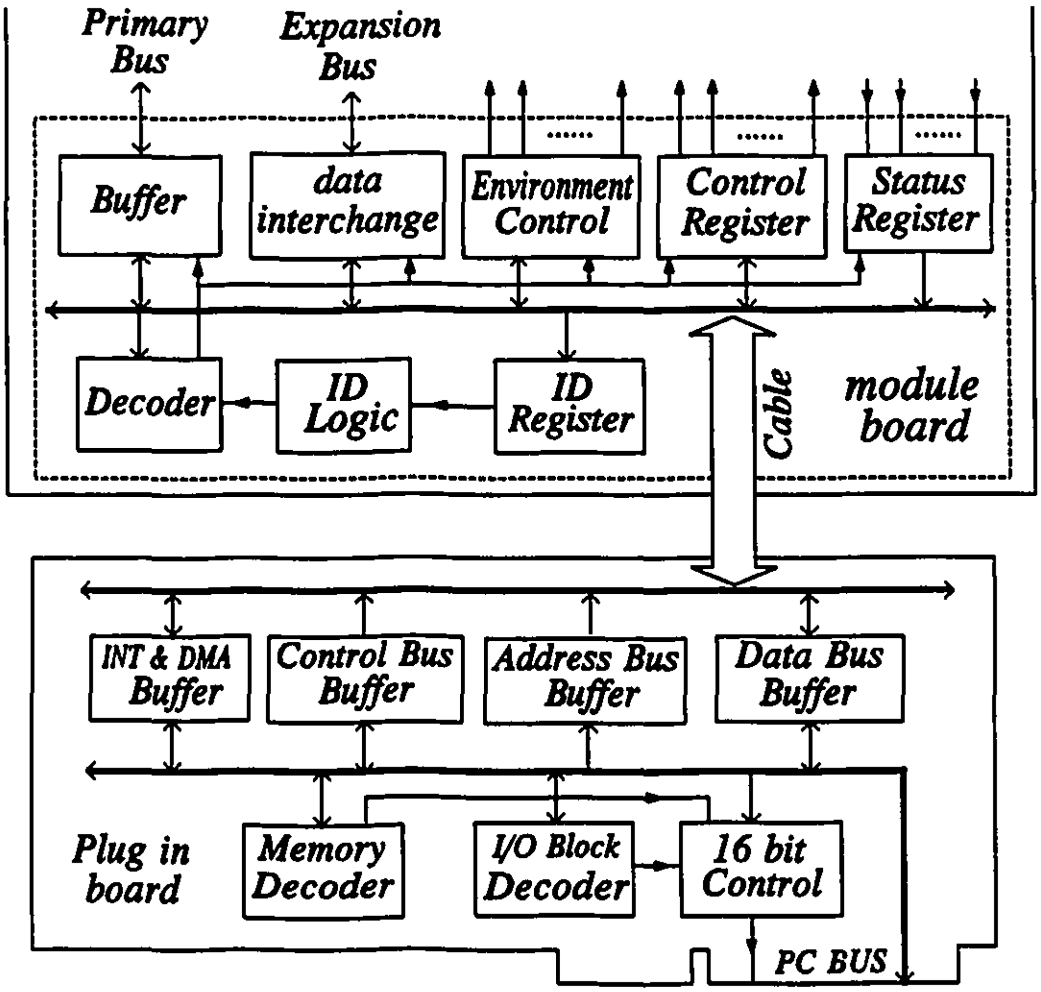


Figure 5-6 Interface circuit between PC and PU.

There is an ID register on each module. All the ID registers in the system are connected in parallel and use the same address of the I/O port of the PC. The PC can read from or write to the ID registers. Every PU is assigned an ID number in its control logic of the PC-PU communication. When the value in the ID registers matches the ID control logic of a PU, the corresponding decoder will be enabled and the PC can work with the memory and I/O ports. For example, if the number i is loaded in the ID register, the decoder for the communication between the PC and PUi is enabled. Then the PC can activate the corresponding buffers or I/O ports. Although the PC can be connected to several PUs in parallel, it can only work with one PU at a time.

There are several function blocks between the buses of the PC and each processor. In the data interchange block are two groups of 16-bit input/output ports. For the PC, they are configured as two input ports and two output ports; but for the processor, they can be configured as either 16-bit or 32-bit input/output ports. The PC and the processor can read from or write to the unidirectional ports for data interchanging at any time. The communication can be implemented under the interrupt control or the polling mode, described in Ref. [71].

There are three registers under the control of the PC. The environment control register is used to define the environment of the PC-PU communication. It includes the enabling or disabling of the interrupts coming from the PU, the determination of the available channels of DMA, and the use of the PC-PU data interchange registers by interrupt control or polling mode. The control register provides separate control signals such as the reset, the hold and release, the interrupt to the processor, and the clear to the interrupt. The status register receives status information from the PU, such as the read buffer full and the write buffer empty of the PC-PU data interchange register, the hold acknowledgement of the processor, and the processor interrupt status.

The PC can access the PU by using two methods. The first one involves the use of the transfer of address and data through the PC-PU data interchange ports. The second method maps the off-chip RAM on the primary bus of the processor onto the unused area of the PC memory space through a group of buffers. Before the PC's access to the RAM, it sends out a signal to request the processor to hold its operation. If its primary bus is permitted to be held at this time, the processor stops its current operation and returns an acknowledgement to the PC. At the same time, the processor sets all the primary address and data buses and the associated strobes in a high impedance state. When the PC receives the acknowledgement, it can work directly with the primary-bus RAM as its own memory.

On the plug-in board, the data, the address, and the control buses of the PC together with the interrupt and the DMA signals are buffered for cable connections to the modules. The 8 most significant bits of the 24-bit address of the PC are used to specify the memory block for the local RAM of the processor. Bits 0-15 specify the memory and I/O address in the block. Bits 4-9 specify the external I/O block and bits 0-3 are used to address the I/O ports in the block. The decoders signal the 16-bit control circuit to put the PC into 16-bit data operation.

5.3.5 Communication Between Processors

Fig. 5-7 is the block diagram of the prototype system which has been built

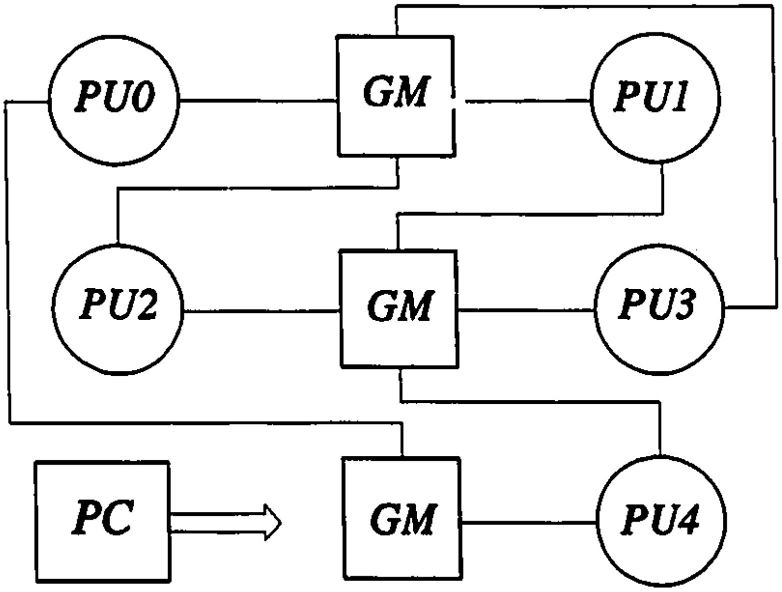


Figure 5-7 Connection of modules in prototype.

and it is made up of two Computational Modules and one I/O Module. Every PU is connected with the other four through a global memory (GM). The communication between PUs on the same board, *PU0* to *PU1* and *PU2* to *PU3* in the Computational Modules, *PU4* in the I/O Module has been hardwired to the GM in the same board. Inter-board communications are through ribbon connectors.

5.3.5.1 External interrupts

There is an interrupt bus in the system for communication between PUs. It contains the signals of interrupt requests and acknowledgements. Fig. 5-8(a) shows the configuration of the interrupt bus. Five identical logic circuit are used

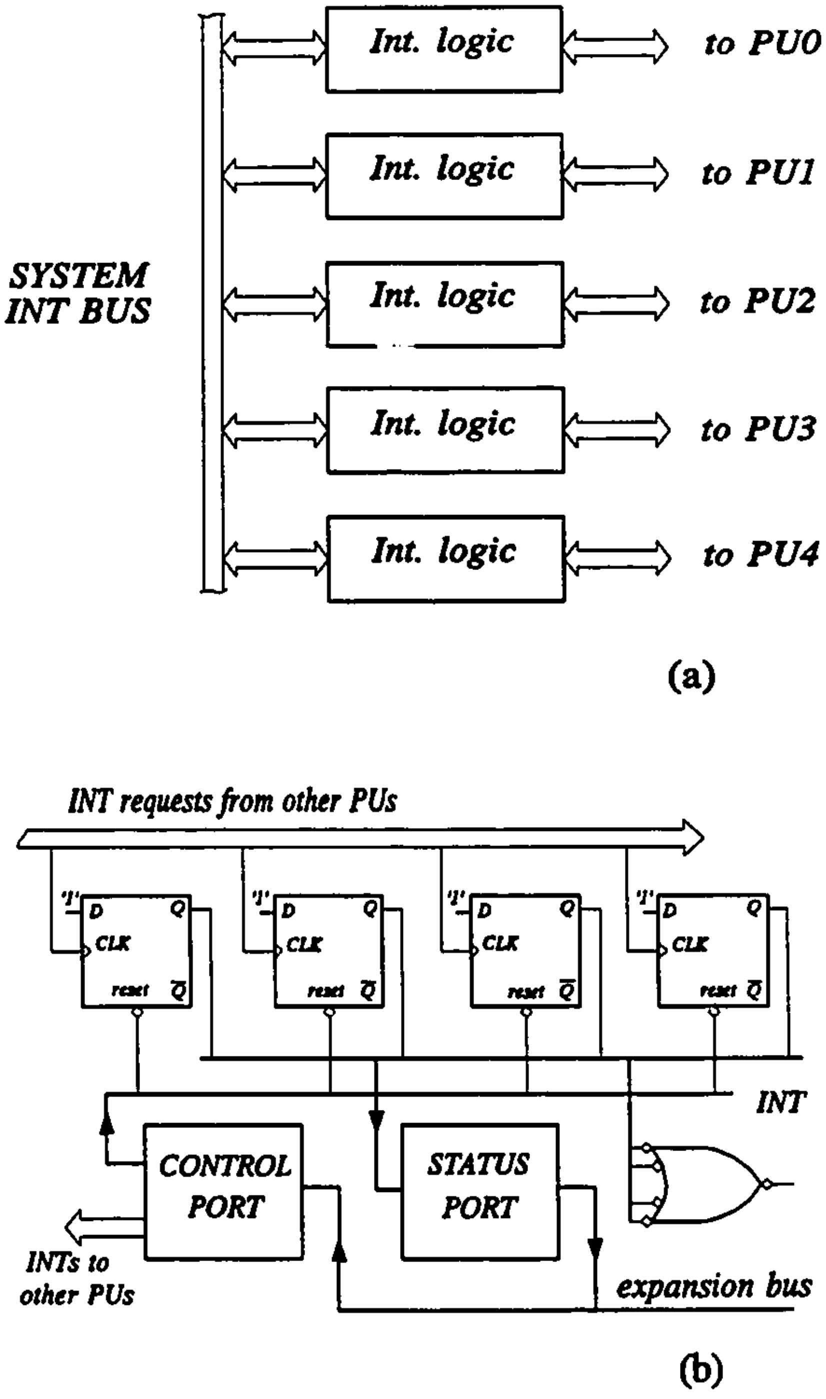


Figure 5-8 Interrupt circuit connections. (a) Block diagram, (b) Logic circuit for *PU0.*

to connect the interrupt bus to the five processors. Each processor has four external interrupt inputs (Ext.Int.0 - Ext.Int.3). Ext.Int.3 is used for communication between the PC and the processor. Ext.Int.2 is used here for the communication between neighbour processors. The remaining two external interrupts will be used for communications between groups and for other external requirements in the future.

Fig. 5-8(b) shows the logic circuit for *PU0.* Four flip-flops are used to register the interrupt requests. An OR gate is used to connect the four interrupt requests together to Ext.Int.2 of the processor. An interrupt signal will be generated whenever one or more interrupt requests are received. Four bits of the status port are used to identify the interrupt sources. The control port generates signals to clear the interrupt requests held in the flip-flops, and to interrupt other processors. The status and the control ports can also be used for 1-bit communication between processors if the interrupt is disabled.

5.3.5.2 Interlock operation

A circuit for the interlock operation has been described in **Section 2.3.** However, in order to minimize the connections between the modules, the circuit is re-designed for the system by using the built-in function of the processor.

Built-in function of the processor: The TMS320C30 processor has two pins XF0 and XF1 which are specially designed for the interlock operation between two processors. Several instructions are given by the manufacturer for operations on the two pins. The operation used here is that when XF0 is activated, it stays in the idle state until XF1 is activated from elsewhere. Then the processor disables XF0 to end the operation. In the case of two processor, the connection of the XF0s to the XF1s will achieve the interlock operation. For more processors, the following circuit is developed for the synchronization.

Synchronization circuit: Fig. 5-9 shows the synchronization circuit. The PUi req. signals (the request signals) are connected to the XF0s of the five processors. When some processors, but not all of them, ask for interlock operation, the corresponding PU req. signals become active. The ACKs of the circuit (the acknowledgement signals), which are connected to XF1s, remain inactive until all the PU reqs. become active. Then the circuit makes all the ACKs active at the same time to respond to the synchronization requests. When the processors receive the acknowledgement signals from their XF1s, they end the interlock operation and then continue to execute the program. After this, the XF0s become inactive automatically and ACKs return to the inactive state.

If for some reason, as an example, a processor is interrupted and it is running a service routine, the processor does not respond to the acknowledgement in time, its XF0 will be kept active by the built-in control logic of the processor, so that its corresponding ACK will be kept active by the logic of the synchronization circuit. When this situation happens, even if all the other processors ask for interlock operation again, the circuit will not respond to them

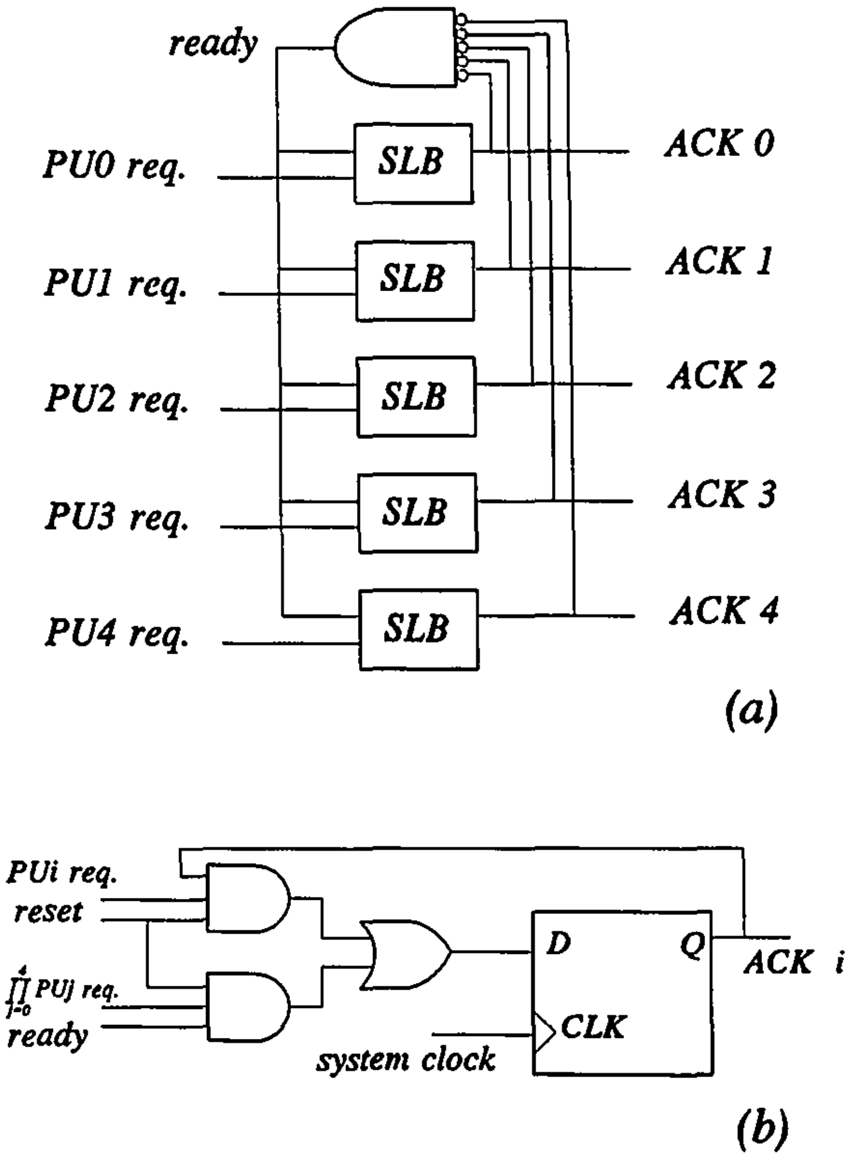


Figure 5-9 Synchronization circuit of system. (a) Block diagram of synchronization circuit, (b) Circuit in each synchronization logic block (SLB).

until all the processors complete the previous interlock instruction and ask for it again.

5.3.5.3 Global Memory

Architecture The global memory is a primary communication medium in the system. Fig. 5-10 shows the block diagram of the global memory. It is composed of a block of static random-access-memory (RAM), four groups of buffers and an arbiter circuit. Each group of the buffers contains a 32-bit bidirectional data buffer, a 13-bit unidirectional address buffer, and the read/write control signal buffers, all of which are between the processor expansion buses and the global memory bus. Although one global memory, as it is designed, can serve four

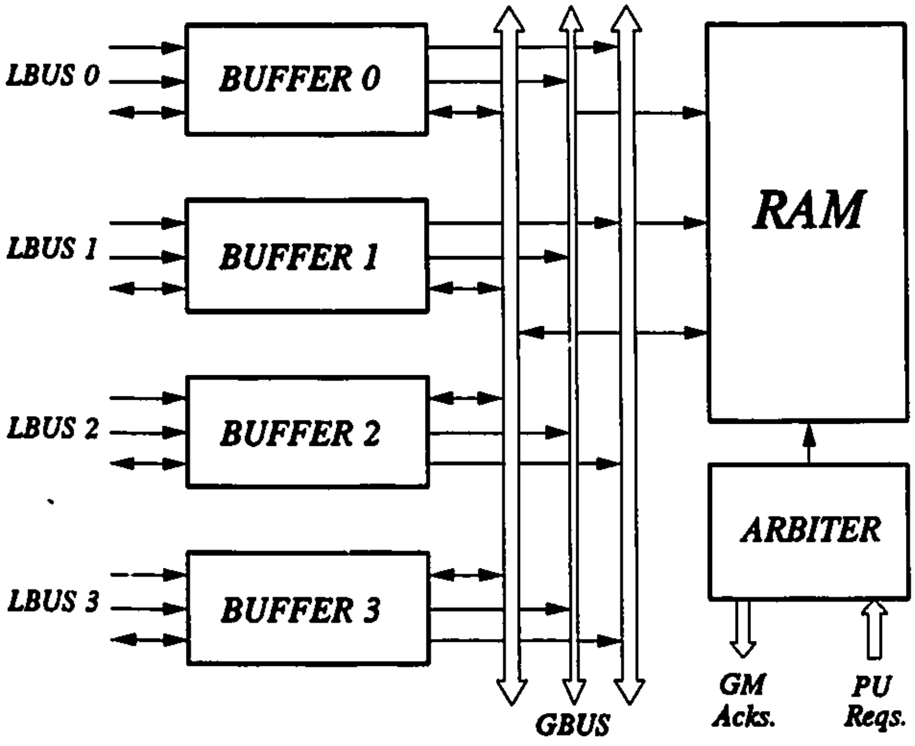
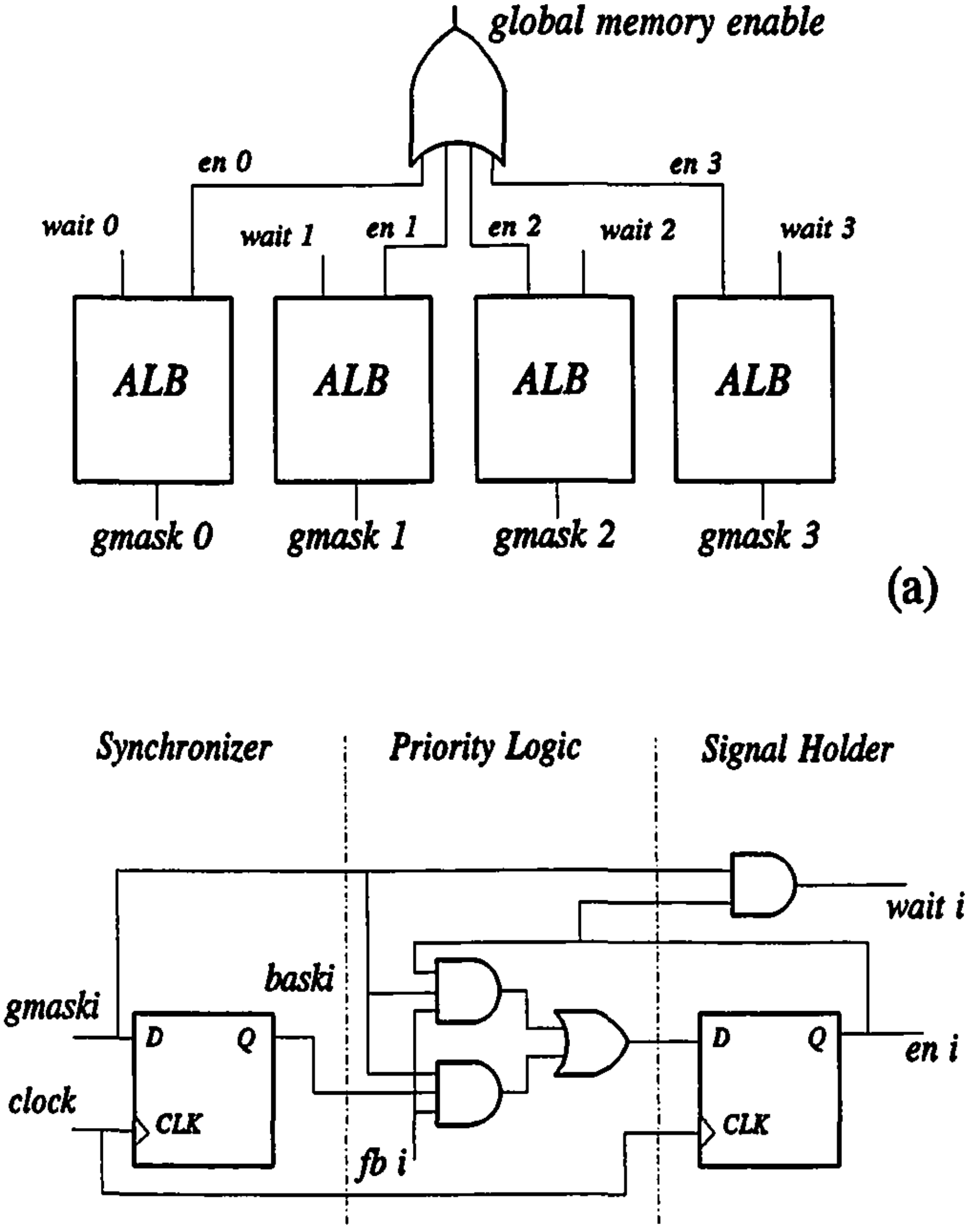


Figure 5-10 Organization of global memory. LBUS represents local bus and GBUS represents global (memory) bus.

processors only one processor can access the global memory at any time so that the conflict of data flow from different processors has to be avoided. The arbiter circuit controls the access to the global memory.

Arbiter logic circuit: Fig. 5-11(a) shows the functional block diagram of the arbiter circuit. The four blocks in the diagram have similar circuits for the four processors, and one of them is shown in Fig. 5-11(b). The operation of the arbiter circuit can be divided into three stages. The first one is the signal synchronizer. As the processors are working asynchronously with different clocks, their request signal "gmaski" cannot satisfy the timing requirement of the combination logic of the arbiter circuit. The second stage is the priority logic. When more than one processor request access to the global memory at the same time, only one request signal can be acknowledged. The priority is designed so that the processor in *PU0* has the highest priority and the processor in *PU3* has the lowest one. The third stage of the circuit is to keep the acknowledged signal active for one clock period so as to meet the need of the timing of the processor. The OR gate outputs the global memory enable signal. Whenever a request signal is acknowledged, the global memory is enabled.

When the circuit is in operation, the clock signal can be anyone from the four processors in the system. The "gmaski" is the input to the arbiter circuit, and the "baski" is the global memory request synchronized with the clock signal. The synchronized requests and the previous status of the arbiter go to the combination



$$fb\ i = \Big(\prod_{\substack{0\le j\le 3\\ j\ne i}} \overline{en\ j}\Big)\Big(\prod_{\substack{0\le l\le 3\\ l<i}} \overline{bask\ l}\Big)$$

Figure 5-11 Arbiter circuit of global memory. (a) block diagram of arbiter circuit, (b) circuit in arbiter logic block (ALB).

logic circuit to be arbitrated and the result is sent to the signal holder for the next output.

