

PARALLEL PROCESSING WITH MULTIPLE MICROPROCESSORS
FOR POWER SYSTEM ANALYSIS

by

ISAIAS ELIZARRARAZ ALCARAZ
B.Sc.(Eng.)

Thesis submitted for the degree of
Doctor of Philosophy
in the Faculty of Engineering

Department of Electrical Engineering
Imperial College of Science and Technology
University of London

London, August 1983

ABSTRACT

The applications of microprocessors and parallel processing techniques for the simulation and analysis of large power systems are discussed in this thesis. It is shown that moderately priced microprocessor units can be effectively employed for both the solution of large power system problems and the development of efficient computational tools for system analysis.

An array of multiple microprocessors has been employed for the analysis of large power systems using off-the-shelf hardware units based on the Texas TMS9900, 16-bit microprocessor. The software is implemented in the high level programming language PASCAL and only special purpose routines such as the microprocessor intercommunication procedures are written in assembly language.

The capabilities of a microprocessor unit for the solution of power system problems are explored using a software package implemented for steady state analysis. This package exploits dynamic data structures in PASCAL when programming the sparsity-oriented techniques required for the efficient handling of large electric power networks.

Multiple microprocessors and piecewise methods are applied to the detailed simulation and analysis of large power systems. A decomposition technique is proposed to divide a network into subnetworks such that each subnetwork can be solved independently on a microprocessor. Parallel algorithms that fully exploit sparsity at all levels during the solution process of a partitioned system are examined and different approaches to their implementation are evaluated using test systems of different sizes and characteristics.

Descriptions are given of the multiple microprocessor configurations, task distribution approaches, synchronization schemes, interchange of information procedures and algorithms implemented for the analysis of bulk power systems using distributed microprocessors.

Parallel processing algorithms for transient stability, load flow, contingency analysis and short circuit computations have been implemented using the array of microprocessors and a versatile user

interface has been incorporated for interactive purposes.

The array of microprocessors can be used as a computational aid in the analysis of a wide spectrum of power system conditions and its usefulness has been demonstrated in a range of applications including planning, design, teaching and research.

ACKNOWLEDGEMENTS

The work presented in this thesis was carried out under the supervision of Dr. M.J. Short, Ph.D., B.Sc., D.I.C., C.Eng., M.I.E.E., M.I.E.E.E., to whom I am greatly indebted for his helpful guidance and constant encouragement.

I would like to thank Dr. B.J. Cory, D.Sc.(Eng.), A.C.G.I., C.Eng., F.I.E.E., Sen.M.I.E.E.E., Reader in Electrical Engineering, for his valuable suggestions and keen interest during the development of this project.

I wish to express my most sincere appreciation to my friends in the Power System Simulation Group R.B.I. Johnson and R. Lopez for their contributions and useful discussions.

I would also like to extend my appreciation to my colleagues in the Escuela de Ingenieria Electrica de la Universidad Michoacana, Morelia, Mexico, for their continuous motivation to the realization of this work. Special thanks are due to M.Sc. Miguel Angeles and M.Sc. Salvador Acha for their constant stimulus and the most invaluable help received throughout all the stages of this project.

I am grateful to the authorities of the Universidad Michoacana and the Secretaria de Educacion Publica, Mexico, for the financial support that has made possible my stay at the Imperial College.

The multiple microprocessor power system simulator project at Imperial College has been supported by the Science and Engineering Research Council in the U.K.

A MIS PADRES

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	2
ACKNOWLEDGEMENTS	4
LIST OF TABLES	12
LIST OF FIGURES	14
LIST OF SYMBOLS AND ABBREVIATIONS	18
CHAPTER 1: INTRODUCTION	22
1.0 GENERAL	22
1.1 PARALLEL PROCESSING BY MULTIPLE	
PROCESSOR STRUCTURES	23
1.1.1 Multiple Processors for NBDF Algorithms.	24
1.1.2 Multiple Processors for BBDF Algorithms.	24
1.1.3 Multiple Processors for NBBDF Algorithms	26
1.1.4 A Partitioned Ring Structure	26
1.2 MICROPROCESSOR ARRAYS FOR POWER SYSTEM ANALYSIS	29
1.2.1 General Objectives	30
1.3 POWER SYSTEM ANALYSIS	32
1.3.1 Steady State Analysis	32
1.3.2 Dynamic State Analysis	33
1.4 THESIS OVERVIEW	34
1.4.1 Synopsis of Chapters	37
CHAPTER 2: FORMULATION OF THE MATHEMATICAL	
MODEL STRUCTURE	39
2.0 INTRODUCTION	39
2.1 THE POWER SYSTEM	39
2.2 INTERCONNECTED TRANSMISSION NETWORK	41
2.2.1 Representation of Unbalanced Networks	41
2.2.2 Representation of Transmission Lines	
and Transformers	43
2.2.3 Network Mathematical Formulation	46

2.3	GENERATION PLANT		47
2.3.1	Generator Model		49
2.3.1.1	Damping		51
2.3.1.2	Saturation		52
2.3.1.3	Saliency		52
2.3.2	Network/Generator Interface		54
2.3.3	Excitation system		56
2.3.4	Power System Stabiliser		56
2.3.5	Speed Governor Systems		58
2.3.6	Turbines		58
2.3.7	Boiler		58
2.4	ELECTRICAL LOADS		62
2.5	RELAY REPRESENTATION		62
2.5.1	Over/Under frequency and Under voltage Relays		63
2.5.2	Distance Relays		63
2.6	SYSTEM CONTROL CENTRE		64
2.7	COMMENTS		65
CHAPTER 3:	SOFTWARE AND HARDWARE CHARACTERISTICS		66
3.0	INTRODUCTION		66
3.1	PASCAL FEATURES		66
3.1.1	Control Structures		67
3.1.2	Data Structures		68
3.1.3	Dynamic Data Structures		69
3.2	SOFTWARE DEVELOPMENT SYSTEM		70
3.2.1	Code Generation		71
3.3	MICROPROCESSOR UNITS		71
3.4	MICROPROCESSORS INTERCONNECTION STRUCTURE		72
3.5	INTER-PROCESSOR COMMUNICATION PROCEDURES		74
3.5.1	Initialisation Procedure		75
3.5.2	Sending/Receiving Procedures		75
3.6	OBSERVATIONS		78
CHAPTER 4:	STEADY STATE POWER SYSTEM ANALYSIS		
	ON A SINGLE MICROPROCESSOR		80
4.0	SUMMARY		80

4.1	SPARSITY PROGRAMMING TECHNIQUES USING	
	DDS IN PASCAL	80
4.1.1	Sparse Storage Scheme and	
	Allocation Procedure	81
4.1.2	Comparing the Dynamic and Array	
	Implementation of Linked Lists	84
4.1.3	Ordering, Factorisation and	
	Solution Procedures	84
4.1.4	Computational Results on a	
	TMS9900 Microprocessor	88
4.2	LOAD FLOW STUDIES	89
4.2.1	Fast Decoupled Load Flow	90
4.2.2	Computational Results	91
4.3	CONTINGENCY ANALYSIS	92
4.3.1	Branch Outage	95
4.3.2	Shunt Capacitor or Reactor Switching	95
4.3.3	Multiple Branch Outages	95
4.3.4	Generator/Load Changes	96
4.3.5	Application and Results	97
4.4	SOFTWARE PACKAGE ORGANIZATION	97
4.5	CONCLUSIONS	99

CHAPTER 5:	MULTIPLE MICROPROCESSORS FOR THE	
	SOLUTION OF LARGE POWER NETWORKS	100
5.0	SUMMARY	100
5.1	THE MULTIPLE MICROPROCESSOR	
	CONFIGURATION ADOPTED	100
5.2	DECOMPOSITION OF NETWORKS	101
5.3	PIECEWISE TECHNIQUES	102
	5.3.1 The Proposed Technique	105
5.4	SOLUTION ALGORITHMS	111
5.5	PARALLEL SOLUTION ALGORITHMS USING	
	MULTIPLE MICROPROCESSORS	113
	5.5.1 Factorisation Procedure	113
	5.5.2 Solution Procedure	115
	5.5.3 Investigation and Results on the Appli-	
	cation of Parallel Solution Algorithms	115
5.6	APPLICATION TO THE FAST DECOUPLED LOAD FLOW	120

5.6.1	Computational Algorithm		122
5.6.2	Results Using Multiple Microprocessors	.	125
5.7	CONCLUSIONS		127
CHAPTER 6:	FAULT ANALYSIS USING MICROPROCESSORS		128
6.0	INTRODUCTION		128
6.1	FAULTED NETWORKS		128
6.1.1	Fault Representation		130
6.2	SOLUTION ALGORITHM		134
6.2.1	System-Wide Short Circuit Studies	.	137
6.2.2	Simultaneous Faults		137
6.3	ALGORITHM IMPLEMENTATION AND RESULTS	. .	138
6.4	CONCLUSIONS		144
CHAPTER 7:	TRANSIENT STABILITY ANALYSIS USING MULTIPLE MICROPROCESSORS		146
7.0	INTRODUCTION		146
7.1	TRANSIENT STABILITY PROBLEM		147
7.2	SOLUTION TECHNIQUES		148
7.2.1	Integration Techniques		149
7.2.2	Integration of State Equations by Model Decoupling		151
7.2.3	The Integration Scheme Implemented	.	152
7.2.4	Network Solution		153
7.2.5	Simulation of Disturbances		156
7.3	APPLICATION OF PARALLEL PROCESSING TECHNIQUES	.	157
7.3.1	Algorithm Implementation		158
7.4	DYNAMIC SIMULATION OF LARGE POWER SYSTEMS	.	161
7.5	TESTS AND RESULTS		170
7.6	CONCLUSIONS		180
CHAPTER 8:	MULTIPLE MICROPROCESSOR POWER SYSTEM SIMULATOR		181
8.0	INTRODUCTION		181
8.1	THE MULTIPLE MICROPROCESSOR DYNAMIC SIMULATOR	.	182
8.1.1	Simulator Features		183
8.1.2	Simulator Architecture		186
8.1.3	User/Simulator Interactive Interface	.	188

8.2	SIMULATOR APPLICATIONS		191
8.2.1	System Planning and Design Studies	.	193
8.2.1.1	Example		193
8.2.2	System Operation		193
8.2.2.1	Example		196
8.2.3	Teaching and Research		200
8.2.3.1	Example		200
CHAPTER 9:	CONCLUSIONS		207
9.1	GENERAL CONCLUSIONS		207
9.2	CONTRIBUTIONS		213
9.3	SUGGESTIONS FOR FURTHER RESEARCH		215
REFERENCES			218
APPENDIX A:	COMPENSATION METHODS		228
A.0	INTRODUCTION		228
A.1	THE COMPENSATION METHOD		228
A.2	COMPENSATION SCHEMES		235
A.2.1	Post-Compensation Methods		236
A.2.2	Pre-Compensation Methods		236
A.2.3	Mid-Compensation Methods		236
APPENDIX B:	NUMERICAL EXAMPLE FOR THE SOLUTION IN PARALLEL OF PARTITIONED SYSTEMS		238
B.0	INTRODUCTION		238
B.1	THE SAMPLE NETWORK		238
B.2	SOLUTION IN PARALLEL OF A PARTITIONED SYSTEM	.	240
APPENDIX C:	DESCRIPTION OF TEST SYSTEMS		246
C.0	INTRODUCTION		246
C.1	5-BUS TEST SYSTEM		247
C.2	WSCC 9-BUS TEST SYSTEM		249
C.3	IEEE 14-BUS TEST SYSTEM		251
C.4	IEEE 24-BUS RELIABILITY TEST SYSTEM		253
C.5	IEEE 30-BUS TEST SYSTEM		256
C.6	NEW ENGLAND 39-BUS TEST SYSTEM		259
C.7	IEEE 57-BUS TEST SYSTEM		262

C.8 IEEE 118-BUS TEST SYSTEM		266
APPENDIX D: DATA FOR THE DYNAMIC SIMULATION OF TEST SYSTEMS		273
D.0 INTRODUCTION		273
D.1 WSCC 9-BUS TEST SYSTEM		275
D.2 NEW ENGLAND 39-BUS TEST SYSTEM		276

LIST OF TABLES

3.1	Average Execution Times for Floating Point Arithmetic Operations
4.1	Characteristics of Test Networks
4.2	Execution Times on the TMS9900 Microprocessor for the Sparsity Procedures Implemented in PASCAL
4.3	Load Flow Execution Times on the TMS9900 Microprocessor
4.4	Execution Times for Contingency Analysis on the TMS9900 Microprocessor
5.1	Substitution Procedures
5.2	Network Partitioning Characteristics
5.3	Execution Times for Parallel Solution Algorithms Using 2 MPUs
5.4	An Extension that Further Parallelises the Performance of Block Matrix Algorithms.
6.1	Fault Parameters
7.1	Execution times for the Dynamic Simulation of Bulk Power Systems Using Four MPUs.
7.2	Computational Efficiencies and Speeding-up Factors for the Dynamic Simulation of Bulk Power Systems Using Four MPUs.
8.1	Power System Scenario
C.1	Network Data. 5-bus Test System
C.2	Base-Case Load Flow. 5-bus Test System
C.3	Network Data. 9-bus Test System
C.4	Base-Case Load Flow. 9-bus Test System
C.5	Network Data. 14-bus Test System
C.6	Base-Case Load Flow. 14-bus Test System
C.7	Network Data. 24-bus Test System
C.8	Base-Case Load Flow. 24-bus Test System
C.9	Network Data. 30-bus Test System
C.10	Base-Case Load Flow. 30-bus Test System
C.11	Network Data. 39-bus Test System
C.12	Base-Case Load Flow. 39-bus Test System
C.13	Network Data. 57-bus Test System

C.14	Base-Case Load Flow. 57-bus Test System
C.15	Network Data. 118-bus Test System
C.16	Base-Case Load Flow. 118-bus Test System
D.1	Generator Data. 9-bus Test System
D.2	Excitation System Data. 9-bus Test System
D.3	Power System Stabiliser Data. 9-bus Test System
D.4	Speed Governing System Data. 9-bus Test System
D.5	Turbine Data. 9-bus Test System
D.6	Generator Data. 39-bus Test System
D.7	Excitation System Data. 39-bus Test System
D.8	Power System Stabiliser Data. 39-bus Test System
D.9	Speed Governing System Data. 39-bus Test System
D.10	Turbine Data. 39-bus Test System

LIST OF FIGURES

- 1.1 Special Matrix Forms
- 1.2 Multiprocessing Structure for NBDF Algorithms
- 1.3 Multiple Processing Structures for BBDF Algorithms
- 1.4 Multiple Processing Structure for NBBDF Algorithms
- 1.5 A Partitioned Ring Structure
- 1.6 Multiple Microprocessor Structure for a Dynamic Power System Simulator
- 1.7 Time Ranges of Power System Dynamic Phenomena
- 2.1 Conceptual Model of an Interconnected Power System
- 2.2 Representation of a Balanced Three-phase Element
- 2.3 Representation of Unbalanced Networks
- 2.4 Sequence Networks
- 2.5 Representation of Transmission Network elements
- 2.6 Generation Plant Model
- 2.7 Synchronous Machine Representation
- 2.8 Block Diagram for a Two-axis Generator Model
- 2.9 Two Frames of Reference for Phasor Quantities for a Voltage V_i
- 2.10 Equivalent Circuit for Generator
- 2.11 Excitation System
- 2.12 Power System Stabiliser
- 2.13 Governor Speed Models
- 2.14 Turbines Models
- 2.15 Boiler Model
- 2.16 Operating Characteristics of an Impedance Relay
- 3.1 Four Basic Control Structures
- 3.2 Examples of Linked Lists
- 3.3 Interconnection Structure of 4 MPU's Suitable for BBDF Algorithms
- 3.4 Communication Buffer
- 3.5 Example Illustrating the Application of PSEND/GETMSG
- 3.6 Coordination Scheme Between Two MPU's to Avoid Losing any Transmitted Message Using PSEND/GETMSG

- 4.1 Allocation Procedure for the Developed Semi-dynamic Sparse Storage Scheme
- 4.2 A Sample Network
- 4.3 Example Illustrating the Sparse Storage of [Y] Elements
- 4.4 Orientation of Networks
- 4.5 Execution times on the TMS9900 Microprocessor for the Sparsity Procedures Implemented
- 4.6 Software Package Organization for Power System Steady State Analysis
- 5.1 Decomposition Pattern for the IEEE 30-bus System
- 5.2 Decomposition of an Electric Power Network
- 5.3 Algorithm for the Factorisation in Parallel of a BBDF Matrix Using Multiple Microprocessors
- 5.4 Parallel Solution Algorithm Using Multiple Microprocessors
- 5.5 Solution of Test Networks Using up to 4 MPUs
- 5.6 Parallel FDLF Algorithm for the Application of Multiple Microprocessors
- 5.7 Speedup for the Load Flow Solution of Test Systems Using Multiple Microprocessors
- 6.1 Representation of Faults
- 6.2 Algorithm Implemented for Fault Analysis Studies
- 6.3 Hardware Configuration for Fault Analysis Studies
- 7.1a Representation of a 3-bus Power System Using Microprocessors
- 7.1b Task Distribution Among the MPUs for Dynamic Simulations
- 7.2 Initialisation Stage for Dynamic Simulation Studies
- 7.3 Simulation Step when a Disturbance Occurs in the System
- 7.4 Main Loop for Transient Stability Simulation
- 7.5 Configuration of Microprocessors for the Simulation of Bulk Power Systems
- 7.6 Parallel Algorithm for the Dynamic Simulation of Bulk Power Systems Using 4 MPUs
- 7.7 Decomposition of the 3-generator, 9-bus Test System
- 7.8 Decomposition of the 10-generator, 39-bus Test System
- 7.9 Generator Rotor Angles for the 9-bus, 3-generator Test System
- 7.10 Generator Rotor Angles for the 39 bus, 10-generator Test System
- 7.11 Timing Diagram for a Simulation Step of the 9-bus System Using 4 MPUs

- 7.12 Timing Diagram for a Simulation Step of the 39-bus System Using 4 MPUs
- 8.1 Representation of a Bulk Power System Using Five Microprocessor Units
- 8.2 Hardware Configuration of the Microprocessor-Based Power System Simulator
- 8.3 Display Page for an MPU Representing Generators
- 8.4 Display Page for an MPU Representing a Network Area
- 8.5 Generator Rotor Angles in Area 2. Three-phase fault on line 26-29 side of bus 29 cleared at 100 ms
- 8.6 Generator Rotor Angles in Area 2. Three-phase fault on line 26-29 side of bus 29 cleared at 120 ms
- 8.7 Generator Frequencies in the System Area 1. Sudden loss of the generation unit 10
- 8.8 Generator Frequencies in the System Area 2. Sudden loss of the generation unit 10
- 8.9 Generator Rotor Angles in the System Area 1. Sudden loss of the generation unit 10
- 8.10 Generator Rotor Angles in the System Area 2. Sudden loss of the generation unit 10
- 8.11 Generator Electrical Powers in the System Area 1. Sudden loss of the generation unit 10
- 8.12 Generator Electrical Powers in the System Area 2. Sudden loss of the generation unit 10
- 8.13 Generator Rotor Angles in the System Area 2 for the Sequence of Events Listed in the Table 8.1
- 8.14 Generator Rotor Angles in the System Area 1 for the Sequence of Events Listed in the Table 8.1
- 8.15 Generator Frequencies in the System Area 2 for the Sequence of Events Listed in the Table 8.1
- 8.16 Generator Electrical Powers in the System Area 2 for the Sequence of Events Listed in the Table 8.1
- 8.17 Generator Mechanical Powers in the System Area 2 for the Sequence of Events Listed in the Table 8.1
- 8.18 Generator Excitation Voltages in the System Area 2 for the Sequence of Events Listed in the Table 8.1

- 8.19 Generator Terminal Voltages in the System Area 2 for the Sequence of Events Listed in the Table 8.1
- A.1 General Thevenin Equivalent Representing an Unmodified Network
- A.2 Representation of Modifications in a Power Network
- A.3 Representation of a Modified Network Using Compensation Methods
- B.1 Sample Network
- B.2 Decomposition of Sample Network
- C.1 5-bus Test System
- C.2 WSCC 9-bus Test System
- C.3 IEEE 14-bus Test System
- C.4 IEEE 24-bus Reliability Test System
- C.5 IEEE 30-bus Test System
- C.6 New England 39-bus Test System
- C.7 IEEE 57-bus Test System
- C.8 IEEE 118-bus Test System
- D.1 Generating Plant Model

LIST OF SYMBOLS AND ABBREVIATIONS

NOTE: Most of the symbols are defined the first time they appear in the text.

abc	Phase variables
A_g, B_g	Generator saturation coefficients
[B]	Susceptance matrix
BBDF	Bordered block diagonal form matrix
BDF	Block diagonal form matrix
cos	Cosine function
[C]	Connectivity matrix
d	Direct axis
D	Damping coefficient
DDS	Dynamic data structures
[D]	Diagonal matrix
exp	Exponential function
f	Frequency
FDLF	Fast decoupled load flow
H	Generator inertia
im	Imaginary part of a complex number
[I]	Vector of bus currents
$[I_g]$	Generation bus current
$[I_l]$	Load bus current
$[I_t]$	Compensation current vector due to network tearing
$[I]^+$	Normalised current
I_d, I_q	Current d-q components
I_f	Fault current

I_{fa}	Generator fictitious current
I_{gc}	Generator compensation current
I_{sal}	Saliency component of the generation current
I/O	Input and output
$j = \sqrt{-1}$	Imaginary operator
[L]	Lower triangular matrix
MPU	Microprocessor unit
NBDF	Near block diagonal form matrix
NBBDF	Nested bordered block diagonal form matrix
P_f	Active load frequency sensitivity parameter
P_v	Active load voltage sensitivity parameter
P	Bus active power
P_a	Acceleration power
P_e	Electrical power
P_g	Generation active power
P_l	Load active power
P_m	Mechanical power
P^{sp}	Scheduled active power
PIO	Parallel I/O ports
q	Quadrature axis
Q	Bus reactive power
q_f	Reactive load frequency sensitivity parameter
q_v	Reactive load voltage sensitivity parameter
Q_g	Generation reactive power
Q_l	Load reactive power
Q^{sp}	Scheduled reactive power
r.h.s.	Right-hand side
re	Real part of a complex number

sal	Saliency
sin	Sine function
$S = P+jQ$	Complex power
S_g	Generator saturation factor
t	Transformer tap position
T	Time constant
T'_{do}, T'_{qo}	Transient open circuit time constants
[T]	Transformation matrix
[U]	Upper triangular matrix
[V]	Vector of bus voltages
[V']	Postcontingency voltage
[ΔV]	Voltage deviation vector
V	Voltage magnitude
V^o	Voltage initial conditions
V_d, V_q	Voltage d-q components
$\omega = 2\pi f$	Angular speed
X_d, X_q	Synchronous reactances
X'_d, X'_q	Transient reactances
[X _t]	Reactance matrix of torn branches
[X _{tt}]	Reactance intersubdivision matrix
[X]	State variables
$y = g+jb$	Branch admittance
y'	Total line charging admittance
y_{sh}	Shunt reactor admittance
[y _t]	Admittance matrix of torn branches
Y_{fa}	Generator fictitious admittance
Y_l	Load admittance
[Y]	Admittance matrix

$[Y_{bdf}]$	Block-diagonal-form admittance matrix
$[Y_{tb}]$	Cut admittance matrix
$z = r+jx$	Branch impedance
$[z_t]$	Impedance matrix of torn branches
Z_{ef}	Equivalent fault impedance
$[Z]$	Impedance matrix
$[Z_{\cdot i}]$	Column i of the impedance matrix
$[Z_{ff}]$	Thevenin's equivalent fault impedance
$[Z_{tt}]$	Interarea impedance matrix
(+)	Positive sequence
(-)	Negative sequence
(o)	Zero sequence
$[]^*$	Complex conjugate
$[]^t$	Transpose matrix
$[]^{-1}$	Inverse matrix
Δ	Increment
$\Delta P, \Delta Q$	Active and reactive mismatches
Δt	Time step
δ	Rotor angle
λ	Generator participation factor
θ	Bus voltage angle
θ_{ij}	Voltage angle across branch ij

CHAPTER ONE

INTRODUCTION

1.0 - GENERAL

The continuing increase in the complexity of interconnected power systems and the need for more detailed planning and operational studies demand the application of modern approaches and powerful techniques to perform system analysis at the lowest possible cost.

Recent investigations have demonstrated the potential of using modern computer hardware architectures to reduce computational difficulties and expense caused by the dimensions of large scale studies [6]-[15]. Pritchard and Pottle [10] estimated that load flow solutions of a 1000-bus base case could be obtained in less than one-half second on an AP-120B array processor. Brash et al. [13] evaluated on a CDC 6600 mainframe computer the performance of a multiprocessor network of up to fifty processors for the solution of the transient stability problem of a 1732-bus, 398-generator system.

Up to the present, most of the proposed multiple processor structures to reduce the computation burden of power system analysis have been confined to feasibility studies [11]-[15] simulating the hardware performance on a serial computer, since the proposed parallel processing structures have not yet been built.

The technological revolution taking place in the computer hardware industry has contributed to a proliferation of microprocessors which offer a high computational power for a price considerable lower than that of standard mainframe computers. The downward trend in price and the constant improvements in the capabilities of microprocessors now offer the attractive alternative of employing several microprocessors in the construction of multiple processor structures. This gives the flexibility of tailoring the hardware architecture to a particular problem.

The availability of complete processors as low-cost building blocks has brought increased attention to multiple processor structures [1],[7]-[9]. A variety of organizations is represented in Pottle et

al.[7], Fuller et al.[8], Civera et al.[9], and examples are given to highlight some of the important considerations in implementing and programming multiprocessor systems. The implementation of distributed processing schemes requires the design of processor interconnection structures and the development of hierarchical computation, decomposition techniques, scheduling schemes for task distribution and the transmission of information between the processors.

The main objective of the work presented in this thesis is to investigate the application of parallel processing techniques for the analysis of large power systems using an existing array of multiple microprocessors.

1.1 - PARALLEL PROCESSING BY MULTIPLE PROCESSOR STRUCTURES

The concept of decomposition of large-scale systems was used by Hachtel and Sangiovanni-Vincentelli in [5] to classify and describe various "third-generation simulators" in terms of the role played by certain matrix forms in their formulation. This classification is adopted in this work to describe multiple processor structures.

The goal of multiple processor structures is to reduce the time required to solve a problem to $(1/n)^{th}$ of the time required for a single processor, where n is the number of processors. This goal is never achieved except for a totally disconnected problem. At some point in the execution process, one or more of the processors will be idle waiting for the data to be computed by another processor. Maximum efficiency of the parallel execution is gained if the set of equations being solved is grouped into nearly equal-sized blocks with a minimum of interconnections and each block is assigned to one or more processors for solution (see, for example, Hatcher et al.[13] or Fong and Pottle [14]). The variables that cannot be ordered into any blocks are known as the cut-set variables because by removing these variables the equations are cut into totally disconnected sets.

It is convenient to represent the variables in matrix form and associate a graph with the matrix to apply ordering or clustering algorithms for optimum decomposition. These algorithms permute the system matrix into special matrix forms and the resulting matrix block structure may be used for the parallelization of computer architectures and for work scheduling purposes [5]. The Near Block Diagonal Form (NBDF) and the Bordered Block Diagonal Form (BBDF) were considered by

Hatcher et al.[13] to propose and evaluate different multiple processor structures for the solution of the transient stability problem. Fong and Pottle [14] considered a BBDF and a Nested Bordered Block Diagonal Form (NBBDF) to propose multiple microcomputer structures. The NBBDF was obtained by applying recursive clustering to individual blocks of a BBDF matrix. These matrix forms are illustrated schematically in Fig. 1.1.

Recently proposed parallel processing structures for power system analysis are briefly reviewed to prepare the ground for this research.

1.1.1 - Multiple Processors for NBDF Algorithms

The simplest multiple processor structure for NBDF algorithms is one using a single shared memory connected by a common bus to a number of processors [13] as depicted in Fig. 1.2, where each processor unit is formed by the processing element and its own memory unit.

In the structure presented in Fig. 1.2, the controller ensures that the execution process proceeds in the prescribed fashion and synchronizes the processors in certain portions of the code which must use the same data. The arbiter is responsible for ensuring that only one processor receives permission to use the bus during one memory cycle. The shared memory is used for the exchange of data and the rate with which the data must be shared depends on the number of processors sharing the data. The processors will be delayed by the process of sharing data when two or more processors attempt to access shared memory on the same cycle. Improvements could be obtained by increasing the access rate of the shared memory or by using multiple copies of the shared memory so that each processor would have several buses, each of which is connected to an individual shared memory.

1.1.2 - Multiple Processors for BBDF Algorithms

The BBDF algorithms require the processors to alternate active periods of execution. With only one set of processors active at a time, the difficulty of sharing memory is eliminated. If the cutset variables are assigned to a single processor, this processor can provide the control functions required in the structure and must be able to access all other processors.

Happ [16] described the implementation of BBDF algorithms considering a multicomputer configuration based on the computers

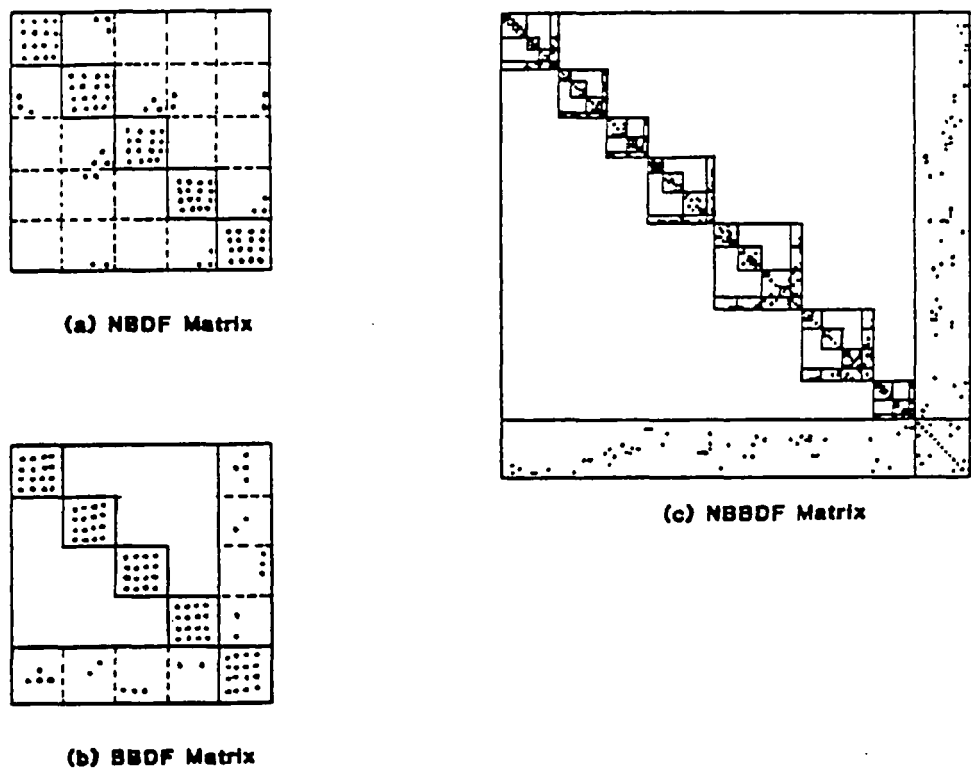


Fig. 1.1: Special Matrix Forms.

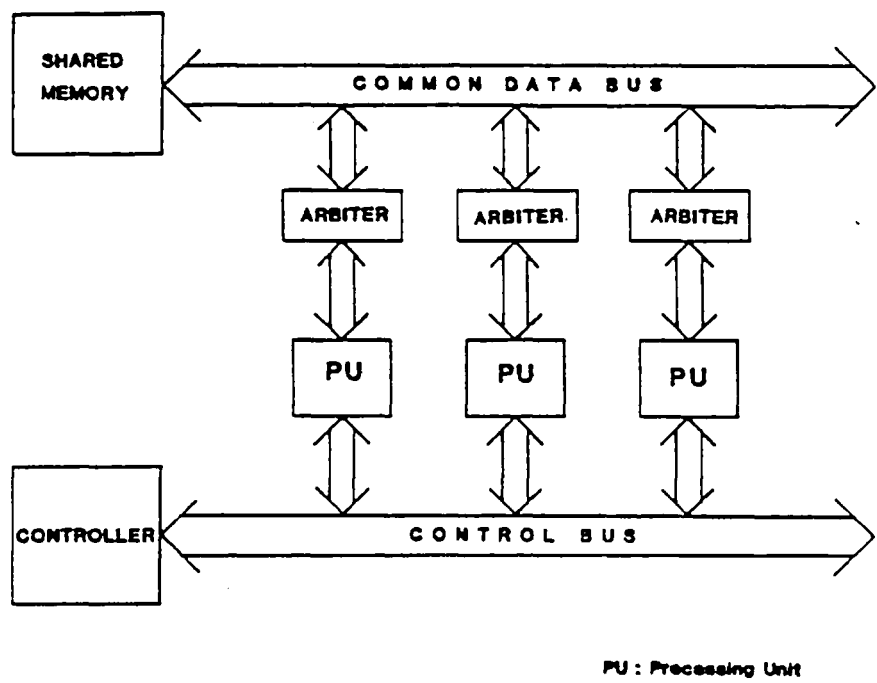


Fig. 1.2: Multiprocessing Structure for NBDF Algorithms.

utilized in the operation of electric power companies. The computer in each area of the system solves the equations of its area model and the computer in the system control centre solves the cutset equations. The computers are linked to each other by communication channels. The computational speed of this configuration which is shown in Fig. 1.3a, is a function of the area computers and the communication channels. If one of the area computers has sufficient capability, it may be used also as a central computer and vice versa.

Fig. 1.3b shows the parallel structure considered in Hatcher et al.[13] for BBDF algorithms. A processor is used to solve the cutset equations and all the processors are connected through a common bus. The bus is inactive when the cutset processor is inactive so the other processors cannot interfere with each other.

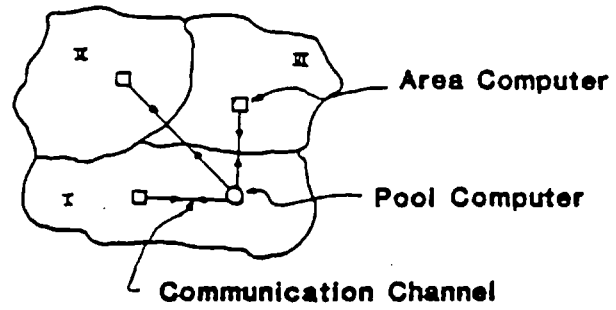
In Fong and Pottle [14], the use of a cutset processor is eliminated and the cutset variables are distributed among all processors which are connected by a single common data bus. This structure is illustrated in Fig. 1.3c. The controls required for synchronization are provided by the communication procedure and the solution algorithm. At any one time only one processor may have control of the bus to broadcast data to the other processors. The solution algorithms are arranged so that, at the time data are broadcast, those processors needing them will be waiting. After each data broadcast, the processors will be working asynchronously up until further data are needed to continue execution.

1.1.3 - Multiple Processors for NBBDF Algorithms

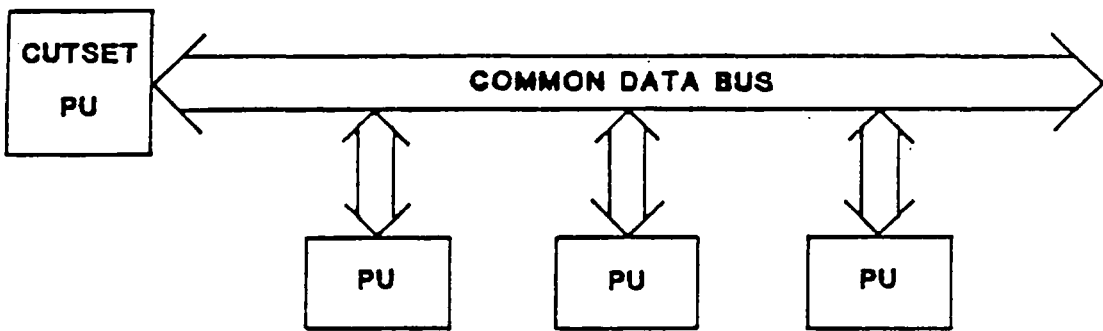
Fig. 1.4 shows an extension to the structure of Fig. 1.3b which possess the parallelism required by NBBDF algorithms. The common data bus is now broken up into sections by bus switches. Groups of processors may now operate simultaneously, independently of each other. The sequence of operation of the bus switches is controlled by the processors (see Fong and Pottle [14]).

1.1.4 - A Partitioned Ring Structure

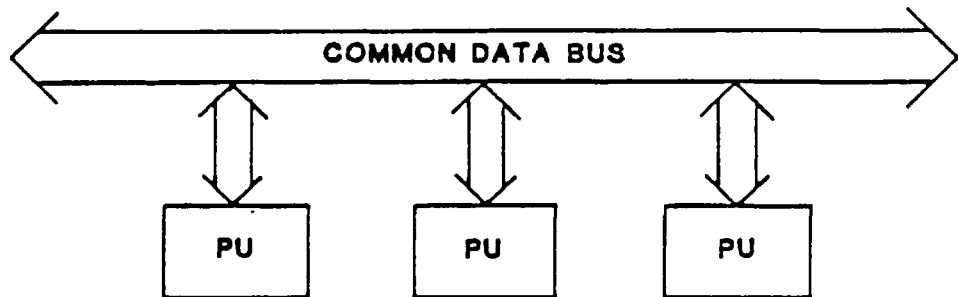
A rather different approach is followed by Brasch et al.[15] to design a multiple processor structure. The designed structure accomplishes all the control coordination through a message transmitting network. Several processors are connected in groups



(a) Three area power system



(b) Structure using a cutset processor



(c) Single common data bus structure

PU : Processing Unit

Fig. 1.3: Multiple Processing Structures for BBDF Algorithms.

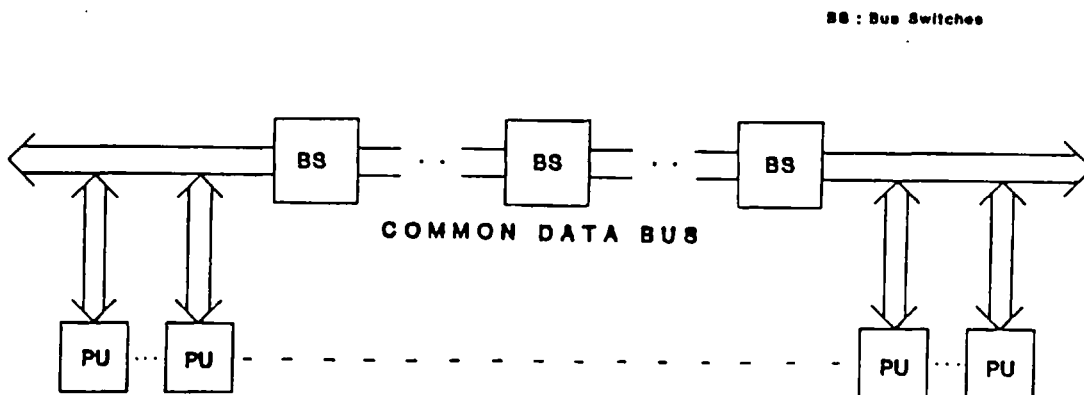


Fig. 1.4: Multiple Processing Structure for NBBDF Algorithms.

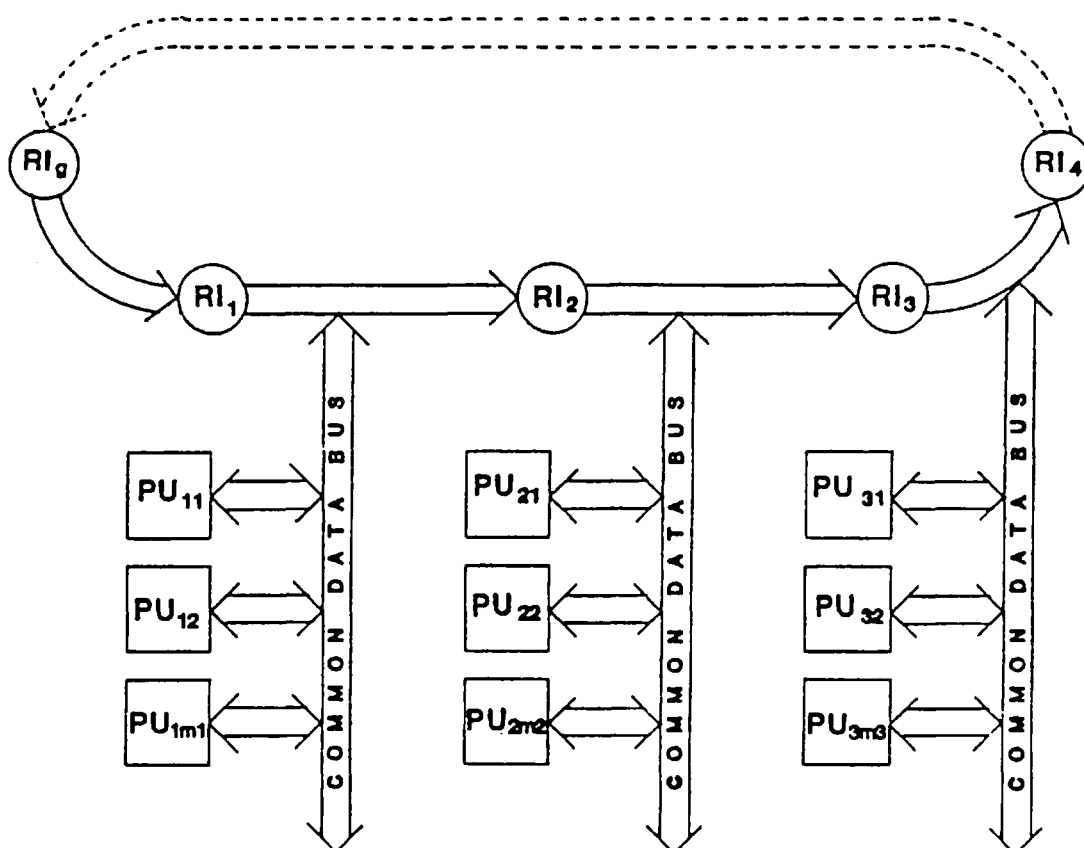


Fig. 1.5: A Partitioned Ring Structure.

through a common bus and these groups of processors on each bus are connected through a ring structure as shown in Fig. 1.5. When a message is sent out by a given processor it is received immediately by all the other processors in that group. But there will be a time delay before the message is transmitted to the next group of processors around the ring.

In the parallel processing structures described above, a key role is played by the communication network responsible for inter-processor information exchange. The choice of the most suitable structure for the solution of a given problem is hardware dependent and many more arrangements of processors may be considered in order to determine the best configuration compatible with the hardware available and the problem formulation.

1.2 - MICROPROCESSOR ARRAYS FOR POWER SYSTEM ANALYSIS

Following the cost-performance improvements of microprocessors in recent years, a timely research project was proposed by Short and Cory at Imperial College [1] to investigate the application of microprocessor arrays to develop a multi-purpose power system simulator. The proposed dynamic simulator would replace the hybrid simulator developed by the Power Systems Group, which demonstrated its usefulness as a research tool in the implementation of on-line real-time digital control techniques for multi-machine electric power systems.

In addition to the fields of research and design, some other applications of the new dynamic simulator would be:

- . for teaching: to demonstrate the principles of power system control.
- . for training: enabling power utilities to train control-room staff.
- . for consultants: in resolving system planning and operating problems.

The construction of such a simulator requires the implementation of a wide range of power system studies incorporating in the solution process the application of distributed processing techniques.

1.2.1 - General Objectives

To conduct the research study on the development of the microprocessor-based power system simulator, the following objectives were stated in the project proposal [1]:

1. To examine and determine the capability of a single microprocessor and associated memory in the solution of power system problems.
2. To determine the simulator architecture and software structure appropriate to the parallel interconnection of microprocessors for power system studies.
3. To design, construct and evaluate a flexible, expandable, multi-machine power system simulator.

An important consideration suggested in [1] for the design of the simulator was to use a modular approach for tailoring the simulator architecture in such a way that the simulator could closely parallel the actual power system configuration. Fig. 1.6 depicts the suggested microprocessors architecture. In this configuration a single microprocessor is used to represent identifiable elements or groups of elements of a power system. Further decompositions are possible by using more than one processor to represent a system element. The support computer could be used to monitor and control the simulator and to perform activities typical of an energy control centre.

This approach to distributing the task among the processors is based on a natural decomposition of a power system, whereas most of the approaches presented previously are based on a mathematical decomposition of the system equations. A similar parallel scheme was outlined by Stott in [19] as a promising future development for faster and cheaper simulations via many inexpensive microprocessors.

The modularity achieved with the natural decomposition of a power system for multi-level and parallel computations may be exploited to expand the simulator for the representation of very large power systems. Identical modules could be reproduced and easily incorporated onto the microprocessors structure according to the system characteristics. Also, this approach would facilitate simulator maintenance, since each module could be modified as a separate entity without affecting any other module. The interconnection, control,

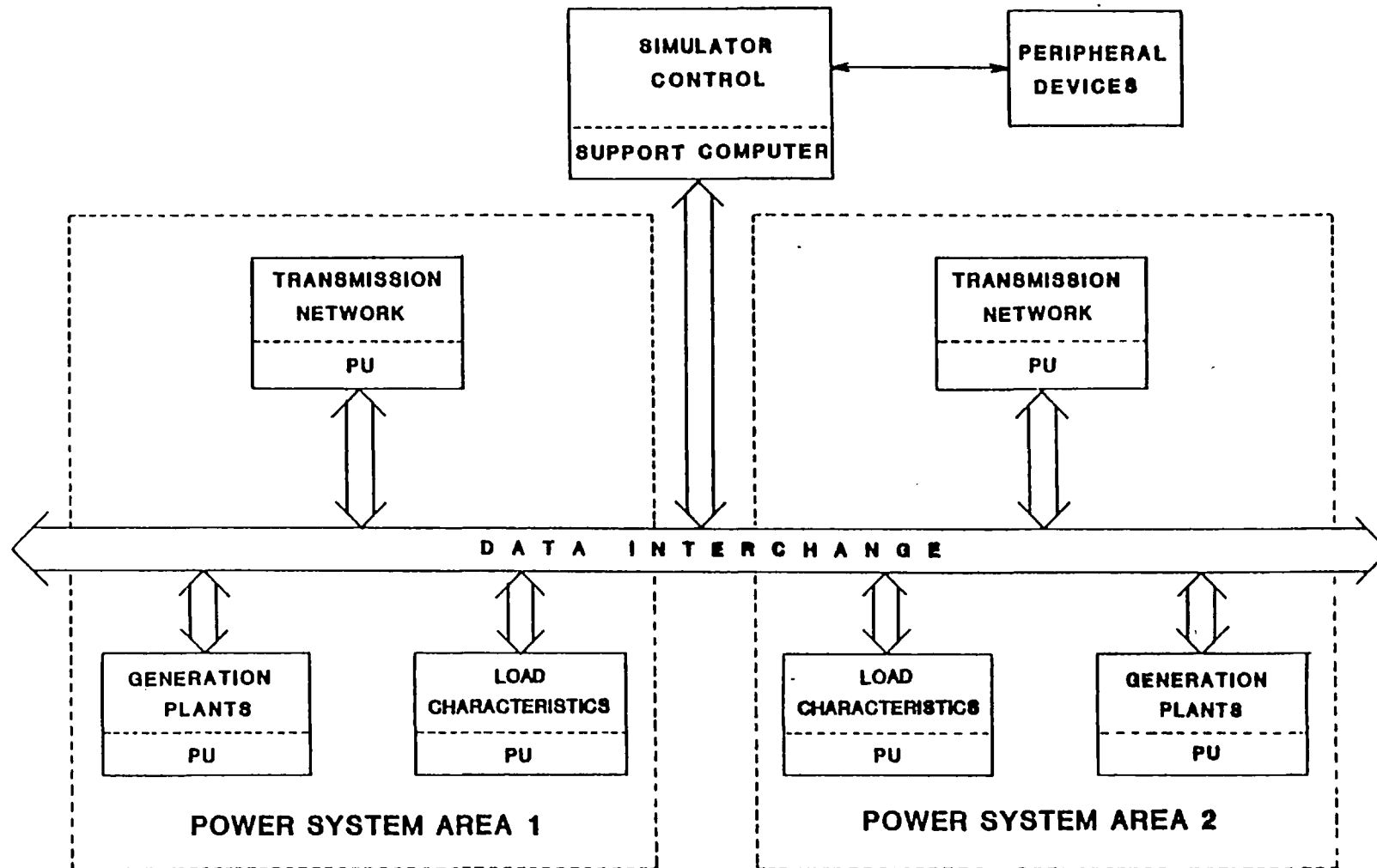


Fig. 1.6: Multiple Microprocessor Structure for a Dynamic Power System Simulator.

synchronization and communication between microprocessors for the proposed configuration are described later in the thesis.

1.3 - POWER SYSTEM ANALYSIS

A multi-purpose simulator should be capable of representing different power system scenarios to analyse system responses to various disturbances and to develop approaches to solving system analysis problems. A power system simulator, as defined in the IEEE Standard Dictionary of Electrical and Electronics Terms [110], "is a device, used to represent the behaviour of a physical system by virtue of its analogous characteristics. In this general sense, all computers are, or can, be simulators. However, in a more restricted definition, a simulator is a device used to interact with or to train a human operator in the performance of a given task or tasks."

Throughout this thesis the term "power system simulator" is used in a general sense to refer to the software and hardware configuration of an array of microprocessors used for power system analysis.

The development of a simulator for power system analysis using digital computers requires the formulation of mathematical models to describe the behaviour of the different power system components and the implementation of solution algorithms.

To obtain economical solutions of power system problems, the analysis of power systems has been divided into dynamic and steady state analysis and these states have been divided further into several studies.

1.3.1 - Steady State Analysis

The steady state operation of an electric power system requires that the power supply and the load demand must be balanced. Furthermore, the system has to be operated within specified limits.

Steady state analysis is used to obtain static security indices and rapid evaluations of system response to different disturbances. For any system changes, the dynamic transition of the system from its operating state to a post-disturbed state is ignored and it is

assumed that a new stable operation condition is reached. The studies generally considered include:

- . Load flow studies for assessing the system performance under normal or emergency conditions and for analysing the effectiveness of alternative plans for system expansion.
- . Contingency analysis to determine new system operating conditions resulting from line outages or changes in load or generation. This analysis is a useful assessment of security and assists the system dispatcher on decision making for appropriate action about impending problems.
- . Short circuit studies for system protection and planning purposes.

Load flow studies provide the initial conditions for the simulation of any disturbance and along with contingency analysis form the basic package of studies available in a power system control centre.

1.3.2 - Dynamic State Analysis

The dynamic state analysis is divided into several time ranges according to the predominant effects of the equations of motion of the system components as illustrated in Fig. 1.7 (see Concordia and Shulz [20] or Anderson and Dembart [21]). The bars represent the time range of the studies, the left end of each bar indicates the fastest dynamics that are usually of importance and the right end of the bar indicates the time over which the study is generally carried out to enable useful conclusions to be drawn.

The different time scales of the dynamic behaviour of power system elements give the flexibility to formulate comprehensive models valid for a specific span of time, thus avoiding unnecessary representations of system components which cause computational inefficiency and do not affect the outcome of the simulation studies. The considerations upon which general models are formulated for different time ranges should be uniform and compatible in order to provide smooth transitions between the studies.

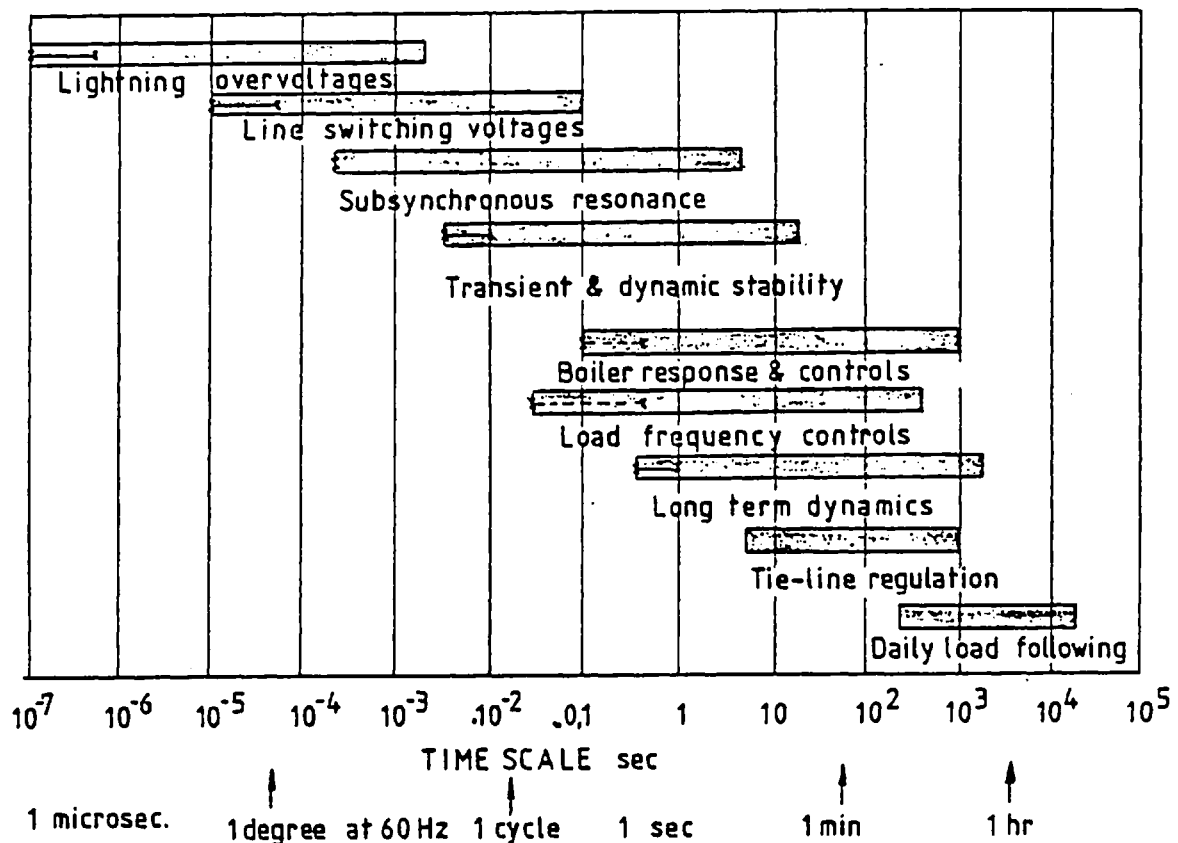


Fig. 1.7: Time Ranges of Power System Dynamic Phenomena.

The time frame to be considered in this work ranges from milliseconds to several seconds encompassing studies such as transient and dynamic stability.

1.4 - THESIS OVERVIEW

Analytical methods for power system analysis have been the subject of considerable effort over several years. Two excellent reviews on solution concepts and computational techniques for steady state and dynamic analysis were presented by Stott in references [18] and [19], respectively. Stott states in these papers that the objective of the calculation process is to compute system analysis problems as rapidly

as possible, consistent with the following requirements:

- . Sufficient accuracy for engineering purposes
- . Reliability, in the sense of not experiencing numerical breakdown on practical problems
- . Economy of computer storage
- . Versatility to handle conventional and special features
- . Flexibility in the range of modelling and in the ease with which models can be altered to accommodate changing study requirements and new types of apparatus
- . Ease of maintenance and enhancement of the whole program.

These requirements tend to conflict with the objective of fast solutions, making the design of the overall calculation scheme a very intricate exercise in tradeoffs. Stott concludes his review in [19], pointing out that the state of the art has not changed essentially for some years and that most of the development potential appears to lie, not in the conventional algorithmic sphere, but in the computation itself, much of which is hardware related. Parallel computations in various different forms are destined to reduce solution times and cost considerably, but are generating new problems of both program-organization and algorithmic nature.

At first glance, it would be expected that solution algorithms for power system analysis will execute much faster using multiple processors. However, there are overhead factors that offset the potential parallelism offered by a multiple processor configuration and decomposition penalties imposed by the solution algorithms. The overhead factors are a consequence of executing an algorithm on a particular hardware and system-software configuration. The decomposition penalties imposed by the solution algorithms are:

- . Synchronization. Some processors are idle waiting for other processors to deliver results.
- . Reconstitution. Combination of the results generated by the individual processors into an overall result for the entire computation.

In this work, close attention is given to the decomposition penalties imposed on the implementation of power system analysis algorithms for the development of parallel processing schemes using an

array of microprocessors.

The microprocessor units (MPUs) available are composed of the TMS9900, 16-bit microprocessor as CPU and a memory expansion board of 64 Kbytes.

With the continuous evolution in microprocessor technology it was considered important to implement the computational algorithms in a high level language and to employ a development system for the production of microprocessor code. The high level language PASCAL was considered because of its structured characteristics which enhance programming and ease program maintenance.

The capabilities of an MPU for the solution of power system problems are explored using a software package implemented in PASCAL for steady state analysis. The software package includes load flow, contingency and fault analysis. The contingency analysis algorithm is used for rapid evaluations of changes in the system from an available base-case load flow.

Multiple microprocessors and piecewise methods are employed for the solution of large power systems. Algorithms to identify optimum decomposition patterns and piecewise techniques are applied for the efficient decomposition of power system problems for parallel computations. Block matrix elimination algorithms that fully exploit sparsity at all levels in the computation process are employed for the solution of the diakoptic equations representing a partitioned system.

A parallel processing algorithm for the solution of load flow problems via multiple microprocessors is developed by applying piecewise techniques to the fast decoupled load flow method. The algorithm is verified and evaluated for the solution of IEEE standard test systems using up to 4 MPUs operating in parallel.

A unified approach for the computation of any structural changes in a transmission network, e.g. branch outage, single-line-to-ground fault, etc., is developed to perform contingency and fault analysis on a single microprocessor or a multiple array.

Multiple microprocessor configurations and solution techniques are presented for transient stability analysis of large power systems. The parallelism achieved with the algorithm implemented and the advantages gained with the parallel execution using multiple microprocessors are evaluated for the detailed dynamic simulation of bulk power systems.

Five microprocessor units are employed to implement a simulator for power system analysis and a TI-TMS990/4 minicomputer is used to provide a versatile user/simulator interface for interactive purposes.

The microprocessor-based simulator can be used as a computational aid in a wide spectrum of power system analysis studies and its usefulness has been demonstrated in applications including planning, design, teaching and research.

1.4.1 - Synopsis of Chapters

The thesis has been organised in nine chapters. Chapter 2 presents some considerations on the choice of mathematical representations for the principal components of power systems. Chapter 3 describes the computational environment and outlines the implementation of multiple microprocessor configurations with the hardware available.

The capabilities of a single microprocessor for the solution of power system problems are explored in Chapter 4 and a software package implemented in Pascal for steady state analysis is described. Chapter 5 deals in detail with the application of multiple microprocessors for the solution of large electric power networks. The performance of various parallel processing algorithms is evaluated using an array of microprocessors. A parallel load flow solution algorithm is developed and results for the solution of standard test systems using up to 4 microprocessors are presented. Chapter 6 considers the application of microprocessors for fault analysis calculations and describes an algorithm for the computation of any balanced or unbalanced disturbance in a transmission network.

The solution of the transient stability problem using multiple microprocessors is examined in Chapter 7. Solution techniques are described and the implementation of parallel processing algorithms is discussed giving emphasis to the interchange of information between the processors during the step-by-step calculation. Results for the detailed simulation of test systems using 4 MPUs are presented and the performance of the parallel processing schemes is evaluated.

Finally, Chapter 8 outlines the main characteristics of a simulator which has been developed using five microprocessor units to represent bulk power systems and a host minicomputer to provide a user/simulator interactive interface. Various examples are presented to demonstrate the application of the simulator for research and teaching purposes, and for power system planning and design studies. General conclusions, suggestions for further research and methods to improve the simulator are presented in Chapter 9.

CHAPTER TWO

FORMULATION OF THE MATHEMATICAL MODEL STRUCTURE

2.0 - INTRODUCTION

The primary concern in simulation studies by digital computers is the development of mathematical models to represent the system physical behaviour. The degree of detail introduced in the modelling and the simplifying assumptions that can be made are a function of the study requirements, the availability of data and the computational resources. With rapidly changing computer technology, the hardware environment can no longer be ignored in the modelling process and advantage should be taken in the formulation of mathematical models to exploit efficiently the hardware and software characteristics. Extensive work has been carried out in power systems from the early days of power generation and transmission to develop fairly good models of the different power system components for various types of studies [20],[25]-[46].

This chapter presents some considerations on the choice of models for the simulation of several kinds of power system situations. The considerations include the nature of the problem and consistency of representations to cover the technical scope of the system studies with adequate accuracy using models not so complex as to make the computer time required prohibitive. Bearing in mind that an array of microprocessors will be employed for the solution of the equations describing the system performance, the overall model structure is formulated using functional block diagrams to visualise the coupling variables between individual models and to ease the identification of the interconnection variables for the exchange of data once the tasks have been assigned to the different microprocessors.

2.1 - THE POWER SYSTEM

The major components of a power system are the transmission network, the generation plants, the electrical loads, the ties to neighboring systems and the system control centre (Anderson and Dembart [21]). Large power systems are normally decomposed into subsystems or

interconnected areas and each area may have a local control centre. DC transmission lines are omitted here for simplicity. A conceptual model of a general power system is shown in Fig. 2.1.

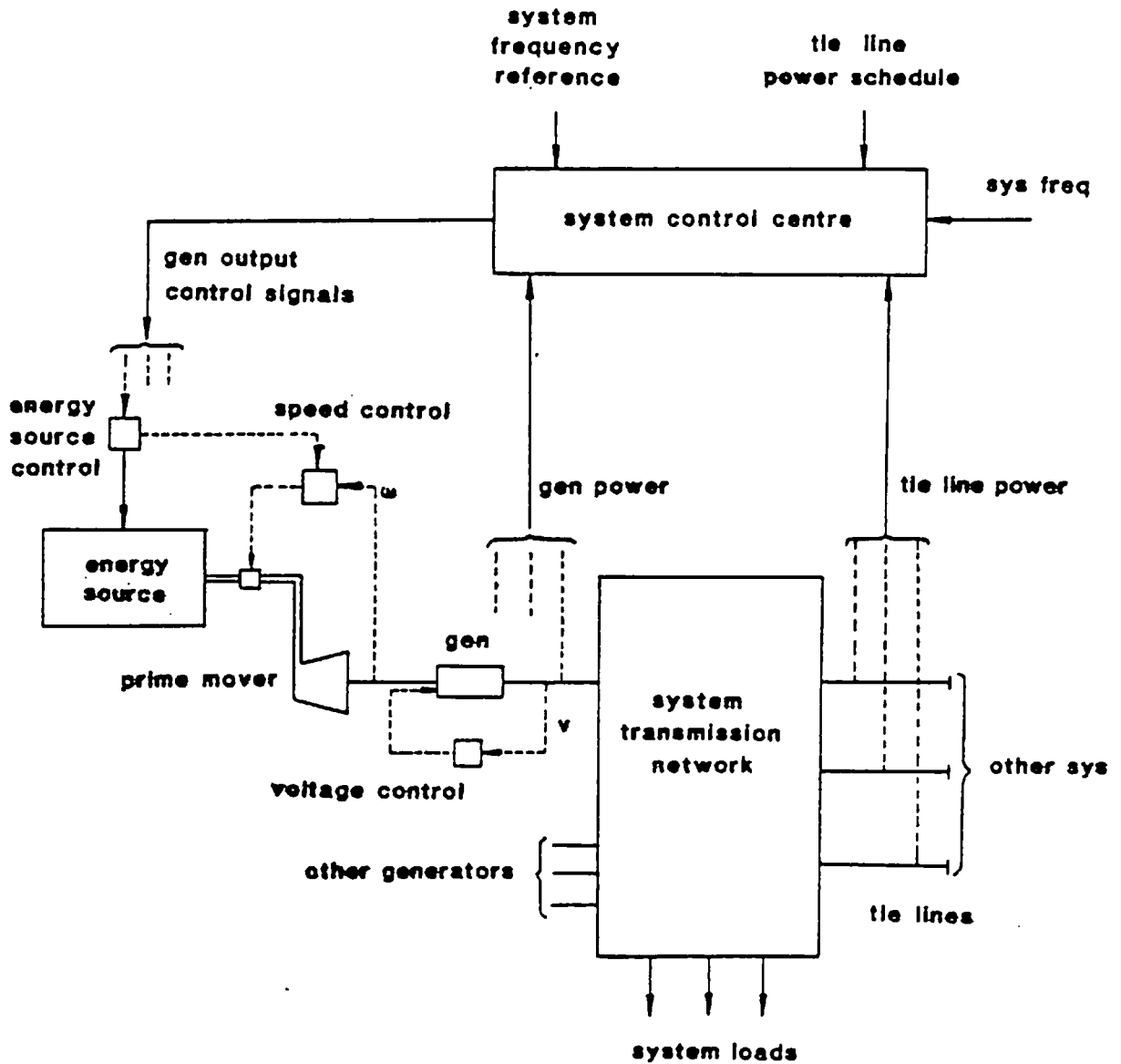


Fig. 2.1: Conceptual Model of an Interconnected Power System.

2.2 - INTERCONNECTED TRANSMISSION NETWORK

The mathematical formulation of a transmission network depends on the frame of reference, namely, bus, branch or loop. The use of the bus frame of reference greatly simplifies the preparation of data and the effects of shunt elements to ground can easily be included. Ground is usually selected as reference bus so that the bus voltages are voltages with respect to ground. If any other bus were chosen as reference, all voltages would be with respect to that bus and the shunt elements would be treated as current sources at the appropriate buses (see, for example, Sterling [22]). The admittance matrix formulation in the bus frame of reference for an interconnected network is

$$[Y][V] = [I] \quad (2.1)$$

For A.C. networks the quantities in (2.1) are complex values and the power at bus i in the network is given by

$$S_i = V_i I_i^* \quad (2.2)$$

Most A.C. transmission networks are usually symmetric, Kimbark [23], or nearly symmetric during normal conditions and during symmetric disturbances. For these conditions, the interconnected network may be represented on a single-phase basis for calculation purposes. In the analysis of unsymmetrical disturbances or unbalanced network operations the simple single-phase representation no longer suffices and a complete phase representation should be used in the calculation (phase coordinates representation). Alternatively the calculation could use transformation methods to represent the unsymmetric conditions as a system of symmetric sequence components [23].

2.2.1 - Representation of Unbalanced Networks

The representation of unbalanced networks using sequence components offers great savings in computer time and storage because the sequence equations are uncoupled. However, it is difficult to handle special unbalance situations such as untransposed transmission lines. The phase coordinates approach offers a much greater flexibility in the types of unbalance that can be treated but demands

extra computer time and storage due to the 3x3 representation of the network elements. Fig. 2.2 illustrates the respective representations of a three phase network element.

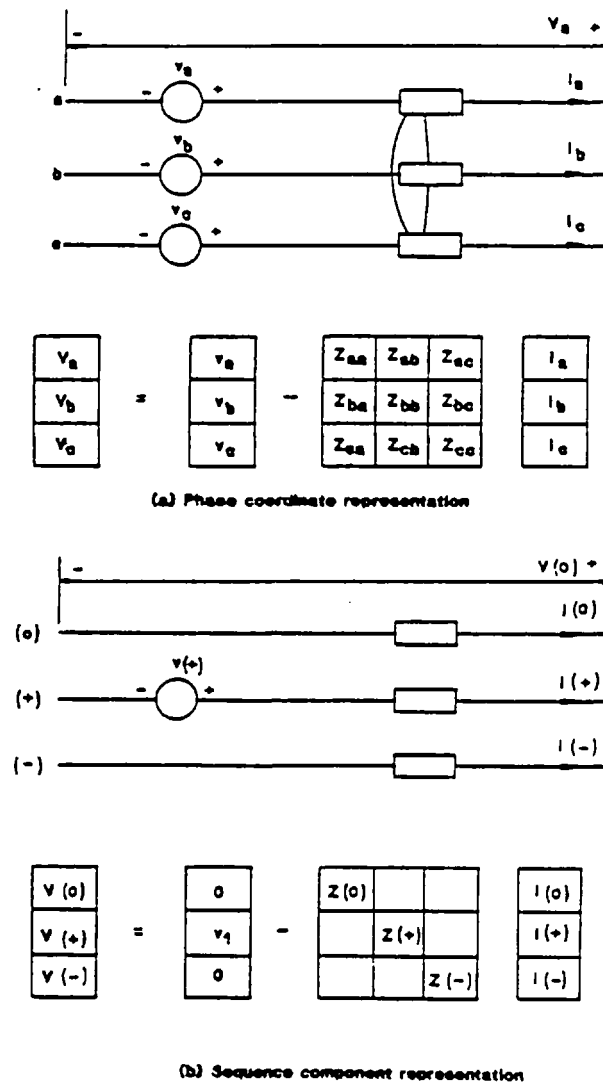


Fig. 2.2: Representation of a Balanced Three-phase Element.

The decoupling of the sequence networks allows each network to be handled separately during calculations. This is not possible when phase coordinates are used, since the phase equations are fully coupled throughout the network. An extension developed by Podmore [30] which combines the respective advantages of sequence components and phase coordinates is adopted to represent electrical networks under any unbalanced condition. This extension is obtained by partitioning the network into a balanced subnetwork and a number of unbalanced subnetworks. The balanced subnetwork, which is usually the main part of the network, is represented using sequence components and the unbalanced subnetworks are represented using phase coordinates. The

representation of a general network using this combined approach is illustrated in Fig. 2.3.

In symmetric components any network is represented by three single-line-to-ground networks: positive, negative and zero sequence. A sequence network is identical to the original network, the only difference being that the value of each impedance is a value unique to each component. Sequence networks are usually treated as one-port or two-terminal networks, where one terminal is the unbalance point and the other terminal is the zero potential bus. Thevenin's theorem is used to model each sequence network. The Thevenin impedance is the network impedance as seen from the unbalance point, Anderson [26], and the Thevenin equivalent voltage in the positive sequence network is the predisturbance voltage at the unbalance point. Thevenin equivalent or open circuit voltages in the negative and zero sequence networks are zero. Fig. 2.4 shows the representation of the sequence networks which are entirely independent under normal conditions and are connected only at the point or points of unbalance under unbalanced conditions.

Simplifications in the representation of unbalanced networks are possible by representing only the unbalance area of the system using the three sequence networks and considering the rest of the network as balanced. Or, by combining the negative and zero sequence networks into an equivalent passive network which is considered as a connected part of the positive sequence network (Han [31]).

In dynamic studies that are not concerned with the details of the negative and zero sequence networks but are concerned with the impact of these sequence parameters, the representation of the negative and zero sequence networks may be considered during the fault period only.

2.2.2 - Representation of Transmission Lines and Transformers

Transmission lines are represented by the equivalent π circuit shown in Fig. 2.5a. The model includes the line series resistance and reactance, and line charging effects. In this representation, the effects of harmonics and line transients at other than the nominal frequency are neglected.