The "eni" is the output which controls the buffers of processor i as well as to the OR gate. The "waiti" goes to the ready pin of processor i.

5.4 SYSTEM SOFTWARE DESCRIPTION

The software for the system includes programs for the processors and for the PC, which are written in C and assembly languages. As the PUs in the system are basically autonomous, the bootstrap and system software for each processor are the same except for some service routines of input-output functions.

5.4.1 System Initialization

For the processors, the bootstrap programs reside in their EPROMs, which are written in assembly language. Fig. 5-12 shows the flow chart of the initialization program. At power up or system reset, the processor starts to execute from address 0. Some registers on the chip such as the status, interrupt mask and peripheral control registers are initialized. The environment or system configuration of the PU is defined through its control port 1. This includes the following: the data interchange ports are configured as 32-bit input/output ports, the global memory is configured for single module (for system initialization only), and the interrupts from the PC and from the reading requests for data

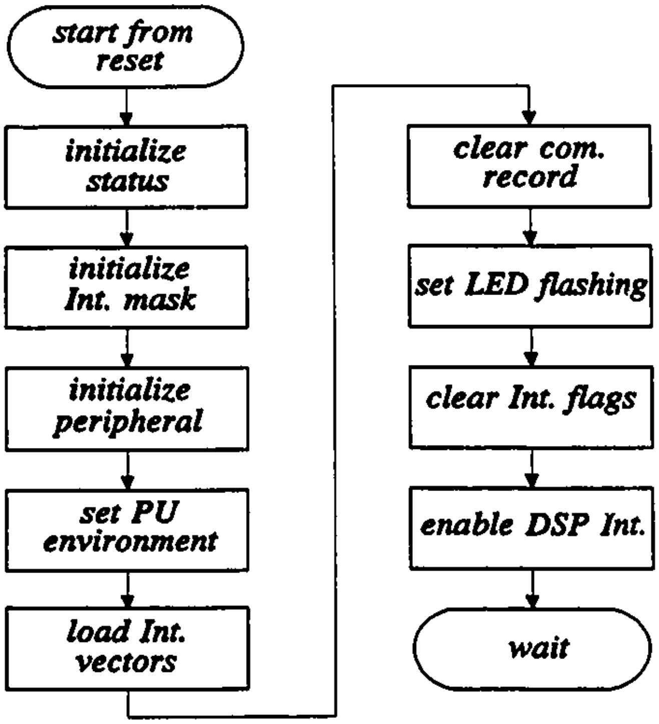


Figure 5-12 Flow chart of initialization program.

interchange port are enabled. Interrupt vectors are loaded in the primary-bus RAM of the processor to point to the interrupt service routines. A communication record register is cleared for PC-PU communication. The flash pattern of the LEDs of the status display port is set up, the external and internal interrupt flag registers are cleared, and the processor interrupt is enabled. After all the above operations, the processor goes into the waiting state.

5.4.2 Monitors and Service Routines

It can be considered that there are three levels of system software: (1) the Host Monitor, (2) the Processor Monitor, and (3) the Service Routines.

(1) Host Monitor This is at the highest level of the operating system hierarchy. It resides in the PC and provides user interface, allowing the user to download programs, commands, and data to any specified processor. Similarly, it enables the user to get access to data from any specified processor or memory locations and have them displayed on the screen. In addition, it enables the user to change the parameter values of the simulated system while in real-time operation.

Fig. 5-13 shows the function block diagram of the Host Monitor. The communication between the PC and the processor is implemented through the

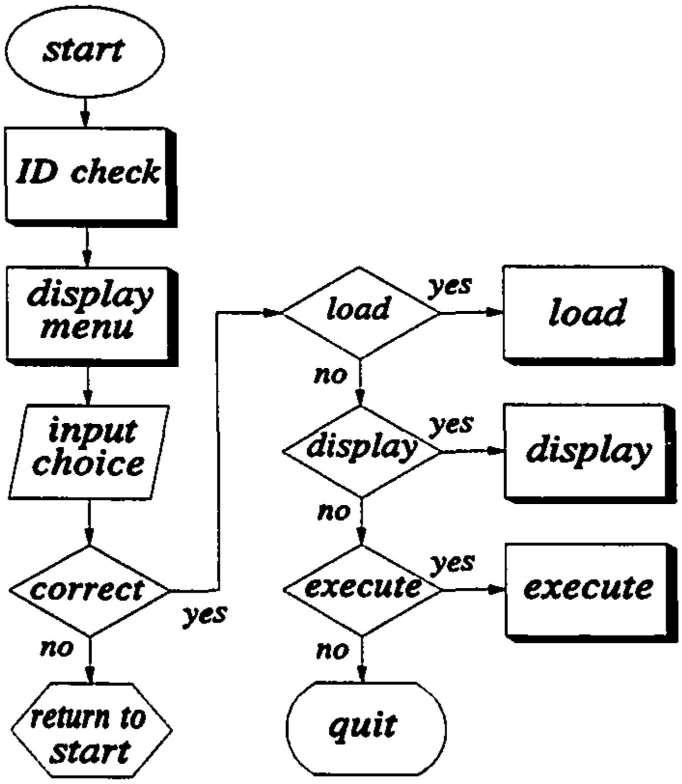


Figure 5-13 Flow chart of Host Monitor.

data interchange ports. It is written in Quick C mixed with 286 assembly language instructions. When the program is initiated, the PC checks the ID register. The ID number can be changed at this time. Then the PC displays a menu on its screen telling the user what kind of functions it has and how to do the operations. Four basic functions are available: 1) load a program or data file to the processor memory, 2) display codes in the processor memory, 3) start the processor to run a program from a specified address, and 4) quit from the parallel port communication program. When one function is completed, the PC can automatically start the next one for ID check and function selection. If a wrong operation is made, the PC can detect it, display the error on the screen and return to the origin of the program.

(2) Processor Monitor This program resides in the EPROMs of the processors. Its main function is to coordinate the information transferred between the PU and the PC. In the waiting state, the monitor is ready to receive commands from the Host Monitor. The Processor Monitor decodes the commands from the Host Monitor and directs program execution to the appropriate service routine(s) to implement the commands by using interrupts.

Fig. 5-14 shows the flow chart of the Processor Monitor. The program is triggered by External Interrupt 3 (INT3) of the processor. When the program is started, the context-save is executed at the beginning. The interrupt status register is read to find out the source of the interrupt. There are three kinds of PC interrupts which can be served for PC-PU communication at this point: to

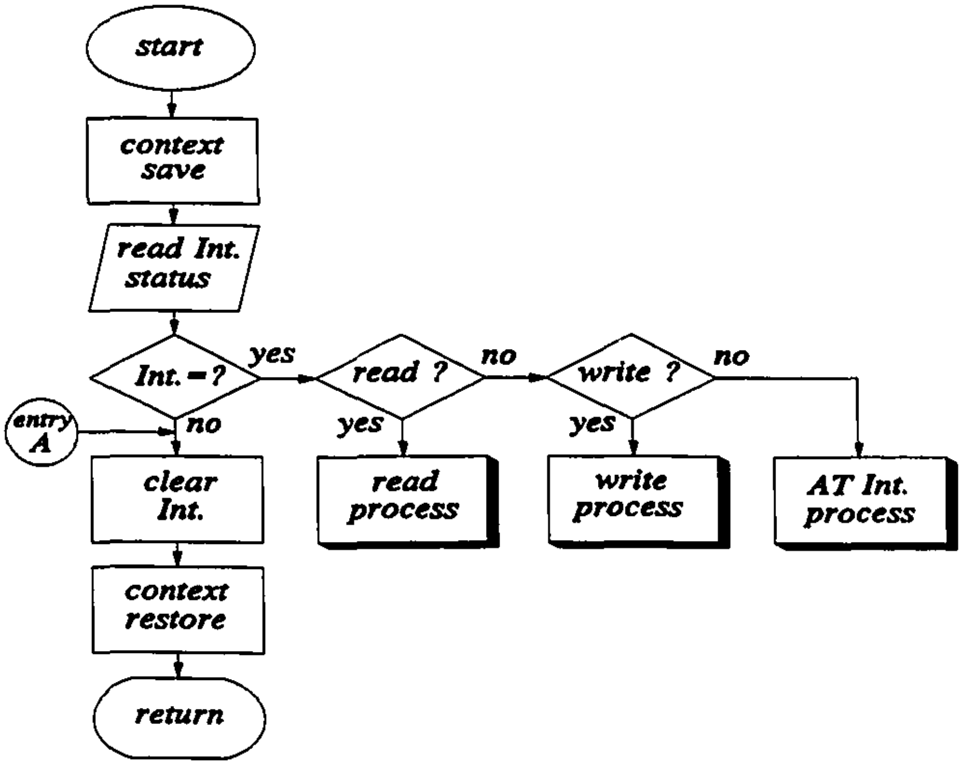


Figure 5-14 Flow chart of Processor Monitor.

read the data interchange ports, to write the data interchange ports and to interrupt the processor directly. No matter which interrupt has been served, the processor will go to the entry A after the interrupt service. Then the PC will clear the external and internal interrupt flag registers, execute the context-restore, and return to the program which executes the LED flashing.

(3) Service Routines The functions of the service routines include: downloading and uploading of program and data, initialization of program execution, data display and so on. These service routines cooperate with the Processor Monitor to execute the command from the Host Monitor. In addition,

the system has other service routines such as data acquisition from the A/D converters, waveform generation from D/A converters and other data transfer.

5.4.3 Interrupt Management

Interrupt vectors In order to provide programming flexibility, each hardware interrupt has an address stored in the interrupt vector area of the EPROM to transfer program execution to a pre-specified location in the program RAM. During system initialization, additional branch instructions are loaded in the program RAM to point to the address of the desired interrupt service routine. Thus, whenever an interrupt occurs, the program execution always jumps first to the address on the EPROM, then to the address in RAM, and finally to the address of the service routine. During real-time operation, the branch instructions in RAM can be changed so that other interrupt service routines can be executed.

Interrupt operation The software of interrupt service is designed around the system hardware. Fig. 5-15 shows the flow chart of the interrupt service routine for External Interrupt 2 (INT2). When the interrupt is invoked, the service routine is activated and all the other interrupts are masked. At the beginning, the context-save is necessary. The processor reads the interrupt status from the status port, from which it can identify the interrupt source. If more than one interrupt occur at the same time, the order at which the interrupts are serviced is based on tests described in the flow-chart of Fig. 5-15. The service priority is assigned through the software. When the processor finds its interrupt source, it turns to run the corresponding service routine. There are four different service routines for the four PUs. After the service, the processor clears the external and internal interrupt flag registers, executes the context-restore and returns to the main program. When the processor returns from the interrupt service routine, its interrupt is enabled. The processor is ready to respond to other interrupt requests.

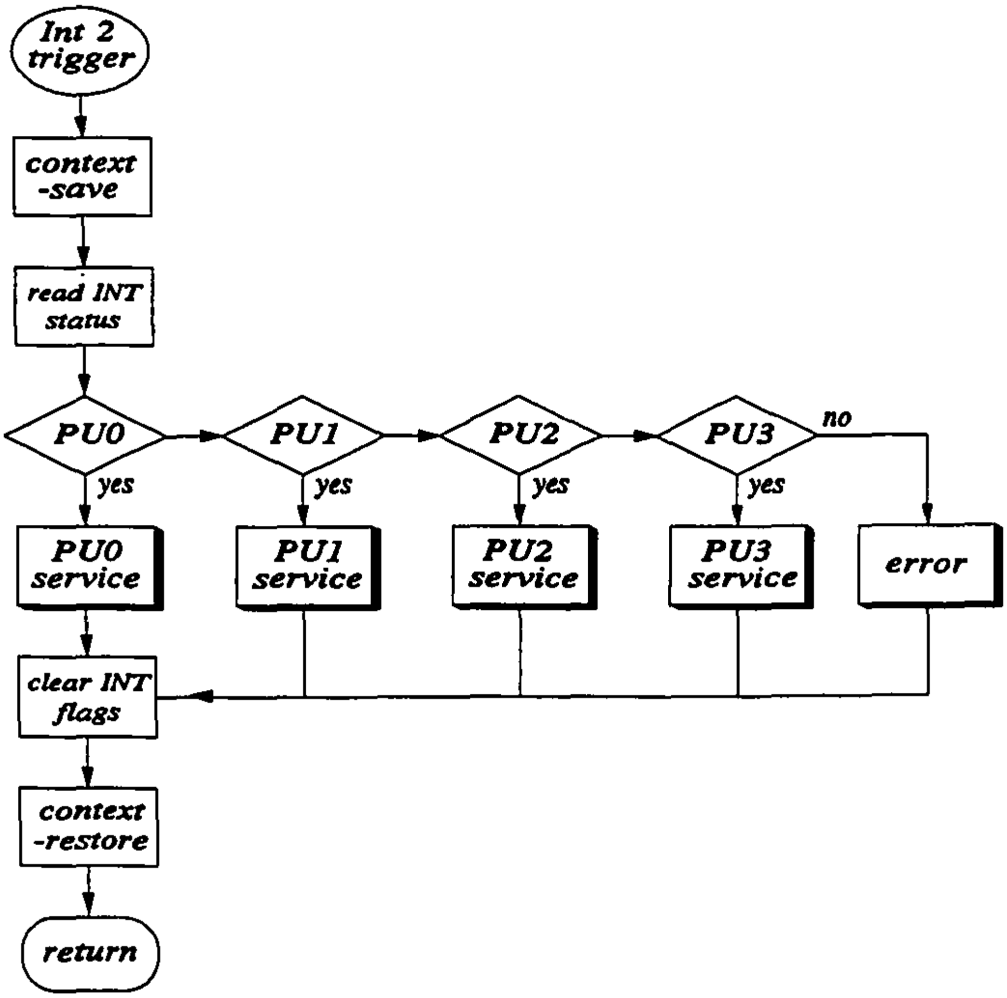


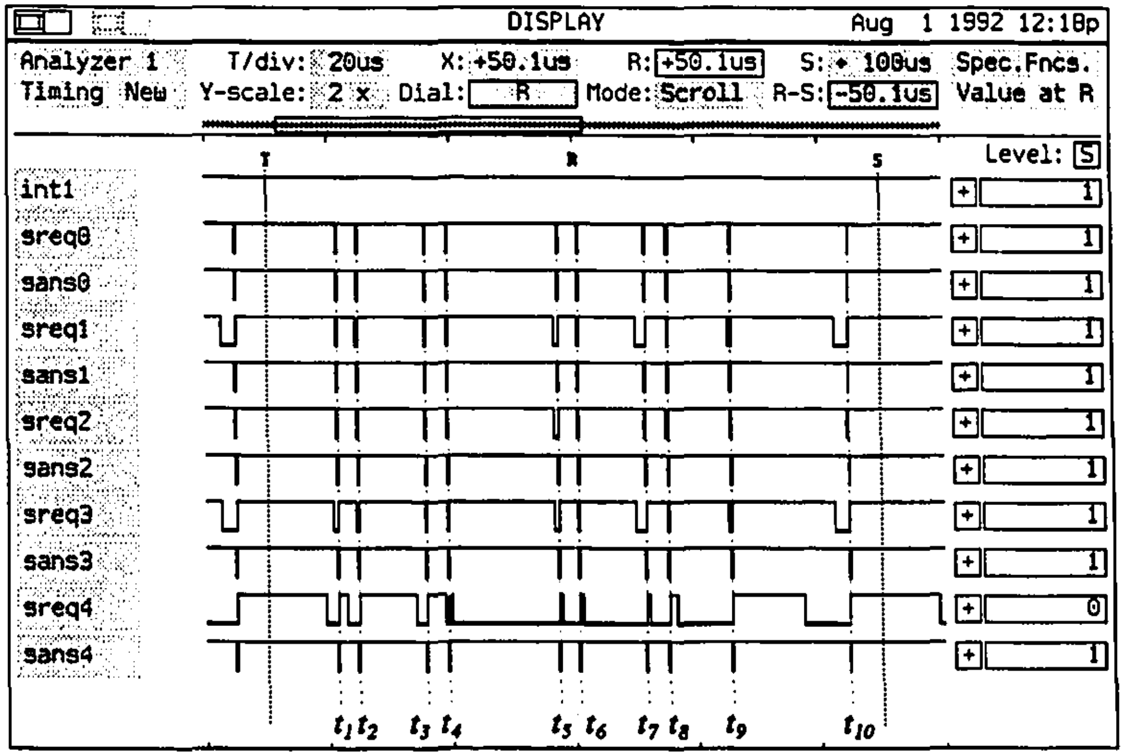
Figure 5-15 Flow chart of interrupt service routine.

5.5 SOME TEST RESULTS

Many tests have been made on the hardware through test programs which have been designed to verify its proper operation. These include: all the local memory read/write, global memory read/write and arbiter logic, interrupt bus, synchronization circuit, A/D and D/A converters, special purpose timers, control and status ports, cross board communication and PC-PU communication. The tests have been conducted with the help of a logic analyzer (PM 3585). The results of two tests are described in the following.

5.5.1 Test of Synchronization

Fig. 5-16 shows a timing diagram from the logical analyzer display for interlock operation of the processors from the application example in **Chapter 7.** The label "sreqi" corresponds to the synchronization request of PUi, which is connected to the XF0 of the processor i. The label "sansi" corresponds to the synchronization acknowledgement of PUi, which is connected to the XF1 of the processor i. If and only if all the processors send the requests for synchronization from "sreqi", then the acknowledgements of "sansi" will be generated at the same time. In the diagram, the low levels of the "sreqi" imply that the processors are waiting for synchronization and the high levels mean that the processors are busy computing. The low level pulses of the "sansi" at t_1 to t_{10} are the synchronization moments of the processors. For example, the "sreq4" in the timing diagram shows



DISPLAY Aug 1 1992 12:18p

Analyzer 1 T/div: 20us X: +50.1us R: +50.1us S: + 100us Spec.Fncs.
Timing New Y-scale: 2 x Dial: R Mode: Scroll R-S: -50.1us Value at R

Level: S

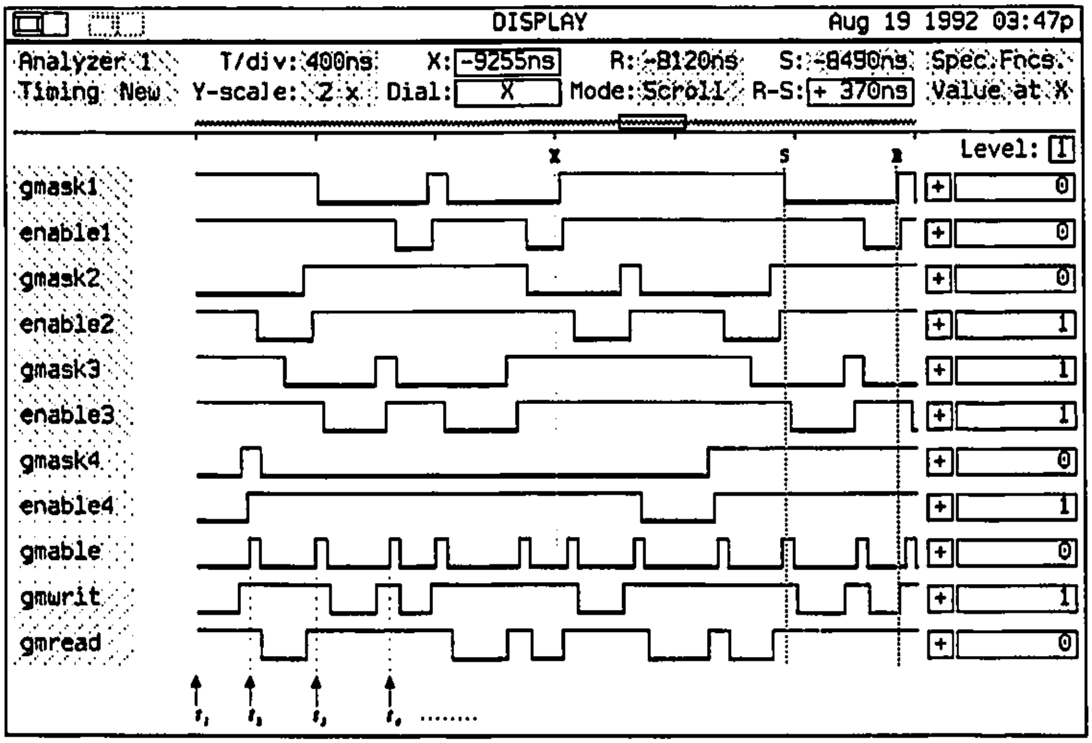
Figure 5-16 Timing diagram of interlock operation recorded by a logic analyzer (PM 3585).

that processor 4 has less computational work. It always finishes its computation before the others and waits for the synchronization acknowledgement from "sans4".

5.5.2 Test of Arbiter

The test programs are designed such that each processor writes different words to different memory locations in a pre-specified block for it, and then read them back for comparison. Each processor is allowed access to the global memory for one read or one write at a time. If an error occurs, the corresponding processor will exit the test program and signal the failure by flashing the LEDs. Because the four processors use the global memory intensively, most global memory requests from a processor will have competition from the others.

Fig. 5-17 is a timing diagram of the arbiter circuit for the global memory. In the timing diagram, the "gmaski" represents the global memory request, the "enablei" represents the permission for the processor to access the global memory, the "gmable" is the enable signal of the global memory, and the "gmwrite" and the "gmread" are the write and read signals of the global memory, respectively. They are all active low.



DISPLAY Aug 19 1992 03:47p

Analyzer 1 T/div: 400ns X: -9255ns R: -8120ns S: -8490ns Spec. Fncs.
Timing New Y-scale: 2 x Dial: X Mode: Scroll R-S: + 370ns Value at X

Level: 1

gmask1 0
enable1 0
gmask2 0
enable2 1
gmask3 1
enable3 1
gmask4 0
enable4 1
gmable 0
gmwrit 1
gmread 0

Figure 5-17 Timing diagram of arbiter circuit.

At time t_1 (the beginning of the timing diagram), the access of the global memory is granted to processor 4, because only "enable4" is enabled while "gmask" is active. At this time, it can also be seen that it is a write operation, because "gmwrite" is also active. At time t_2, only "gmask2" is active, the access to the global memory is granted to processor 2. It is a read operation. At time t_3, "gmask3" and "gmask4" are both active, only processor 3 is granted access to the global memory because it has the higher priority.

As shown in this example, the access time of the processors are separated by the arbiter circuit, no comparison error is found.

5.6 CONCLUSION

Two types of hardware digital signal processing modules have been designed, which can be connected together easily to form an Extensible Modular Multiprocessor System of as many modules as needed to within practical limits. Some basic software for the system has been written, which can help the user to use the system in control and simulation purposes [71]. A prototype of 3 modules consisting of 5 high-speed digital signal processors have been built. The test results in **Chapter 7** demonstrate that the system has been designed and built properly. In general, it can be used for real-time control and simulation in power electronics and power systems.

CHAPTER
SIX

CAPACITOR BREAK-POINT PARTITIONING

6.1 INTRODUCTION

The goal expressed previously is that the tests on the Extensible Modular Multiprocessor System described in **Chapter 5** should be combined with research and development on some challenging problems. The challenges are found in digitizing the Transient Network Analyzer (TNA). Presently, several electric utilities, which in the past had depended on the analog Transient Network Analyzers [23, 24] as their design and planning tools, have projects in increasing the digital content of their installations by using the multiprocessors to replace the op-amp based models of turbo-generators, governors, excitation systems, etc. The Manitoba HVdc Research Centre [25-33] is by far the most advanced, having developed its own parallel hardware microprocessor architecture. On the software side, the ElectroMagnetic Transient Program (EMTP) [38, 39] is being adopted as the basis of modelling the power system. Other research groups depend on commercial hardware such as the transputers [89]. Certain research groups [34, 90] prefer to use the digital models of the turbo-generators (after D/A conversion and power amplification) to drive their existing installation of transmission lines which are hardware based ladder networks of capacitors and inductors.

The need to convert the digital turbo-generator outputs by D/A converter [34, 68] to drive the analog transmission lines high-lights the nature of an important challenge-which is the knowledge of how to break up of the multi-turbo-generator power system into parcels so that each parcel can be numerically integrated by microprocessors concurrently.

The EMTP program of Professor Dommel also uses the transmission lines as the means of partitioning the power system [38, 39]. The transmission lines are modelled as ideal time delay elements and the time-delays enable the power system equations to be broken up systematically. The method suffers from the disadvantage that the transmission line lengths must be longer than twice the length for the electromagnetic wave to travel in the time Δt, the increment of the numerical integration.

Recognizing that there is the need for alternative methodologies in partitioning the power network, this chapter proposes a new approach based on using the capacitors which abound in the power network as the natural break-points. After clarifying the mathematical preliminaries, the method is implemented in **Chapter 7** to simulate a small power system to validate the method.

It is worth repeating that capacitors abound in the power systems but they are almost invariably neglected in the mathematical models for good reasons. Firstly, this is because they are usually fairly small compared with the inductive elements and their omissions have negligible effects on most phenomena of interests to the power engineers. Secondly, it would be time consuming to include them in numerical integration in the era of the single-processor.

The promise of the abundance of real-time computing power in the multiprocessor era requires a reassessment of the mathematical models to represent the power system. The additional time required to integrate the equations of the capacitors may dwarf in significance when compared with naturalness and convenience in programming the multiprocessors so that they represent the turbo-generators on a one-to-one basis.

In this chapter, **Section 6.2** describes the principle of the Capacitor Break-Point Partitioning method for power systems. **Section 6.3** develops the partitioned modules which represent different components of the power systems. The module of the turbo-generator is described in detail. In fact, the generator equations have to be reformulated so that they are amenable to the Capacitor Break-Point

Partitioning method and so that the numerical integrations are efficiently computed. Passing references are given to the modules of the transmission lines, the load bus, etc. **Section 6.4** is a discussion of the capacitor size in simulations and **Section 6.5** concludes this chapter.

6.2 CAPACITOR BREAK-POINT PARTITIONING

6.2.1 Coupled Network Equations

The modelling of the utility networks by R, L elements pre-dates the computer era and is necessary even in the main-frame computer era because the system dimensionality must be kept as low as possible in order to minimize computation time. The example of a small network shown in Fig. 6-1 is used here to highlight the desirability of Capacitor Break-Point Partitioning. The currents are solved by the numerical integration of:

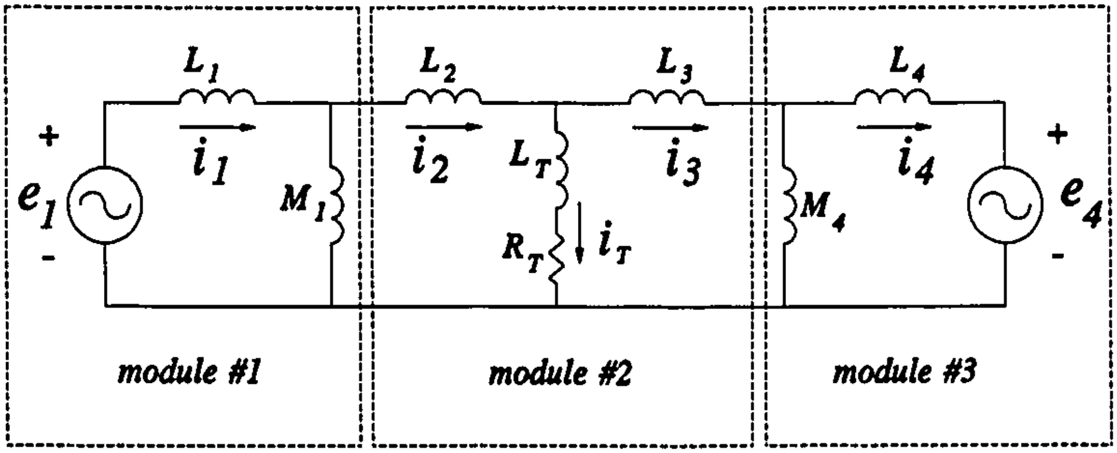


Figure 6-1 Small network.

$$\frac{d\underline{i}}{dt} = [L]^{-1}\{\, \underline{e} - [R]\underline{i} \,\} \tag{6-1}$$

where $\underline{i}$ and $\underline{e}$ are the current and *emf* vectors (4×1), [L] and [R] are the (4×4) inductance and the resistance matrices respectively.

The kth row of Eq. (6-1) is of the form:

$$\frac{di_k}{dt} = a_{k1}e_1 + a_{k4}e_4 + \sum_{j=1}^{4} b_{kj}i_j \tag{6-2}$$

where a_{kj} and b_{kj} are constant coefficients from $[L]^{-1}$ and $-[L]^{-1}[R]$.

Since the numerical integration of i_k involves a knowledge of all the states, $i_1 \ldots i_4$, and the inputs, e_1, e_4, this information must be communicated to the integrator performing the numerical integration of i_k. Suppose the numerical integration is to be performed concurrently by 3 computational modules (of **Chapter 5**) so that each module is responsible for a part of the network as shown in Fig. 6-1, one sees that the information from module *#3* must be relayed by way of module *#2* to module *#1* and vice-versa. For this reason, it is very desirable to find a method of formulating the equations so that information transfers for the concurrent computation are between the contiguous modules.

6.2.2 Natural Partitioning

The physical world is naturally amenable to the partitioning suitable for the extensible architecture. The generators, the transformers, the transmission lines, etc. are "aware" of each other only through the information at their immediate terminals. Thus the requisite information should be available from the contiguous neighbours.

Ref. [91] has shown that associated with the inductive voltages, which are represented by the inductances L and M, there are the accompanying electrostatic voltages which would have been represented by "stray" capacitors. The "stray" capacitors are very small and should rightfully be neglected unless one is interested in the study of the steep switching surges which have caused dielectric failures in transformer [92] and motor windings [93].

6.2.3 Capacitor Break-Point Partitioning

Taking the clue from nature, a capacitor C_T is added across the load as shown in Fig. 6-2. The capacitor C_T may be the capacitance of the transmission line and the load. It may also have already existed for power factor correction or

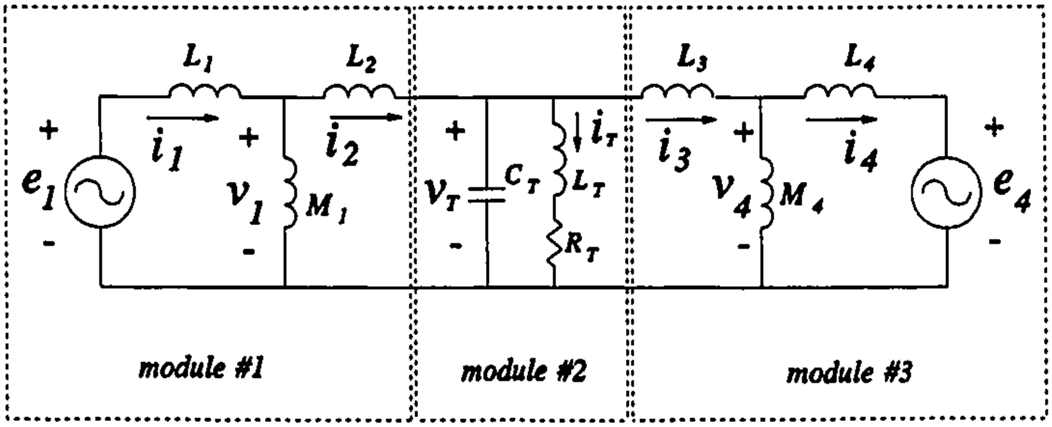


Figure 6-2 Capacitance added to the circuit of Fig. 6-1 for partitioning.

for voltage support purposes [67]. In the latter situations, C_T is by no means small but it is normally omitted in the modelling for main-frame computation because the interest is to keep the system order low.

With the addition of C_T, it is possible to partition the computation burden to 3 modules, each module (*#1*, *#2* and *#3* in Fig. 6-2) being required to exchange input/output information with its contiguous neighbours as listed in Table 6-I.

Table 6-I Information Exchanges

	State Variables	Inputs	Outputs
module *#1*	i_1, i_2	v_T	i_2
module *#2*	i_T, v_T	i_2, i_3	v_T
module *#3*	i_3, i_4	v_T	i_3

The equations which are numerically integrated are:

module *#1*

$$L_1 \frac{di_1}{dt} = e_1 - v_1 \tag{6-3}$$

$$L_2 \frac{di_2}{dt} = v_1 - v_T \tag{6-4}$$

where v_1, the voltage across M_1 is:

$$v_1 = \frac{\dfrac{e_1}{L_1} + \dfrac{v_T}{L_2}}{\dfrac{1}{M_1} + \dfrac{1}{L_1} + \dfrac{1}{L_2}} \tag{6-5}$$

module #2

$$C_T \frac{dv_T}{dt} = i_2 - i_3 - i_T \tag{6-6}$$

$$L_T\, d\frac{i_T}{dt} = v_T - R_T\, i_T \tag{6-7}$$

module #3

$$L_3\, d\frac{i_3}{dt} = v_T - v_4 \tag{6-8}$$

$$L_4\, d\frac{i_4}{dt} = v_4 - e_4 \tag{6-9}$$

where v_4, the voltage across M_4 is:

$$v_4 = \frac{\dfrac{v_T}{L_3} + \dfrac{e_4}{L_4}}{\dfrac{1}{M_4} + \dfrac{1}{L_3} + \dfrac{1}{L_4}} \tag{6-10}$$

The simple example of Fig. 6-2 illustrates the general principle of using capacitors in the network as break-points to achieve partitioning. Since each module requires information from its contiguous neighbours only, any number of modules can be joined together to simulate the power system. The local selfsufficiency of information means that as the power system grows one needs only to add on computational modules to simulate the growth.

6.3 POWER SYSTEM MODULES

There are many possible schemes to partition the electric power utility systems. Based on the architecture and the characteristics of the multiprocessor system described in **Chapter 5**, it is envisaged that each multiprocessor module has the ports to be connected to four neighbours. One or more multiprocessor module as a group will represent a sub-system of the utility network and the Capacitor Break-Point Partitioning of **Section 6.2** will be used to join the sub-systems together.

6.3.1 Turbo-Generator Module

The turbo-generator module includes the multi-inertia torsional shaft system, the governor regulator system, the voltage regulator-field excitation system, the power system stabilizer (PSS), the output transformers and so on. Their mathematical models [94] are presented in **Chapte~ 7**.

The focus here is on the treatment of the electrical equations of the generator because it is where the difficulty lies. This is because the large integrated power system is based on connecting the terminals of the generators to the transmission line network. The transmission line interconnection is the cause of coupled linear equations. The objective is to show how the Capacitor Break-Points method can be integrated with the generator equations for the desired partition.

Fig. 6-3 shows the limits of responsibility of each turbo-generator module with respect to its 4 possible contiguous neighbours. At time t_k, it receives as inputs the voltages $v_{aj}(t_k)$, $v_{bj}(t_k)$, $v_{cj}(t_k)$, $j = 1,...4$ from its neighbours and transmits the currents outputs $i_{aj}(t_k)$, $i_{bj}(t_k)$, $i_{cj}(t_k)$, $j = 1,...4$ to the neighbours. As shown in Fig. 6-3, the line impedances, $R_j + j\omega L_j$, $j = 1,..4$, between the transformers and

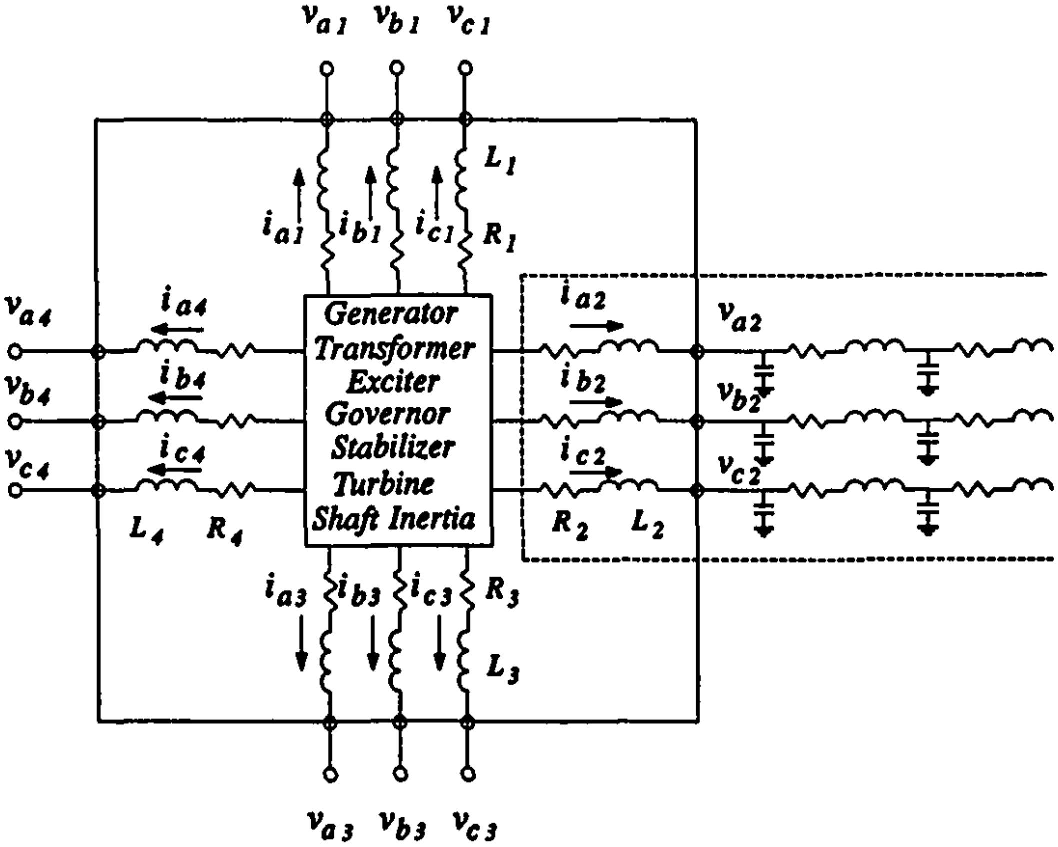


Figure 6-3 Turbo-generator module: electrical circuit connections.

the capacitor nodes belong to the turbo-generator module.

The 3-phase voltages (v_{aj}, v_{bj} and v_{cj}) are transformed to the *o-d-q* frame voltages [22, 87].

$$\begin{bmatrix} v_{oj} \\ v_{dj} \\ v_{qj} \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \cos\delta & \sin\delta \\ \frac{1}{\sqrt{2}} & \cos(\delta-\frac{2\pi}{3}) & \sin(\delta-\frac{2\pi}{3}) \\ \frac{1}{\sqrt{2}} & \cos(\delta-\frac{4\pi}{3}) & \sin(\delta-\frac{4\pi}{3}) \end{bmatrix} \begin{bmatrix} v_{aj} \\ v_{bj} \\ v_{cj} \end{bmatrix} \tag{6-11}$$

where the torque angle $\delta(t_k)$ has been solved from the mechanical equations:

$$\frac{d\delta}{dt} = \omega \tag{6-12}$$

$$2H \frac{d\omega}{dt} = T_{gt} - T_e \tag{6-13}$$

H is the rotor inertia constant, T_{gt} is the mechanical torque acting on the rotor and T_e is the electromagnetic torque of the generator (see Eq.(6-25)).

The *d*- and *q*-axis equivalent circuits of Fig. 6-4 (a) and (b) show the locations of the input voltages v_{dj}, v_{qj} ($j=1, 2, 3, 4$). It should be emphasized that the parameters R_j, L_j ($j=1, 2, 3, 4$) between the transformer mutual inductance M and the voltages v_{dj}, v_{qj} are considered to be part of parameters of the turbo-generator module. The transformer resistance and leakage inductances are lumped together with r, l_d and l_q of the generator armature circuit.

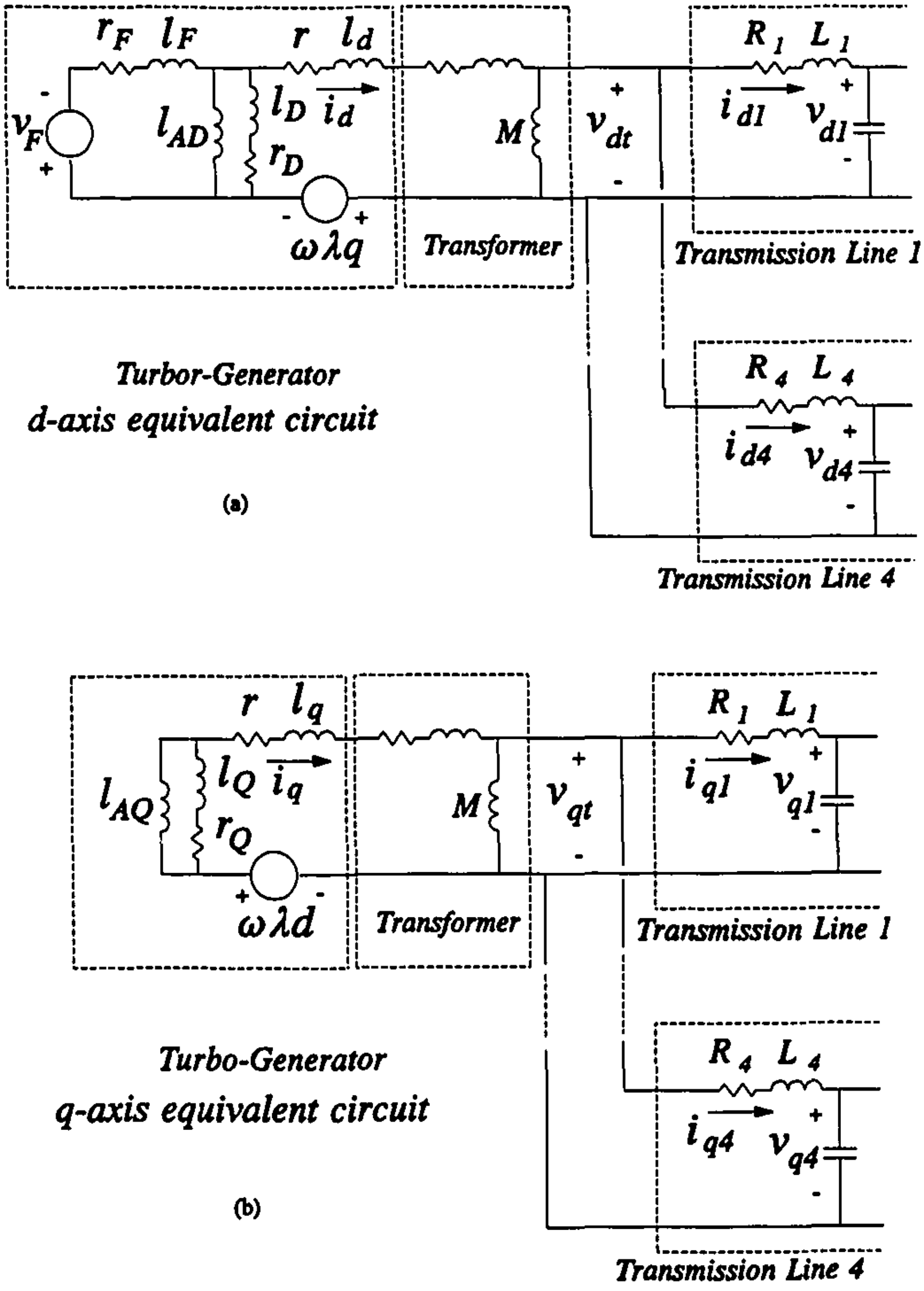


Figure 6-4 Turbo-generator module: generator and transformer equivalent circuit. (a) *d*-axis model, (b) *q*-axis model.

6.3.2 Generator Equations

For numerical integration, one of the most efficient formulation is the flux-linkage state space model of the generator taken from [94, 95]. A familiarity of the generator equations is assumed and the objective here is to organize them for efficient computation by the Capacitor Break-Point Partitioning method. Using the symbols of Ref. [94], the subscripts (d, q) refer to the armature, (D, Q) to the amortisseur windings, F to the field windings, (AD, AQ) to the magnetization inductances of the generator.

***d*-axis**

In the d-axis, the state variables are the flux linkages. λ_d, λ_f and λ_D. One evaluates the magnetization flux linkage λ_{AD}, where

$$\lambda_{AD} = L_{MD} \left[\frac{\lambda_d}{l_d} + \frac{\lambda_F}{l_F} + \frac{\lambda_D}{l_D} \right] \tag{6-14}$$

and the inductance L_{MD} is obtained from the formula:

$$\frac{1}{L_{MD}} = \frac{1}{L_{AD}} + \frac{1}{l_d} + \frac{1}{l_F} + \frac{1}{l_D} \tag{6-15}$$

The numerical integration is performed on the 3 equations.

$$\frac{d\lambda_d}{dt} = \frac{r}{l_d}\left(\lambda_{AD} - \lambda_d\right) - \omega\lambda_q - v_{dt} \tag{6-16}$$

$$\frac{d\lambda_F}{dt} = \frac{r_F}{l_F}\left(\lambda_{AD} - \lambda_F\right) + e_f \tag{6-17}$$

$$\frac{d\lambda_D}{dt} = \frac{r_D}{l_D}(\lambda_{AD} - \lambda_D) \tag{6-18}$$

In Eq. (6-16), ω is a solution from Eq. (6-13). The voltage v_{dt} is the voltage across the transformer and is still an unknown until is solved in Eq. (6-29). In Eq. (6-17), e_f is the output of the field excitation system which will be described in Section 7.2.4. The field excitation equations are described in Ref. [94] and since they can be represented in block diagrams, they present no special problems as Ref. [34] has shown.

q-axis

The q-axis state variables are λ_q and λ_Q. From the state variables, one evaluates the magnetization flux linkage λ_{AQ}, where

$$\lambda_{AQ} = L_{MQ}\left[\frac{\lambda_q}{l_q} + \frac{\lambda_Q}{l_Q}\right] \tag{6-19}$$

where the inductance L_{MQ} is obtained from:

$$\frac{1}{L_{MQ}} = \frac{1}{L_{AQ}} + \frac{1}{l_q} + \frac{1}{l_Q} \tag{6-20}$$

The numerical integration is performed on the 2-equations:

$$\frac{d\lambda_q}{dt} = \frac{r}{l_q}\left(\lambda_{AQ} - \lambda_q\right) + \omega\lambda_d - v_{qt} \tag{6-21}$$

$$\frac{d\lambda_Q}{dt} = \frac{r_Q}{l_Q}(\lambda_{AQ} - \lambda_Q) \tag{6-22}$$

In Eq. (6-21), the voltage v_{qt} is the voltage across the transformer and it is the objective to find its solution which is given in Eq. (6-30).

Armature Currents

The armature currents are:

$$i_d = \frac{\lambda_d - \lambda_{AD}}{l_d} \tag{6-23}$$

$$i_q = \frac{\lambda_q - \lambda_{AQ}}{l_q} \tag{6-24}$$

Electromechanical Torque

The electromechanical torque is:

$$T_e = \lambda_d i_q - \lambda_q i_d \tag{6-25}$$

and is used in the numerical integration of Eq. (6-13).

Transformer Voltages v_{dt}, v_{qt}

The transformer voltages v_{dt} and v_{qt} have to be known before Eq. (6-16) and (6-21) can be integrated. Their closed form solutions can be obtained by some manipulations of the equations. From Fig. 6-4(a)

$$\begin{aligned} v_{dt} &= M\frac{d}{dt}i_{md} \\ &= M\left[\frac{d}{dt}i_d - \frac{d}{dt}\sum_{j=1}^{4} i_{dj}\right] \end{aligned} \tag{6-26}$$

In each of the output lines

$$\frac{di_{dj}}{dt} = \frac{v_{dt} - v_{dj} - R_j i_{d_j}}{L_j} \qquad (j=1,..4) \tag{6-27}$$

Substituting Eq. (6-14) into Eq. (6-23) and on differentiation

$$\frac{di_d}{dt} = \frac{1}{l_d}\frac{d\lambda_d}{dt} - \frac{L_{MD}}{l_d}\frac{d\lambda_d}{dt} - \frac{L_{MD}}{l_F}\frac{d\lambda_F}{dt} - \frac{L_{MD}}{l_D}\frac{d\lambda_D}{dt} \tag{6-28}$$

On substituting Eqs. (6-27) and (6-28) into Eq. (6-26) and thereafter substituting Eqs. (6-16), (6-17) and (6-18) it can be shown that:

$$v_{td} = d_1 \omega \lambda_q + d_2 \lambda_{AD} + d_3 \lambda_d + d_4 \lambda_D + d_5 \sum_{j=1}^{4} \frac{v_{dj} + R_j i_{dj}}{L_j} + d_6 v_F + d_7 \lambda_F \tag{6-29}$$

Defining

$$D = \frac{1}{M} - \frac{L_{MD}}{l_d^2} + \frac{1}{l_d} + \sum_{j=1}^{4} \frac{1}{L_j}$$

the coefficients in Eq. (6-29) are:

$$d_1 = \frac{L_{MD}/l_d - 1}{l_d D}$$

$$d_2 = \frac{\dfrac{r}{l_d}\left(\dfrac{L_{MD}}{l_d} - 1\right) + \dfrac{L_{MD} r_F}{l_F^2} + \dfrac{L_{MD} r_D}{l_D^2}}{l_d D}$$

$$d_3 = \frac{-\dfrac{r}{l_d}\left(\dfrac{L_{MD}}{l_d} - 1\right)}{l_d D}$$

$$d_4 = \frac{-\ L_{MD} r_D \,/\, l_D^2}{l_d D}$$

$$d_5 = \frac{1}{D}$$

$$d_6 = \frac{L_{MD} \,/\, l_F}{l_d D}$$

$$d_7 = \frac{-\ L_{MD} r_F \,/\, l_F^2}{l_d D}$$

Likewise, by manipulation of the q-axis equations it can be shown that:

$$\begin{aligned} v_{tq} &= q_1 \omega \lambda_d + q_2 \lambda_{AQ} + q_3 \lambda_q + q_4 \lambda_Q \\ &\quad + q_5 \sum_{j=1}^{4} \frac{v_{qj} + R_j i_{qj}}{L_j} \end{aligned} \tag{6-30}$$

Defining

$$Q = \frac{1}{M} - \frac{L_{MQ}}{l_q^2} + \frac{1}{l_q} + \sum_{j=1}^{4} \frac{1}{L_j}$$

the coefficients in Eq. (6-30) are:

$$q_1 = \frac{-\ (L_{MQ} \,/\, l_q - 1)}{l_q Q}$$

$$q_2 = \frac{\frac{r}{l_q}\left(\frac{L_{MQ}}{l_q} - 1\right) + \frac{L_{MQ} r_Q}{l_Q^2}}{l_q Q}$$

$$q_3 = \frac{-\frac{r}{l_q}\left(\frac{L_{MQ}}{l_q} - 1\right)}{l_q Q}$$

$$q_4 = \frac{-L_{MQ} r_Q / l_Q^2}{l_q Q}$$

$$q_5 = \frac{1}{Q}$$

Generator Input/Outputs

At time t_k, the state variables are $\lambda_d(t_k)$, $\lambda_q(t_k)$, $\lambda_F(t_k)$, $\lambda_D(t_k)$, $\lambda_Q(t_k)$, $\omega(t_k)$, $\delta(t_k)$, and $i_{dj}(t_k)$, $i_{qj}(t_k)$, $j = 1, 2...4$. The internal input from the voltage regulator exciter subsystem $e_f(t_k)$ in Eq. (6-17) is the solution of Eq. (7-9). Likewise from the generator-turbine sub-system, the internal input T_{gt} in Eq. (6-13) is the solution of Eq. (7-11). The inputs from outside the turbo-generator module are the voltages $v_{aj}(t_k)$, $v_{bj}(t_k)$, $v_{cj}(t_k)$, $(j = 1, 2,..4)$.