Transformers and tap changing transformers are modelled by the equivalent π circuit shown in Fig. 2.5b. The equivalent circuit parameters are the series resistance, reactance and the transformer tap position. If the tap position is set to 1, the shunt elements become open circuits and the equivalent network reduces to the transformer

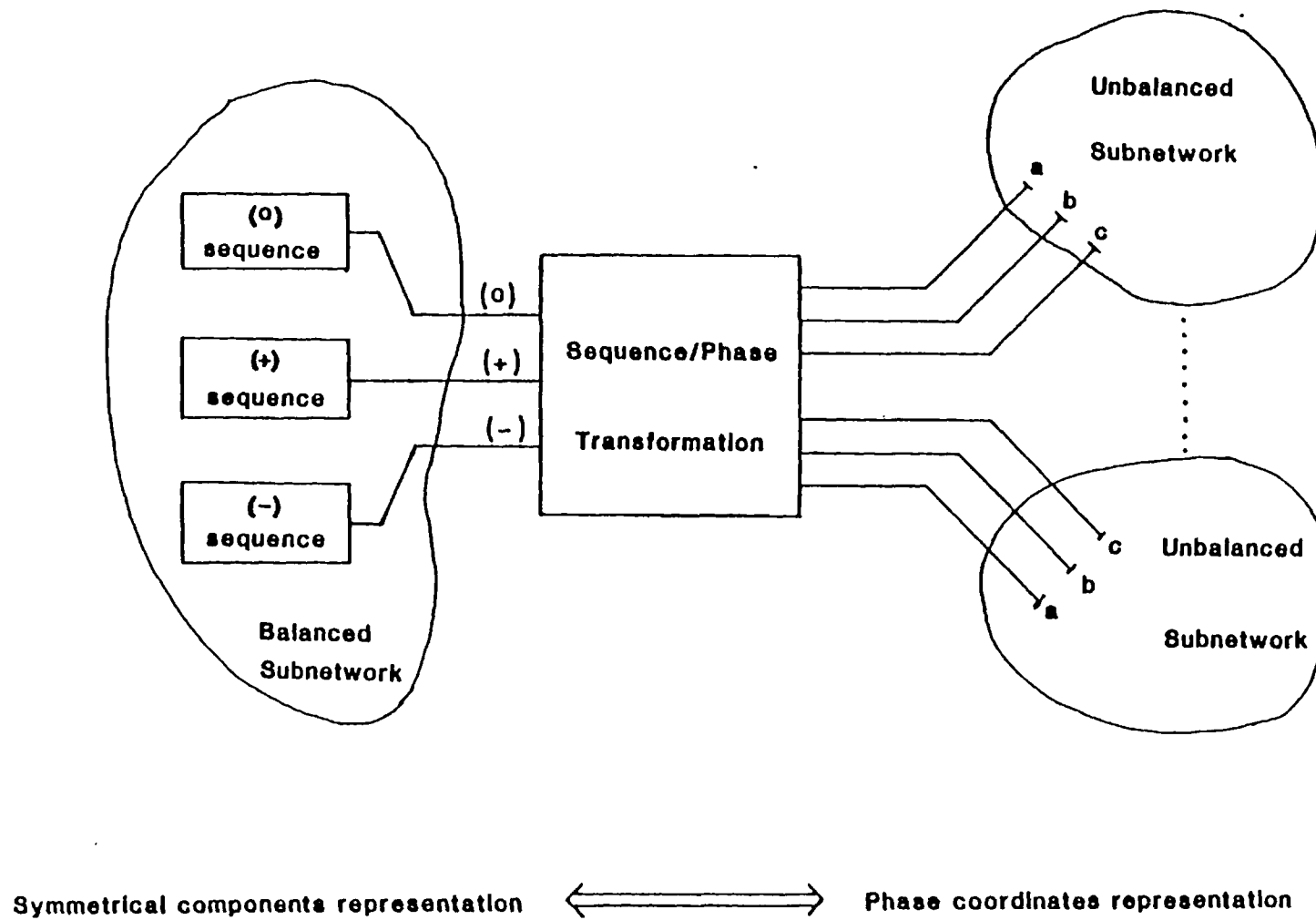
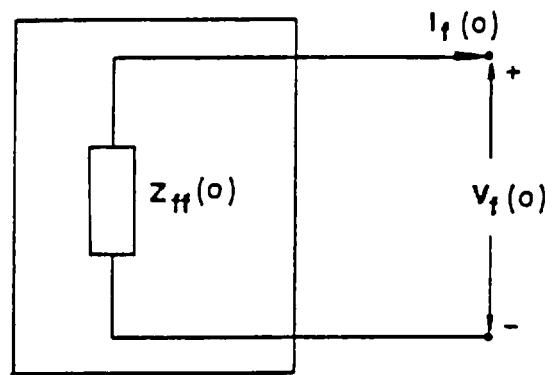
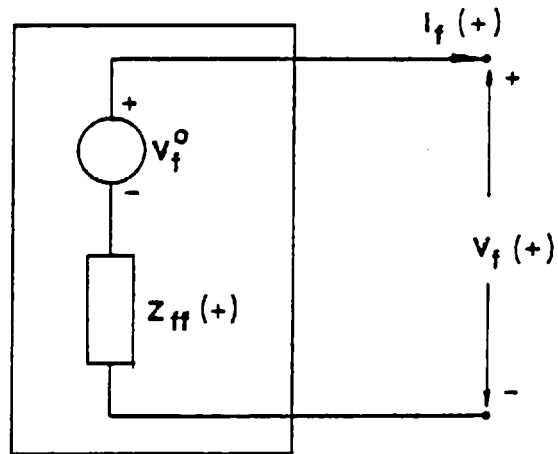


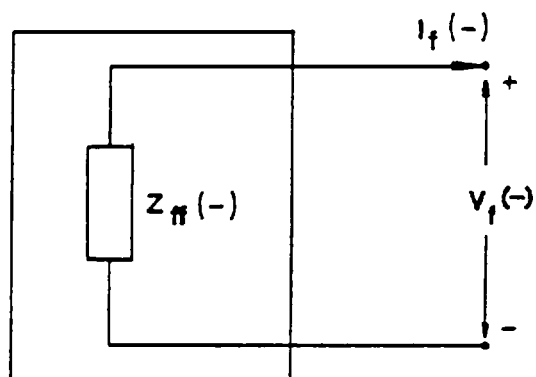
Fig. 2.3: Representation of Unbalanced Networks



Zero sequence



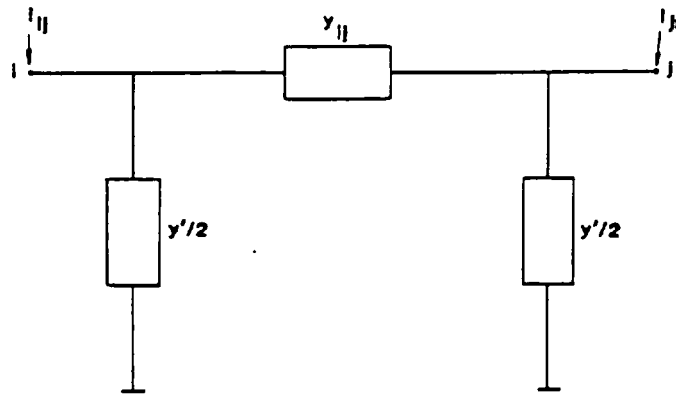
Positive sequence



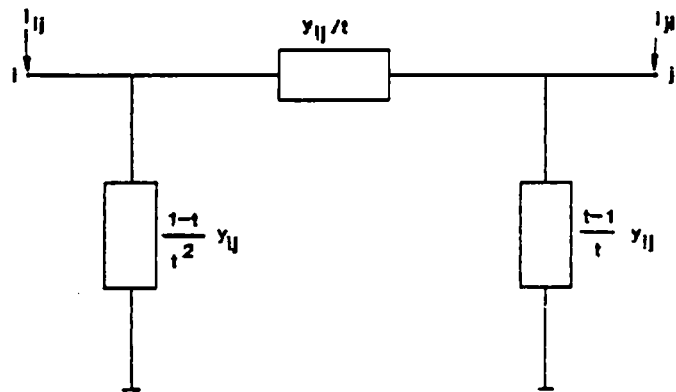
Negative sequence

Fig. 2.4: Sequence Networks

series impedance.



(a) Transmission lines



(b) Transformers

Fig 2.5: Representation of Transmission Network Elements.

Static shunt reactors and capacitors, which are used to supply the requisite reactive power to maintain the network voltage profile, are represented by fixed admittances.

2.2.3 - Network Mathematical Formulation

The formation of the bus admittance matrix $[Y]$ that describes an interconnected network depends upon the network topology and the physical characteristics of the network elements. This matrix may be determined directly from transmission network data since the data are normally specified by a bus connection code together with the impedance or admittance of each network branch (transmission line or

transformer).

The $[Y]$ diagonal elements are computed from

$$Y_{ii} = \sum_j (y_{ij} + y'_{ij}/2) + y_{shi} \quad (2.3)$$

and the offdiagonal elements of $[Y]$ are determined by

$$Y_{ij} = -y_{ij} \quad (2.4)$$

If a tap changing transformer is connected between busses i and j with bus i being the bus on the tap side, then when forming the diagonal elements, y_{ij}/t^2 is used for Y_{ii} and y_{ij} for Y_{jj} . The corresponding offdiagonal element is $-y_{ij}/t$.

The offdiagonal elements, where there is no branch connecting buses i and j , are zero.

The matrix $[Y]$ is of dimension $n \times n$, it is usually symmetrical and for large networks is very sparse.

2.3 - GENERATION PLANT

A generation plant is the primary element of an electric power system in which a basic fuel is converted into electricity. The fuel may take many forms from coal or oil to nuclear fission, or recovery of the kinetic energy in water flow or wind speed. According to the energy source used, the generating plants may be broadly grouped as:

- . Steam Generation Plants
- . Hydro Generation Plants
- . Non conventional Generation Plants

Fig. 2.6 shows the main components of a power plant and their interactions (Adibi et al.[42]). The terminal voltage V and the injected current I couple the generating plant to the network. Auxiliary couplings through control signals, come from the system control centre. The exciter, which drives the generator field coils at voltage E_f , may use as part of its input signals the supplementary voltage V_s to ensure positive damping. The power system stabiliser

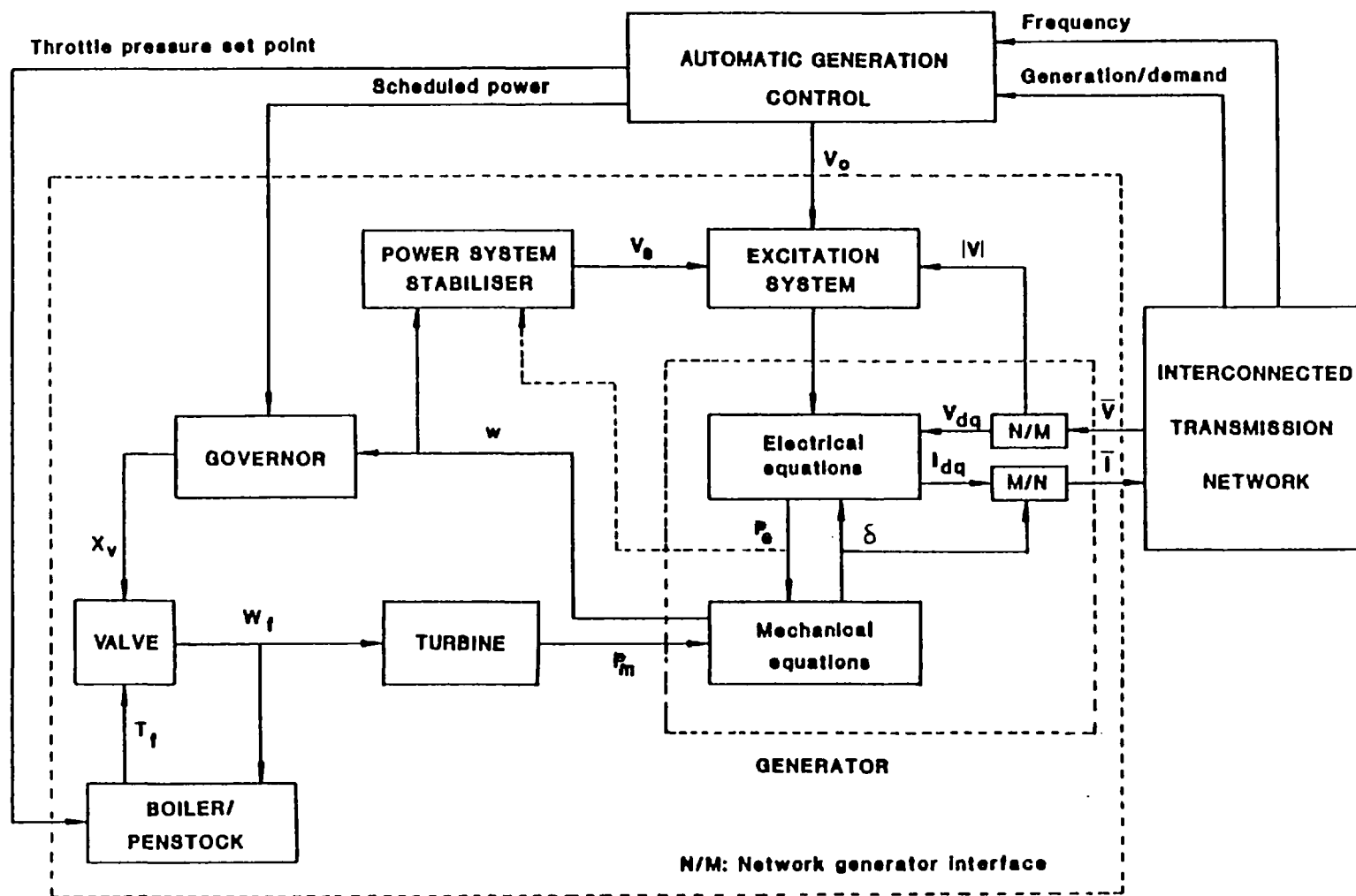


Fig. 2.6: Generation Plant Model

may be sensitive to angular frequency slip signals, to the electrical power changes or both. The coupling of the generator to the turbine-governor system is through the angular frequency ω and the mechanical power P_m . The element representing the energy source, boiler, water column, etc., will depend on the type of plant. The model structure of Fig. 2.6 may be used to construct flexible models with different degrees of complexity for a variety of system studies. The representation of certain elements can be altered to accommodate changing study requirements and new types of apparatus can be incorporated without difficulty. When the inclusion of an element is not required in the power plant model, then the coupling variables that are outputs from the omitted element will appear as constant inputs to the elements that were coupled and are considered in the model.

2.3.1 - Generator Model

In most power system studies, the synchronous generator models are based upon the (dqo) variables obtained by transformation of the machine variables (abc). The transformation used is usually called Park's transformation [32],[33], in which the machine electrical behaviour may be represented by equivalent circuits in the rotor direct and quadrature d-q axes.

For large scale stability studies, it is recommended to use an intermediate order model which has a single quadrature axis rotor circuit and a field circuit (see, for example, Dandeno et al.[34]). This model has proved to be very accurate in the simulation of major disturbances (faults) and virtually "exact" for small disturbances (line trip without fault). The representation of a three-phase synchronous machine using this model is depicted schematically in Fig. 2.7. The stator is considered to be three identical, symmetrically placed, lumped windings (abc). The rotor is considered to be two unequal lumped windings (f,kq) each placed with its axis 90° around the periphery (Anderson and Fouad [27]).

To be consistent with the network equations the model is formulated using voltages, reactances and associated time constants. Thus, the mathematical representation for the synchronous machine of Fig. 2.7, neglecting stator transients, is given by (2.5)-(2.11) [27].

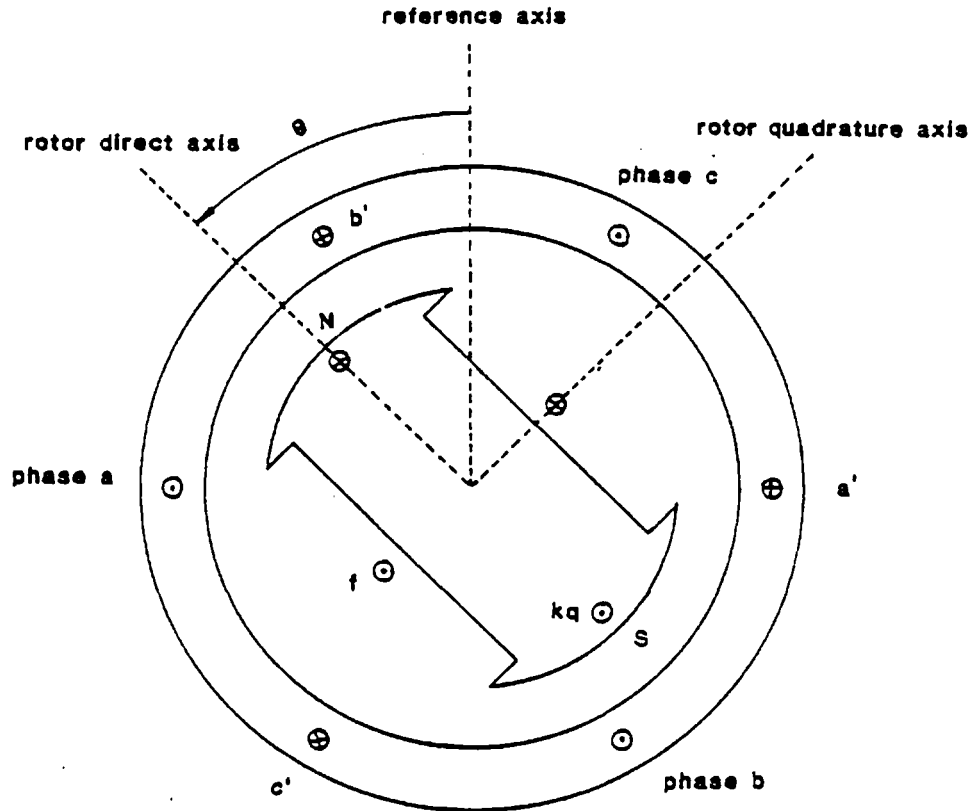


Fig. 2.7: Synchronous Machine Representation.

- Stator equations

$$\begin{bmatrix} I_q \\ I_d \end{bmatrix} = K_{yg} \begin{bmatrix} R & \bar{x}_d' \\ -\bar{x}_q' & R \end{bmatrix} \begin{bmatrix} E_q' - V_q \\ E_d' - V_d \end{bmatrix} \quad (2.5)$$

- Rotor equations

. Mechanical equations

$$\begin{bmatrix} \dot{\delta} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -D/2H \end{bmatrix} \begin{bmatrix} \delta \\ w \end{bmatrix} + \begin{bmatrix} 0 & -1 \\ 1/2H & 0 \end{bmatrix} \begin{bmatrix} P_a \\ w_s \end{bmatrix} \quad (2.6)$$

. Electrical equations

$$\dot{E}_q' = -[E_q' - (\bar{x}_d - \bar{x}_d')I_d - E_f]/T_{d0}' \quad (2.7)$$

$$\dot{E}_d' = -[E_d' + (\bar{x}_q - \bar{x}_q')I_q]/T_{q0}' \quad (2.8)$$

where: $K_{yg} = (R^2 + \bar{x}_d \bar{x}_q')^{-1}$ (2.9)

$P_a = P_m - P_e$ (2.10)

$P_e = V_q I_q + V_d I_d$ (2.11)

w is the synchronous speed

It should be noted that for hydro generators which are generally salient poles machines the q-axis effective reactance in the transient period is the same as the synchronous reactance since the i decays to zero after the subtransient period (a few cycles of occurred^q a fault) [27]. For hydro units $T'_{q0} = 0$ and $X'_q = X_q$ therefore the solution of (2.8) is not required. The transient voltage E'_d is calculated instead by the algebraic equation

$$E'_d = V_d + RI_d + X'_q I_q \quad (2.12)$$

The classical stability model of the generator, which uses a constant voltage magnitude behind the transient reactance is represented by the two mechanical equations of the rotor (2.6), which are better known as the swing equations.

2.3.1.1 - Damping

At any given oscillation frequency, synchronizing torques are developed in phase with the machine rotor angle and damping torques with the machine rotor speed. The synchronizing torques restore the machine rotor angle following an arbitrarily small displacement of this angle and they have been automatically included in the formulation of the machine electrical equations (Olive [35], DeMello and Concordia [36]).

The damping torques always tend to slow the machine down and are formed by the following components:

- . Negative sequence braking
- . Direct current braking
- . Positive sequence damping
- . High frequency variable torques

To avoid the continuous calculation of the damping torque components, a damping coefficient D proportional to the shaft speed or angular slip, is introduced to account for the system damping as shown in (2.6) [27].

2.3.1.2 - Saturation

Saturation effects in synchronous machines during transient conditions are quite complex if treated precisely. The representation of saturation requires that the stator reactances x'_d and x'_q be modified by saturation factors that depend on the Potier voltage (a function of V and E'). A less rigorous representation of saturation, which is incorporated in the model, is not to modify the stator reactances at all, but to modify the stator internal voltage instead by an amount $S_g E'$ from the unsaturated value. As a practical matter the q-axis seldom saturates, so only E'_q needs to be adjusted for saturation [27]. The saturation factor S_g may be expressed as

$$S_g = A_g \exp[B_g(E'_q - 0.8)] \quad (2.13)$$

As the coefficient A_g is usually a very small number, it is assumed in (2.13), that there is no saturation for $E'_q < 0.8$ pu.

2.3.1.3 - Saliency

Saliency is strongly manifested in salient poles machines, Kimbark [25], where there is a narrow air gap under the pole structure and a long air path of the interpolar space which creates a difference of permeance on the d-q axis and therefore a reluctance torque. The difference in the d-q reactances due to this effect introduce a complication in the representation of the stator machine equations, since they cannot be combined to have a phasorial representation consistent with the network equations (2.1). This is illustrated by rewriting the stator equations as

$$\bar{V} = \bar{E}' - (R - jx'_d)I + j(x'_q - x'_d)I_q \quad (2.14)$$

where $\bar{V} = V_q + jV_d$ and $\bar{E}' = E'_q + jE'_d$

The last term on the r.h.s. of (2.14), represents the saliency effect which is only eliminated when $x'_d = x'_q$.

In the model selected to represent the generators, saliency effects cannot be assumed negligible. Since the d-q reactances cannot

be expressed in a complex form, the network equations (2.1) have to be solved iteratively. A direct solution of the network can be achieved by introducing a higher order model in which a winding is added on each axis of the synchronous machine representation and by assuming that the subtransient saliency is negligible (Dandeno et al.[37]). Although the resulting model eliminates the need for iterations in the network solution, it requires a shorter integration time step than the chosen model. Since only two or three iterations are normally sufficient to adjust for saliency, the higher order model does not afford any computational simplifications.

The functional block diagram shown in Fig. 2.8, illustrates the representation of the model selected for the generators.

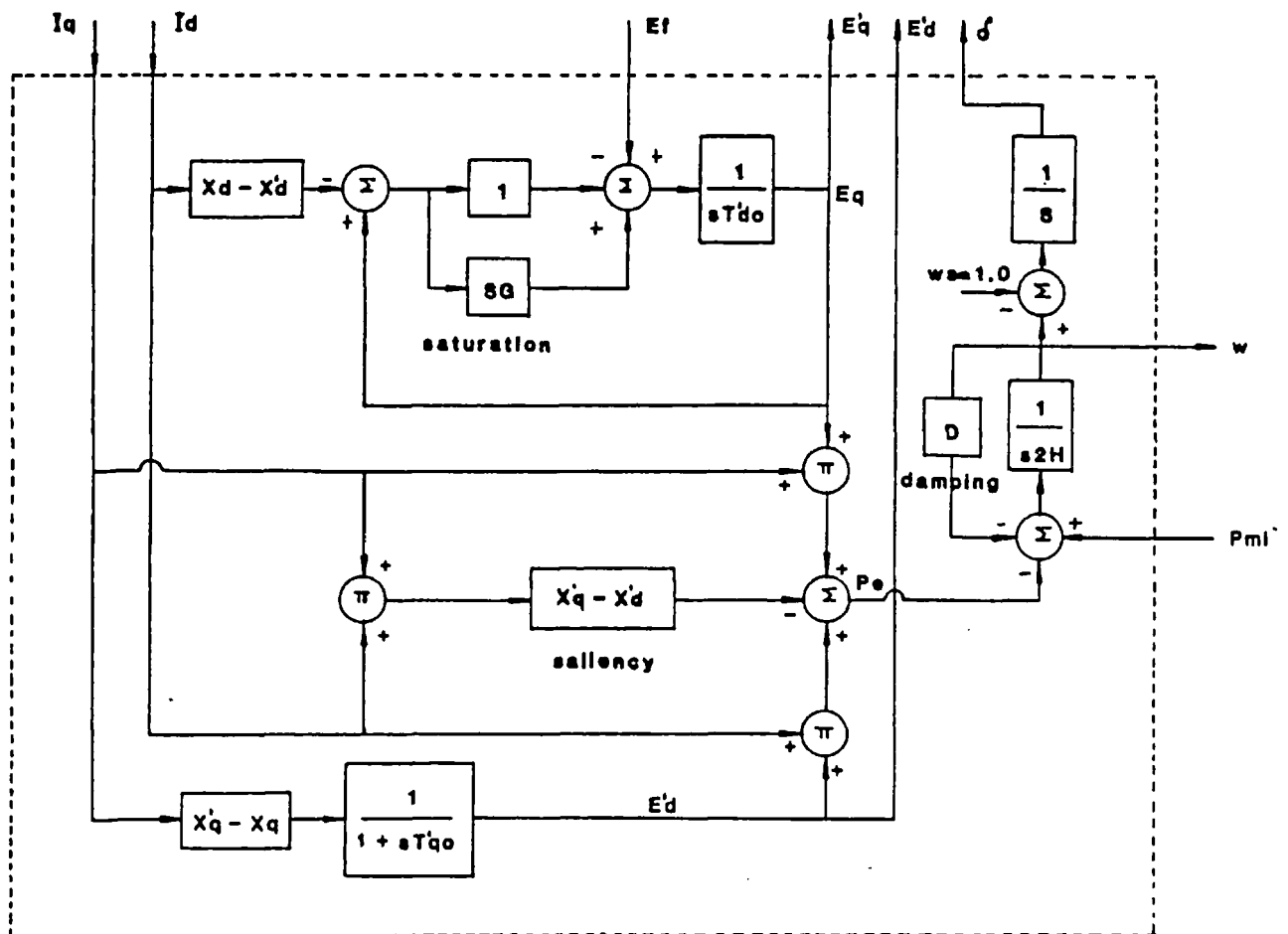


Fig. 2.8: Block Diagram for a Two-axis Generator Model.

2.3.2 - Network/Generator Interface

In the exchange of data between generators and network, a coordinates transformation must be performed since the network variables are referred to a common complex reference frame and the generator variables are referred to the d-q reference frame of each machine. This transformation is simply a rotation of phasors [27] as shown in Fig. 2.9 and expressed mathematically by

$$\begin{bmatrix} d \\ q \end{bmatrix} = \begin{bmatrix} -\sin & \cos \\ \cos & \sin \end{bmatrix} \begin{bmatrix} re \\ im \end{bmatrix} = [T] \begin{bmatrix} re \\ im \end{bmatrix} \quad (2.15)$$

Before transferring any values from the generators, the transformation $[re, im] = [T]^{-1}[d, q]$ is performed and when receiving values from the network, the transformation $[d, q] = [T][re, im]$ must be done before using the values in the generator equations.

An equivalent circuit consistent of a current source in parallel with a fictitious admittance (Norton equivalent), is recommended by Dommel and Sato [38] to represent the generators as viewed from the transmission network to improve convergence in the saliency correction process. This equivalent circuit is shown in Fig. 2.10. The value of the current source is obtained by expressing the generator stator equations in the complex reference frame of the network and adding to the corresponding diagonal term of the admittance matrix the fictitious admittance. The values of the equivalent circuit parameters are given by

$$Y_{fa} = K_{yg} [R - j(\chi_d' + \chi_q')/2] \quad (2.16)$$

$$I_{fa} = Y_{fa} \bar{E}' + I_{sal} \quad (2.17)$$

$$I_{sal} = -j(\chi_q' - \chi_d')(\bar{E}'^* - \bar{V}^*) \exp(j2\delta) K_{yg}/2 \quad (2.18)$$

With this representation, the network transfers to the generators the bus voltage at generator terminals and the generators transfer the fictitious current I_{fa} to the network.

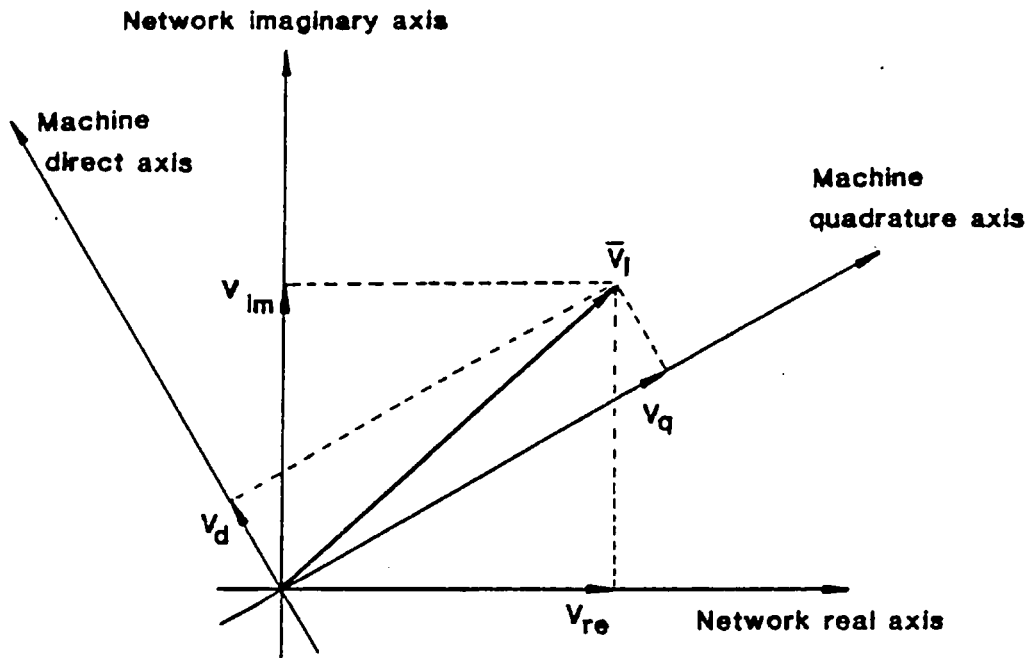


Fig. 2.9: Two Frames of Reference for Phasor Quantities for a Voltage $\bar{V}_i$

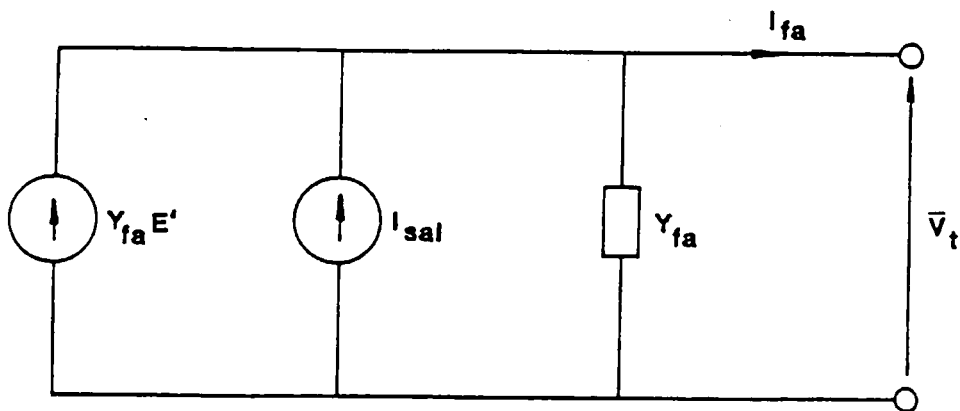


Fig. 2.10: Equivalent Circuit for Generator

2.3.3 - Excitation system

The function of an excitation system is to control the generator terminal voltage and the reactive power. Excitation system models for large scale system stability studies have been recently recommended by the IEEE Working Group on Computer Modelling of Excitation Systems [39].

The block diagram shown in Fig. 2.11 describes the Type DC1 Excitation System model which is used to represent field controlled DC commutator exciters with continuously acting voltage regulators. The principal input to this model is the generator terminal voltage V_t . The exciter may be either separately or self-excited. When a self-excited shunt field is used, K_e represents the setting of the shunt field rheostat and its numerical value is negative. The non-windup limit on the amplifier, represents the amplifier power supply limitations. The exciter saturation is given by

$$S_e = A_e \exp(B_e E_f) \quad (2.19)$$

2.3.4 - Power System Stabiliser

Power system stabilisers supply control signals to the voltage regulators to provide positive damping to machine-system oscillations and to damp tie line oscillations [27],[40]. Different quantities including electrical power, shaft speed and frequency deviations are used as the input signals to a power system stabiliser. The shaft speed signal has proved to be ideally suited for stability studies. With the speed as the input quantity, it is necessary to provide a phase lead component to compensate for lags between the regulator input and the resulting electrical torque change.

Fig. 2.12 shows the model of a power system stabiliser with two stages of lead-lag compensation. The reset term is used to "washout" the compensation effect after a time T_o . The use of reset control assures no permanent offset in the terminal voltage due to a prolonged error in frequency, such as might occur in an overload or islanding condition. The limiter assures that the stabiliser output never dominates the terminal voltage feedback in the excitation system.

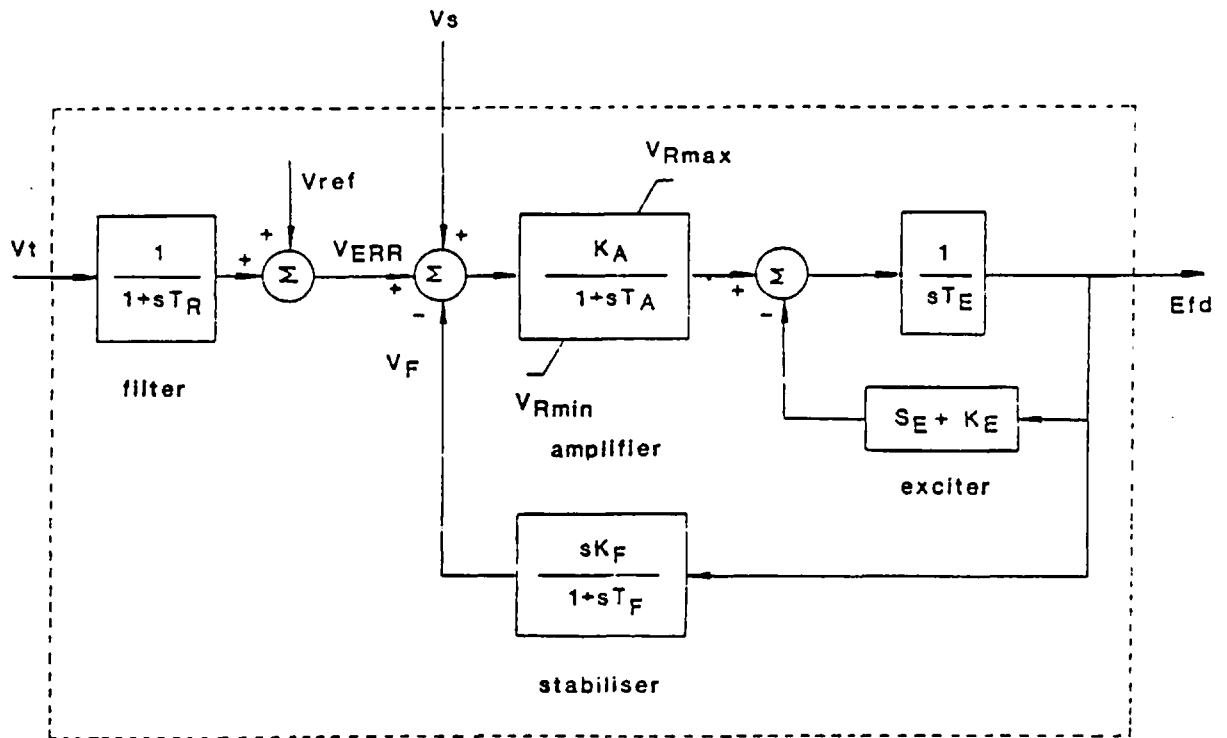


Fig. 2.11: Excitation System Model

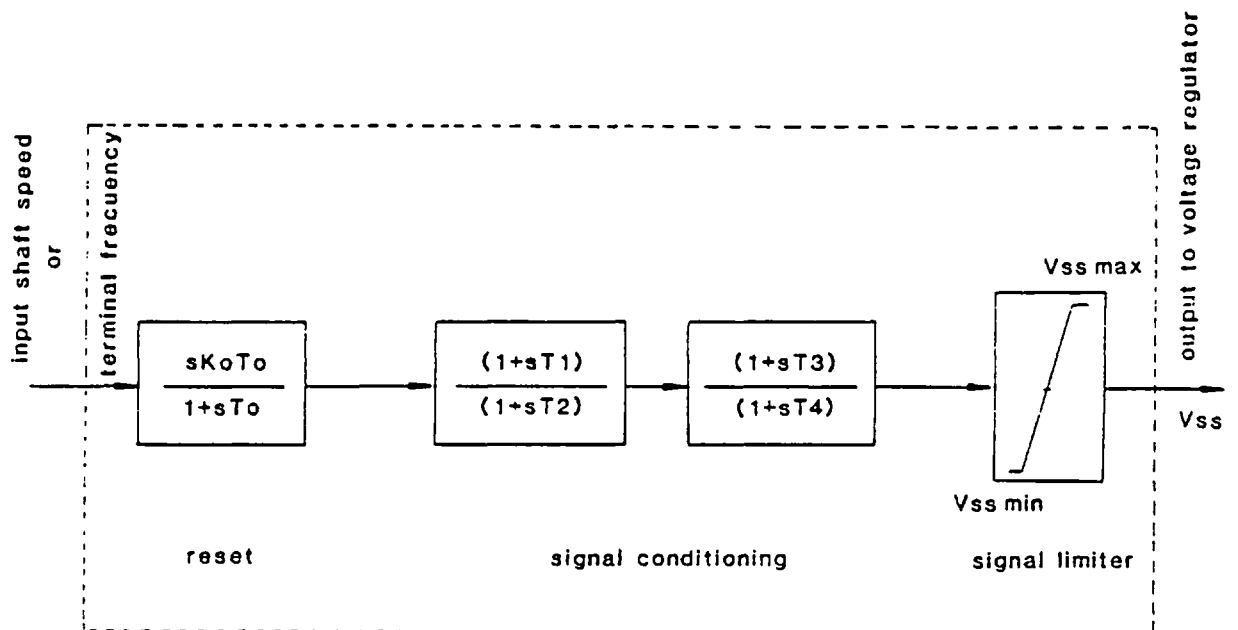


Fig. 2.12: Power System Stabiliser Model

2.3.5 - Speed Governor Systems

Speed governor systems are used to keep under prescribed limits the power output of a generator by controlling the input flow to the turbine. Many forms of governor system have been devised but all include in some way the variation of the turbine-generator shaft-speed as the basis on which to alter the current position of the turbine working fluid control valve [21].

A general model of a speed governing system for steam turbines is shown in Fig. 2.13a (IEEE Committee Report [41]). A distinction can be made between a mechanical hydraulic control system and an electro-hydraulic control system by an appropriate selection of values for the model parameters [41].

Fig. 2.13b shows an approximate nonlinear model of a speed governing system for hydro turbines.

Although, governor deadband (backlash) is not shown in the models, provision is made to include their effects in studies where tie line flows and frequency control are analysed.

2.3.6 - Turbines

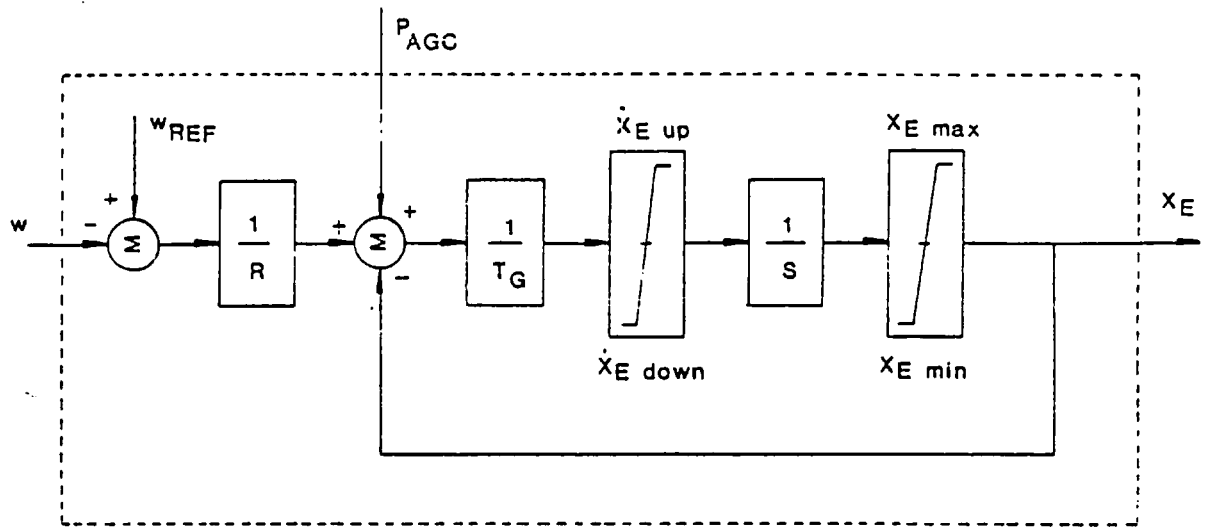
The turbines supply the mechanical energy to drive the generators and directly influence the electrical MW outputs and frequency. Mathematical models for different types of steam and hydro turbines are recommended in [41] for dynamic studies. The turbine equations are usually relatively simple. A single reheat steam system configuration with its approximate linear model is shown in Fig. 2.14a, where flows into and out of any steam vessel are related by a simple time constant but the steam extraction taken at various turbine stages to heat feed-water is not shown in the model.

If the alternative of fast valving is required in the model, then it can be simulated by incorporating the representation of an interceptor valve in the system as shown in Fig. 2.14a.

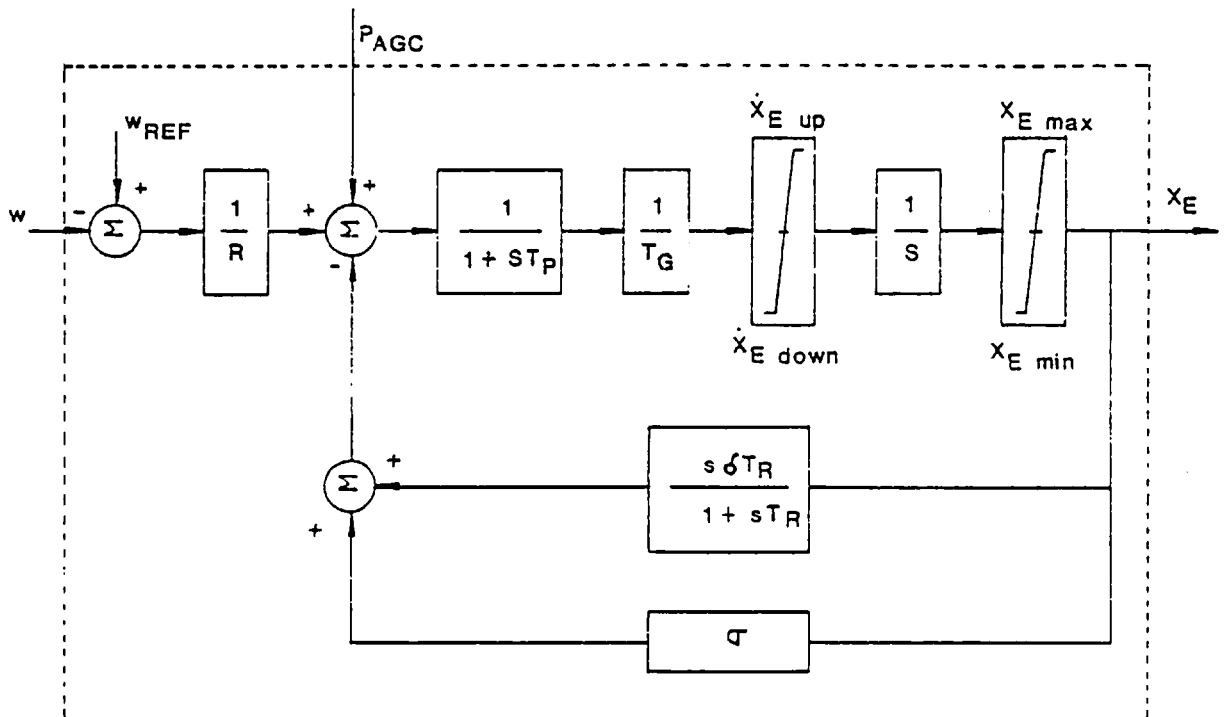
The transient characteristics of hydro turbines are determined by the dynamics of water flow in the penstock. A general simplified model for hydro turbines is shown in Fig. 2.14b.

2.3.7 - Boiler

Fig. 2.15 shows an approximate model to represent a boiler (DeMello et al.[42]).

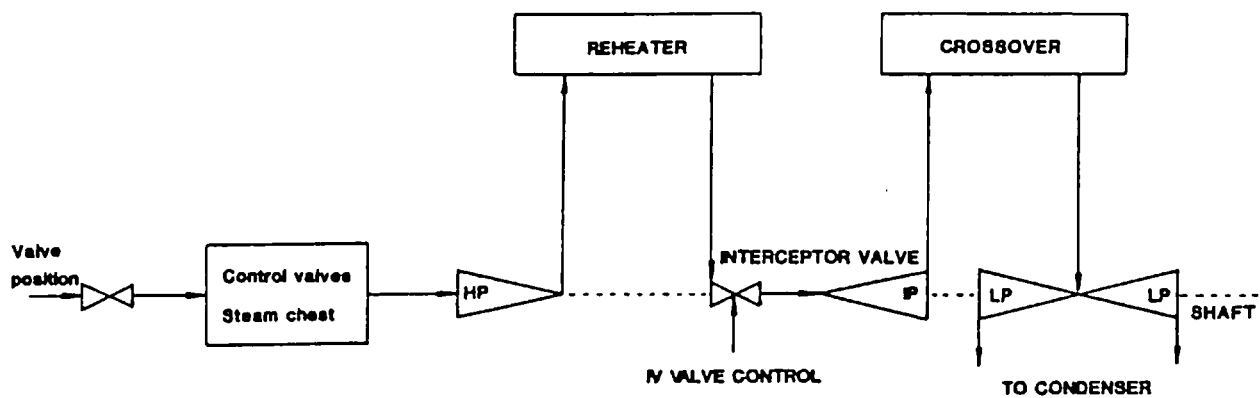


(a) Speed governing system for steam turbines

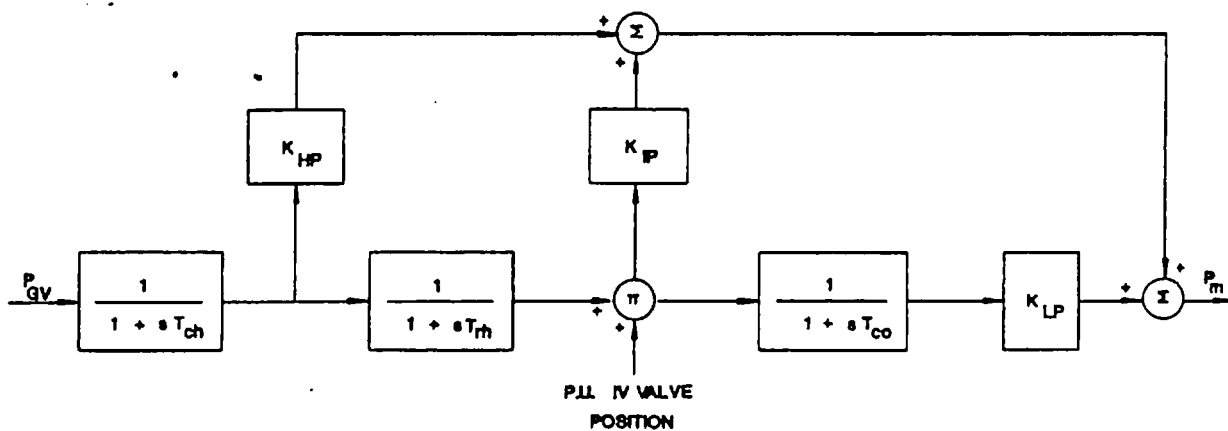


(b) Speed governing system for hydro turbines

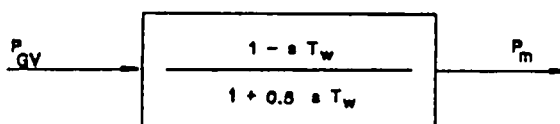
Fig. 2.13: Governor Speed Models



Steam system configuration



(a) Approximate linear model of a tandem compound single reheat steam turbine



(b) Approximate linear model for hydro turbines

Fig. 2.14: Turbine Models

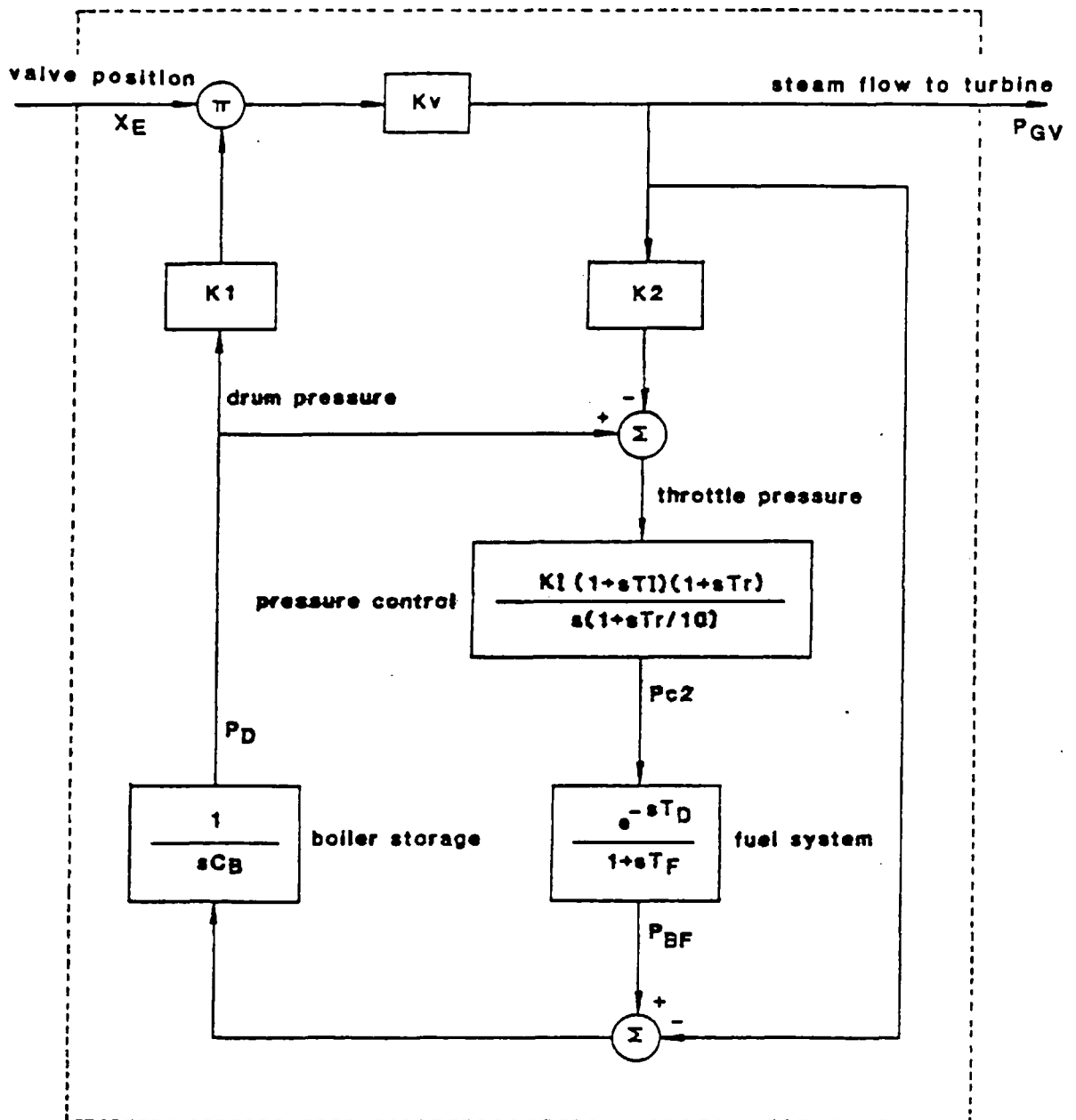


Fig. 2.15: Boiler Model

Steam temperature and boiler-feed-water level controls have been neglected in the model and the drum and all heating tubes are represented as a single energy store which is proportional to steam pressure. The model includes the effects of long term dynamics of fuel flow, steam flow on boiler drum pressure and the combustion controls. The effect of controls applied to the energy supply system is to restore pressure within response capabilities of the fuel supply system. In this model, the change in steam flow will be transient as energy is drawn from boiler storage and the contribution to change in steam flow is due to boiler pressure deviations.

Although, the model is basically that for a drum type boiler, similar responses in simulation studies have been observed for once-through boilers and pressurized water reactors [42].

During the delayed fuel feed boost T_d the boiler output depends on the storage response represented by the large time constant C_b . As there is little or no change of heat input to the boilers for several seconds after following a disturbance (see Ashmole et al. [44]), a good assumption to simplify the generation plant model is to omit the boiler representation for at least a period of time T_d , holding the boiler output to initial value.

2.4 - ELECTRICAL LOADS

A general model which relates changes in active and reactive load demand to changes in voltage and frequency (Berg [45]) is employed to represent the loads according to the following functions

$$P_l = K_p |V|^{p_v} (f)^{p_f} \quad (2.20) \quad \text{where} \quad K_p = P_o |V_o|^{-p_v} (f_o)^{-p_f} \quad (2.20a)$$

$$Q_l = K_q |V|^{q_v} (f)^{q_f} \quad (2.21) \quad K_q = P_o |V_o|^{-q_v} (f_o)^{-q_f} \quad (2.21a)$$

The frequency is the bus frequency which is computed as the rate of change of absolute bus angle. The voltage and frequency exponents are the characteristic load parameters and for common load types, normally range from 0 to 3 in magnitude, with q_v having the highest value. For static loads $p_f = q_f = 0$.

This load representation is very useful in studies where the effect of different types of loads are considered, since it is possible to switch from one representation to another just by changing the exponent value.

2.5 - RELAY REPRESENTATION

The principal purpose of relays is to protect the power system from effects of faults by initiating circuit breaker operations to disconnect the faulted equipment (Kimbark [24]).

The following types of relays are considered for the protection of power system elements:

- . Over/under frequency relays
- . Under voltage relays
- . Distance relays

2.5.1 - Over/Under frequency and Under voltage Relays

Frequency and voltage relays are used for load shedding/restoration and generation unit protection; the voltage relays are also used for switching shunt capacitors and reactors (Podmore et al.[4]).

These relays will operate if the frequency or voltage magnitude of a sensed network bus violates (for certain time) either an upper or lower limit. The information required for the representation of the relays is the tripping time (relay plus breaker time) and the upper and lower settings. When the bus frequency or voltage magnitude violates the relay settings, the time is stored and the relay begins to operate, if the values come off the limits the relay is assumed to reset instantaneously, unless the time since the relay began to operate is more than the relay operating time then the relay proceeds to trip the protected element.

2.5.2 - Distance Relays

Distance relays or impedance relays are offset mho relays which monitor the apparent impedance of lines and trip the line if the impedance enters a specified protection zone (Young [46]). The operating characteristics of these relays are represented by circles drawn on an $R\text{-}X$ diagram with the radius equal to the impedance settings as shown in Fig. 2.16. Information related to the operating characteristics of the relay includes, the diameter or diameters of the circles for each protective zone, the angle ϕ with respect to the R axis of the line along which the centres of the circles lie, and the positions of the centres of the circles along this line (Stagg and El-Abiad [47]).

The apparent impedance for bus 1, is calculated from

$$Z_1 = R_1 + jX_1 = z_{1j} V_1 / V_{1j} \quad (2.22)$$

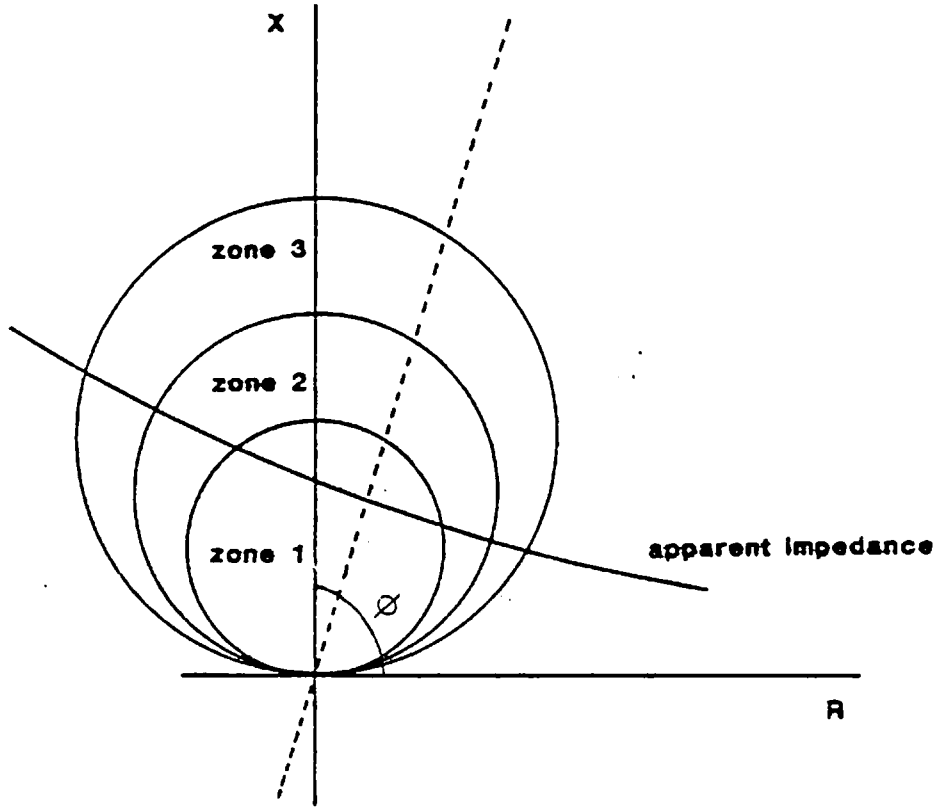


Fig. 2.16: Operating Characteristics of a Distance Relay.

2.6 - SYSTEM CONTROL CENTRE

The energy control centre is the most important decision making element in the system. Features and on-line functions that the control centre of a bulk power system would have, are listed in Dy Liacco [48]. As the implementation of all the functions enumerated in [48] is beyond the scope of this work, only the following will be considered:

- **Security Analysis.** The operator can simulate disturbances to evaluate the system response and determine whether the system is secure or not under those disturbances. The most common types of disturbances are: Changes in network configuration and changes in the power injection to the network caused by the partial or complete loss of a generator, load or tie line.

- . Limit Checking. Key variables are checked to see if they are within prescribed limits. Any out of range condition is signaled to the operator.
- . Emergency Control. When there are major violations on network variables, the operator may take corrective actions such as: Alteration of the scheduled power output of generators, changes in network configuration, and load disconnections.
- . Restorative Control. The operator may restore service to disconnected loads and reconnect generation units or reclose lines that had been tripped out during an emergency.

The only on-line function considered here is the monitoring of system variables. All the others are manual functions that will be set by the operator.

2.7 - COMMENTS

The linear models selected to represent the control elements of a generator are valid for short/mid term dynamic studies. The complete model of a power plant comprises a 20th order system of state equations for a steam generation plant and a 16th order model for hydro plants.

Due to the great differences in the rate of response and associated effects of the power system components it is customary to include in the system model only the elements relevant to the study under consideration. For example, the boiler representation is relevant for studies carried out beyond 10 s [17], therefore, the boiler model can be omitted from the simulation for 10 s after following the initial disturbance without introducing significant inaccuracies in the simulation.

The different time constants of the generation plant models and the nonlinearities introduced by dynamic saliency or nonadmittance loads in the network equations have to be taken into consideration for the selection of efficient solution techniques.

Solution techniques to perform different power system studies are discussed in chapter 4 through 7 and typical values for the parameters of the models described in this chapter are given in the Appendices C and D.

CHAPTER THREE

SOFTWARE AND HARDWARE CHARACTERISTICS

3.0 - INTRODUCTION

With the availability of software development systems, software portability, life-cycle maintenance and program support environment no longer represent serious restrictions on the utilization of different families of microprocessors for a given application, since the development systems have the capability of generating executable microprocessor code for a variety of microprocessors from source programs written in a high level language.

With the continuous evolution in microprocessor technology it was considered important in this project to implement the computational algorithms in a high level language and to employ a development system for the production of microprocessor code. The high level language PASCAL was considered because of its structured characteristics which enhance programming and ease program maintenance.

The computational equipment already available for the project consists of a software development system supporting PASCAL and five microprocessor units.

In this chapter, descriptions are given of the software and hardware characteristics, the interconnection between microprocessors to construct multiple processor structures, and important features of communication procedures for the implementation of parallel processing algorithms.

3.1 - PASCAL FEATURES

PASCAL is a highly structured language that allows the efficient implementation of large programs giving programmers enormous flexibility to design data type structures, promoting top-down programming and minimizing life-cycle cost maintenance. Although PASCAL code typically takes twice as much memory space and does not execute as fast as machine code, it has been found to be well suited

for microprocessor software programs (Wickham [49]). PASCAL has been suggested in Enns et al.[52] as a competitive alternative to traditional languages in implementing power system analysis programs.

Without attempting to give a detailed description of the PASCAL language, programming features of interest from a point of view of power system applications are highlighted next.

3.1.1 - Control Structures

Structured programming as defined by N. Wirth [50] is the formulation of programs as hierarchical nested structures of statements and objects of computations.

Four basic control structures are sufficient in structured programming for expressing any flowchartable program logic (see, for example, Jensen [51]). With these basic structures, shown in Fig. 3.1 it is possible to write programs in languages like PASCAL without an explicit GO TO statement. Each control structure has exactly one entry and one exit. The process structure can be as simple as an elementary assignment statement or as complex as an entire program segment. Even though some systems support the exit structure it is not a standard PASCAL feature. The control structures in PASCAL are:

- . Selection:- IF-THEN-ELSE
 CASE-OF
- . Iteration:- WHILE-DO
 REPEAT-UNTIL
 FOR-DO
- . Exit:- ESCAPE

3.1.2 - Data Structures

The data structures available in PASCAL are the ARRAY, the RECORD, the SET and the FILE. The structures are made of different data types. Its components can be simple standard data types, data structures or user-defined data types.

The size of an array in a PASCAL program must be specified at

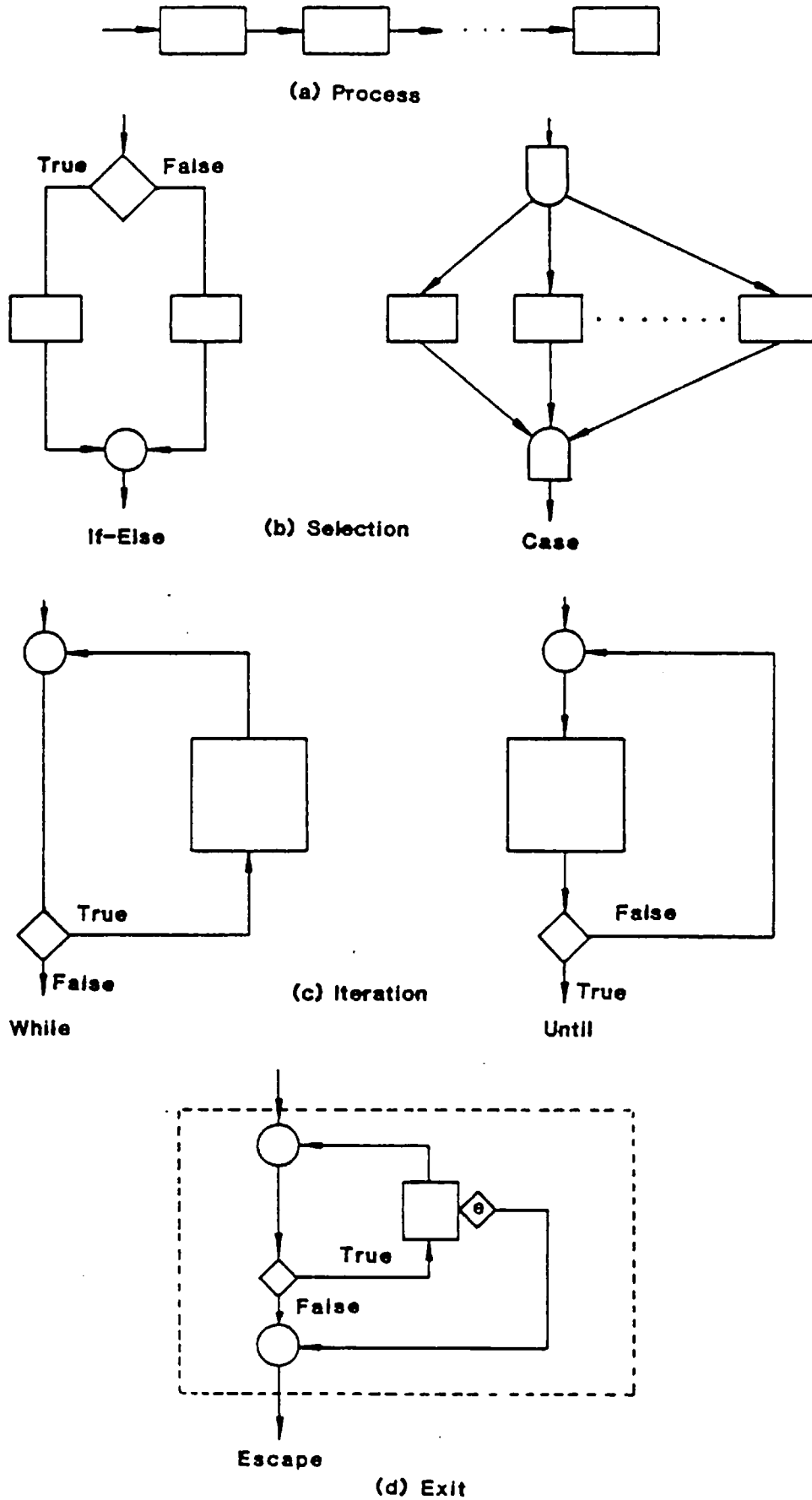


Fig. 3.1: Four Basic Control Structures

compile time and there is no way to change it during program's execution (Tenenbaum and Augenstein [53]). PASCAL allows the implementation of variant records permitting a RECORD variable to be interpreted in different ways. If a RECORD is regarded as a road map to an area of memory, a variant record provides several different road maps for the same area of memory, and a tag field value determines which road is in current use. All basic SET operations are supported by PASCAL, i.e. union, intersection, etc.

The COMPLEX data type is not predefined in PASCAL but can be defined as

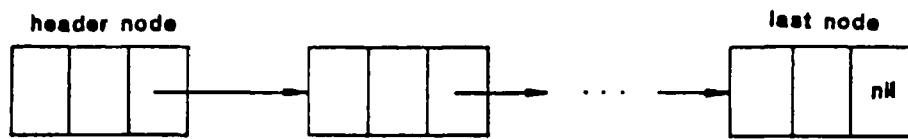
```
TYPE      COMPLEX = RECORD
                                re, im : REAL
                                END;
```

Arithmetic operations on complex numbers in PASCAL must be performed explicitly or by procedures coded for that purpose.

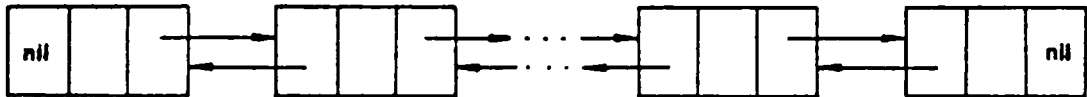
3.1.3 - Dynamic Data Structures

One of the most powerful PASCAL features is the POINTER type manipulation which allows the creation of dynamic data structures (DDS) that can be used instead of variable size arrays in situations where the number of elements varies from run to run as in the case of algorithms using sparsity techniques. A pointer is the actual address of a portion of storage that can be dynamically allocated and freed. By altering the values of pointers, portions of storage called nodes can be attached, detached and reassembled in patterns that grow and shrink as execution of a program progresses. A node generally contains an information field and a next address field. The value NIL is assigned to the next address field if the node is the last of a list [53].

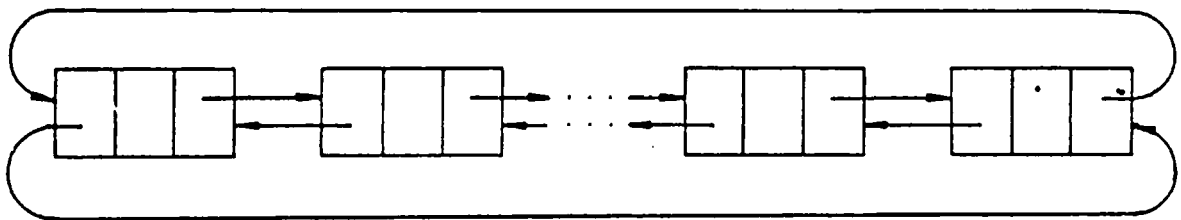
The implementation of linked lists using DDS does not require the specification in advance of the storage needed to allocate the list. Three examples of linked lists are shown in Fig. 3.2. By using the standard procedures NEW and DISPOSE it is very easy to insert or delete elements in a list. All that is required in each case is an adjustment of the links.



(a) Linear linked list



(b) Linear doubly linked list



(c) Circular doubly linked list

Fig. 3.2: Examples of Linked Lists.

3.2 - SOFTWARE DEVELOPMENT SYSTEM

The software development system consists of a Texas Instruments FS990/4 minicomputer, a TI-AMPL (Advanced Microprocessor Prototyping Laboratory) system and the TI Microprocessor PASCAL system. AMPL is a special software/hardware system with emulator and trace capabilities that is used in conjunction with the FS990 minicomputer for prototype development or software debugging purposes.

The Microprocessor PASCAL system is a software package supporting development and execution of a superset of the PASCAL language for execution on host computers or target systems of the TI-9900 family of microprocessors. The TI PASCAL System supports concurrent or multitasking execution whereby a number of routines share

simultaneously a single processor during execution. Also, reentrancy is supported by the system, allowing the independent execution of a code module on separate areas of data at the same time [54].

Compilation of groups of modules (programs, procedures, etc.) that are to be linked or executed together may be carried out separately and each segment may then be saved for later use. This facility enables the user to create library packages and to work independently on smaller files. Various options are available for controlling the execution and output of the PASCAL compiler. For more details of TI PASCAL features and differences with the standard PASCAL language, see the system user's manual [54].

3.2.1 - Code Generation

The generation of microprocessor code from PASCAL, involves the collection and linking of all code specified to the system, including required modules of the interpreter and required routines of the Run Time Support system. The code production process consists of the following steps [54]:

- . Compile and save all user segments
- . Collect required routines from the Run Time Support library
- . Link the user segments, the user initialisation module and the user configuration module

The configuration module, allows the user to specify the organization of the target microprocessor's memory (RAM/ROM boundaries) and the locations of the system restart and reload vectors. The user initialisation module basically comprises assembly language interrupt handlers and user coded crash routines. For a detail description of the code generation process see [54].

Once produced, the code is stored on floppy disks and is ready to be loaded onto the target microprocessor unit for execution.

3.3 - MICROPROCESSOR UNITS

Each microprocessor unit (MPU) is formed by a TI-TMS9900 microprocessor and a memory expansion board. The TMS9900 is a 16-bit processor operated at a 3-MHz clock rate with a memory to memory architecture, having its work space (working registers) in the system

random access memory. The maximum memory addressability is 64 Kbytes and the I/O functions are performed through a serial I/O port called the CRU (Communication Register Unit). The MPUs have stored in PROM a program that permits the loading of programs to RAM memory.

The microprocessor board contains two serial I/O RS-232C ports and a programmable parallel port. The parallel port has the ability to transmit interrupts or transfer fields from 1 to 16 bits of data as inputs or outputs under the command of software instructions. Parallel I/O (PIO) can be expanded by the addition of auxiliary boards with three extra PIO ports each.

The TMS9900 computational speed for the basic floating point arithmetic operations is given in Table 3.1. Floating point values are represented by 2 words of 2 bytes each, using 8 bits for the exponent and 24 bits for the mantisa.

TABLE 3.1

Average Execution Times for Floating Point
Arithmetic Operations (ms)

Addition	1.100
Subtraction	1.300
Multiplication	1.075
Division	1.852

3.4 - MICROPROCESSORS INTERCONNECTION STRUCTURE

With the hardware available, the interconnection between MPU's is via 20-way ribbon cables connecting the MPU's through the PIO ports. The interconnection of four MPU's in a "star" configuration suitable for BBDF algorithms is illustrated in Fig. 3.3. The central MPU contains an auxiliary PIO board to make possible this interconnection. If more than four MPU's are to be connected to the central MPU, additional PIO boards may be included in the central MPU.