From the inputs, one can compute $v_{dj}(t_k)$, $v_{qj}(t_k)$ using Eq. (6-11), and v_{dt}, v_{qt} using Eqs. (6-29) and (6-30). By numerical integration of Eqs. (6-12), (6-13), (6-16), (6-17), (6-18) (6-21), (6-22), (6-27) and a corresponding set of equations for the q-axis currents, one obtains the states at t_{k+1}: $\omega(t_{k+1})$, $\delta(t_{k+1})$, $\lambda_d(t_{k+1})$, $\lambda_F(t_{k+1})$ $\lambda_D(t_{k+1})$, $\lambda_q(t_{k+1})$, $\lambda_Q(t_{k+1})$, $i_{dj}(t_{k+1})$, $i_{qj}(t_{k+1})$, $(j = 1, 2..4)$. Using the inverse transformation of Eq. (6-11) for $\delta(t_{k+1})$, one obtains the output currents $i_{aj}(t_{k+1})$, $i_{bj}(t_{k+1})$, $i_{cj}(t_{k+1})$, $(j = 1, 2..4)$, which are transmitted to the contiguous neighbours.

Governor, Field Excitation and Power System Stabilizer

The turbo-generator module includes the governor system, the field excitation system and the power system stabilizer. They are modelled in block-diagrams in Figs. 7-3, 7-4 and 7-5 [94] and they are treated more fully in **Chapter 7**. Their inclusion means that the block diagrams have to be translated into ordinary differential equations to be numerically integrated. This adds to the volume of real-time computation. Otherwise, they do not present any difficulty.

6.4 INTEGRATION OF PARTITIONED MODULES

Because of research time constraint, the scope of module development in this thesis is limited to the turbo-generator module. In order to complete the modelling of the power system for the digital Transient Network Analyzer, partitioned modules must be developed to represent: (1) the transmission lines, (2) the loads, (3) the static VAR compensators, (4) the converters of the high voltage direct current transmission system, (5) the phase shifters, etc.

The integration of the modules will be based on prescribing the "protocols" for the information exchanges between modules. The "protocols" will follow the steps outlined for the information exchanges betwe he turbo-generator and transmission line modules in Fig. 6-3 and described in **Section 6.3.1**. The turbo-generator module receives as inputs the voltages $v_{aj}(t_k)$, $v_{bj}(t_k)$, $v_{cj}(t_k)$, $j=1, \dots 4$ from its neighbours and transmits the current outputs $i_{aj}(t_k)$, $i_{bj}(t_k)$, $i_{cj}(t_k)$, $j=1, \dots 4$ to the neighbours. The modules which connect directly to the turbo-generator modules must have complementary inputs and outputs which have to be current and voltage quantities respectively. Thus the transmission line modules receive at one of its ports, the inputs which are currents $i_{aj}(t_k)$, $i_{bj}(t_k)$, $i_{cj}(t_k)$, and the outputs of the voltages, $v_{aj}(t_k)$, $v_{bj}(t_k)$, $v_{cj}(t_k)$.

6.5 CAPACITOR SIZE FOR PARTITIONING

The voltage across a small capacitor has a high frequency of oscillation, so that the numerical integration step-size must be kept small to satisfy spectral radius requirements. The "stray capacitances" in the power apparatus are very small. Some values cited in the literature for the winding capacitance-phase to ground are:

736 KV transformers:	*2156 pf*
rotating machinery **[93]**:	*8060 pf*

The parameters for the *765 KV* transmission lines are **[96]**:

$$R_1 = 0.019 \ \text{Ohm/mi}$$
$$X_{L1} = 0.548 \ \text{Ohm/mi}$$
$$X_{c1} = 0.128 \ \text{megaohm-mi}$$

However, the load buses invariably contain substantial capacitors for the purposes of improving voltage profiles, reducing line loading and losses by power-factor improvement. In the power transmission system itself, capacitor banks are installed for power factor correction, voltage control and stabilization, phase balancing and the handling of harmonics [67]. While these capacitor installations are too small to be included in the methodology of the single-processor, they are large enough to ensure that the integration step-size does not have to be minuscule for the multiprocessor modules.

6.6 CONCLUSION

In this chapter, the method of Capacitor Break-Point Partitioning of the power system equations has been described. In **Section 6.3.1**, the generator equations have been reformulated for the Capacitor Break-Point Partitioning and for fast numerical integration by parallel processing. In the next chapter, the principle has been adopted in the real-time simulation of a small power system made up of two turbo-generators and one load bus.

CHAPTER
SEVEN

IMPLEMENTATION OF A PROTOTYPE DIGITAL TNA

7.1 INTRODUCTION

In **Chapter 5**, the design and the implementation of an Extensible Modular Multiprocessor System have been described. In **Chapter 6**, the Capacitor Break-Point Partitioning method has been proposed for breaking up the power system equations for numerical integration by the multiprocessors. This chapter focuses on the implementation of a prototype digital Transient Network Analyzer (TNA) by the Extensible Modular Multiprocessor System to simulate a small power system. Because of time and budgetary constraints, the Extensible Modular Multi-processor System is limited to three boards: two Computational Modules and one I/O Module.

The simulated power system consists of two turbo-generators and an infinite bus. Each Computational Module in the Extensible Modular Multiprocessor System simulates one turbo-generator module. This includes the generator-turbine system, the transformer, the local load, the transmission lines, the governor system, the excitation system, and the power system stabilizer system [94]. The I/O Module computes the system load, post-processes the state variables of the turbo-generator and presents selected information for analog display. The numerical integration is implemented by the modified Euler's method [98] at a step-size of *100 μs* in real time. Assembly language is used to achieve the fast processing speed.

In this chapter, **Section 7.2** describes the simulated power system and the equations modelling the subsystems. **Section 7.3** concentrates on the implementation of the digital TNA. This includes the integration of the turbo-generator modules to form the interconnected power system, the task scheduling in the multiprocessor computer and the parallel programming for the simulation. **Section 7.4** reports on the experimental proofs of the correct operation of the TNA, which are carefully gathered from the tests on individual turbo-generator, the tests on multi-machines and the tests for the ability to predict subsynchronous resonance phenomena of the power system. **Section 7.5** concludes the chapter.

7.2 MATHEMATICAL MODEL OF SIMULATED POWER SYSTEM

The small power system used for the feasibility study of the prototype TNA is shown in Fig. 7-1. In the network, generator $\underline{e}_1$ is connected to an infinite bus

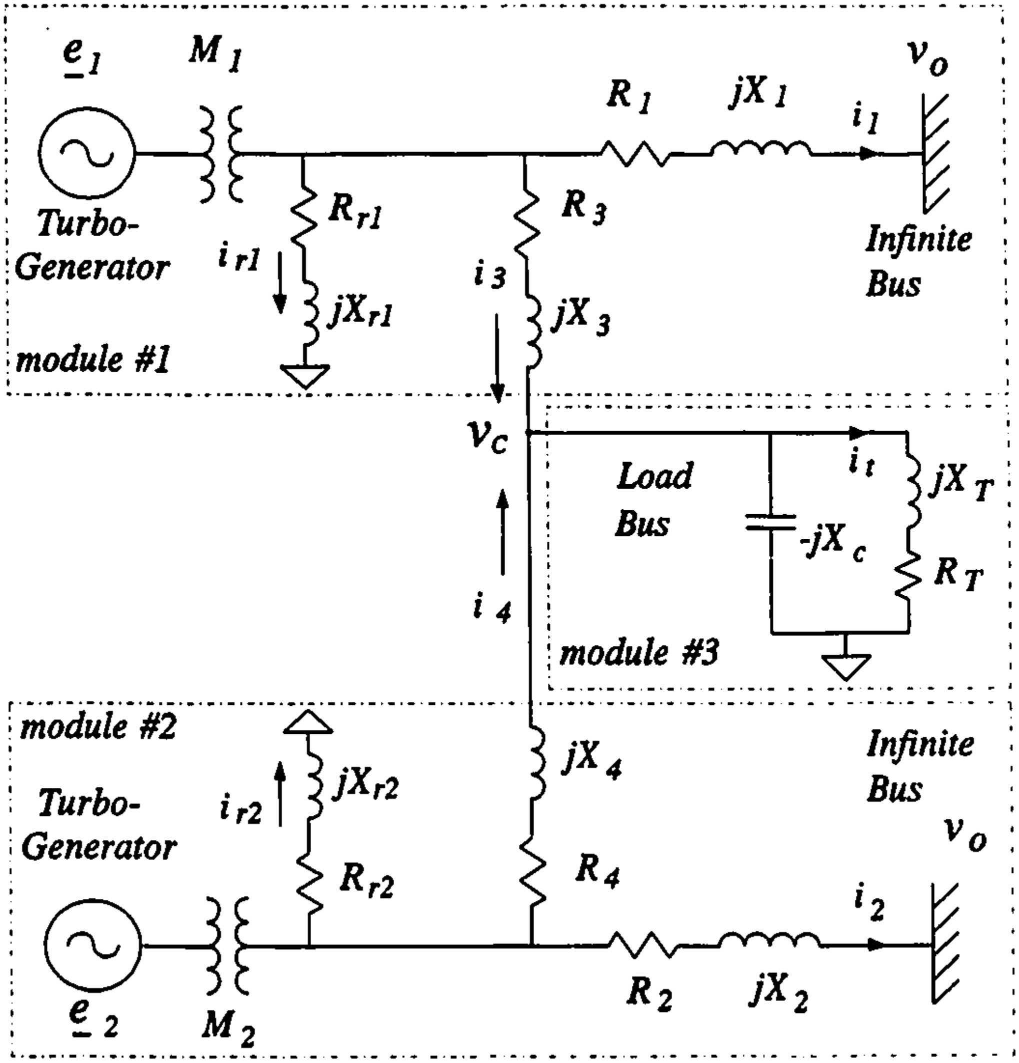


Figure 7-1 Power system for test of digital TNA.

through the transformer, M_1, and a transmission line, modelled as an impedance R_1+jX_1. Generator e_2 is connected similarly through M_2 and R_2+jX_2 to the same infinite bus. The load bus is connected between the transformer M_1 bus and the transformer M_2 bus through transmission lines modelled as R_3+jX_3 and R_4+jX_4. The capacitance in the reactance $-jX_c$ is the break-point which enables the partitioning method of Chapter 6 to be implemented. X_c includes the transmission line capacitance and the capacitance in the load. As power factor correction is widely practised, the compensating capacitor, which is not negligibly small, is also a contribution to X_c. $R_{r1}+jX_{r1}$ and $R_{r2}+jX_{r2}$ are the models of the local loads of the generator units.

7.2.1 Generator Equations

The generator equations have been presented in Section 6.3.2. and for completeness are reproduced here with slight modifications. Thus v_{dt}, v_{qt} and v_F in Eqs. (6-16), (6-17) and (6-21) are replaced by the symbols e_d, e_q and e_f in Eqs. (7-1), (7-2) and (7-3) respectively. These changes have been made to take into account of the specific network of Fig. 7-1 so that the formulae of e_d and e_q are taken from Eqs. (7-37) and (7-38). Also e_t is a solution to Eq. (7-21). The generator equations are:

$$\dot{\lambda}_d=\omega_o[e_d+\omega\lambda_q+r_a(\lambda_{md}-\lambda_d)/x_l] \tag{7-1}$$

$$\dot{\lambda}_q=\omega_o[e_q-\omega\lambda_d+r_a(\lambda_{mq}-\lambda_q)/x_l] \tag{7-2}$$

$$\dot{\lambda}_{fd}=\omega_o[e_f+r_f(\lambda_{md}-\lambda_{fd})/x_{fd}] \tag{7-3}$$

$$\dot{\lambda}_{kd} = \omega_o r_{kd} (\lambda_{md} - \lambda_{kd})/x_{kd} \tag{7-4}$$

$$\dot{\lambda}_{kq} = \omega_o r_{kq} (\lambda_{mq} - \lambda_{kq})/x_l \tag{7-5}$$

where λ_d, λ_q, λ_{fd}, λ_{kd}, λ_{kq} are the fluxes of d-axis and q-axis in the stator, field and rotor windings, e_d, e_q, e_f are the voltage sources of d-axis, q-axis in the stator and field windings, r_a, x_l, x_{ad}, x_{aq}, r_f, x_{fd}, r_{kd}, x_{kd}, r_{kq}, x_{kq} are the circuit parameters in stator, field and rotor winding, and

$$\lambda_{md} = \frac{\lambda_d/x_l + \lambda_{fd}/x_{fd} + \lambda_{kd}/x_{kd}}{1/x_{ad} + 1/x_l + 1/x_{fd} + 1/x_{kd}} ,$$

$$\lambda_{mq} = \frac{\lambda_q/x_l + \lambda_{kq}/x_{kq}}{1/x_{aq} + 1/x_l + 1/x_{kq}} .$$

7.2.2 Torsional Shaft-Inertia System

The hydro turbine-generator shaft system of Fig. 7-2 is modelled by the equations [97]:

$$\dot{\omega}_g = \frac{1}{2H_g}\left[T_{gt} - (D_{gt} + D_g)\,\omega_g + D_{gt}\,\omega_t + T_e\right] \tag{7-6}$$

$$\dot{\omega}_t = \frac{1}{2H_t}\left[T_{in} - (D_{gt} + D_t)\,\omega_t + D_{gt}\,\omega_g - T_{gt}\right] \tag{7-7}$$

$$\dot{\delta}_g = \omega_o(\omega_g - 1) \tag{7-8}$$

$$\dot{\delta}_t = \omega_o(\,\omega_t - 1) \tag{7-9}$$

where the electrical torque T_e is

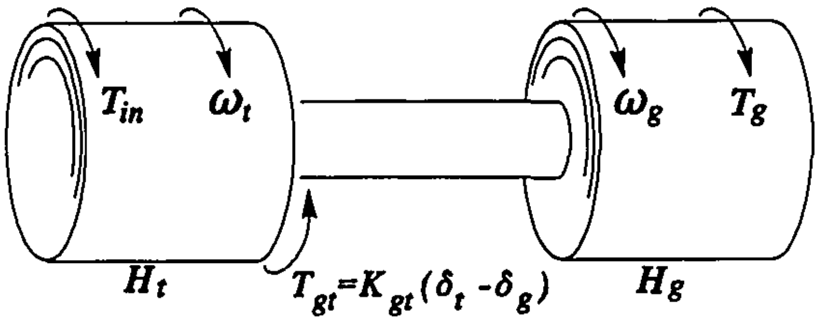


Figure 7-2 Generator-turbine system.

$$T_e = \frac{\lambda_d(\lambda_q - \lambda_{mq}) - \lambda_q(\lambda_d - \lambda_{md})}{3x_l} \quad , \qquad (7\text{-}10)$$

the shaft mechanical torque T_{gt} is

$$T_{gt} = K_{gt}(\delta_t - \delta_g) \ . \qquad (7\text{-}11)$$

The states ω_g and ω_t, δ_g and δ_t are respectively the angular velocities and the angles of the generator and the turbine, H_g, H_t, D_g, D_t, D_{gt} are the moment of inertias and the damping factors of the generator and turbine. The subscripts g and t denote generator and turbine quantities. K_{gt} is the torsional stiffness constant of the shaft. T_{in} is the turbine driving torque which is obtained from Eq. (7-16).

7.2.3 Governor System

Fig. 7-3 shows the block diagram of the governor system taken from Fig.

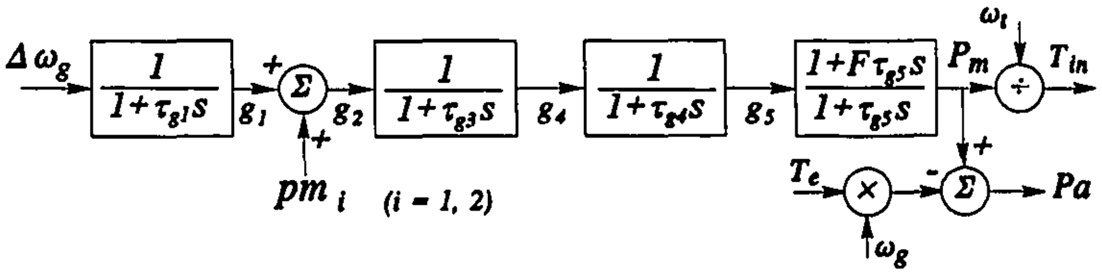


Figure 7-3 Block diagram of governor system.

D13 of Ref. [94]. From the individual blocks, the differential equations to be integrated are:

$$\dot{g}_1 = -\frac{g_1}{\tau_{g1}} + \frac{K_g}{\tau_{g1}}(\omega_g - 1) \qquad (7\text{-}12)$$

$$\dot{g}_4 = \frac{g_1}{\tau_{g3}} - \frac{g_4}{\tau_{g3}} + \frac{pm_i}{\tau_{g3}} \qquad (7\text{-}13)$$

$$\dot{g}_5 = \frac{g_4}{\tau_{g4}} - \frac{g_5}{\tau_{g4}} \qquad (7\text{-}14)$$

$$\dot{P}_m = \frac{F}{\tau_{g4}}g_4 - \left(\frac{F}{\tau_{g4}} - \frac{1}{\tau_{g5}}\right)g_5 - \frac{1}{\tau_{g5}}P_m \qquad (7\text{-}15)$$

$$T_{in} = P_m/\omega_t \ . \qquad (7\text{-}16)$$

7.2.4 Excitation System

Fig. 7-4 shows the block diagram of the excitation system taken from the Fig. D4 of Ref. [94]. From the individual blocks, the differential equations to be integrated are:

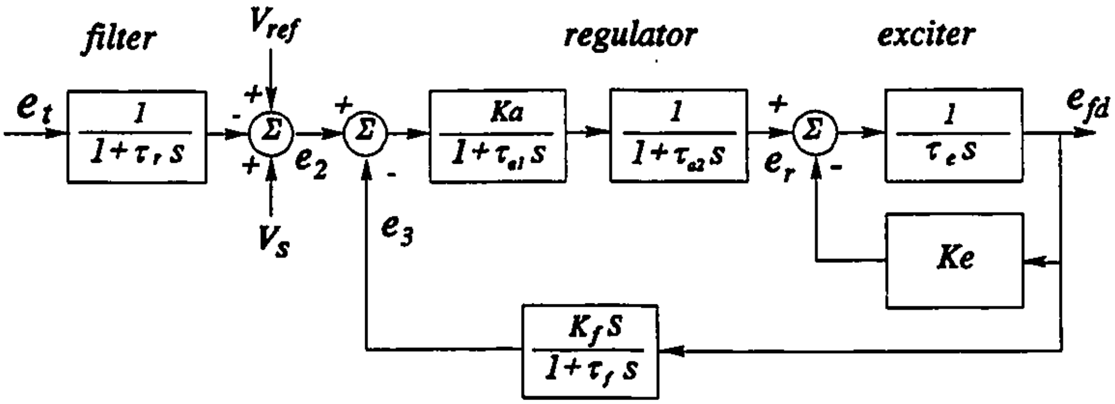


Figure 7-4 Block diagram of excitation system.

$$\dot{e}_3 = -\frac{e_3}{\tau_f} + \frac{K_f}{\tau_f \tau_e} e_r - \frac{K_f K_e}{\tau_f \tau_e} e_{fd} \tag{7-17}$$

$$\dot{e}_r = -\frac{K_a}{\tau_z} e_3 - \frac{e_r}{\tau_a} + \frac{K_a}{\tau_a}(V_{ref} + V_s - e_t) \tag{7-18}$$

$$\dot{e}_{fd} = \frac{e_r}{\tau_e} - \frac{K_e}{\tau_e} e_{fd} \tag{7-19}$$

$$e_f = \frac{\sqrt{3}\, r_f}{x_{ad}} e_{fd} \tag{7-20}$$

where

$$e_t = \frac{\sqrt{e_d^2 + e_q^2}}{\sqrt{3}} \; . \tag{7-21}$$

7.2.5 Power System Stabilizer

Fig. 7-5 shows the block diagram of the power system stabilizer taken from Fig. D16 of Ref. [94]. From the individual blocks, the differential equations

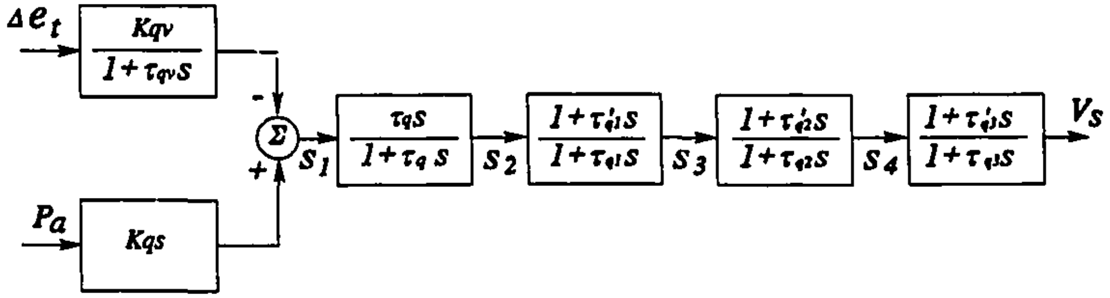

Δe_t
$\frac{Kqv}{1+\tau_{qv}s}$
Σ
-
+
S_1
$\frac{\tau_q s}{1+\tau_q s}$
S_2
$\frac{1+\tau'_{q1}s}{1+\tau_{q1}s}$
S_3
$\frac{1+\tau'_{q2}s}{1+\tau_{q2}s}$
S_4
$\frac{1+\tau'_{q3}s}{1+\tau_{q3}s}$
V_S
Pa
Kqs

Figure 7-5 Block diagram of power system stabilizer.

which have to be integrated are:

$$\dot{s}_1 = K_{qs}\dot{P}_a \tag{7-22}$$

$$\dot{s}_2 = \dot{s}_1 - \frac{s_2}{\tau_q} \tag{7-23}$$

$$\dot{s}_3 = \frac{\tau'_{q1}}{\tau_{q1}}\dot{s}_1 - \left(\frac{\tau'_{q1}}{\tau_q\,\tau_{q1}} - \frac{1}{\tau_{q1}}\right)s_2 - \frac{s_3}{\tau_{q1}} \tag{7-24}$$

$$\dot{s}_4 = -\frac{\tau'_{q1}\,\tau'_{q2}}{\tau_{q1}\,\tau_{q2}}\dot{s}_1 - \left(\frac{\tau'_{q1}\,\tau'_{q2}}{\tau_q\,\tau_{q1}\,\tau_{q2}} - \frac{\tau'_{q2}}{\tau_{q1}\,\tau_{q2}}\right)s_2 - \left(\frac{\tau'_{q2}}{\tau_{q1}\,\tau_{q2}} - \frac{1}{\tau_{q2}}\right)s_3 - \frac{s_4}{\tau_{q2}} \tag{7-25}$$

where

$$\dot{P}_a = \dot{P}_m - \dot{P}_e \tag{7-26}$$

$\dot{P}_m$ is from Eq. (7-15), and

$$\dot{P}_e = \dot{\omega}_g T_e + \omega_g \dot{T}_e$$

$$\dot{T}_e = \frac{\dot{\lambda}_d(\lambda_q - \lambda_{mq}) - \dot{\lambda}_q(\lambda_d - \lambda_{md}) + \lambda_d(\dot{\lambda}_q - \dot{\lambda}_{mq}) - \lambda_q(\dot{\lambda}_d - \dot{\lambda}_{md})}{3x_l}$$

$$\dot{\lambda}_{md} = \frac{\dot{\lambda}_d/x_l + \dot{\lambda}_{fd}/x_{fd} + \dot{\lambda}_{kd}/x_{kd}}{1/x_{ad} + 1/x_l + 1/x_{fd} + 1/x_{kd}}$$

$$\dot{\lambda}_{mq} = \frac{\dot{\lambda}_q/x_l + \dot{\lambda}_{kq}/x_{kq}}{1/x_{aq} + 1/x_l + 1/x_{kq}}$$

$\dot{\omega}_g$ is from Eq. (7-6).