This particular method of interconnecting MPU's, represents a very rigid scheme and makes it difficult to perform variations in the hardware configurations and to implement algorithms which would be

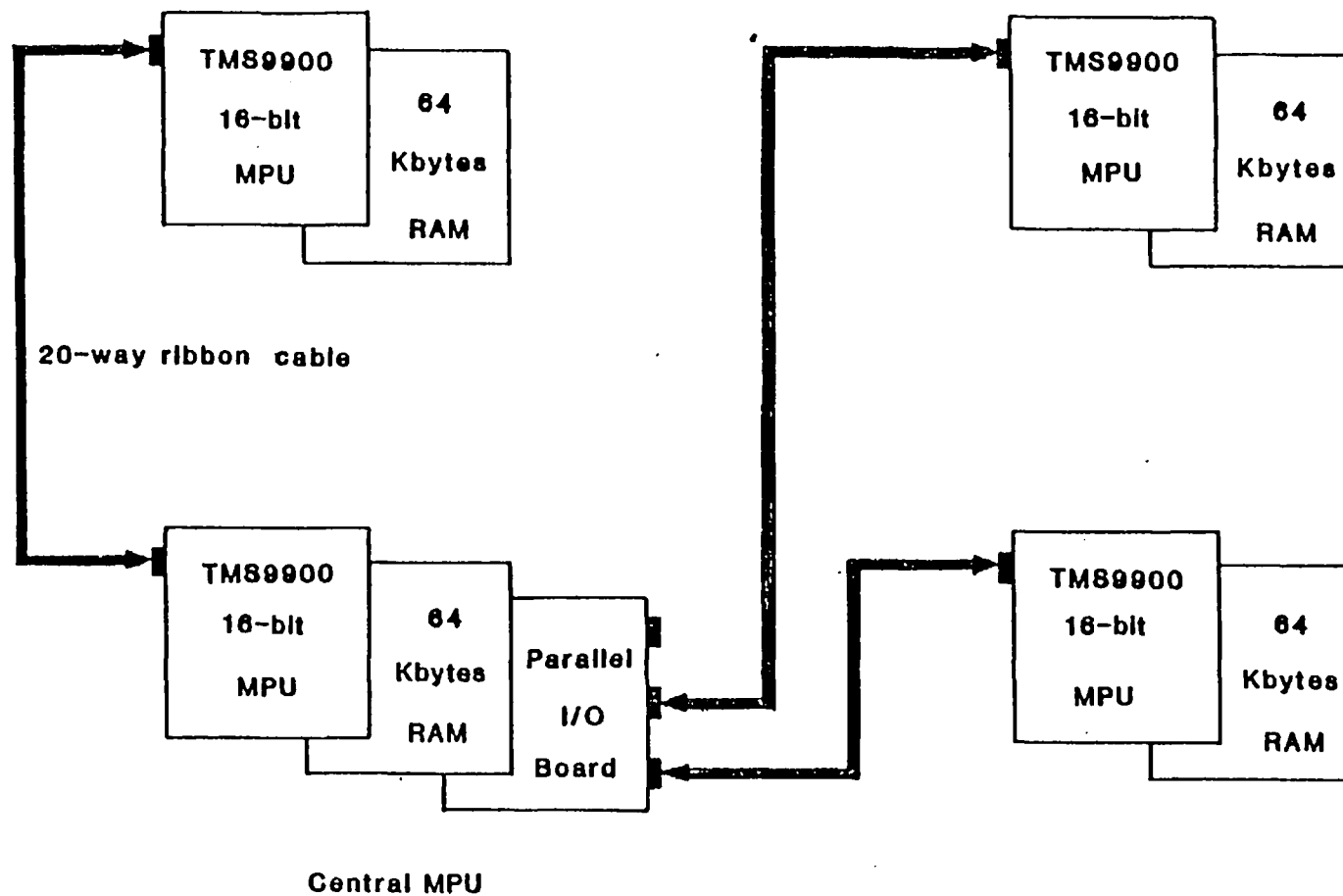


Fig. 3.3: Interconnection Structure of 4 MPUs Suitable for BBDF Algorithms

possible with the use of a parallel data bus.

3.5 - INTER-PROCESSOR COMMUNICATION PROCEDURES

The communication between MPU's through the PIO ports via the 20-way ribbon cable is accomplished by interrupt driven procedures which transfer a real variable between two MPUs in less than 1.0 ms. The procedures implemented in assembly language by Johnson [109] can be called from PASCAL programs and in addition to communicating information, the procedures may be used to synchronize the MPU's during the execution of parallel processing algorithms. For convenience, only the procedure characteristics that play a major role in the implementation of parallel algorithms are pointed out here. A detailed description of implementation and communication protocols is given in [109].

The procedures transmit information between MPU's using communication buffers consisting of the following information:

DESTINATION	MESSAGE	STATUS
-------------	---------	--------

Fig. 3.4 - Communication Buffer

The destination contains the parallel port address through which a message is sent or from which a message is fetched. The message contains the transferred information which can be any data structure with a variable length of an even number of words. The length of a message is given by half the number of binary words forming the message. For real values, the length is equal to the number of values and for integers, it is equal to one half of the number of values in the message.

The status gives users at the application level an indication whether the message in the buffer is old or new, or whether any messages have been lost, or if the length of a received message is different from that of the message transmitted.

3.5.1 - Initialisation Procedure

Before attempting to transfer any information between MPU's the procedure INITIO should be called to initialise the parallel ports and to clear the communication buffers.

The information required by INITIO is the number of ports to be initialised, the communication buffers and the message's length. The first port initialised always corresponds to the PIO on the microprocessor board. If variable length messages are to be transmitted then the maximum length should be specified.

3.5.2 - Sending/Receiving Procedures

Sending and receiving information between MPU's is performed by two pairs of procedures:- PSEND/GETMSG and PGET/PUTMSG.

PSEND is used to transfer information from the communication buffer of the sender MPU into the receiver's buffer. As the communication buffers are not accessible to users, the procedure GETMSG is used to get a message from the communication buffer into a user buffer and to flag the current buffer status at the application level (PASCAL program).

When PSEND is called, the sender MPU issues an interrupt to the receiver MPU in order to place the message in the receivers buffer. After transferring the information, both sender and receiver continue execution. When GETMSG is invoked, the buffer status is checked and if there is not a new message the MPU waits until a new message arrives. To transfer the message into the user buffer, GETMSG disables any incoming interrupts to protect the message in the buffer from possible overwriting by new incoming information. After getting the message, control is transferred to the user program.

The information required by PSEND is the port through which the message is to be sent, message length and the message itself. GETMSG requires the buffer from which a message is to be fetched, message length and user buffer into which the message is to be placed.

As an example to illustrate the application of these communication procedures, consider the interconnection of two MPUs as shown in Fig. 3.5. Assume that the MPU 2 needs updated values of V , which are calculated in the MPU 1, to obtain new values of I , and that MPU 1 needs updated values of I to calculate new values of V and so on. Suppose that an initial value for I is known, then the implementation of a

repetitive procedure for the process described is given in Fig. 3.5. Notice that no matter how long it takes one MPU to calculate new values of V or I , the other will always wait for updated values to calculate its new values respectively.

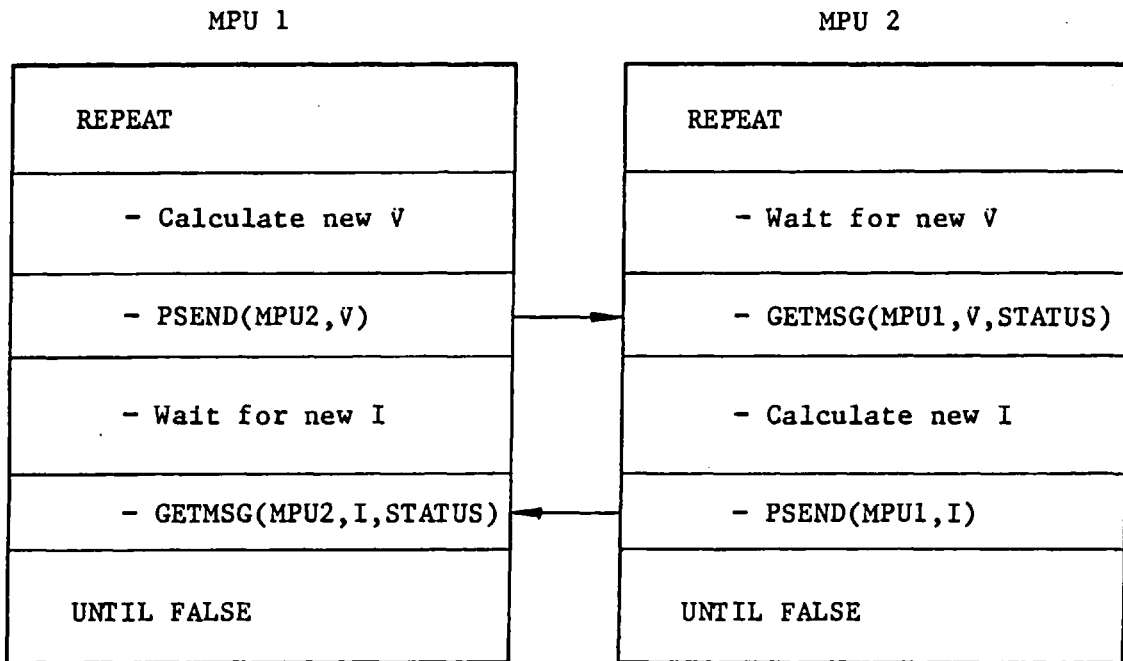


Fig. 3.5: Example Illustrating the Application of PSEND/GETMSG

Coordination between MPU's during the execution process may be accomplished using these procedures to transmit acknowledgment signals between MPU's instead of messages. For instance, a control scheme to avoid losing any messages is presented in Fig. 3.6. The sender MPU will always wait for a ready-to-receive message from the receiver MPU before sending the message, thus when the message is sent the receiver is waiting for it.

PUTMSG/PGET may be used for the same purposes of PSEND/GETMSG but work in a completely different manner. PUTMSG puts a message in a specified communication buffer on the MPU calling it without checking if it is overwriting values not yet read by the destination MPU. PGET gets a message through the PIO from the addressed MPU buffer without checking whether it is a new message or not. In both cases it is the user's responsibility to do status checking if required.

The distinction between the communication buffers used by

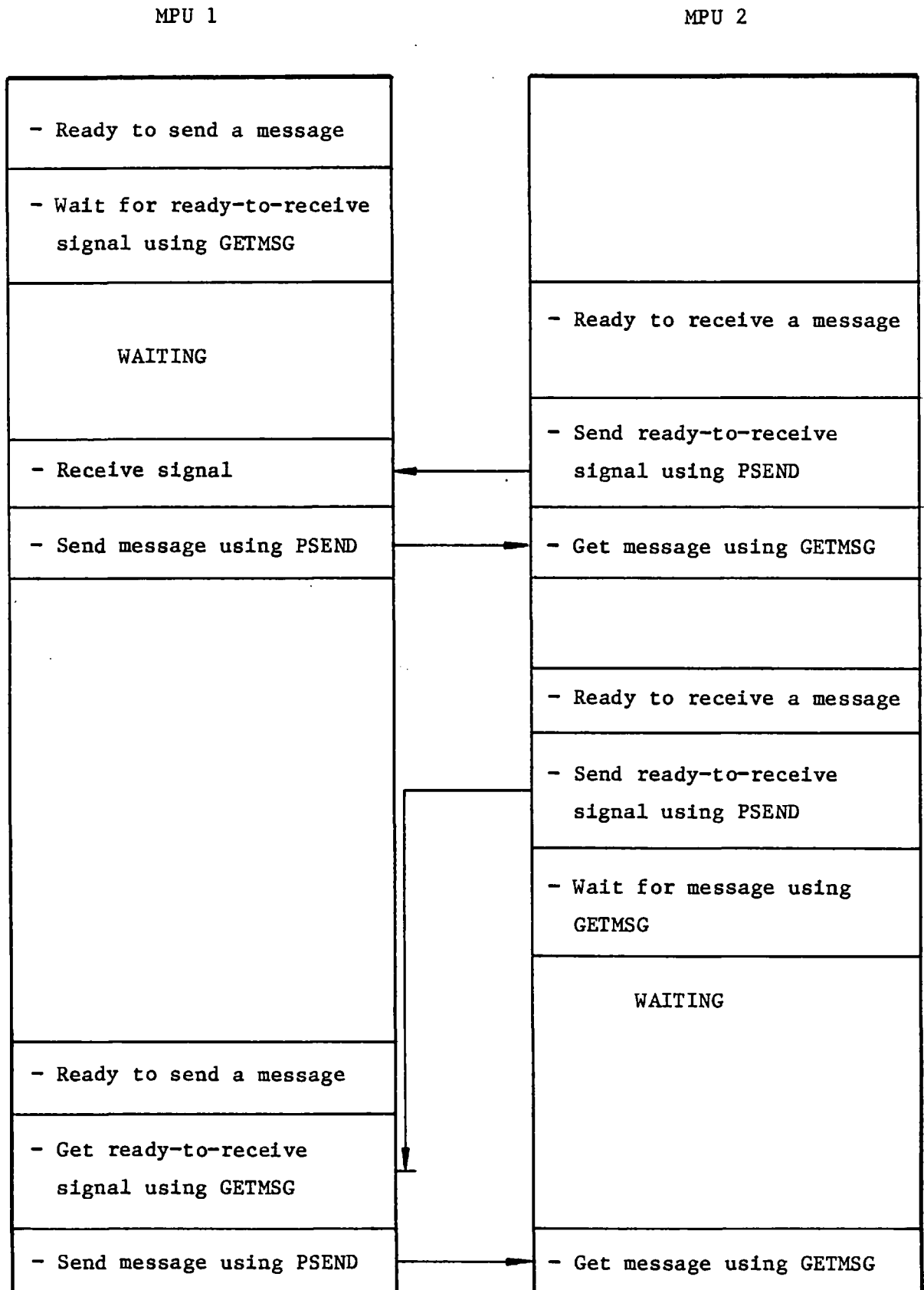


Fig. 3.6: Coordination Scheme Between Two MPUs to avoid Losing any Transmitted Message Using PSEND/GETMSG

PSEND/GETMSG and PUTMSG/PGET is made by assigning a negative sign to the number of ports to be initialised when calling INITIO.

The pair of procedures PUTMSG/PGET is particularly suited for the implementation of parallel chaotic relaxation algorithms, where the latest values available from MPU's are used to carry on calculations without regard to which stage of the calculation produced them [13]. As soon as new values have been computed by an MPU, they are made available to other MPUs by calling PUTMSG to update the message in the communication buffers. When an MPU needs values from another MPU to continue execution, it calls PGET to get the available values from the addressed MPU regardless of how old or new the values are.

The communication procedures described above are designed to communicate with directly connected MPUs only. If the communication between indirectly connected MPUs is required, a buffering of messages scheme could be implemented. For example, in the MPU configuration of Fig. 3.3 all MPUs can communicate directly with the central MPU and vice versa but the communication between MPUs in the periphery cannot be performed directly because they are not linked. In this case, the central processor may get the message from the sender MPU and transfer it to the receiver MPU. Of course, this problem could be solved as well by linking all MPUs using auxiliary PIO boards.

3.6 - OBSERVATIONS

The software and hardware characteristics must be taken into consideration for the development of computational algorithms in the most efficient manner possible.

Highly structured programs can be written in PASCAL by eliminating the use of GO TO statements. The flexibility to define and manipulate data structures in PASCAL should be exploited to implement power system analysis programs especially when programming the sparsity-oriented techniques required for the efficient handling of large power networks.

The TMS9900 execution times for floating point arithmetic operations should be considered when programming repetitive inner loops or iterative calculations. Whenever possible, multiplications and additions should be used instead of divisions and subtractions respectively. In general, the number of arithmetic operations

should be kept to a minimum. Bear in mind that a 32-bit floating point representation is often inadequate for power system computations.

For the design of multiple microprocessor configurations with the available hardware it should be noted that each MPU has a local memory which only its own processor can access and that the interconnection between MPUs is through their parallel I/O ports. It is emphasized that the communication procedures play a major role in the implementation of parallel processing algorithms for the synchronization and interchange of information between the microprocessors.

The implementation of parallel processing algorithms using the described communication procedures for the simulation and analysis of power systems is discussed in chapters 5 and 7.

CHAPTER FOUR

STEADY STATE POWER SYSTEM ANALYSIS ON A SINGLE MICROPROCESSOR

4.0 - SUMMARY

The capabilities of a single microprocessor for the solution of power system problems are explored using a software package implemented in PASCAL for steady state analysis.

A TMS9900 microprocessor with 64 K-byte of memory was used to perform ac load flow studies and to evaluate the steady state response of power systems under specified contingencies. The contingency analysis algorithm is a practical computational tool that can be used for rapid evaluations of changes in the system from an available base-case load flow. The sparsity techniques required for the manipulation of large electric networks are efficiently implemented by taking advantage of properties of the network admittance matrix and by exploiting fully the application of dynamic data structures (DDS) which are possible in PASCAL. With the developed programs it is possible to handle networks of up to 400 buses on a single TMS9900.

The software package organization and the implementation of algorithms are described in this chapter and computational results obtained with the TMS9900 in the steady state analysis of several test systems of different sizes and characteristics are presented.

4.1 - SPARSITY PROGRAMMING TECHNIQUES USING DDS IN PASCAL

Sato and Tinney [57], and Carpentier [58] demonstrated that significant reductions in memory and processing time are possible from careful exploitation of sparsity in matrix calculations. Duff surveyed the state of the art in sparse matrix research and examined design considerations when implementing sparse matrix algorithms [59].

The different computational aspects involved in the application of sparsity techniques to the analysis of large network problems may be

organised as four basic procedures [60]:

- . Compact storage of the elements describing the problem,
- . Determination of the elimination sequence,
- . Factorisation of the coefficient matrix and
- . The numerical solution of the problem.

Brameller [60] recommends a separate implementation of these procedures when programming the sparsity-directed techniques to avoid unnecessary repetition of calculations. For example, when the solution is required for several r.h.s. vectors the same factorised matrix should be used. Similarly, when several solutions are required with different coefficient matrices having the same structure, the original elimination sequence should be used for all the required solutions.

Schemes for storing and manipulating large sparse matrices efficiently in a compact form consist of linked lists which describe the structure of the matrix and identify its elements. Linked lists have been implemented in power system programs using a set of parallel arrays (for example Brameller [60]), using a contiguous set of memory locations (Ogbouobiri [61]) or using DDS in PASCAL (Enns et al. [52]).

This work follows the recommendation in [60] to program in separate procedures the compacting, ordering, factorisation and solution of large sparse sets of equations, and the linked lists required for storing nonzero elements are implemented using DDS in PASCAL as described below.

4.1.1. - Sparse Storage Scheme and Allocation Procedure

One important consideration in the implementation of a sparse storage scheme is its efficiency, that is, the stored elements should be readily accessible, the storage required should be kept low and modifications should be accomplished easily. If a sparse matrix is to be factorised and used for computations, the storage scheme has a significant impact on the implementation of the factorisation and solution, and hence on the complexity of the programs and the execution time (George and Liu [62]).

The sparse admittance matrix $[Y]$ of a power system network is used to describe the developed storage scheme using DDS. For practical networks the admittance matrix $[Y]$ is symmetric and all the diagonal elements are nonzero and equal in number to the buses in the network.

The nonzero off-diagonal elements correspond to the branches in the network. The elements can be stored either by rows or columns and only the upper or lower part of the matrix needs to be stored. If the diagonal elements are stored in their natural order no indexing information is required to identify these elements since their relative position is known implicitly. If $[Y]$ is stored by rows, then only the column index of the off-diagonal terms needs to be kept for identification.

During the elimination or factorisation process the number of diagonal elements remains unchanged and the off-diagonal elements are continuously generated or deleted, these facts are taken into account in implementing an efficient sparse storage scheme. In the implemented scheme, the diagonal elements are stored in an array of dimension equal to the number of buses in the network and are selected as entry points to allow easy access to any stored element. The off-diagonal elements of each row in the upper part of the matrix are stored in a linear linked list created dynamically using pointer type variables in PASCAL and the entry points to the lists are stored with the corresponding diagonal elements. To retrieve a given element, determine the row in which the element belongs, obtain the entry point of the linked list of that row and search through the list until the element is found. The condensation or garbage collection process [61] is implemented by calling the PASCAL procedure DISPOSE to free memory space occupied by elements no longer needed. The released memory is automatically reused during program execution.

The allocation algorithm for the storage scheme outlined above is presented in Fig. 4.1 and may be described as follows: An array of type NONZERO and dimension N is declared to store the diagonal elements and the entry points to the linked list of each row, N being equal to the number of buses. Once a branch data is read in the PASCAL procedure NEW is called to allocate enough storage to hold the data and the dynamically generated storage is inserted in the linked list of the corresponding row. Typically, branch data are given by the branch end buses (i,j) , the branch impedance z_{ij} and one-half of total line charging y' . The branch admittance is calculated and the $[Y]$ elements are computed using (2.3)-(2.7). The number of nonzero elements or branches connected to each bus, required later by the ordering procedure, is stored in the field INDEX of the diagonal array.

```

PROGRAM sparsity ( INPUT,OUTPUT );
CONST  n = 400;(* maximum number of buses *)
TYPE   complex = RECORD
        re,im :REAL
      END;
branchptr = @nonzero;
nonzero = RECORD
        index   : INTEGER;
        value   : complex;
        nextcol : branchptr
      END;
diagonal = ARRAY[ 1..n ] OF nonzero;
VAR     y : diagonal;
        buses,branches : INTEGER,

PROCEDURE store(VAR buses,branches:INTEGER;VAR y:diagonal);
VAR      branch : branchptr;
        r,x,yc,g,b : REAL;
        i,j,k : INTEGER;
BEGIN
  READLN( buses,branches );
  (* initialise diagonal array *)
  FOR k := 1 TO nbuses DO
    WITH y[k] DO
      BEGIN
        index := 0;
        value.re := 0.0;
        value.im := 0.0;
        nextcol := NIL
      END;
  FOR k := 1 TO branches DO
    BEGIN
      (* read branch data *)
      READLN( i,j,r,x,yc );

      (* calculate branch admittance g+jb *)

      (* allocate dynamic storage *)
      NEW( branch );
      WITH branch@ DO
        BEGIN
          index := j;
          value.re := -g;
          value.im := -b;
          nextcol := y[i].nextcol
        END;
      (* insert element in corresponding row *)
      y[i].nextcol := branch;

      (* form diagonal elements *)

      (* update no. of connected branches in i and j *)
      y[i].index := 1+y[i].index;
      y[j].index := 1+y[j].index
    END
  END(* store *),

```

Fig. 4.1: Allocation Procedure for the Developed Semi-dynamic Sparse Storage Scheme.

4.1.2 - Comparing the Dynamic and Array Implementation of Linked Lists

To compare the sparse scheme implemented using DDS and the conventional array approach consider the simple network shown in Fig. 4.2. The storage of the [Y] elements for this network using both schemes is illustrated in Figs. 4.3a and 4.3b respectively.

As the number of generated elements during the elimination process is not known at the time that a program is compiled, the arrays used to allocate the linked lists in the array approach should be dimensioned to provide enough storage to avoid overflow. The disadvantage of overdimensioning the arrays, which is necessarily incurred with this approach, is that the storage reserved for the declared arrays remains allocated throughout the program execution.

The major advantages of DDS over arrays is that there is no need to reserve in advance the storage needed for the linked lists; no storage is allocated until needed and as long as sufficient storage is available in the computer system memory no overflow will occur. Another advantage of the dynamic implementation is that reference to a particular element does not involve the address computation that is necessary in computing the address of a given element within an array.

To show the gains obtained when accessing elements in the dynamic implementation consider the retrieval of all stored elements in the linked list of the first row for the example given in Figs. 4.3a and 4.3b. If this process is performed 100 times the execution time on the TMS9900 for the dynamic scheme implemented is 606.6 ms whereas for the array approach it is 967.8 ms.

From the simple example given it is evident that for large networks the compact semi-dynamic storage scheme implemented is much more advantageous than the conventional array approach in terms of flexibility in manipulating the linked lists, storage requirements and computational speed in retrieving stored elements.

4.1.3 - Ordering, Factorisation and Solution Procedures

The basic objective of the ordering is to minimize the creation of nonzero terms during the factorisation process of a sparse matrix and, as a result, reduce the execution time for direct solutions.

Comparisons of the computational performance of a variety of ordering schemes [62]-[64] have shown that the minimum degree ordering algorithm [62], better known as Tinney's scheme 2 [65], is the most

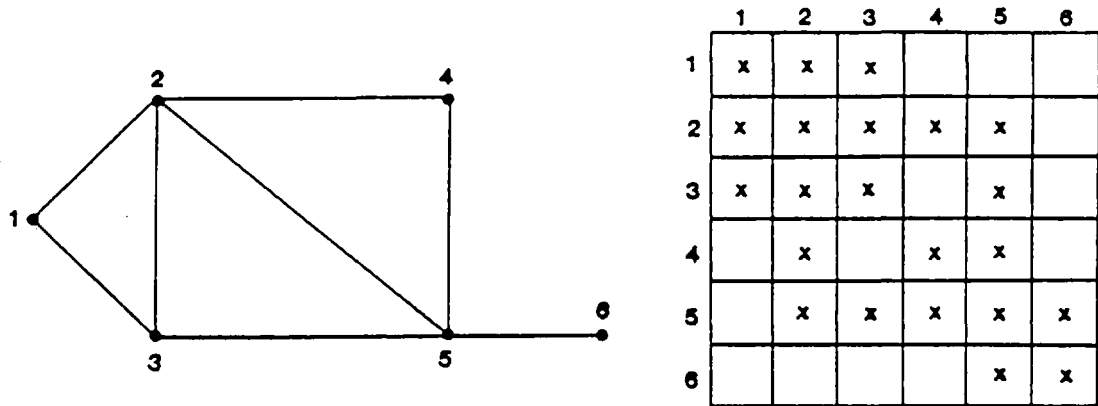
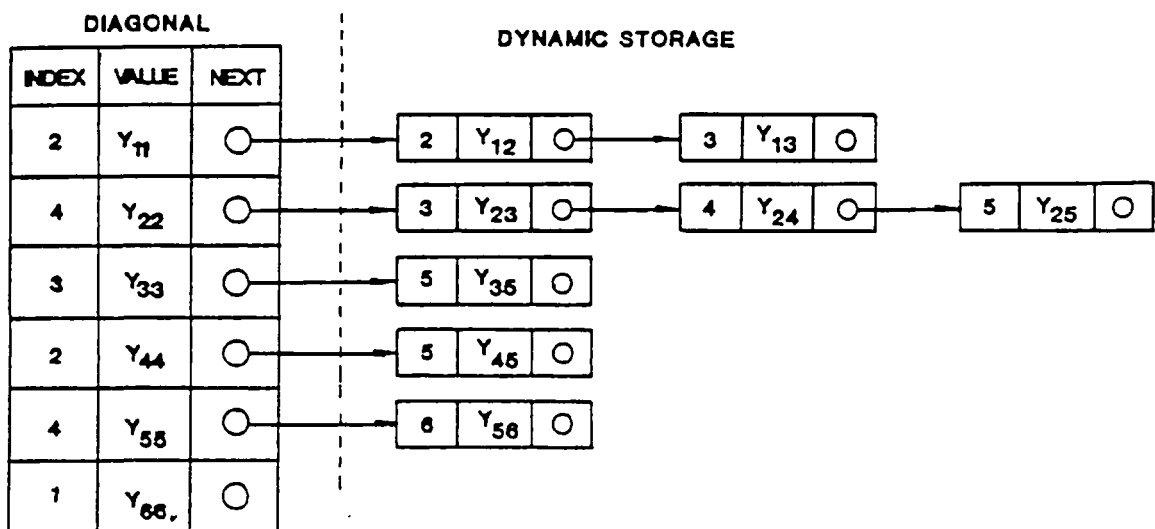


Fig. 4.2: A Sample Network

LCOL	NOZE	DIAG	ITAG	LNBT	CZ
1	1	2	1	2	Y_{11}
2	3	4	2	3	Y_{12}
3	7	3	3	1	Y_{13}
4	10	2	4	3	Y_{21}
5	12	4	5	4	Y_{23}
6	16	1	6	5	Y_{24}
			7	1	Y_{25}
			8	2	Y_{31}
			9	5	Y_{32}
			10	3	Y_{33}
			11	2	Y_{42}
			12	5	Y_{43}
			13	2	Y_{52}
			14	3	Y_{53}
			15	4	Y_{54}
			16	6	Y_{56}
			17	5	Y_{65}
			18	-	-
			19	-	-

(a) Sparse storage using arrays



(b) Sparse storage using DDS

Fig. 4.3: Example Illustrating the Sparse Storage of [Y] Elements

attractive ordering scheme in terms of computational speed and optimality for the solution of power system networks with sparsity directed Gaussian elimination. This dynamic ordering algorithm which is adopted here, simply consists of selecting at each stage of elimination the network bus with the least number of connected branches as the next bus to be eliminated. If more than one bus meet this criterion the choice is arbitrary [65].

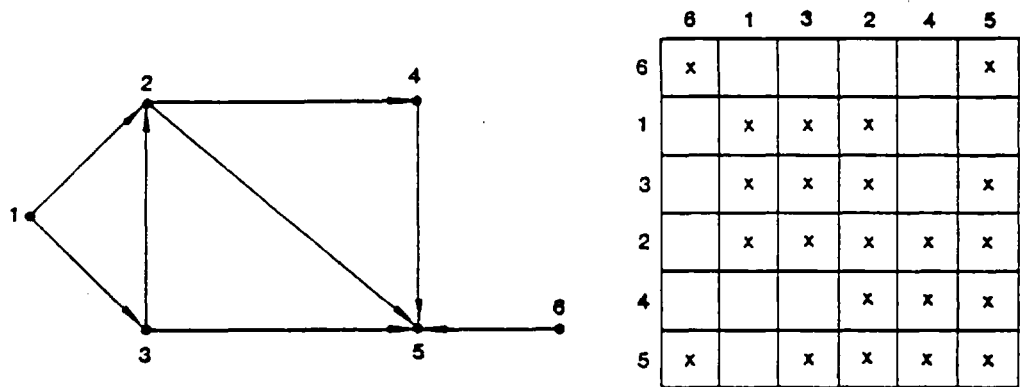
Advantage is taken of the fact that in a power network there is a one-to-one relationship between the admittance matrix and the network graph to program the ordering algorithm and factorisation procedures using linear graph theory [62]. The ordering procedure simulates the elimination process and the field INDEX of array Y containing the number of branches connected to each bus is overwritten by the established elimination sequence since that information is not needed after the ordering.

During the factorisation process most of the time is dedicated to searching the sparse storage for the elements to be eliminated and their interconnections. This time is normally reduced at the expense of memory storage by storing all nonzero elements in a matrix, including the case of symmetric matrices in which the stored elements are duplicated.

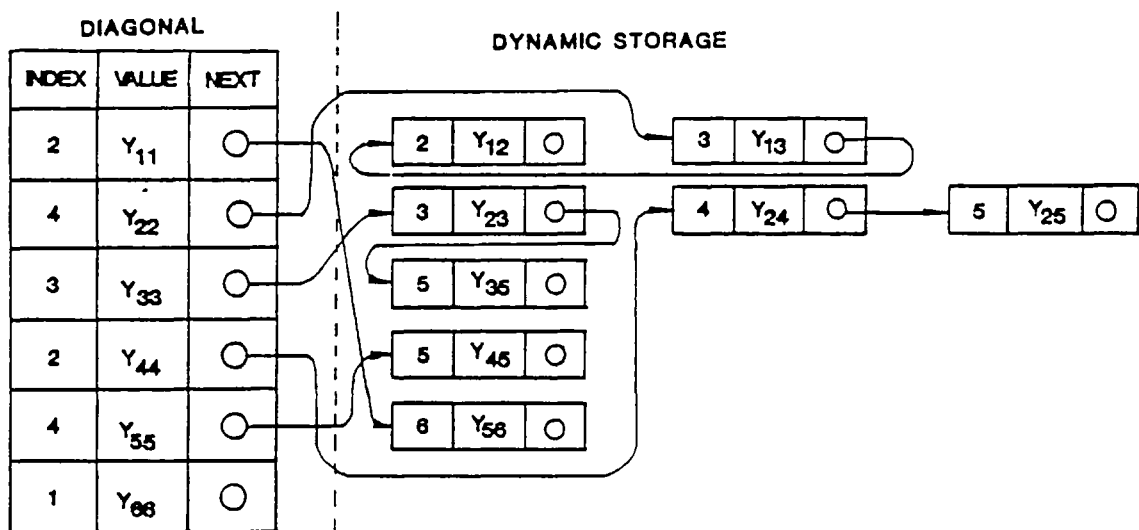
Searching time during the factorisation process is minimized in the software package implemented without increasing memory requirements by introducing an intermediate procedure between the ordering and factorisation. The intermediate procedure orientates the network by renumbering the buses according to their order in the elimination sequence and adjusts the links of the stored elements in such a way that the elements of the linked lists are chained in the elimination order. The information in the field INDEX of the diagonal array Y is used to access any element in the oriented data storage if that element is specified using the original bus identification of the network.

To illustrate the application of this procedure consider the example given in Fig. 4.2. The elimination sequence for this network is 6 1 3 2 4 5. The oriented network is shown in Fig. 4.4a. The diagonal elements are stored in the elimination order and the links are adjusted accordingly as shown in Fig. 4.4b. Finally, Fig. 4.4c shows the storage of the oriented network after renumbering the buses.

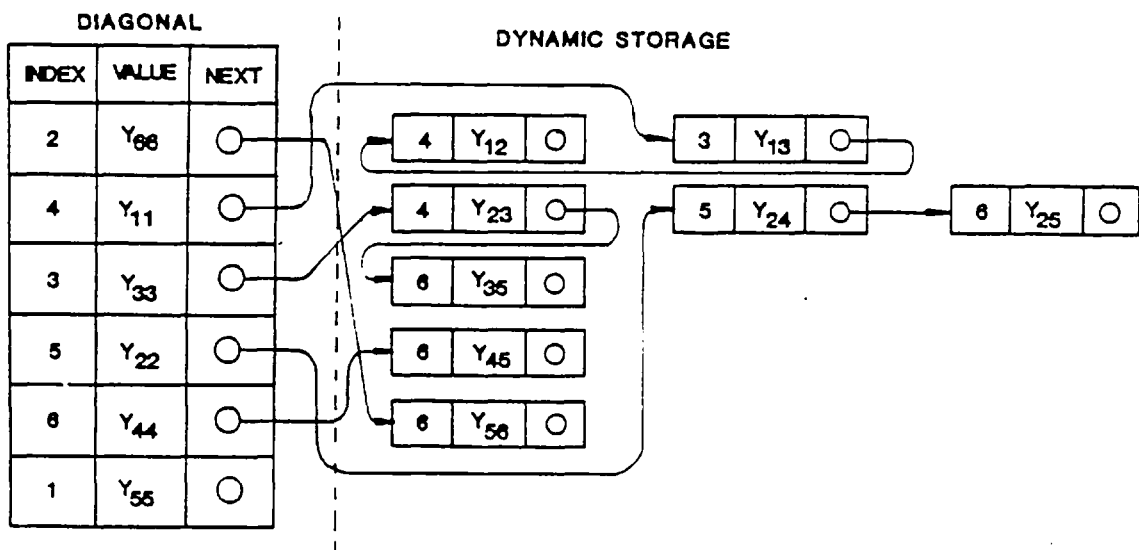
This network orientation procedure not only speeds up the factorisation but also greatly simplifies the programming and reduces the required indexing in the solution process.



(a) Oriented network



(b) Links adjustment



(c) Oriented sparse storage

Fig. 4.4: Orientation of Networks

The procedure implemented for the factorisation is based on the Zollenkopf bi-factorisation technique [66]. A factorisation step is described as follows: If bus i is eliminated, the numerical value of all elements connected to it are overwritten by the computed factors. The reduced network is determined by creating new elements between any pair of buses (j,k) if both j and k are neighbours of i but are not neighbours of each other. If an element between j and k already exists then its value is modified to reflect the bus elimination. All other elements remain unaltered.

After factorising the sparse coefficient matrix the direct solution of the unknown quantities of a linear system of equations can be obtained by performing forward and backward substitutions with the matrix triangular factors [66].

4.1.4 - Computational Results on a TMS9900 Microprocessor

The performance of the sparsity algorithms described in the previous section and programmed in PASCAL was evaluated using test power system networks of different sizes and characteristics. The test systems used are: The WSCC (Western States Coordinating Council) 9-bus test system [27], The IEEE 24-bus reliability test system [67], the 14, 30 and 57-bus IEEE load flow test systems [68], the New England 39-bus system (EPRI test system), the IEEE 118-bus system [100] and a 400-bus network which is part of the British Grid System [70]. The test systems are described in the Appendix C except the 400-bus network. Table 4.1 lists general characteristics of the networks and gives the

TABLE 4.1

Characteristics of Test Networks and Storage
of Nonzero Elements in [Y]

Buses (n)	PV Buses	Branches (b)	Ratio b/n	No. of Nonzeros in [Y]	No. of Nonzeros in the factorised [Y]
5	1	6	1.200	11	12
9	2	9	1.000	18	21
14	4	20	1.429	34	38
24	10	34	1.417	58	76
30	5	41	1.367	71	85
39	9	46	1.179	85	106
57	6	78	1.368	135	194
118	53	179	1.517	297	383
400	58	515	1.288	915	1216

number of nonzero elements stored in [Y] before and after the factorisation.

The execution times obtained on the TMS9900 microprocessor for the different sparsity procedures are given in the Table 4.2 and plotted in Fig. 4.5. From these results it can be seen that the time for the factorisation and solution varies linearly with the number of buses in the network and that the time taken for the factorisation is less than the time required by two repeat solutions. Most of the computing time is spent in simulating the ordering process and performing the network orientation. For large networks, the orientation is very time consuming but its incorporation as a preprocessing stage simplifies the handling of large networks on the microprocessor.

TABLE 4.2

Execution Times on the TMS9900 Microprocessor for
the Sparsity Procedures Implemented in PASCAL (s)

Network	Ordering	Orientation	Factorisation	Direct Solution
5	0.258	0.066	0.337	0.238
9	0.635	0.163	0.681	0.442
14	1.178	0.425	1.040	0.736
24	2.653	1.117	2.060	1.345
30	3.222	1.643	2.132	1.537
39	4.534	2.957	2.560	1 868
57	9 448	6.186	6.065	3.523
118	22.066	24.912	10.044	6.761
400	144.015	289.788	32.401	22.084

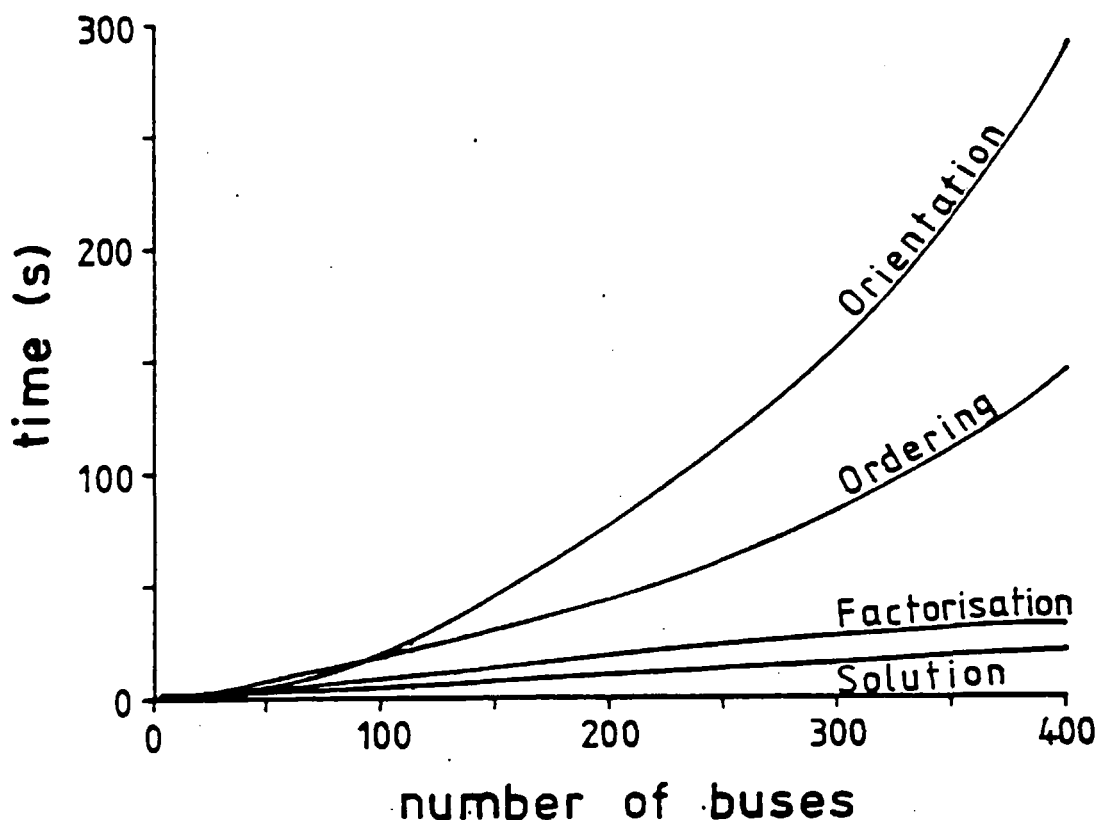


Fig. 4.5: Execution Times on the TMS9900 Microprocessor for the Sparsity Procedures Implemented.

4.2 - LOAD FLOW STUDIES

The load flow problem consists of the calculation of bus voltages and line power flows of an electric power network for specified terminal or bus conditions. The problem is specified by a schedule of complex power injections at load buses (PQ type) and real power injections and voltage magnitudes at generator buses (PV type). One generator bus is selected as slack bus, with only its complex voltage scheduled and its complex power is determined after the network solution [47].

The power flow equations are obtained by combining (2.1) and (2.2) to give

$$S_i = v_i \sum_{j \in 1} Y_{ij}^* v_j^* \quad (4.1)$$

Because of the product of voltages which are the problem unknowns,

the power flow equations (4.1) are non-linear and therefore some kind of iterative solution technique must be used. Many different solution methods have been published but the most satisfactory solution schemes are forms of Newton's method [18]. The choice of a load flow method for its incorporation in the software package being described was based on a study carried out to compare the performance and computer requirements of different solution algorithms using a CDC Cyber 174 computer [93]. From the results obtained the Fast Decoupled Load Flow (FDLF) by Stott and Alsac [69] was selected because of the many advantages that the method offers in terms of calculation speed, storage requirements and reliability.

4.2.1 - Fast Decoupled Load Flow

The FDLF method has had a widespread acceptance in the power industry since its publication [69]. The method's formulation is given by the decoupled equations:

$$[B'][\Delta\theta] = [\Delta P/|V|] \quad (4.2)$$

$$[B''][\Delta|V|] = [\Delta Q/|V|] \quad (4.3)$$

The matrices $[B']$ and $[B'']$, which are derived directly from $[Y]$, are square, symmetric, real matrices of susceptances only. The dimension of $[B']$ is one less than the number of buses and the dimension of $[B'']$ is equal to the number of PQ buses. Shunt susceptances and transformer off-nominal taps are omitted from $[B']$. Similarly, phase-shifts are not represented in $[B'']$. The matrices need to be factorised only once at the beginning of the solution.

The power mismatches are given by

$$\Delta P_i = P_i^{SP} - |V_i| \sum_{j \in i} |V_j| (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \quad (4.4)$$

$$\Delta Q_i = Q_i^{SP} - |V_i| \sum_{j \in i} |V_j| (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) \quad (4.5)$$

where $\theta_{ij} = \theta_i - \theta_j$

The iteration scheme of the FDLF solves (4.2) and (4.3) alternately, always using the most recent voltage values in the mismatch computation. Each iteration cycle comprises one solution for $[\Delta\theta]$ to update $[\theta]$ (half iteration) and then one solution for $[\Delta|V|]$ to update $[|V|]$. To avoid unnecessary extra iterations a tolerance of 0.01 MW/MVAR for the power mismatches is recommended in [69] to obtain acceptably accurate load flow solutions.

In the implementation of this algorithm the calculation of (4.4) and (4.5) is simplified by approximating the sine and cosine functions by the first three terms of the Taylor's series expansion and $\Delta P/|V|$ is used to check for convergence instead of ΔP , reducing the calculation of (4.4) by $(n-1)$ multiplications as shown in (4.6)

$$\Delta P_i / |V_i| = P_i^{SP} / |V_i| - \sum_{j \in i} |V_j| (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \quad (4.6)$$

$\Delta Q/|V|$ is not used for convergence testing since the reactive power directly depends on voltage magnitude changes, and as might be expected for voltage magnitudes far from unity the obtained solution may not be as accurate as when convergence is checked on ΔQ .

The treatment of reactive constraints at PV buses, voltage constraints at all buses and other common load flow adjustments when using the FDLF is described in detail in Stott and Alsac [69].

4.2.2 - Computational Results

The FDLF algorithm was programmed in PASCAL using the sparse programming techniques described in the previous sections and execution times are given in Table 4.3 for different calculation stages of the FDLF in the load flow solution of several test systems using a TMS9900 microprocessor. The numerical solution obtained for the different test systems with the TMS9900 was identical to that obtained on a CDC Cyber 174 computer up to the fourth decimal place and required the same number of iterations. The execution speed of the TMS9900 as compared with the CDC was slower by an average factor of 60 in the solution of load flow problems.

TABLE 4.3

Load Flow Execution Times on the
TMS9900 Microprocessor (s)

Network	Factorisation of		Iterative	Time per	No. of
	B'	B''	Process	Iteration	Iters.
5	0.075	0.023	0.912	0.608	1.5
9	0.176	0.158	1.896	0.948	2.0
14	0.305	0.119	6.286	1.572	4.0
24	0.736	0.188	7.442	2.977	2.5
30	0.800	0.691	12.469	3.563	3.5
39	1.119	0.889	20.290	4.058	5.0
57	2.625	1.790	30.622	6.805	4.5
118	4.403	0.493	43.396	14.465	3.0
400*	15.182	7.778	237.288	47.458	5.0

*Part of the CEGB network of 465 buses

The British Grid System consisting of 465 buses and 731 lines [70] was used to explore the maximum network size for which a load flow solution could be obtained on a single TMS9900. An attempt was made to solve the complete system but the program ran out of memory during the factorisation of [B'] after having formed [Y] and factorised [B'']. All the buses connected by a single branch were eliminated from the system and a solution was achieved for the reduced network of 361 buses. Then, buses previously eliminated were added to the reduced network until the program execution failed because of memory requirements. The largest network that could be solved on the TMS9900 with 64 Kbytes of memory consisted of 400 buses. However, as memory requirements depend on network characteristics, a rough estimate indicates that it would be possible to study networks of about 300 buses considering 3 branches per bus, without resorting to the use of a backing store for intermediate solution stages.

4.3 - CONTINGENCY ANALYSIS

Contingency analysis is a valuable analytical tool to determine the ability of a power system to withstand the impact of sudden changes due to equipment outage such as the loss of a generator, transmission line, etc. A practical strategy for contingency analysis is based on

the application of an approximate dc load flow to measure the relative severity of a contingency and contingency cases violating predetermined system constraints are then retained to be processed in more detail by an ac contingency power flow technique (see, for example, Enns et al.[71]). However, a major drawback of the dc technique is that it does not account for the effects of deviations in the bus voltage magnitudes even though such deviations have a great effect in the ranking of contingencies and the development of performance indices when the postcontingency system voltage profile is taken into consideration (Medicherla and Rastogi [72]). Peterson et al.[74] formulated a decoupled algorithm based on a linearization of the real and reactive nodal power to incorporate voltage magnitudes in contingency calculations. Enns et al.[71] developed a contingency analysis algorithm based on the compensation method (see Tinney [73]) to measure the system voltage deviation from an ac base-case solution following a contingency.

The compensation method which is reviewed in the Appendix A, provides fast and efficient solutions of problems involving changes in a system from available base-case solutions.

A fast, approximate ac linear contingency analysis algorithm based on the compensation method [73] was implemented to provide a meaningful measure of the relative severity of a given contingency.

A simple logical rule is used to analyse contingency case results. If any system component experiences an out-of-limit condition the user is alerted by displaying the abnormal conditions for the case being examined. The program allows user-specified single next-contingency cases which can be of the following types:

- . Branch outage
- . Capacitor or reactor switching
- . Generator/Load changes

The initial conditions are provided by a base case ac load flow solution and the algorithm tests each contingency for applicability and rejects those contingencies for which:

- . The branch specified is absent from the base case
- . The removal of the specified branch disconnects the network
- . A specified bus is absent from the network
- . The specified bus for a generator outage is the slack bus

In the simulation of a contingency, voltage magnitudes at generator buses are allowed to change in order to determine the impact of the contingency at those buses. In order to avoid iterations in the calculation it is assumed that for any system changes the current injection vector remains constant at its base-case value. Thus, the postcontingency voltages $[V']$ are obtained without altering the network initial conditions by the expression

$$[V'] = [V^0] + [\Delta V] \quad (4.7)$$

where the bus voltage deviation $[\Delta V]$ is a compensation vector that depends on the type of contingency and indicates the severity of the contingency.

Application of the compensation method [73] to calculate $[\Delta V]$ requires the computation of columns of the network impedance matrix $[Z]$ corresponding to the buses involved in the contingency. Designating the i th column of $[Z]$ by $[Z_{.i}]$, the following relation holds [58]:

$$[Y][Z_{.i}] = \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ 1 \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \quad (4.8)$$

Thus, a column of $[Z]$ can be readily obtained from the factorised $[Y]$ matrix by performing a repeat solution using a null r.h.s. vector containing only a +1 in the location corresponding to the required column. Computational savings are achieved in this calculation process by incorporating the method of sparsity exploitation in the forward substitution as described by Stott and Hobson in [75]. Once the required $[Z]$ columns have been obtained they are used with the compensation method to simulate a specified contingency as described below.

4.3.1 - Branch Outage

If a branch connecting buses i and j is to be removed from the network then the voltage compensation vector is computed from

$$[\Delta V] = -(v_i^0 - v_j^0) (Z_{ii} + Z_{jj} - 2Z_{ij} - z_{ij})^{-1} ([Z_{.i}] - [Z_{.j}]) \quad (4.9)$$

If the branch outage is a tap changing transformer then $1/t$ is used instead of $+1$ when calculating the $[Z]$ column for the bus on the tap side of the transformer.

4.3.2 - Shunt Capacitor or Reactor Switching

If a capacitor bank or a shunt reactor is to be connected to or disconnected from bus i then

$$[\Delta V] = -v_i^0 (Z_{ii} \pm z_{sh})^{-1} [Z_{.i}] \quad (4.10)$$

where the negative sign is used for a disconnection.

4.3.3 - Multiple Branch Outages

Simultaneous multiple network-element changes can be handled in a case-by-case basis as described by Alsac et al. in [74]. If m multiple outages are to be calculated then the solution is given by the recursive formula:

$$\begin{aligned} [V^1] &= [V^0] + [\Delta V^1] \\ [V^2] &= [V^1] + [\Delta V^2] \\ &\vdots \\ [V^m] &= [V^{m-1}] + [\Delta V^m] \end{aligned} \quad (4.11)$$

where for each case $[\Delta V^m]$ is given by (4.9) or (4.10).

4.3.4 - Generator/Load Changes

Power changes in a system due to step changes in load, partial or total loss of generation are accommodated either by the slack generator or by all generators according to participation factors based on natural governor response (Stott and Hobson [75]). The participation factor for the generation unit i may be computed from the steady state expression

$$\lambda_i = \frac{P_{gi}^{\text{rated}}}{R_i \left(\sum_j R_j^{-1} P_{gj}^{\text{rated}} + D_{lf} \sum_j P_{dj} \right)} \quad (4.12)$$

which is used to redistribute the scheduled power of generation units due to primary speed control action [76]. R_i is the unit droop given by frequency change over generator rated output change, D_{lf} is the load frequency sensitivity expressed in percent decrease in load per 1% frequency decrease.

When the generation output of a unit has reached the generation limit the participation factor is set to zero. If only the slack generator is used to accommodate any generator/load changes all participation factors are set to zero except for the slack generator which is set to 1.

The different cases of changes in load, partial or total loss of generation are treated in the same way. The system voltage deviation $[\Delta V]$ due to a change in power S at bus i is computed from

$$[Y][\Delta V] = [\Delta I] \quad (4.13)$$

where $[\Delta I]$ is formed as follows

$$\Delta I_k = \begin{cases} 0 & \text{If bus } k \text{ is not participating} \\ -\lambda_k \Delta S_i^* / V_i^* & \text{if bus } k \text{ is participating} \\ \Delta S_i^* / V_i^* & \text{at bus } i \end{cases}$$

4.3.5 - Application and Results

Computation times are given in Table 4.4 for some of the contingencies studied in the test power systems with the TMS9900. The execution times for line flow calculations include the checking of system postcontingency conditions. To simplify computations in the calculation of participation factors a uniform regulating unit droop R_i of 4% and a load frequency sensitivity factor D_{lf} of 1% are assumed for all generation units. From these results it can be seen that most of the computation time is spent in line flow calculations and the checking of system postcontingency conditions.

TABLE 4.4

Execution Times for Contingency Analysis
on the TMS9900 Microprocessor (s)

Network	Branch Outage	Capacitor/Reactor Switching	Generator/Load Outage	Line Flows
5	0.622	0.297	0.271	2.805
9	1.045	0.509	0.483	3.350
14	1.555	0.704	0.628	5.132
24	2.180	1.163	1.155	6.906
30	2.662	1.260	1.276	6.914
39	2.481	1.283	1.191	10.488
57	5.612	2.771	2.717	13.771
118	9.202	4.707	4.974	24.686
400*	29.697	15.375	14.871	111.125

*Part of the CEGB network of 465 buses

4.4 - SOFTWARE PACKAGE ORGANIZATION

The organization and interaction between the main procedures of the software package for steady state analysis are illustrated in the Fig. 4.6. The code and the data files are stored on floppy disk in the disk units of the FS990/4 host minicomputer and only the module required for execution is loaded onto the microprocessor's memory. The module for short circuit calculations is described in chapter 6.

The PASCAL programs implemented can be executed without any changes on the CDC Cyber 174 computer, and on the TI 990/4 minicomputer, and by replacing the I/O procedures on the TMS9900 microprocessor.

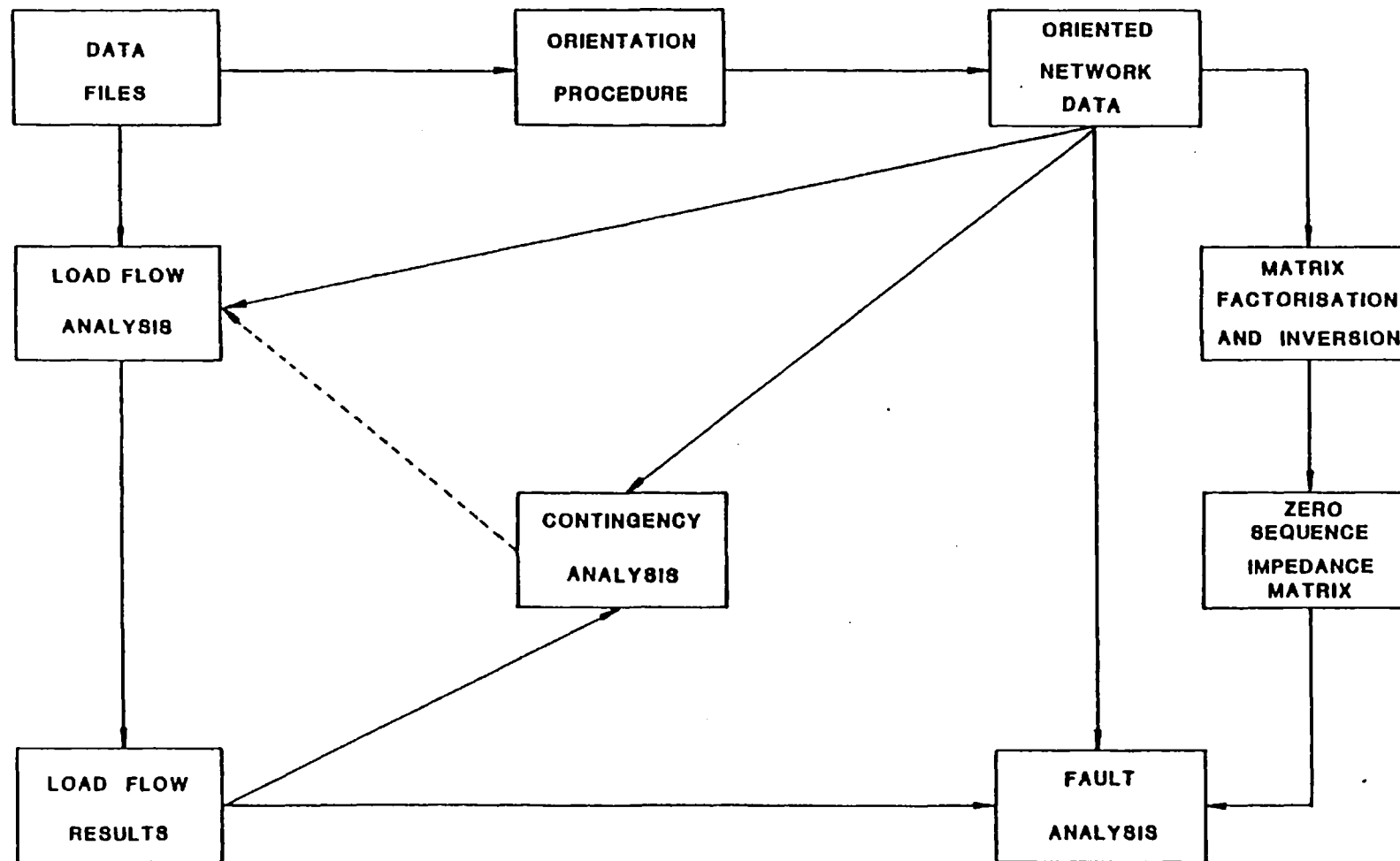


Fig. 4.6: Software Package Organization for Power System Steady State Analysis

4.5 - CONCLUSIONS

The potential of a single microprocessor in the solution of power system problems has been explored with the application of a software package implemented in PASCAL for steady state analysis.

With the developed programs it is possible to perform load flow and contingency analysis studies for systems with up to 400 buses on a single microprocessor unit. Although it was possible to analyse large power systems on a microprocessor, accuracy problems may occur even when small systems with high and low impedance, capacitive compensation, etc., are analysed on processing units which use a 32-bit floating point representation.

The computation time of an MPU when compared with mainframe computers is rather slow. The execution speed of the TMS9900 microprocessor was slower by an average factor of 60 with respect to the execution time obtained on a CDC 174 Cyber computer using the same PASCAL program for the solution of load flow problems. However, microprocessors are improving their capabilities constantly and are becoming even more inexpensive. Already there exist in the market microprocessors with greater processing throughput than the TMS9900. For example, the TMS99000, Motorola 68000, Intel 8086, Zilog Z8000, etc.

Dynamic data structures in PASCAL are fully exploited for the implementation of the sparsity-oriented techniques required for the efficient handling of large networks.

Since the software package for steady state analysis is written in a high level language, it could be used with other processor systems supporting the PASCAL language without any modifications at all.

In addition to the applications presented, the developed software package may be used as a building block for other algorithms such as automatic contingency selection, dynamic contingency analysis, network equivalents and in other areas where the solution of systems of symmetric linear algebraic equations is required.

CHAPTER FIVE

MULTIPLE MICROPROCESSORS FOR THE SOLUTION OF LARGE POWER NETWORKS

5.0 - SUMMARY

The decomposition of large networks enables solutions to be calculated concurrently to increase processing power or serially to extend storage capability. This chapter describes the application of piecewise methods for the solution of large electric power networks using multiple microprocessors operating in parallel.

A heuristic cluster algorithm is employed to select an optimum decomposition pattern for a given network and an efficient piecewise technique is proposed to divide the network into subnetworks. The technique retains the same properties as the untorn approach and it is suitable for the application of short word length microcomputers. The solution of the torn network is then carried out in parallel by solving each subnetwork on one microprocessor.

The implementation of computational algorithms that fully exploit parallelism and sparsity at all levels in the solution process is examined. A parallel processing algorithm based on the fast decoupled load flow method is outlined and the results are presented for the load flow solution of IEEE standard test systems using an array of microprocessors.

5.1 - THE MULTIPLE MICROPROCESSOR CONFIGURATION ADOPTED

Some of the most important aspects of the construction of multiple microprocessor configurations are the interconnection schemes and the procedures for exchanging information between the processors. It was pointed out in sec. 3.5 that the interconnection between MPUs with the available hardware would be through the parallel I/O ports of the MPUs using a 20-way ribbon cable. This scheme restricts the microprocessor configuration to architectures requiring few interconnections, since the implementation of a fully interconnected structure would be more

expensive than using a parallel data bus to interconnect all the processors.

The chosen parallel structure for the solution of power networks is a "star" configuration for BBDF algorithms (see sec. 1.1). A configuration of processors for BBDF algorithms requires that the processor assigned to the block of cut-set variables be capable of sharing data with the other processors. The remaining processors do not have to share data between each other and therefore no interconnections between them are required. A multiple microprocessor configuration for BBDF algorithms using available hardware was illustrated in Fig. 3.4.

Since each MPU has a local memory which only its own processor can access, the solution of a network using the chosen microprocessor structure requires the application of piecewise techniques to partition the network into subnetworks such that each subnetwork can be analyzed independently. To achieve computational efficiency, the subnetworks should be chosen roughly of the same size with as few interconnections as possible in order to balance the tasks assigned to the different MPUs.

5.2 - DECOMPOSITION OF NETWORKS

Appropriate structures for the decomposition of large power networks may be determined by inspection of the network geographical layout or by the application of tearing algorithms. Various decomposition algorithms that exploit the topology of an interconnected network to produce near optimum decompositions are discussed in references [5] and [13]. These tearing algorithms transform the network adjacency matrix into special matrix forms which are suitable for decomposition. The adjacency matrix is an $n \times n$ matrix whose ij -th entry is nonzero for $i=j$ and $i \neq j$ if buses i and j are connected by a common branch.

The heuristic cluster algorithm described by Sangiovanni-Vincentelli et al. in [77] is used in this work to determine the most suitable decomposition of a network into a specified number of subnetworks. The number of subnetworks is directly related to the number of microprocessors that will be used for solution. This clustering algorithm permutes the adjacency matrix of the network into a BBDF matrix such that the diagonal blocks have nearly equal size and the number of elements in the border is minimized. The number of

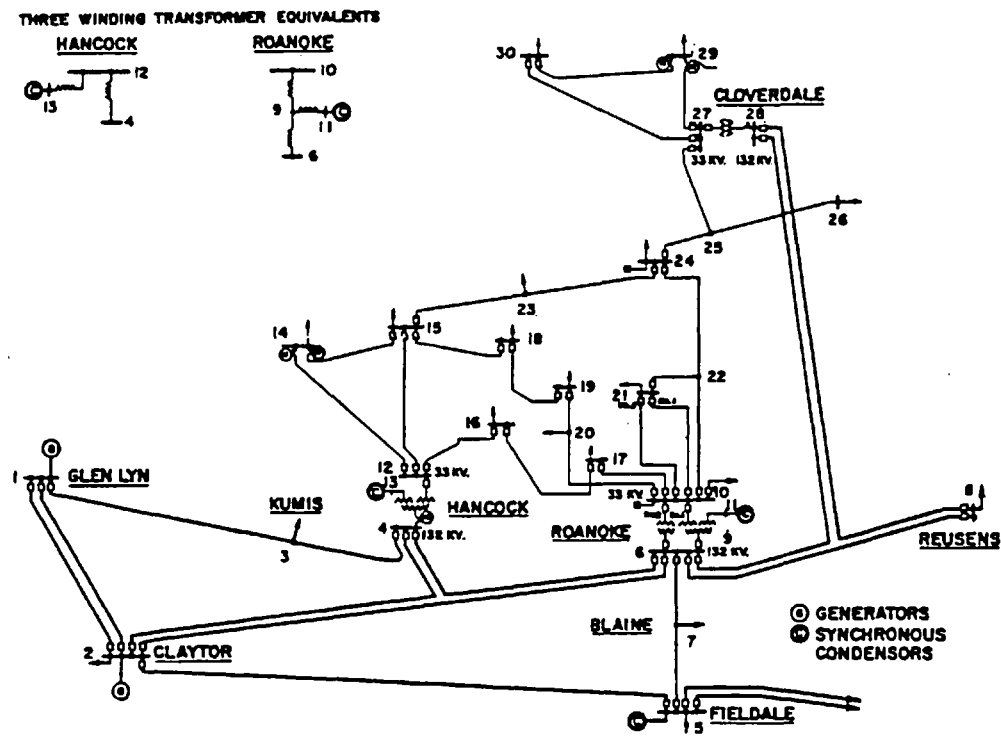
blocks formed in the diagonal may be controlled by specifying the size of the largest block to be allowed. The algorithm is based on the concept of a contour tableau [63]. The tableau construction is started by selecting a bus of minimum degree (fewest number of branches). The sequence of buses which are subtended by the last rise and fall in the sequence of indices of the contour form a cluster [63], see Fig. 5.1a.

To illustrate how the algorithm is used to identify clusters in a network, consider the application to the IEEE 30-bus standard test network depicted in Fig. 5.1. Assume that 3 MPUs will be used for solution, then the size of the largest subnetwork to be allowed is 10 buses. In this network, the buses 11, 13 and 26 have the fewest number of branches, therefore one of them is selected by the algorithm as the starting point for the tableau construction. If the tableau formed is not found to be suitable for decomposition the algorithm can be rerun selecting a different starting bus. The tableau which was more suitable for decomposition in this example was found when the algorithm started at bus 13. The resulting contour tableau is shown in Fig. 5.1a and the clusters identified are marked by the dotted lines in Fig. 5.1b which depicts the graph associated with the network. The buses outside the clusters 4, 10, 24 and 27 are identified as the bordered buses. By reordering the adjacency matrix so that the rows and columns corresponding to the bordered buses are concentrated in the bottom rows and the right-most columns of the matrix, the original adjacency matrix is transformed into a BBDF as shown in Fig. 5.1c. The algorithm was programmed in PASCAL using DDS to manipulate the network adjacency matrix.

5.3 - PIECEWISE TECHNIQUES

After a partitioning pattern has been chosen to decompose a network, an appropriate formulation is required for the equations representing the torn network or diakoptic equations.

In the formulation of the diakoptic equations some authors have followed Kron's original development [78], considering diakoptics as a physical network tearing approach (Happ [79]), while others have emphasized that diakoptics is an algebraic process (for example, Laughton [81] and Sasson [82]) or have derived the diakoptic equations by partitioning and rearranging Kirchhoff's law equations (Brameller [80], Wu[84], Chua and Chen [85]). Although these derivations are quite different from each other, it is recognized that the theory of



IEEE 30-bus standard test system

	Iteratins Node	A d j a c e n t S e t					Cardinality
1	13	12					1
2	12	4	14	15	16		4
3	14	4	15	16			3
4	16	4	15	17			3
5	17	4	15	10			3
6	15	4	10	18	23		4
7	18	4	10	23	19		4
8	23	4	10	19	24		4
9	19	4	10	24	20		4
10	20	4	10	24			3
11	3	1					1
12	1	2					1
13	2	5	6				2
14	5	6	7				2
15	7	6					1
16	6	8	9	28			3
17	8	9	28				2
18	9	28	11				2
19	11	28					1
20	28	27					1
21	21	22					1
22	22	0					0
23	25	26					1
24	26	0					0
25	29	30					1
26	30	0					0

First cluster [13,12,14,16,17,15,18,23,19,20]

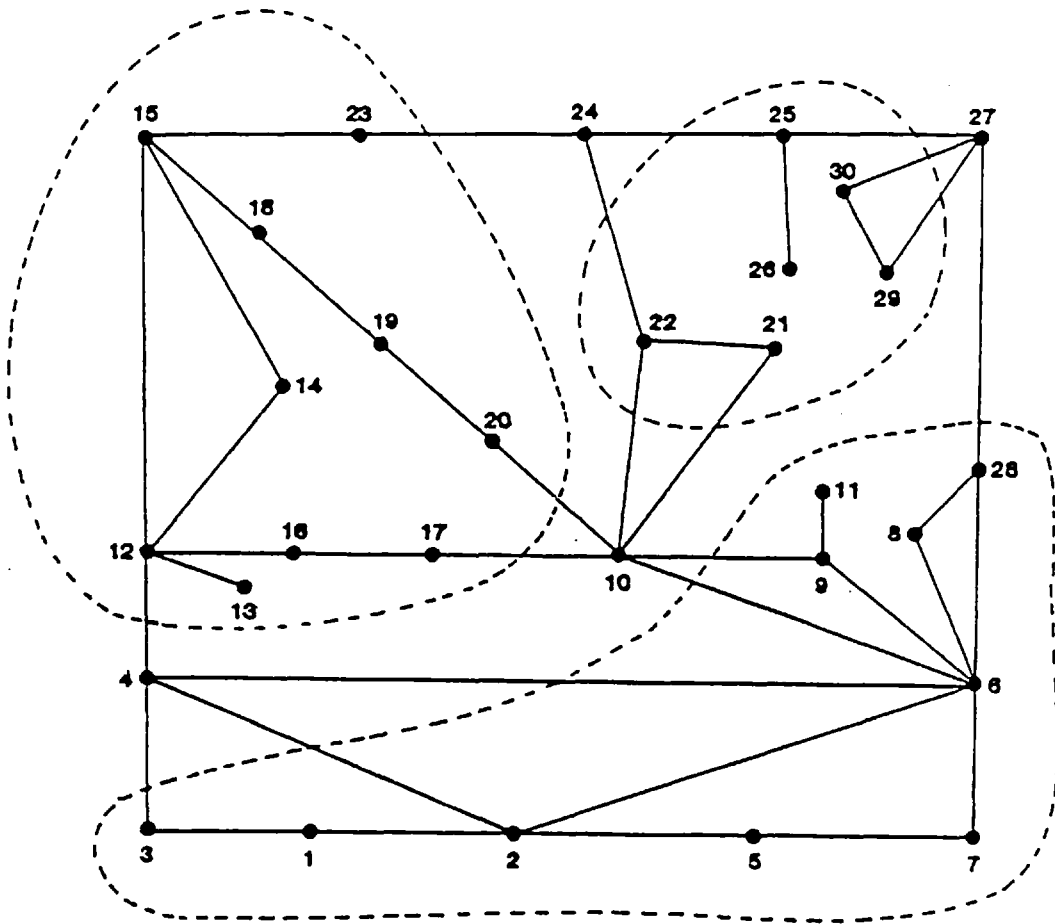
Bordered buses [4,24,10]

Second cluster [3,1,2,5,7,6,8,9,11,28]

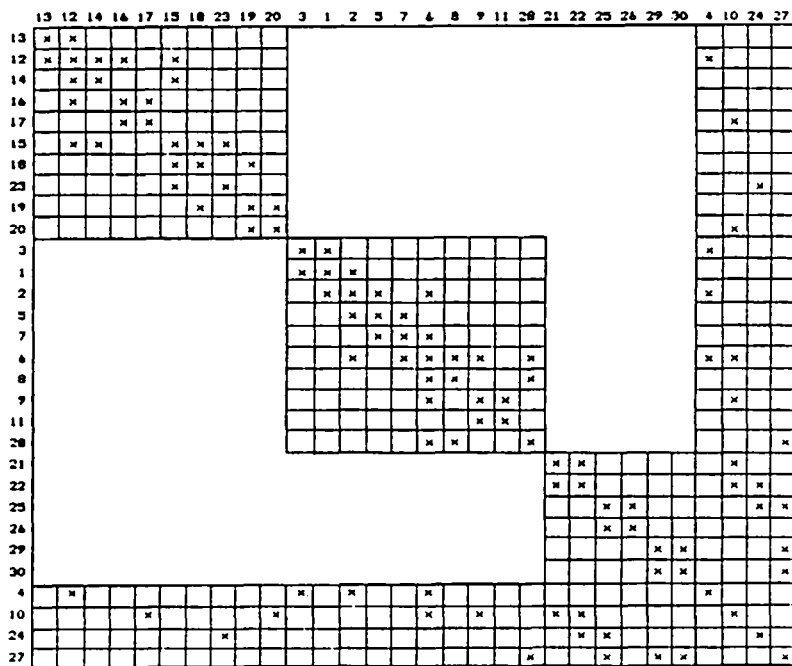
Bordered buses [27]

Third cluster [21,22,25,26,29,30]

(a) Contour tableau



(b) Clusters in the network



(c) Transformation of the adjacency matrix into BBDF

Fig. 5.1: Decomposition Pattern for the IEEE 30-bus System.

diakoptics is the basis for all existing decomposition techniques, since the essence of this theory is to solve a large network by first tearing it apart, solving each subdivision separately and then combining the subdivision solutions to obtain the solution of the entire network [78].