7.2.6 Local Load, Transformer and Transmission Lines

The local load, the transformer and the transmission lines are modelled according to Kirchhoff's voltage and current laws. For example, in module *#1* for the turbo-generator g_1 in Fig. 7-1, the differential equations of the local load, the transformer and the transmission lines are:

$$\dot{i}_{1d} = -\frac{\omega_o R_1}{X_1} i_{1d} + \omega_o \omega_g i_{1q} + \frac{\omega_o}{X_1} e_d - \frac{\omega_o}{X_1} v_{od} \tag{7-27}$$

$$\dot{i}_{1q} = -\omega_o \omega_g i_{1d} - \frac{\omega_o R_1}{X_1} i_{1q} + \frac{\omega_o}{X_1} e_q - \frac{\omega_o}{X_1} v_{oq} \tag{7-28}$$

$$\dot{i}_{3d} = -\frac{\omega_o R_3}{X_3} i_{3d} + \omega_o \omega_g i_{3q} + \frac{\omega_o}{X_3} e_d - \frac{\omega_o}{X_3} v_{cd} \tag{7-29}$$

$$\dot{i}_{3q} = -\omega_o \omega_g i_{3d} - \frac{\omega_o R_3}{X_3} i_{3q} + \frac{\omega_o}{X_3} e_q - \frac{\omega_o}{X_3} v_{cq} \tag{7-30}$$

$$\dot{i}_{1rd} = -\frac{\omega_o R_{rl}}{X_{rl}} i_{1rd} + \omega_o \omega_g i_{1rq} + \frac{\omega_o}{X_{rl}} e_d \tag{7-31}$$

$$\dot{i}_{1rq} = -\omega_o \omega_g i_{1rd} - \frac{\omega_o R_{rl}}{X_{rl}} i_{1rq} + \frac{\omega_o}{X_{rl}} e_q \tag{7-32}$$

where v_{od}, v_{oq} are the voltages of infinite bus, and v_{cd}, v_{cq} are the voltages across the load bus.

7.2.7 System Load

The system load is also modelled according to Kirchhoff's voltage and current laws as follows:

$$\dot{v}_{cd} = \omega_o \omega_g v_{cq} - \omega_o X_c i_{td} + \omega_o X_c i_{3d} + \omega_o X_c i_{4d} \tag{7-33}$$

$$\dot{v}_{cq} = -\omega_o \omega_g v_{cd} - \omega_o X_c i_{tq} - \omega_o X_c i_{3q} + \omega_o X_c i_{4q} \tag{7-34}$$

$$\dot{i}_{td} = \frac{\omega_o}{X_t} v_{cd} - \frac{\omega_o r_t}{X_t} i_{td} + \omega_o \omega_g i_{tq} \tag{7-35}$$

$$\dot{i}_{tq} = \frac{\omega_o}{X_t} v_{cq} - \omega_o \omega_g i_{td} - \frac{\omega_o r_t}{X_t} i_{tq} \tag{7-36}$$

It should be stressed that the Capacitor Break-Point Partitioning method revolves around the capacitance in X_c and the voltages v_{cd}, v_{cq} in Eqs. 7-33 to 7-36.

Formulae of e_d, e_q

The voltages across the mutual reactance X_m of the module *#1* for the transformer interconnection with the local load $R_{r1}+jX_{r1}$, the transmission lines R_1+jX_1 and R_3+jX_3 can be developed as:

$$e_d=\frac{\dfrac{(k_d-1)(\omega_g\lambda_q+r_l i_d)+k_{fd}(e_f-r_f i_f)-k_{kd}r_{kd}i_{kd}}{x_l}+\dfrac{R_1 i_{1d}+v_{od}}{X_1}+\dfrac{R_3 i_{3d}+v_{cd}}{X_3}+\dfrac{R_{rl}}{X_{rl}}i_{rd}-\omega_g i_q}{\dfrac{1}{X_m}+\dfrac{1}{X_1}+\dfrac{1}{X_3}+\dfrac{1}{X_{rl}}+\dfrac{1-k_d}{x_l}} \tag{7-37}$$

$$e_q=\frac{\dfrac{(k_q-1)(-\omega_g\lambda_d+r_l i_q)-k_{kq}r_{kq}i_{kq}}{x_l}+\dfrac{R_1 i_{1q}+v_{oq}}{X_1}+\dfrac{R_3 i_{3q}+v_{cq}}{X_3}+\dfrac{R_{rl}}{X_{rl}}i_{rq}+\omega_g i_d}{\dfrac{1}{X_m}+\dfrac{1}{X_1}+\dfrac{1}{X_3}+\dfrac{1}{X_{rl}}+\dfrac{1-k_q}{x_l}} \tag{7-38}$$

where

$$k_d=\frac{1/x_l}{1/x_{ad}+1/x_l+1/x_{fd}+1/x_{kd}}\ ,$$

$$k_q=\frac{1/x_l}{1/x_{aq}+1/x_l+1/x_{kq}}\ ,$$

$$k_{fd}=\frac{1/x_{fd}}{1/x_{ad}+1/x_l+1/x_{fd}+1/x_{kd}}\ ,$$

$$k_{kd}=\frac{1/x_{kd}}{1/x_{ad}+1/x_l+1/x_{fd}+1/x_{kd}}\ ,$$

$$k_{kq}=\frac{1/x_{kq}}{1/x_{aq}+1/x_l+1/x_{kq}}\ ,$$

The voltages e_d , e_q of Eqs. (7-37) and (7-38) are substituted in the generator equations (7-1) and (7-2).

The voltages e_d , e_q across the transformer M_2 in module #2 have similar formulae.

The magnetic saturation in the generator and the transformer iron has not been modelled in this study. The ferromagnetic nonlinearity can be represented by a "look-up table" but it is felt that its detail modelling will extend the scope of this research. For this reason, this work is deferred for future thesis research.

7.3 IMPLEMENTATION OF DIGITAL TNA

7.3.1 Capacitor Break-Point Partitioning

As the subsystems of the generator are or can be put in the block diagram form, there is no difficulty in organizing them so that the differential equations can be assigned to several processors to be numerically integrated in parallel. The major difficulty lies in the transmission line network tying together the stator equations of many turbo-generators. Chapter 6 has resolved this difficulty by using the capacitors, which are present in the power network but normally neglected in the simplifying assumptions, as the breakpoints for the partitioning. In the system of Fig. 7-1, the breakpoint is the capacitance in the system load. From the breakpoint, the system is divided into three modules and in fact, the generator equations (7-1) to (7-5) and (7-27) to (7-32), and the system load equations from Eqs. (7-33) to (7-36) have incorporated the Capacitor Break-Point Partitioning method.

7.3.2 Multi *d-q* Reference Frames

The generator equations are formulated with respect to the *d-q* axes which rotate with the rotor inertia. The angular velocity of the rotor inertia, ω_g, and the torque angle δ_g are solutions to Eqs. (7-6) and (7-8). Depending on the dynamics in each turbo-generator shaft system, the *d-q* axes of module *#1* and *#2* will, in general, take on different angles as illustrated in Fig. 7-6. In principle, the voltages and the currents of each of the turbo-generators have to be referred to the *a-b-c* frame (through Eq. (6-11)) or the *D-Q* frame of the infinite bus, so as to ensure that the rotor swings are accounted for in the mathematical formulations. Using the torque angle δ_g solved from Eqs. (7-6) and (7-8) in Eq. (6-11), all the voltages and currents of different modules can be transformed into the common *D-Q* frame.

In order to shorten the computation time in the real-time simulation of the

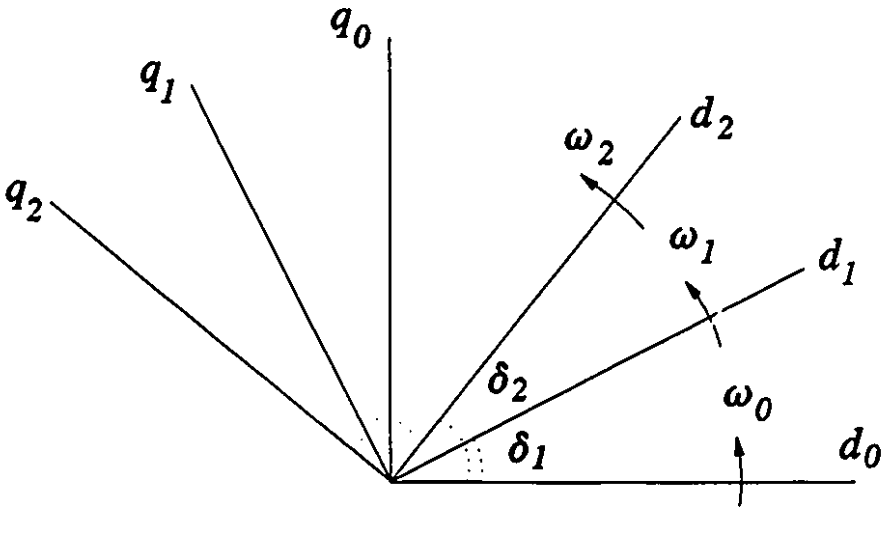


Figure 7-6 Relationship of *d-q* coordinates of modules #1, #2 and infinite bus.

small system in Fig. 7-1, the d_1-q_1 frame of g_1 of module *#1* is selected as the primary coordinate frame in the simulation problem. The system equations of module *#3* are also written in the d_1-q_1 coordinate frame. Therefore, there need only data interchanges between the d_1-q_1 and the d_2-q_2 frames. That is, the computation results from module *#2* for the g_2 generator and from module *#3* representing the load bus are expressed in the d_1-q_1 coordinates. For example, the currents i_{4d}, i_{4q} computed by module *#2* become i_{41d}, i_{41q} in module *#1*. The coordinate transformation is by:

$$\alpha = \delta_2 - \delta_1 \tag{7-39}$$

$$i_{41d} = i_{4d} \cos\alpha + i_{4q} \sin\alpha \tag{7-40}$$

$$i_{41q} = -i_{4d} \sin\alpha + i_{4q} \cos\alpha \tag{7-41}$$

Likewise, v_{cd}, v_{cq} of module *#3* which are framed in the d_1-q_1 reference frame becomes v_{2cd}, v_{2cq} when used by module *#2*. The transformation is:

$$v_{2cd} = v_{cd} \cos\alpha - v_{cq} \sin\alpha \tag{7-42}$$

$$v_{2cq} = v_{cd} \sin\alpha + v_{cq} \cos\alpha \tag{7-43}$$

7.3.3 Modified Euler's Method

As the generator system equations are mathematically nonlinear, they have to be numerically integrated. The 2-step, modified Euler's method (predictor-corrector) has been used for module *#1* and *#2*. The algorithm of the modified Euler's method is [98]:

$$y_{n+1}=y_n+h(K_1+K_2)/2 \tag{7-44}$$

$$K_1=f(t_n,\ y) \tag{7-45}$$

$$K_2=f(t_n+h,\ y_n+K_1h) \qquad n=1, 2, 3, \ldots \tag{7-46}$$

where t_n represents time, h represents the step size and y represents the state variables.

7.3.4 Discrete-Time Approach

The load bus system in module *#3* of Fig. 7-1 is linear and can be put in the first order standard form:

$$\dot{\underline{x}}=[A]\underline{x}+[B]\underline{u} \tag{7-47}$$

$$\underline{y}=[C]\underline{x} \tag{7-48}$$

where $[A]$, $[B]$ and $[C]$ are constant matrices from the circuit parameters.

Instead of using modified Euler's method, the discrete-time representation [78, 79] is used so that Eqs. (7-47) and (7-48) become

$$\underline{x}_{k+1}=[F]\underline{x}_k+[G]\underline{u}_k \tag{7-49}$$

$$\underline{y}_k=[C]\underline{x}_k \tag{7-50}$$

where

$$t_{k+1}=t_k+h. \tag{7-51}$$

The matrices are:

$$[F]=\exp[A]h \tag{7-52}$$

$$[G]=(\int_0^h [exp[A]\tau]d\tau)[B] \tag{7-53}$$

The discrete-time approach has been found to be more numerically stable than the modified Euler's method when the capacitive reactance is large. Because both the $[F]$ and $[G]$ matrices contain the exact numerical integration in the step-size h, it avoids the inaccuracies of the modified Euler's method. In retrospect, all the functional blocks in the transfer functions in the excitation system, the governor and the power system stabilizer are linear and are amenable to the same discrete-time approach.

7.3.5 Parallel Processing of the Power System

The power system of Fig. 7-1 is divided into three modules using the capacitance X_c as the break-point. The turbo-generator subsystems are regarded as independent modules *#1* and *#2*. The missing information is the capacitor voltage v_c which is computed by the load bus module *#3*. The state variables of the load bus module are also calculated independently except for the currents i_3 and i_4 which have to be computed by the turbo-generator modules. Thus the two turbo-generator modules and the load bus module are programmed to be executed in parallel with minimal information exchanges which consist of v_c , i_3 and i_4.

In the turbo-generator module, the governor system of Fig. 7-3 outputs the torque T_{in} for the input set at p_{mi}. The excitation system of Fig. 7-4 builds the field voltage e_{fd} from the generator terminal voltage e_t and the power system stabilizer output V_s from Fig. 7-5. The calculations for the governor system and for the excitation system are carried out using the pipeline parallel processing concept [55]. This is necessary because the input V_s of the excitation system of Fig. 7-4 must follow the output of the power system stabilizer which in turn must receive P_a as output of the governor. For the two groups of calculations to be done in parallel, the computation is performed for the excitation system at step k, while the computation performed for the governor is at step $k+1$.

For the generator-turbine system and the transmission lines, the outputs are the currents of the generator and the inputs are the torque T_{gt}, field voltage e_f and its terminal voltages e_d and e_q. For the transmission lines, the transformer and the local load, the outputs are the line currents and the inputs are e_d, e_q, and v_{cd} , v_{cq} the load bus voltages. As the terminal voltages e_d and e_q can be determined in advance, the two groups of calculations are executed in parallel.

In each of the turbo-generator modules, certain computations are processed in parallel. Three guidelines are followed to allocate the computing tasks for processors:

1. All the processors should be put to work in parallel as often as possible so as to maximize the computation throughput.

2. The data interchanges through global memories are as infrequent as

possible so as to minimize the competitions for access to the global memories and to reduce the waiting times of the processors.

3. The tasks arranged for the processors should follow as closely to the function blocks of the sub-systems as possible so as to minimize mistakes in writing the programs and to facilitate easy debugging.

In addition, the programming should avoid interrupting any pipeline executions of the processors.

7.3.6 Arrangement of Tasks

From the guidelines for parallel processing stated above, the tasks have been arranged for the processors as follows:

The two Computational Modules have been assigned the computational burden of module *#1* and *#2* as shown in Fig. 7-1. Each Computational Module simulates one turbo-generator system which includes the turbine, the generator, the transmission lines, the transformer, the local load, the governor, the excitation system, and the power system stabilizer.

The I/O Module has been assigned the computational burden of module *#3* together with the post-processing of output information for analog display on a 4-channel oscilloscope.

In each Computational Module, one of the processors simulates the governor system and the turbine-generator system. The other processor simulates the power system stabilizer, the excitation system and the transmission lines and the remaining tasks. The division of the computational tasks performed by the three hardware processor-modules are shown in Fig. 7-7.

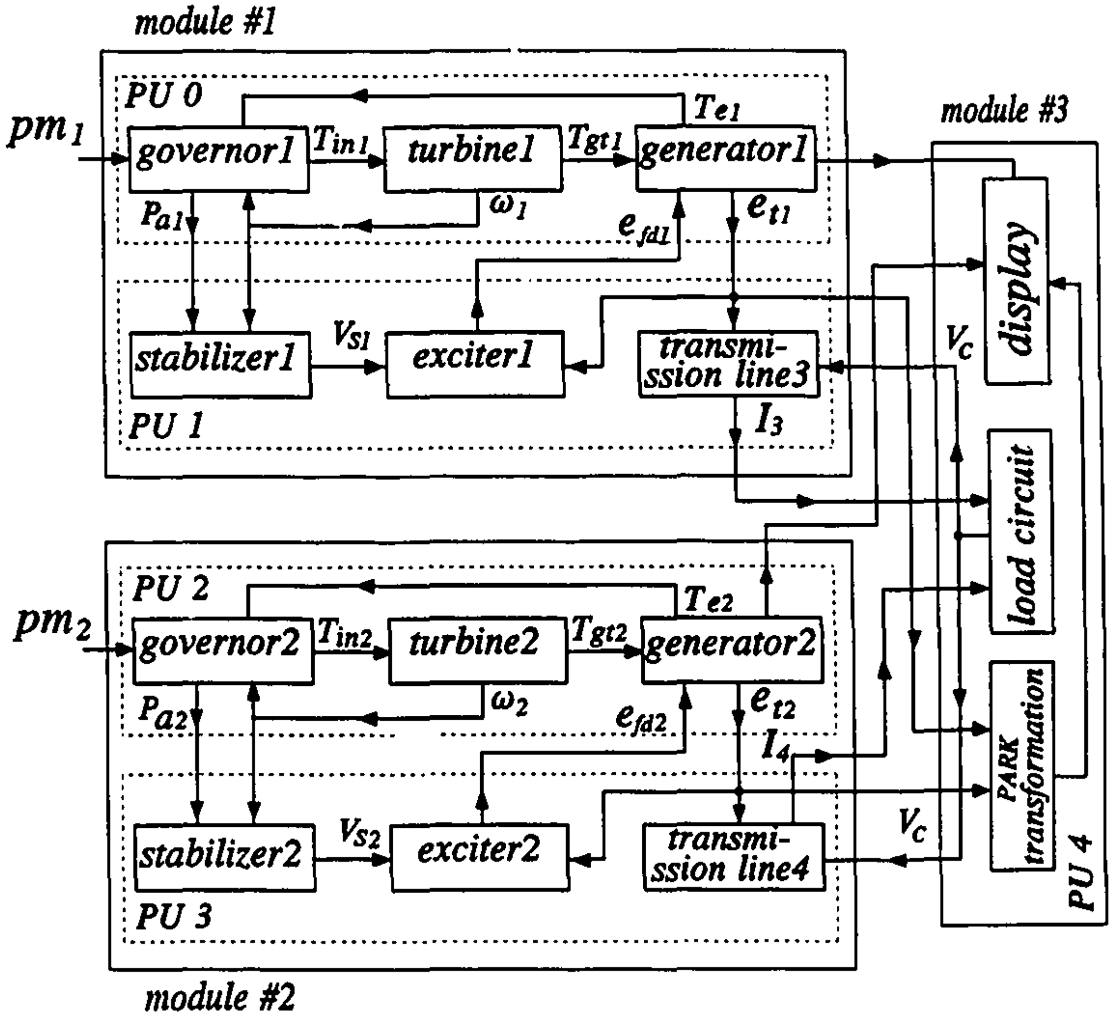


Figure 7-7 Division of tasks executed by processor units (PUs).

7.3.7 Parallel Programming

Fig. 7-8 is the flow chart of the initialization program. The initialization program includes the first step of the simulation of the governor so as to implement the pipelining parallel processing concept [55]. The PC has been used to code and load the simulation programs and to start the processors in the simulation runs.

The initialization programs are almost identical for the five processors. At the beginning of the program, the internal environment of the processor is defined. The on-chip registers are set for proper interrupt, synchronization and status control. The external environment of the processor is also defined for the Extensible Modular Multiprocessor System. For example, the multi-module configuration of the processing units (*PU*s) are defined through the control ports. The interrupt vectors are loaded to point the positions of service routines in the memory. Then the control block for the real-time simulation is initialized. In order to shorten the real-time computation, as many of the repeated arithmetic operations which involve fixed parameters are grouped together as pre-calculated constant coefficients. The initial conditions of all the state variables for the power system are inputted and calculated. For *PU0* and *PU2*, which simulate the governor systems, the first step calculations are executed. For *PU4*, the integer corresponding to the integration step-size of *100* μs is loaded into the on-chip timer, and the counter acting as a real-time clock is reset to zero. The initial

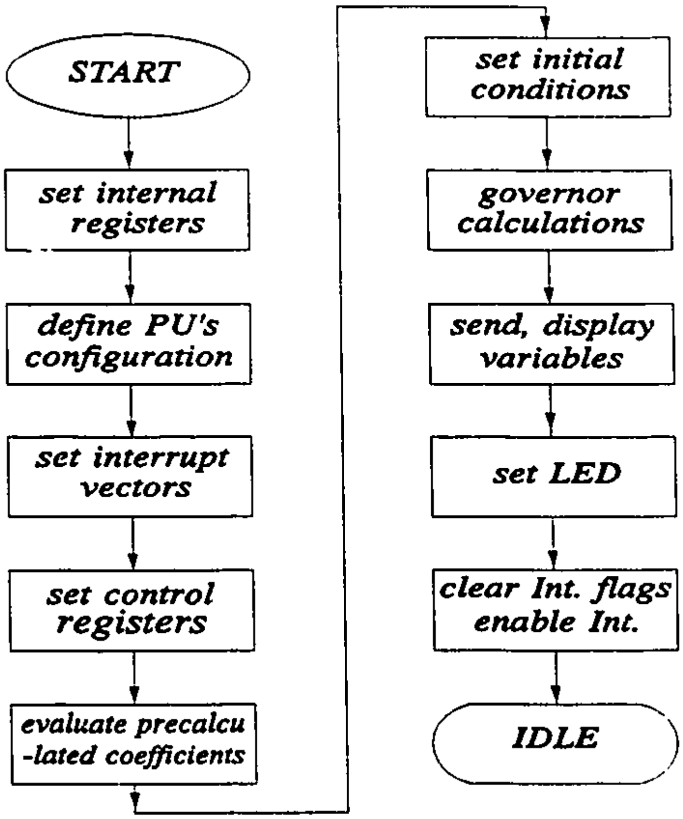


Figure 7-8 Flow chart of initialization program.

states of the two generator variables are sent to *PU4* for display. The LED (Light Emitting Diodes) are flashed to signal that the simulation is working properly. Finally, the interrupt signals are reset externally and internally; the processor interrupts are enabled and the processors remain in the idle state waiting for interrupt signals to come.

Fig. 7-9 is the flow chart of the parallel program. Each column represents the work of each PU. Each row of the blocks represents the work in each synchronized stage. *PU4* starts each step of the calculation by the interrupt

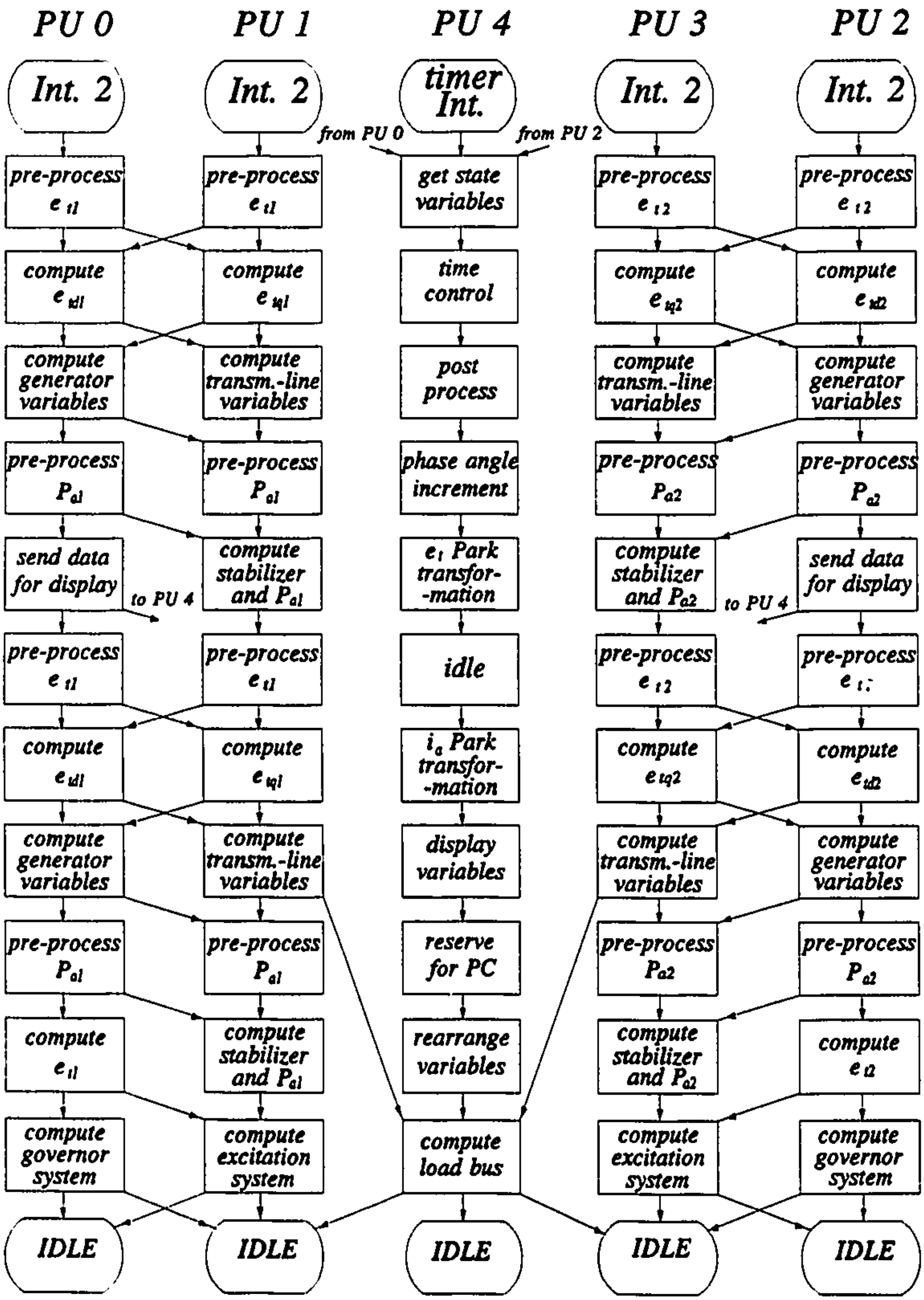


Figure 7-9 Flow chart of parallel programs.

signals of its on-chip timer. Then it sends out signals to the other four processors to start them working in parallel. In each real-time step of *100* μs, the processors are synchronized ten times by the interlock circuit. Before each synchronization, the processors prepare the data and put them into global memories so that they can be exchanged. After each synchronization, the processors fetch the data needed from the global memories and then proceed to execute the algorithm.