When, in conventional diakoptics, a network is partitioned into subdivisions and the torn branches are removed from the matrix representing each of the subsystems, singularity problems may occur, especially if $[Y_{bus}]$ methods are employed for the solution of low voltage networks [81] or in methods that exclude all shunt elements, as in the case of the matrix $[B']$ of the FDLF. This problem has been overcome by:

- . Selecting a temporary bus for reference in each of the subdivisions, except in the subdivision containing the slack bus of the system [80],[91].
- . Cutting twice the interconnecting branches (Overlapping decomposition technique). The torn branches are not removed and remain part of the areas they were interconnecting. All buses on the far side of interconnection branches appear as slack buses [83],[90].

In the first approach an exact formulation of the effects of tearing is represented by adding the cut injected currents in the subsystems. A policy for current direction or sign assignment of the cut injected currents is required and the temporary reference buses have to be adjusted during network solutions. Buses connected with torn branches only should be treated as a special case [84]. With the second approach the requirements in the first are not necessary but provision has to be made during the solution process to properly account for the effect of the torn branches for an exact representation [90].

An efficient piecewise technique that combines the best computational features of the approaches described above is derived in the following section by performing mathematical manipulations of the network equations.

5.3.1 - The Proposed Technique

The derivation procedure of the proposed technique is illustrated

by considering the simple power network shown in Fig. 5.2a. For simplicity, it is assumed that there is no mutual coupling either between torn branches and subnetworks or among the torn subnetworks.

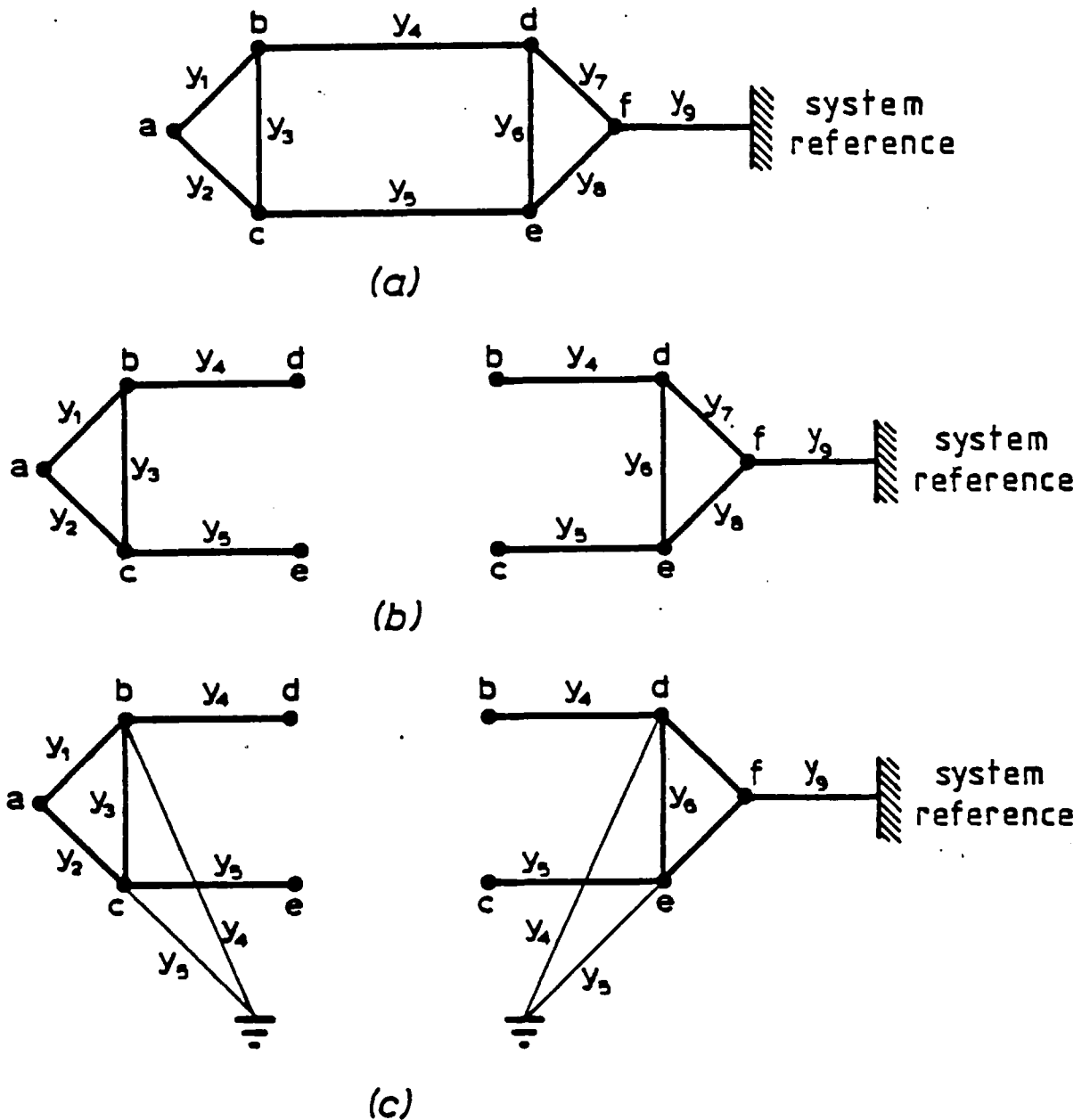


Fig. 5.2: Decomposition of an Electric Power Network.

(a). Original network. (b). The network is torn using the overlapping decomposition technique. (c). Boundary buses of the subnetworks are grounded.

1. The admittance matrix $[Y]$ for the network of Fig. 5.2a. is given by

	a	b	c	d	e	f
a	y_1+y_2	$-y_1$	$-y_2$			
b	$-y_1$	$y_1+y_3+y_4$	$-y_3$	$-y_4$		
c	$-y_2$	$-y_3$	$y_2+y_3+y_5$		$-y_5$	
d		$-y_4$		$y_4+y_6+y_7$	$-y_6$	$-y_7$
e			$-y_5$	$-y_6$	$y_5+y_6+y_8$	$-y_8$
f				$-y_7$	$-y_8$	$y_7+y_8+y_9$

(5.1)

Note that this is not a singular matrix.

2. The network is partitioned into two subnetworks by cutting twice the interconnection lines b-d and c-e as shown in Fig. 5.2b. This is equivalent to removing the off-diagonal-block elements in (5.1).

	a	b	c	d	e	f
a	Y_{aa}	Y_{ab}	Y_{ac}			
b	Y_{ba}	Y_{bb}	Y_{bc}			
c	Y_{ca}	Y_{cb}	Y_{cc}			
d				Y_{dd}	Y_{de}	Y_{df}
e				Y_{ed}	Y_{ee}	Y_{ef}
f				Y_{fd}	Y_{fe}	Y_{ff}

+

	a	b	c	d	e	f
a						
b				$-y_4$		
c					$-y_5$	
d		$-y_4$				
e			$-y_5$			
f						

(5.2)

The submatrices of the diagonal blocks are not singular matrices.

3. The cut admittance matrix on the right hand side of (5.2) is transformed into a matrix similar to that represented in conventional diakoptics. The buses with torn branches (boundary buses) are grounded through an admittance equal to the series admittances of the torn branches, as illustrated in Fig. 5.2c.

and represented in (5.3).

$$\begin{array}{c}
 \begin{array}{c} a \\ b \\ c \\ d \\ e \\ f \end{array}
 \begin{array}{|c|c|c|c|c|c|}
 \hline
 a & b & c & d & e & f \\
 \hline
 Y_{aa} & Y_{ab} & Y_{ac} & & & \\
 Y_{ba} & Y_{bb}+y_4 & Y_{bc} & & & \\
 Y_{ca} & Y_{cb} & Y_{cc}+y_5 & & & \\
 \hline
 & & & Y_{dd}+y_4 & Y_{de} & Y_{df} \\
 & & & Y_{ed} & Y_{ee}+y_5 & Y_{ef} \\
 & & & Y_{fd} & Y_{fe} & Y_{ff} \\
 \hline
 \end{array}
 +
 \begin{array}{c} a \\ b \\ c \\ d \\ e \\ f \end{array}
 \begin{array}{|c|c|c|c|c|c|}
 \hline
 a & b & c & d & e & f \\
 \hline
 & & & & & \\
 & -y_4 & & -y_4 & & \\
 & & -y_5 & & -y_5 & \\
 & -y_4 & & -y_4 & & \\
 & & -y_5 & & -y_5 & \\
 & & & & & \\
 \hline
 \end{array}
 \end{array} \quad (5.3)$$

Expressing (5.3) in a condensed form

$$[Y] = [Y_{bdf}] + [Y_{tb}] \quad (5.4)$$

It should be noted that matrix $[Y_{bdf}]$ has a block-diagonal-form (BDF) and that matrix $[Y_{tb}]$ contains torn branches only.

4. The cut admittance matrix $[Y_{tb}]$ is decomposed into the product of three matrices

$$\begin{array}{c}
 \begin{array}{c} 4 \ 5 \\ a \\ b \\ c \\ d \\ e \\ f \end{array}
 \begin{array}{|c|c|}
 \hline
 & \\
 \hline
 1 & \\
 & 1 \\
 1 & \\
 & 1 \\
 & \\
 \hline
 \end{array}
 \begin{array}{c} 4 \ 5 \\ 4 \\ 5 \end{array}
 \begin{array}{|c|c|}
 \hline
 -y_4 & \\
 & -y_5 \\
 \hline
 \end{array}
 \begin{array}{c} 4 \ 5 \\ a \ b \ c \ d \ e \ f \\ 4 \\ 5 \end{array}
 \begin{array}{|c|c|c|c|c|c|}
 \hline
 & 1 & & 1 & & \\
 & & 1 & & 1 & \\
 \hline
 \end{array}
 \end{array} \quad (5.5)$$

or in short form

$$[Y_{tb}] = [C] [-y_t] [C]^t \quad (5.6)$$

where the superscript t denotes the transpose. $[-y_t]$ is a diagonal matrix containing the negative of the series admittance of the torn branches and $[C]$ is a connectivity matrix which is easily formed by inspection. The number of rows is equal to the

the number of buses and the number of columns is equal to the number of torn branches. Each column of $[C]$ contains only nonzero entries of +1 in the rows corresponding to the buses connected by each torn branch (boundary buses).

5. The equations representing the torn network are obtained by substituting (5.6) in (5.4), replacing (5.4) in the network equation (2.1) and rearranging

$$\begin{bmatrix} Y_{bdf} & C \\ C^t & z_t \end{bmatrix} \begin{bmatrix} V \\ I_t \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} \quad (5.7)$$

where $[z_t] = [y_t]^{-1}$ is the impedance matrix of the torn branches and $[I_t] = [-y_t][C]^t[V]$ is a vector containing the set of compensation currents due to the tearing.

The elimination of -1 's from $[C]$ in the derived technique destroys the physical concept of currents through the torn branches but the computational advantages gained are evident since there is no need of extra information relating to the classification of subsystems according to the convention adopted for the current flow direction in the cutset.

The exact representation of the tearing gives to the derived decomposition technique the same properties as the untorn approach. Equation (5.7) has the same structure as the conventional diakoptic equations (see, for example, Brameller [80] or Wu [84]) but differs in that:

- . Torn branches are not removed and appear in each of the subsystems they were interconnecting.
- . Temporary reference buses in each subdivision are not necessary.
- . The boundary buses are grounded through the series admittances of the torn branches.
- . The current source modelling each torn branch does not represent the current through the branch.
- . The connectivity matrix $[C]$ does not contain -1 's.

- . The torn branch impedance matrix $[z_t]$ appears with positive sign.

These differences give to the proposed decomposition technique the following computational characteristics:

- . No extra calculations for temporary bus adjustments are required.
- . Buses connected with torn branches only are treated as normal buses.
- . No extra information is needed relating to the convention adopted for the current flow direction in the torn branches.
- . The computational effort at the subsystem level depends on the number of boundary buses rather than on the number of torn branches.

The numerical calculation of the decomposition technique described above can be reinforced by adding a fixed admittance to the $[Y_{bdf}]$ diagonal elements to assure a matrix diagonal dominance. The admittance is obtained by converting the loads into admittances using an estimate of the bus voltages [79]. If the estimated voltage is chosen as 1.0 p.u. the load admittance is equal to the complex conjugate of the load power

$$Y_{li} = P_{li} - jQ_{li} \quad (5.8)$$

Then, the bus injection currents should be corrected using an equivalent value as

$$I'_i = I_i + (P_{li} - jQ_{li})V_i \quad (5.9)$$

When Point Jacobi algorithms or Gauss-Seidel iterative procedures are employed for network solutions the convergence is improved if the approach of converting the loads into fixed admittances is included.

Although the decomposition technique was derived by assuming a symmetric system of equations, it may be applied to any set of linear simultaneous equations, whether symmetric or not.

5.4 - SOLUTION ALGORITHMS.

It has been shown that all recently proposed sparse matrix solution algorithms for diakoptic analysis of power systems belong to a set of algorithms which is derived by applying block matrix elimination to a partitioned form of system equations (Hajj [89]).

Thus, applying block matrix elimination algorithms (George [86]) to (5.7) for solution, the following expression is obtained

$$\begin{bmatrix} Y_{bdf} & C \\ C^t & z_t \end{bmatrix} = \begin{bmatrix} L & \\ C^t U^{-1} & L_t \end{bmatrix} \begin{bmatrix} U & A \\ & U_t \end{bmatrix} \quad (5.10)$$

$$\text{where } [A] = [L]^{-1}[C] \quad (5.11)$$

$$[L_t][U_t] = [Z_{tt}] = [z_t] - [C]^t [U]^{-1} [A] \quad (5.12)$$

[L] and [U] are lower and upper triangular matrices respectively containing the factors of $[Y_{bdf}]$. The size of matrix [A] depends on the number of torn branches and when two or more torn branches are incident with the same bus the corresponding columns in [A] are equal to each other. In this case the size of [A] is reduced by eliminating the redundant columns at the subsystem level. The matrix $[Z_{tt}]$ is referred to as the interarea network impedance matrix [88].

Equation (5.10) represents the factorisation of (5.7) and corresponds to the factorisation scheme denoted by F1 in [86]. Other factorisation schemes may be obtained by rearranging (5.10).

With the factors of (5.10) available for solution different variations in the substitution procedure can be obtained by carefully selecting an appropriate sequence in the order of computations to obtain an efficient solution.

Wu [87], Alvarado [88] and Hajj [89] have recently studied and proposed substitution procedures for the solution of diakoptic analysis

in power systems. The procedures denoted here by S1, S2 and S3 respectively are summarised in Table 5.1. In the formulation of the algorithms the identities of individual subsystems are preserved so that they can be implemented in a parallel computer configuration. To achieve maximum computational efficiency with these algorithms, the system sparsity should be exploited at all levels of the decomposition in the solution process [88],[89]. Due to the composition of the matrix $[C]$ (0 and 1), products indicated with this matrix operate as selectors only and require no numerical operations.

TABLE 5.1

Substitution Procedures

S1	S2	S3
$[L][V_1] = [I]$	$[L][U][V_1] = [I]$	$[L][V_1] = [I]$
$[V_t] = ([C]^t[U]^{-1})[V_1]$	$[V_t] = [C]^t[V_1]$	$[V_t] = [C]^t[U]^{-1}[V_1]$
$[L_t][U_t][I_t] = [V_t]$	$[L_t][U_t][I_t] = [V_t]$	$[L_t][U_t][I_t] = [V_t]$
$[V_2] = [V_1] - [L]^{-1}[C][I_t]$	$[L][U][V_2] = [C][I_t]$	$[V_2] = [A][I_t]$
$[U][V] = [V_2]$	$[V] = [V_1] + [V_2]$	$[U][V] = [V_1] + [V_2]$

The substitution procedures given in the Table 5.1 are equivalent to the Sherman-Morrison-Woodbury formula [5]

$$[V] = [Y_{bdf}]^{-1}[I] + [Y_{bdf}]^{-1}([C](z_t - [C]^t[Y_{bdf}]^{-1}[C])^{-1}[C]^t)[Y_{bdf}]^{-1}[I] \quad (5.13)$$

The procedures S1, S2 and S3 not only differ in the computation sequence but in the ordering strategy used for elimination. In S1 all buses are ordered to minimize fill-in. In S2 the boundary buses are ordered last and the remaining buses are ordered to partially minimize

fill-in. The ordering strategy used with S3 orders the buses in the subnetworks to reduce fill-in delaying as long as possible the choice of the boundary buses as pivots but without necessarily ordering them last.

Hajj [89] demonstrated that S3 performs better than S1 and S2, and that it produces the most economical solutions when the auxiliary matrix [A] is formed and stored for computations. Hajj also suggested that in order to obtain overall computational efficiency, different algorithms could be used to solve different subsystems within the same system.

The performance of these algorithms is evaluated in sec. 5.5.3 using multiple microprocessors and the application of different algorithms to solve different subsystems is investigated.

5.5 - PARALLEL SOLUTION ALGORITHMS USING MULTIPLE MICROPROCESSORS.

In the implementation of parallel solution algorithms for the multiple microprocessor structure illustrated in Fig. 3.4, the central microprocessor of the star configuration should be used to synchronise all the microprocessors during execution (see sec. 1.1.2). In addition, the subnetwork containing the network bordered buses and the cutset equations should be assigned to this processor for solution.

5.5.1 - Factorisation Procedure

After distributing the tasks among the microprocessors, the factorisation in parallel of (5.7) may be carried out using the algorithm outlined in Fig. 5.3. In this algorithm the MPUs are synchronised during execution through the ready-to-receive signal issued at step 4c by the central MPU. The MPUs in the periphery wait until this signal is received to send a message to the central MPU and to continue execution. The central MPU will send a ready-to-receive signal only to those MPUs from which it is awaiting a new message. The number of messages and the length of each message to be sent by an MPU in the periphery is equivalent to the number of boundary buses in the subnetwork. All information regarding to how many and which length of messages are to be received in the central MPU is available from the cutset equations.

From steps 1-3 of the algorithm in Fig. 5.3, note that it is important to have nearly equal size subnetworks, since this would

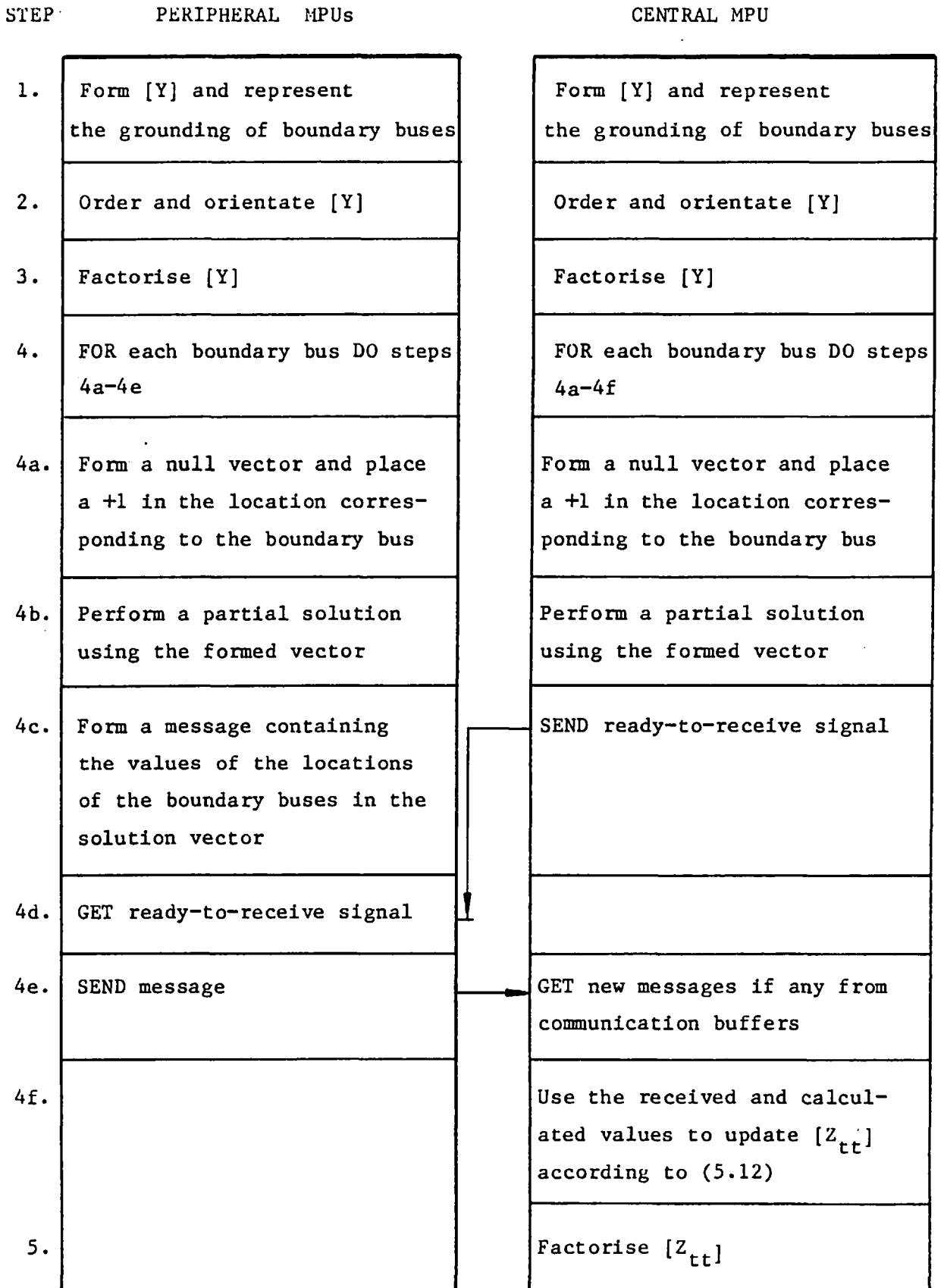


Fig. 5.3: Algorithm for the Factorisation in Parallel of a BBDF Matrix Using Multiple Microprocessors.

balance the tasks among the microprocessors. And from step 4 it should be noticed that the computational effort at the subsystem level depends on the number of boundary buses. The MPU solving the subnetwork with the fewest number of boundary buses will be first to finish the computation of its contribution to the formation of the interarea matrix. After the interarea matrix has been formed, the central MPU proceeds to factorise it.

5.5.2 - Solution Procedure

After the factorisation, a repeat solution can be performed on the multiple MPUs configuration using the parallel algorithm described in Fig. 5.4. This algorithm can be used with the substitution procedures S1, S2 and S3. The synchronization of MPUs in this algorithm is provided automatically by the scheme used for exchanging information. The central MPU will not proceed to solve the cutset equations until it has received all the information required from the peripheral MPUs and all MPUs in the periphery will wait for the cutset solution to modify their individual solutions in order to obtain an exact solution. Hence the justification for not dedicating a single MPU for the cutset solution, since this will not result in an improvement of the overall parallel configuration performance. The importance of having a small cutset is evident from the algorithm since all the peripheral MPUs are idle during the cutset solution.

A numerical example using a simple power network is given in the Appendix B to illustrate the implementation of the parallel processing algorithms described.

5.5.3 - Investigation and Results on the Application of Parallel Solution Algorithms.

The preceeding computational algorithms have been programmed in PASCAL using the sparsity techniques presented in sec. 4.1 and the assembly language procedures described in sec. 3.5.2 for the communication between microprocessors.

The standard IEEE 30, 57 and 118-bus systems were chosen as test systems to compare the performance of the solution algorithms presented in Table 5.1. Some characteristics of the network partitionings obtained with the algorithm described in sec. 5.2 are given in Table 5.2.

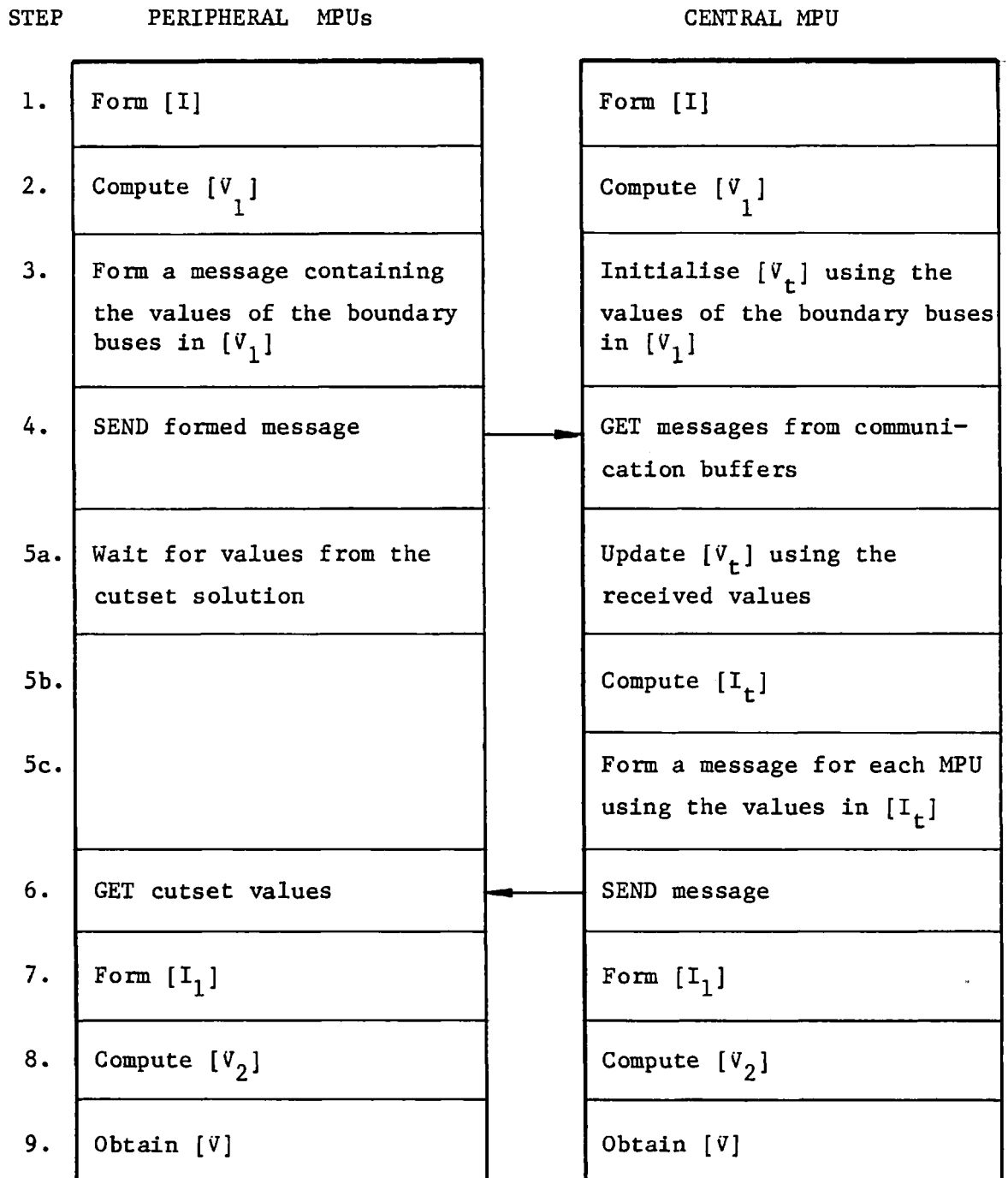


Fig. 5.4: Parallel Solution Algorithm Using Multiple Microprocessors

TABLE 5.2

Network Partitioning Characteristics

System	Subnetworks	Buses in Subnetworks	Torn Branches	Boundary Buses	Cut Buses
30-bus	2	16,14	4	4,3	3,4
	3	10,10,10	10	5,4,4	3,3,9
	4	7,8,8,7	14	4,4,4,6	3,4,3,12
57-bus	2	29,28	6	5,6	6,5
	3	20,20,17	15	6,8,8	4,7,14
	4	15,15,15,12	19	5,8,4,11	4,6,5,17
118-bus	2	59,59	9	6,8	8,6
	3	35,40,43	9	4,4,7	4,3,
	4	35,31,31,21	17	4,7,3,10	4,5,4,14

The preparation of data for the determined subnetworks is carried out systematically by a program that selects the data for each specified subnetwork from the main data file of the network. The program also includes in the created data file the cutset information containing the torn branches and the boundary buses.

In the evaluation of the parallel solution algorithms S1, S2 and S3 a linear network representation $[Y][V] = [I]$ of real quantities was assumed. Since the algorithms differ only in the way the subsystems are solved, for clarity, only the computational results obtained in the solution of the test systems using 2 MPUs are presented. Execution times corresponding to the processor that finishes the calculation last are given in Table 5.3. The factorisation times include the factorisation of the interarea matrix. The time required by an MPU to transfer 1 to 10 real values varied from 0.9 ms to 1.1 ms.

From the execution times given in Table 5.3 it is observed that the formation of the iterarea matrix is the most time consuming calculation in the solution process of a partitioned system.

TABLE 5.3

Execution Times for Parallel Solution
Algorithms Using 2 MPUs (s)

Algorithm		T e s t 30	S y s t e m s 57	118
S1	Factorisation	0.466	0.769	1.957
	Formation of			
	Interarea Matrix	0.929	1.459	3.540
	Direct Solution	0.559	1.067	2.005
	Total	1.954	3.295	6.274
S2	Factorisation	0.466	1.184	2.099
	Formation of			
	Interarea Matrix	0.512	1.915	0.834
	Direct Solution	0.559	1.153	1.937
	Total	1.537	4.252	4.870
S3	Factorisation	0.465	0.769	1.921
	Formation of			
	Interarea Matrix	0.539	1.441	3.106
	Direct Solution	0.423	0.885	1.461
	Total	1.427	3.095	6.488
S4	Factorisation	0.465	0.769	1.921
	Formation of			
	Interarea Matrix	0.323	0.993	1.064
	Direct Solution	0.388	0.819	1.332
	Total	1.176	2.581	4.317
Solution Factorisation		0.800	2.625	4.403
Without Direct Solution		0.409	0.939	1.871
Tearing Total		1.209	3.564	6.274

The interarea matrix is formed faster with algorithm S2 for the 30 and 118-bus systems. The algorithms S1 and S3 require more or less the same time for factorisation but S3 performs better in the direct solution. The factorisation in parallel is faster than the sequential factorisation without tearing. If the total calculation times for each algorithm are considered, then S2 produces the fastest solution for the 118-bus system and S3 for the 57-bus system.

The algorithm S4, which produces the fastest solution in all the cases presented, is the result of an investigation carried out on the combination of different algorithms to solve different subnetworks. It was found that the most economical solution is achieved when the subnetworks are solved using a variation in the implementation of S3 that further parallelises its performance. In this variation, the peripheral MPUs perform the product of the $[L]$ diagonal elements in the forward substitution while the central MPU is solving the cutset. And the product $[A]^t[v_1^-]$ is performed instead of the backward substitution $[U]^{-1}[v_1]$, as shown in Table 5.4.

TABLE 5.4

An Extension that Further Parallelises the
Performance of Block Matrix Algorithms.

Peripheral MPUs	Central MPU
$[L^-][v_1^-] = [I]$	$[L][v_1] = [I]$
$[v_t] = [A]^t[v_1^-]$	$[v_t] = [C]^t([U]^{-1}[v_1])$
$[D][v_1] = [v_1^-]$	$[L_t][U_t][I_t] = [v_t]$
$[v_2] = [A][I_t]$	$[v_2] = [A][I_t]$
$[U][v] = [v_1] + [v_2]$	$[U][v] = [v_1] + [v_2]$

where $[L] = [L^-][D]$ and $[U] = [L]^t$

Since the formation of the interarea matrix requires the calculation of elements of the inverse matrix corresponding to boundary buses, a fast procedure is incorporated to compute these elements. Instead of performing partial repeat solutions as suggested in previously proposed algorithms, the required elements are obtained using the recursive scheme in Takahashi et al.[92] which is given by (5.14) and (5.15).

$$Z_{ij} = \sum_{k>j} Z_{ik} L_{kj} \quad i < j \quad (5.14)$$

$$Z_{jj} = L_{jj} + \sum_{k>j} Z_{jk} L_{kj} \quad (5.15)$$

Where Z_{ij} and L_{kj} are elements of $[Z]$ and $[L]$ respectively. The advantage of this scheme is that it is restricted to the backward substitution.

Execution times for the solution of the test networks with the algorithm of Table 5.4 using 2 3 and 4 MPUs are depicted in Fig. 5.5, where the execution for the sequential solution has been chosen as unity. From this figure it is seen that larger networks benefit more from parallel diakoptics than smaller networks. The parallel solution of the 30-bus and 57-bus systems is faster than the uniprocessor solution when 2 MPUs are used for the computation but not for the cases when 3 and 4 MPUs are employed.

5.6 - APPLICATION TO THE FAST DECOUPLED LOAD FLOW

The solution of load flow problems with the FDLF [69] using multiple microprocessors requires the formulation of diakoptic equations for (4.2) and (4.3), since the FDLF does not work directly on the matrix $[Y]$ for which the diakoptic equations were derived in section 5.3.1. Application of the decomposition procedure described in sec. 5.3.1 to (4.2) and (4.3) leads to

$$\begin{array}{|c|c|c|} \hline B'_{bdf} & C' & \Delta \theta \\ \hline C'^t & x'_t & I'_t \\ \hline \end{array} = \begin{array}{|c|} \hline \Delta P / |V| \\ \hline 0 \\ \hline \end{array} \quad (5.16)$$

and

$$\begin{array}{|c|c|c|} \hline B''_{bdf} & C'' & \Delta |V| \\ \hline C''^t & x''_t & I''_t \\ \hline \end{array} = \begin{array}{|c|} \hline \Delta Q / |V| \\ \hline 0 \\ \hline \end{array} \quad (5.17)$$

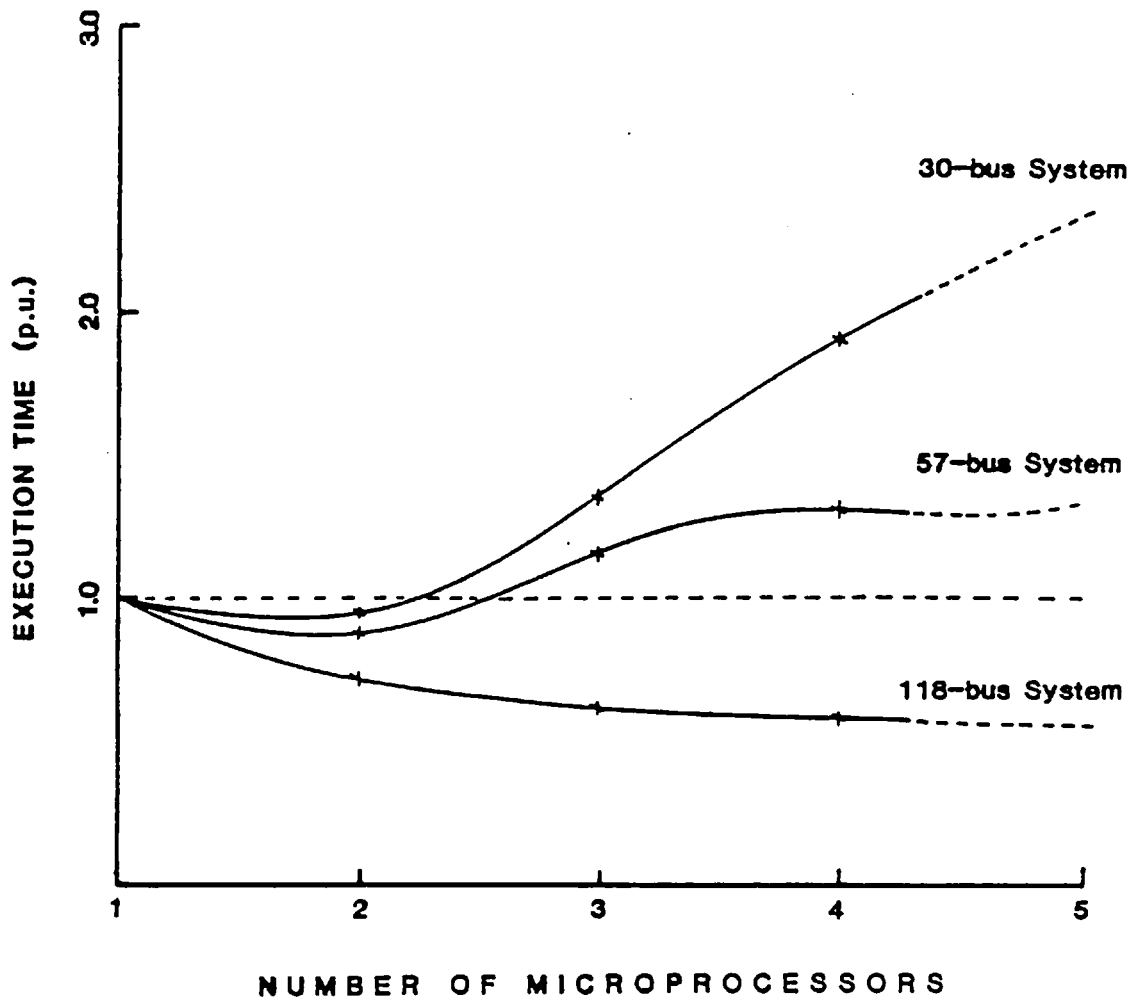


Fig. 5.5: Solution of Test Networks Using up to 4 MPUs

Equations (5.16) and (5.17) have exactly the same structure as (5.7) and, therefore, the factorisation and substitution procedures described in sec. 5.5. can be applied for solution. Thus the intersubdivision equations are expressed as

$$[X_{tt}'] = [x_t'] - [C']^t [U']^{-1} [L']^{-1} [C'] \quad (5.18)$$

$$[X_{tt}''] = [x_t''] - [C'']^t [U'']^{-1} [L'']^{-1} [C''] \quad (5.19)$$

If the slack bus of the system is chosen as a boundary bus, all branches attached to it are excluded from the intersubdivision matrices. When one end of a cut branch is connected to a PV bus, the branch is not included in $[x_t'']$ and it has no elements in $[X_{tt}'']$. Hence, from the computational point of view, the cutting of branches connecting PV buses is recommended.

The grounding of boundary buses is represented in $[B']$ and $[B'']$ only, not in $[Y]$. When the fixed load admittances are included in the matrices, only the imaginary part is relevant to $[B']$ and $[B'']$. To avoid unnecessary computations, only the reactive power Q^{sp} is added to the diagonal elements and the reactive power mismatch should be corrected as

$$\Delta Q_1' / |V_1| = \Delta Q_1 / |V_1| - Q^{sp} |V_1| \cos 2\theta_1 \quad (5.20)$$

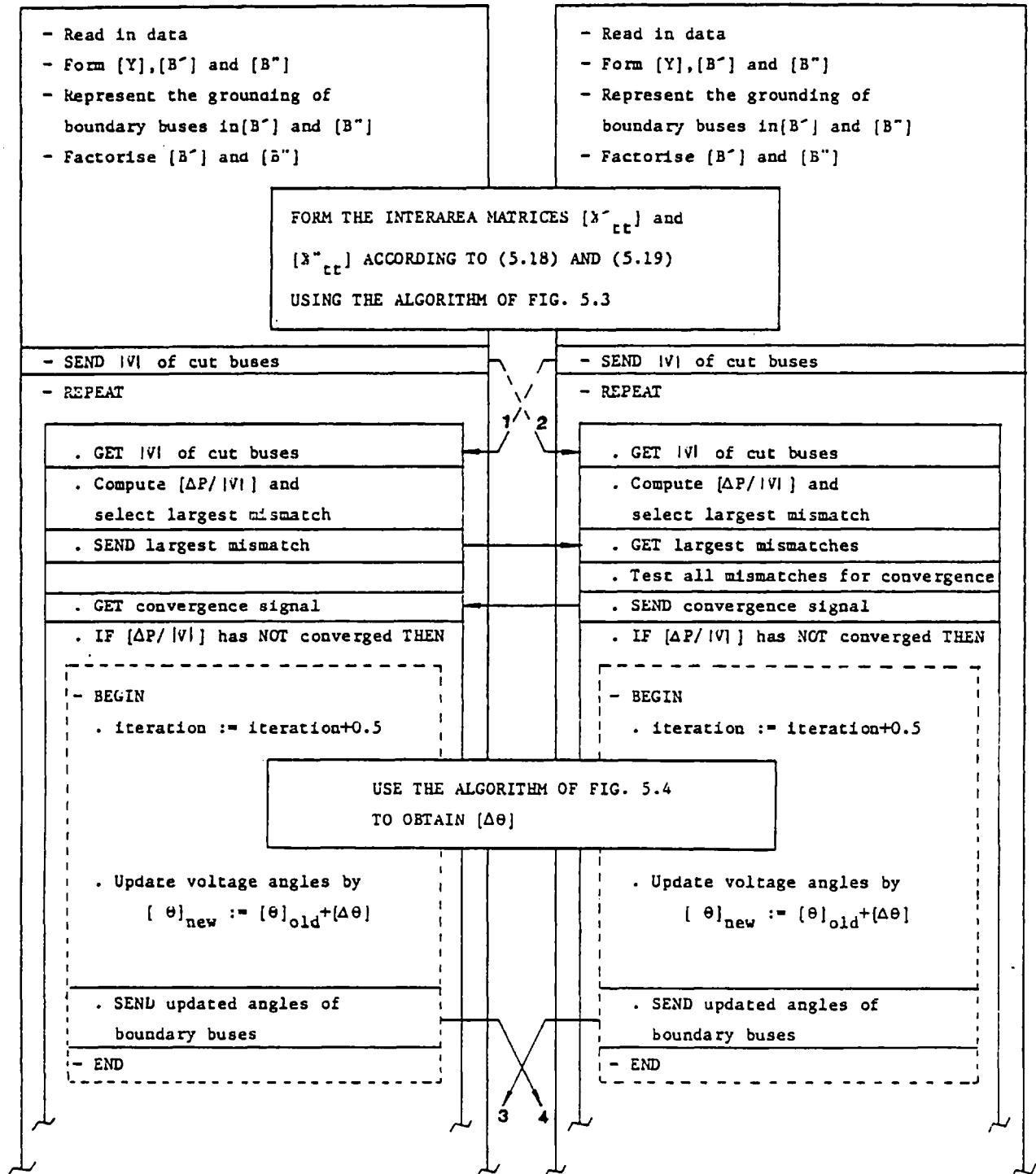
The application of this decomposition procedure does not affect the convergence properties of the FDLF. The main difference between the piecewise formulation of the FDLF given by (5.16)-(5.19) and that in Magdy El-Marsafawy et al.[91] is that expressions to calculate the factors for the adjustment of temporary reference buses are not required.

5.6.1 - Computational Algorithm

A computational algorithm for the solution of load flow problems with the FDLF using multiple microprocessors is given in Fig. 5.6. In the algorithm, the central MPU checks the convergence of the overall solution by testing the largest mismatch on each MPU. If and only if

PERIPHERAL MPUs

CENTRAL MPU



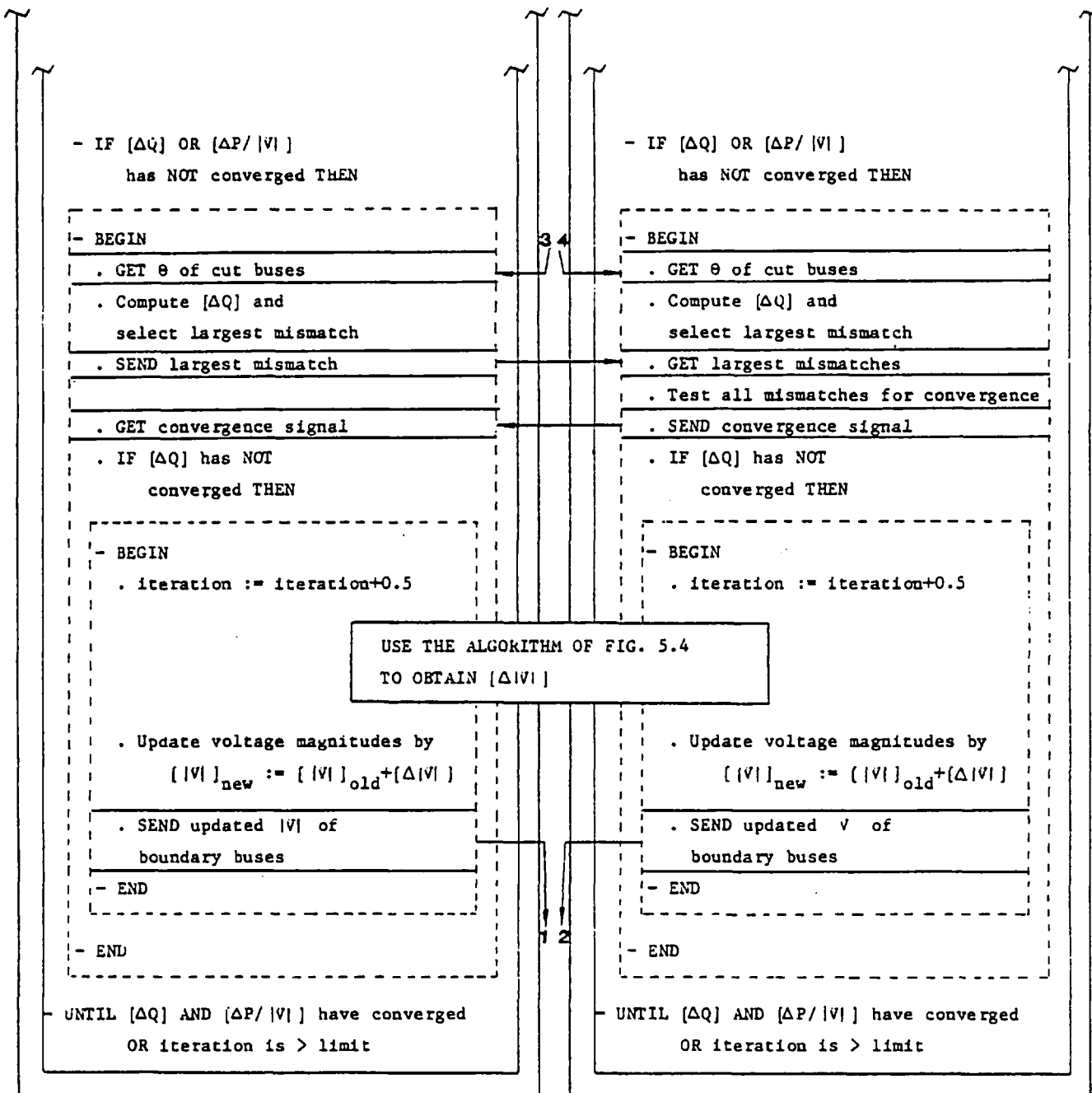


Fig. 5.6: Parallel FDLF Algorithm for the Application of Multiple Microprocessors.

all mismatches are within tolerance the solution has converged. The central MPU signals to the MPUs whether convergence has been achieved or not. On completion of a solution, or if a specified maximum number of iterations is reached, all MPUs proceed to display the individual solutions. For the calculation of ΔP the MPUs receive the voltage magnitude of cut buses (far end buses of torn branches) and the voltage angle for the ΔQ calculation. The length of the messages to be sent from an MPU is equal to the number of boundary buses in the subnetwork and the length of the received messages is equal to the number of cut buses. All MPUs proceed to update either $|V|$ or θ and send and receive only updated values.

It should be noted that for all the MPUs in the periphery the algorithm is exactly the same.

A numerical example illustrating the implementation of the algorithm is given in the Appendix B.

5.6.2 - Results Using Multiple Microprocessors

Since the algorithm outlined in Fig. 5.6 is essentially the same as the FDLF without tearing, the FDLF program described in sec. 4.2 was modified to implement this parallel processing algorithm. Communication procedures were included to send/receive $|V|$ or θ , to send the largest power mismatch and to receive the convergence signal to continue or stop execution. The parallel algorithm S4 presented in section 5.5.3 for factorisation and direct solutions was embeded in the program to solve the diakoptic equations.

The IEEE standard test systems were used to verify and evaluate the performance of the developed algorithm. The parallel algorithm produced numerical solutions indistinguishable from those obtained with the untorn approach and required the same number of iterations. The speedup obtained in the load flow solution of the test systems using up to 4 MPUs is depicted in Fig. 5.7. A linear speedup is obtained if the time taken by an n-processor algorithm is one nth of the time required by the uniprocessor algorithm [11]. Fig. 5.7 shows the effectiveness of the application of multiple microprocessors for the load flow solution of large power systems. It is observed that for the 30-bus system the fastest solution is obtained when 3 MPUs are used for the computation. For the solution with 4 MPUs Fig. 5.7 shows speeding-up factors of 1.9 and 2.7 for the 57-bus and 118-bus systems respectively. This represents a 63% reduction in the execution time for the load flow

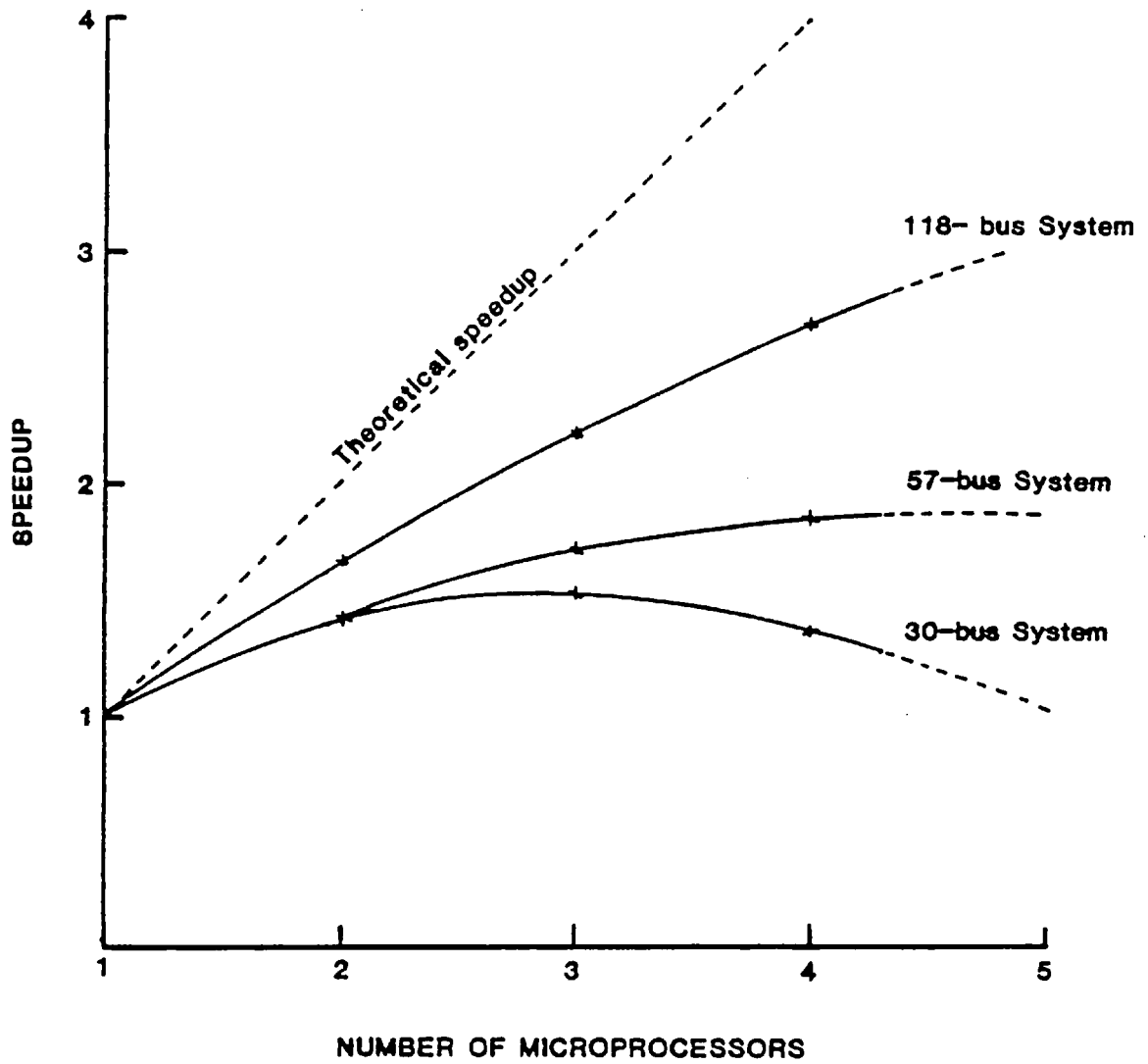


Fig. 5.7: Speedup for the Load Flow Solution of Test Systems Using Multiple Microprocessors

solution of the 118-bus system.

For all the cases studied the parallel solution was faster than the sequential solution without tearing. The gains achieved with parallel processing in this particular application correspond mainly to the power mismatch computation, which is the most time consuming calculation in the FDLF.

5.7 - CONCLUSIONS

A "star" configuration of multiple microprocessors for BBDF algorithms was implemented for the parallel solution of large power networks. A heuristic cluster algorithm is used to identify a practical decomposition pattern for a given network and an efficient piecewise technique is proposed to partition a network into the chosen subnetworks. Each subnetwork is then assigned to one microprocessor for solution.

Recently proposed matrix block elimination algorithms for the solution of the diakoptic equations representing a partitioned system have been evaluated for the solution of the linear equation $[Y][V] = [I]$ for different IEEE test systems. A variation developed for the implementation of matrix block algorithms produced faster execution times for both the formation of the interarea matrix and the direct solution computation than the algorithms proposed by Wu [87], Alvarado [88] and Hajj [89] for all the cases studied.

The results show that the formation of the interarea matrix, a problem usually disregarded in the theoretical evaluation of parallel processing algorithms, is the most time consuming calculation in the solution of a partitioned system. Also, it is noted that larger networks benefit more from parallel processing than smaller networks and that unless used with great caution the decomposition of networks for concurrent calculations may result in greater computational requirements than the uniprocessor calculation without tearing.

Results from the parallel solution of load flow problems show the effectiveness of the application of multiple microprocessors for the solution of large power systems. With the application of 4 MPUs for the solution of the IEEE 118-bus system the execution time is reduced to 37% of that required by a single MPU.

CHAPTER SIX

FAULT ANALYSIS USING MICROPROCESSORS

6.0 - INTRODUCTION

Fault analysis in power systems is performed primarily for system protection and planning purposes. The results obtained from the analysis are used to develop effective relaying and switching systems or to design specifications for future installations [23],[24],[26].

Faulted system studies constitute an integral part of transient stability analysis where a great variety of system situations is investigated to detect operating situations that make the system vulnerable to certain disturbances having potentially serious consequences.

This chapter describes the approach adopted for representing symmetrical and unsymmetrical faults in power transmission networks. The representation described allows the treatment of any network switching condition, i.e. branch outage, capacitor bank disconnection, Single-line-to-ground fault, etc., in the same manner.

The fault analysis algorithm developed for the efficient calculation of different system faulted conditions is particularly effective in practical studies where only the positive sequence parameters are of interest.

The algorithm provides a unified approach for the representation and computation of different network switching conditions in the dynamic power system simulator described in the following chapters. Also, the algorithm can be employed to perform detailed short circuit calculations using one microprocessor or a multiple array.

6.1 - FAULTED NETWORKS

A novel approach for the representation of unbalanced networks is presented by Podmore in [30]. The approach combines the computational

advantages of symmetrical components and the flexibility to handle special network unbalances in the phase coordinates frame of reference. The approach allows the efficient calculation of short circuit problems in bulk power systems and provides the flexibility of representing in detail special situations such as untransposed transmission line effects. In this approach, which was described in sec. 2.2.1, two types of unbalances are distinguished in a power network:

- . Unbalances caused by unsymmetrical faults
- . Unbalances due to untransposed transmission lines, unbalanced loads, etc.

An unfaulted network can be appropriately partitioned into a balanced subnetwork and a number of unbalanced subnetworks simply by removing the unbalanced elements from the main part of the network [30]. With this special form of partitioning, the unbalanced subnetworks are generally of a very simple form. For example, if there is only one untransposed transmission line in the network, by removing this line from the network, the rest of the passive elements form a balanced subnetwork and the untransposed line forms an unbalanced subnetwork. The balanced subnetwork is then represented using symmetrical components and the unbalanced subnetwork using phase coordinates.

A solution algorithm with the approach described may be summarised as follows [30]: the solution of the faulted network is initially obtained in terms of sequence components and is then transformed to phase coordinates. The phase coordinate solutions of all the subnetworks are then combined to form the solution of the original unpartitioned network.

The solution of the faulted network can be carried out advantageously by exploiting the structural properties of the sequence networks in the symmetrical components representation. In this work, attention is drawn to the natural decoupling of the sequence networks to formulate an algorithm for fast computations of different fault conditions in electric power networks.

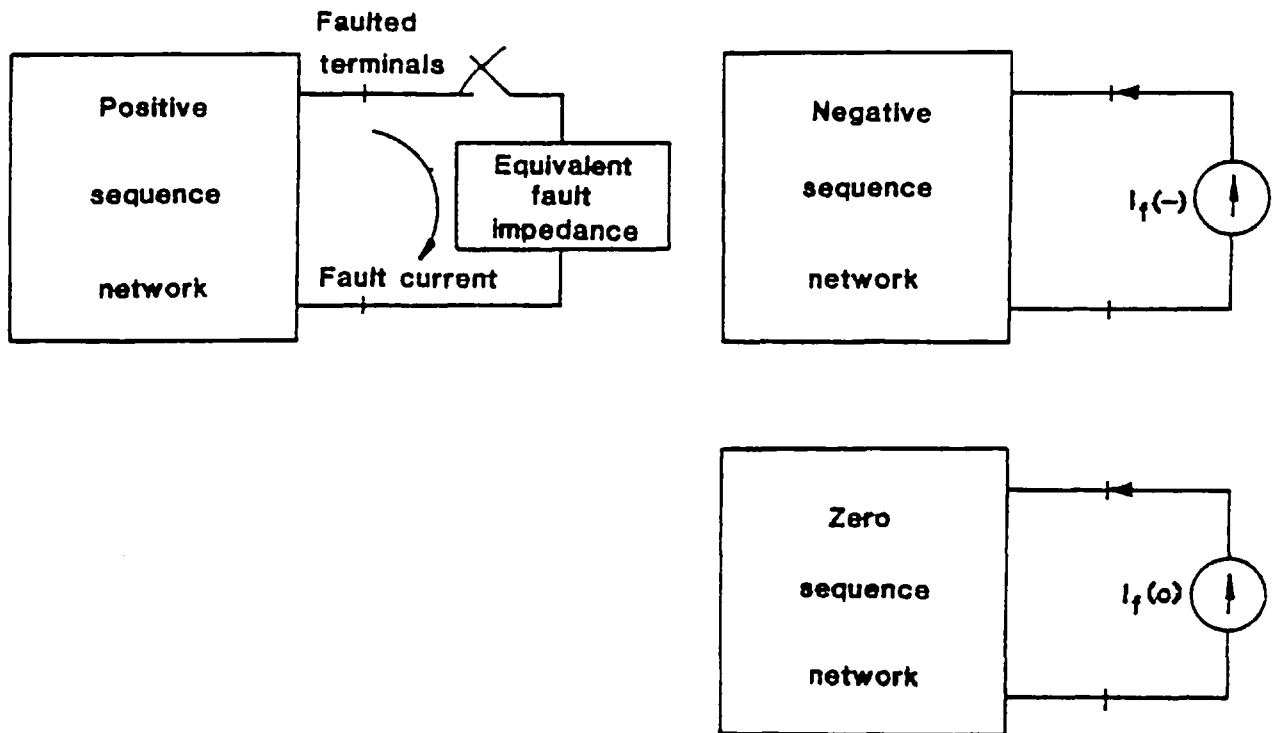
6.1.1 - Fault Representation

According to Anderson [26], the four types of fault of most general interest because of frequency of occurrence in bulk power systems are: single-line-to-ground, line-to-line, double-line-to-ground and balanced three—phase faults. The first three types constitute severe unbalanced operating conditions and the single line to ground fault is by far the most prevalent, making up to 70% of all transmission line faults.

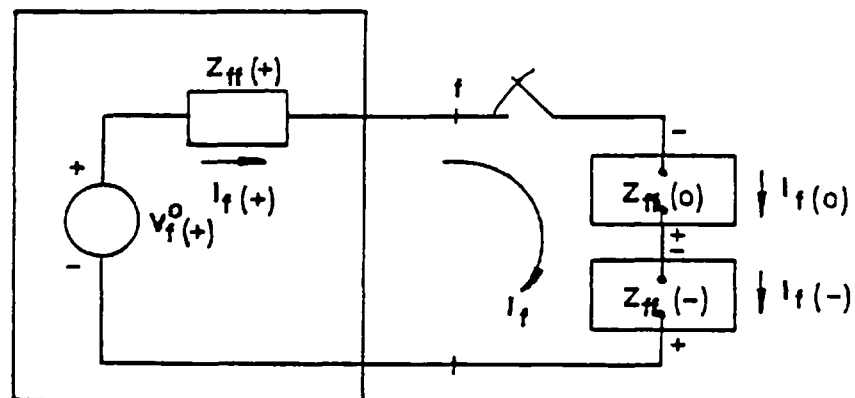
In symmetrical components, the faults are represented by interconnecting the sequence networks according to the type of fault to be analysed. When the sequence networks are represented by their Thevenin equivalents "seen" from the faulted terminals as illustrated in Fig. 2.2, each type of fault can be appropriately represented by interconnecting only the Thevenin equivalents regardless of the size and complexity of the networks. Furthermore, since the open circuit voltage of the Thevenin's equivalent of the (o) and (-) sequence networks is zero [26], the corresponding impedances may be combined into an equivalent fault impedance which can be connected to the (+) sequence network at the faulted terminals to represent a specified fault condition. The (+) sequence network then models the prefault network conditions and the representation of any types of faults may be considered as the addition of a new branch from the faulted point to ground in the (+) sequence network [95], as shown in Fig. 6.1. This modified (+) sequence network may be conveniently used to study the effect of any symmetric or unsymmetric fault on the (+) sequence network. In this representation, the fault current is given by the current flowing through the fault equivalent impedance [23].

In this approach, the effects of the faults on the (-) and (o) sequence networks are represented by injecting into these networks at the faulted terminals the corresponding sequence component of the fault current. The sequence components of the fault current are determined from the sequence interconnections which depend on the type of fault. For example, in the representation of a single-line-to-ground fault the sequence networks are interconnected in series [23], therefore the sequence components of the fault current are equal to the fault current.

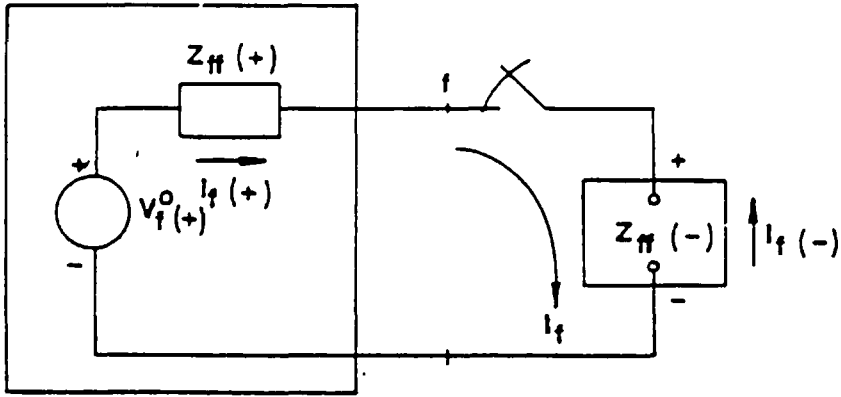
When the occurrence of a fault is represented on an unbalanced



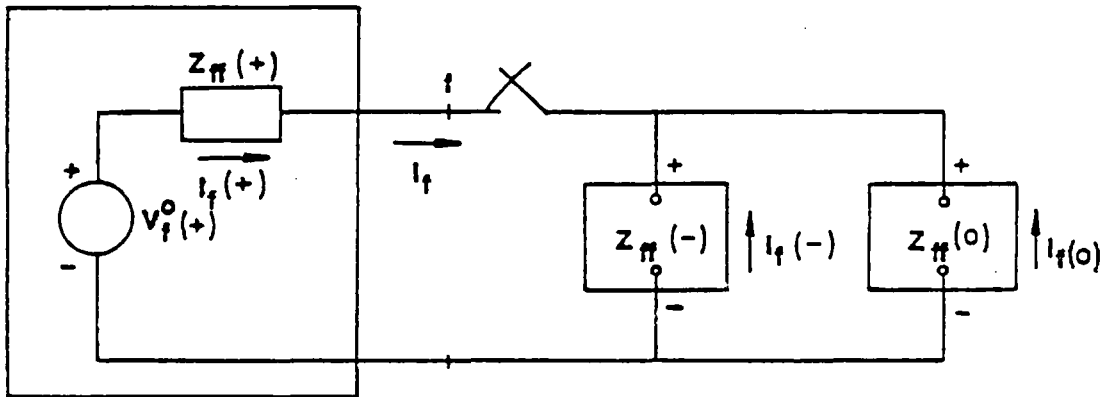
(a) General fault representation



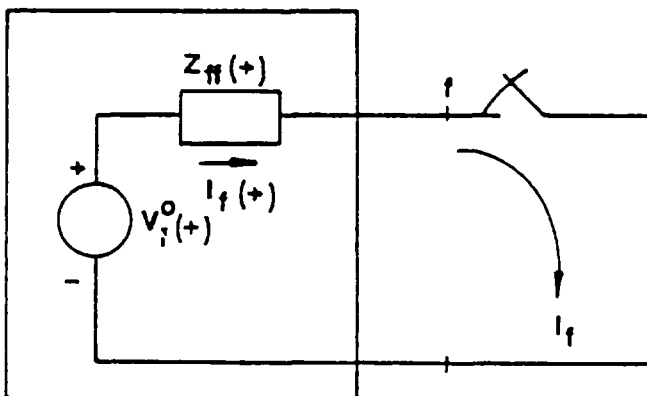
(b) Single-line-to-ground Fault



(c) Line-to-line Fault



(d) Double line-to-ground Fault



(e) Three-phase Fault

Fig. 6.1: Representation of Faults

network element, the fault current is calculated using phase coordinates representation as described by Laughton [28]-[29]. Then, the phase quantities are transformed into sequence components using the transformation (6.1) to represent the effects of the fault in the rest of the network.

$$\begin{bmatrix} (o) \\ (+) \\ (-) \end{bmatrix} = 1/3 \begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix} \begin{bmatrix} (a) \\ (b) \\ (c) \end{bmatrix} \quad (6.1)$$

The representation of faults using the procedure described is illustrated in Fig. 6.1. The fault equivalent impedance and the fault current for each of the faults considered are given in Table 6.1.

TABLE 6.1

Fault Parameters

Type of Fault	Equivalent Fault Impedance (Z_{ef})	F a u l t C u r r e n t		
		(+)	(-)	(o)
Single-Line-to-Ground	$Z_{ff(o)} + Z_{ff(-)}$	I_f	I_f	I_f
Line-to-Line	$Z_{ff(-)}$	I_f	$-I_f$	0
Double-Line-to Ground	$Z_{ff(-)}Z_{ff(o)}Z_x$	I_f	$-I_fZ_{ff(o)}Z_x$	$-I_fZ_{ff(-)}Z_x$
Three-Phase	0	I_f	0	0

where $I_f = V_f^0 [Z_{ef} + Z_{ff(+)}]^{-1}$

and $Z_x = [Z_{ff(-)} + Z_{ff(o)}]^{-1}$

The major advantage of representing the faults as given in Fig. 6.1 is that all types of fault may be treated as branch modifications in the (+) sequence network, permitting the computation of any network switching condition in the same manner.

6.2 - SOLUTION ALGORITHM

The main objective of short circuit analysis is the calculation of voltages and currents during fault conditions for the determination of circuit-breaker capacity and protective relay performance [24]. Many short circuit analysis methods are based on the evaluation of the bus impedance matrix and are variants of the classical $[Z_{bus}]$ method by Brown, et. al in [96]. Recently, work focused on the development of more generalised techniques [95],[97]. Undrill and Kostyniak [97] implemented a technique using optimally ordered triangular factorisation [65] and an iterative scheme for treating mutuals. Gross and Hong [95] developed an extension of the compensation method referred to as the two-step compensation method for the short circuit calculation of large power systems. Here, a variant of the latter technique based on the compensation method, which is reviewed in Appendix A, is used for short circuit calculations.

The solution of short circuit problems using the compensation method requires the replacement of the equivalent fault impedance attached to the (+) sequence network at the faulted terminals by a compensation current that produces identical effects. This compensation current in the fault representation described corresponds to the current through the fault impedance as given in Table 6.1. Thus, the system fault voltages are easily computed using the expression for the compensation method (4.7) introduced in chapter 4, where the voltage deviation vector $[\Delta V]$ will depend now on the type of fault, and it is given by

$$[\Delta V] = -I_f(+)[Z_{f(+)}] \quad (6.2)$$

where I_f is calculated according to the type of fault.

The sequence voltages are computed directly from

$$[V(+)] = [V^0(+)] + [\Delta V] \quad (6.3)$$

$$[V(-)] = -I_f(-)[Z_f(-)] \quad (6.4)$$

$$[V(o)] = -I_f(o)[Z_f(o)] \quad (6.5)$$

The negative signs reverse the positively defined direction of fault currents in Table 6.1 to indicate that the current is injected into the networks.

If there were any unbalances in the system before the fault, the voltages should be corrected as described in [30].

Since the $[Y]$ matrices of the (+) and (-) sequence networks differ slightly in the diagonal elements corresponding to buses with generation units or induction machines it is customary to assume that $[Y(+)] = [Y(-)]$ and therefore $[Z_f(-)] = [Z_f(+)]$.

The power of the formulation given by (6.2)-(6.5) is that factors usually neglected in fault analysis, such as shunt elements, off-nominal transformer ratios, load representations and loading conditions, as given by a base case ac load flow solution, may be taken into consideration without any difficulties. Furthermore, once the fault current has been calculated according to the specified type of fault, the calculations are performed independently on the sequence networks requiring only one column of the $[Z]$ matrix.

In stability analysis where only the (+) sequence parameters are of primary interest, the calculations are reduced to solve (6.2) and (6.3) requiring for the representation of any unsymmetrical fault the Thevenin's equivalent from the (o) sequence network at the faulted point. If the fault locations on transmission elements are assumed to occur near to the branch end buses, then the Thevenin's equivalents of the sequence networks are given by the diagonal elements of $[Z]$ corresponding to the nearest bus. Therefore, in dynamic simulations, the diagonal elements of $[Z(+)]$, $[Z(-)]$ and $[Z(o)]$ can be precomputed and stored with the (+) sequence network data for the fast computation of different network faulted conditions during the simulation.

Once the network fault voltages are known, the current flowing in the element i, j of each sequence element is calculated from

$$I_{ij}(+) = y_{ij}(+) [V_i(+)-V_j(+)] \quad (6.6)$$

$$I_{ij}(-) = y_{ij}(-) [V_i(-)-V_j(-)] \quad (6.7)$$

$$I_{ij}(o) = y_{ij}(o) [V_i(o)-V_j(o)] \quad (6.8)$$

If the element i, j is a transformer with delta-wye connections, the phase values of voltages and currents should be shifted $\pm (\pi/6)$ radians depending on which is the high voltage bus. The (o) sequence quantities do not require adjustment [98].

Line flow calculations are carried out to detect any overcurrents or to determine the performance of line protection relays. This calculation can be simplified if the line current is normalised to the line admittance as [99]

$$I_{ij}^+ = I_{ij}/y_{ij} = V_i - V_j \quad (6.9)$$

Then, the apparent impedance "seen" by a relay at the i th terminal can be computed directly from the bus voltages as

$$Z_{ij}^+ = V_i / I_{ij}^+ = V_i (V_i - V_j)^{-1} \quad (6.10)$$

If the relays are set to protect the total length of the line, the relay would operate if the normalized apparent impedance locus of the protected transmission line crosses the segment $[0,1]$ [99].