In executing the modified Euler's method, the program is divided into two parts. The first part, Eq. (7-45), is for the predictor, while the second part, Eq. (7-46), is for the corrector. The Extensible Modular Multiprocessor System devotes two stages in calculating the generator terminal voltages e_t represented by the Eqs. (7-37) and (7-38) in each part. Then the simulation of the generator-turbine system, the transmission lines and power system stabilizer are implemented in parallel by integrating Eqs. (7-1) to (7-11) and (7-27) to (7-32). The variable P_a from (7-26) is the accelerating power which is used as one of the inputs of the power system stabilizer in the equations from (7-22) to (7-25). After the calculation for the predictor (7-45), the corrector equations (7-46) and (7-44) are implemented by using the same procedure.

In the program, eleven sets of the data interchanges are executed. Ten sets are controlled by the synchronization circuit which is resident in each processor and is controlled by software instruction. The eleventh set of data transferred is after an interrupt, and the data for analog display from the other processors are read by *PU4* from the global memories. *PU4* displays the results of the last step. The last stage of the calculations is for the equations of the governor system from (7-12) to (7-16), the excitation system from (7-17) to (7-21) and the load bus from (7-33) to (7-36).

The calculation of the governor equations is for step $k+1$ while the calculations of the excitation system and the load bus are for step k. The computation of the governor equations in the last stage of the program ensures that processors are optimally busy all the time and are sharing the burden of work equitably. It is to be recalled that the outputs T_{in} and P_a of Fig. 7-3 are used by the power system stabilizer of Fig. 7-5 and in Eq. (7-7). This means that the governor equations should have been computed earlier in the program. Since this is not possible, the pipelining principle has been used. This explains why the governor equations are solved in the initialization program of Fig. 7-8. To keep all the processors equally busy, the governor equations are solved by *PU0* and *PU2* in the last stage of the iterative parallel program for the $k+1$ step.

When the processors complete all the work of the $\underline{k}$th step, they go into the idle state and wait for the next interrupt.

7.4 EXPERIMENTAL RESULTS

This section presents the oscillograms from a test program which includes (i) symmetry checks, (ii) behavioral checks against well known waveforms of hunting oscillations, synchronization out-of-phase torques, and (iii) checks to its ability to predict subsynchronous resonance phenomena.

The experimental tests have been planned for easy verification by symmetry to ensure that: (a) the hardware has been faultlessly assembled, (b) the parallel programs have been meticulously written, (c) the Capacitor Break-Point Partitioning method is correct, (d) the data transfers and communications between processor modules are executed as planned. Thus, the parameters in the turbo-generator modules *#1* and *#2* have been set to be identical as listed in Appendix-B. In addition, the load bus module *#3* is placed at the mid-point of the transmission line joining the two turbo-generators. Unless perfect, the test results of Figs. 7-12, 7-13(a), and 7-15(a), (b) would have stood out as asymmetric.

The test results are recorded as continuous waveforms on a 4-channel oscilloscope. The time scale testifies to the real-time conversion by the D/A converters of the I/O Module.

On completion of the verification tests, one can use the digital TNA with confidence and run the programs with the parameters which are actually in the power system.

7.4.1 Individual Turbo-Generator Tests

Governor, Exciter, Power System Stabilizer

Fig. 7-10 (a), (b) and (c) shows the responses of: the angular velocity deviation $\Delta\omega_l$ of the generator, the torque angle δ_l, and the electromechanical torque ΔT_e for a step change in pm_l, which is the power demand reference in the governor system as shown in Fig. 7-3. In (a), there is neither the excitation system nor the power system stabilizer so that the *2 Hz* oscillatory transient has a long time constant. In (b), the excitation system is added and the shortening of the oscillation is apparent. In (c), the power system stabilizer is added to dampen the

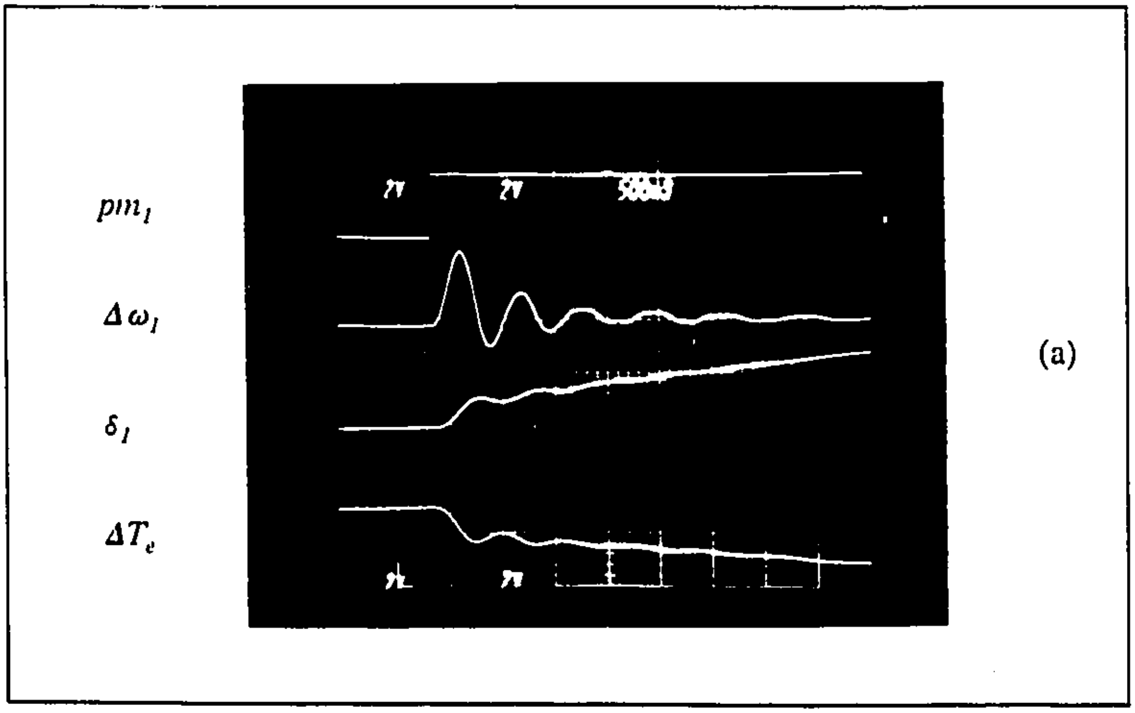


Figure 7-10 Responses of $\Delta\omega_l$, δ_l, ΔT_e to step change in pm_l. (a) no excitation system and no power system stabilizer.
(continued on the next page)

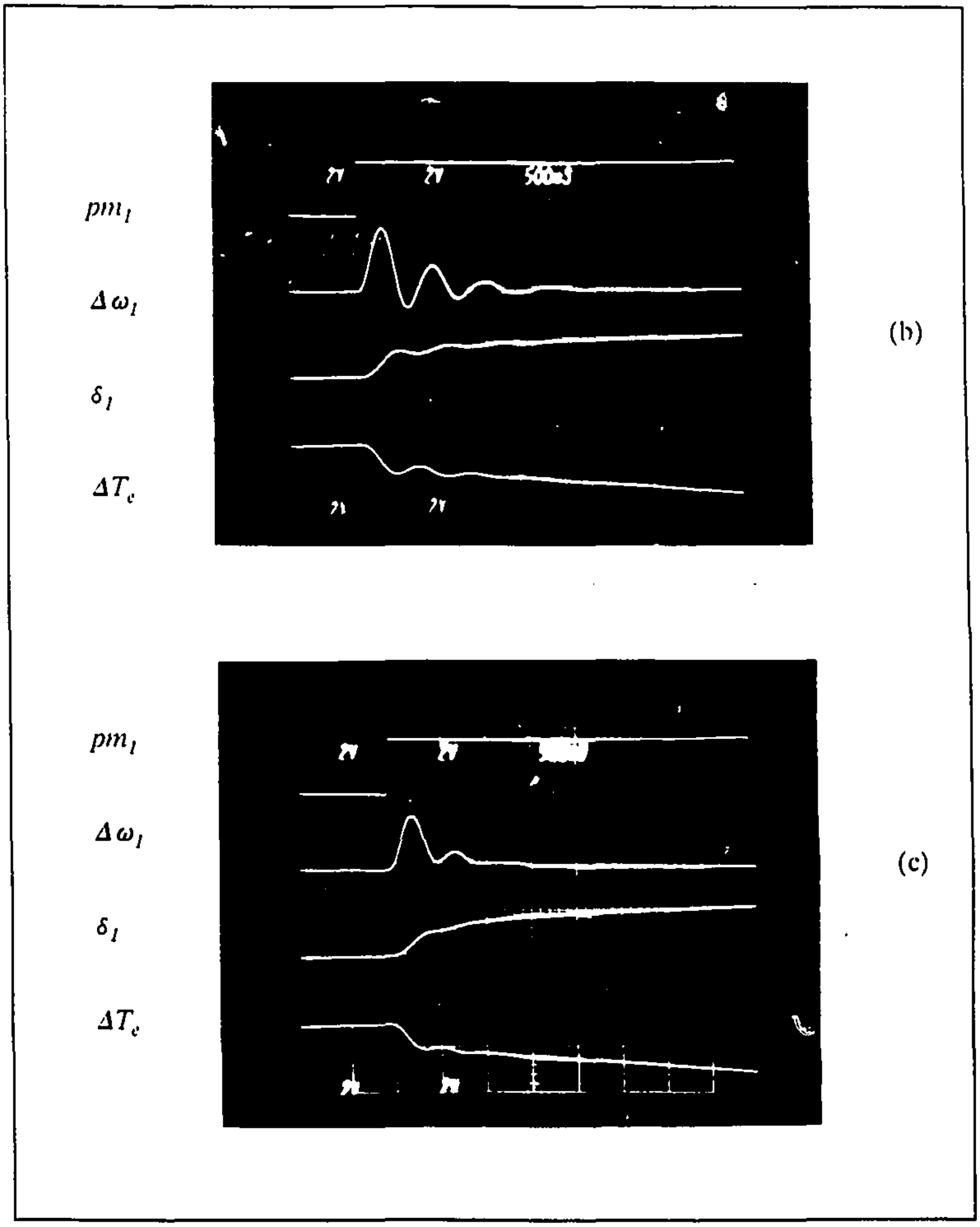


Figure 7-10 (continued from the last page)

(b) excitation system added, (c) power system stabilizer added.

oscillations further. The improvements in the damping in the time responses serve as proofs that the excitation system and the power system stabilizer have been correctly modelled.

At the end of all the 3 oscillograms, the system has not arrived at a steady-state because the torque angle δ and the electromechanical torque ΔT_e are still adjusting to the governor system which has a long time constant.

Fig. 7-11 shows the responses of the turbo-generator with the excitation system and the power system stabilizer for a step change in power demand reference pm_l in (a). The trend of the responses of the state variables (b) λ_d, (c) λ_q and (d) λ_f correspond to well-known published results such as in Ref. [94]. This is a spot-check on the correct modelling of the generator equations.

Post-Processing

The currents in Fig. 7-11 (e) i_d, (f) i_q and (g) i_f are not the state-variables of the numerical integration but have to be post-processed by the I/O Module from the state variables which are the flux linkages $\underline{\lambda}$. These results demonstrate the real-time post-processing, which in this case is not time-consuming.

7.4.2 Multi-Machine Tests

Steady-State Stability Boundary

For the single generator, the steady-state stability limit is at the peak of the generator T_e-v_s-δ curve around $\delta = 90°$. Experimentally, one approaches this limit

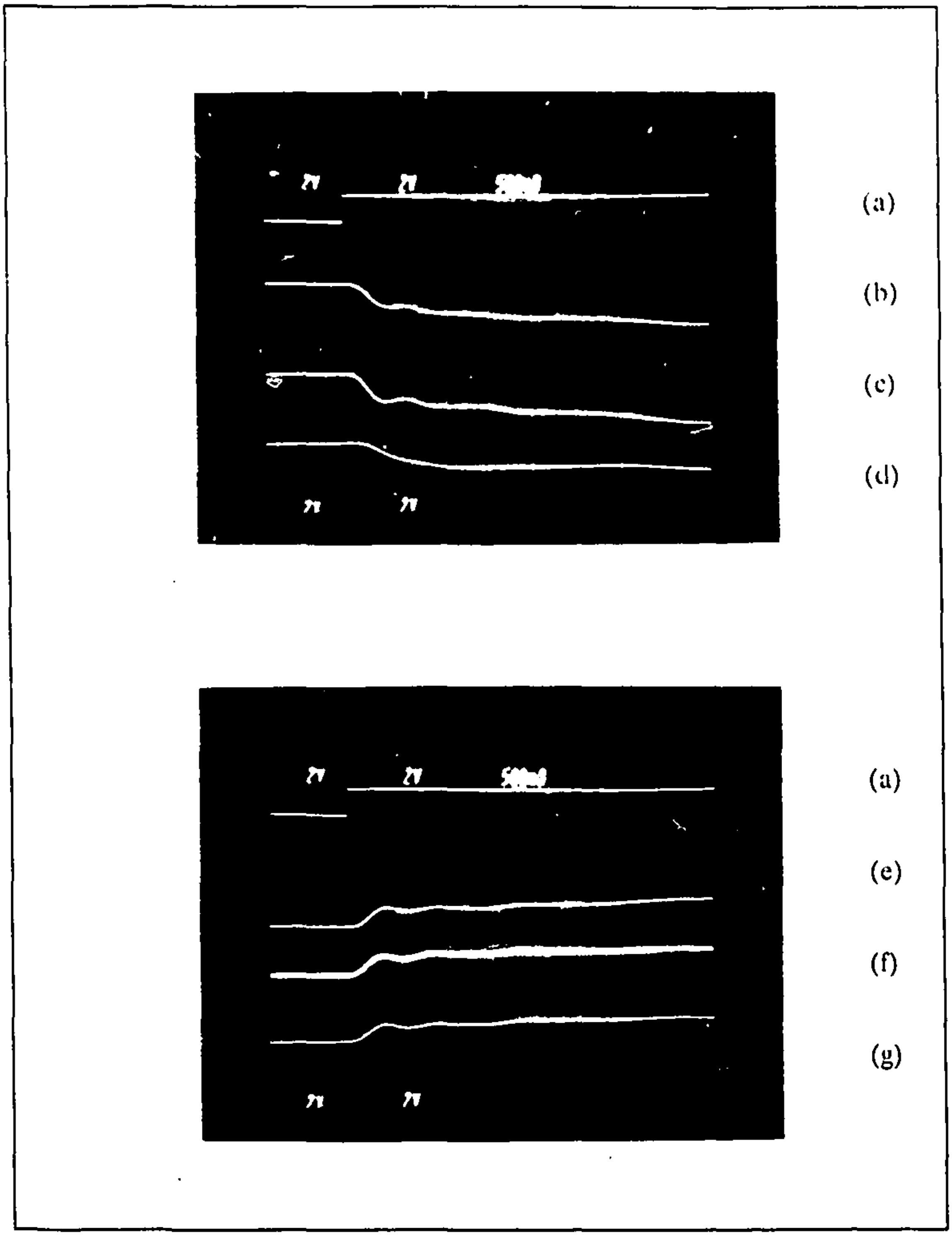


Figure 7-11 Responses of turbo-generator to step change in pm_l. (a) pm_l, (b)λ_d, (c)λ_q, (d)λ_f, (e)i_d, (f)i_q, (g)i_f.

by slowly increasing the power demand reference pm_1 until instability sets in.

Fig. 7-12 shows the steady-state boundary in the pm_1-pm_2 plane for the multi-generator system of Fig. 7-1. The power demand references pm_i ($i=1$, 2) are shown in Fig. 7-3. The boundary limit has been found experimentally by setting pm_2 at different constant values and increasing pm_1 until instability sets in. The mirror symmetry about the 45° line is in agreement with the fact that turbo-generator modules #1 and #2 are identical and the load bus module #3 is situated midway between them. Fig. 7-12 constitutes a steady-state symmetry check on correctness.

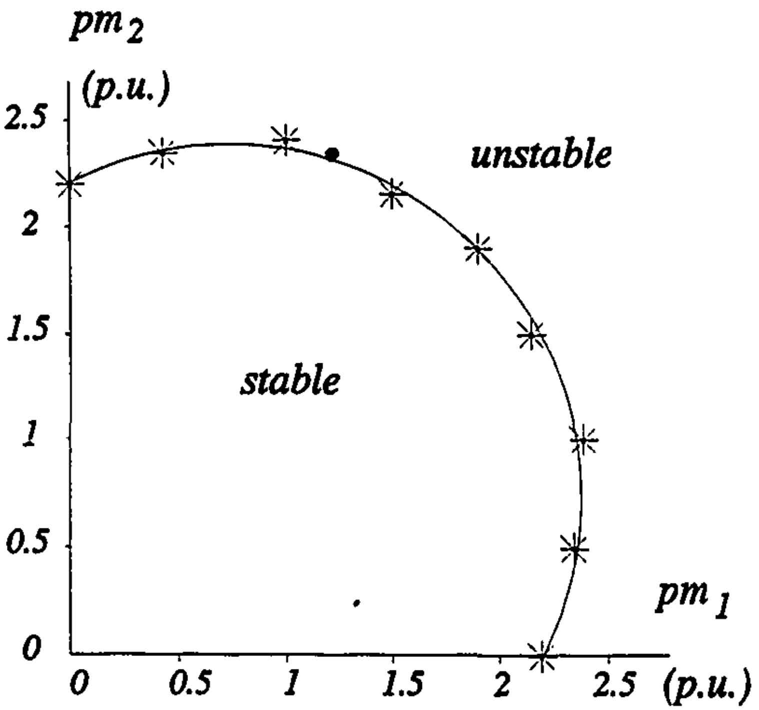


Figure 7-12 Steady-state boundary in pm_1-pm_2 plane.

Small perturbation-Zero Input Response (Free Motion)

In this test, the angular velocities deviations from the synchronous velocity $\Delta\omega_1$, $\Delta\omega_2$ and the torque angles δ_1, δ_2 of turbo-generators of modules *#1* and *#2* are monitored. The initial portions along the time axis of Fig. 7-13 (a) and (b) show the 4 oscilloscope traces of $\Delta\omega_1$, δ_1, $\Delta\omega_2$, δ_2 which are constant because the system has already settled to its steady state. A small perturbation step change is introduced simultaneously to the state variables δ_1 and δ_2. It should be pointed out that such a step change cannot be implemented in practice. However, there is no difficulty in programming these perturbation changes in the processors for the purpose of testing. The subsequent oscilloscope traces show the " free motion" of the small disturbances as they are damped out.

Symmetrical Small Perturbation

In Fig. 7-13 (a), the small perturbations in δ_1 and δ_2 are both equal and negative. The almost identical traces of $\Delta\omega_1$ and $\Delta\omega_2$, and of δ_1 and δ_2 are good checks by symmetry as to the correctness of the work under dynamic conditions.

Non-Symmetrical Small Perturbation

In Fig. 7-13 (b), the small perturbations introduced in δ_1 and δ_2 are of opposite signs. One sees that eventually both turbo-generators have the tendency to cohere.

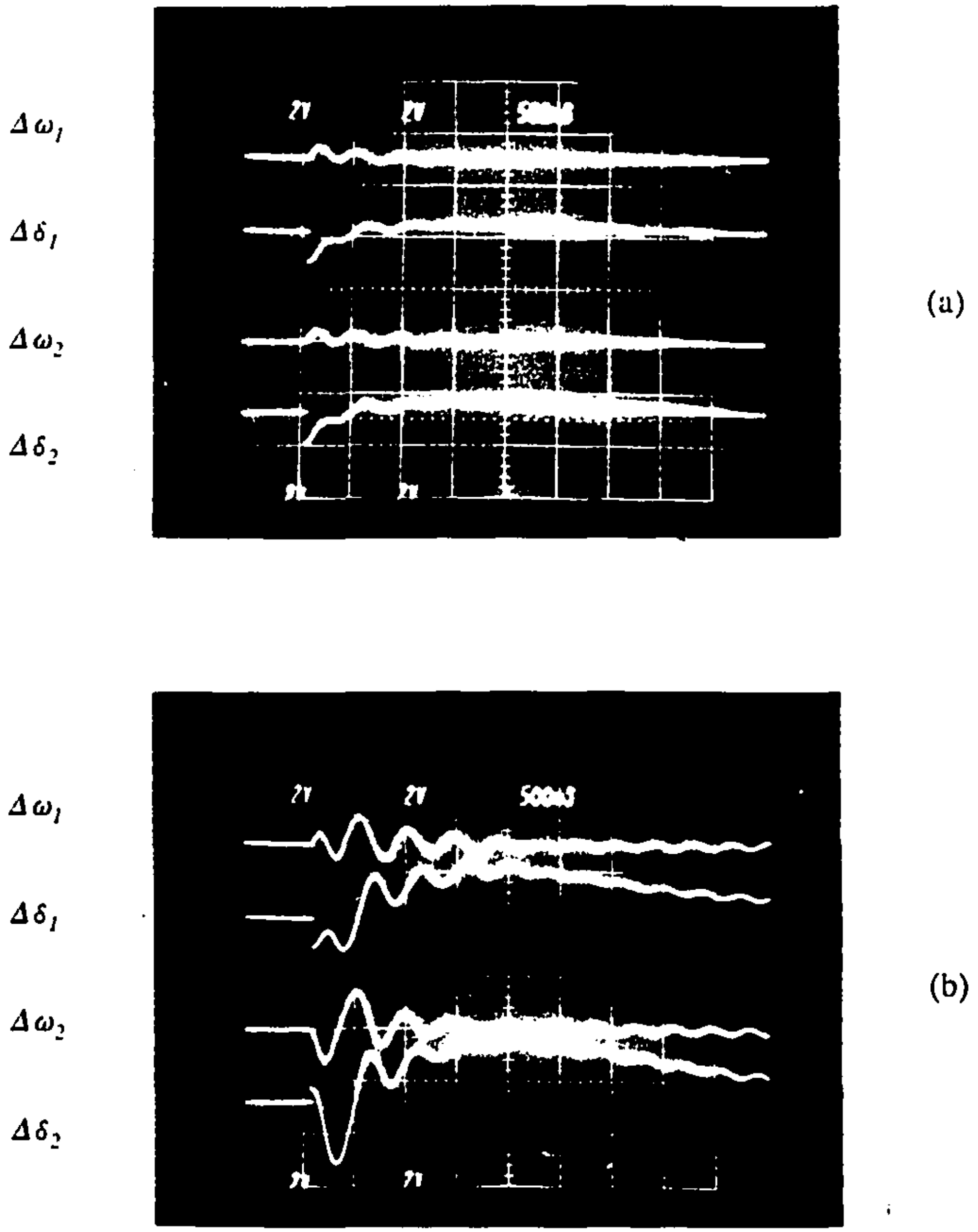


Figure 7-13 Small perturbation - zero input responses. (a) Symmetrical small perturbation. (b) Non-symmetrical perturbation.