When the generators are represented by their Norton equivalents (see sec. 2.3.2), the prefault Norton's equivalent current is calculated from the base-case load flow solution using

$$I_{gi} = y_{gi} V_i + (P_{gi} - jQ_{gi}) / V_i^* \quad (6.11)$$

and the generators fault contribution is given by

$$I_{gfi} = I_{gi} - y_{gi} V_{fi} \quad (6.12)$$

6.2.1 - System-Wide Short Circuit Studies

System-wide short circuit studies are performed very quickly and efficiently using the solution technique described above. A system-wide short circuit study is a sequence of independent single fault studies typically carried out by protection engineers [95]. In such studies, a set of fault locations and a set of fault conditions are specified. After all the fault conditions at one location have been analyzed, the procedure is repeated at the next fault location. A computational algorithm for this type of study using the solution technique described is as follows: For each type of fault specified at a particular location, calculate the system fault conditions using (6.2)-(6.5) and compute line flows if required. The process continues until all of the specified fault conditions at all of the specified locations have been analyzed.

6.2.2 - Simultaneous Faults

The solution of a faulted network involving multiple simultaneous faults can be obtained by performing the calculation on a case-by-case basis using the recursive formula

$$[V^i] = [V^{i-1}] - Y_f^i V_f^{i-1} [Z_{.f}^i] \quad (6.13)$$

for $i = 1 \dots s$

where s = number of simultaneous faults and

$$Y_f^i = [Z_{.ff}^i(+) + Z_{ef}^i]^{-1}$$

The column of the impedance matrix for the i -th fault $[Z_{.f}^i]$ is

evaluated using the following recursive procedure [69]

$$\begin{aligned}
 [R^0] &= [Y]^{-1} [M^1]^t \\
 [R^1] &= [R^0] - Y_f^1 [Z_{.f}^1] [M^1] [R^0] \\
 &\vdots \\
 &\vdots \\
 [Z_{.f}^i] &= [R^{i-1}] = [R^{i-2}] - Y_f^{i-1} [Z_{.f}^{i-1}] [M^{i-1}] [R^{i-2}]
 \end{aligned} \tag{6.14}$$

where $[R]$ is a column vector

$[M]$ is a row vector which is null except for $M_f = 1$.

The procedure (6.14) is repeated for $i = 1 \dots s$

Although this procedure avoids partial or complete matrix refactorisations, the execution time and storage requirements increase as the number of simultaneous faults increases since for each fault it is necessary to store and compute a column of the impedance matrix using (6.14) and to perform a series of operations to calculate the network voltages according to (6.13). This recursive procedure is computationally preferable to matrix refactorisation only when the number of simultaneous faults is small. The formula (6.13) applies similarly to the (-) and (o) sequence networks for which $V^0(-) = V^0(o) = 0$.

6.3 - ALGORITHM IMPLEMENTATION AND RESULTS

In the implementation of the fault analysis algorithm described above it is assumed that $[Y(+)] = [Y(-)]$. The prefault network conditions are obtained from a base case ac load flow solution, generators are represented by their Norton equivalents (see sec. 2.3.2) and the loads by constant admittances. Resistance, line charging susceptance, shunt elements and off-nominal transformer ratios are also taken into consideration.

The storage of both $[Y(+)]$ and $[Y(o)]$ in main memory reduces approximately by one-half the maximum size of the network that can be handled on a microprocessor for short circuit calculations. A flexible scheme that extends the storage capability of a microprocessor was implemented by making use of a backing store in such a way that only the $[Y(+)]$ matrix is stored in main memory and the $[Z(o)]$ matrix, which

is precomputed, is stored on floppy disk in the disk units of the TI-990/4 host minicomputer.

When fault analysis calculations are performed on a microprocessor, an assembly code program [109] running on the minicomputer is used to retrieve the required elements of $[Z(o)]$ from the data file stored on floppy disk. For each specified fault location, the microprocessor sends the identification of the required $[Z(o)]$ column, and the program in the minicomputer searches in the data file for the record of that column and sends back all column elements to the microprocessor. These elements are used with the elements of the $[Z(+)]$ column computed on the microprocessor to calculate the fault current for each type of fault specified at the same location. The fault current sequence components and the (+) sequence fault voltages are displayed as soon as they are calculated. The calculation of the (-) and (o) sequence voltages and line currents is optional.

Most of the execution time in this fault analysis program is taken in retrieving the $[Z(o)]$ elements from backing storage. For instance, the short circuit calculation in the IEEE 30-bus test system of a single-line-to-ground fault on the transmission line 4-6 near to bus 6, required 10.1255 s to retrieve the $[Z(o)]$ column elements and 0.9574 s to calculate the required column of $[Z(+)]$.

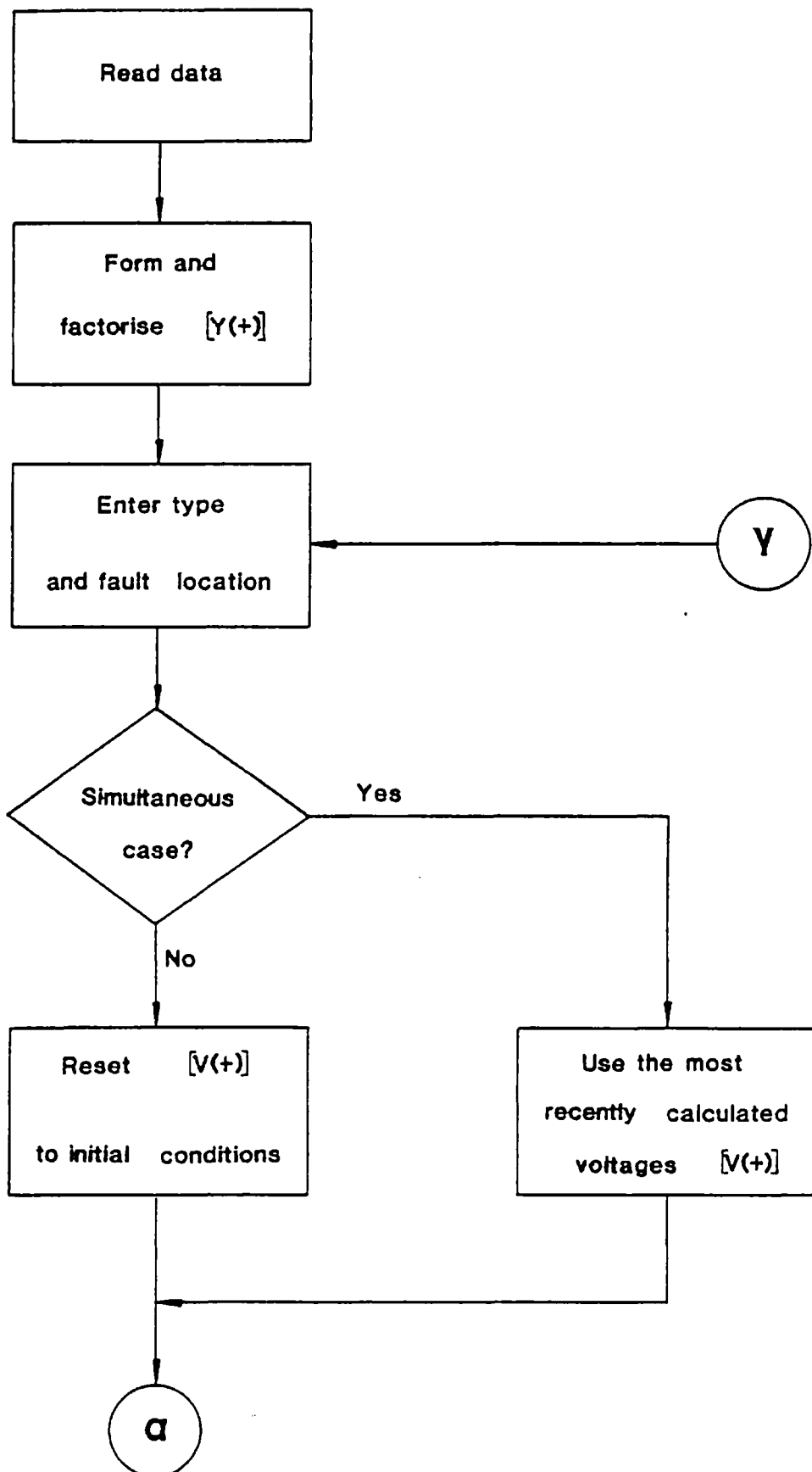
The fault analysis algorithm was programmed in PASCAL making use of the sparsity techniques described in chapter 4 to manipulate the admittance matrix of the sequence networks. The required columns of the $[Z(+)]$ matrix on the microprocessor are calculated using (4.8).

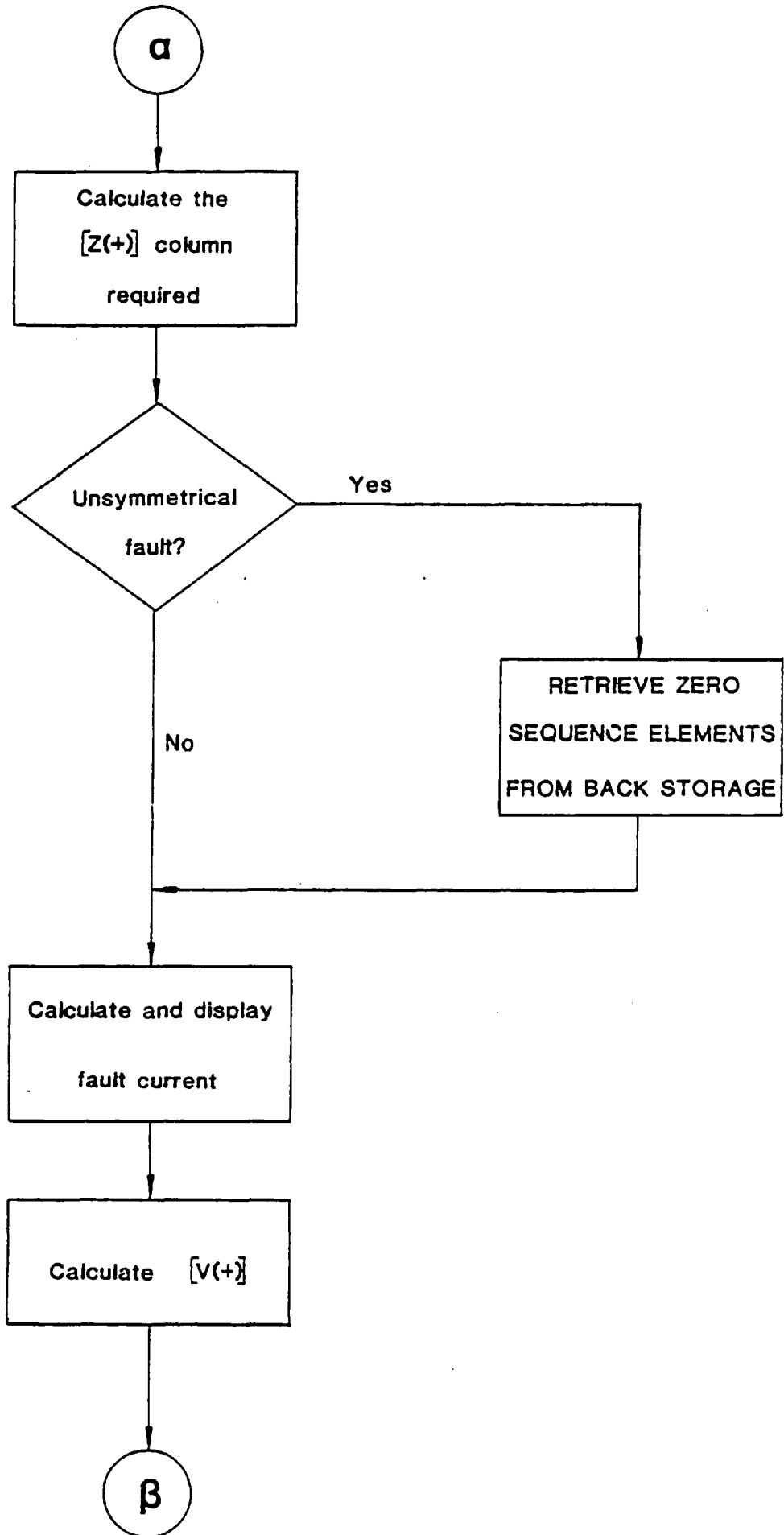
The program includes a microprocessor/user dialogue such that the user can select the type of study required from the following possibilities:

- . Single fault short circuit calculation
- . System-wide short circuit study
- . Simultaneous faults

At the end of the calculation of the fault (+) sequence voltages the user is given the option to request the computation of (-) and (o) sequence voltages, line flows and phase quantities of voltages and currents.

Fig. 6.2 illustrates the main parts of the implemented algorithm





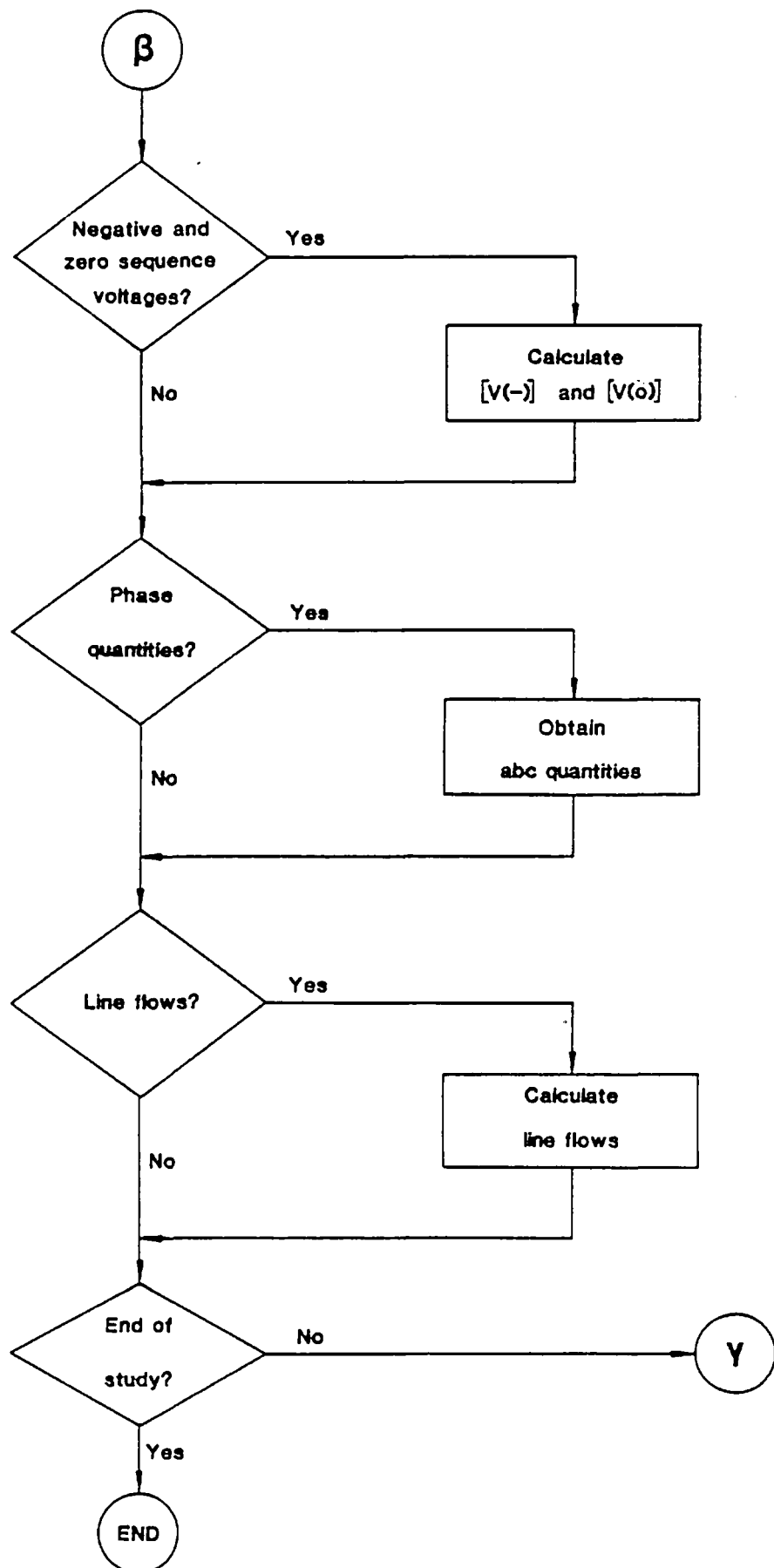


Fig. 6.2: Algorithm Implemented for Fault Analysis Studies

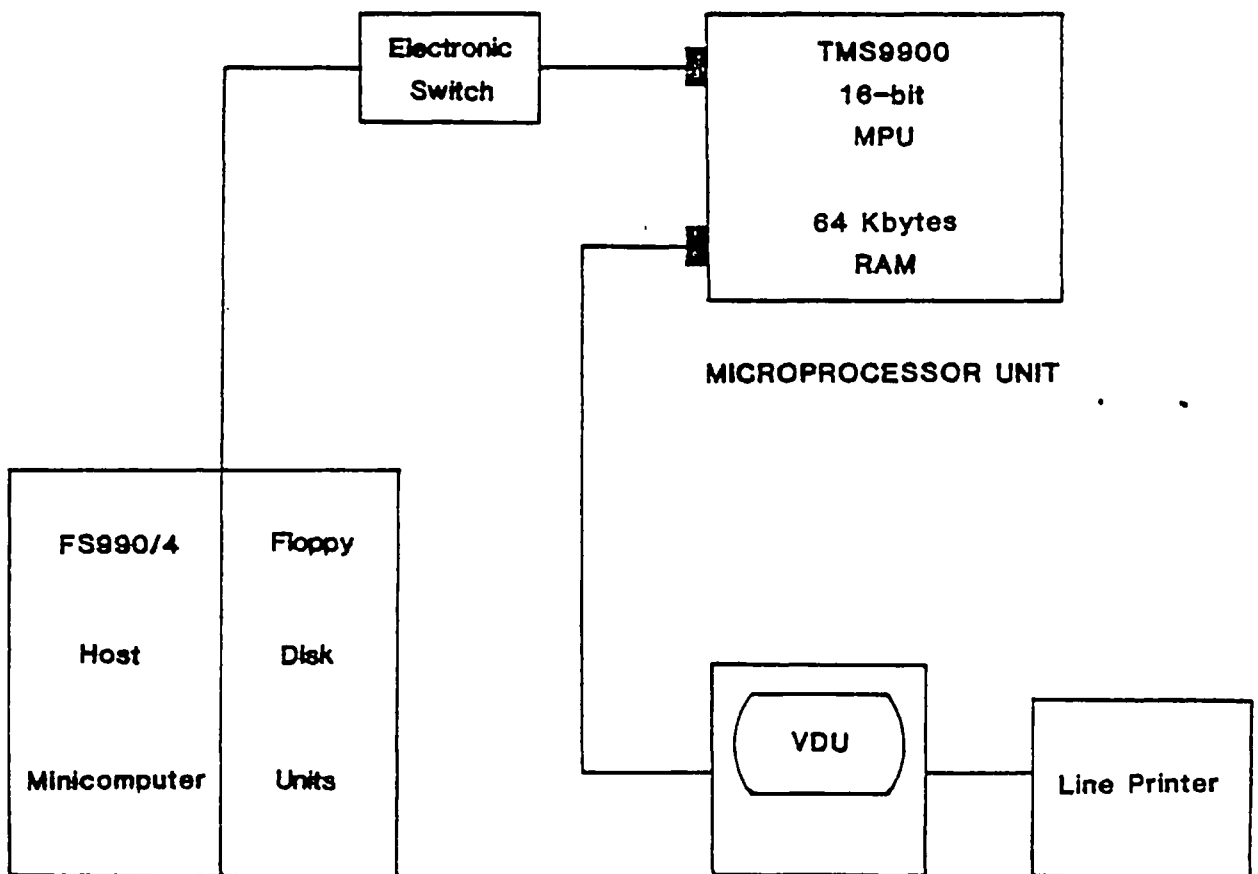


Fig. 6.3: Hardware Configuration for Fault Analysis Studies

and Fig. 6.3 shows the hardware configuration used.

The 6-bus and 14-bus test systems in Anderson [26] were used to verify the algorithm performance for the detailed solution of short circuit problems. For the purposes of investigating the capabilities of a TMS9900 microprocessor in the calculation of short circuit problems on large power networks, (o) sequence data for the IEEE standard test systems were obtained by multiplying the series branch reactances of the (+) sequence network by random factors in the range 2.0-5.0 [23]. As it may be anticipated, the execution time required to calculate the (+) sequence voltage and currents was very similar to the timings given in Table 4.3 for contingency analysis.

If, instead of using as backing storage the floppy disk unit in the minicomputer, another microprocessor is used to solve the (o) sequence network, then only the driving point impedance of the faulted point is needed on the microprocessor solving the (+) sequence network to calculate the fault current which is required on both processors to compute the corresponding sequence voltages and currents. The (-) sequence components of voltages and currents are calculated on the microprocessor solving the (+) sequence network. The time to transfer the required information between the microprocessors is of the order of 1 ms and the time for line flow calculations is effectively reduced by two thirds with this processing scheme.

Other parallel processing schemes are possible with the application of diakoptic techniques to partition the sequence networks into subnetworks in order to use more than two microprocessors for solution. The solution of the torn networks can be obtained straight forwardly using the parallel solution algorithms described in the previous chapter.

6.4 - CONCLUSIONS

The representation of symmetrical and unsymmetrical faults as the addition of branches to the (+) sequence network allows the treatment of any structural changes in a power system in essentially the same manner.

An algorithm based on the compensation method has been developed for short circuit calculations using microprocessors. The prefault network conditions are provided by a base-case load flow and the loads

are represented as constant admittances. Shunt elements and off-nominal transformer ratios are taken into consideration.

A very effective processing scheme was implemented for fault analysis using a microprocessor unit and a backing store. When fault studies are performed, the (+) sequence parameters are stored in main memory on the microprocessor unit and the elements of the (o) sequence impedance matrix required for the computation of unsymmetrical faults are retrieved from a data file in the back storage.

With the approach implemented the storage capabilities of a TMS9900 microprocessor have been extended for short circuit calculations of large power networks.

CHAPTER SEVEN

TRANSIENT STABILITY ANALYSIS USING MULTIPLE MICROPROCESSORS

7.0 - INTRODUCTION

The digital simulation of transient stability has evolved as a multiple-purpose time domain simulation. Dynamic simulations are performed in planning, design and operational studies to examine the behaviour of a system following a disturbance. A simulation yields the time response of all the variables included in the system representation, which in turn provide a "yes" or "no" answer to the question: Is the system stable?

Both the detail and size of the system representation directly affect the complexity and computational cost of the simulation. Approaches to alleviate the computation burden of transient simulations have been along three general directions. One approach is the reduction of the number of equations needed for the simulation while still providing an accurate account of the transient behaviour of the system. The second approach for faster solution time consists of the improvement of solution methods and numerical algorithms. The third and most recent approach is the application of multiple processors operating in parallel for the efficient solution of the transient stability problem. The first two approaches have shown substantial advance over the last few years. The application of multiprocessors has been mostly confined to feasibility studies. Brash et. al concluded in a recent publication [15] that research on the feasibility of using multiple processors for transient stability has progressed to the point where a start should be made on actual hardware implementation.

The challenge has been met and the application of an array of microprocessors operating in parallel for the transient stability analysis of bulk power systems is discussed in this chapter. The

parallel processing algorithms developed and the results obtained using different tests systems are presented.

7.1 - TRANSIENT STABILITY PROBLEM

The fast electro-mechanical transient behaviour of a power system is simulated in stability analysis to determine whether loss of synchronism will occur in the first few swings following a fault for normal and stuck breaker fault clearing condititons [17]. Transient stability studies typically involve simulations of around 10 s, see Fig. 1.7.

The transient stability problem requires the solution of all the mathematical expressions representing the behaviour of the different power system components. The general characteristics of the mathematical models to represent the major system components, as described in chapter 2, may be summarised as follows:

1. Transmission network - Algebraic equations of the form

$$[Y][V] = [I] \quad (7.1)$$

where $[I]$ is formed by generation and load currents. $[Y]$ includes the generator-terminal Norton shunts, see sec. 2.3. The structure of (7.1) will change at certain moments in time due to the simulation of disturbances in the system.

2. Generation plants - Nonlinear differential equations of the form

$$[\dot{X}] = f([X], [V]) \quad (7.2)$$

where $[X]$ is the vector of state variables.

3. Generator/Network interface - Algebraic equations of the form

$$[I_g] = h([X], [V]) \quad (7.3)$$

$[I_g]$ is the generator stator current transformed into the network frame of reference. If saliency is not represented

then $[I_g] = h([X])$.

4. Electric loads - Nonlinear algebraic equations of the form

$$[I_L] = l([V]) \quad (7.4)$$

Any loads represented by constant admittances may be directly included in the network admittance matrix and the corresponding elements in $[I_L]$ are set to zero.

Notice that when saliency is not represented in the generators, and the loads are modeled as constant admittances, the network equation (7.1) is linear.

7.2 - SOLUTION TECHNIQUES

The solution of the transient stability problem can be approached in many different ways. Stott [19] classified the different solution approaches into partitioned and simultaneous according to the way in which the differential equations (7.2) are interfaced with the algebraic equations (7.1). In the partitioned approaches, the solutions of (7.1)-(7.4) are alternated in some manner and the numerical techniques to solve (7.1) and (7.2) may be chosen independently of each other. In the simultaneous approach, the differential equations (7.2) are converted into algebraic equations using an implicit integration formula and these algebraized equations are grouped together with (7.1), (7.3) and (7.4) to form a single set of nonlinear algebraic equations. All the equations are then solved simultaneously using an iterative technique.

The essential problem to solve when choosing an approach for the solution of the transient stability problem is to ensure a reliable and accurate solution while minimizing the solution time. The solution scheme implemented in this work corresponds to a partitioned approach in which the identity of the power system components is retained. The solution steps of a partitioned approach are outlined below:

1. Form the bus current vector $[I] = [I_g] + [I_L]$ and solve (7.1) for $[V]$.

2. Transform the terminal voltage of each generator into the d-q frame of reference and solve (7.2) using an integration method.
3. Obtain the generator currents in the network frame of reference using (7.3).
4. Calculate load currents according to the load characteristic using (7.4).

Thus a partitioned approach consists of solving a set of algebraic equations and a set of differential equations at every time step during the transient stability simulation.

7.2.1 - Integration Techniques

The basic requirements of a method for the integration of ordinary differential equations are accuracy, stability and flexibility. It should be possible to obtain the required accuracy of a numerical approximation without a prohibitive amount of computational work and the integration formulas should be flexible for ease of implementation and use.

The numerical integration methods for the solution of ordinary differential equations fall into the main categories [19]

- . Implicit or Explicit
- . Single step or Multistep.

Implicit integration methods have good numerical stability properties and allow the use of a large integration step, but require the solution of a system of nonlinear algebraic equations by an iterative technique which can impose a step size restriction. Explicit methods are not very stable ones, compared with implicit methods but have the advantages that the integration formulas are applied directly to each of the individual differential equations and do not require an iterative solution. Multistep methods require the storage of previous values of the variables and/or their derivatives, and thus, in principle are efficient. However the process has to be restarted whenever a discontinuity occurs. Single step methods do not require information from previous time steps and are self-starting methods

[19].

The integration methods most commonly used in the solution of the transient stability problem are:

- Midpoint integration - This technique was one of the earliest methods used in classical stability studies [23]. The technique is sufficiently accurate for integration steps of 1 to 5 cycles.

$$x_{t+\Delta t} = x_{t-\Delta t} + 2\Delta t \dot{x}_t \quad \text{for } \epsilon \sim \Delta t^3 \quad (7.5)$$

- Euler's method - The method is simple to implement but it is very inaccurate, requiring small integration time steps.

$$x_{t+\Delta t} = x_t + \Delta t \dot{x}_t \quad \text{for } \epsilon \sim \Delta t^2 \quad (7.6)$$

- Runge-Kutta methods - These methods have been used extensively in transient stability calculations. They are more accurate than Euler but require more computational effort.

$$x_{t+\Delta t} = x_t + (K_1 + 2K_2 + 2K_3 + K_4)/6 \quad \text{for } \epsilon \sim \Delta t^5 \quad (7.7)$$

$$K_1 = \Delta t f(t, x_t)$$

$$K_2 = \Delta t f(t + \Delta t/2, x_t + K_1/2)$$

$$K_3 = \Delta t f(t + \Delta t/2, x_t + K_2/2)$$

$$K_4 = \Delta t f(t + \Delta t, x_t + K_3)$$

- Trapezoidal rule - The trapezoidal rule of integration has proven to be a most reliable and fast technique for the solution of differential equations. Perhaps its most salient property is its numerical stability, which allows the solution of stiff system of equations. A problem is stiff if the ratio between the largest and smallest time constant is high [19].

$$x_{t+\Delta t} = x_t + (\dot{x}_t + \dot{x}_{t+\Delta t}) \Delta t/2 \quad \text{for } \epsilon \sim \Delta t^3 \quad (7.8)$$

Dommel and Sato [38] developed a powerful technique by combining the trapezoidal rule with the midpoint integration method for the fast solution of the transient stability problem. This technique has become the choice for the implementation of transient stability programs because of the many advantages that the technique offers in terms of numerical stability, accuracy and computational efficiency.

7.2.2 - Integration of State Equations by Model Decoupling

The state equations representing different power plants do not directly interact, but couple only through the network. This natural decoupling has been generally exploited to solve the equations of each generation plant separately [21]. Further, the state equations of a generation plant may be decoupled according to the power plant elements represented as

- . Generator

$$\dot{x}_g = g(x_g, x_e, x_t, v) \quad (7.9)$$

- . Excitation system

$$\dot{x}_e = e(x_e, x_s, v) \quad (7.10)$$

- . Power system stabiliser

$$\dot{x}_s = s(x_s, x_g) \quad (7.11)$$

- . Governor-Turbine system

$$\dot{x}_t = t(x_t, x_g) \quad (7.12)$$

This decoupling allows the solution of the state equations using

different integration steps for each set of equations depending on the magnitude of the time constants associated with the individual sets. The slow response elements with longer time constants (governor-turbine) could be solved using a larger integration step than that required by fast response elements such as the AVR and power system stabiliser.

A more refined decoupling of state equations may be achieved by eigenvalue analysis. Lopez [101] uses a participation matrix based on the eigenvectors of the system of state equations to identify variables with similar oscillation frequencies. The variables are classified into different groups according to their rate of response, and each group is integrated using different integration steps.

7.2.3 - The Integration Scheme Implemented

The heuristic decoupling of state equations presented above is used to implement an integration scheme for the solution of the state equations in the transient stability problem. The integration scheme is mostly based on the technique by Dommel and Sato [38] in which the generator swing equations are solved using the combination midpoint integration as predictor and the trapezoidal rule as corrector. The electrical state equations of the generator and the rest of the state equations representing the generator control elements are algebraised using the trapezoidal rule of integration. The algebraised equations are solved iteratively at every simulation step. In order to avoid this iterative solution the generator equations (7.9) are solved using the following second order predictor/corrector scheme:

- . Predict the value $x_g(t+\Delta t)$ using midpoint integration

$$\dot{x}_g(t+\Delta t) = \dot{x}_g(t-\Delta t) + 2\Delta t \ddot{x}_g(t) \quad (7.13)$$

- . Correct the predicted value using trapezoidal integration

$$x_g(t+\Delta t) = x_g(t) + [\dot{x}_g(t) + \dot{x}_g(t+\Delta t)] \Delta t / 2 \quad (7.14)$$

In this integration scheme the network equations are solved only

once at the end of the correction process during the simulation, but immediately after a discontinuity in the network due to faults or switching operations, Euler's method is used to restart the midpoint integration and the network equations are solved twice, once at the end of the predictor stage and once at the end of the correction process.

A similar predictor/corrector scheme is used to solve the differential equations of the generator controls, but instead of using midpoint integration, Euler's method is used as predictor to avoid restarting the midpoint integration whenever a variable violates its limits.

7.2.4 - Network Solution

Modelling aspects such as dynamic saliency and nonadmittance loads, e.g. constant power or variable load as a function of bus voltage and frequency, play an important part in the solution of the network equations (7.1). If saliency and/or nonadmittance loads are represented in the system, (7.1) must be solved iteratively.

The IEEE Computer Analysis of Power Systems Working group [102] pointed out the various effects of using different load-voltage characteristics on system dynamic simulations and it was shown that network solutions may not be obtained for voltages lower than normal when the loads are represented as a function of the bus voltages. To overcome this problem it is recommended in [102] to change the load representation to a constant admittance whenever the voltage falls below a certain minimum value. The load-voltage characteristic could be restored when the voltage is restored above the minimum. This procedure not only allows network solutions for low voltage levels but eliminates unrealistic load representations. Arrillaga [103] suggests a very effective approach to implement the change in the load representation according to changes in the bus voltages. In this approach the loads are converted at the beginning of the simulation into constant admittances using the initial conditions available from a load flow and the load admittance obtained Y_{10} is included in the admittance matrix. The loads represented as a function of bus voltage and frequency are converted into a characteristic admittance Y_1 which is computed for each network solution by (7.15).

$$Y_1 = (P_1 - jQ_1) / |V|^2 \quad (7.15)$$

where P_1 and Q_1 are given by the load characteristics.

Thus, the load current $[I_1]$ can be computed as the deviation of the load characteristic from an admittance characteristic

$$[I_1] = (Y_{10} - Y_1)V \quad (7.16)$$

When the voltage drops below some predetermined value, say 0.8 p.u., the load representation is changed to a constant admittance. To preserve the continuity of the load power in the change of representations, at a voltage V_{low} , the coefficients (2.20a) and (2.21a) are recalculated using

$$K_p^{new} = K_p^{old} |V_{low}|^{pv-2(f)pf} \quad (7.16a)$$

$$K_q^{new} = K_q^{old} |V_{low}|^{qv-2(f)qf} \quad (7.16b)$$

The nonlinearities introduced by the nonadmittance loads in the network equations may be eliminated by making the assumption that the voltage of the previous time step can be used to compute the load current I_1 [104]. This assumption is acceptable because the changes in magnitude and angle of the bus voltages from one time step to the next are very small, except immediately following a switching operation. Dandeno and Kundur presented in [104] various computational results of several tests on a complex multi-machine system to demonstrate the validity of this approximation and they presented a non-iterative technique for the solution of the network equations, while allowing the load MVA-voltage relationship to vary. An iterative solution is performed only for the time step immediately following a switching operation or some major electrical system discontinuity.

The technique implemented to solve the network equations is based on the approach suggested by Dandeno and Kundur [104] and the load characteristics are considered to be valid only for a small voltage deviation from nominal (0.8 1.2 p.u.), as recommended by Arrillaga in [103]. When the voltage magnitude of a given bus is out of the specified bounds the load representation at that bus is changed automatically to constant admittance using the procedure described above.

Techniques for avoiding the 2 or 3 iterations due to dynamic saliency (see sec. 2.3.1.3) are hardly worthwhile [38]. The saliency corrective process can be performed quickly if the network equations are rearranged for solution as

$$[Y][V_1] = [I_g] - [I_1] \quad (7.17)$$

$$[Y][V_{sal}] = [I_{sal}] \quad (7.18)$$

$$[V] = [V_1] + [V_{sal}] \quad (7.19)$$

If saliency is not represented then $[I_{sal}] = 0$ and $[V] = [V_1]$.

$[I_{sal}]$ is a vector with nonzero elements in the locations corresponding to buses with salient generators.

The solution process of the network equations (7.17)-(7.19) proceeds as follows:

1. Form the current vector $[I]$
2. Obtain the network voltages by performing a forward and backward substitution with the factorised $[Y]$ in (7.17)
3. If saliency is represented then perform steps 3.1-3.5 to correct the voltages for dynamic saliency.
 - 3.1 Solve (7.18) for $[V_{sal}]$
 - 3.2 Correct network voltages for buses with salient generators only using (7.19)
 - 3.3 Obtain a new estimate of $[I_{sal}]$ using (2.18)
 - 3.4 Repeat 3.1-3.3 until $[I_{sal}]$ is within some specified tolerance
 - 3.5 Correct network voltages using (7.19)

The iterative process to correct for saliency is accelerated if the bus voltage angles are advanced by the same amount by which the generator internal angle has changed from $t-\Delta t$ to t . Dommel and Sato pointed out [38] that omission of this rotation would double the number

of iterations.

With the approach implemented for the network solution, iterations are performed only when saliency is represented in the generator models and a discontinuity occurs in the network. The algorithm was implemented using the sparsity oriented programming techniques described in chapter 4.

7.2.5 - Simulation of Disturbances

The simulation of disturbances plays an important part in transient stability analysis. Disturbances in the network can be simulated using the algorithms for contingency and fault analysis described in chapters 4 and 6 respectively.

The disconnection of loads can be simulated using (7.16) simply by setting $Y_1 = 0$ which eliminates the effect of the load admittance Y_o included in $[Y]$.

The disconnection of a generator may be simulated in the network by injecting the compensation current

$$I_{gc} = Y_{fa} V \quad (7.20)$$

instead of the generation current I_g to cancel out the effect of the generator Norton admittance Y_{fa} included in $[Y]$. When a generator is reconnected, the generator is automatically synchronised by setting the generator phase angle equal to the angle of the bus terminal voltage prior to reconnection.

With this approach to simulate changes in the system, two types of compensation methods are performed. A post-compensation method for faults and branch outages, and a pre-compensation method for switching out generators, loads and capacitor banks. The post-compensation method requires the calculation of compensation factors to obtain the voltage deviation vector $[\Delta V]$ which is used to compensate the network solution to represent the effects of faults and/or branch outages. In the pre-compensation method the current vector $[I]$ is compensated before the network solution to eliminate the effects of the shunt elements in $[Y]$.

The simulation of various changes in the system as a result of a

cascade of events is easily handled with the approach implemented. Imagine the following power system scenario. A large generator is lost in the system at time t_1 , after a short period of time an overloaded line is tripped out by relay protective action at t_2 and at time t_3 an operator emergency measure is taken by cutting off a large load in the system. At time t_3 and thereafter during the 3 events-on period the solution of the network equations is performed as follows:

- . Compensate $[I]$ to represent the generator and load outages.
- . Solve the network equations $[Y][V] = [I]$ for $[V]$.
- . Compensate $[V]$ to represent the effects of the line outage.

The major advantage of this unified approach to simulate changes in the system is that it requires very little arithmetic, since refactorisations of the network admittance matrix are not necessary for a small number of changes in the network.

7.3 - APPLICATION OF PARALLEL PROCESSING TECHNIQUES

Several algorithms have been proposed in the literature for the solution of the transient stability problem using multiple processors [11]-[16]. Hatcher, et. al [13] grouped the transient stability equations into a single set of nonlinear algebraic equations and proposed several multiple processor structures for solution using different iterative algorithms. Fong and Pottle [14] solved the set of nonlinear algebraic equations simultaneously using the Newton-Raphson method and exploited the Jacobian matrix properties to propose multiple microprocessor structures for solution. Brash, Vann Ness and Kang [11],[15] solved the differential and algebraic equations using the technique by Dommel and Sato in [38]. They evaluated on a general purpose computer the performance of a multiprocessor network of up to 50 processors for power system dynamic analysis. These parallel processing algorithms were not implemented on existing hardware since the multiple processor structures proposed, which were briefly reviewed in sec. 1.1, have not yet been built.

The parallel processing scheme implemented for the solution of the transient stability problem using microprocessors differs from previously proposed techniques in the way in which the computational

tasks are distributed among the processors and in the techniques used for solution. The task distribution is based on a natural decomposition of a power system, whereby each microprocessor solves the mathematical representation of identifiable elements or groups of elements of a given system. To illustrate this, consider the small power system depicted in Fig. 7.1a. Three microprocessors may be used to represent this system. One microprocessor for each generation plant and one for the transmission network and load. Fig. 7.1b shows the equations assigned to each microprocessor and the interchange of information required for solution. This decomposition of the system is independent of the detail and complexity included in the modelling. Identical or completely different models could be used to represent the generation plants. It should be noted that only two quantities are exchanged between an MPU representing generators and the one representing the network. The bus terminal voltage is sent from the MPU representing the network to MPUs representing generators and these send back to the network the generation injected current referred to the network reference frame after solving the generator equation (7.9) using the received voltage.

7.3.1 - Algorithm Implementation

The techniques presented in previous sections to solve the power plant and the network equations, and the communication procedures (chapter 3) for exchanging information between the processor units (MPUs) are required for the implementation of the parallel processing scheme described above. Also, procedures are required to simulate disturbances, e.g. faults (chapter 6) and branch switchings (chapter 4) and a load flow solution (chapter 4) to provide the system steady state initial conditions.

The algorithm implemented is described assuming a detailed modelling of the power system components. The algorithm detects from the generator data if saliency is represented in the model in order to correct the network solutions accordingly. The algorithm is organised into three main sections: initialisation, changes in the system and simulation.

The initialisation phase illustrated in Fig. 7.2, is executed only at the beginning of the simulation for each system. At this

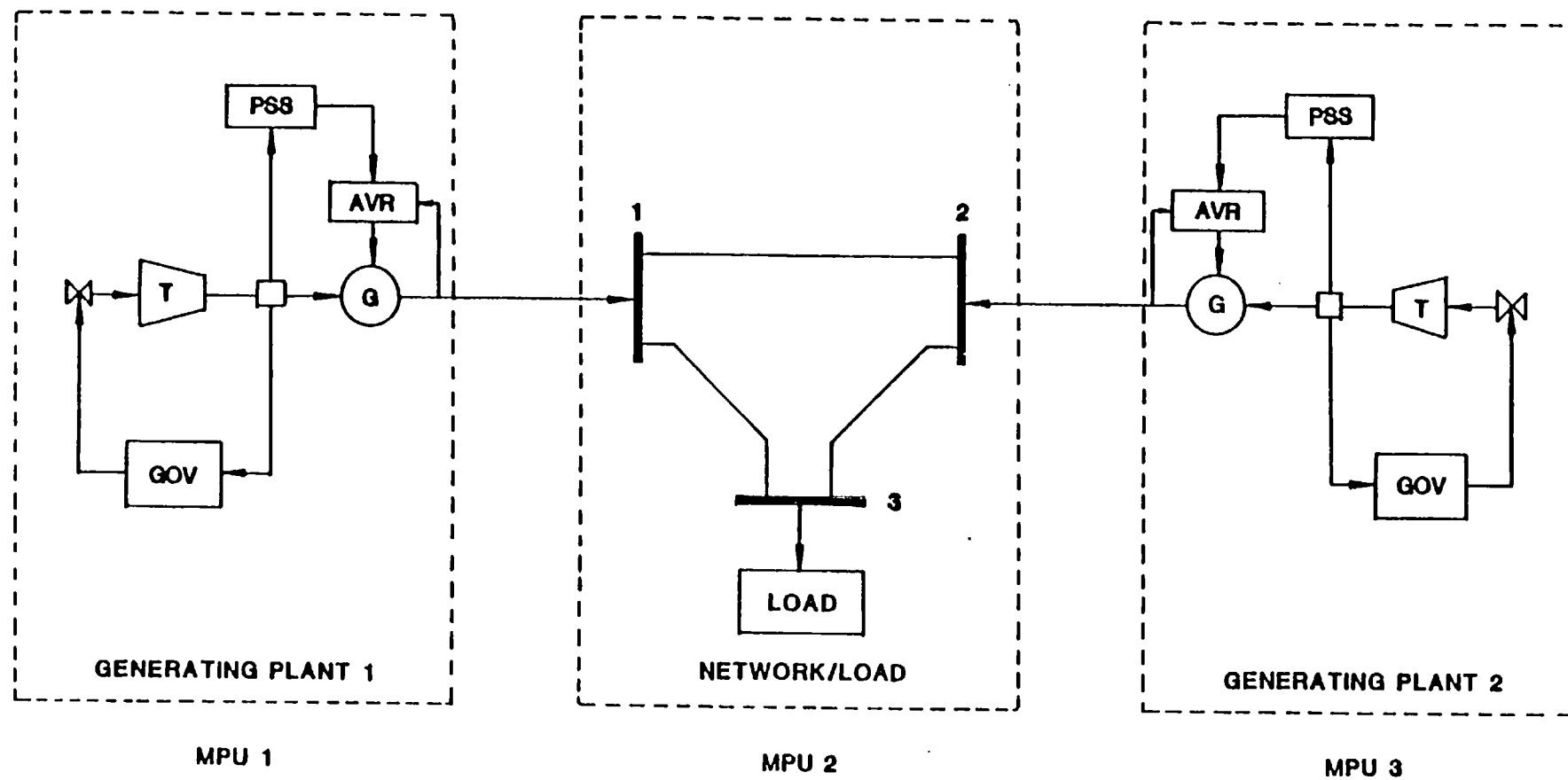


Fig. 7.1a: Representation of a 3-bus Power System Using Microprocessors

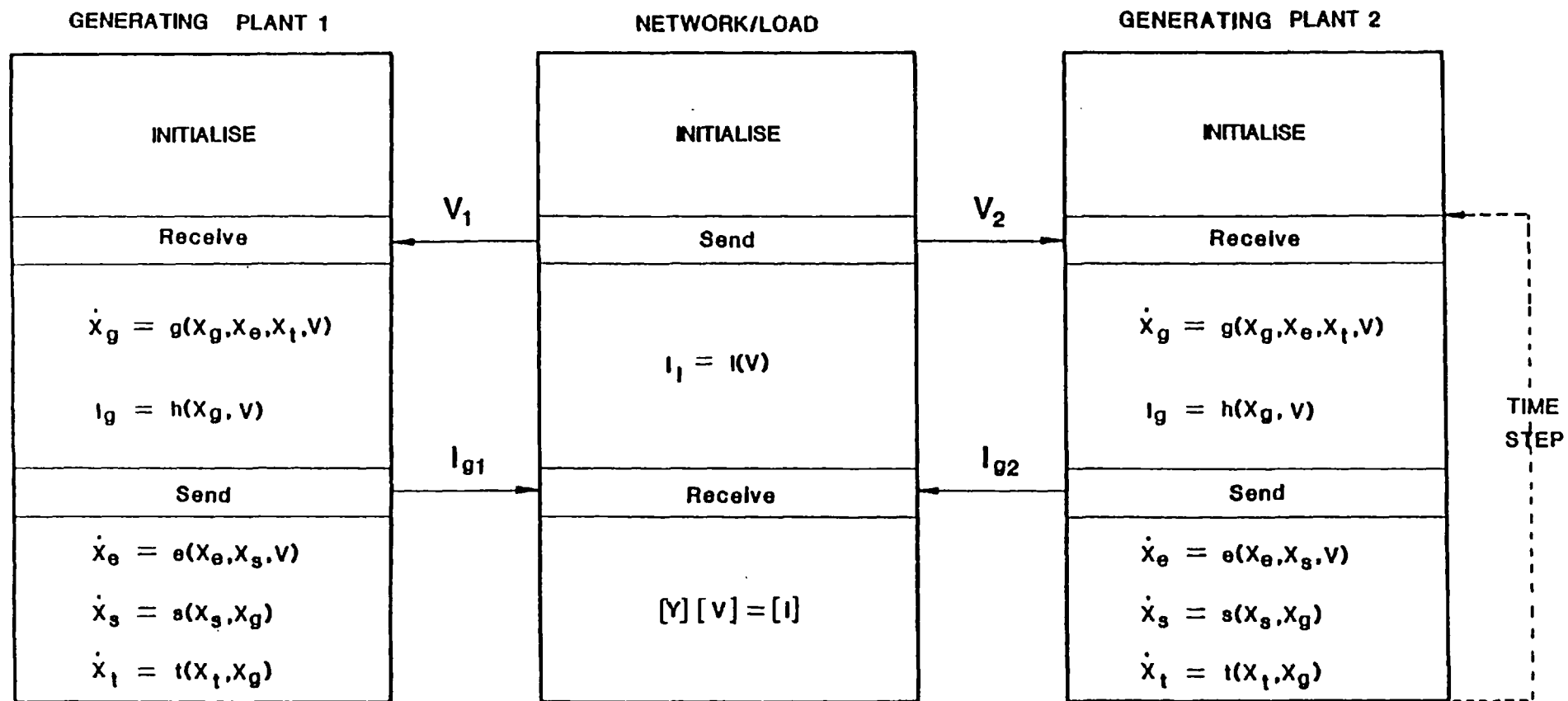


Fig. 7.1b: Task Distribution Among the MPUs for Dynamic Simulations

initialization stage, the MPU solving the network sends to the MPUs solving the generators the initial loading conditions of generator buses, available from a base case load flow, to initialise the state variables of the power plant model and the latter send back the generator Norton admittances and the consideration of saliency in the generator model. If saliency is represented, the MPUs solving salient generators will transfer 5 real values to the MPU solving the network at each time step. Two values correspond to the complex components of I_g and two values to I_{sal} . The fifth value corresponds to the change in the generator internal angle from $t-\Delta t$ to t . MPUs solving nonsalient generators will transfer only the 2 real values of the generation current I_g .

The section dealing with the simulation of changes in the system due to faults or switchings is shown in Fig. 7.3. This section is executed only at the instant a disturbance is applied to or removed from the system. At this particular time step the network equations are solved twice. Once after the predictor stage in the solution of the differential equations of the generators and once after the correction phase.

The third section of the algorithm which is shown in Fig. 7.4 represents the main simulation loop. This part is executed at each time step during the simulation.

All the control and timing of the simulation is centralised on the MPU solving the network. At the beginning of each time step a signal is sent to the MPUs solving the generators to indicate any changes in the system or the end of the simulation. The synchronisation during the simulation is provided by the scheme used for exchanging information. The MPUs solving generators will not proceed to solve the generator state equations until they have received the terminal voltage and the MPU solving the network will not compute new system voltages until the new generation currents are available.

7.4 - DYNAMIC SIMULATION OF LARGE POWER SYSTEMS

The parallel processing scheme described in the preceeding section can be used for the dynamic analysis of bulk power systems but requires a large number of processors if the number of generation units in the system is considerable. For example, to simulate the New England test

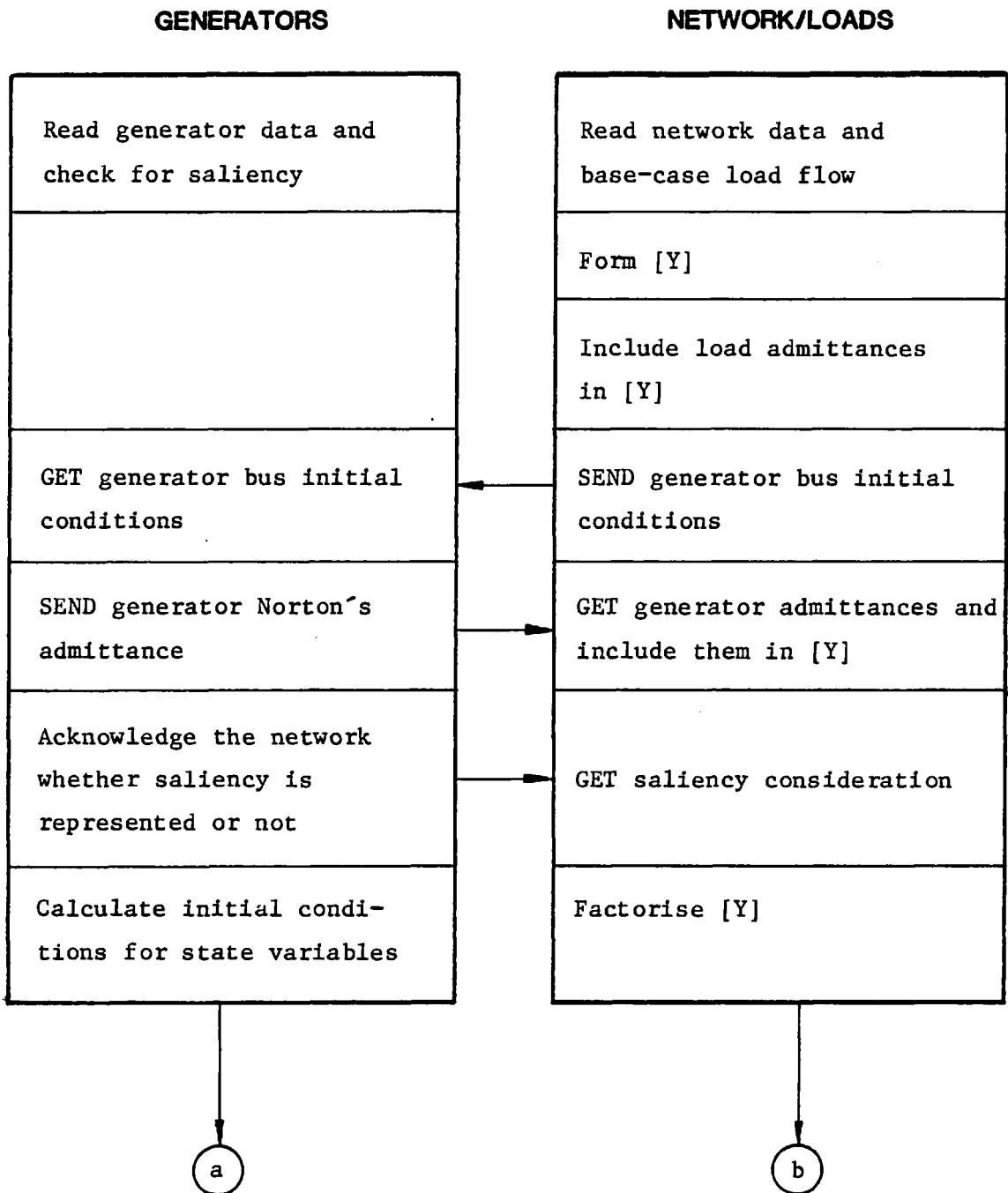


Fig. 7.2: Initialisation Stage for Dynamic Simulations

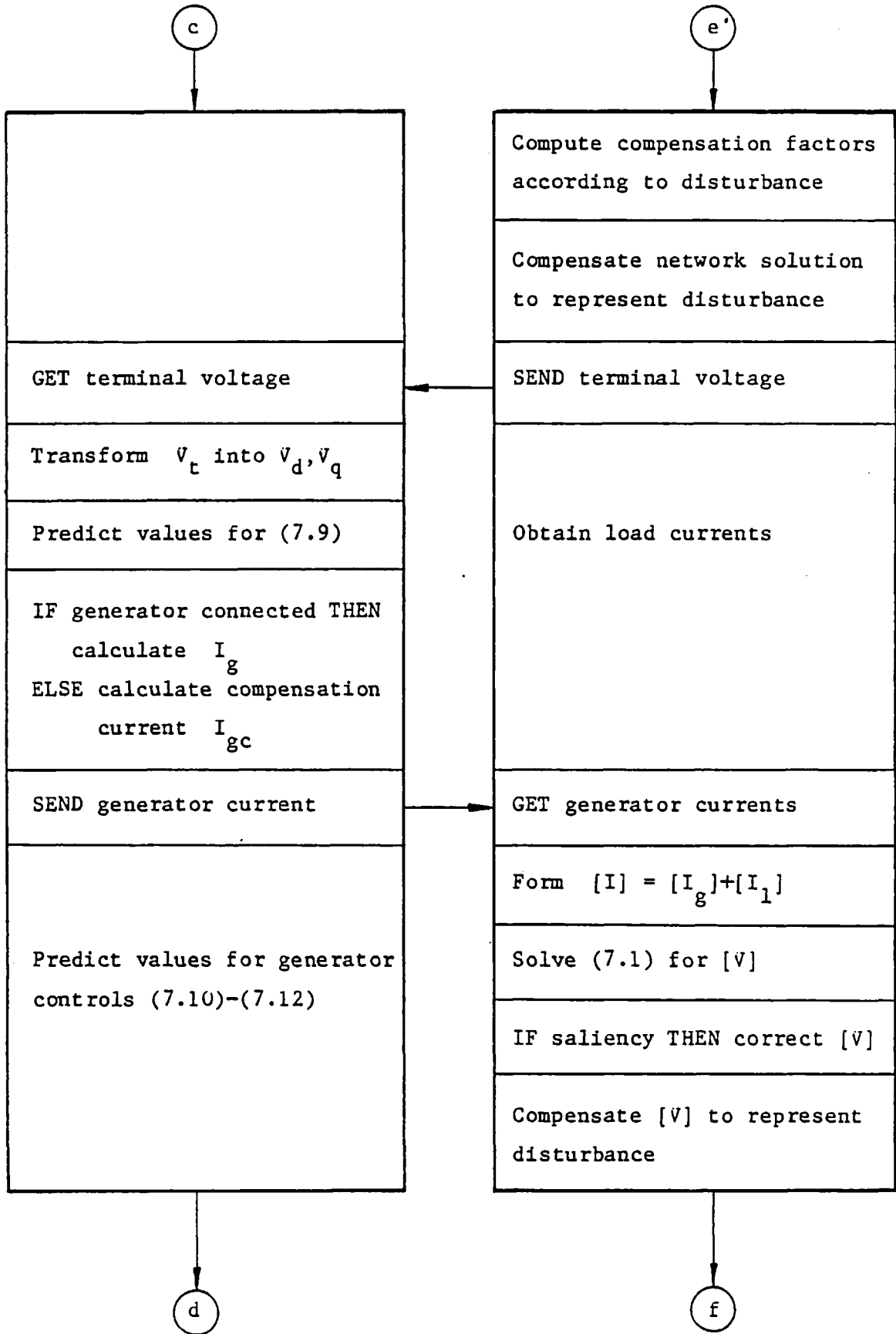
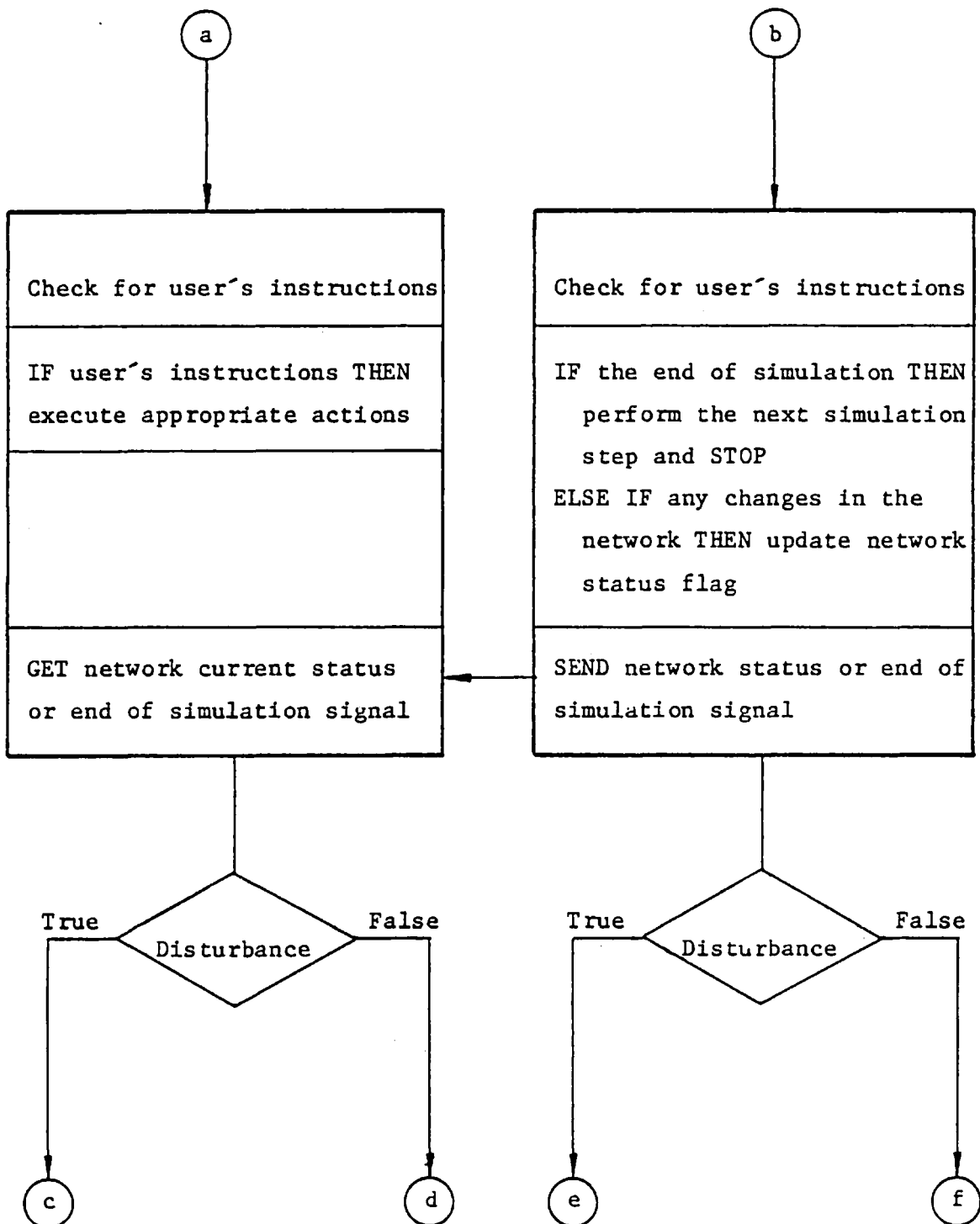
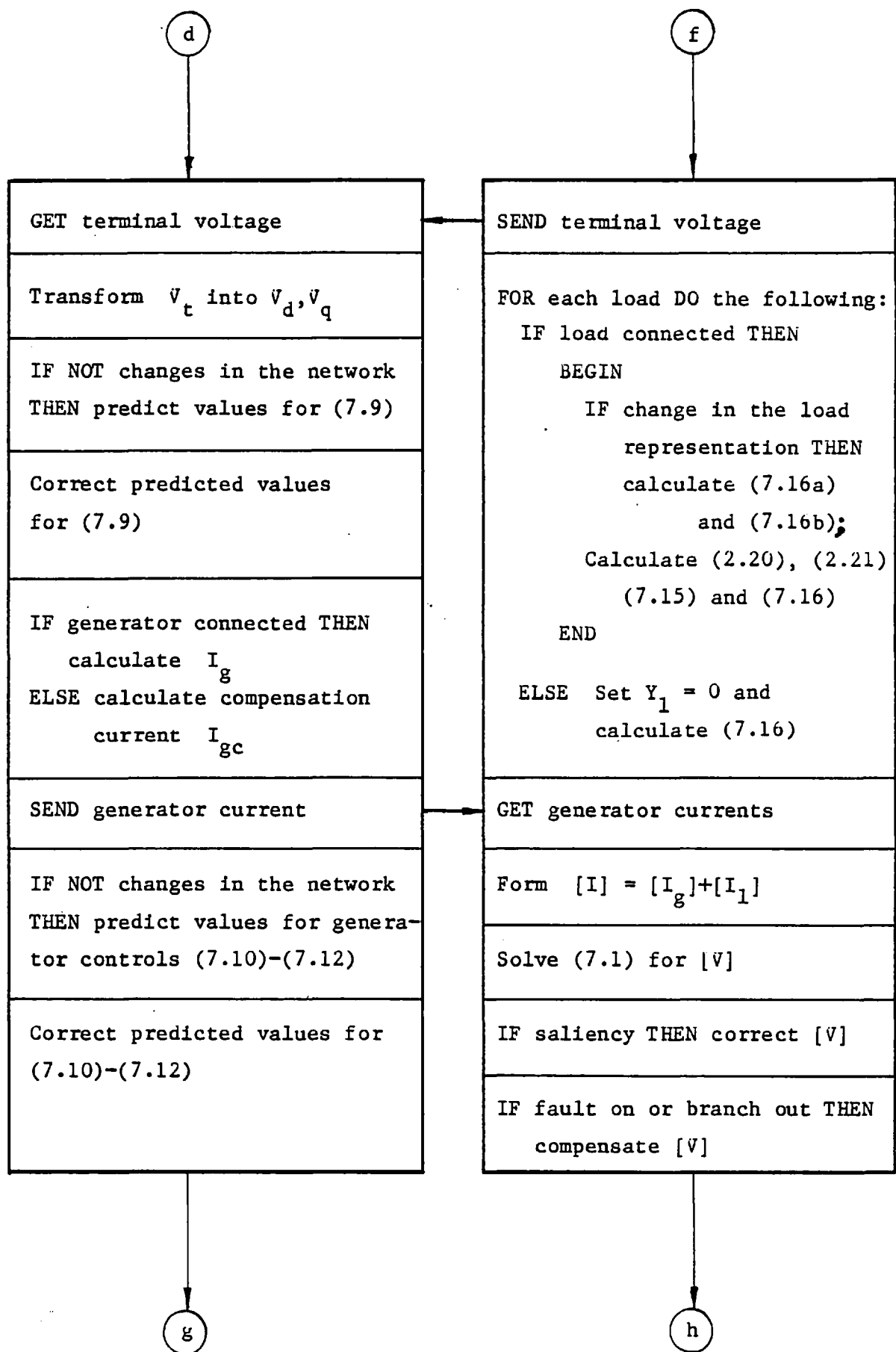


Fig. 7.3: Simulation Step at the Instant a Disturbance Occurs in the Network





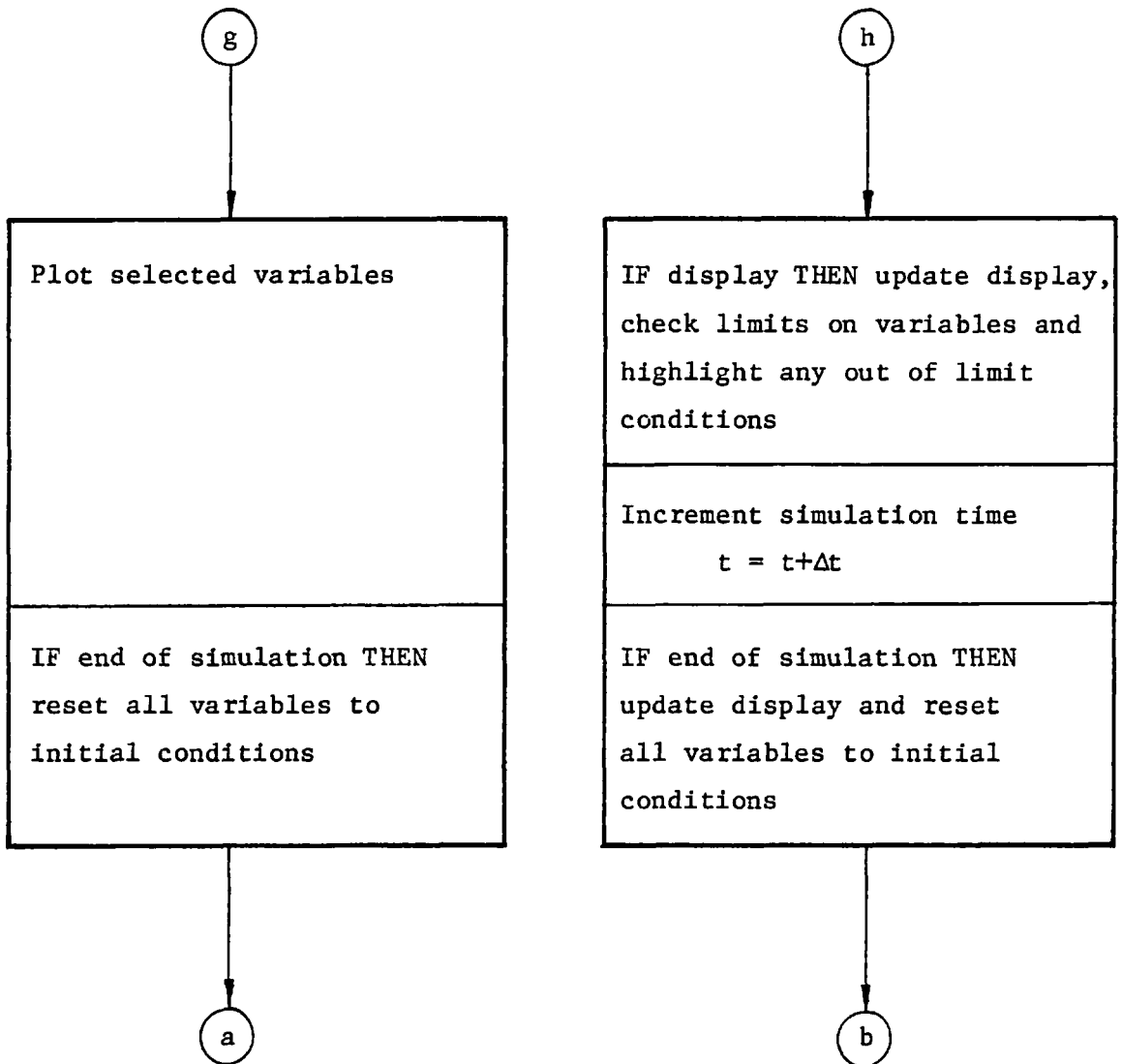


Fig. 7.4: Main Loop of the Transient Stability Simulation Algorithm

system [100], which has 10 generation units, 11 MPUs would be needed for the simulation. This represents more than twice the hardware available.

A parallel processing scheme that uses two levels of decomposition is proposed for the simulation of large power systems using 4 MPUs. The first decomposition level consists of dividing any power system into two subsystems. This decomposition can be performed using the tearing algorithm implemented in sec. 5.2 or using a given criterion. In the second decomposition level, each subsystem is divided into a transmission network area, loads and generation units. Then, two MPUs are used to represent each subsystem. The transmission network and loads of a subsystem are represented on one MPU and all the generators in that subsystem on the other. Fig. 7.5 depicts the configuration of microprocessors for the implementation of the proposed parallel processing scheme.

Few modifications are required in the algorithm shown in Figs. 7.2-7.4 to implement the parallel processing scheme for the simulation of bulk power systems using 4 MPUs. The number of real values to be transferred between MPUs in a subsystem is $2m$, where m is the number of generators in that subsystem. For salient generators $5m$ real values are transmitted from the MPU solving the generators. The diakoptic algorithms implemented in chapter 5 must be used to solve both network areas and to simulate the occurrence of disturbances. The interarea equations can be solved on any of the MPUs solving the transmission network areas.

When there are salient generators in just one of the areas, the iterative process needed to correct for saliency may be restricted to that area and after achieving convergence a complete solution of (7.18) is carried out to correct the voltages of both areas.

The algorithm implemented for the dynamic simulation of bulk power systems using 4 MPUs is shown in Fig. 7.6.

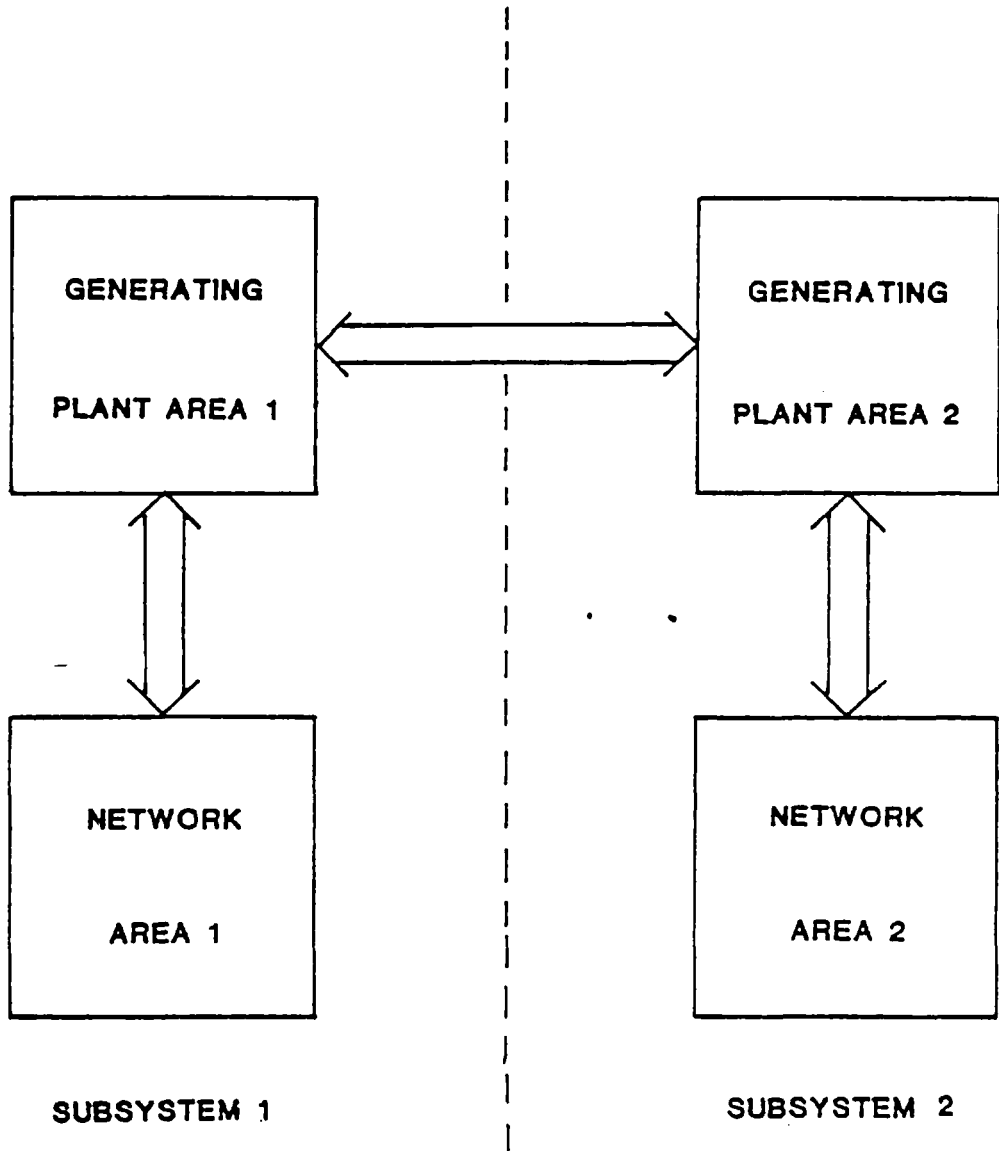


Fig. 7.5: Configuration of Microprocessors for the Simulation, of Bulk Power Systems

GENERATING PLANT 1

NETWORK AREA 1

NETWORK AREA 2

GENERATING PLANT 2

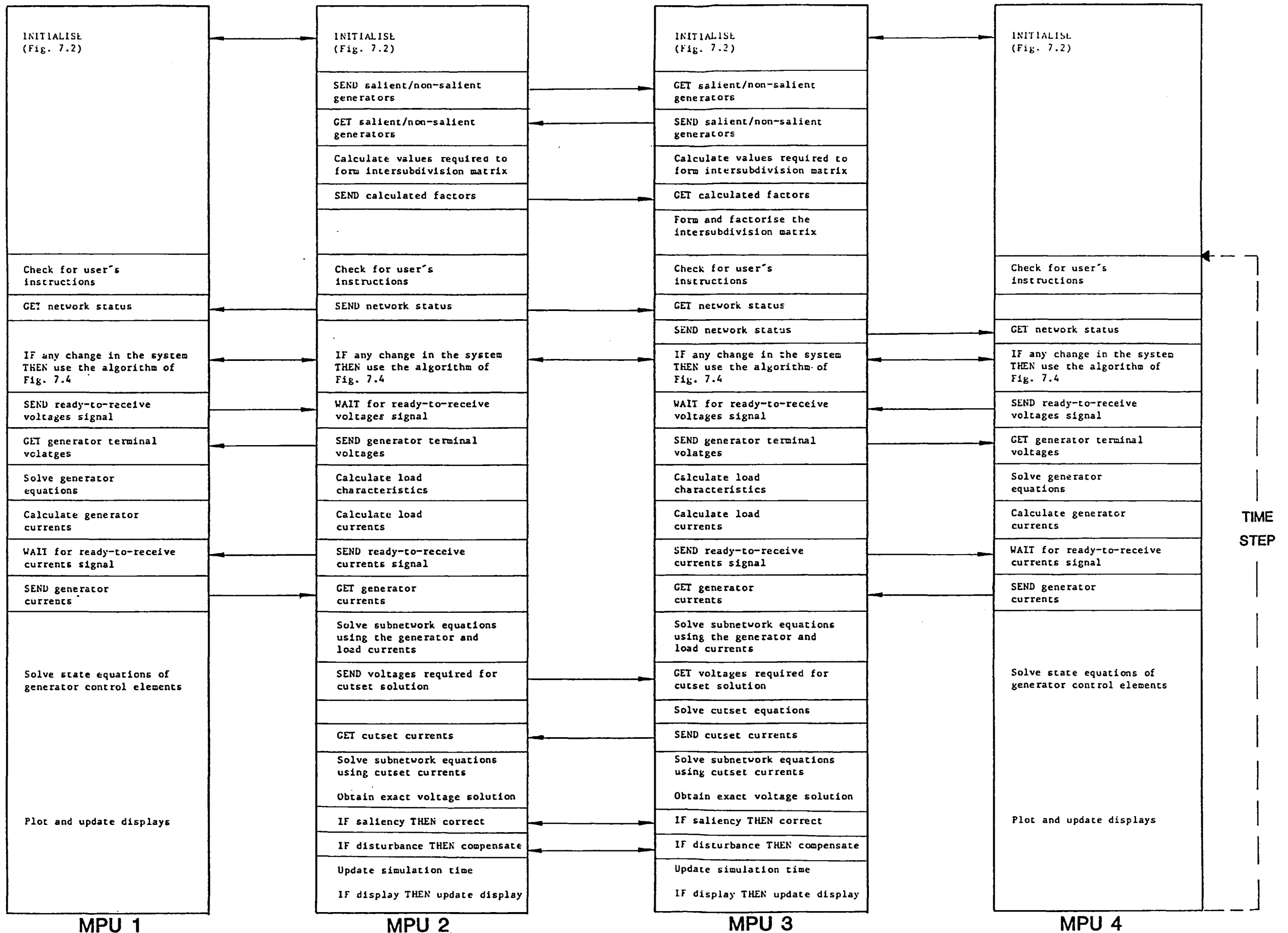


Fig. 7.6: Parallel Algorithm for the Dynamic Simulation of Bulk Power Systems Using 4 MPUs.

7.5 - TESTS AND RESULTS

Transient stability studies of different test systems have been carried out using the parallel processing schemes described in the previous sections.

The evaluation of the performance of 4 microprocessors for the solution of the transient stability problem of a 3 bus system with 2 generators and an induction motor is discussed in some detail in an early publication on this research work [105]. The speeding-up factor obtained with the 4 MPUs was 2.4 resulting in a 59% computational efficiency of the parallel simulation.

Two test systems are considered in this section to evaluate the proposed two level decomposition algorithm. The systems, which have been used by many investigators, are:

- . A 3 generation units, 9-bus system (WSCC test system)
- . A 10 generation units, 39-bus system which is the EPRI representation of the New England 345 kV bulk transmission system.

These systems have different characteristics in terms of generation units, transmission R/X ratios and load density. Data for the systems are given in the Appendices C and D.

Each of the generation plants was modeled in great detail using the following representations:

- . Generator - 4 state and 4 algebraic equations (two-axis model)
- . Excitation system - 3 state equations (IEEE Type DC1-DC commutator exciter)
- . Power system stabiliser - 3 state equations (Two stages of lead-lag compensation)
- . Governor-Turbine system - 4 state equations (IEEE Tandem compound, single reheat)

Specific data for the model parameters of each generation unit are given in the Appendix D.

The systems were divided into subsystems using the heuristic

clustering algorithm implemented in sec. 5.2. The 9-bus system was divided into subsystems of 5 buses with 2 generators and 4 buses with 1 generation unit. The 39-bus system was partitioned into subsystems of 19 and 20 buses with 5 generation units each. The decomposition of the test systems is illustrated in Figs. 7.7 and 7.8 respectively.

Transient stability studies were performed on these test systems simulating different types of faults in different locations in the systems. A time step of 0.02 s was used in the simulations. Fig. 7.9 shows the generator rotor angles with respect to the synchronously rotating network reference frame for the 9-bus system when the following sequence of events is simulated:

- . A three-phase fault is applied on the line 8-9 near bus 9
- . The fault is cleared after 100 ms by opening the faulted line
- . The line is successfully reclosed after 300 ms of the opening

Similarly, Figs. 7.10a and 7.10b show the generator rotor angles of the 39 bus system for a single-line-to-ground fault applied on the 26-29 line side of the 29 bus. A breaker's failure to operate was simulated and the faulted line was removed after 300 ms that the fault occurred. Line reclosure was not attempted.

One second of transient response for the 9-bus system is computed in about 47 seconds and for the 39-bus system in 131 seconds approximately. These times include fault occurrence and clearing but do not include output and initialization of the simulation which is done once for each system.

Timing diagrams of a simulation time step for the test systems are shown in Figs. 7.11 and 7.12 respectively. Table 7.1 gives total execution times and Table 7.2 presents the computational efficiencies and speeding-up factors obtained with the configuration of 4 MPUs implemented for the dynamic simulation of bulk power systems.

From the timing diagrams it can be seen that the plotting of swing curves for a time step takes longer than the computation itself. The execution times given in Figs. 7.11 and 7.12 may be used to measure the

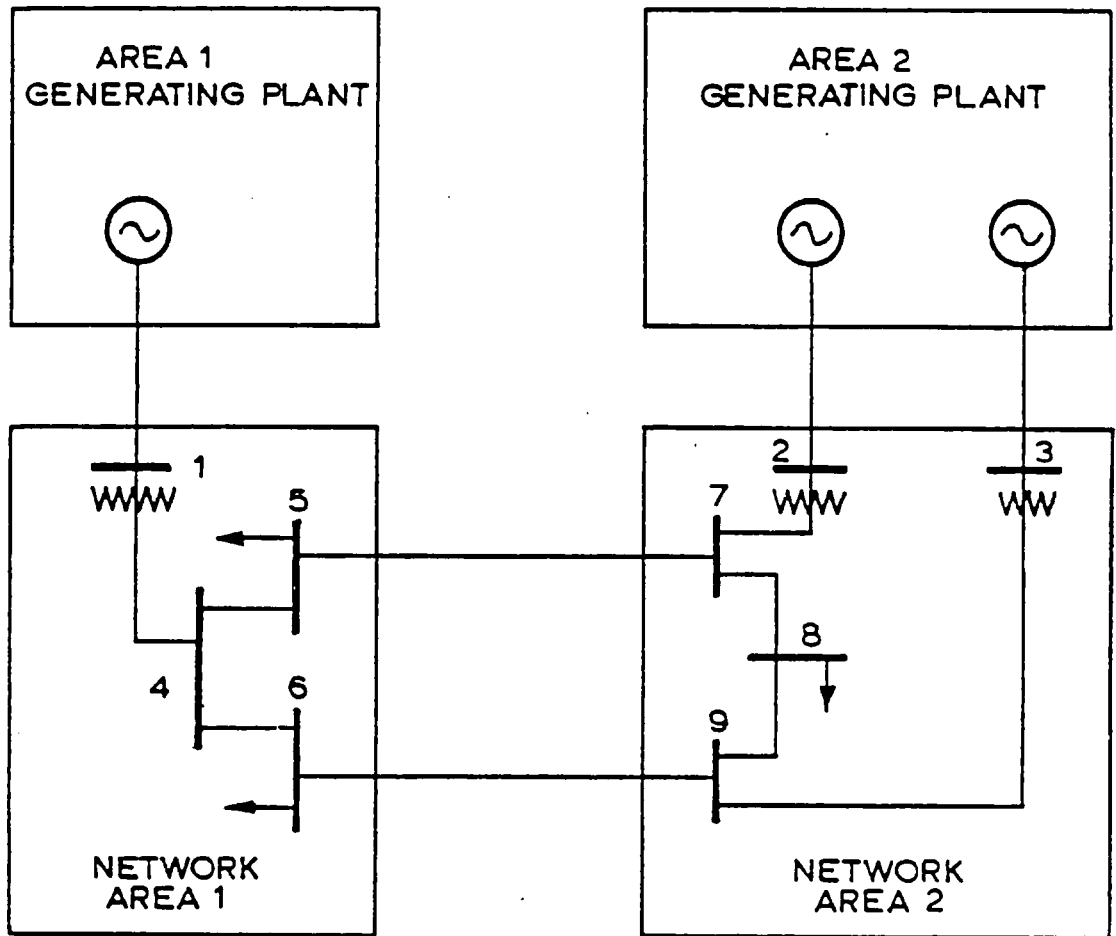


Fig. 7.7: Decomposition of the 9-bus, 3-generator Test System

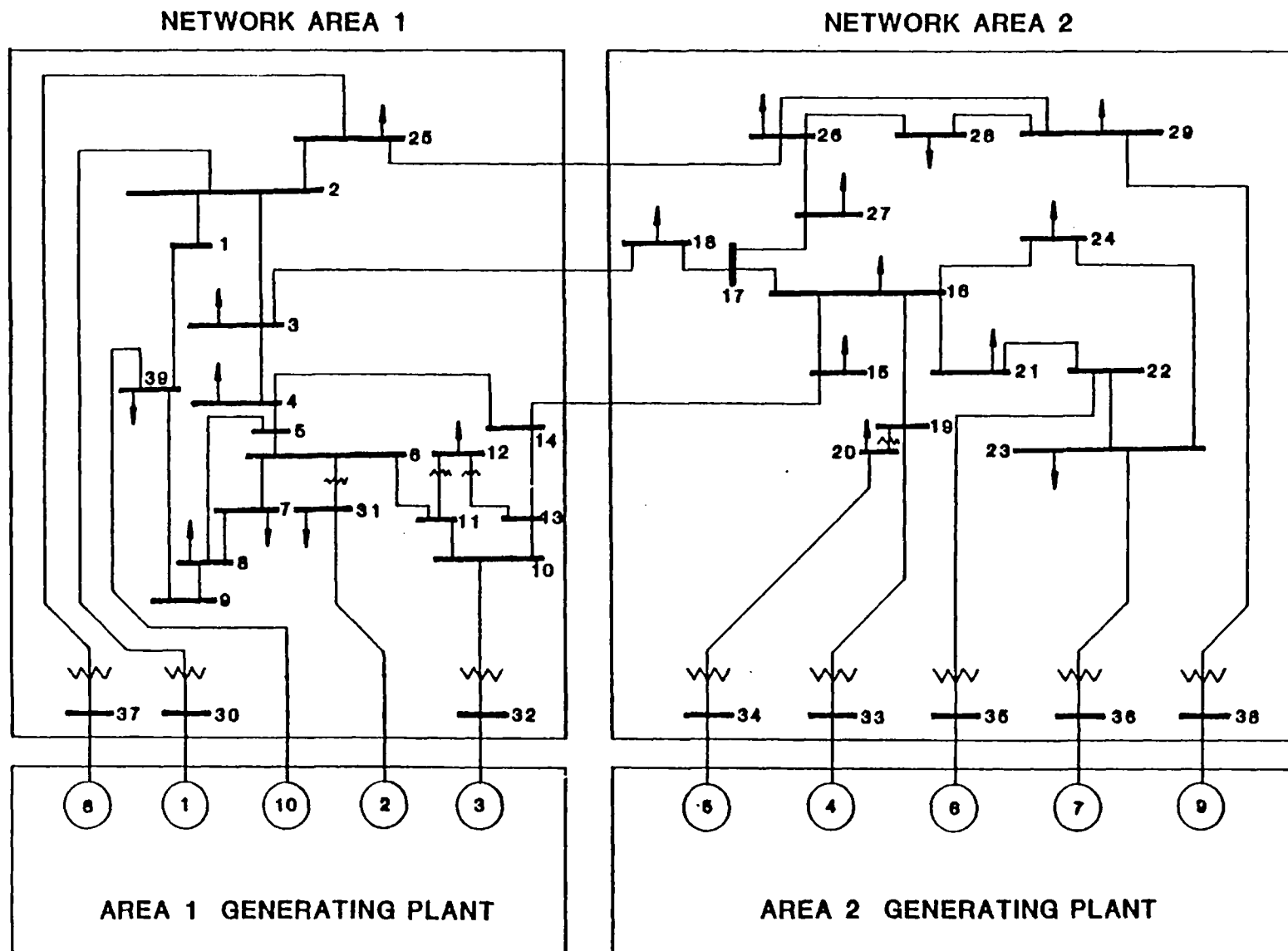


Fig. 7.8: Decomposition of the 39-bus, 10-generator Test System

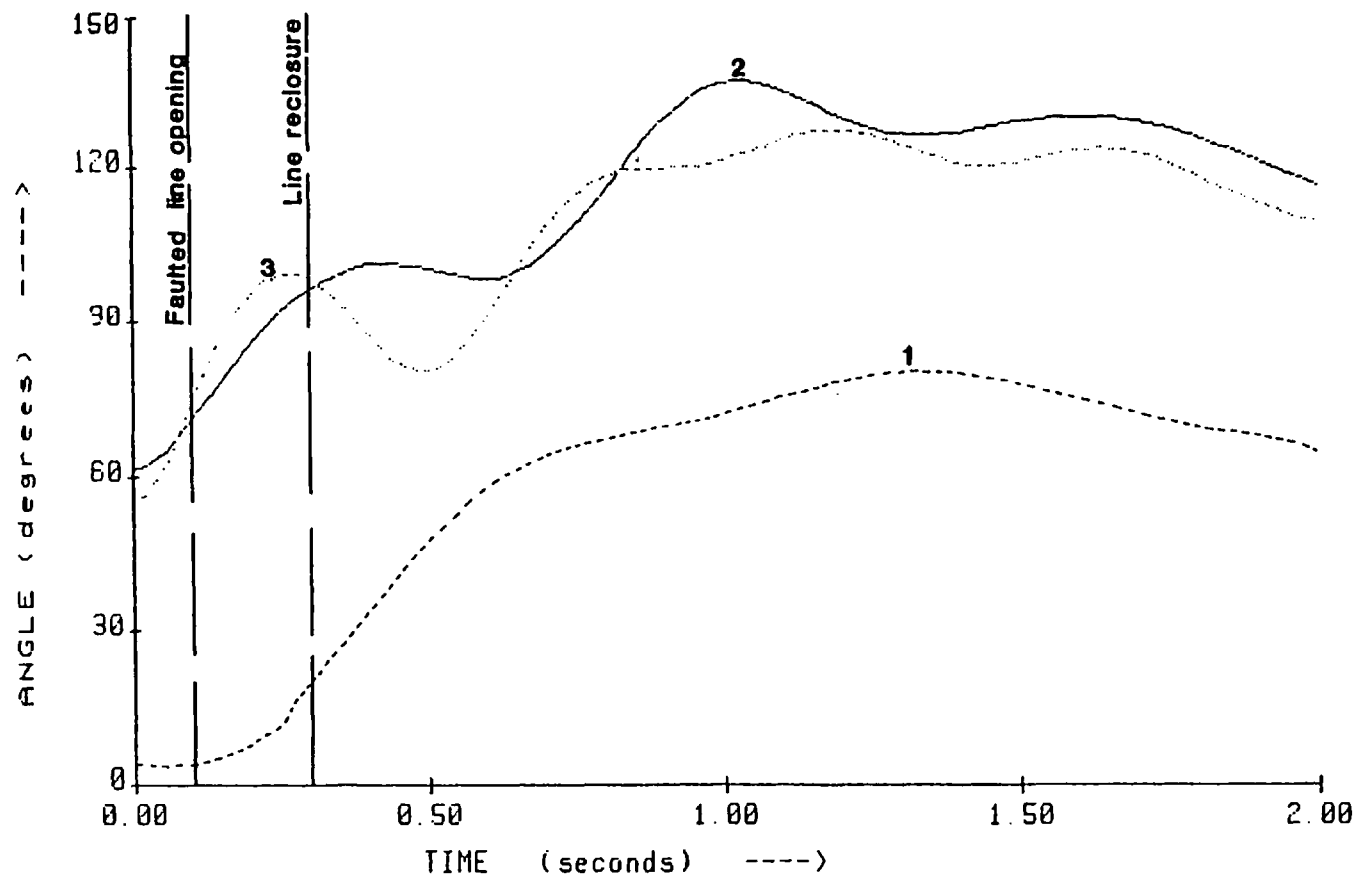
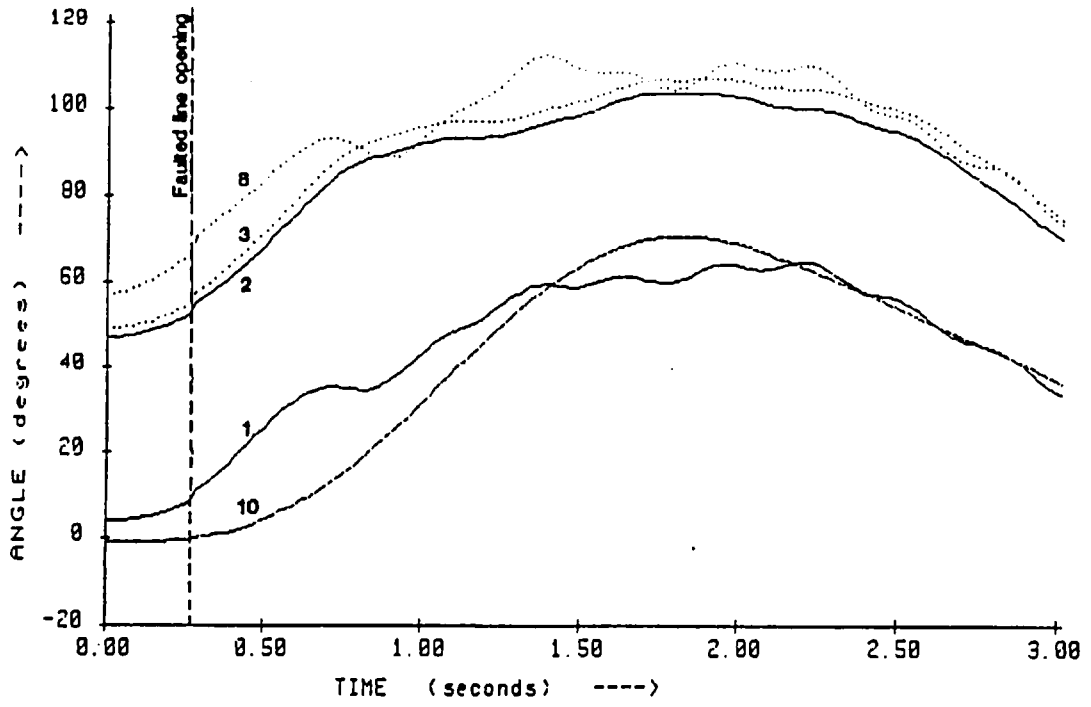
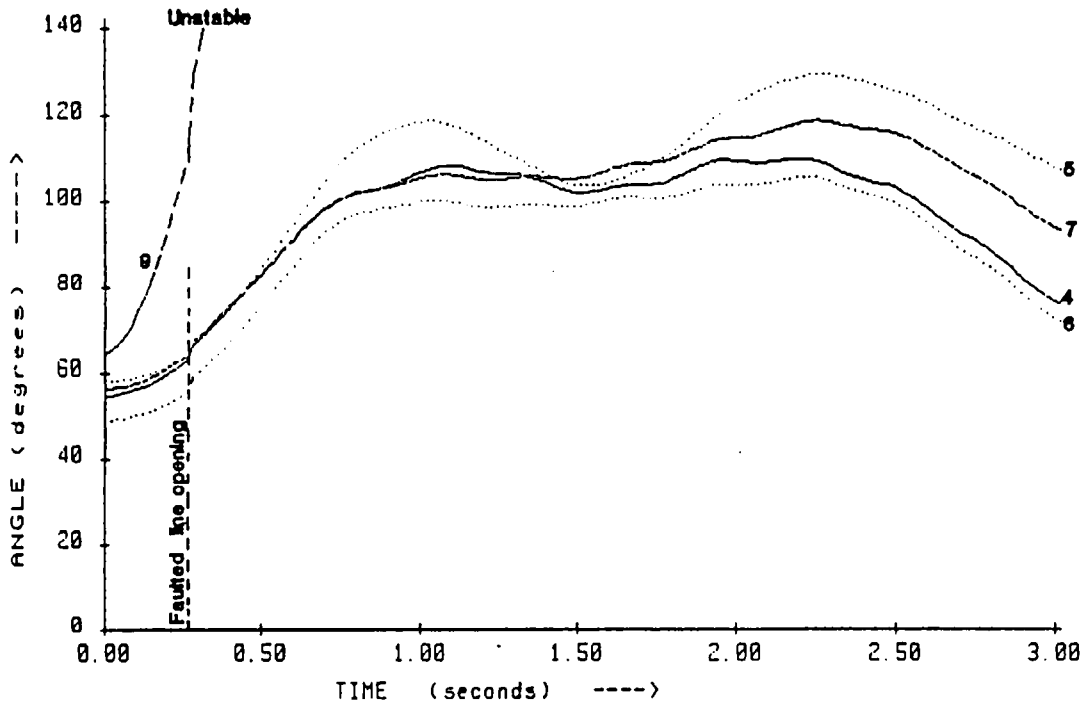


Fig. 7.9: Generator Rotor Angles for the 9-bus, 3-generator Test System. Three-phase fault on line 8-9 near bus 9 cleared by opening the line at 100 ms. The line is reclosed at 300 ms from the opening.



(a) Area 1 generators



(b) Area 2 generators

Fig. 7.10: Generator Rotor Angles for the 39-bus, 10-generator Test System. Single-line-to-ground fault on line 26-29 side of bus 29. The faulted line is removed after 300 ms that the fault occurred.

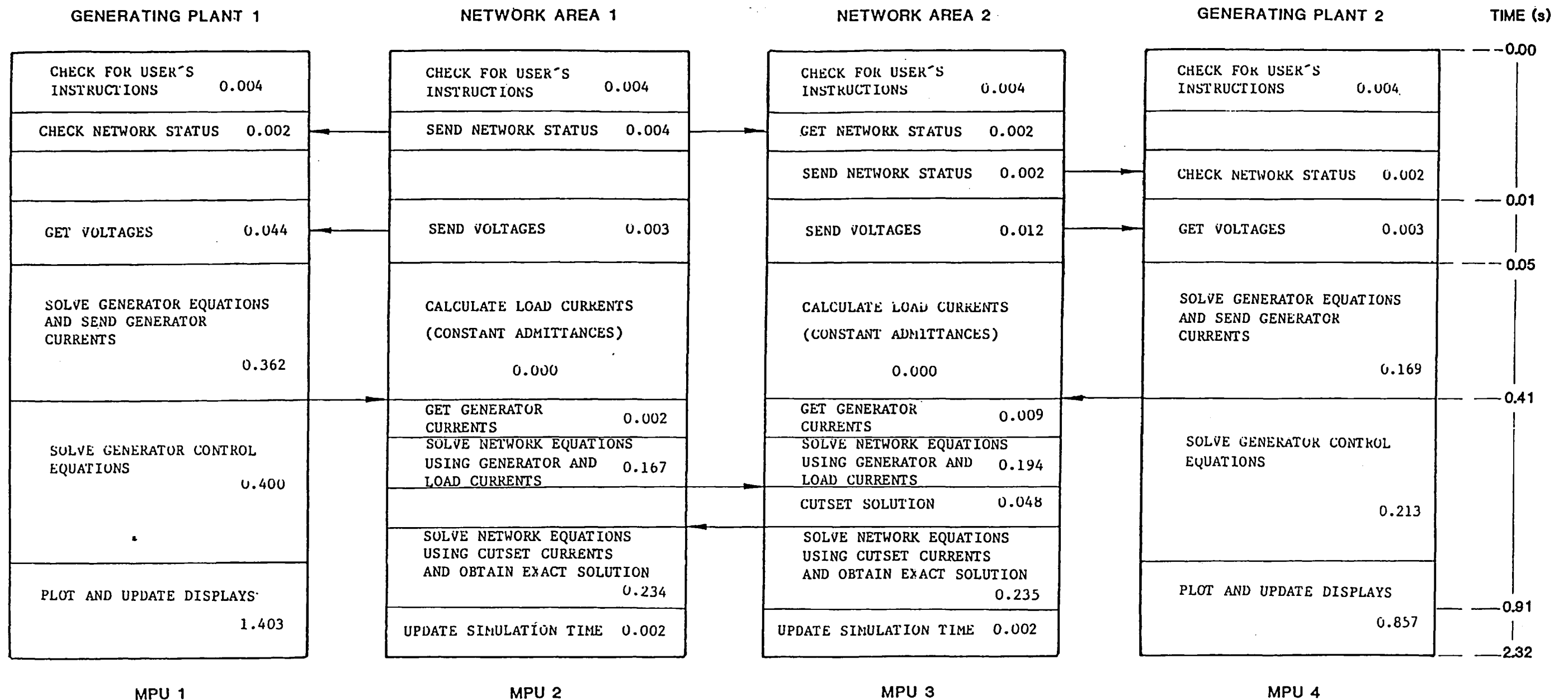


Fig. 7.11: Timing Diagram of a Simulation Time Step for the 9-bus, 3-generator Test System.

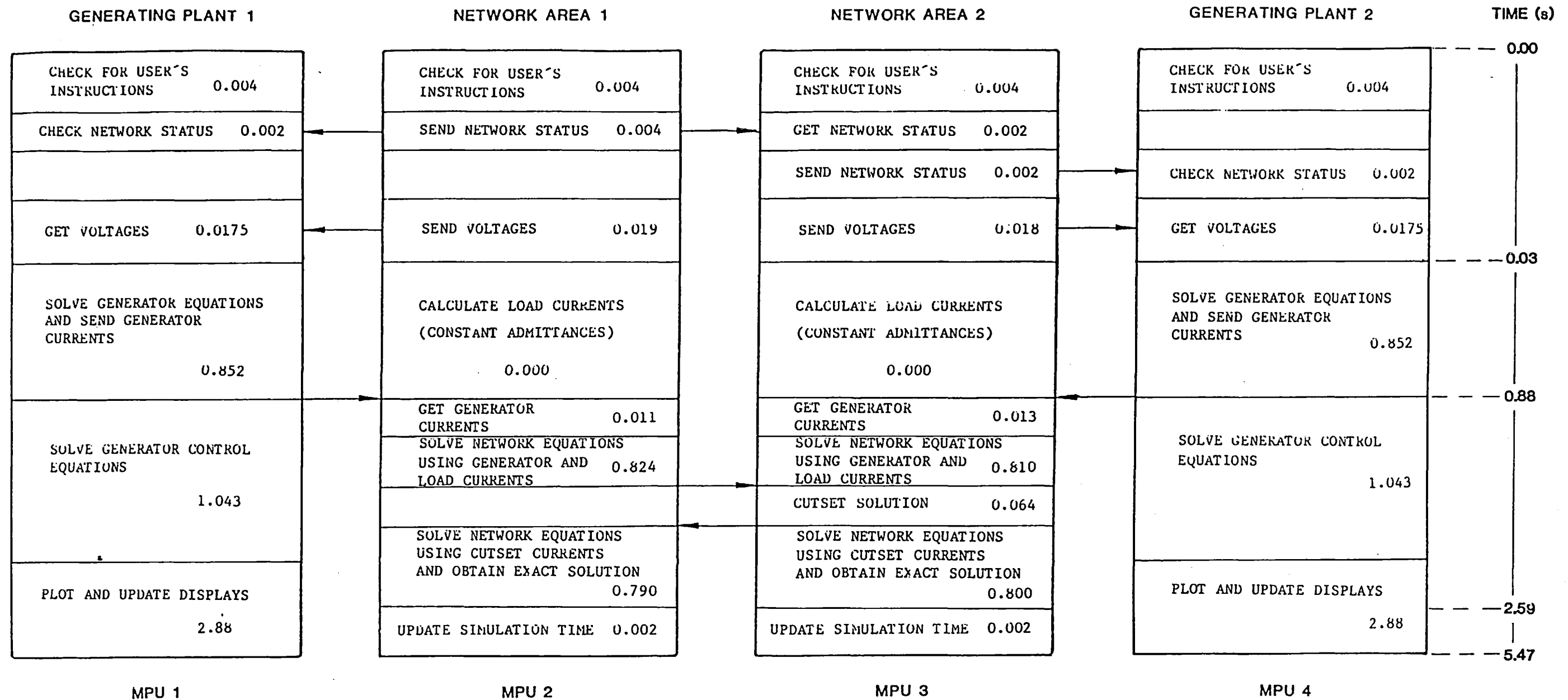


Fig. 7.12: Timing Diagram of a Simulation Time Step for the 39-bus, 10-generator Test System.

TABLE 7.1

Execution Times for the Dynamic Simulation
of Bulk Power Systems Using Four MPUs. (s)

MPUs	9-bus System			39-bus System		
	Active Time	Idle Time	Loading %	Active Time	Idle Time	Loading %
Generating Plant 1	0.390	0.520	43	1.920	0.670	74
Generating Plant 2	0.770	0.140	84	1.920	0.670	74
Network Area 1	0.416	0.498	46	1.650	0.930	64
Network Area 2	0.508	0.406	56	1.730	0.870	67

TABLE 7.2

Computational Efficiencies and Speeding-up
Factors for the Dynamic Simulation of Bulk
Power Systems Using Four MPUs.

	9-bus System	39-bus System
Simulation Time Step (Including Output)	2.32 s	5.47 s
Simulation Time Step (Excluding Output)	0.91 s	2.59 s
One Second of Simulation	47.00 s	131.00 s
Uniprocessor Time*	1.56 s	5.63 s
Speeding-up Factor	1.71	2.17
Computational Efficiency	42.75%	54.25%

* Estimated

parallelism achieved in the simulation and to estimate the advantages gained with the concurrent execution using multiple microprocessors. Excluding the time for results output, it is found that the MPU solving 2 of the generator plants of the 9-bus system is busy 84% of the time with respect to the global computation time required by the 4 MPUs used in the simulation, and that the MPU solving 1 generation plant is busy 43%. Similarly, the MPUs representing generators for the simulation of the 39-bus system are busy 74% while the ones solving the network are busy 64% and 67% of the time. In order to determine the overall computational efficiency of the configuration of MPUs used in the simulation, the execution time for the computation of a simulation step using a single MPU may be roughly estimated by adding to the times giving in the Table 4.2 for network direct solutions the computation times for the solution of the generation plant equations. Thus, the estimated times are 1.62 s for the 9-bus system and 5.69 s for the 39-bus system. Using the definitions given in [105] for speeding-up factors and computational efficiency, an speeding-up factor of 1.7 and an efficiency of 43% are obtained for the simulation of the 9-bus system using 4 MPUs. Similarly, for the 39-bus system the speeding-up factor is 2.17 and the corresponding efficiency is 54%. These figures show that the parallel processing scheme used for the simulation is more suitable for the 39-bus system than for the 9-bus system.

The information in Figs. 7.11 and 7.12 may be used in combination with the execution times given in chapters 4 and 5 for network solutions using one and multiple microprocessors to estimate the performance of different configurations of microprocessors for transient stability simulations. For example, if the 9-bus system had been simulated using the first approach proposed in which 1 MPU is employed to represent each generator in the system and 1 MPU to represent the network, the estimated speeding-up factor is 2.4 and the computational efficiency 60% which means that this approach is more suitable for the simulation of the 9-bus system than the proposed approach for the representation of large systems. Now, if 11 MPUs were employed for the simulation of the 39-bus system using 1 MPU to represent each generation unit and 1 MPU to represent the transmission network and loads, the estimated speeding-up factor is 2.1 and the corresponding computational efficiency is 19%. These figures tell us that the simulation of the 39-bus system using 11 MPUs would not result

in any gains and that it would be much more inefficient than the proposed configuration of 4 MPUs since many of the MPUs would be idle most of the time. In this manner it is possible to estimate the number of MPUs and the best configuration for the efficient simulation of a particular system.

7.6 - CONCLUSIONS

The implementation of parallel processing algorithms for the solution of the transient stability problem using multiple microprocessors has been discussed in this chapter. The parallel processing scheme implemented offers the capability for simulating in great detail large and complex power systems and with the developed algorithms the computation time is effectively reduced by factors between 1 and 3 when 4 microprocessors operating in parallel are used for transient stability solutions.

Computational results have been presented for the dynamic simulation of test systems using multiple microprocessors and it has been shown that for a given system there exists an optimum number of processors, configuration and task distribution for which the most economical simulation is obtained.

With the information provided in this and previous chapters it is possible to predict the performance of an existing configuration of processors for the simulation of a certain system or to determine the most suitable configuration of processors for that system.

- 181 -

CHAPTER EIGHT

MULTIPLE MICROPROCESSOR POWER SYSTEM SIMULATOR

8.0 - INTRODUCTION

The simulation of power systems with adequate detail and accuracy continues to be of fundamental concern to analysts, designers, operators and managers throughout the world. At the present time, attention is focused on finding the best ways to solve power system problems in a number of areas using modern computer hardware architectures such as microprocessors and array processors [5]-[16]. In a timely research project proposal [1] at Imperial College, Short and Cory launched an investigation on the utilization of microprocessor arrays for the construction of a flexible multi-purpose power system simulator.*

Following the guidelines stated in [1], a microprocessor-based power system simulator has been developed using a high level programming language (PASCAL) to write the simulator software and minimizing the utilization of special hardware by using off-the-shelf hardware wherever possible. Parallel processing techniques enhance the simulator by permitting a detailed analysis of large and complex systems at a modest cost and interactive dynamic facilities that greatly improve job turnaround are available to provide immediate answers to the user's question "what if.....?"

Some characteristics of the simulator are described in this chapter and various simulation studies are performed using a realistically complex test system to illustrate typical applications of the simulator.

* Throughout this thesis the term "power system simulator" refers to the software and hardware configuration of an array of microprocessors used for power system analysis.

8.1 - THE MULTIPLE MICROPROCESSOR POWER SYSTEM SIMULATOR

Presently available power system simulators have the capability for modelling, in real time, the long-term dynamic behaviour of systems with several hundred buses [3]-[4]. However, there is an increasing need to include transient stability phenomena as a study option since system security is becoming a more and more important issue in the viability of electric power systems. Existing production programs for transient stability, e.g. Bonneville Power Administration (BPA) program, Philadelphia Electric Co. (PECO) program, Arizona Public Service (APS) program, etc., require a large amount of computer storage and in general do not provide interactive dynamic simulations. Recently, ESCA Corporation developed a unified short/long-term simulation program for EPRI [17], using a DEC VAX 11/780 computer, which required 500 Kbytes of main memory and 1.5 Megabytes of virtual memory for a 100 bus system base case. Nevertheless, Shulz, one of the authors of the long-term dynamic part of the program designated as LOTDYS (see discussion to [17]), argued that a simulation with both the short and long term simulation modes in the same program will be more expensive than one with a single simulation mode. Despite the significant advance achieved with the unified program for short/long-term dynamics the quest for computational tools to provide accurate and economical simulation studies continues.

The availability of complete processors as low-cost building blocks now makes economically feasible the application of multiple processors to reduce the computation burden of complex power system analysis. A power system simulator has been developed as a user-oriented tool using multiple microprocessors to meet the demand for computational tools which provide detailed simulations of bulk power systems at a fraction of present day cost and execution time. The simulator can be used as an effective aid for system analysis in planning, design and operational studies where numerous runs are conducted for a variety of system conditions, e.g. network configuration, equipment outages, etc., to determine how stable or unstable the system is for specified disturbances.

The system analysis studies implemented for the simulator are:

- . Load flow analysis
- . Fault analysis
- . Transient stability analysis

Parallel to this work, research has been undertaken for the implementation of long-term dynamic studies [101] and real-time simulation for operator training in load scheduling.

8.1.1 - Simulator Features

The power system simulator comprises detailed models of power system components for short/mid term dynamics, a set of typical disturbances, a series of control actions and a data base.

The mathematical models available for the power system componenets are:

- Transmission network

- . Equivalent pi-circuit for transmission lines
- . Tap changing transformers
- . Shunt reactors/capacitors

- Generation plant

- . Generator:- Two axis model including
 - Saliency
 - Damping
 - Saturation
- . Excitation systems:- IEEE excitation system models
- . Power system stabiliser:- Two stages of lead-lag compensation without high frequency filters represented
- . Speed-Governing systems
 - Mechanical-hydraulic control system
 - Electro-hydraulic control system

- . Turbines

- Steam

- Non-reheat

- Tandem compound single reheat

- Fast turbine valving

- Hydro

- Electric load

- . Constant admittance

- . Exponential function of bus voltage and frequency

For details on the modelling see chapter 2. An option has been included in the representation of generation plants to allow for the inclusion in the model of only those generator control elements whose effects are desired in the representation.

The disturbances and control actions that can be simulated are:

- Transmission network

- . Limit checking of key variables

- . Symmetrical and unsymmetrical faults

- Single-line-to-ground fault

- Line-to-line fault

- Double-line-to-ground fault

- Three-phase fault

- . Fault clearing with or without line reclosure

- . Line outage/reconnection

- . Transformer outage/reconnection

- . Shunt reactor/capacitor switching

- Generation plant

- . Generator disconnection/reconnection

- . Fast turbine valving

- . Modification of the AVR reference voltage
- . Alteration of scheduled generation power

- Electric load

- . Disconnection/reconnection
- . Change in load demand
- . Change in load characteristic

The different events associated with the system components have been implemented as manual actions to be initiated by the user interactively with the simulator. The initialization of events in the transmission network and loads have been centralised as a function of the power system control centre, see sec. 2.7.

The data base comprises data for the following test systems:

- IEEE test systems
 - . 14-bus, 2-generator
 - . 24-bus, 10-generator
 - . 30-bus, 2-generator
 - . 57-bus, 4-generator
 - . 118-bus, 54-generator
- 3-bus, 2-generator system
- 5-bus, 2-generator system
- 9-bus, 3-generator WSCC test system
- 39-bus, 10-generator New England system

The data for some of the test systems are given in the Appendices C and D.

8.1.2 - Simulator Architecture

The multiple microprocessor power system simulator was developed following the guide lines by Short and Cory in [1]. Three of the main guidelines stated in [1] for the construction of the simulator are:

- . To implement a simulator architecture and software that is modular, flexible and readily expandable
- . To minimize special hardware construction by using off-the-shelf hardware wherever possible
- . To use high level programming languages so that simulator software is hardware independent

The architecture of the simulator was tailored to closely parallel the actual power system configuration. To represent bulk power systems using microprocessors, a power system is decomposed into subsystems and a system control centre. Each subsystem is further decomposed into generation plants, transmission network and electric loads. To make optimum use of the hardware available (5MPUs) the systems are decomposed into two subsystems. One MPU is then used to represent the control centre and 2 MPUs to represent each subsystem. One of the MPUs representing a subsystem is used to represent the transmission network and loads, and the other to represent all generation plants in that subsystem. A schematic diagram showing the architecture of the simulator developed for the representation of bulk power systems using 5 MPUs is illustrated in Fig. 8.1.

The coordination of MPUs during the simulation is provided by the parallel processing algorithms implemented for the simulator which have been discussed thoroughly in previous chapters. The MPU representing the control centre is not directly connected to the MPUs representing generation units. Any interchange of information between these MPUs is performed through the MPUs representing network areas which receive and pass the messages.

Each MPU is composed of the TMS9900, 16-bit microprocessor as CPU and a 64 Kbytes memory expansion board. The 5 MPUs are cabled permanently through one of their serial I/O RS-232C ports to a programmable control switch which links the microprocessors with the

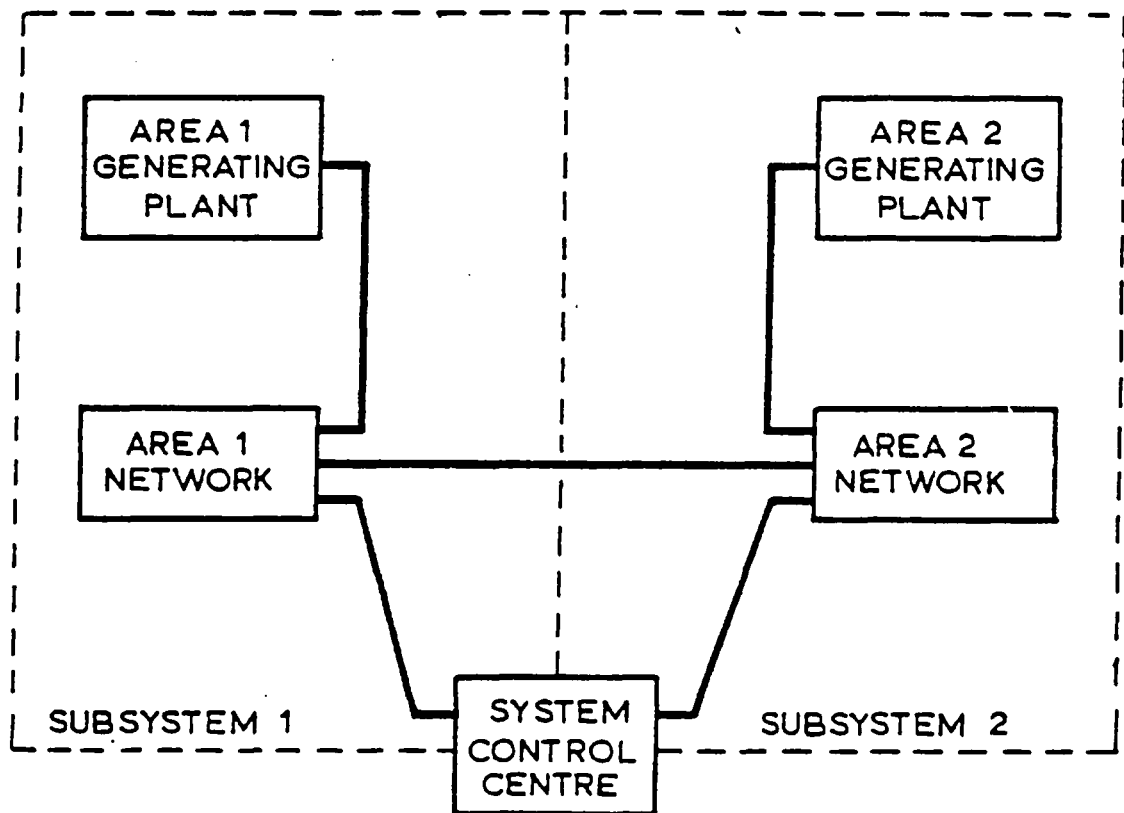


Fig. 8.1: Representation of Bulk power Systems Using Five Microprocessor Units.

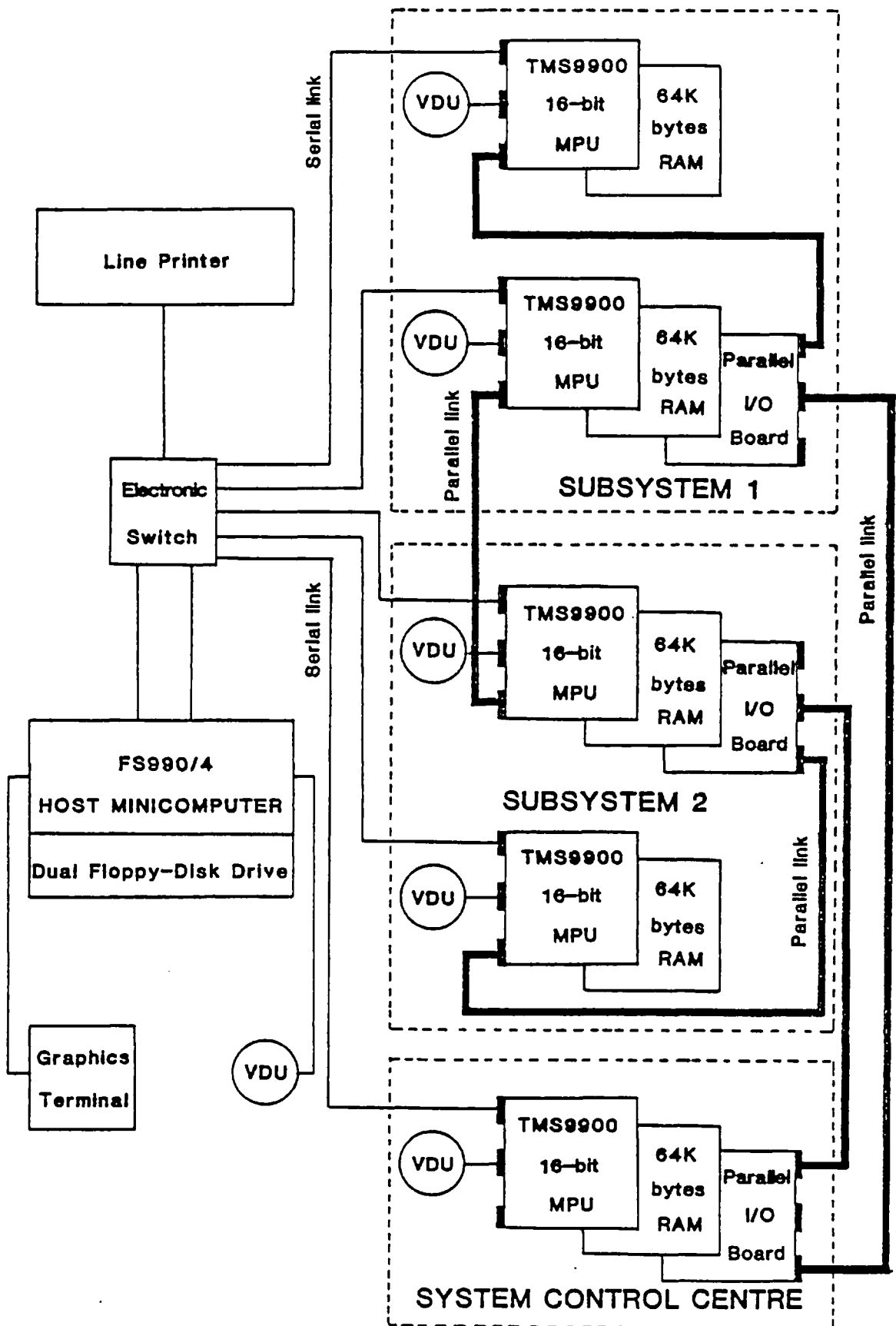
support minicomputer TI-FS990/4. The minicomputer can communicate with any of the 5 MPUs through the programable switch and it is used to provide a versatile interface between user and simulator. The interface program enables the control, monitoring and on-line interaction with the simulator. A user sitting in front of the console of the minicomputer can monitor from there every single microprocessor. He can halt, resume or restart the simulation and he can simulate changes in the power system at any point during the simulation. Fig. 8.2 depicts the hardware configuration of the simulator and support devices. The support devices include VDUs for each MPU, a printer, a graphics terminal and the host minicomputer. All the simulator software and the data base are stored on floppy disks in the disk units of the minicomputer.

The software for both minicomputer and simulator are completely independent of each other. Modifications of the simulator software do not affect the interface program, and vice versa. The minicomputer/simulator communication is performed by messages. To communicate a message to a certain MPU, the minicomputer selects the destination MPU on the programable switch and places the message into a predefined communication buffer on that MPU. During the simulation, the MPUs check the communication buffers for any messages from the minicomputer and send back an acknowledgment message to the minicomputer confirming the received instructions and then proceed to execute the appropriate action. Details on communication protocols can be found in the simulator documentation [109].

8.1.3 - User/Simulator Interactive Interface

The interactive environment of the multiple microprocessor power system simulator is provided by a control program via the host minicomputer FS990/4. A user can initiate one of several possible actions at his disposal by pressing command keys. The commands available may be broadly divided into three main groups according to their functions:

- Control and initialisation commands. These commands allow the user to set up and control the operation of the simulator. Commands are included to download programs, send data, reset,



INTERACTIVE MULTIPLE MICROCOMPUTER SIMULATOR : HARDWARE STRUCTURE

Fig. 8.2: Hardware Configuration of the Microprocessor-Based Power System Simulator.

halt and select the mode of operation of the simulator.

- . Utility Commands. These commands include various useful routines for file handling. A pre-edited command file may be used to initialise and control a complete simulation session without user intervention. Facilities are provided that allow the user to produce high resolution graphics and store results in a data file for a later analysis.
- . Interactive commands. These enable the user to interact dynamically with the simulator. The user can alter the values of variables and set different events as the simulation progresses by using the commands:

MONIT - Selects the communication port of a specified MPU

MODIFY - Sends the new value of a modifiable variable

EVENT - Sends the parameters of a user requested event

Control, initialization and utility commands are available for each MPU. The interactive commands are associated with the actions that the user is allowed to simulate on each MPU. For example, the interactive commands for MPUs representing generators are the disconnection/reconnection of a generator, application of fast turbine valving, modification of the reference voltage in the AVR and the alteration of the scheduled generation power. The last two actions are initiated with the command MODIFY and the first two with EVENT. The actions initiated with MODIFY are immediately executed whereas actions with the EVENT command need further specifications which are given to the simulator program. The interaction with MPUs representing the transmission network and loads is centralised on the MPU representing the system control centre. In addition, commands have been included in this MPU to restart a new simulation study on the same system without reloading data and to select the power plant controls whose representation is desired in the modelling for the new study.

For each MPU a display page has been formatted according to the power system component represented, e.g. generators, network area, etc. The displays show the status, the current simulation time, the values of scheduled variables and the menu of commands available on a particular MPU. The user can interact with the simulator by selecting commands as required from the menu displayed which labels the function

of the command keys. For instance, to initiate an event the user selects the MPU where the event is to be simulated by pressing the key-MONIT (monitor) followed by the key identifying the MPU (network or generator) to be selected. Having done so, the display, of the selected MPU appears on the screen. Then he presses the key-EVENT and an immediate response appears on the screen as soon as the command is executed reconfiguring the function keys to show the events available. On selection of the event the user is prompted to enter the event parameters, e.g., for a fault, type, near bus, far bus and duration. An acknowledgement message from the MPU confirming the received instructions appears on the display as soon as the action is taken and an audible alarm is sounded on the VDU connected to the MPU to signal the operator that an event is on. The audible alarm goes off when the event is cancelled. The displays for MPUs representing generators and a transmission network area are illustrated in Figs. 8.3 and 8.4 respectively.

For a detailed description of the commands available and examples of typical simulation sessions see the interactive power system simulator user's manual [109].

8.2 - SIMULATOR APPLICATIONS

The EPRI representation of the New England 345 kV bulk transmission system is used in this section to illustrate typical applications of the multiple microprocessor power system simulator in a range of power system analysis studies. The system has been divided into two subsystems as shown in Fig. 7.7. One subsystem is made up of 19 buses and 5 generation units with a total initial generation power of 3003 MW and an initial load of 2924 MW. The second subsystem contains 19 buses and 5 generation units with a total initial generation of 3180 MW and an initial load of 2897 MW. One of the 10 generators (unit 10) represents parts of the U.S.-Canada interconnected system. The data have been expanded from the EPRI data in order to incorporate in each generation plant the representation of excitation systems, power system stabilisers and governor-turbine systems. The loads were modelled as constant admittances and a simulation time step of 0.02 s was used in all the cases. The data used for each generation plant are given in the Appendix D.

POWER SYSTEM DYNAMIC SIMULATOR

STATUS OF GENERATORS IN AREA NO. 2

GENERATOR PARAMETERS

GENERATOR	ANGLE	FREQUENCY	ELEC. MW	MECH. MW	AVR VOLT	VOLTAGE
UNIT1	78.90	59.93	621.19	632.01	2.61	1.01
UNIT2	107.19	59.68	449.45	506.40	3.06	0.99
UNIT3	68.61	59.91	734.91	651.49	2.90	1.00
UNIT4	79.80	60.01	643.84	560.28	0.90	1.03
UNIT5	94.47	59.99	773.27	830.67	2.12	1.02

PLOTVAR:-

78.90 107.19 68.61 79.80 94.47

MESSAGE:- Change of Scheduled Power in Unit 3

SIMULATION TIME : 3.34

UNIT1

UNIT2

UNIT3

UNIT4

UNIT5

4

5

6

7

9

ENTER UNIT ID FOR DISCONNECTION

F1-DISCON F2-RECON F3-FASTV F4-AVREF F5-SCH.POW

F1

F2

F3

F4

F5

F6

F7

F8

CMD

Fig. 8.3: Display Page for an MPU Representing Generators

POWER SYSTEM DYNAMIC SIMULATOR

NETWORK AREA NO. 1 STATUS

BUS BAR VOLTAGES							
BUS	MAG.	ANGLE	STAT	BUS	MAG.	ANGLE	STAT
1	1.06	6.86		2	1.09	6.03	
3	1.06	2.76		4	1.21	2.58	*HI*
5	1.04	22.43		6	1.22	5.16	*HI*
7	1.03	3.23		8	1.03	2.26	
9	1.00	5.52		10	1.06	7.21	
11	0.75	6.50	*LO*	12	1.04	6.34	
13	0.78	6.31	*LO*	14	1.05	4.17	
25	0.92	7.00		30	1.08	8.24	
31	1.01	14.03		32	1.02	15.62	
37	1.04	13.21		39	1.03	7.40	

SIMULATION TIME : 3.34

MESSAGE : Line 4 14 Disconnected

ON/OFF

NEARBUS

FARBUS

MODELS

FAULT

OUTAGE

LOADS

CAP.SWITCH

F1

F2

F3

F4

F5

F6

F7

F8

CMD

Fig. 8.4: Display Page for an MPU Representing a Network Area.

8.2.1 - System Planning and Design Studies

An extensive number of transient stability simulations are routinely required by planning and design engineers to determine how stable or unstable the system is for specified severe disturbances.

The microprocessor-based simulator may be used as an effective tool to determine the system stability to various disturbance conditions. Also, the simulator can be used to obtain information on the sensitivity of the system stability with respect to changes in certain parameters for example:- type of fault, fault location, fault clearing time, etc.

8.2.1.1 - Example

In power system planning and design the critical fault clearing time has always been considered as a measure of the system transient stability properties, Kimbark [23].

The "remote generation" unit 9 which is loosely coupled to the system through the transmission lines 26-29 and 28-29 is chosen to perform critical clearing time studies for fault conditions. A three-phase fault is applied on the line 26-29 side of the bus 29 and different fault clearing times are considered to analyse the stability behaviour of the test generator.

The rotor angles with respect to the synchronously rotating network reference frame of the generators in the faulted area are shown in Fig. 8.5 for a fault clearing time of 100 ms. After a large initial oscillation ($\approx 150^\circ$) the generator 9 stabilises and swings together with the other area generators. However, for a fault clearing time of 120 ms the generator 9 goes unstable as shown in Fig. 8.6. The rest of the generators in the system remain stable for both cases.

8.2.2 - System Operation

Transient stability studies are currently performed off-line in system operation studies for a wide range of operating conditions and postulated outages to determine whether such disturbances will cause transient instability in the system. The only on-line application

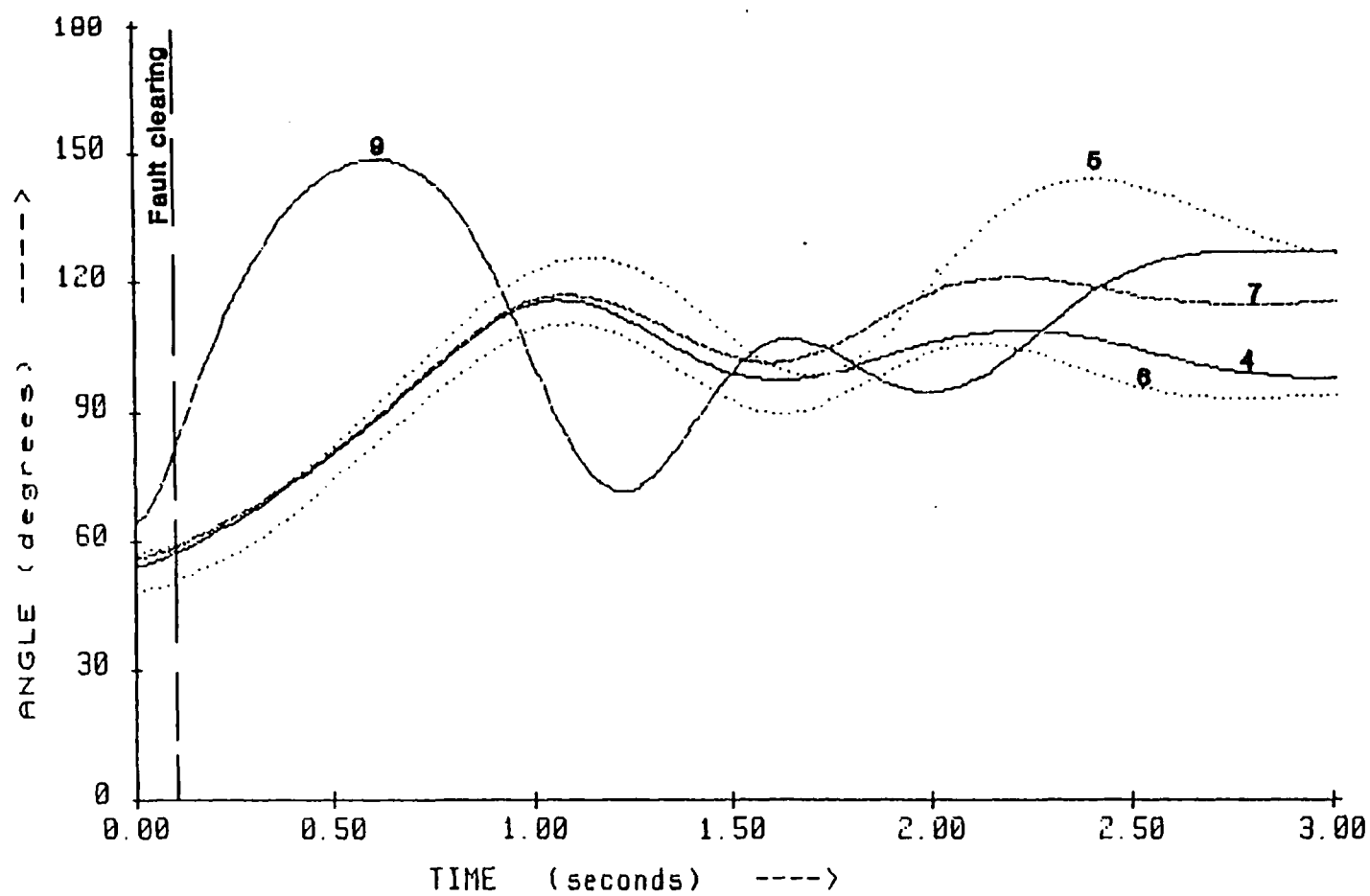


Fig. 8.5: Rotor Angles for the Generators in Area 2. Three-phase fault on line 26-29 side of bus 29 cleared at 100 ms.

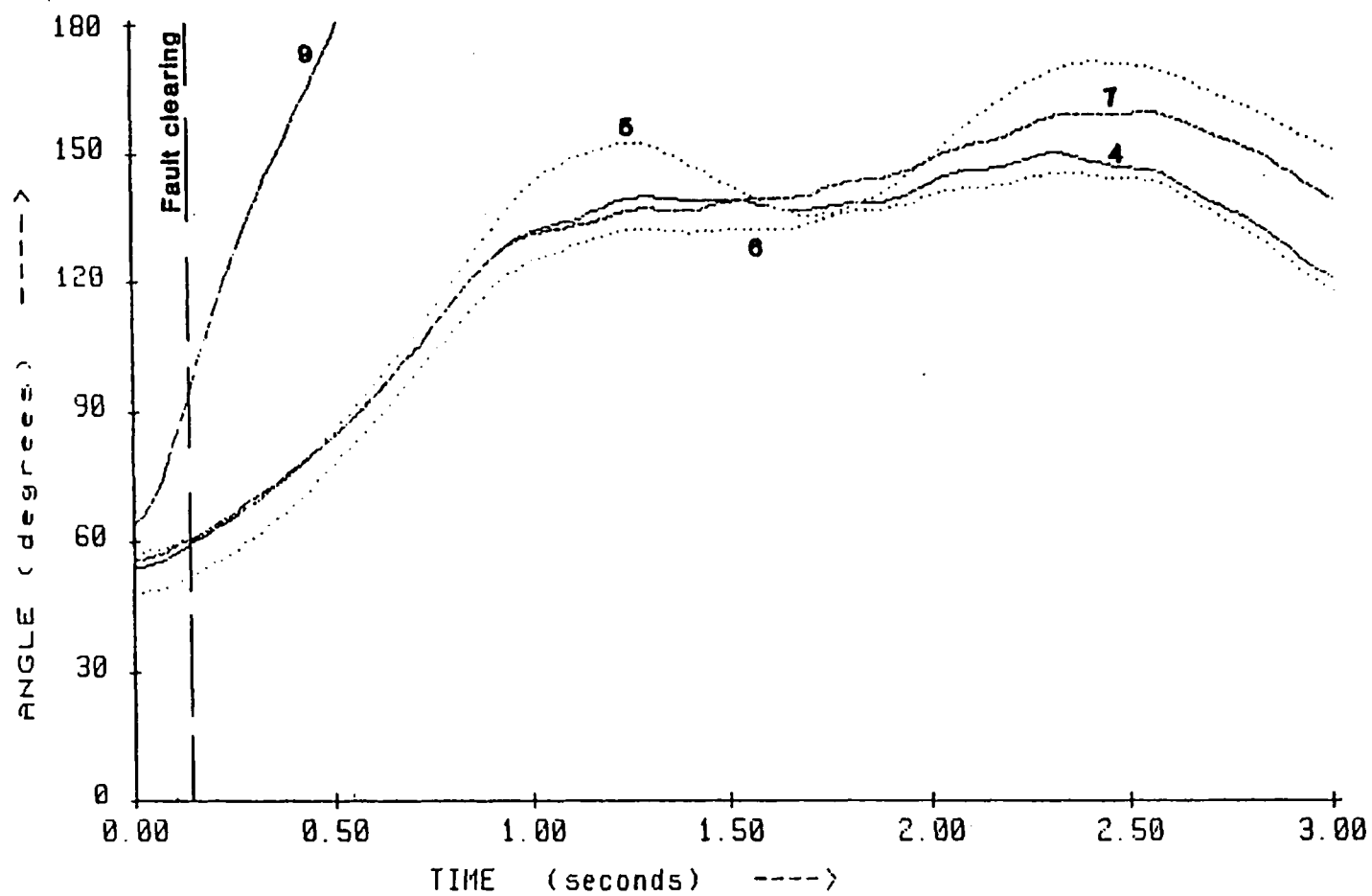


Fig. 8.6: Rotor Angles for the Generators in Area 2. Three-phase fault on line 26-29 side of bus 29 cleared at 120 ms.

which has been implemented appears to be that of the Tokyo Electric Power Company [106].

An operator is typically interested in studying system conditions for up to several hours in advance for the purpose of assessing the impact of future loading conditions and/or generator and transmission outages. The operator could be significantly assisted by the use of a practical tool such as the power system simulator to assess the security of the system and to evaluate preventive and corrective measures for future system conditions.

8.2.2.1 - Example

The case considered to illustrate the application of the simulator in system operation studies involves an assumed sudden loss of the interconnection between the New England area and the Canadian system at a time when the New England area is importing 1000 MW. The loss of the tie was simulated by disconnecting the generation unit 10 which represents parts of the U.S.-Canada interconnected system. The sudden loss of the imported generation results in a generation deficiency in the New England area which causes a decline of the frequency, as shown in Figs. 8.7 and 8.8.

The resulting angular displacements and the electrical power of the generators are plotted in Fig. 8.9 through 8.12.

The general pattern shown by the generator angle curves is that the angles tend to retard with time and eventually will settle down to some new steady state value. The loss of the imported power is temporarily made up by the stored energy in all the system generators. The electrical power curves in Figs. 8.11 and 8.12 show the instantaneous increase in the power output of the generators to supply the generation deficiency in the system. The frequency shows an initial recovery but continues dropping until it reaches a new value of around 59.8 Hz.

If the new system steady state conditions violate some constraints established by the philosophy of operation, the operator should take corrective measures to restore the system within acceptable operational margins.

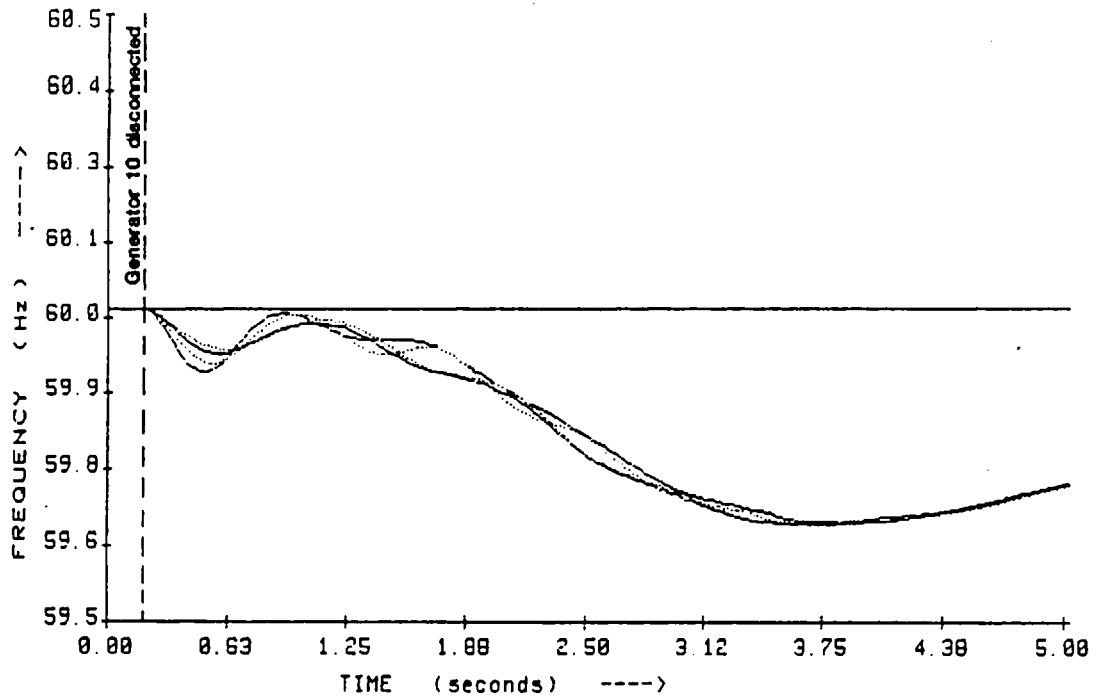


Fig. 8.7: Generator Frequencies in the System Area 1.
Sudden loss of the generation unit 10.

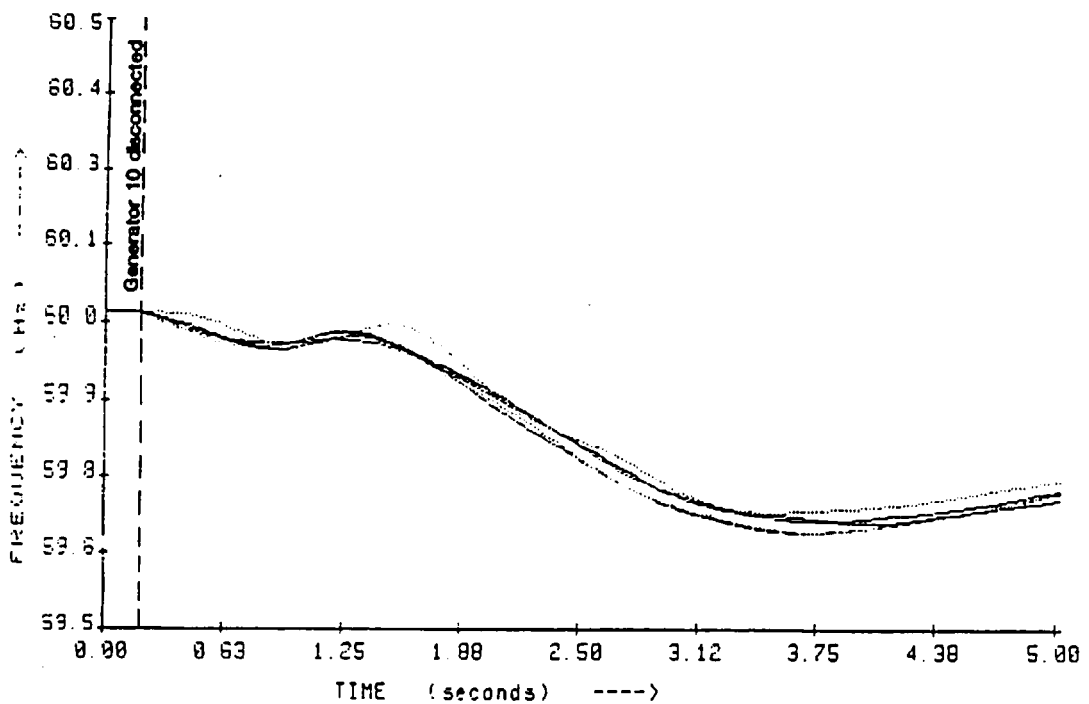


Fig. 8.8: Generator Frequencies in the System Area 2.
Sudden loss of the generation unit 10.