7.4.3 Out-of-Phase Synchronization or Reclosing

Although the tests using small perturbations in the torque angle are contrived, the transients following these step changes are of the same nature as those from the out-of-phase synchronization or reclosing of the circuit-breakers of the turbo-generator. Fig. 7-14 is a record, in expanded time scale (*100 ms/div*), of the terminal voltage e_t, the line current i_a, the torque in the shaft between the generator and the turbine ΔT_{gt} and the generator torque ΔT_e of one turbo-generator. The transient generator torque ΔT_e under synchronization and reclosing has been studied extensively **[99-102]** and ΔT_e in Fig. 7-14 bears all the well known characteristics. Initially, there is the *60 Hz* oscillating component which subsides with the generator subtransient and transient. The *2 Hz* oscillation

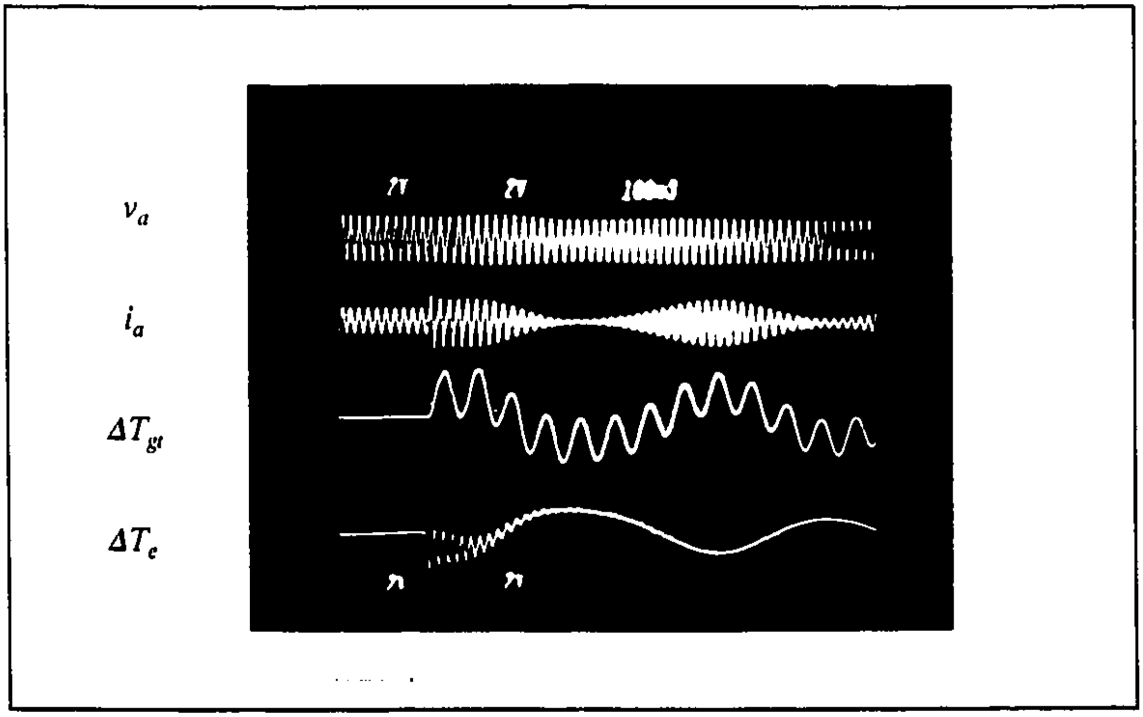


Figure 7-14 Out-of-phase synchronization or reclosing.

comes from the "hunting" of the rotor.

The almost constant magnitude of the voltage waveform e_t testifies to the close regulation by the excitation system in the face of the changes in the magnitude of the line current i_a. The *2 Hz* current envelope is a consequence of the "hunting". The shaft torque ΔT_{gt} has a *15 Hz* torsional oscillation as the inertias of the generator and the turbine in Fig. 7-2 turn against the elastic shaft of torsional stiffness K_{gt}.

7.4.4 Step Inputs in Power Demand References

Fig. 7-15 shows the 4 oscilloscope traces of $\Delta\omega_1$, δ_1, $\Delta\omega_2$, δ_2. Initially, in Fig. 7-15 (a), the steady-state has been achieved for the power demand references $pm_1=pm_2=P$. Simultaneous step changes are made so that finally $pm_1=pm_2=P+\Delta P$. As the upper two traces are identical to the lower two traces, the correctness of the work is given another symmetry check.

In Fig. 7-15 (b), initially the power demand reference settings are $pm_1=pm_2=P$. After the simultaneous step changes are introduced, $pm_1=P+\Delta P$ and $pm_2=P-\Delta P$. The two turbo-generators are seen to adjust in opposite directions to come to the new steady-state, thus giving a further symmetry check.

In Fig. 7-15 (c), initially the power demand reference settings are $pm_1=pm_2=P$. A step change is introduced to module *#1* only so that finally $pm_1=P+\Delta P$ and $pm_2=P$. The perturbations in $\Delta\omega_2$ and δ_2 are brought about

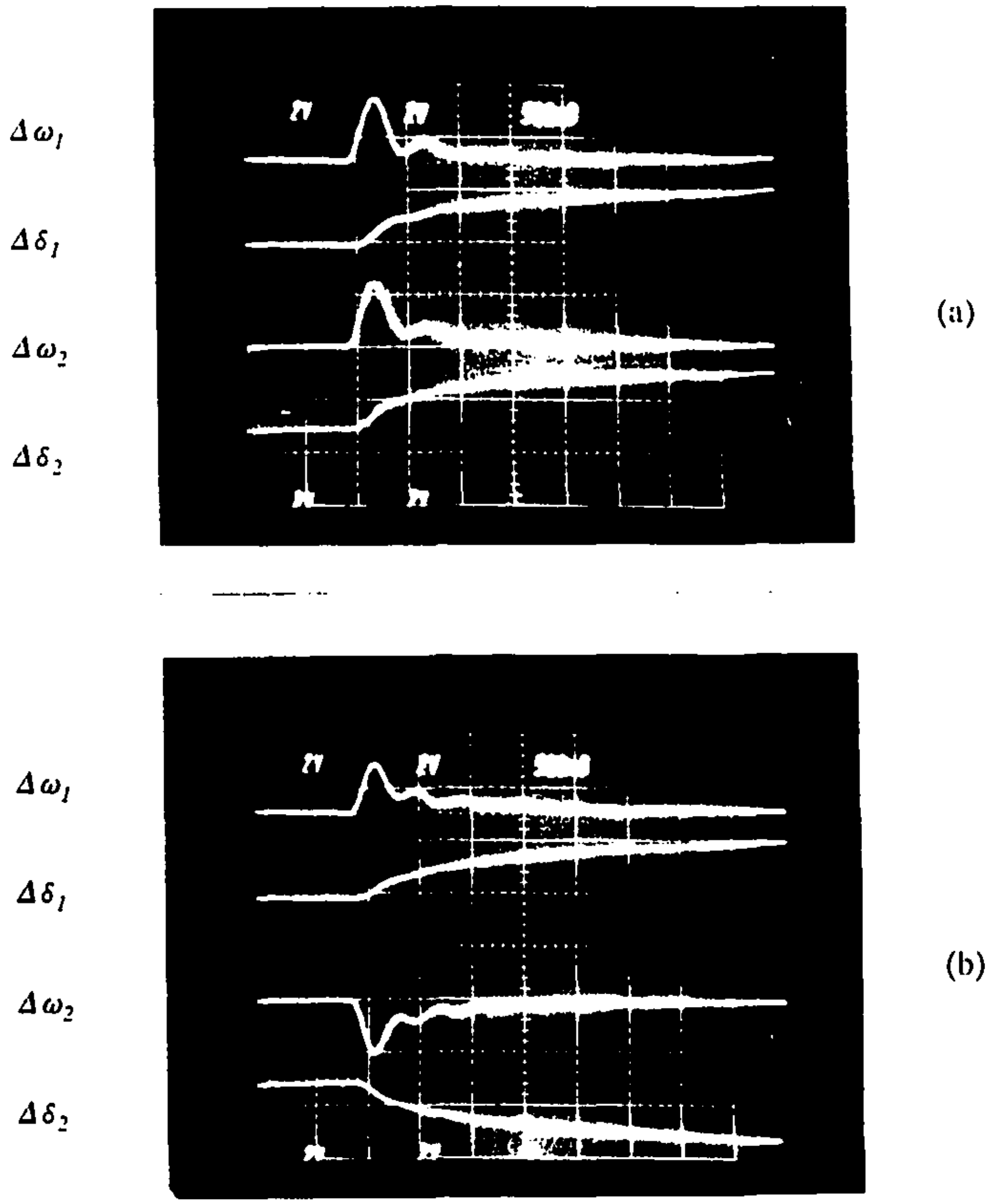


Figure 7-15 Responses to step inputs in power demand references. (a) simultaneous step changes are made from $pm_1=pm_2=P$ to $pm_1=pm_2=P+\Delta P$. (b) simultaneous step changes are made from $pm_1=pm_2=P$ to $pm_1=P+\Delta P$, $pm_2=P\text{-}\Delta P$. (continued on the next page)

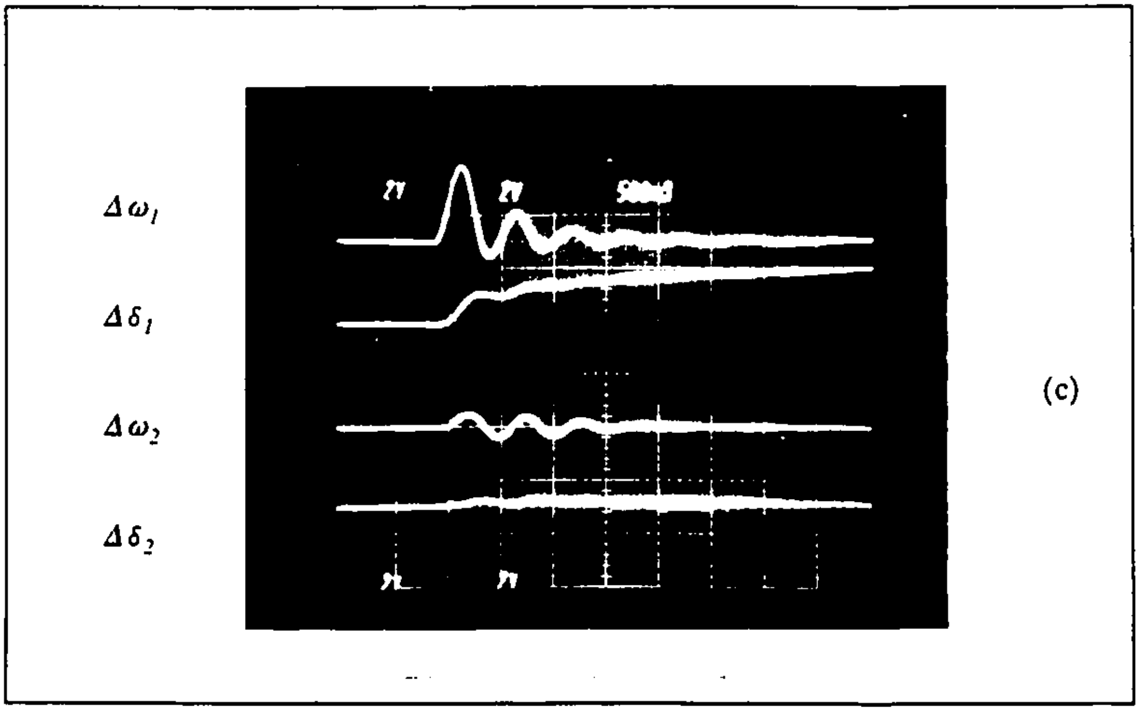


Figure 7-15 (continued from the last page)
(c) a step change is made from $pm_1=pm_2=P$ to $pm_1=P+\Delta P, pm_2=P$.

through the transmission lines joining the two turbo-generators.

7.4.5 Subsynchronous Resonance Phenomena

As part of the extensive testing, it is the aim to ensure that the subsynchronous resonance (SSR) Instability will manifest itself under the adverse conditions for its occurrence. Ref. **[103, 104]** have shown that shunt capacitance, when sufficiently large will resonate with the system inductances at a frequency which falls below *60 Hz*, which by definition is the subsynchronuous resonance frequency. For this reason, the X_c in Fig. 7-1 is reduced so as to have an electrical resonance frequency of *(60-15)=45 Hz*, which is the condition for the torsional interaction phenomena with the shaft resonant frequency of *15 Hz* **[105]**.

Fig. 7-16(a) is the oscillogram of e_t , i_a , ΔT_{gt} and ΔT_e (time scale: *200ms/div*) of this contrived experiment. One sees that at the beginning of the transient, the *15 Hz* oscillation appears in the generator torque ΔT_e showing that processor-modules predict the torsional interaction phenomena. However, the *15 Hz* oscillation in ΔT_e is positively damped out in a few cycles. This stability is expected of the hydro system where the generator inertia is very much large than the turbine inertia **[106]**.

In order to confirm the effect of the generator-to-turbine inertia ratio to damping **[106]**, the experiment has been repeated and recorded with the same scales as in Fig. 7-16 (b), but for the numerical values of the inertias interchanged. Thus the generator inertia is smaller than the turbine inertia but the *15 Hz* shaft torsional resonance frequency is preserved. The *15 Hz* oscillation in the generator torque ΔT_e is sustained in the entire oscillogram. One sees that the *15 Hz* oscillation of the shaft torque ΔT_{gt} in (b) is several times larger than in (a). The results of Fig. 7-16 (a) and (b) have been noticed in Ref. **[107]** and confirmed in Ref. **[108]**.

7.4.6 Numerical Stability Tests

The digital TNA has been run in excess of 5 minutes of real time to ensure that numerical instability does not occur. For this reason, additional tests were

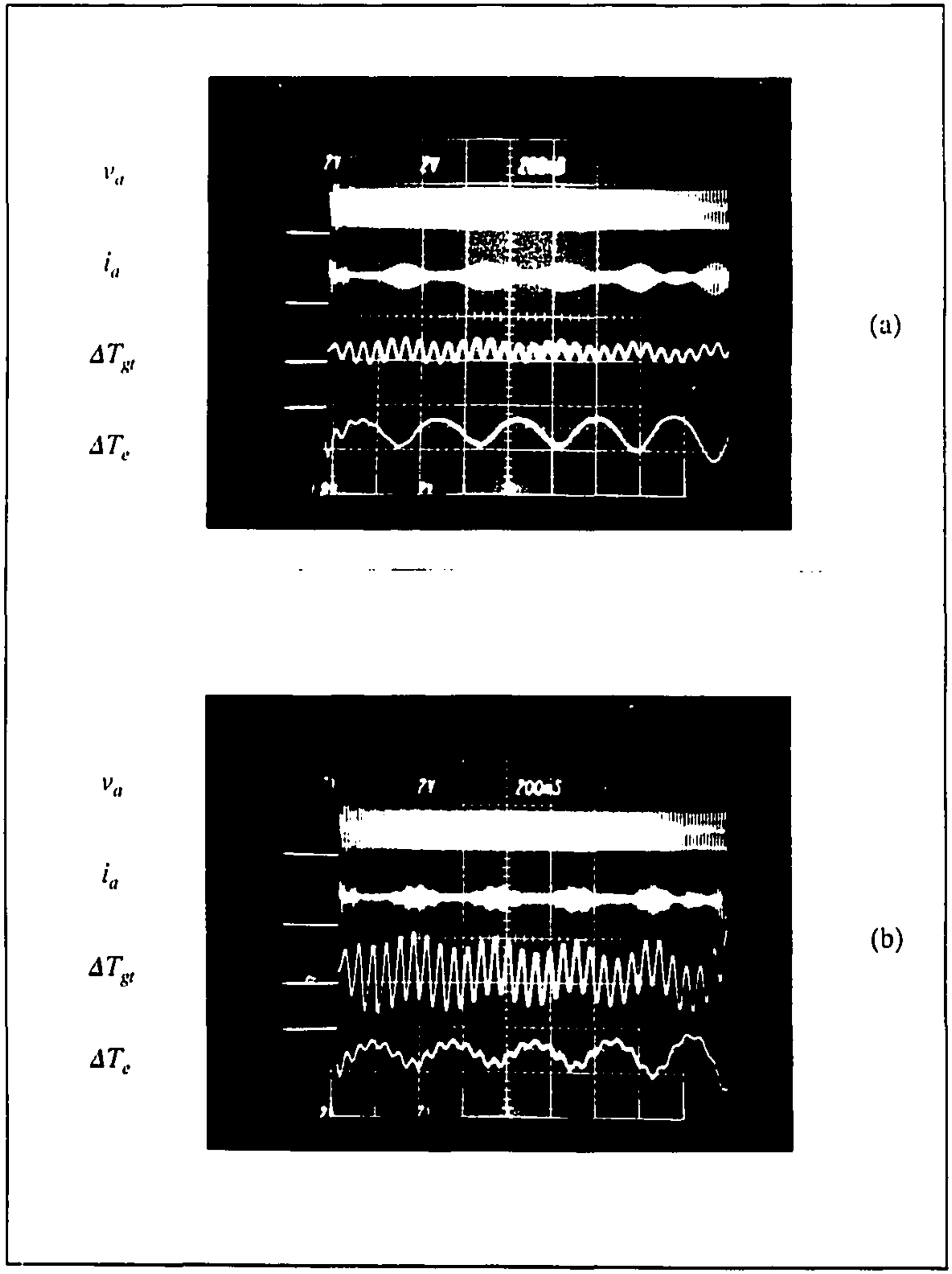


Figure 7-16 Subsynchronous resonance phenomena. (a) (H_g/H_t=2.22/0.633), (b) (H_g/H_t=0.633/2.22).

made with small perturbations in the power demand references. Instability did not occur in tests which exceeded 5 minutes.

7.5 CONCLUSION

The extensive test program, which includes symmetry checks and behavioral checks with respect to well known hunting oscillations, synchronization out-of-phase torques, and subsynchronous resonance phenomena has verified that the prototype digital TNA works reliably. The success of the digital TNA depends on: (a) the theoretical method of Capacitor Break-Point Partitioning so that different portions of the power system can be allocated to different processor-modules, (b) the architecture of the processor modules which can communicate the numerical integration results of one module to its contiguous neighbours with minimum delay, and (c) the programming for real-time concurrent multi-processing.

CHAPTER
EIGHT

CONCLUSION

8.1 INTRODUCTION

This thesis has contributed to advances in real-time parallel microprocessor control and simulation for power applications. The advances have been made possible by designing, building and programming the laboratory hardware realizations of two systems of multiple high-speed digital signal processors (DSPs):

(1) the Mark I (the single board Multiprocessor Controller) described in **Chapter 2**, and

(2) the Mark II (the multi-board Extensible Modular Multiprocessor System) described in **Chapter 5**.

The Mark I has been proven in applications as a real-time digital feedback controller of a power electronics bridge converter, as described in **Chapter 3** and **Chapter 4**. At the time when the research papers were presented in the conferences of the *IEEE Industry Application Society* [84], *Industrial Electronics Society* [109], *Power Systems Society* [83], Mark I was the first real-time parallel microprocessor controller ever to appear in the literature. The Mark II has been proven as a prototype digital Transient Network Analyzer capable of simulating in real time a small power system made up of more than one turbo-generator [111].

The advances are described and discussed below under the following headings:

(1) Advances in real-time digital control of power electronic converters.

(2) Advances in real-time digital simulation for power application.

(3) Advances in real-time parallel computation for power application.

8.2 ADVANCES IN REAL-TIME DIGITAL CONTROL OF POWER ELECTRONIC CONVERTERS

1. By using a pole-placement control strategy implemented in the single board Multiprocessor Controller of **Chapter 2**, it has been demonstrated in **Chapter 4** that the stability of the voltage-source type, pulse width modulated (PWM) converter can be substantially improved. More

importantly, the experimental results show that the voltage-source type, PWM High Voltage direct current (HVdc) transmission system is viable. Prior to the research of this thesis, the dc link capacitor size needed to stabilize the dc voltage regulator has to be very large. In addition to the high cost, the large capacitor slows the response time. The experiment, reported in **Section 4.5.2**, has proven that the system stability can be improved and the dc link capacitor size can be reduced by a factor of *1/120* through pole-placement control. This brings the voltage-source type, PWM HVdc transmission system within the economic range **[88]**.

2. The real-time computation power harnessed by the three TMS320C25s in the Multiprocessor Controller has enabled the PWM converter to be controlled digitally at a sampling rate of around *2520 Hz*. At this frequency, the harmonics up to the 7th harmonic of the *60 Hz* current and voltage waveforms can be shaped by the pulse width modulation (PWM) strategy **[110]**.

   It should be pointed out, that *2520 Hz* represents the bandwidth requirement of the fairly slow bipolar transistor (BJT) technology which unfortunately is within the human audio range. The trend is to use the faster insulated gate bipolar transistor (IGBT) or metal oxide semiconductor field effect transistor (MOSFET) to push the switching frequency above the *10,000 Hz* audio-frequency limit to avoid annoyance to hearing. The real-time computation power for such frequencies requires the Extensible Modular Multiprocessor System described in **Chapter 5.**

3. Within the sampling period of around *398 μs*, fairly sophisticated control strategies have been shown to be implementable in real time:

   (a) In **Chapter 3**, the dc link voltage is regulated by P-I feedback. The reactive power is controlled so that the bus voltage of the converter can be adjusted.

   (b) In **Chapter 4**, the algorithm of pole-placement with state feedback through an observer has been computed in real time by the three DSPs in parallel.

4. The ability to implement a control algorithm directly instead of being constrained by the "look-up table" is valuable because it gives greater flexibility to the designer and it broadens the range of applicability of the control method. However, the "look-up table" method continues to serve important functions. For instance in **Chapter 4**, it has been used to handle the nonlinearity of the system equations. Based on the samples of the feedback signal, the nearest operating point of the small perturbation linearization method in the "look-up table" is recognized. The pole-placement control is implemented by using the precalculated matrices of the linearized equations corresponding to the operating point in the "look-

up table" which are stored in the memories .

5. The programming of the pole-placement control in **Chapter 4** has to be written almost entirely in assembly language. The C language instructions have been combined with assembly language programming to convert numerical data for manipulations by the TMS320C25s which can handle fixed point arithmetic only. There is insufficient computation power to run Mark I to execute the same pole-placement control, programmed entirely in a higher language such as C.

8.3 ADVANCES IN REAL-TIME DIGITAL SIMULATION FOR POWER APPLICATION

1. The Capacitor Break-Point Partitioning method described in **Chapter 6** enables the power system equations to be broken up into computational packages, each package being assigned to a Computational Module of the Extensible Modular Multiprocessor System described in **Chapter 5**. The extensive and successful tests of the prototype digital Transient Network Analyzer (TNA), described in **Chapter 7**, are experimental proofs of:

   (a) the correctness of the Capacitor Break-Point Partitioning method,

   (b) the correctness of the extensible architecture.

   The Capacitor Break-Point Partitioning method is a new alternative method to Dommel's classic -- ElectroMagnetic Transient Program (EMTP) [38] which employs the transmission line time delays as the means of partitioning the power system.

2. **Section 6.3.2** contributes in showing how the generator equations should be pre-conditioned in the light of the Capacitor Break-Point Partitioning method, so that the numerical integrations are efficiently computed.

3. The integration step-size of *100 μs* is achieved using the modified Euler's method (predictor/corrector). The equations, which are numerically integrated by two TMS320C30s in each Computational Module, contain:

   (1) the generator model,

   (2) the torsional shaft, rotor-turbine inertia model,

   (3) the governor model,

   (4) the field exciter model,

   (5) the power system stabilizer model,

   (6) the transformer, local load, *R-L* representation of transmission lines.

4. The continuing development of the digital Transient Network Analyzer will require the other subsystems such as the transmission line module, the static VAR compensator modules, the HVdc converter station modules, etc. to be included. Presently the different modules have been developed and tested in isolation or in relatively small systems. Large scale tests,

however, can only be carried out by the utility research centres which have the resources to do them.

8.4 ADVANCES IN REAL-TIME PARALLEL COMPUTATION FOR POWER APPLICATION

1. The Multiprocessor Controller (Mark I) of **Chapter 2**, is a proven design for power electronic converter applications.

2. The Extensible Modular Multiprocessor System (Mark II) of **Chapter 5** is a proven design in applications where the computation power requirement needs flexibility and expandability.

3. The architecture based on extensible modules is very suitable for control and simulation applications. This is because the information exchanges between immediate neighbouring processing modules correspond closely to the physical interactions in engineering subsystems. The parallel programming is more straight-forward because each module is made responsible for one partition of the engineering system in carrying out all the allocated computations within the sampling time interval.

8.5 CONTRIBUTIONS TO ORIGINAL KNOWLEDGE

To the best of the author's knowledge, the following are contributions to original knowledge.

1. The research in this thesis shows how multiple high-speed digital signal processors (DSPs) can be harnessed to work concurrently for real-time control and simulation in power applications. The published papers **[83, 84, 109, 111, 112]** based on this work are the first to appear in the fields of power electronics and power systems.

2. The stability region of the voltage-source type, pulse width modulated (PWM) converter (under voltage angle control) is substantially extended and the dc link capacitor size is reduced by a factor of *1/120* by using pole-placement control with state feedback through an observer (**Chapter 4**). The reduction of the dc link capacitor size brings the voltage-source type, PWM HVdc transmission system within acceptable economic range, thus contributing to establish PWM HVdc as a potential next generation transmission system.

3. The research described in **Chapters 6** and **7** demonstrate that the power system equations can be broken up into computational packages by using the Capacitor Break-Point Partitioning method. The method is a new alternative to Dommel's classic - ElectroMagnetic Transient Program

(EMTP) [38] which employs the transmission line time delays as the means of partitioning the power system.

4. The extensive and successful tests of the prototype digital Transient Network Analyzer (TNA) given in **Chapter 7**, implemented by using the Extensible Modular Multiprocessor System of **Chapter 5**, demonstrate a new modular approach to digitize the TNA.