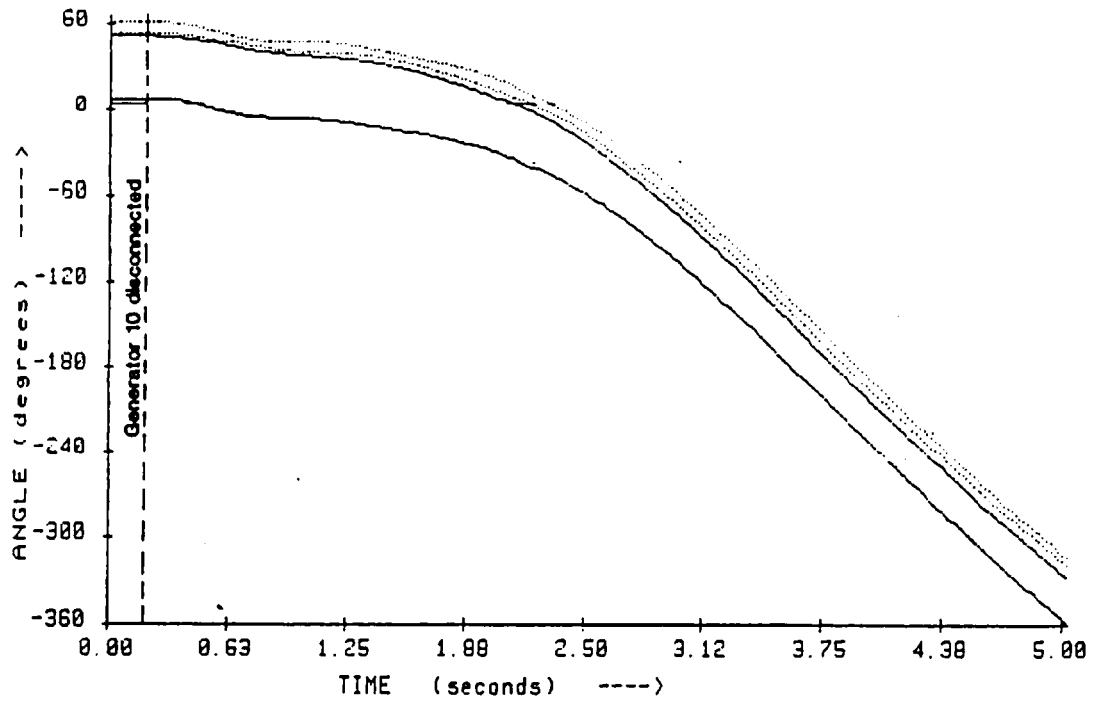


Fig. 8.9: Generator Rotor Angles in the System Area 1.
Sudden loss of the generation unit 10.

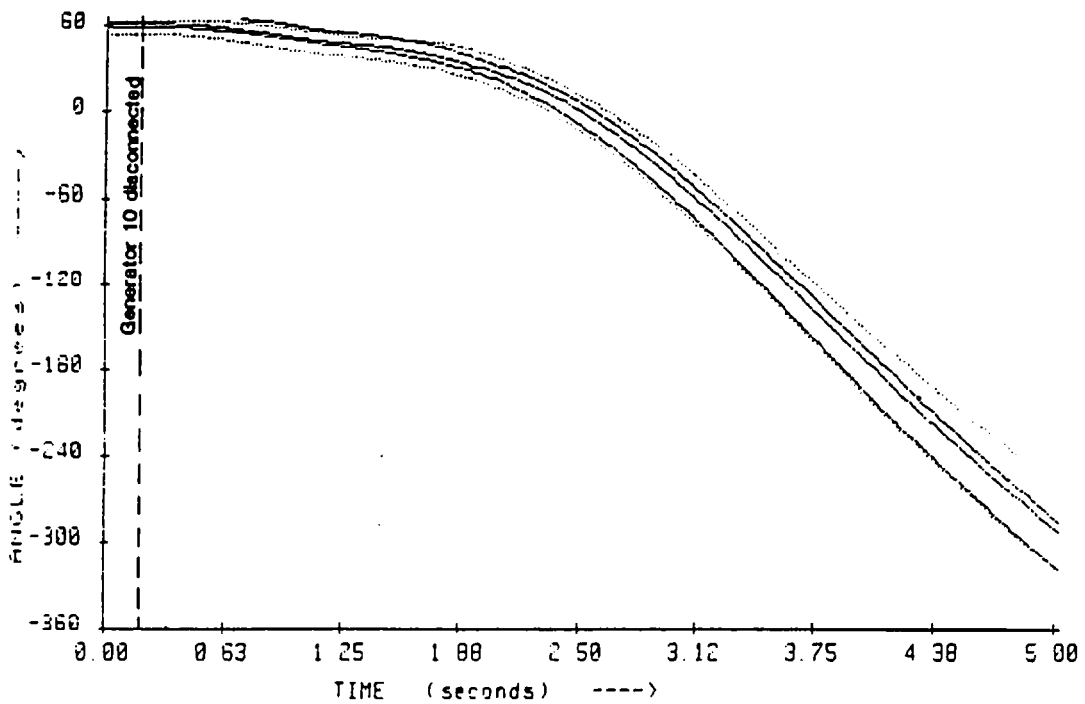


Fig. 8.10: Generator Rotor Angles in the System Area 2.
Sudden loss of the generation unit 10.

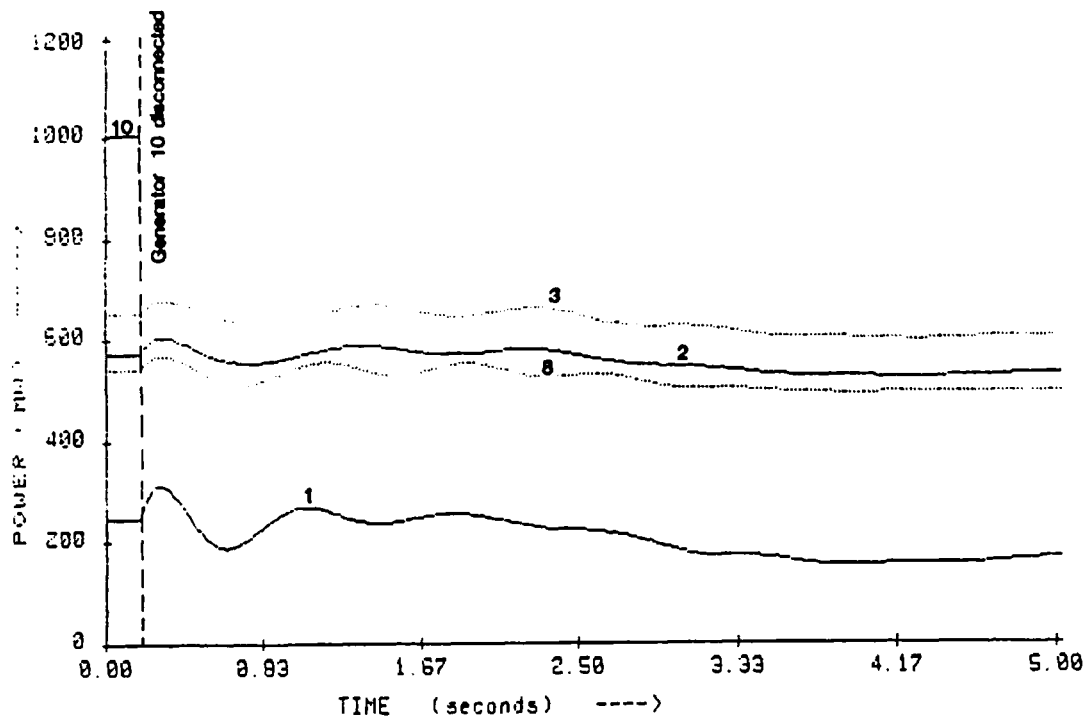


Fig. 8.11: Generator Electrical Powers in the System Area 1.
Sudden loss of the generation unit 10.

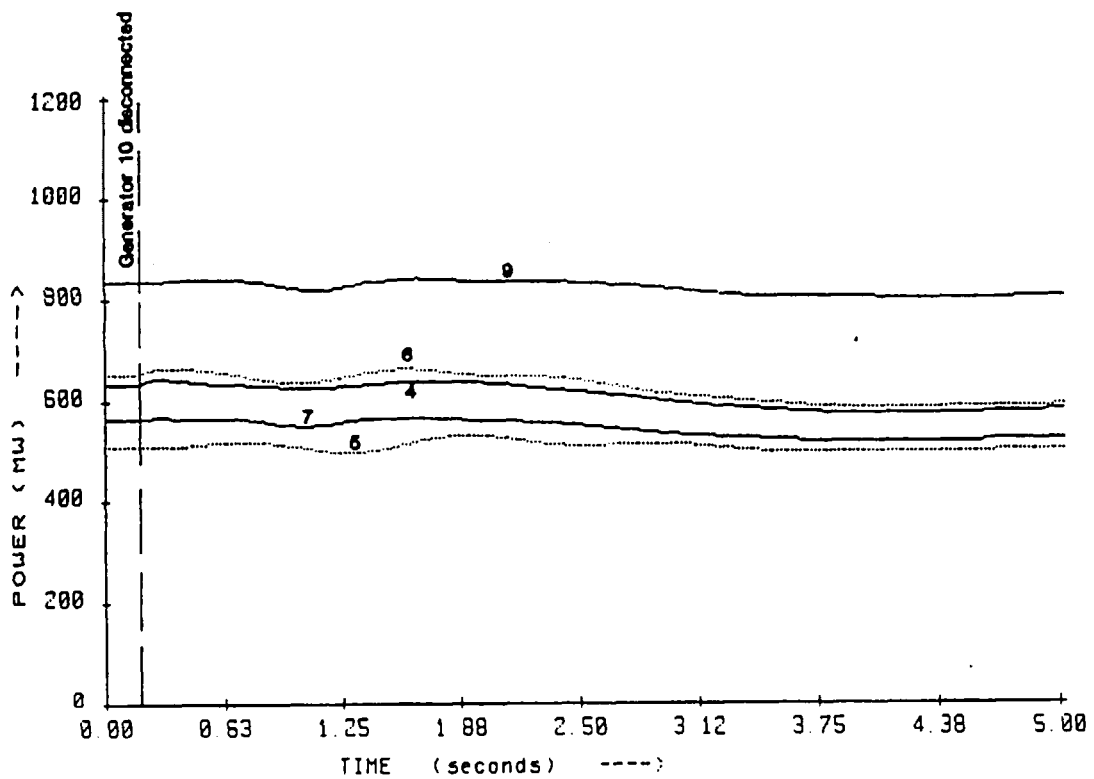


Fig. 8.12: Generator Electrical Powers in the System Area 2.
Sudden loss of the generation unit 10.

8.2.3 - Teaching and Research

The application of the simulator for teaching and research purposes is wide ranging. It could be used for testing new algorithms, modelling special characteristics of new equipment or examining the effects on system stability by changes in parameters such as generator inertias, damping factors, line reactances, etc. The latter can be done by creating data files containing the modified parameters before the initialization of the simulation, whereas the former applications require the replacement of software modules in the simulator to include the new solution techniques or modelling features.

The simulator can be used to investigate the implementation of emergency control actions for stability enhancement such as fast turbine valving. The improvement of system stability by opening and closing switches to counteract the effects of disturbances could also be examined. For instance, the loss of generation causes a decline in frequency whereas the loss of a large supply increases the frequency. Thus, the loss of generation could be counteracted by a load disconnection.

For teaching, the simulator offers a hands-on opportunity for studying the stability properties of multi-machine power systems and for demonstrating the principles of power system control hitherto confined chiefly to theoretical textbook treatment. A student could simulate system emergency conditions, e.g. faults, line outages, generator/load outage, etc., and experience emergency control measures, e.g. generator/load disconnection/reconnection, line tripping/reclosure, etc.

8.2.3.1 - Example

The power system scenario chosen in this example consists of a sequence of events which is listed in Table 8.1.

TABLE 8.1

Power System Scenario

EVENT		TIME (s)
1	Single-line-to-ground fault on line 22-23 side of bus 22	0.00
2	Fast turbine valving on unit 7	0.10
3	Opening of circuit breakers at the far end bus of faulted line	0.24
4	Fault removed by opening the breakers at the near end bus of faulted line	0.36
5	Reclosing of line 22-23	0.54
6	50% Opening of the interceptor valves in unit 7	0.60
7	Generation unit 7 disconnected	1.00
8	Rapid resynchronization of generation unit 7	1.50
9	Raise of scheduled generation power in unit 6	2.00
10	End of study	3.00

The dynamic behaviour of the system for the sequence of events listed in Table 8.1 is summarised in Fig. 8.13 through 8.19. Figs. 8.13 and 8.14 show the rotor angle oscillations for the

generation units in the areas 2 and 1 respectively.

Fig. 8.17 shows how after the application of fast turbine valving in the generation unit 7, the mechanical power falls instantaneously to 330 MW from 550 MW and to 200 MW in around 0.5 s. Then, the mechanical power raises to 340 MW after a quick reopening of the interceptor valves to 50%. These control actions cause a reduction in the generator power output as shown in Fig. 8.16 and a deceleration as illustrated in Figs. 8.13 and 8.15.

The electrical power curves in Fig. 8.16 show the instantaneous pick-up power of the generator 7 when it is rapidly resynchronised to the system and the acceleration/deceleration caused by the variation in the initial loading of the generator can be observed in Figs. 8.13 and 8.15.

The excitation system response and the variation in the terminal voltage magnitude for the generating plants in the system area 2 are plotted in Figs. 8.18 and 8.19 respectively.

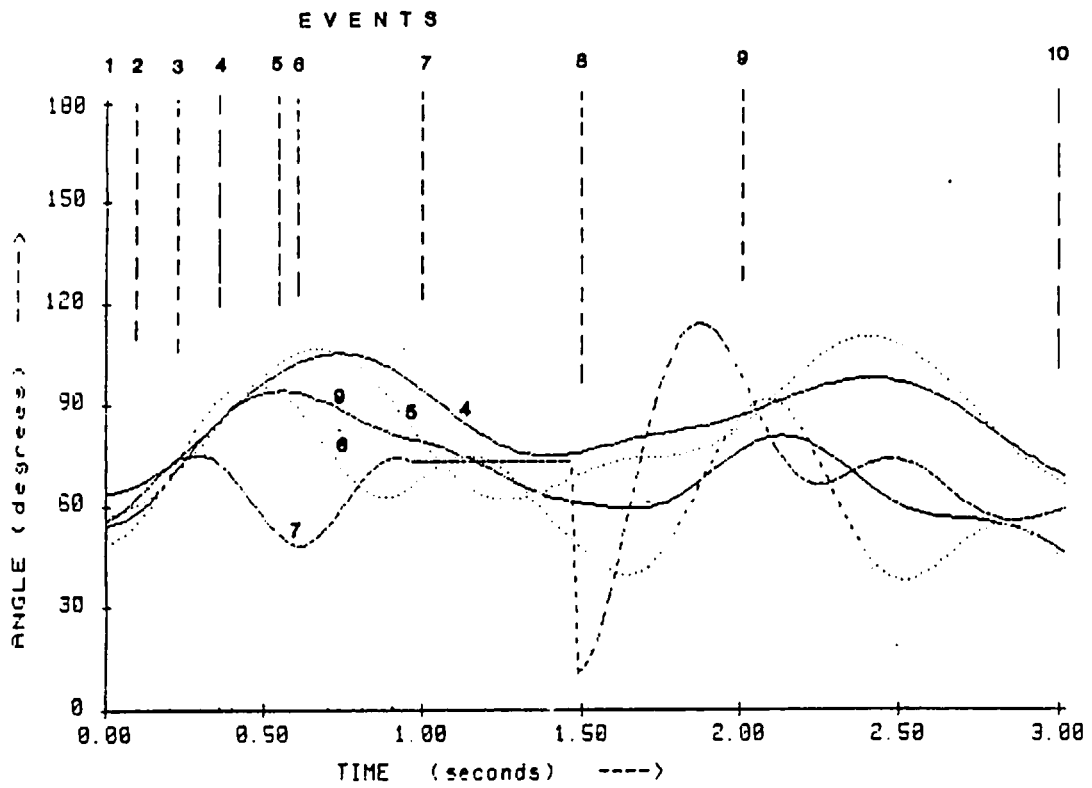


Fig. 8.13: Generator Rotor Angles in the System Area 2 for the Sequence of Events Listed in the Table 8.1.

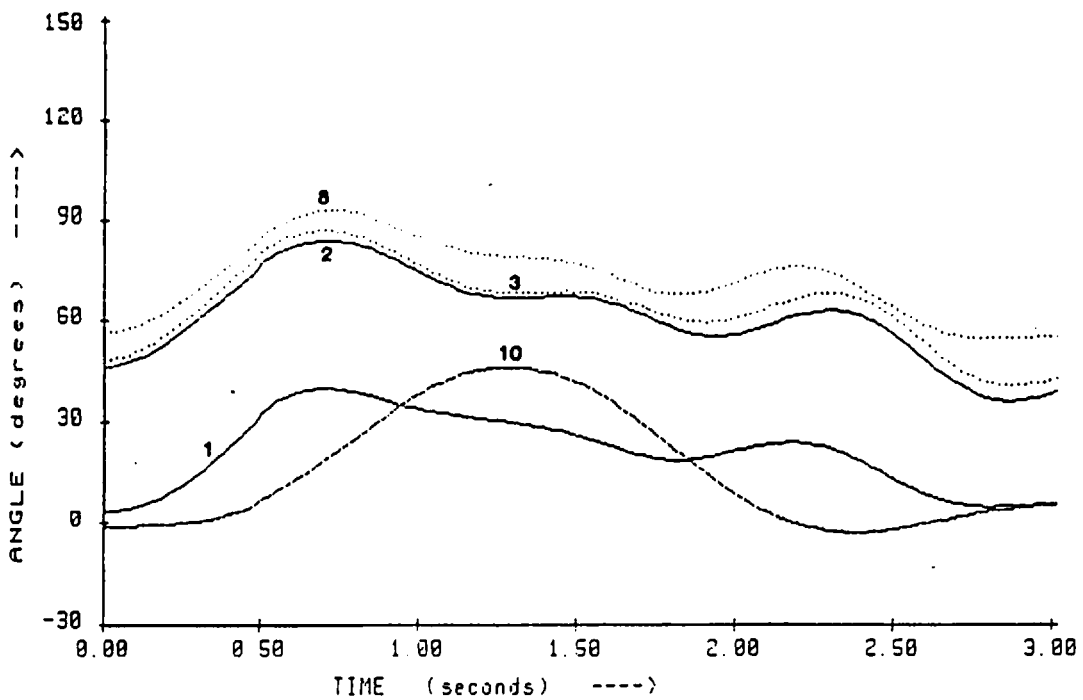


Fig. 8.14: Generator Rotor Angles in the System Area 1 for the Sequence of Events Listed in the Table 8.1.

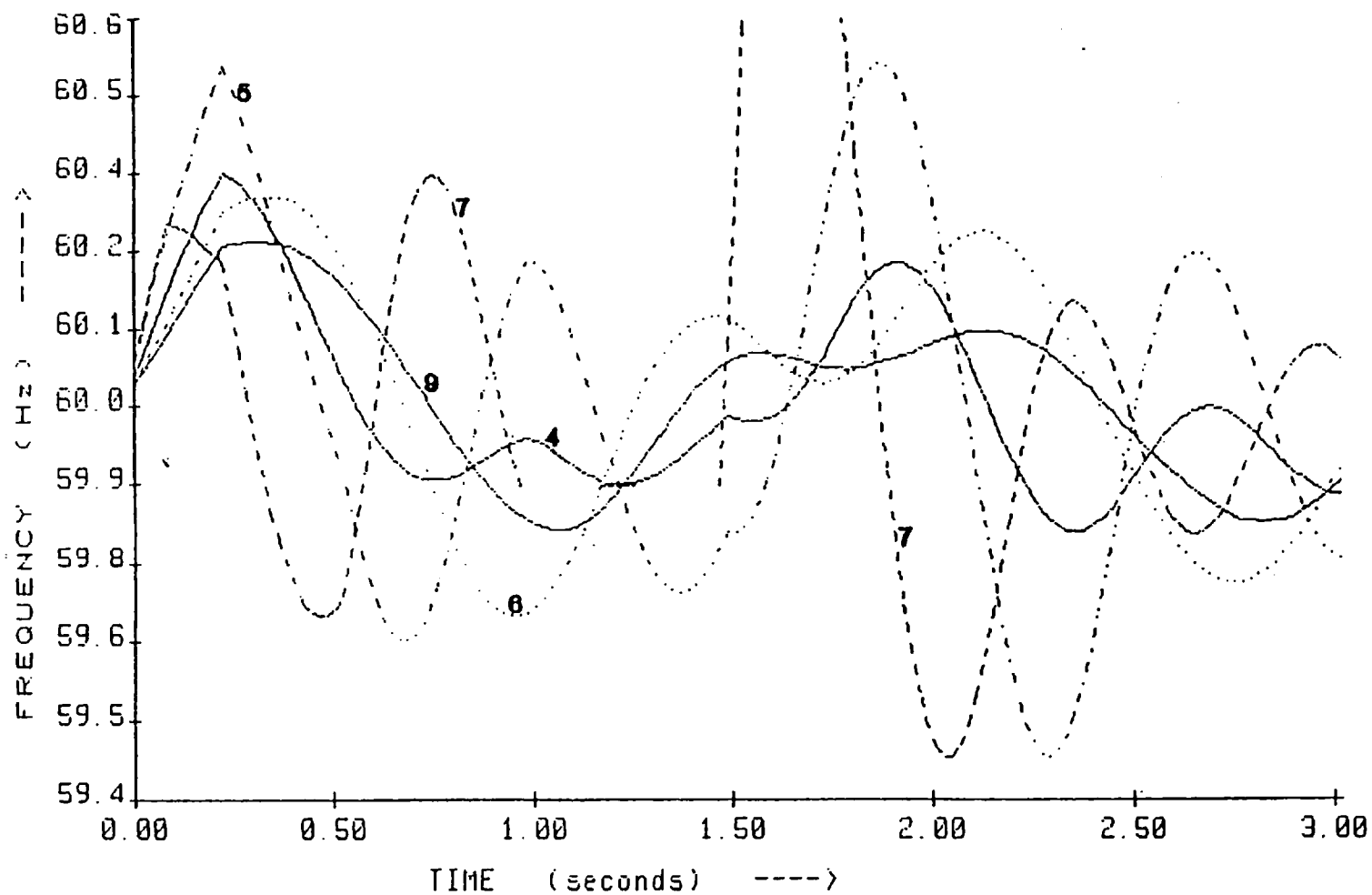


Fig. 8.15: Generator Frequencies in the System Area 2 for the Sequence of Events Listed in the Table 8.1

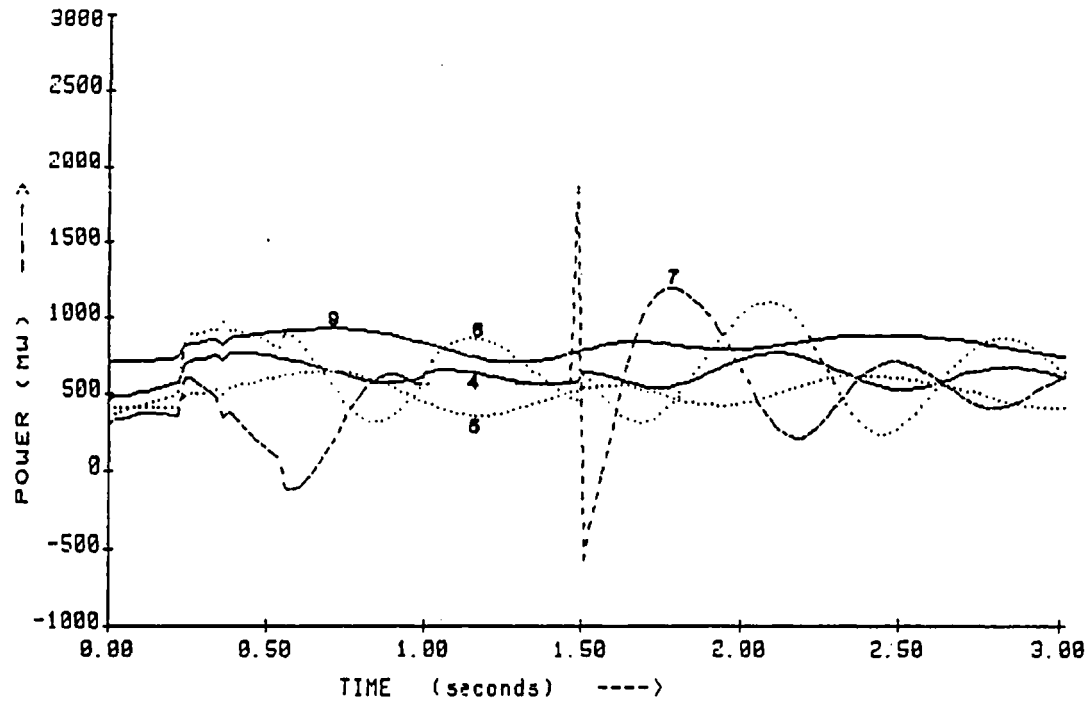


Fig. 8.16: Generator Electrical Powers in the System Area 2 for the Sequence of Events Listed in the Table 8.1.

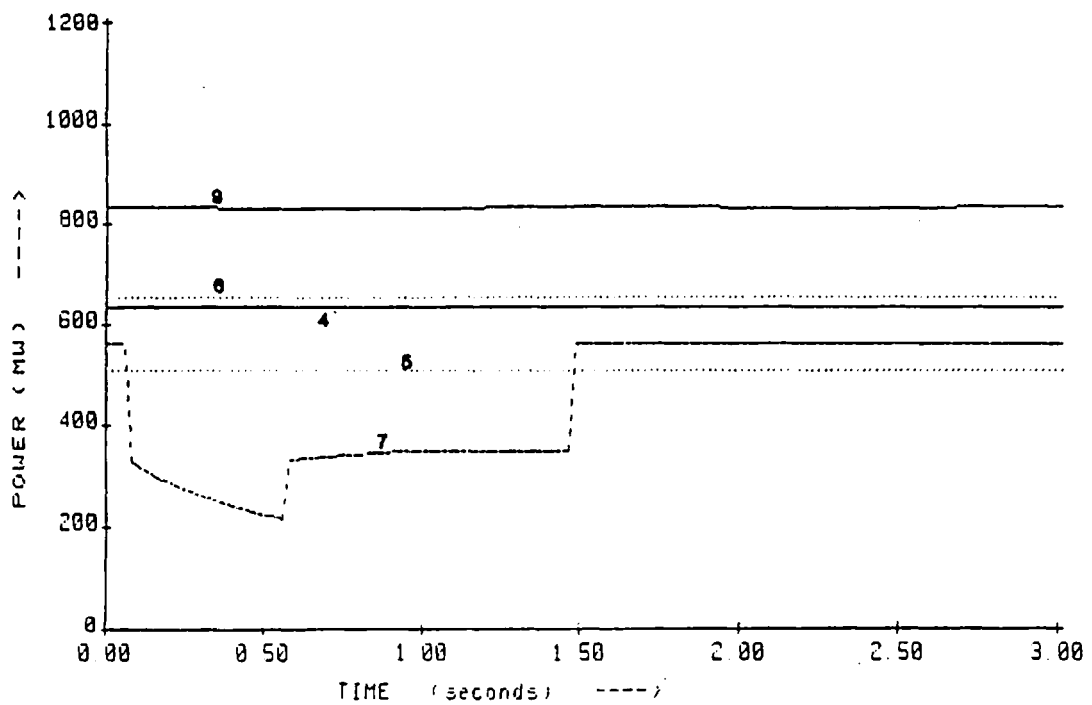


Fig. 8.17: Generator Mechanical Powers in the System Area 2 for the Sequence of Events Listed in the Table 8.1.

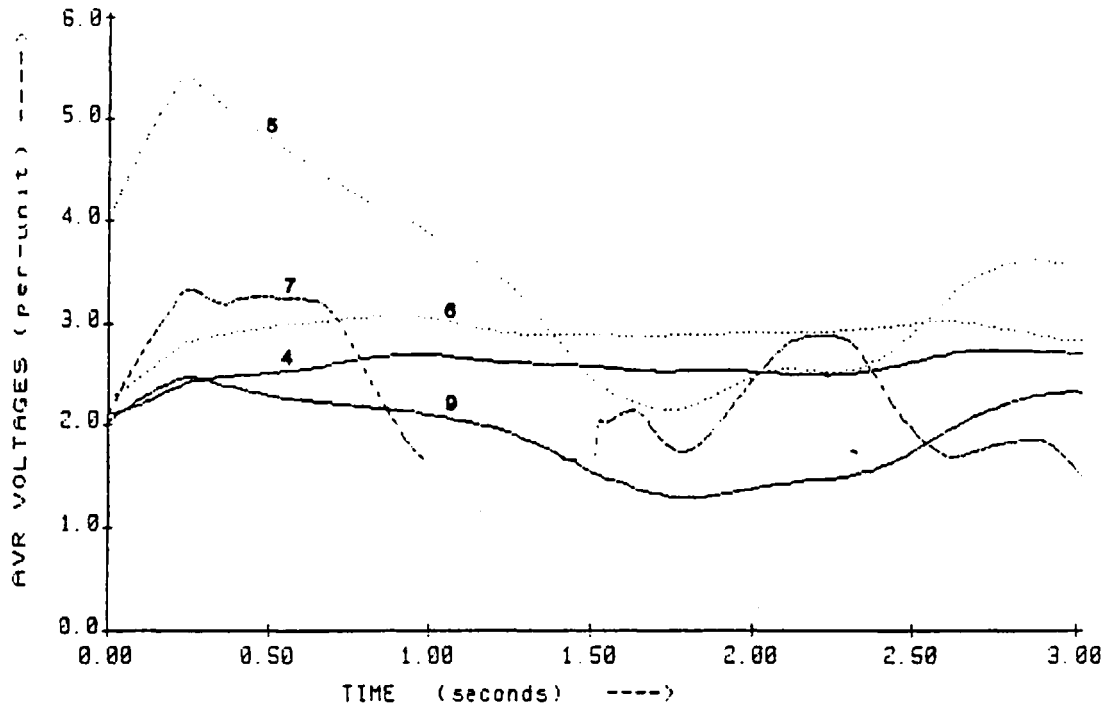


Fig. 8.18: Generator Excitation Voltages in the system Area 2 for the Sequence of Events Listed in the Table 8.1.

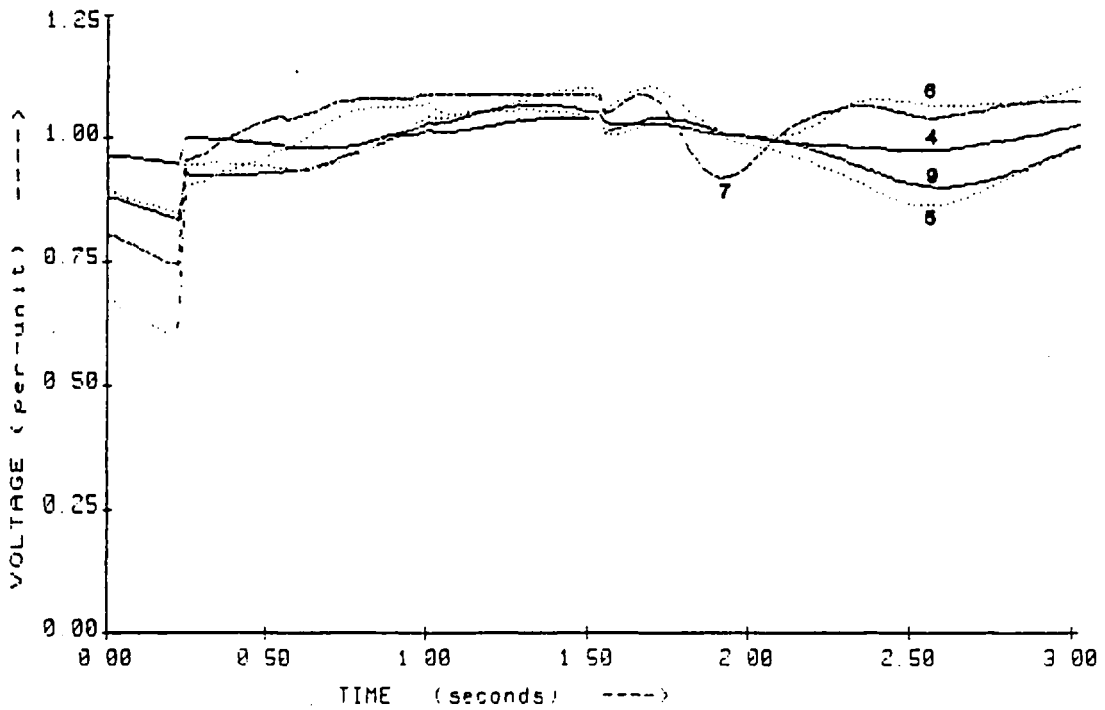


Fig. 8.19: Generator Terminal Voltages in the System Area 2 for the Sequence of Events Listed in the Table 8.1.

CHAPTER NINE

CONCLUSIONS

9.1 - GENERAL CONCLUSIONS

The application of microprocessors and parallel processing techniques for power system analysis has been investigated in this thesis. It has been shown that inexpensive microprocessor units can be used effectively for both the solution of large power system problems and the development of efficient computational tools for system analysis.

A simulator has been developed utilizing multiple microprocessors connected in parallel for the economical analysis of complex power systems. The simulator was constructed using off-the-shelf hardware units so as to minimise the building of specialised hardware wherever possible. The simulator is composed of five microprocessor units, each one of which consists of a TI-TMS9900, 16-bit microprocessor as CPU and 64 Kbytes of main memory. A TI-FS990/4 minicomputer is used to provide a versatile user/simulator interface for interactive purposes.

A modular approach that uses two levels of decomposition is proposed to tailor the simulator architecture in such a way that the configuration of microprocessors closely parallels the natural structure of a power system. In the first decomposition level, a system is divided into subsystems using a heuristic clustering algorithm that exploits the topology of the interconnected network to obtain a practical decomposition. In the second decomposition level, each subsystem is further divided into generation plants, electric loads and transmission network. One microprocessor is then used to represent the transmission network and loads in one of the subsystems and another microprocessor is employed to represent all the generation plants in that subsystem.

The software is written in the high level programming language PASCAL and only special purpose routines such as the microprocessor intercommunication procedures are written in assembly language. Software portability, modularity and ease of maintenance represent some of the major advantages of programming the algorithms in a high level language.

The PASCAL language is a possible alternative to traditional languages such as FORTRAN and BASIC for the implementation of power system analysis programs. The structured properties of the PASCAL language and the flexibility to create and manipulate data structures can be exploited fully for the development of computational algorithms, especially when programming the sparsity-oriented techniques required for the efficient handling of large power networks. A disadvantage of PASCAL for power system applications is that it does not support the COMPLEX data type and therefore arithmetic operations on complex numbers must be performed explicitly or by procedures coded for that purpose.

The development of algorithms for the simulator provided the opportunity: to explore the capabilities of microprocessors for the solution of power system problems, to implement multiple processor configurations for power system analysis hitherto confined to feasibility studies and to evaluate on the actual array of microprocessors the performance of parallel solution algorithms previously evaluated on hypothetical hardware.

A software package for steady state analysis was employed to explore the capabilities of a single microprocessor for the solution of power system analysis problems. Dynamic data structures in PASCAL are fully exploited for the implementation of the sparsity-oriented programming techniques required for the efficient handling of large power networks. A semi-dynamic storage scheme is proposed to store the sparse admittance matrix and a network orientation procedure is incorporated to speed up the factorisation process and to simplify the handling of large networks on a microprocessor. The sparsity procedures implemented have been used as a building block for the development of several power system algorithms.

It was found that with the developed programs it is possible to perform load flow contingency and fault analysis studies for power

systems of about 400 buses on a single microprocessor unit. Although an MPU has enough capability for the analysis of large systems, the computation time is rather slow when compared with mainframe computers. One iteration of the fast decoupled load flow for the solution of the IEEE 118-bus test system is approximately 300 times slower on the TMS9900 microprocessor than the execution time reported by Stott and Alsac [69] for a CDC 7600 mainframe computer. However, microprocessors are improving their capabilities constantly and are becoming even more inexpensive. Already there exist in the market microprocessors with greater processing throughput than the TMS9900. For example, the TMS99000, 16-bit microprocessor uses a 24-MHz clocking rate and offers an overall processing throughput of 12 times that of the TMS 9900. It should be stressed that since the developed programs are written in a high level language they could be run on different microprocessors.

Hierarchical computation, decomposition techniques and parallel solution algorithms have been developed for a wide spectrum of system studies using multiple microprocessor configurations.

A multiple microprocessor configuration suitable for the implementation of BBDF algorithms is employed for the solution of large electric power networks. A heuristic clustering algorithm is used to choose a decomposition pattern to divide a network into nearly equal-sized subnetworks with a minimum of interconnections and an efficient piecewise technique is proposed to partition the network into the chosen subnetworks. Each subnetwork is then assigned to one microprocessor for solution.

The proposed piecewise technique combines the best computational features of previously proposed decomposition techniques, retains the same properties as the untorn approach and eliminates singularities created by decomposition.

Recently proposed matrix block elimination algorithms that fully exploit sparsity at all levels in the computation process have been implemented for the factorisation and solution in parallel of the diakoptic equations representing a partitioned network. The performance of the algorithms is evaluated using multiple microprocessors for the solution of the linear equation $[Y][V] = [I]$ for IEEE standard test systems.

From the results obtained it is found that the formation of the

interarea matrix, a problem commonly disregarded in the theoretical evaluation of parallel solution algorithms, is the most time consuming calculation in the solution process of a partitioned system. Also, the results show that larger networks benefit more from parallel computations than smaller networks and that the performance of the algorithms is system dependent. The solution of the 57-bus system is obtained faster with the algorithm proposed by Hajj [89], whereas Alvarado's algorithm [88] produced the fastest solution for the 118-bus system.

A variation in the implementation of matrix block algorithms was developed to achieve overall computational efficiency in the parallel solution of decomposed systems. Instead of calculating the elements of the inverse matrix required for the formation of the interarea matrix by performing partial repeat solutions, as suggested in previously proposed algorithms, the recursive scheme by Takahashi et al. [92] is incorporated for the rapid computation of these elements. Moreover, an extension that further parallelises and consequently improves the performance of the parallel solution algorithm proposed by Hajj [89] was developed by modifying the computation sequence of the direct solution calculation. The variation developed for the parallel solution of partitioned networks produced faster execution times for both the formation of the interarea matrix and the direct solution computation than the algorithms by Wu [87], Alvarado [88] and Hajj [89] in all the cases studied.

From the computational results obtained, when 2, 3 and 4 MPUs are used for solution it is noted that the optimum number of MPUs (subdivisions) required for solution is system dependent and that unless used with great caution the decomposition of networks for concurrent calculations may result in greater computational requirements than a uniprocessor calculation without tearing. It is, therefore, recommended that a tearing algorithm be used to identify suitable decomposition patterns for a given network in order to achieve maximum computational efficiency.

The developed piecewise algorithms were applied to the fast decoupled load flow method for the solution of load flow problems using multiple microprocessors. The proposed piecewise technique does not affect the convergence characteristics of the FDLF and offers the

following computational characteristics:

- . There is no need to select temporary reference buses in the subsystems. Hence the calculation of correction factors and the adjustment of temporary reference buses at each iteration are not required.
- . No extra information is needed relating to the convention adopted for the current flow direction in the torn branches.
- . Buses connected with torn branches only are treated as normal buses.
- . The computational effort at the subsystem level depends on the number of boundary buses rather than on the number of torn branches.

From the computational point of view it is recommended that the branches connecting PV buses be cut, preferably when partitioning a network, since this will result in a smaller interarea matrix of the decomposed matrix $[B'']$ of the FDLF.

A parallel computation algorithm based on the FDLF has been implemented for the solution of load flow problems using a "star" configuration of microprocessor units. In this configuration each MPU solves a subnetwork and in addition the central MPU solves the intersubdivision equations of the decomposed matrices $[B']$ and $[B'']$. The performance of the algorithm was verified and evaluated for the solution of IEEE standard test systems. The parallel algorithm produced numerical solutions indistinguishable from those obtained with the untorn approach and required the same number of iterations. With the application of 4 MPUs to solve the IEEE 118-bus system the execution time is reduced to 37% of that required by a single MPU. For all the cases studied using 2-4 MPUs the parallel solution was always superior to the uniprocessor solution without tearing. The results have demonstrated the effectiveness of the application of parallel processing techniques for the load flow problem solution of large power systems.

Parallel processing schemes are proposed for the solution of the transient stability problem using multiple microprocessors. In the approach implemented, the state equations describing the dynamic

behaviour of generation plants are solved using a second order predictor (midpoint integration)/corrector (trapezoidal rule) scheme, and the solution of the interconnected network is obtained using the factorised admittance matrix. The load-voltage characteristics are considered to be valid only for a small voltage deviation from nominal. With the developed technique for the network solution, iterations are performed only when dynamic saliency is represented in the generator models and a discontinuity occurs in the network.

Detailed simulations of different test systems have been carried out using 4 MPUs. The decomposition approach for tailoring the simulator architecture is used to distribute the tasks among the MPUs for transient stability simulations of bulk power systems. One second of transient response for a 9-bus, 3-generator system is computed in about 47 seconds and for a 39-bus, 10-generator system in 131 seconds approximately (excluding output).

It has been shown that for a given system there exists an optimum number of processors, suitable configuration and task distribution for which the most economical simulation is obtained.

A unified approach for both contingency and fault analysis has been developed for the simulation of balanced and unbalanced disturbances in a power system. The approach uses a post-compensation method for the computation of faults and branch outages, and a pre-compensation method for switching out generators, loads and capacitor banks. The major power of the approach presented is that it requires very little arithmetic, since partial or complete refactorisations of the network admittance matrix are not necessary for the simulation of a small number of simultaneous changes in the network.

With the unified approach, the simulation of disturbances as a result of a cascade of events is easily handled in dynamic simulation studies. Also, the approach can be employed to predict system steady state conditions following equipment outages, or to perform detailed short circuit calculations using one microprocessor or a multiple array.

A very effective processing scheme for short circuit calculations of large power systems was implemented using a single microprocessor unit and the TI-FS990/4 host minicomputer. The zero sequence impedance

matrix is precomputed and stored on floppy disk in the disk units of the minicomputer, and the positive sequence parameters of the system to be studied are stored in main memory on the MPU. When fault analysis studies are performed, the MPU retrieves from the data file in the disk units of the minicomputer the zero sequence elements required for the computation of unsymmetrical faults.

The microprocessor-based simulator developed for power system analysis comprises detailed models of the main power system components for short/mid term dynamics a set of typical disturbances, a series of control actions and a data base. The simulation of events and control actions have been implemented as manual operations to be initiated by the user interactively with the simulator.

The simulator can be used as a computational aid in a number of power system studies including:

- . Load flow analysis,
- . Contingency analysis,
- . Fault analysis and
- . Transient stability analysis.

The application of the simulator has been demonstrated in a range of power system analysis studies for design, planning, teaching and research.

9.2 - CONTRIBUTIONS

The original contributions presented in this thesis are:

- (i) Exploration of the capabilities of inexpensive microprocessor units for the development of a microprocessor-based simulator for power system analysis.
- (ii) Implementation of multiple processor configurations for power system analysis. Strategies are proposed to decompose power system problems for the solution in parallel using an array of microprocessors. Software tools have been implemented to identify practical

decomposition patterns to partition a system into subsystems.

- (iii) The derivation of an efficient piecewise technique that combines the best computational features of previously proposed decomposition techniques. The technique retains the same properties as the untorn approach and does not require the selection of temporary reference buses in the subnetworks to eliminate singularities created by decomposition.
- (iv) The implementation and evaluation of parallel algorithms for matrix factorisation and substitution procedures for the solution of partitioned systems using multiple microprocessors. A variation in the formulation of matrix block algorithms was developed for the parallel solution of decomposed systems. The variation is more efficient than previously proposed parallel processing algorithms.
- (v) The implementation of a parallel fast decoupled load flow algorithm for the solution of load flow problems using multiple microprocessors connected in a "star" configuration.
- (vi) The development of parallel processing techniques for the solution of the transient stability problem using microprocessor arrays. A post-compensation method is used for the simulation of symmetrical and unsymmetrical faults and branch outages. A pre-compensation method is employed for switching out generators, loads and capacitor banks.
- (vii) Development of a software package for steady state analysis using a single microprocessor unit. An effective processing scheme for short circuit calculations has been developed using microprocessors. The zero sequence impedance matrix is stored in a backing store and only the positive sequence parameters are stored in main memory on the microprocessor unit. The contingency analysis part of the software package is a practical computational tool that can be used for rapid

evaluations of a single or a small number of simultaneous changes in the system from an available base-case load flow.

- (viii) The use of dynamic data structures in PASCAL for the implementation of the sparsity-oriented programming techniques required for the efficient manipulation of large power networks. A semi-dynamic storage scheme is presented and a network orientation procedure is incorporated to simplify the handling of large networks on the microprocessors.

9.3 - SUGGESTIONS FOR FURTHER RESEARCH

Dynamic simulations of bulk power systems generate a massive output of results that makes it difficult to analyse the system conditions during the step-by-step calculation. Methods need to be devised to identify and summarise the important information from the simulation of the system dynamic response to indicate potential loss of synchronism. The usefulness of the actual simulator could be greatly enhanced if stability indices are developed to summarise the response of a power system to a disturbance or a set of disturbances. These indices will be useful in planning studies to identify "worst cases". In operation studies these indices could be represented as system operating constraints to define secure system configurations with respect to postulated fault conditions (see [99]).

The simulator could be enhanced by including an option to simulate in an automatic mode the emergency control functions implemented as manual actions initiated by the user. This would require the development of algorithms to simulate system protective schemes that disconnect/reconnect system equipment automatically during the simulation and the implementation of automatic generation control to raise/lower the scheduled generation power of a generator without user intervention.

Control mechanisms such as fast valving, fault clearing, capacitor switching, exciter control, etc., operate on the basis of local

information regardless of conditions elsewhere in the system. Methods are required to control a system during emergencies by coordinating the operation of the multiple control mechanisms existing in the system. With the implementation of dynamic stability indices and the automatic operation of emergency control mechanisms, the simulator could be used as a test-bed for the development of emergency control strategies. The motivation for work in this area is the prevention of widespread system failures, if this is possible.

There is a growing industry interest in the development of a probabilistic stability analysis tool. This is visualised by some as a fast, approximate stability assessment tool that can be used to identify problem areas of greatest risk or to evaluate the system exposure level, when in a given operating state. Billinton and Kuruganty [108] have shown that fault occurrence, location and clearance in a power system are probabilistic in nature and therefore they should be treated that way in practical transient stability assessment. The implementation of probabilistic analysis in the simulator will require a set of probabilistic system models as well as new methods of analysis.

The extension of the simulation to a longer time frame (several minutes) will expand enormously the simulator applications. In the longer time frame ranging to many minutes, the following studies could be implemented:

- . Load frequency control
- . Underfrequency load shedding
- . Relief of overloaded lines
- . Relief of excessive phase angles across lines
- . Amount and distribution of reserve requirements
- . Effect of quick-starting units on reserve equipment
- . Daily load following
- . Economic dispatch

With the inclusion of long-term dynamic studies the simulator can be used for the training of system operation personnel or control-room staff. In order to provide a realistic environment to the trainee the

simulator should be capable of representing in great detail systems of a substantial size say 100 buses or more, and to perform the simulation in real-time or faster than real-time. It has been shown in the thesis that although it is possible to represent large power systems using microprocessor units based on the TI-TMS9900 microprocessor the execution speed is rather slow. Therefore, the simulation of bulk power systems in real-time will require the utilization of a faster microprocessor.

In order to have a reliable accuracy in the computation of power system problems when 16-bit microprocessors such as the TI-TMS9900 are employed for solution, a larger word length is required for floating point representation since a 32-bit format is very often inadequate for power system computations.

The simulator performance can be improved by the incorporation of a parallel data bus to interconnect all the microprocessors and by the utilization of hardware floating point calculations. With a parallel data bus more efficient parallel processing schemes could be implemented by using scheduling algorithms [11],[15] instead of the matrix block algorithms presented.

There will be a continuing need for the development and enhancement of computational tools for the economical analysis of ever increasing in both size and complexity power systems. The application of multiple microcomputer configurations will certainly become predominant in the solution of large-scale problems in the near future. It is hoped that this thesis will make a small contribution towards the development of new techniques for power system analysis that make full use of modern computer technology.

REFERENCES

- [1] M.J. Short and B.J. Cory, "Application of microprocessor arrays in power system simulation and analysis," Research Project Proposal, Department of Electrical Eng., Imperial College, London, March 1979.
- [2] E. Arriola-Valdes, L.L. Freris, C.G. Giles and M.J. Short, "Real-time hybrid power system simulator for on-line control studies," IEE Conference Publication No.140 On-line Operation & Optimisation of Transmission & Distribution Systems, London, 1976.
- [3] E.B. Janofsky and C.J. Durkin, "Dispatcher simulator training at Consolidated Edison," IEEE PES Winter Meeting, Paper 81 WM 033-0, Atlanta, Georgia, February 1981.
- [4] R. Podmore, J.C. Giri, M.P. Gorenberg, J.P. Britton and N.M. Peterson, "An advanced dispatcher training simulator," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-101, No.1, pp.17-24, January 1982.
- [5] G.D. Hachtel and A.L. Sangiovanni-Vincentelli, "A survey of third-generation simulation techniques," Proceedings of the IEEE, Vol.69, No.10, pp.1264-1288, October 1981.
- [6] H.H. Happ, C. Pottle and K.A. Wirgan, "An assessment of computer technology for large scale power system simulations," Proceedings IEEE PICA Conference, pp.316-324, 1979.
- [7] C. Pottle, R. Norin, B. Smith, J. Storer and S. Talukdar, "Proceedings of a computer hardware workshop," Proceedings of the SIAM International Conference on Electric Power Problems: The mathematical challenge, pp.499-512, Seattle, Washington, March 1980.
- [8] S.H. Fuller, J.K. Ousterhout, L. Raskin, P.I. Rubinfeld, J.S. Pradeep and R.J. Swan, "Multi-microprocessors: An overview and working example," Proceedings of the IEEE, Vol.66, No.2, pp.216-228, February 1978.
- [9] P. Civera, G. Conte, D. Del Corso, F. Gregoretti and E. Pasero, "The μ^* Project: An experience with a multiprocessor system," IEEE Micro, Vol.2, No.2, pp.38-50, May 1982.

- [10] R. Pritchard and C. Pottle, "High-speed power flows using attached scientific ("array") processors," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-101, No.1, pp.249-253, January 1982.
- [11] F.M. Brasch, J.E. Van Ness and Sang-Chul Kang, "The use of a multiprocessor network for the transient stability problem," Proceedings IEEE PICA Conference, pp.337-344, 1979.
- [12] I. Durham, R.C. Dugan, A.K. Jones and S.N. Talukdar, "Power system simulation on a multiprocessor," IEEE PES Summer Meeting, Paper A 79 442-5, Vancouver, Canada, July 1979.
- [13] W.L. Hatcher, F.M. Brasch and J.E. Van Ness, "A feasibility study for the solution of transient stability problems by multiprocessor structures," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-96, No.6, pp.1789-1797, Nov/Dec 1978.
- [14] J. Fong and C. Pottle, "Parallel processing of power system analysis problems via simple parallel microcomputer structures," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-97, No.5, pp.1834-1841, Sept/Oct 1978.
- [15] F.M. Brasch, J.E. Van Ness and S.C. Kang, "Simulation of a multiprocessor network for power system problems," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-101, No.2, pp.295-301, February 1982.
- [16] H.H. Happ, "Multicomputer configurations and diakoptics - Stability analysis of large power systems," Proceedings IEEE PICA Conference, pp.101-104, 1973.
- [17] R.J. Frowd, J.C. Giri and R. Podmore, "Transient stability and long-term dynamics unified", IEEE Transactions on Power Apparatus and Systems, Vol.PAS-101, No.10, pp.3841-3850, October 1982.
- [18] B. Stott, "Review of load-flow calculation methods," Proceedings of the IEEE, Vol.62, No.7, pp.916-929, July 1974.
- [19] B. Stott, "Power system dynamic response calculations," Proceedings of the IEEE, Vol.67, No.2, pp.219-241, February 1979.
- [20] C. Concordia and R.P. Shulz, "Appropriate component representation for the simulation of power system dynamics," IEEE Symposium on Adequacy and Philosophy of Modelling: System Dynamic Performance, IEEE 75 lCH0970-4-PWR, pp.16-23, 1975.

- [21] P.M. Anderson and B. Dembart, "Computational aspects of transient stability analysis," Proceedings of the SIAM International Conference on Electric Power Problems: The mathematical challenge, pp.159-189, Seattle, Washington, March 1980.
- [22] M.J.H. Sterling, Power System Control, England: Peter Peregrinus Ltd., 1978.
- [23] E.W. Kimbark, Power System Stability: Elements of Stability Calculations, Vol.I, New York: John Wiley & Sons, Inc., 1948.
- [24] E.W. Kimbark, Power System Stability: Power Circuit Breakers and Protective Relays, Vol.II, London: Chapman & Hall, 1950.
- [25] E.W. Kimbark, Power System Stability: Synchronous Machines, Vol.III, New York: Dover Publications, Inc., 1968.
- [26] P.M Anderson, Analysis of Faulted Power Systems, Iowa: The Iowa State University Press, 1973.
- [27] P.M Anderson and A.A. Fouad, Power System Control and Stability, Iowa: The Iowa State University Press, 1973.
- [28] M.A. Laughton, "Analysis of unbalanced polyphase networks by the method of phase co-ordinates - Part 1. System representation in phase frame of reference," Proceedings of the IEE, Vol.115, No.8, pp.1163-1172, August 1968.
- [29] M.A. Laughton, "Analysis of unbalanced polyphase networks by the method of phase co-ordinates - Part 2. Fault Analysis," Proceedings of the IEE, Vol.116, No.5, pp.857-865, May 1969.
- [30] R. Podmore, "General method for unbalanced fault analysis," Proceedings IEEE PICA Conference, pp.56-62, 1973.
- [31] Z.X. Han, "Generalized method of analysis of simultaneous faults in electric power systems," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-101, No.10, pp.3933-3942, October 1982.
- [32] R.H. Park, "Two reaction theory of synchronous machines, Generalized method of analysis - Part I," AIEE Transactions, Vol.48, pp.716-730, July 1929.
- [33] R.H. Park, "Two reaction theory of synchronous machines - Part II," AIEE Transactions, Vol.52, pp.352-355, June 1933.
- [34] P.L. Dandeno, R.L. Hauth and R.P. Shulz, "Effects of synchronous machine modeling in large scale system studies," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-92, pp.574-582, Mar/Apr 1973.

- [35] D.W. Olive, "New techniques for the calculation of dynamic stability," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-85, No.7, pp.767-782, July 1966.
- [36] F.P. DeMello and C. Concordia, "Concepts of synchronous machine stability as affected by excitation control," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-88, No.4, pp.316-329, April 1969.
- [37] P.L. Dandeno, Prabhashankar Kundur and R.P. Shulz, "Recent trends and progress in synchronous machine modeling in the electric utility industry," Proceedings of the IEEE, Vol.62, No.7, pp.941-950, July 1974.
- [38] H.W. Dommel and N. Sato, "Fast transient stability solutions," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-91, pp.1643-1650, July/Aug 1972.
- [39] IEEE Comittee Report, "Excitation system models for power system stability studies," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-100, No.2, pp.494-509, February 1981.
- [40] P. Kundur, D.C. Lee and H.M. Zein El-Din, "Power system stabilizers for thermal units: Analytical techniques and on-site validation," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-100, No.1, pp.81-95, January 1981.
- [41] IEEE Comittee Report, "Dynamic models for steam and hydro turbines in power system studies," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-92, pp.1904-1915, Nov/Dec 1973.
- [42] M.M. Adibi, P.M. Hirsch and J.A. Jordan, "Solution methods for transient and dynamic stability," Proceedings of the IEEE, Vol.62, No.7, pp.951-958, July 1974.
- [43] F.P. DeMello, R.J. Mills and W.F. B'ells, "Automatic generation control, Part I - Process modeling," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-92, pp.710-715, Nov/Dec 1973.
- [44] P.H. Ashmole, D.R. Battlebury and R.K. Bowdler, "Power system model for large frequency disturbances," Proceedings of the IEE, Vol.121, No.7, pp.601-608, July 1974.
- [45] G.J. Berg, "Power system load representation," Proceedings of the IEE, Vol.120, No.3, pp.344-348, March 1973.
- [46] C.C. Young, "Equipment and system modelling for large-scale stability studies," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-91, pp.99-109, Jan/Feb 1972.

- [47] G.W. Stagg and A.H. El-Abiad, Computer Methods in Power System Analysis, New york: McGraw-Hill, 1968.
- [48] T.E. Dy Liacco, "Real-time computer control of power systems," Proceedings of the IEEE, Vol.62, No.7, pp.884-891, July 1974.
- [49] K. Wickham, "Pascal is a 'natural'," IEEE Spectrum, Vol.16, No.3, pp.35-41, March 1979.
- [50] N. Writh, "On the composition of well-structured programs," Computing Surveys, Vol.6, No.4, pp.247-259, December 1974.
- [51] R.W. Jensen, "Tutorial Series-6: Structured Programming," IEEE Computer Society Magazine, Vol.14, No.3, pp.31-48, March 1981.
- [52] M.K. Enns, F.L. Alvarado and Ko-Chih Liu, "Power system programming in Pascal," Proceedings IEEE PICA Conference, pp.238-244, 1979.
- [53] A.M. Tenenbaum and M.J. Augenstein, Data Structures Using Pascal, Englewood Cliffs, N.J.: Prentice-Hall, 1981.
- [54] The 9900 Microprocessor Pascal System: User's manual, Texas Instruments, Inc., 1979.
- [55] 9900 Family System Design and Data Book, 1st Edition, Houston, Texas: Texas Instruments Inc.
- [56] L.H. Pollard, "Multiprocessing with the TI 9900," Proc. of the 11th Annual Asilomar Conference on Circuit Systems and Computers, pp.461-465, Pacific Grove, CA, USA, November 1977.
- [57] N. Sato and W.F. Tinney, "Techniques for exploiting the sparsity of the network admittance matrix," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-82, pp.944-950, December 1963.
- [58] J. Carpentier, "Ordered eliminations," Proccedings of Power System Computation Conference, London, 1963.
- [59] I.S. Duff, "A survey of sparse matrix research," Proccedings of the IEEE, Vol.65, No.4, pp.500-535, April 1977.
- [60] A. Brameller, R.N. Allan and Y.M. Haman, Sparsity, Its Practical Application to System Analysis, London: Pitman 1976.
- [61] E.C. Ogbouobiri, "Dynamic storage and retrieval in sparsity programming," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-89, No.1, pp.150-155, January 1970.
- [62] A. George and J.W. Liu, Computer Solution of Large Positive Definite Systems, Englewood Cliffs, N.J.: Prentice-Hall, 1981.

- [63] E.C. Ogbouobiri, W.F. Tinney and J.W. Walker, "Sparsity-directed decomposition for Gaussian elimination on matrices," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-89, No.1, pp.141-150, January 1970.
- [64] B. Stott and E. Hobson, "Solution of large power-system networks by ordered elimination: A comparison of ordering schemes," Proceedings of the IEE, Vol.118, No.1, pp.125-134, January 1971.
- [65] W.F. Tinney and J.W. Walker, "Direct solutions of sparse network equations by optimally ordered triangular factorisation," Proceedings of the IEEE, Vol.55, No.11, pp.1801-1809, November 1976.
- [66] K. Zollenkopf, "Bi-factorisation-basic computational algorithm and programming techniques," Large Sparse Sets of Linear Equations, pp.75-96, Edited by J.K. Reid, Academic Press, 1971.
- [67] IEEE Committee Report, "IEEE Reliability test system," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-98, No.6, pp.2047-2054, Nov/Dec 1979.
- [68] L.L. Freris and A.M. Sasson, "Investigation of the load-flow problem," Proceedings of the IEE, Vol.115, No.10, pp.1459-1470, October 1968.
- [69] B. Stott and O. Alsac, "Fast decoupled load flow," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-93, pp.859-869, May/Jun 1974.
- [70] B.J. Cory, Private Communication.
- [71] M.K. Enns, J.J. Anada and B. Sackett, "Fast linear contingency analysis," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-101, No.4, pp.783-791, April 1982.
- [72] T.K.P. Medicherla and S.C. Rastogi, "A voltage-criterion based contingency selection technique," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-101, No.9, pp.3523-3531, September 1982.
- [73] W.F. Tinney, "Compensation methods for network solutions by optimally ordered triangular factorisation," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-91, No.1, pp.123-127, Jan/Feb 1972.
- [74] N.M. Peterson, W.F. Tinney and D.W. Bree, "Iterative linear AC power flow solution for fast approximate outage studies," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-91, No.5,

- pp.2048-2056, Sept/Oct 1972.
- [75] B. Stott and E. Hobson, "Power system security control calculations using linear programming, Part I and II," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-97, No.5, pp.1713-1731, Sept/Oct 1978.
 - [76] IEEE Working Group, "MW response of fossil fueled steam units," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-92, pp.455-463, 1974.
 - [77] A. Sangiovanni-Vincentelli, L.K. Chen and L.O. Chua, "An efficient heuristic cluster algorithm for tearing large-scale networks," IEEE Transactions on Circuits and Systems, Vol.CAS-24, No.12, pp.709-717, December 1977.
 - [78] G. Kron. Diakoptics-Piecewise Solution of Large-Scale Systems. London: MacDonald, 1963.
 - [79] H.H. Happ. Piecewise Methods and Applications to Power Systems. New York: John Wiley & Sons, Inc., 1980.
 - [80] A. Brameller, M.N. John, and M.R. Scott. Practical Diakoptics for Electrical Networks. London: Chapman & Hall, 1969.
 - [81] M.A. Laughton, "Decomposition techniques in power system network load flow analysis using the nodal impedance matrix," Proceedings of the IEE, Vol.115, No.4, pp.539-542, April 1968.
 - [82] A.M. Sasson, and H.E Brown, "Deductions and clarifications on the diakoptics approach," Paper XIII-A, 71 C26-PWR, Proceedings of the 7th PICA Conference, Boston, pp.433-439, 1971.
 - [83] A.M. Sasson, "Decomposition techniques applied to the nonlinear programming load flow method," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-89, No.1, pp.78-82, January 1970.
 - [84] F.F. Wu, "Diakoptic network analysis," Proceedings of the 7th PICA Conference, New Orleans, LA, pp.364-371, 1975.
 - [85] L.O. Chua, and L.K. Chen, "Diakoptic and generalized hybrid analysis," IEEE Transactions on Circuits and Systems, Vol.CAS-23, pp.694-705, December 1976.
 - [86] A. George, "On block elimination for sparse linear systems," SIAM J., Numer. Anal., Vol.11, No.3, pp.585-603, June 1974.
 - [87] F.F. Wu, "Solution of large-scale networks by tearing," IEEE Transactions on Circuits and Systems, Vol.CAS-23, pp.706-713, December 1976.

- [88] F.L. Alvarado, D.K. Reitan, and M. Bahari- Kashani, "Sparsity in diakoptic algorithms," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-96, No.5, pp.1450-1459, Sept/Oct. 1977.
- [89] I.N. Hajj, "Diakoptics and block elimination," Paper A 78 564-7, PES Summer Meeting, Los Angeles, CA, July 1978.
- [90] R. Kasturi, and M.S.N. Potti, "Piecewise Newton- Raphson load flow-an exact method using ordered elimination," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-95, No.4, pp.1244-1253, Jul/Aug. 1976.
- [91] Magdy El-Marsafawy, R.W. Menzies, and R.M. Mathur, " A new, exact, diakoptic, fast-decoupled load-flow technique for very large power systems," Paper A 79 440-9, IEEE PES Summer Meeting, Vancouver, Canada, July 1979.
- [92] K. Takahashi, J. Fagan and M. Chen, "Formation of a sparse bus impedance matrix and its application to short circuit study," Proceedings of the IEEE PICA Conference, Minneapolis, pp.63-69, 1973.
- [93] I. Elizarraraz, "Algorithms in PASCAL for power system simulation and analysis," Seminar notes, Department of Electrical Engineering, Imperial College, November 1981.
- [94] O. Alsac, B. Stott and W.F. Tinney, "Sparsity-oriented compensation methods for modified network solutions," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-102, No.5, pp.1050-1060, May 1983.
- [95] G. Gross and H.W. Hong, "A two-step compensation method for solving short circuit problems," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-101, No.6, pp.1322-1331, June 1982.
- [96] H.E. Brown, L.K. Kirchmayer, C.E. Person and C.W. Stagg, "Digital calculation of three-phase short-circuits by matrix methods," AIEE , Paper No. CP 60-181, 1959.
- [97] J.M. Undrill and T.E. Kostyniak, "Advanced power system fault analysis method," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-94, pp.2141-2148, Nov/Dec 1975.

- [98] C.A. Gross, Power System Analysis, New York: John Wiley & Sons, 1979.
- [99] Bulk Power Security Assessment, Technical Report Vol. 1, IBM Research Project RP90-3, San Jose, Ca., November 1970.
- [100] T. Athay, V. Sherkat, R. Podmore, S. Virmani and C. Puech, "Transient energy stability analysis," Section IV of the Conf. Proc. Systems Engineering for Power: Emergency Operating State Control, Davos, Switzerland, Sept. 30 to Oct. 5, 1979.
- [101] R. Lopez, Dynamic Simulation of Power Systems on Multiple Microprocessors, Ph.D. Thesis in preparation, Department of Electrical Engineering, Imperial College, July 1983.
- [102] "System load dynamics-simulation effects and determination of load constants," IEEE Computer Analysis of Power Systems Working Group, IEEE Transactions on Power Apparatus and Systems, Vol.PAS-92, pp.600-609, Mar/Apr 1973.
- [103] J. Arrillaga, C.P. Arnold and B.J. Harker, Computer Modelling of Electrical Power Systems, Chichester, England: John Wiley & Sons Ltd., 1983.
- [104] P.L. Dandeno and P. Kundur, "A non-iterative transient stability program including the effects of variable load-voltage characteristics," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-92, No.5, pp.1478-1484, Sept/Oct 1973.
- [105] I. Elizarraraz, R. Johnson, R. Lopez and M.J. Short, "Simulation of multi-machine power systems using distributed microcomputers," Proceedings of the 17th UPEC, Paper 3B.1, Manchester, England, March 1982.
- [106] O. Saito, K. Koizumi, M. Udo, M. Sato, H. McKae and T. Tsumi, "Security monitoring system including fast transient stability studies," IEEE PES Winter Meeting, Paper T 75085-6, New York, N.Y., 1975.
- [107] I. Elizarraraz, "Test power systems for network solution algorithms," Power system Report, Department of Electrical Engineering, Imperial College, July 1983.
- [108] R. Billinton and P. Kuruganty, "probabilistic assessment of transient stability in a practical multimachine system," IEEE Transactions on Power Apparatus and Systems, Vol.PAS-100, No.7, pp.3634-3641, July 1981.

- [109] R. Johnson, R. Lopez and I. Elizarraraz, "Interactive power system simulator user's manual and report," Power System Report, Department of Electrical Engineering, Imperial College, July 1983.
- [110] IEEE Standard Dictionary of Electrical and Electronics Terms, Second Edition, New York, N.Y.. The Institute of Electrical and Electronics Engineers, Inc. 1977.

APPENDIX A

COMPENSATION METHODS

A.0 - INTRODUCTION

The solution of electric power networks that undergo modifications can be performed by matrix refactorisation, factor updating techniques or compensation methods. The choice among these techniques for the computation of changes in a power system depends on the problem conditions. In general, the compensation methods are computationally more efficient when the network changes are temporary and the number of buses involved is small compared to the buses of the system.

Tinney described in [73] the application of the compensation method with sparsity-oriented techniques to solve problems involving network changes using the factors of the unmodified bus admittance matrix. Alsac et al. derived in [94] three basic compensation methods, pre-, mid- and post-compensation, for the solution of modified power networks and presented several variants for their implementations.

This appendix briefly reviews the theoretical concepts of the compensation methods which are the basis of the unified approach implemented for the efficient simulation of disturbances in a power system.

A.1 - THE COMPENSATION METHOD

The compensation theorem states that a passive network element can be replaced by an active element that causes the same network response [73]. This means that a passive branch element can be substituted for a current source without affecting the voltages or currents in the network.

To illustrate the application of the compensation method to represent the addition or removal of elements in a power network consider a balanced network which can be described in its normal

operating state by the nodal equation

$$[Y][V^0] = [I^0] \quad (A.1)$$

Now, consider that several changes are introduced in the values of some of the passive elements of the original network. In this case, the equation (A.1) becomes

$$[Y'] [V'] = [I^0] \quad (A.2)$$

where $[Y']$ is the admittance matrix of the modified network and $[V']$ is the resulting bus voltage vector which is to be determined. Note that in (A.2) it is assumed that the current injection vector is unaltered. When the compensation method is used to obtain the solution of the modified network, $[V']$ is calculated without explicitly forming and factorising $[Y']$ or solving (A.2). Instead, the method makes use of the factors of $[Y]$, which were obtained to solve the unmodified network using (A.1).

The changes in the network can be defined as

$$[\Delta Y] = [Y'] - [Y] \quad (A.3)$$

where $[\Delta Y]$ is the admittance modification matrix.

To analyse the effect of modifications in the elements of a network, the original network can be represented by the general Thevenin equivalent shown in Fig. A.1. Suppose that a single element between buses k and m is to be modified. Then the formation of the Thevenin equivalent network is as follows. The open circuit voltage is given by the voltage between buses k and m

$$V_t^0 = V_k^0 - V_m^0 \quad (A.4)$$

where V_k^0 and V_m^0 are elements of $[V^0]$.

The Thevenin equivalent impedance is the impedance of the unmodified network as it would be measured between buses k and m . This impedance can be obtained by removing all the original current

injections, i.e. setting $[I^0] = 0$, applying a 1 p.u. current across the element km, and measuring the resulting voltage across the element km. If V_{tk} and V_{tm} are defined as the voltages at buses k and m respectively which result in response to the application of a 1 p.u. current across the element km, then these voltages can be calculated by solving

$$[Y][V_t] = \begin{matrix} & k & 1 \\ & \left| \right. & \\ & m & -1 \end{matrix} \quad (A.5)$$

The Thevenin equivalent impedance is then obtained from

$$Z_t^0 = V_{tk} - V_{tm} \quad (A.6)$$

where V_{tk} and V_{tm} are elements of $[V_t]$.

The admittance matrix $[\Delta Y]$ representing the change in the element km can be expressed using the branch-oriented modification approach [94] as

$$[Y] = \begin{matrix} & [M] & [\Delta Y] & [M]^t \\ & k & \begin{bmatrix} +1 \\ -y_{km} \end{bmatrix} & \begin{bmatrix} +1 & -1 \\ k & m \end{bmatrix} \\ & m & \begin{bmatrix} -1 \end{bmatrix} \end{matrix} \quad (A.7)$$

or using the node-oriented modification approach as

$$[Y] = \begin{matrix} & \begin{matrix} [M] \\ \begin{matrix} k \\ \vdots \\ m \end{matrix} \end{matrix} \begin{bmatrix} +1 \\ \vdots \\ -1 \end{bmatrix} \end{matrix} \begin{matrix} \begin{matrix} [\Delta Y] \\ \begin{bmatrix} -y_{km} & y_{km} \\ y_{km} & -y_{km} \end{bmatrix} \end{matrix} \begin{matrix} \begin{matrix} [M]^t \\ \begin{matrix} k & m \end{matrix} \end{matrix} \begin{bmatrix} +1 \\ \vdots \\ -1 \end{bmatrix} \end{matrix} \quad (A.8)$$

The expressions (A.7) and (A.8) can be written in a condensed form as

$$[\Delta Y] = [M][\Delta y][M]^t \quad (A.9)$$

The choice between the branch and node-oriented approaches for symmetric modifications depends on the elements involved in the modifications. For instance, if the number of branches to be changed is smaller than the number of buses involved then the branch-oriented approach is preferred. Non-symmetric modifications should be handled with the node-oriented approach.