APPENDIX-A

Circuit and Test Parameters of the PWM converter system:

AC Circuit Inductance:

$j\omega_o L = 5\ ohm$

DC Link Capactance:

$C_{dc} = 200\ \mu F \sim 24000\ \mu F$

Valve Resistance:

$R = 1\ ohm$

Experimental Conditions:

$\omega_0 = 377\ rad/s$

$M = 0.8$ (modulation index)

$v_{ac} = 31.1\ volt$

$V_{dc} = 110\ volt$

DC Load:

$R_{load} = 22\ ohm \sim \infty$

APPENDIX-B

Test Parameters of the simulated turbo-generator from [94]:

H, τ = seconds, otherwise per unit values.

Generator:

$x_{ad}=1.550$ $x_{aq}=1.490$ $x_{kd}=0.055$ $x_{kq}=0.036$ $x_{fd}=0.101$ $x_{l}=0.150$

$r_{a}=0.001096$ $r_{fd}=0.00074$ $r_{kd}=0.0131$ $r_{kq}=0.0540$ $H_{g}=2.22$

Turbine:

$H_{t}=0.633$ $d_{g}=0.5$ $d_{t}=0.5$ $d_{tg}=0.6$ $k_{tg}=25.0$

Governor:

$\tau_{g1}=0.100$ $\tau_{g3}=0.200$ $\tau_{g4}=0.050$ $\tau_{g5}=8.000$ $F=0.300$

Excitation System:

$k_{r}=1.000$ $\tau_{r}=0.000$ $k_{a}=400.000$ $\tau_{a1}=0.200$ $\tau_{a2}=0.000$

$k_{e}=1.000$ $\tau_{e}=0.800$ $k_{f}=0.030$ $\tau_{f}=1.000$

Power System Stabilizer:

$k_{qv}=0.000$ $k_{qs}=0.050$ $\tau_{qs}=0.000$ $\tau_{q}=10.000$ $\tau_{q1}=0.020$ $\tau_{q1}'=0.300$

$\tau_{q2}=0.020$ $\tau_{q2}'=0.300$ $\tau_{q3}=0.000$ $\tau_{q3}'=0.000$ $\tau_{qv}=0.000$

Transmission Line:

$r_1=0.010$ $x_1=0.200$ $r_2=0.010$ $x_2=0.200$

Transformer:

$x_m=50.000$

System Load:

$r_n=2.000$ $x_n=0.400$ $X_c=200.000$

Local Load:

$r_{r1}=10.0$ $x_{r1}=4.83$ $r_{r2}=10.0$ $x_{r2}=4.83$

SSR Test:

$X_c=0.175$

REFERENCES

[1] Bimal K. Bose, "Microcomputer Control of Power Electronics and Drives", IEEE Press, 1987.

[2] Bimal K. Bose, "Recent Advances in Power Electronics", *IEEE Trans. on Power Electronics*, Vol.PE-7, No.1, Jan. 1992, pp.2-16.

[3] R. Comerford, "PCs and Workstations", *IEEE Spectrum*, Vol.30, No.1, Jan. 1993, pp.26-29.

[4] S. R. Bowes and A. Midour, "Microprocessor implementation of new optimal PWM switching strategies", *IEE Proc.*, Vol.135, Pt.B, No.5, Sept. 1988, pp.269-280.

[5] S. R. Bowes and M. J. Mount, "Microprocessor Control of PWM Inverters", *IEE Proc.*, Vol.128, Pt.B, No.6, Nov. 1981, pp.293-305.

[6] S. R. Bowes and A. Midoun, "New PWM Switching Strategy for Microprocessor Controlled Inverter Drives", *IEE Proc.*, Vol.133, Pt.B, No.4, July 1986, pp.237-254.

[7] S. R. Bowes and T. Davies, "Microprocessor-Based Development System for PWM Variable-Speed Drives", *IEE Proc.*, Vol.132, Pt.B, No.1, Jan. 1985, pp.18-45.

[8] B. K. Bose, "Power Electronics and AC Drives", Prentice-Hall, 1986.

[9] G. N. Acharya, S. S. Shakhawat, William Shepherd, U. M. Rao and Yuan M. Ng, "Microprocessor-Based PWM Inverter Using Modified Regular Sampling Techniques", *IEEE Trans. on Industry Applications*, Vol.IA-22, No.2, March/April, 1986, pp.286-291.

[10] Y. Y. Tzou, R. W. Wang, and Y. C. Wu, "Design and Implementation of a

Multiprocessor-Based Digital PWM Inverter Controller", pp.136-145.

[11] E. Dwyer and B.T. Ooi, "A 'Lookup' Table Based Micro-processor controller for a three Phase PWM Inverter", 5th Annual Conference, Industrial and Control Application of Microprocessors, pp.19-22, March 1979, 79CHI444.9 IECI.

[12] Bimal K. Bose, Paul M. Szczesny and Robert L. Steigerwald, "Microcomputer Control of a Residential Photovoltaic Power Conditioning System", IEEE/IAS annual meeting, Vol.IA-21, Sept./Oct. 1984, pp.852-859.

[13] Axel Brickwedde, "Microprocessor-Based Adaptive Speed- and Position-Control for Electrical Drives", IEEE/IAS annual meeting, Sept./Oct. 1984, pp.411-417.

[14] Masato Koyama, Masao Yano, Isao Kamiyama and Sadanari Yano, "Microprocessor-Based Vector Control System for Induction Motor Drives with Rotor Time Constant Identification Function", IEEE/IAS annual meeting, 1985, pp.564-569.

[15] Syunichi Hirose, Nagataga Seki, Osamu Higa, Yasukiko Miguchi, Kazutoshi Takahashi and Toshiaki Shibata, "Multi-Microcomputer-Based Controller for 12 MW GTO Power Conditioning Systems", IEEE Power Electronics Specialists Conference, PESC'88 Record, April, 1988, pp.611-618.

[16] Hideki Hashimoto, Yuko Ohno, Seiji Kondo and Fumio Harashima, "Torque Control of Induction Motor Using Predictive Observer", IEEE, 1989, pp.271-278.

[17] B. De Fornel, F. Ajuria, J. C. Hapiot, "Numerical Control of an Induction Machine Fed by a PWM Inverter", IEEE, 1985, pp.647-652.

[18] Kalyan P. Gokhale, Atsuo Kawamura and Richard G. Hoft, "Dead Beat Microprocessor Control of Inverter for Sinusoidal Output Waveform

Synthesis", IEEE, 1985, pp.28-36.

[19] Lazhar Ben-Brahim and Atsuo Kawamura, "Digital Control of Induction Motor Current with Deadbeat Response Using Predictive State Observer", IPEC, 1990, Tokoy, pp.489-496.

[20] Ajit K. Chattopadhyay and Nidhi Meher, "Microprocessor Implementation of a State Feedback Control Strategy for a Current Source Inverter-Fed Induction Motor Drive", IEEE, 1987, pp.146-155.

[21] Stephen R. Macminn and Robert J. Thomas, "Microprocessor Simulation of Synchronous Machine Dynamics in Real-Time", *IEEE Trans. on Power Systems*, Vol.PWRS-1, No.3, August 1986, pp.220-225.

[22] P. C. Krause, "Analysis of Electric Machinery", McGraw-Hill, 1986.

[23] G. Jasmin, J. P. Bowles, A Leroux, D. Mukhedkar, "Electronics simulation of a hydro-generator with static excitation", *IEEE Trans. on Power Apparatus and Systems*, Vol.PAS-100, No.9, Sept. 1981, pp.4207-4215.

[24] J. Belanger and L. Vaughan, "Hydro-Quebec Technology for The Simulation of Very Large and Complex Transmission Systems", 8th CEPSI Conference, Singapore, 1990.

[25] W. J. Giesbrecht, R. P. Wierkx, A. Gole, G. Mazur, T. Weeks, "Modelling Manitoba Hydro HVdc Controls on the Real-Time Digital Simulator", Canada Electrical Association, March 1993.

[26] P. G. Mclaren, R. Kuffel, R. Wierckx and J. Giesbrecht, "A Real Time Digital Simulator for Testing Relays", *IEEE Trans. on Power Delivery*, Vol.7, No.1, Jan. 1992, pp.207-213.

[27] P. G. Mclaren, R. Kuffel, W. J. Giesbrecht, W. Keerthipala, A. Castro, D. Fedirchuk, S. Innes, K. Mustaphi, K. Sletten, "On site relay transient testing for a series compensation upgrade", to be published.

[28] X. Wang, J. Giesbrecht, D. Woodford, L. Arendt, R. Wierckx, R. Kuffel, "Enhanced performance of a conventional HVdc analogue simulator with a real-time digital simulator", to be published.

[29] P. G. McLaren, R. Kuffel, R. Wierckx, J. Giesbrecht, L. Arendt, "Developments in Digital TNA's for Relay Testing", to be published.

[30] D. Brandt, R. Wachal, R. Valiquette and R. Wierckx, "Closed loop testing of a joint var controller using a digital real-time simulator", *IEEE Trans. on Power Systems*, Vol.6, No.3, August 1991, pp.1140-1146.

[31] R. P. Wierckx, "Fully digital real-time electromagnetic transients simulator", to be published.

[32] G. Irwin, T. Maguire, R. Wierckx and D. A. Woodford, "Developments in electromagnetic transients simulation", to be published.

[33] R. P. Wierckx, T. L. Maguire, D. A. Woodford, G. K. Rosendahl, "Canadian developments in power system simulation", IEEE Canadian Review, No. 6, Dec.1989, p15.

[34] Van-Qué Do and Alpha Oumar Barry, "A Real-Time Model of the Synchronous Machine Based on Digital Signal Processors", IEEE/PES 1992 Winter Meeting, New York, 92WM 180-0 PWRS.

[35] G. Morin, L. X. Bui, S. Casoria and J. Reeve, "Modelling of the Hydro-Quebec-New England HVDC System and Digital Controls with EMPT", IEEE/PES 1992 Winter Meeting, New York, 92 WM 274-1 PWRD.

[36] S. Kyuwa, T. Yoshida, S. Yuasa, K. Omata, K. Mitamura, "Operator Training Simulator with Real-Time Transient Stability Analysis", Proceedings of the 1993 IEEE Power Industry Computer Application Conference, pp.255-261.

[37] Hisao Taoka, Isao Lyoda, Hideo Noguchi, Nobuyuki Sato and Taro Nakazawa, "Real-Time Digital Simulator for Power System Analysis on a

Hypercube Computer", *IEEE Trans. on Power Systems*, Vol.7, No.1, Feb. 1992, pp.1-7.

[38] H. W. Dommel, "Electromagnetic Transients Program Reference Book (EMTP Book)", Bonnerille Power Administration, Portland OR, 1986.

[39] H. W. Dommel, "Digital computer solution of electromagnetic transients in single- and multi- phase", *IEEE Trans. on Power Apparatus and Systems*, Vol.PAS-88, No.4, April 1969, pp.388-399.

[40] S. R. Bowes and P. R. Clark, "Transputer Based-Harmonic-Elimination PWM Control of Inverter Drives", Conference Record of 1989 IEEE Industry Applications Society Annual Meeting, pp.744-752.

[41] B.T. Ooi and X. Wang, "Boost Type PWM HVDC Transmission System", *IEEE Trans. on Power Delivery*, Vol.6, No.4, Oct. 1991, pp.1557-1563.

[42] B.T. Ooi and X. Wang, "Voltage Angle Lock Loop Control of the Boost Type PWM Converter for HVDC Application", *IEEE Trans. on Power Electronics*, Vol.PE-5, No.2, April 1990, pp.229-232.

[43] B.T. Ooi, Y. Guo, X. Wang, H.C. Lee, H.L. Nakra and J.W. Dixon, "Stability of PWM Voltage Regulator Based on P-I Feedback", Conference Record of 4th European conference on Power Electronics and Applications, Florence, Italy, September 1991.

[44] Glenn Zorpette, "The power of parallelism", *IEEE Spectrum*, Vol.29, No.9, Sept. 1992, pp.28-33.

[45] Fumio Harashima, Seiji Kondo, Kouhei Ohnishi, Masatoshi Kajita and Motohisa Susono, "Multi-Microprocessor Based Control System for Quick Response Induction Motor Drive", IEEE/IAS annual meeting, 1984, pp.605-611.

[46] Kenji Kubo, Masahiko Watanabe, Tsutomu Ohmae and Kenzo Kamiyama, "A Fully Digitalized Speed Regulator Using Multi-microprocessor System

for Induction Motor Drives", *IEEE Trans. on Industry Applications*, Vol.IA-21, No.4, July/August, 1985, pp.1001-1007.

[47] Ying-Yu Tzou and Hsiang -Jui Wu, "Multi-microprocessor-Based Robust Control of an AC Induction Servo Motor", *IEEE Trans. on Industry Applications*, Vol.IA-26, No.3, May/June 1990, pp.441-449.

[48] S. R. Bowes and P. R.. Clark, "Transputer Based Optimal PWM Control of Inverter Drives", Conference Record of 1988 IEEE Industrial Applications Society Annual Meeting, pp.315-321.

[49] "The Transputer Family", Ref. No.72, TRN056 01, 1987, INMOS Ltd., Bristol, U. K..

[50] S. Martel, S. Lafontaine and I. Hunter, "A Parallel Processing System for Intensive I/O", Canadian Conference on Electrical and Computer Engineering, 1990, pp.27.4.1-27.4.5.

[51] S. Martel, I. Hunter and S. Lafontaine, "Integration of Parallel Processing Structures Using a Matrix Approach", Canadian Conference on Electrical and Computer Engineering, 1991, pp.36.5.1-36.5.4.

[52] "TMS320C25 User's Guide", Texas Instruments, 1986.

[53] "TMS320C3x User's Guide", Texas Instruments, 1990.

[54] J. Mogal, "Workstation Metrics", *IEEE Spectrum*, May 1992, pp.50-58.

[55] Kai Hwang, Faué A. Briggs, "Computer Architecture and Parallel Processing", McGraw-Hill Book Company, 1985.

[56] Marshall C. Yovits, "Advances in Computers", Vol.35, Academic Press, Inc., 1992, pp.255-324.

[57] G. Cybenko, D. J. Kuck, "Revolution or Evolution", *IEEE Spectrum*, Vol.29, No.9, Sept. 1992, pp.39-41.

[58] W. L. Hatcher, F. M. Brasch Jr, J. E. Van Ness, "A feasibility study for the

solution of transient stability problems by multiprocessor structures", *IEEE Trans. on Power Apparatus and Systems*, Vol.PAS-96, No.6, Nov./Dec. 1977, pp.1789-1797.

[59] J. Fong and C. Pottle, "Parallel processing of power system analysis problems via simple parallel micrccomputer structures", *IEEE Trans. on Power Apparatus and Systems*, Vol.PAS-97, No.5, Sept./Oct. 1978, pp.1834-1841.

[60] D. J. Boraty ka-Stadnicka, M. G. Lauby and J. E. Van Ness, "Converting an existing computer code to a hupercube computer", Proceedings of the 1989 IEEE Power Industry Computer Application Conference, pp.394-399.

[61] S. Y. Lee, H. D. Chiang, K. G. Lee and B. Y. Ku, "Parallel Power System Transient Stability Analysis on Hypercube Multiprocessors", *IEEE Trans. on Power Systems*, Vol.6, No.3, August 1991, pp.1337-1343.

[62] M. L. Crow, M. Ilić-Spong, "The Parallel Implementation of the Waveform Relaxation Method for Transient Stability Simulations", *IEEE Trans. on Power Systems*, Vol.5, No.3, August 1990, pp.922-932.

[63] M. La Scala, A. Bose, D. J. Tylavski and J. S. Chai, "A highly parallel method for transient stability analysis", *IEEE Trans. on Power Systems*, Vol.5, No.4, Nov., 1990, pp.1439-1446.

[64] J. E. Van Ness, G. Molina, "The Use of Multiple Facting in The Parallel Solution of Algebraic Equations", *IEEE Trans. Power Apparatus and Systems*, Vol.PAS-102, No.10, Oct. 1983, pp.3433-3438.

[65] F. M. Brasch, J. E. Van Ness and S. C. Kang, "Simulation of a Multiprocessor Network for Power System Problems", *IEEE Trans. on Power Apparatus and Systems*, Vol.PAS-101, No.2, Feb. 1982, pp.295-301.

[66] S. L. Lin and J. E. Van Ness, "Parallel Solution of Sparse Algebraic Equations", Proceedings of the 1993 IEEE Power Industry Computer

Application Conference, pp.380-386.

[67] J. E. Miller (editor), "Reactive power control in electric systems", 1982, John Wiley.

[68] H. Taoka, I. Iyoda, H. Noguchi, N. Sato, T. Nakazawa, A. Yamazaki, "Real-time digital simulator with digital/analog conversion interface for testing power instruments", Proceedings of the 1993 IEEE Power Industry Computer Application Conference, pp.336-342.

[69] X. Wang, "Advances in Pulse Width Modulation Techniques", Ph. D thesis, Departmemt of Electrical Engineering, McGill University, Montreal, Canada, 1993.

[70] Yan Guo, "Technical Manual of Multiprocessor Controller", Power Lab., Dept. of Elect. Eng., McGill University, 1990.

[71] Yan Guo, "Technical Manual of Extensible Modular Multiprocessor System", Power Lab., Dept. of Elect. Eng., McGill University, 1993.

[72] R. C. Dorf, "Modern Control Systems", Addison-Wesley Publishing Company, 1986, 1992.

[73] P. Dorato and R. K. Yedavalli, "Robust Control", IEEE Press, 1987 and 1992.

[74] K. S. Narendra and A. M. Annaswamy, "Stable Adaptive Systems", Prentice-Hall International Inc., 1989.

[75] K. J. Astrom and B. Wittenmark, "Adaptive Control", Addison-Wesley Publishing Company, 1989.

[76] J.W. Dixon, A.B. Kulkarni, M. Nishimoto and B.T. Ooi, "Characteristics of a Controlled-Current PWM Rectifier-Inverter Link", *IEEE Trans. on Industry Applications*, Vol.IA-23, No.6, Nov./Dec. 1987, pp.1022-1028.

[77] B. Friedland, "Control System Design", McGraw-Hill International Edition, 1987.

[78] Karl J. Aström and Bjorn Wittenmark, "Computer Controlled Systems: Theory and Design", Prentice-Hall, Inc., Englewood Cliffs, N. J., 1984.

[79] Ogata Katsuhiko, "Discrete-Time Control Systems", Englewood Cliffs, N. J., Prentice-Hall, 1987.

[80] J. C. Salmon, "Three Phase Current Controlled PWM Inverter Using Bipolar Transistors, Master Thesis, Department of Electrical Engineering, McGill University, Montreal, Canada, 1984.

[81] J. W. Dixon, "Boost Type PWM Rectifiers for High Power Applications", Ph. D. Thesis, Department of Electrical Engineering, McGill University, Montreal, Canada, 1988.

[82] E. W. Kimbark, "Direct Current Transmission", Vol. I, New York: John Wiley & Sons, Inc., 1971.

[83] Y. Guo, H. C. Lee, X. Wang, and B.T. Ooi, "A Multiprocessor Digital Signal Processing System for Real-Time Power Converter Application", 1991 IEEE Power Industry Computer Application Conference and *IEEE Trans. on Power Systems*, Vol.7, No.2, May 1992, pp.805-811.

[84] Y. Guo, X. Wang, H.C. Lee and B.T. Ooi, "Concurrent Microprocessor-Based PWM Converter Control", Conf. Rec. of 1990 IEEE Industry Applications Society Annual Meeting, Oct. 1990, Seattle, pp.991-996.

[85] O. I. Elgerd, "Electric Energy Systems Theory: An Introduction", New York: McGraw Hill, 1982.

[86] N. Mohan, T. M. Undeland and W. P. Robbins, "Power Electronics: Converter Applications and Design" John Wiley 1989.

[87] David C. White and H. H. Woodson, "Electromechanical Energy Conversion", John Wiley and Sons., Inc, 1959.

[88] X. Wang and B. T. Ooi, "High voltage direct current transmission system based on voltage source converters", Conference Record of PESC'90, 1990,

pp. 325-332.

[89] R. C. Durie, C. Pottle, "An extensible real-time digital transient network analyzer", IEEE/PES 1992 Winter Meeting, 92 WM 175-0 PWRS.

[90] H. Taoka, I. Iyoda, H. Noguchi, N. Sato, T. Nakazawa and A. Yamazaki, "Real-time digital simulator with digital/analog conversion interface for testing power instruments", Proceedings of the 1993 IEEE Power Industry Computer Application Conference, pp.336-342.

[91] R. M. Fano, L. J. Chu and R. B. Adler, "Electromagnetic Fields Energy and Forces", John Wiley 1963, pp.249-259.

[92] R. C. Degeneff, W. J. McNutt, W. Neugebauer, J. Panok, M. E. McCallum, C. C. Honey, "Transfermor response to system switching voltages", *IEEE Trans. on Power Apparatus and Systems*, Vol.PAS-101, No.6, June 1982, pp.1457-1470.

[93] H. S. Chandrasekharaiah, "Natural frequencies and transient responses of 3-phase rotating machine windings", *IEEE Trans. on Energy Conversion*, Vol.EC-1, No.3, Sept. 1986, pp.167-173.

[94] P. M. Anderson and A. A. Fouad, "Power System Control and Stability", Iowa State University Press 1977, pp.109-111.

[95] K. Prabhashankar and Wasyl Janischewsyj, "Digital Simulation of Multimachine Power Systems for Stability Studies", *IEEE Trans. on Power Apparatus and Systems*, Vol.PAS-87, No.1, Jan.1968, pp.73-79.

[96] D. G. Fink and H. W. Beaty, "Standard Handbook for Electrical Engineers", McGraw Hill 12th edition, Section 14-8.

[97] Yao-nan Yu, "Electric Power System Dynamics", Academic Press Inc., 1983.

[98] M. J. Maron, "Numerical Analysis, A Practical Approach", New York, Macmillan Publishing Company, 1987.

[99] A.J.Wood, "Synchronizing Out of Phase," *AIEE Trans. on Power Apparatus and Systems*, No.29, April 1957, pp.1-10.

[100] H. S. Kirschbaum, " Transient Electrical Torques of Turbine Generators During Short Circuits and Synchronizing", *AIEE Trans. on Power Apparatus and Systems*, Vol.64, Feb.1945, pp.65-70.

[101] P.C.Krause, W.C.Hollopeter, D.M.Triezenberg, P.A.Rusche, "Shaft Torques During Out-of-Phase Synchronization", *IEEE Trans. on Power Apparatus and Systems*, Vol.PAS-96, No.4, July/Aug.1977, pp.1318-1323.

[102] T. Komukai and M. Udo, " Analysis of Transient Torques in Synchronous Machine at Multi-Phase Reclosing, " *IEEE Trans. on Power Apparatus and Systems*, Vol.PAS-92, No.1, Jan./Feb.1973, pp.365-373.

[103] B. T. Ooi, " Steady state stability and subsynchronous resonance of shunt capacitor compensated transmission lines", *IEEE Trans. on Power Apparatus and Systems*, Vol.PAS-101, No.9, Sept.1982, pp.3269-3275.

[104] M. H. Banakar and B. T. Ooi, " Resonance in long distance radial transmission lines with static VAR compensation, Part II super synchronous limit", *IEEE Trans. on Power Apparatus and Systems*, Vol.PAS-103, No.6, June 1984, pp.1233-1248.

[105] C. E. J. Bowler, D. N. Ewart and C. Concordia, " Self excited torsional frequency oscillations with series capacitors", *IEEE Trans. on Power Apparatus and Systems*, Vol.PAS-92, No.5, Sept./Oct.1973, pp.1688-1695.

[106] G. Andersson, R. Atmuri, R. Rosenqvist, S. Torseng, "Influence of Hydro Units' Generator-to-Turbine Inertia Ratio on Damping of Subsynchronous Oscillations", *IEEE Trans. on Power Apparatus and Systems*, Vol.PAS-103, No.8, Aug.1984, pp.2352-2361.

[107] K. Murotani, M. Asano, " Subsynchronous Resonance Oscillations of Series-Compensated Transmission System and Their Suppression",

Electrical Engineering in Japan, Vol.96, No.6, 1976, pp.113-121.

[108] M. M. Sartawi, " Suppression and Avoidance of Subsynchronous Resonance in Synchronous Generator", Ph. D Thesis, June 1978, McGill University, pp.240-244.

[109] Y. Guo, X. Wang, H.C. Lee and B.T. Ooi, "Pole-Placement Control of Voltage-Regulated PWM Rectifier Through Real-Time Multi-processing", Proceedings IECON'91, 1991 International Conference on Industrial Electronics, Control and Instrumentation, October 1991, Kobe, Japan, pp.513-518, also accepted for publications in *IEEE Trans. on Industrial Electronics.*

[110] X. Wang and B. T. Ooi, "Real-Time Multi-DSP Control of Three Phase Current-Source Unity Power Factor PWM Rectifier", accepted by the *IEEE Trans. on Power Electronics* for publication.

[111] Y. Guo, B.T. Ooi and H.C. Lee, "Integration of Turbo-Generator Modules in Digital Transient Network Analyzer", Proceedings of 1993 IEEE Power Industry Computer Application Conference pp.343-349 and accepted by *IEEE Trans. on Power Systems.*

[112] Y. Guo, H.C. Lee and B.T. Ooi, "Extensible Digital Signal Processing Modules for Real-Time Control and Simulation", accepted by 1993 International Conference on Industrial Electronics, Control and Instrumentation (IECON), Hawaii, Nov. 15, 1993.