Next, the admittance $[\Delta Y]$ representing the change in the network is connected between buses k and m as shown in Fig. A.2a. Applying Kirchhoff's current and voltage laws to the resulting network the following relationships are obtained

$$[I_{km}] = -[\Delta Y][V'] \quad (A.10)$$

$$[V'] = [V_t^0] + [Z_t^0][I_{km}] \quad (A.11)$$

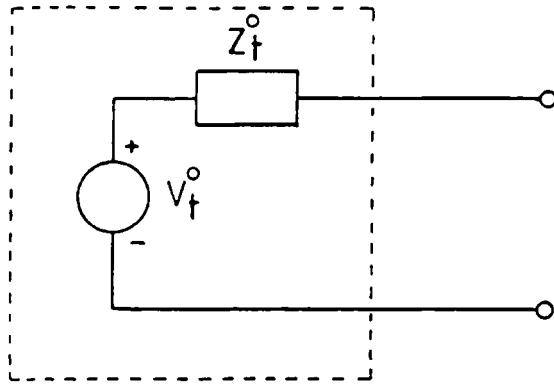
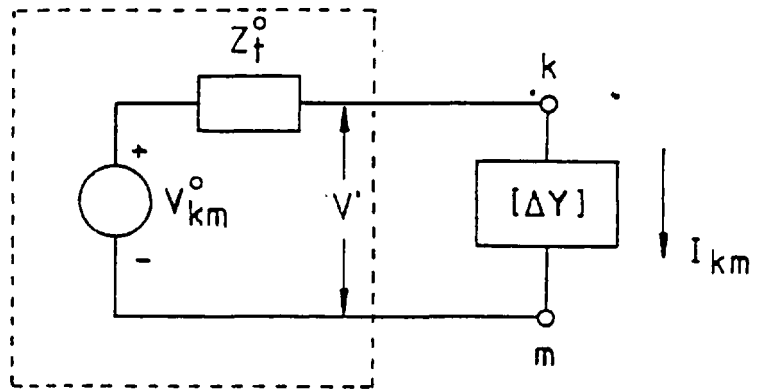
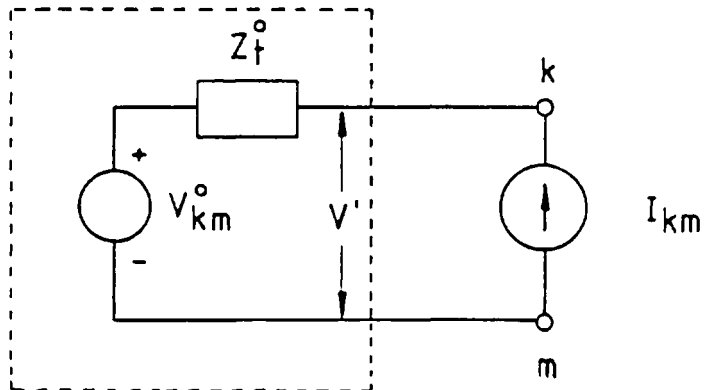


Fig. A.1: General Thevenin Equivalent Representing an Unmodified Network



(a)



(b)

Fig. A.2: Representation of Modifications in a Power Network

Combining (A.10) and (A.11)

$$[I_{km}] = -[\Delta Y]([U_n] + [Z_t^0][\Delta Y])^{-1}[V_t^0] \quad (A.12)$$

where $[U_n]$ is the unity matrix.

$$\text{Then with } [\Delta Z] = [\Delta Y]^{-1}$$

$$[I_{km}] = -([Z_t^0] + [\Delta Z])^{-1}[V_t^0] \quad (A.13)$$

and both equations (A.12) and (A.13) hold.

According to the compensation theorem the admittance modification matrix $[\Delta Y]$ can be replaced by the current source $[I_{km}]$ without affecting the resultant voltage vector $[V']$ as shown in Fig. A.2b. In the original network, it follows that the effect of network modifications may be represented by adding the compensating current $[I_{km}]$ between the terminal buses of the modified element. Therefore, by superposition, the nodal equations of the modified system can be expressed as

$$[Y][V'] = [I^0] + [\Delta I] \quad (A.14)$$

This is illustrated in Fig. A.3.

From (A.1) and (A.14), and the assumed non-singularity of $[Y]$

$$[V'] = [V^0] + [\Delta V] \quad (A.15)$$

where the voltage vector $[\Delta V]$ is given by

$$[Y][\Delta V] = [\Delta I] \quad (A.16)$$

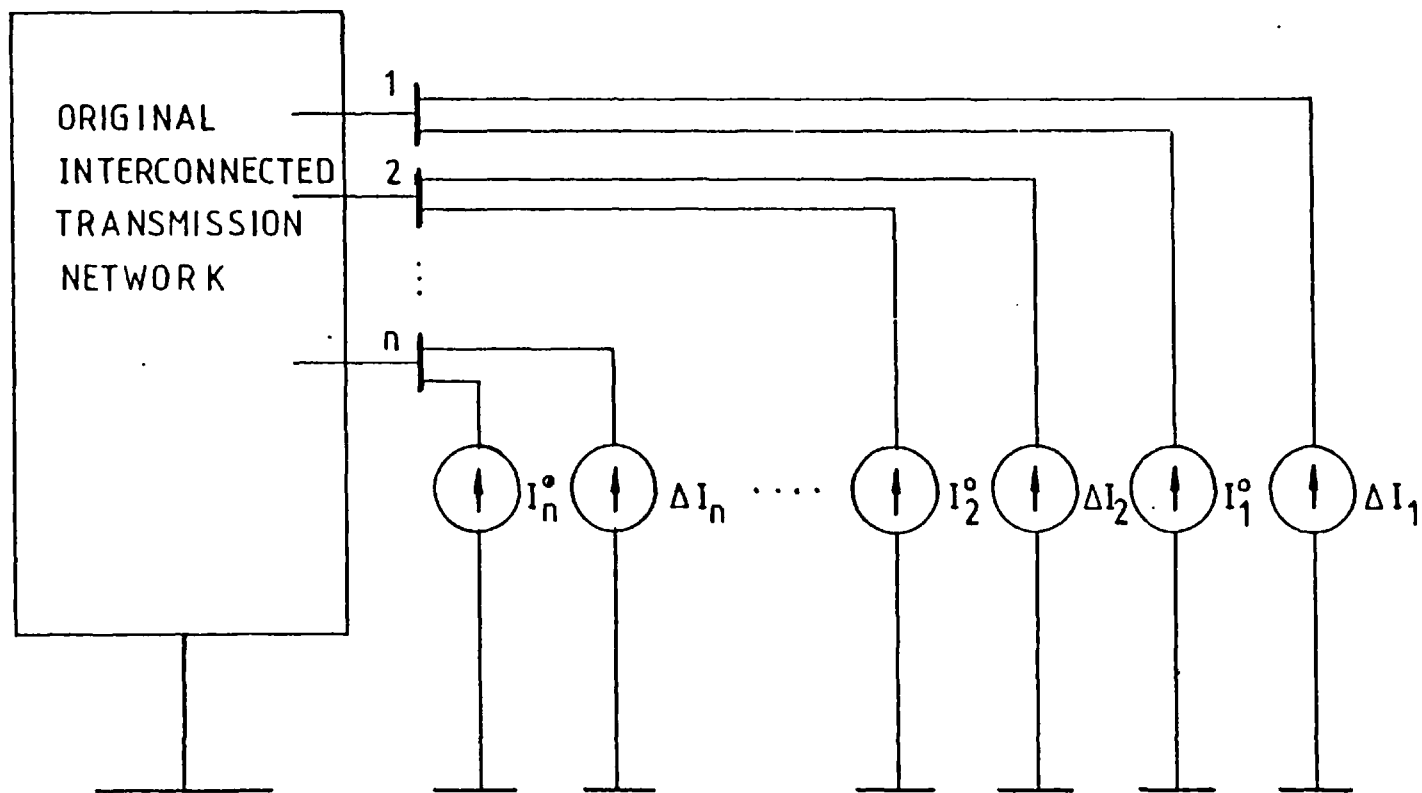


Fig. A.3: Representation of a Modified Network Using Compensation Methods

and

$$[\Delta I] = [M]([Z_t^0] + [z_{km}])^{-1} [M]^t [V^0] \quad (A.17)$$

Therefore, $[V']$ can be calculated using the compensation method by simply modifying the original voltage vector $[V^0]$ without changing or performing a complete or partial refactorisation of the admittance matrix $[Y]$.

The repeat network solution implied in (A.16) to obtain the voltage correction vector $[\Delta V]$ can be avoided if the values of $[V_t]$ calculated to determine the Thevenin equivalent impedance are saved for the computation of $[\Delta V]$ as shown below.

Expressing (A.5) as

$$[V_t] = [Y]^{-1} [M] \quad (A.18)$$

Then

$$[\Delta V] = [V_t]([Z_t^0] + [z_{km}])^{-1} [M]^t [V^0] \quad (A.19)$$

It should be noted from (A.18) that the computation of required columns of the inverse matrix $[Y]^{-1}$ is all that is needed for the application of the compensation method to the solution of modified power networks.

A.2 - COMPENSATION SCHEMES

The solution process of a modified network using the compensation method can be divided into two phases [94]:

- . The preparatory phase - the calculation of certain terms of the inverse matrix $[Y]$ or its factors.
- . The solution phase - the solution for $[V']$ using (A.15). This comprises a standard network solution using the factors of the original unmodified matrix $[Y]$, plus the compensating steps that introduce the effect of the modifications.

The preparatory phase is in principle performed only once. During the solution phase the compensating steps can be performed after, before or in the middle (after the forward but before the backward substitution) of the network solution. These different compensation schemes have been classified in [94] as:

- . Post-compensation methods
- . Pre-compensation methods
- . Mid-compensation methods

A.2.1 - Post-Compensation Methods

In the post-compensation methods the compensating steps are performed after the network solution has taken place. First, the unmodified network equation (A.1) is solved for $[V^0]$. Second, the compensation vector $[\Delta V]$ is calculated from (A.19) and the network solution $[V^0]$ is corrected to represent the effect of the network modifications using (A.15).

A.2.2 - Pre-Compensation Methods

In the pre-compensation methods the current vector $[\Delta I]$ is calculated first using (A.17) and the bus current vector $[I^0]$ is modified as indicated in (A.14) before obtaining the network solution. Thus the solution of the modified network is computed directly by solving (A.14) for $[V]$ using the compensated vector of bus currents $[I^0] + [\Delta I]$.

A.2.3 - Mid-Compensation Methods

In the Mid-compensation method the compensation of the network solution is performed after the forward substitution

$$[F] = [L]^{-1} [I^0] \quad (A.20)$$

according to

$$[F'] = [F] - [L]^{-1} [M] ([Z_t^0] + [z_{km}])^{-1} ([L]^{-1} [M])^t [F] \quad (A.21)$$

and the solution of the modified network is then completed by performing the backward substitution

$$[V'] = [U]^{-1} [F'] \quad (A.22)$$

The selection of the best compensation scheme is problem dependent. In a problem involving a series of modifications, different compensation methods or even refactorisation may well be appropriate for different cases. Apart from computer memory considerations, the compensation methods are normally selected on the basis of software and hardware characteristics and total computing effort: preparatory plus compensation phases.

In the unified approach implemented for the simulation of disturbances in a power system, a pre-compensation scheme is used to simulate generator, load and shunt reactor outages, and a post-compensation method is employed to simulate branch outages and to represent symmetrical and unsymmetrical faults.

The application of the compensation method to contingency analysis is described in chapter 4 and to short-circuit calculations in chapter 6. The simulation of cascades of events in dynamic studies using compensation methods is discussed in section 7.2.5.

APPENDIX B

NUMERICAL EXAMPLE FOR THE SOLUTION IN PARALLEL OF PARTITIONED SYSTEMS

B.0 - INTRODUCTION

In this appendix, a simple numerical example is given to illustrate the application of the piecewise technique proposed in chapter 5 to partition a power network for its solution in parallel using multiple processors. The example also shows the application of the parallel FDLF algorithm developed for the solution of load flow problems using distributed microprocessors. This includes the algorithms implemented for the factorisation and solution in parallel of a partitioned system.

For simplicity, only the first half iteration of the FDLF algorithm is considered. The voltage angle corrections are calculated first using the untorn approach and these results are used to verify the accuracy of the parallel solution.

B.1 - THE SAMPLE NETWORK

The sample network depicted in Fig. B.1 [80] is considered to illustrate the application of the algorithms developed for the solution in parallel of a partitioned system. The bus s is the system slack bus.

In this example, line resistances are neglected, it is assumed that all line reactances are 0.2 p.u. and it is supposed that the vector $[\Delta P/|V|]^t$ for the system of Fig. B.1 is given by:

$$[\Delta P/|V|]^t = \begin{array}{c} \begin{array}{cccccccc} a & b & g & c & d & h & e & f \end{array} \\ \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0.06 & -0.01 & 0.00 & -0.06 & -0.06 & 0.09 & 0.03 & -0.02 \\ \hline \end{array} \end{array}$$

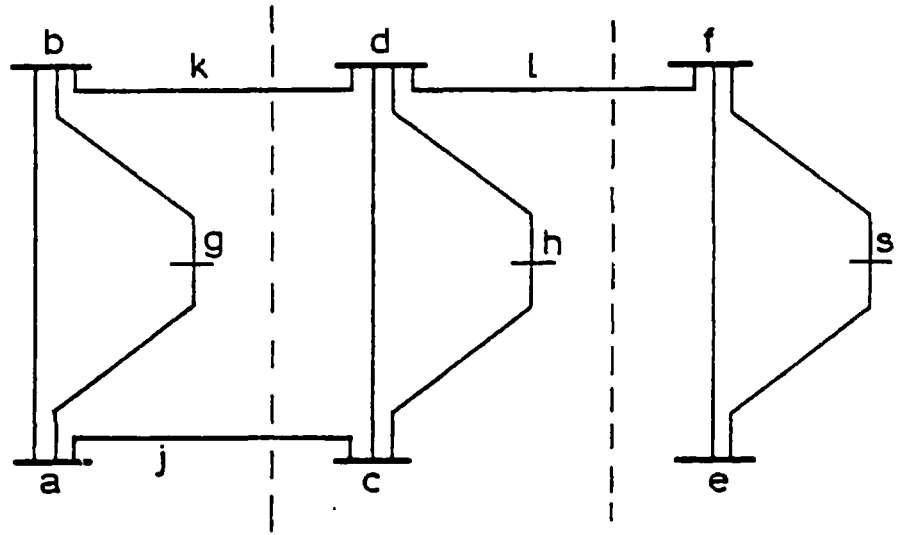


Fig. B.1: Sample Network

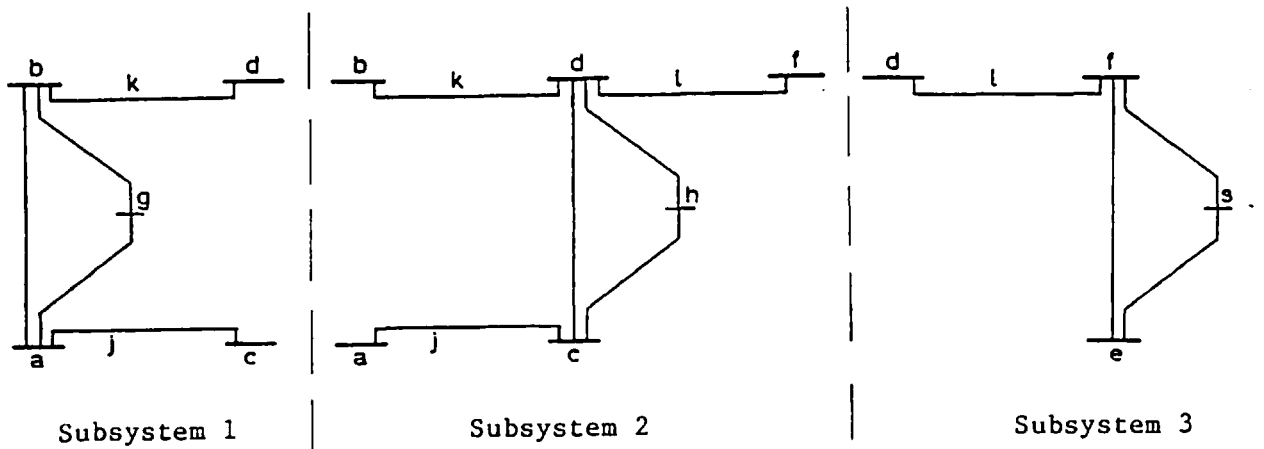


Fig. B.2: Decomposition of Sample Network

The first half iteration of the FDLF using the untorn approach consists of solving (4.2) to obtain new estimated values of the bus voltage angles. The voltage angle corrections obtained from the solution of (4.2) for this example are

$$\begin{bmatrix} \Delta \theta \end{bmatrix}^t = 1/5 \begin{bmatrix} a & b & g & c & d & h & e & f \end{bmatrix} \begin{bmatrix} 0.07 & 0.05 & 0.06 & 0.04 & 0.03 & 0.08 & 0.02 & 0.01 \end{bmatrix}$$

B.2 - SOLUTION IN PARALLEL OF A PARTITIONED SYSTEM

This section illustrates the application of the parallel FDLF algorithm developed (see Fig. 5.6) for the solution of load flow problems using multiple processors.

Assume that 3 processors are to be used for the solution of the sample system given in Fig. B.1. Then, the system is partitioned into 3 subsystems by cutting the lines j, k and l using the overlapping decomposition technique as shown in Fig. B.2. Each of the resulting subsystems is then assigned to one processor for solution.

The parallel solution of the partitioned system proceeds as follows:

- Read in data and store the elements of the matrix $[C']$ in a single vector of dimension equal to the number of boundary buses in each subsystem

$$\begin{bmatrix} a \\ b \end{bmatrix} \quad \text{MPU 1}$$

$$\begin{bmatrix} c \\ d \end{bmatrix} \quad \text{MPU 2}$$

$$\begin{bmatrix} f \end{bmatrix} \quad \text{MPU3}$$

- Form $[Y]$

$$\begin{array}{c} a \\ b \\ g \end{array} \begin{bmatrix} a & b & g \\ -15 & 5 & 5 \\ 5 & -15 & 5 \\ 5 & 5 & -10 \end{bmatrix} \quad \text{MPU 1}$$

$$\begin{array}{c} c \\ d \\ h \end{array} \begin{bmatrix} c & d & h \\ -15 & 5 & 5 \\ 5 & -20 & 5 \\ 5 & 5 & -10 \end{bmatrix} \quad \text{MPU 2}$$

$$\begin{array}{c} e \\ f \end{array} \begin{bmatrix} e & f \\ -10 & 5 \\ 5 & -15 \end{bmatrix} \quad \text{MPU 3}$$

- Form the matrix $[B']$

	a	b	g
a	15	-5	-5
b	-5	15	-5
g	-5	-5	10

MPU 1

	c	d	h
c	15	-5	-5
d	-5	20	-5
h	-5	-5	10

MPU 2

	e	f
e	10	-5
f	-5	15

MPU 3

- Ground the boundary buses and order $[B']$ to reduce fill-in, delaying as long as possible the choice of boundary buses as pivots

	g	a	b
g	10	-5	-5
a		20	-5
b		symm.	20

MPU 1

	h	c	d
h	10	-5	-5
c		20	-5
d		symm.	30

MPU 2

	e	f
e	10	-5
f		20

MPU 3

- Factorise $[B']$ independently for each subsystem

	g	a	b
g	.1000	.5000	.5000
a		.0571	.4286
b			.0700

MPU 1

	h	c	d
h	.1000	.5000	.5000
c		.0571	.4286
d			.0412

MPU 2

	e	f
e	.1000	.5000
f		.0571

MPU 3

- Form the intersubdivision matrix (ISM) $[X'_{tt}]$ according to (5.18). Each MPU computes $[A'] = [U']^{-1}[L']^{-1}[C']$ by performing partial repeat solutions using the factorised $[B']$ with the columns of $[C']$, which are formed in turn for each boundary bus by filling with zeros an array of dimension equal to the number of buses in the subsystem and placing a +1 in the location corresponding to the boundary bus. The processor in charge of the solution of the cutset equations, in this case the MPU 2, combines the values obtained for $[A']$ in each MPU according to the product $[C'] [A']$ as follows

	j	k	l
j	$x_j - A'_{jj} - A'_{cc}$	$-A'_{ab} - A'_{cd}$	$-A'_{cd}$
k		$x_k - A'_{bb} - A'_{dd}$	$-A'_{dd}$
l			$x_l - A'_{dd} - A'_{ff}$

Following the procedure outlined above the ISM $[X'_{tt}]$ is obtained as shown below:

- Place in the diagonal elements of $[X'_{tt}]$ the reactances of the torn branches

	j	k	l
j	0.2000	0.0000	0.0000
k		0.2000	0.0000
l			0.2000

MPU 2

- Form a null vector for the first boundary bus in each subsystem and place a +1 in the corresponding location

	a
g	0
a	1
b	0

MPU 1

	c
h	0
c	1
d	0

MPU2

	f
e	0
f	1

MPU3

- Obtain the values of the column of $[A']$ required for the formation of the ISM by performing a partial repeat solution

	a
g	*
a	0.0700
b	0.0300

MPU 1

	c
h	*
c	0.0647
d	0.0177

MPU 2

	f
e	*
f	0.0571

MPU 3

* Element not required or calculated

- Send the calculated values to the MPU solving the cutset equations (central MPU) which updates the ISM with the computed and received values

$$[x'_{tt}] = \begin{matrix} & \begin{matrix} j & k & l \end{matrix} \\ \begin{matrix} j \\ k \\ l \end{matrix} & \begin{bmatrix} 0.0653 & -0.0477 & -0.0177 \\ & 0.2000 & 0.0000 \\ & & 0.1429 \end{bmatrix} \end{matrix}$$

MPU 2

The MPU 3 has finished with its calculation of factors for the formation of the ISM $[x'_{tt}]$.

- Form a null vector for the next boundary bus if there is any and place a +1 in the location corresponding to that boundary bus

$$\begin{matrix} & b \\ \begin{matrix} g \\ a \\ b \end{matrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{matrix}$$

MPU 1

$$\begin{matrix} & d \\ \begin{matrix} h \\ c \\ d \end{matrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{matrix}$$

MPU 2

- Then the required values of $[A']$ are

$$\begin{matrix} & b \\ \begin{matrix} g \\ a \\ b \end{matrix} & \begin{bmatrix} * \\ * \\ 0.0700 \end{bmatrix} \end{matrix}$$

MPU 1

$$\begin{matrix} & d \\ \begin{matrix} h \\ c \\ d \end{matrix} & \begin{bmatrix} * \\ * \\ 0.0412 \end{bmatrix} \end{matrix}$$

MPU 2

* Element not required or calculated

- Send the factors calculated in the peripheral MPUs and update the ISM in the central MPU

$$[x'_{tt}] = \begin{matrix} & \begin{matrix} j & k & l \end{matrix} \\ \begin{matrix} j \\ k \\ l \end{matrix} & \begin{bmatrix} 0.0653 & -0.0477 & -0.0177 \\ & 0.0888 & -0.0412 \\ & & 0.1017 \end{bmatrix} \end{matrix}$$

MPU 2

The ISM $[x'_{tt}]$ has been formed

- Factorise $[x'_{tt}]$

$$[L'_t][U'_t] = \begin{matrix} & \begin{matrix} j & k & l \end{matrix} \\ \begin{matrix} j \\ k \\ l \end{matrix} & \begin{bmatrix} 15.3153 & 0.7297 & 0.2703 \\ & 18.5000 & 1.0000 \\ & & 23.3333 \end{bmatrix} \end{matrix}$$

MPU 2

- Form the vector $[\Delta P/|V|]$ for each subsystem.

$$\begin{matrix} g \\ a \\ b \end{matrix} \begin{bmatrix} 0.00 \\ 0.06 \\ -0.01 \end{bmatrix}$$

MPU 1

$$\begin{matrix} h \\ c \\ d \end{matrix} \begin{bmatrix} 0.09 \\ -0.06 \\ -0.06 \end{bmatrix}$$

MPU 2

$$\begin{matrix} e \\ f \end{matrix} \begin{bmatrix} 0.03 \\ -0.02 \end{bmatrix}$$

MPU 3

- Use the substitution procedure S2 (see Table 5.1) to obtain $[\Delta\theta]$

• Solve $[L'] [U'] [\Delta\theta_1] = [\Delta P/|V|]$ for $[\Delta\theta_1]$

$$\begin{matrix} 1/100 \\ g \\ a \\ b \end{matrix} \begin{bmatrix} 0.2500 \\ 0.3900 \\ 0.1100 \end{bmatrix}$$

MPU 1

$$\begin{matrix} 1/100 \\ h \\ c \\ d \end{matrix} \begin{bmatrix} 0.7941 \\ -0.1235 \\ -0.0882 \end{bmatrix}$$

MPU 2

$$\begin{matrix} 1/100 \\ e \\ f \end{matrix} \begin{bmatrix} 0.2857 \\ -0.0286 \end{bmatrix}$$

MPU 3

- Send to the central MPU the values in the locations of boundary buses in the solution vector $[\Delta\theta_1]$. With the received values the central MPU forms the vector $[\Delta\theta_t]$ according to $[\Delta\theta_t] = [C']^t [\Delta\theta_1]$

$$[\Delta\theta_t] = \begin{matrix} j \\ k \\ l \end{matrix} \begin{bmatrix} \Delta\theta_{1a} + \Delta\theta_{1c} \\ \Delta\theta_{1b} + \Delta\theta_{1d} \\ \Delta\theta_{1d} + \Delta\theta_{1f} \end{bmatrix} = 1/100 \begin{bmatrix} 0.2665 \\ 0.0218 \\ -0.1168 \end{bmatrix}$$

MPU 2

- The central MPU solves the cutset equations $[L'_t][U'_t][I'_t] = [\Delta\theta_t]$ for $[I'_t]$

$$[I'_t] = \begin{matrix} j \\ k \\ l \end{matrix} \begin{bmatrix} 0.110 \\ 0.080 \\ 0.040 \end{bmatrix}$$

MPU 2

- The central MPU sends to the peripheral MPUs the values in the locations of $[I'_t]$ corresponding to the branches that were interconnecting the respective subsystems. The MPUs form a vector using the values from the cutset solution according to $[\Delta P_t / |V_t|] = [C'] [I'_t]$

$$\begin{array}{c} g \\ a \\ b \end{array} \begin{array}{|c|} \hline 0 \\ I'_{tj} \\ I'_{tk} \\ \hline \end{array}$$

MPU 1

$$\begin{array}{c} h \\ c \\ d \end{array} \begin{array}{|c|} \hline 0 \\ I'_{tj} \\ I'_{tk} + I'_{t1} \\ \hline \end{array}$$

MPU 2

$$\begin{array}{c} e \\ f \end{array} \begin{array}{|c|} \hline 0 \\ I'_{t1} \\ \hline \end{array}$$

MPU 3

Then $[\Delta P_t / |V_t|]$ is equal to

$$\begin{array}{c} g \\ a \\ b \end{array} \begin{array}{|c|} \hline 0.00 \\ 0.11 \\ 0.08 \\ \hline \end{array}$$

MPU 1

$$\begin{array}{c} h \\ c \\ d \end{array} \begin{array}{|c|} \hline 0.00 \\ 0.11 \\ 0.12 \\ \hline \end{array}$$

MPU 2

$$\begin{array}{c} e \\ f \end{array} \begin{array}{|c|} \hline 0.00 \\ 0.04 \\ \hline \end{array}$$

MPU 3

- Solve $[L'] [U'] [\Delta \theta_2] = [\Delta P_t / |V_t|]$ for $[\Delta \theta_2]$

$$\begin{array}{c} 1/100 \\ g \\ a \\ b \end{array} \begin{array}{|c|} \hline 0.9500 \\ 1.0100 \\ 0.8900 \\ \hline \end{array}$$

MPU 1

$$\begin{array}{c} 1/100 \\ h \\ c \\ d \end{array} \begin{array}{|c|} \hline 0.8059 \\ 0.9235 \\ 0.6882 \\ \hline \end{array}$$

MPU 2

$$\begin{array}{c} 1/100 \\ e \\ f \end{array} \begin{array}{|c|} \hline 0.1143 \\ 0.2286 \\ \hline \end{array}$$

MPU 3

- Obtain the exact voltage angle corrections by $[\Delta \theta] = [\Delta \theta_1] + [\Delta \theta_2]$

$$\begin{array}{c} 1/5 \\ g \\ a \\ b \end{array} \begin{array}{|c|} \hline 0.06 \\ 0.07 \\ 0.05 \\ \hline \end{array}$$

MPU 1

$$\begin{array}{c} 1/5 \\ h \\ c \\ d \end{array} \begin{array}{|c|} \hline 0.08 \\ 0.04 \\ 0.03 \\ \hline \end{array}$$

MPU 2

$$\begin{array}{c} 1/5 \\ e \\ f \end{array} \begin{array}{|c|} \hline 0.02 \\ 0.01 \\ \hline \end{array}$$

MPU 3

These values correspond exactly to the results obtained with the untorn approach.

The formation and factorisation of the matrices $[B'']$ and $[X''_{tt}]$, and the corrections of bus voltage magnitudes are carried out in a similar manner.

APPENDIX C

DESCRIPTION OF TEST SYSTEMS

C.0 - INTRODUCTION

This appendix describes the test systems used for the evaluation of the algorithms implemented for different power system studies. For each of these systems, a descriptive title containing the system characteristics, a schematic diagram, transmission network data and a base-case load flow solution are given.

The transmission network data are given in p.u. on a 100 MVA basis. The type of buses in the systems are specified in the data as follows:

- . Type 1 - Slack bus
- . Type 2 - Controlled voltage bus (PV bus)
- . Type 3 - Load bus (PQ bus)

The base-case load flow solution was obtained on a TMS9900, 16-bit microprocessor using the FDLF method with a tolerance of 0.01 MW/MVAR for convergence. The execution times and the number of iterations required for solution are summarised in Table 4.3.

C.1 - 5-BUS TEST SYSTEM

The 5-bus test system consists of:

- . 2 Generators
- . 0 Synchronous compensators
- . 3 Loads
- . 6 Lines
- . 0 Tap changing transformers
- . 0 Shunt capacitors
- . 0 Shunt reactors

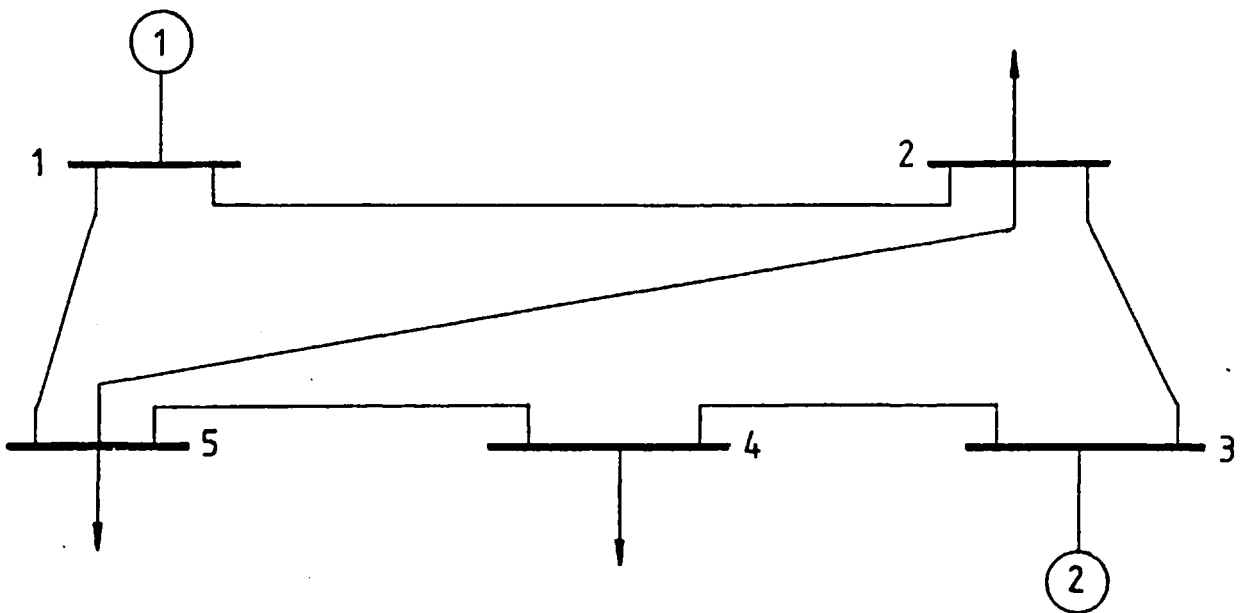


Fig. C.1: 5-bus Test System

TABLE C.1

TRANSMISSION NETWORK DATA
5-BUS TEST SYSTEM

BUS	TO	BUS	R	X	B/2
1		5	0.1500	0.5500	0.0000
1		2	0.0900	0.3800	0.0000
2		5	0.0970	0.4100	0.0000
2		3	0.3000	0.7500	0.0000
3		4	0.1200	0.5100	0.0000
4		5	0.1000	0.4500	0.0000

TABLE C.2

BASE-CASE LOAD FLOW SOLUTION
5-BUS TEST SYSTEM

BUS	TYPE	VOLTAGE	ANGLE	GEN MW	GEN MVAR	LOAD MW	LOAD MVAR
1	1	1.100	0.000	72.180	47.030	0.000	0.000
2	3	0.981	-6.335	0.000	0.000	50.000	10.000
3	2	1.048	3.773	65.000	16.790	0.000	0.000
4	3	0.937	-6.896	0.000	0.000	35.000	10.000
5	3	0.946	-8.098	0.000	0.000	45.000	15.000

SYSTEM GENERATION: 137.180 MW 63.820 MVAR
SYSTEM LOAD: 130.000 MW 35.000 MVAR

C.2 - WSCC 9-BUS TEST SYSTEM

The Western States Coordinating Council (WSCC) 9-bus test system [27] consists of:

- . 3 Generators
- . 0 Synchronous compensators
- . 3 Loads
- . 9 Lines
- . 3 Tap changing transformers
- . 0 Shunt capacitors
- . 0 Shunt reactors

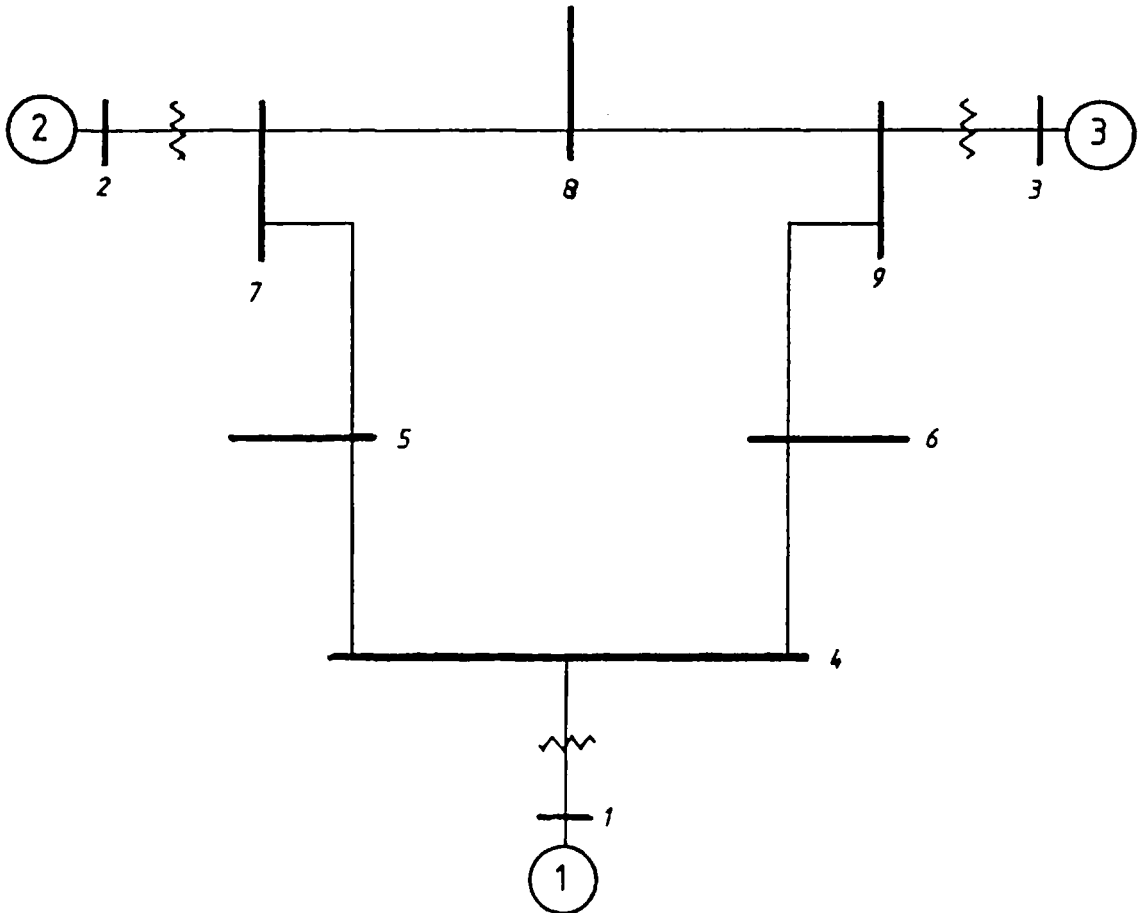


Fig. C.2: WSCC 9-bus Test System

TABLE C.3

TRANSMISSION NETWORK DATA
9-BUS TEST SYSTEM

BUS	TO	BUS	R	X	B/2
1		4	0.0000	0.0576	0.0000
2		7	0.0000	0.0625	0.0000
3		9	0.0000	0.0586	0.0000
4		6	0.0170	0.0920	0.0790
4		5	0.0100	0.0850	0.0880
5		7	0.0320	0.1610	0.1530
6		9	0.0390	0.1700	0.1790
7		8	0.0085	0.0720	0.0745
8		9	0.0119	0.1008	0.1045

TABLE C.3.1

TRANSFORMER		TAP SETTING
2	7	1.000
3	9	1.000
1	4	1.000

TABLE C.4

BASE-CASE LOAD FLOW SOLUTION
9-BUS TEST SYSTEM

BUS	TYPE	VOLTAGE	ANGLE	GEN MW	GEN MVAR	LOAD MW	LOAD MVAR
1	1	1.040	0.000	71.460	27.090	0.000	0.000
2	2	1.012	9.482	163.000	6.700	0.000	0.000
3	2	1.022	4.779	85.000	-10.820	0.000	0.000
4	3	1.025	-2.268	0.000	0.000	0.000	0.000
5	3	0.993	-3.961	0.000	0.000	125.000	50.000
6	3	1.011	-3.717	0.000	0.000	90.000	30.000
7	3	1.024	3.824	0.000	0.000	0.000	0.000
8	3	1.016	0.751	0.000	0.000	100.000	35.000
9	3	1.032	2.034	0.000	0.000	0.000	0.000

SYSTEM GENERATION: 319.460 MW 22.970 MVAR
SYSTEM LOAD: 315.000 MW 115.000 MVAR

C.3 - IEEE 14-BUS TEST SYSTEM

The IEEE 14-bus test system [68] consists of:

- . 2 Generators
- . 3 Synchronous compensators
- . 11 Loads
- . 20 Lines
- . 3 Tap changing transformers
- . 1 Shunt capacitor
- . 0 Shunt reactors

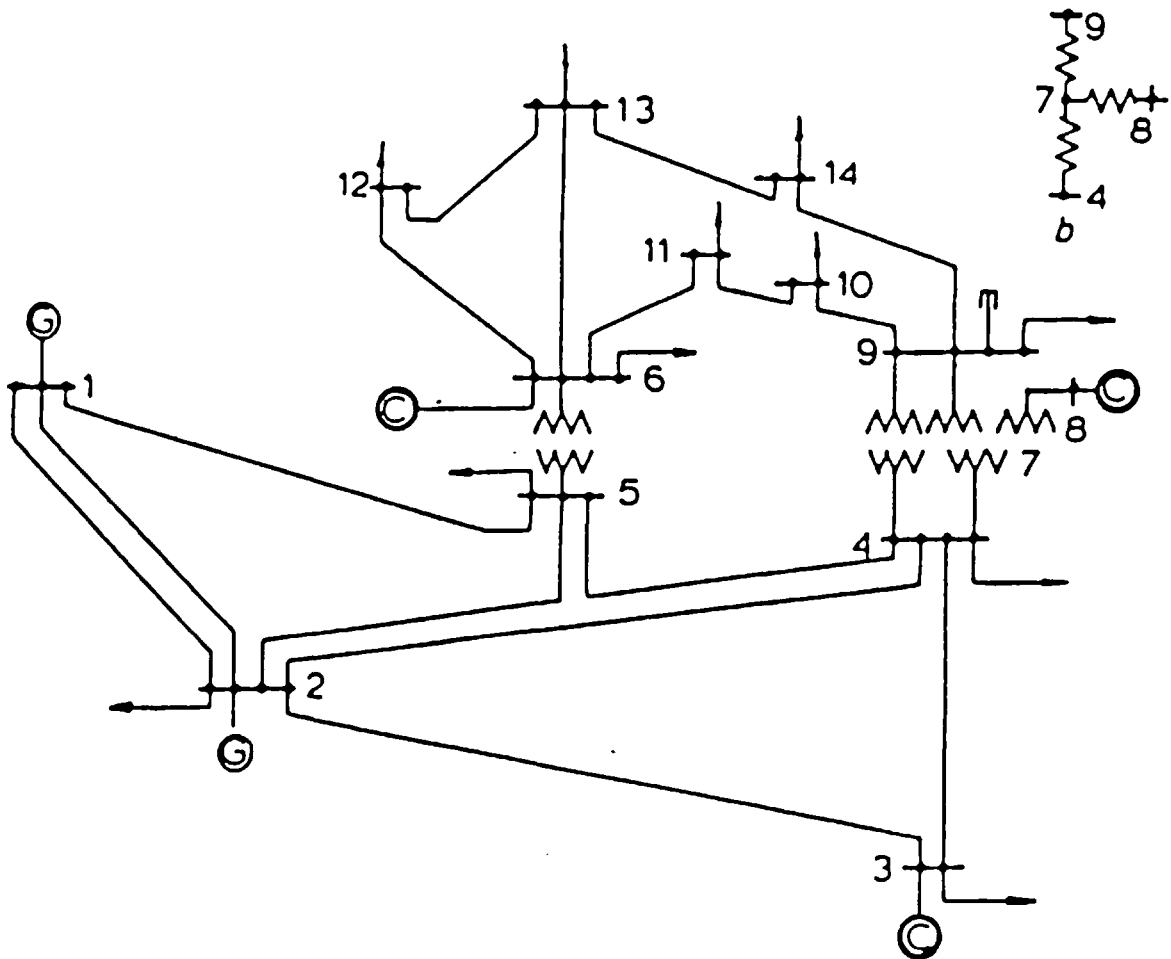


Fig. C.3: IEEE 14-bus Test System

TABLE C.5

TRANSMISSION NETWORK DATA
14-BUS TEST SYSTEM

BUS	TO	BUS	R	X	B/2
1	2		0.0194	0.0592	0.0264
1	5		0.0540	0.2230	0.0246
2	3		0.0470	0.1980	0.0219
2	4		0.0581	0.1763	0.0187
2	5		0.0569	0.1739	0.0170
3	4		0.0670	0.1710	0.0173
4	5		0.0133	0.0421	0.0064
4	7		0.0000	0.2091	0.0000
4	9		0.0000	0.5562	0.0000
5	6		0.0000	0.2520	0.0000
6	11		0.0950	0.1989	0.0000
6	12		0.1229	0.2558	0.0000
6	13		0.0661	0.1303	0.0000
7	8		0.0000	0.1761	0.0000
7	9		0.0000	0.1100	0.0000
9	10		0.0318	0.0845	0.0000
9	14		0.1271	0.2704	0.0000
10	11		0.0820	0.1921	0.0000
12	13		0.2209	0.1999	0.0000
13	14		0.1709	0.3480	0.0000

TABLE C.5.1

TRANSFORMER	TAP	SETTING
4	7	0.978
4	9	0.969
5	6	0.932

TABLE C.5.2

BUS	SHUNT ELEMENT
9	0.190

+ = Capacitor
- = Reactor

TABLE C.6

BASE-CASE LOAD FLOW SOLUTION
14-BUS TEST SYSTEM

BUS	TYPE	VOLTAGE	ANGLE	GEN MW	GEN MVAR	LOAD MW	LOAD MVAR
1	1	1.060	0.000	232.460	-26.400	0.000	0.000
2	2	1.041	-5.161	40.000	20.630	21.700	12.700
3	2	0.986	-12.613	0.000	13.910	94.200	19.000
4	3	1.017	-10.797	0.000	0.000	47.800	-3.900
5	3	1.028	-9.350	0.000	0.000	7.600	1.600
6	2	0.995	-15.325	0.000	44.370	11.200	7.500
7	3	1.010	-14.325	0.000	0.000	0.000	0.000
8	2	1.058	-15.002	0.000	30.460	0.000	0.000
9	3	0.989	-15.882	0.000	0.000	29.500	16.600
10	3	0.981	-15.972	0.000	0.000	9.000	5.800
11	3	0.984	-15.744	0.000	0.000	3.500	1.800
12	3	0.975	-16.028	0.000	0.000	6.100	1.600
13	3	0.972	-15.999	0.000	0.000	13.500	5.800
14	3	0.958	-16.667	0.000	0.000	14.900	5.000

SYSTEM GENERATION: 272.460 MW 82.970 MVAR
SYSTEM LOAD: 259.000 MW 73.500 MVAR

C.4 - IEEE 24-BUS RELIABILITY TEST SYSTEM

The IEEE 24-bus reliability test system [67] consists of:

- . 10 Generators
- . 1 Synchronous compensators
- . 17 Loads
- . 34 Lines of 138 & 230 kV
- . 5 Tap changing transformers
- . 0 Shunt capacitors
- . 1 Shunt reactors

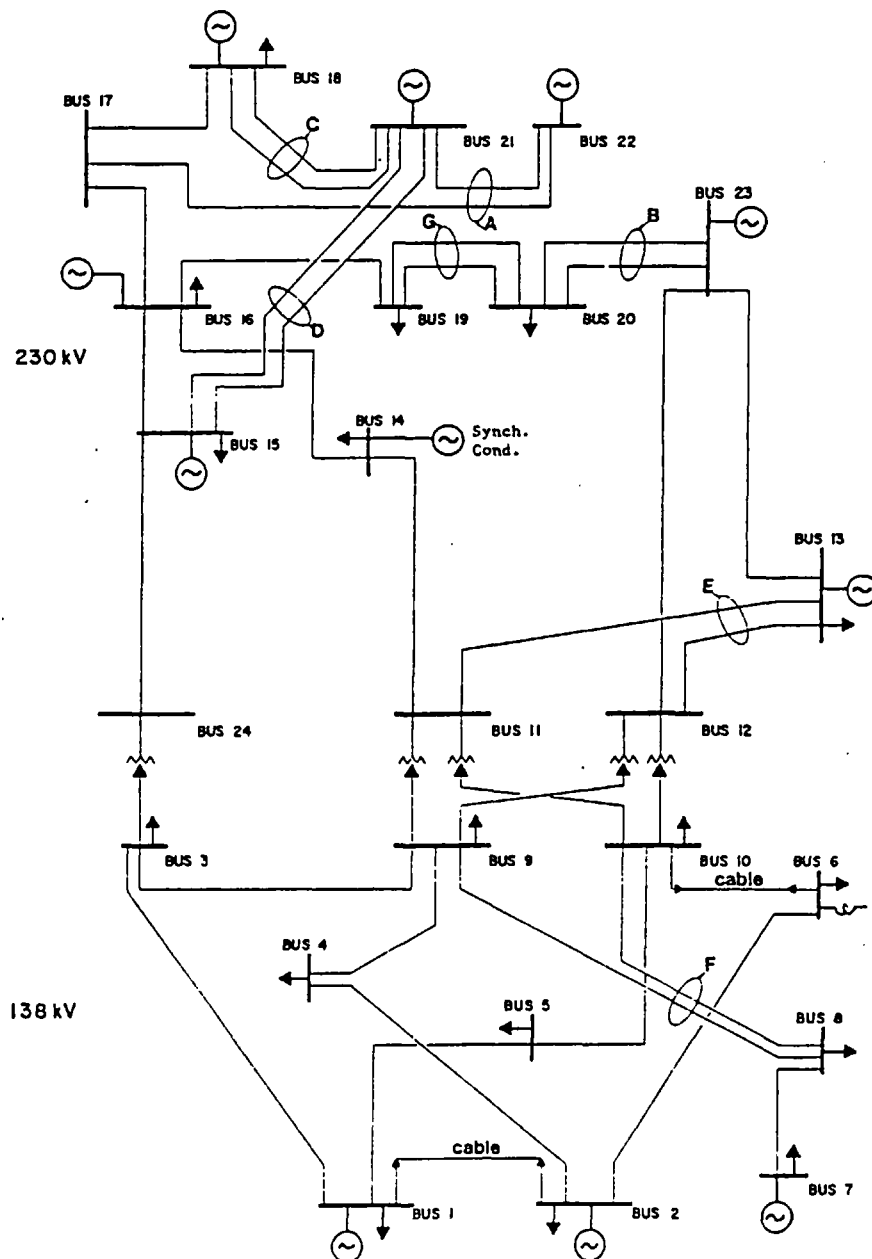


Fig. C.4: IEEE 24-bus Reliability Test System

TABLE C.7

TRANSMISSION NETWORK DATA
24-BUS TEST SYSTEM

BUS	TO	BUS	R	X	B/2
18		20	0.0026	0.0139	0.2306
18		24	0.0546	0.2112	0.0286
6		18	0.0218	0.0845	0.0115
5		20	0.0328	0.1267	0.0172
20		22	0.0497	0.1920	0.0260
19		24	0.0308	0.1190	0.0161
13		24	0.0023	0.0839	0.0000
5		19	0.0268	0.1037	0.0141
6		21	0.0228	0.0883	0.0120
21		22	0.0139	0.0605	1.2295
13		15	0.0067	0.0519	0.0546
2		7	0.0159	0.0614	0.0083
7		19	0.0427	0.1651	0.0224
7		21	0.0427	0.1651	0.0223
17		19	0.0023	0.0839	0.0000
14		19	0.0023	0.0839	0.0000
17		21	0.0023	0.0839	0.0000
14		21	0.0023	0.0839	0.0000
8		17	0.0061	0.0476	0.0499
9		17	0.0054	0.0418	0.0879
8		14	0.0061	0.0476	0.0499
1		14	0.0124	0.0966	0.1015
1		8	0.0111	0.0865	0.0909
9		23	0.0050	0.0389	0.0409
15		23	0.0022	0.0173	0.0182
12		15	0.0031	0.0123	0.1030
16		23	0.0033	0.0259	0.0273
4		23	0.0030	0.0230	0.0243
10		16	0.0018	0.0144	0.0152
11		16	0.0135	0.1053	0.1106
10		12	0.0016	0.0129	0.0545
3		4	0.0025	0.0198	0.0833
1		3	0.0014	0.0108	0.0455
11		12	0.0087	0.0678	0.0712

TABLE C.7.1

TRANSFORMER	TAP	SETTING
18	13	1.000
18	16	1.000
20	13	1.000
20	16	1.000
23	12	1.000

TABLE C.7.2

BUS	SHUNT ELEMENT
22	-1.000
+ = Capacitor	
- = Reactor	

TABLE C.8

BASE-CASE LOAD FLOW SOLUTION
24-BUS TEST SYSTEM

BUS	TYPE	VOLTAGE	ANGLE	GEN MW	GEN MVAR	LOAD MW	LOAD MVAR
1	1	1.050	0.000	660.900	41.380	0.000	0.000
2	2	0.914	-28.468	136.000	88.940	125.000	25.000
3	3	1.042	-1.850	0.000	0.000	128.000	26.000
4	3	1.034	-3.879	0.000	0.000	181.000	37.000
5	3	0.919	-22.605	0.000	0.000	74.000	15.000
6	3	0.936	-23.615	0.000	0.000	71.000	14.000
7	3	0.880	-27.202	0.000	0.000	171.000	35.000
8	2	1.035	-6.069	594.000	97.620	265.000	54.000
9	2	1.024	-10.469	0.000	110.210	194.000	39.000
10	2	1.040	-0.391	360.000	56.240	333.000	68.000
11	2	1.034	6.471	300.000	-40.720	0.000	0.000
12	2	1.040	-0.005	360.000	-105.600	0.000	0.000
13	3	0.997	-9.152	0.000	0.000	0.000	0.000
14	3	0.998	-11.053	0.000	0.000	0.000	0.000
15	2	1.038	-3.411	122.000	214.730	317.000	64.000
16	3	1.040	-1.039	0.000	0.000	0.000	0.000
17	3	1.000	-12.508	0.000	0.000	0.000	0.000
18	2	0.959	-23.060	122.000	33.750	108.000	22.000
19	3	0.946	-19.014	0.000	0.000	175.000	36.000
20	2	0.958	-23.153	122.000	30.810	97.000	20.000
21	3	0.960	-21.712	0.000	0.000	195.000	40.000
22	3	0.916	-24.724	0.000	0.000	136.000	37.000
23	2	1.038	-4.021	124.000	43.710	100.000	20.000
24	3	0.942	-18.651	0.000	0.000	180.000	28.000

SYSTEM GENERATION: 2900.897 MW 571.069 MVAR
SYSTEM LOAD: 2849.996 MW 579.998 MVAR

C.5 - IEEE 30-BUS TEST SYSTEM

The IEEE 30-bus test system [68] consists of:

.	2	Generators
.	3	Synchronous compensators
.	21	Loads
.	41	Lines of 138 & 69 kV
.	4	Tap changing transformers
.	2	Shunt capacitors
.	0	Shunt reactors

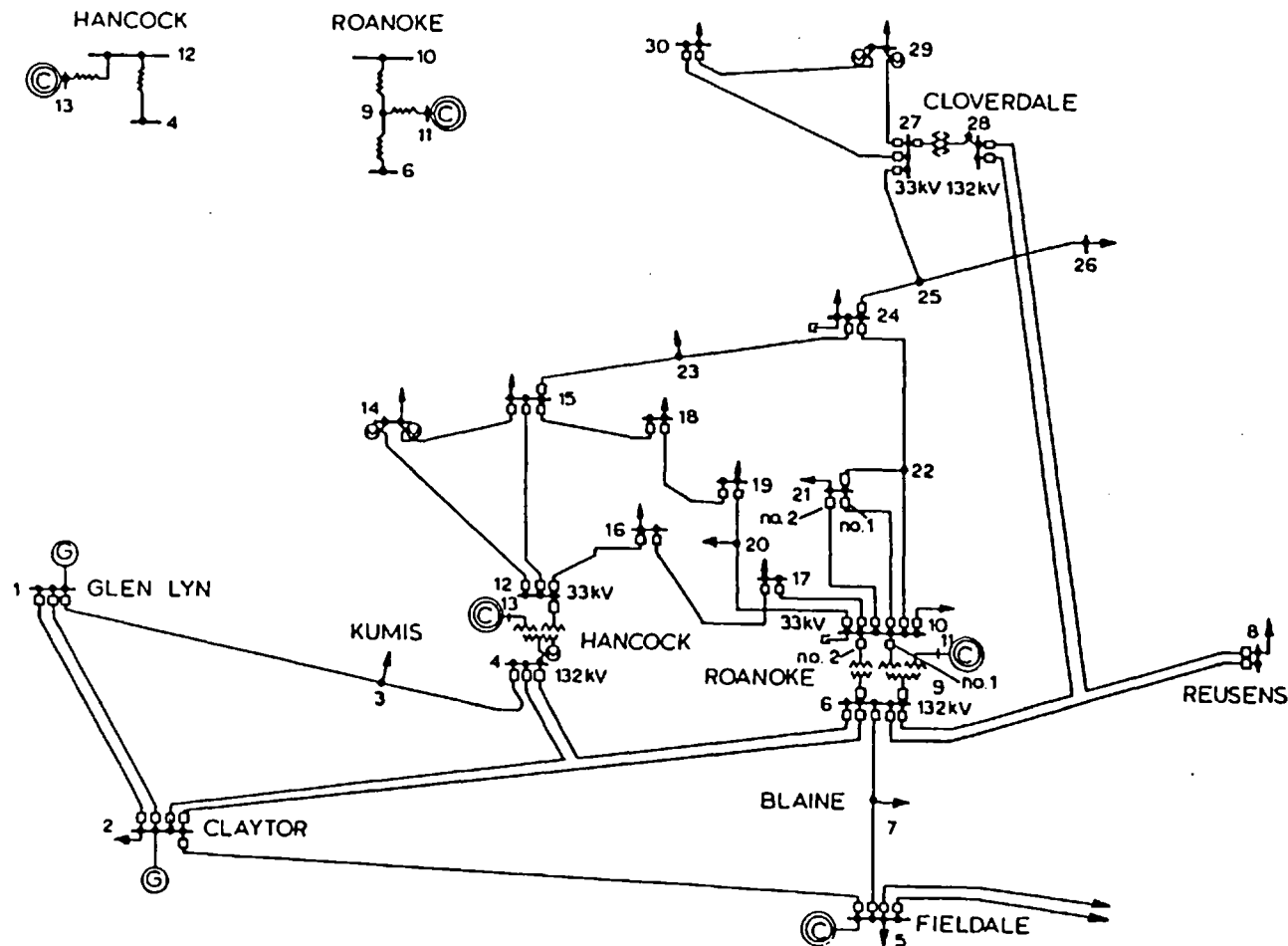


Fig. C.5: IEEE 30-bus Test System

TABLE C.9

TRANSMISSION NETWORK DATA
30-BUS TEST SYSTEM

BUS	TO BUS	R	X	B/2
1	2	0.0192	0.0575	0.0264
1	3	0.0452	0.1852	0.0204
2	4	0.0570	0.1737	0.0184
3	4	0.0132	0.0379	0.0042
2	5	0.0472	0.1983	0.0209
2	6	0.0581	0.1763	0.0187
4	6	0.0119	0.0414	0.0045
5	7	0.0460	0.1160	0.0102
6	7	0.0267	0.0820	0.0085
6	8	0.0120	0.0420	0.0045
6	9	0.0000	0.2080	0.0000
6	10	0.0000	0.5560	0.0000
9	11	0.0000	0.2080	0.0000
9	10	0.0000	0.1100	0.0000
4	12	0.0000	0.2560	0.0000
12	13	0.0000	0.1400	0.0000
12	14	0.1231	0.2559	0.0000
12	15	0.0662	0.1304	0.0000
12	16	0.0945	0.1987	0.0000
14	15	0.2210	0.1997	0.0000
16	17	0.0824	0.1923	0.0000
15	18	0.1070	0.2185	0.0000
18	19	0.0639	0.1292	0.0000
19	20	0.0340	0.0680	0.0000
10	20	0.0936	0.2090	0.0000
10	17	0.0324	0.0845	0.0000
10	21	0.0348	0.0749	0.0000
10	22	0.0727	0.1499	0.0000
21	22	0.0116	0.0236	0.0000
15	23	0.1000	0.2020	0.0000
22	24	0.1150	0.1790	0.0000
23	24	0.1320	0.2700	0.0000
24	25	0.1885	0.3292	0.0000
25	26	0.2544	0.3800	0.0000
25	27	0.1093	0.2087	0.0000
27	28	0.0000	0.3960	0.0000
27	29	0.2198	0.4153	0.0000
27	30	0.3202	0.6027	0.0000
29	30	0.2399	0.4533	0.0000
8	28	0.0636	0.2000	0.0214
6	28	0.0169	0.0599	0.0065

TABLE C.9.1

TRANSFORMER	TAP SETTING
4	12
6	9
6	10
28	27

TABLE C.9.2

BUS	SHUNT ELEMENT
10	0.190
24	0.043

+ = Capacitor
- = Reactor

TABLE C.10

BASE-CASE LOAD FLOW SOLUTION
30-BUS TEST SYSTEM

BUS	TYPE	VOLTAGE	ANGLE	GEN MW	GEN MVAR	LOAD MW	LOAD MVAR
1	1	1.060	0.000	261.080	-26.890	0.000	0.000
2	2	1.040	-5.746	40.000	44.610	21.700	12.700
3	3	1.022	-8.361	0.000	0.000	2.400	1.200
4	3	1.012	-10.007	0.000	0.000	7.600	1.600
5	2	0.979	-14.309	0.000	32.140	94.200	19.000
6	3	0.997	-11.552	0.000	0.000	0.000	0.000
7	3	0.980	-13.111	0.000	0.000	22.800	10.900
8	2	0.988	-12.088	0.000	16.920	30.000	30.000
9	3	0.988	-15.085	0.000	0.000	0.000	0.000
10	3	0.964	-16.622	0.000	0.000	5.800	2.000
11	2	1.045	-15.967	0.000	31.070	0.000	0.000
12	3	0.972	-16.111	0.000	0.000	11.200	7.500
13	2	1.029	-17.048	0.000	45.040	0.000	0.000
14	3	0.953	-16.799	0.000	0.000	6.200	1.600
15	3	0.948	-16.742	0.000	0.000	8.200	2.500
16	3	0.960	-16.496	0.000	0.000	3.500	1.800
17	3	0.956	-16.709	0.000	0.000	9.000	5.800
18	3	0.938	-17.196	0.000	0.000	3.200	0.900
19	3	0.936	-17.325	0.000	0.000	9.500	3.400
20	3	0.942	-17.200	0.000	0.000	2.200	0.700
21	3	0.948	-16.841	0.000	0.000	17.500	11.200
22	3	0.948	-16.830	0.000	0.000	0.000	0.000
23	3	0.936	-16.884	0.000	0.000	3.200	1.600
24	3	0.930	-16.868	0.000	0.000	8.700	6.700
25	3	0.924	-16.143	0.000	0.000	0.000	0.000
26	3	0.903	-16.211	0.000	0.000	3.500	2.300
27	3	0.930	-15.582	0.000	0.000	0.000	0.000
28	3	0.991	-12.096	0.000	0.000	0.000	0.000
29	3	0.903	-16.481	0.000	0.000	2.400	0.900
30	3	0.886	-17.148	0.000	0.000	10.600	1.900

SYSTEM GENERATION: 301.080 MW 142.890 MVAR
SYSTEM LOAD: 283.400 MW 126.200 MVAR

C.6 - NEW ENGLAND 39-BUS TEST SYSTEM

The New England 39-bus test system [100] consists of:

•	10	Generators
•	0	Synchronous compensators
•	19	Loads
•	46	Lines of 345 kV
•	12	Tap changing transformers
•	0	Shunt capacitors
•	0	Shunt reactors

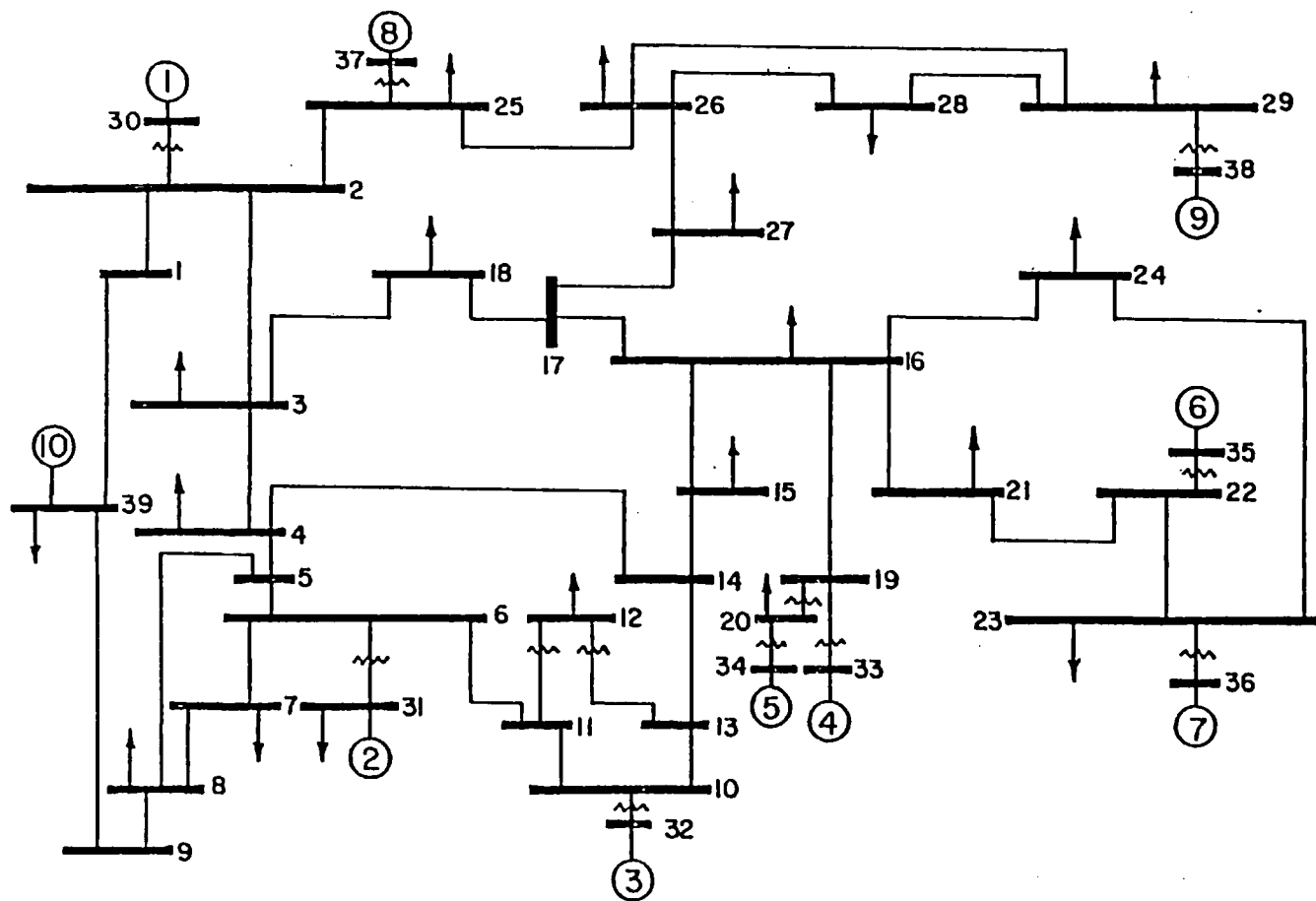


Fig. C.6: New England 39-bus Test System

TABLE C.11

TRANSMISSION NETWORK DATA
39-BUS TEST SYSTEM

BUS	TO BUS	R	X	B/2
1	2	0.0035	0.0411	0.3493
1	39	0.0010	0.0250	0.3750
2	3	0.0013	0.0151	0.1286
2	25	0.0070	0.0086	0.0730
3	4	0.0013	0.0213	0.1107
3	18	0.0011	0.0133	0.1069
4	5	0.0008	0.0128	0.0671
4	14	0.0008	0.0129	0.0691
5	6	0.0002	0.0026	0.0217
5	8	0.0008	0.0112	0.0738
6	7	0.0006	0.0092	0.0565
6	11	0.0007	0.0082	0.0695
7	8	0.0004	0.0046	0.0390
8	9	0.0023	0.0363	0.1902
9	39	0.0010	0.0250	0.6000
10	11	0.0004	0.0043	0.0364
10	13	0.0004	0.0043	0.0364
13	14	0.0009	0.0101	0.0861
14	15	0.0018	0.0217	0.1830
15	16	0.0009	0.0094	0.0855
16	17	0.0007	0.0089	0.0671
16	19	0.0016	0.0195	0.1520
16	21	0.0008	0.0135	0.1274
16	24	0.0003	0.0059	0.0340
17	18	0.0007	0.0082	0.0659
17	27	0.0013	0.0173	0.1608
21	22	0.0008	0.0140	0.1282
22	23	0.0006	0.0096	0.0923
23	24	0.0022	0.0350	0.1805
25	26	0.0032	0.0323	0.2565
26	27	0.0014	0.0147	0.1198
26	28	0.0043	0.0474	0.3901
26	29	0.0057	0.0625	0.5145
28	29	0.0014	0.0151	0.1245
11	12	0.0016	0.0435	0.0000
12	13	0.0016	0.0435	0.0000
6	31	0.0000	0.0250	0.0000
10	32	0.0000	0.0200	0.0000
19	33	0.0007	0.0142	0.0000
20	34	0.0009	0.0180	0.0000
22	35	0.0000	0.0143	0.0000
23	36	0.0005	0.0272	0.0000
25	37	0.0006	0.0232	0.0000
2	30	0.0000	0.0181	0.0000
29	38	0.0008	0.0156	0.0000
19	20	0.0007	0.0138	0.0000

TABLE C.11.1

TRANSFORMER	TAP SETTING
12	11
12	13
6	31
10	32
19	33
20	34
22	35
23	36
25	37
2	30
29	38
19	20

TABLE C.12

BASE-CASE LOAD FLOW SOLUTION
39-BUS TEST SYSTEM

BUS	TYPE	VOLTAGE	ANGLE	GEN MW	GEN MVAR	LOAD MW	LOAD MVAR
1	3	1.033	-9.983	0.000	0.000	0.000	0.000
2	3	1.041	-7.338	0.000	0.000	0.000	0.000
3	3	1.015	-10.112	0.000	0.000	322.000	2.400
4	3	0.986	-10.638	0.000	0.000	500.000	184.000
5	3	0.991	-9.477	0.000	0.000	0.000	0.000
6	3	0.995	-8.801	0.000	0.000	0.000	0.000
7	3	0.978	-10.871	0.000	0.000	233.800	84.000
8	3	0.976	-11.355	0.000	0.000	522.000	176.000
9	3	1.008	-11.547	0.000	0.000	0.000	0.000
10	3	1.011	-6.481	0.000	0.000	0.000	0.000
11	3	1.004	-7.270	0.000	0.000	0.000	0.000
12	3	0.992	-7.195	0.000	0.000	8.500	88.000
13	3	1.006	-7.182	0.000	0.000	0.000	0.000
14	3	1.000	-8.837	0.000	0.000	0.000	0.000
15	3	1.002	-9.291	0.000	0.000	320.000	153.000
16	3	1.022	-8.008	0.000	0.000	329.400	32.300
17	3	1.022	-9.042	0.000	0.000	0.000	0.000
18	3	1.017	-9.874	0.000	0.000	158.000	30.000
19	3	1.048	-3.313	0.000	0.000	0.000	0.000
20	3	0.988	-4.522	0.000	0.000	680.000	103.000
21	3	1.027	-5.540	0.000	0.000	274.000	115.000
22	3	1.049	-0.973	0.000	0.000	0.000	0.000
23	3	1.044	-1.175	0.000	0.000	247.500	84.600
24	3	1.028	-7.927	0.000	0.000	308.600	-92.200
25	3	1.052	-5.970	0.000	0.000	224.000	47.200
26	3	1.044	-7.251	0.000	0.000	139.000	17.000
27	3	1.025	-9.222	0.000	0.000	281.000	75.500
28	3	1.048	-3.560	0.000	0.000	206.000	27.600
29	3	1.049	-0.662	0.000	0.000	283.300	26.900
30	2	1.044	-4.804	250.000	143.360	0.000	0.000
31	1	0.982	0.000	573.060	207.490	9.200	4.600
32	2	0.983	1.585	650.000	205.850	0.000	0.000
33	2	0.996	2.055	632.000	109.270	0.000	0.000
34	2	1.012	0.631	508.000	166.820	0.000	0.000
35	2	1.046	4.231	650.000	211.330	0.000	0.000
36	2	1.056	7.141	560.000	99.950	0.000	0.000
37	2	1.028	1.160	540.000	2.210	0.000	0.000
38	2	1.020	6.593	830.000	22.400	0.000	0.000
39	2	1.011	-11.376	1000.000	88.470	1103.999	250.000

SYSTEM GENERATION: 6193.050 MW 1257.148 MVAR
SYSTEM LOAD: 6150.290 MW 1408.896 MVAR

C.7 - IEEE 57-BUS TEST SYSTEM

The IEEE 57-bus test system [68] consists of:

- . 4 Generators
- . 3 Synchronous compensators
- . 42 Loads
- . 78 Lines
- . 15 Tap changing transformers
- . 3 Shunt capacitors
- . 0 Shunt reactors

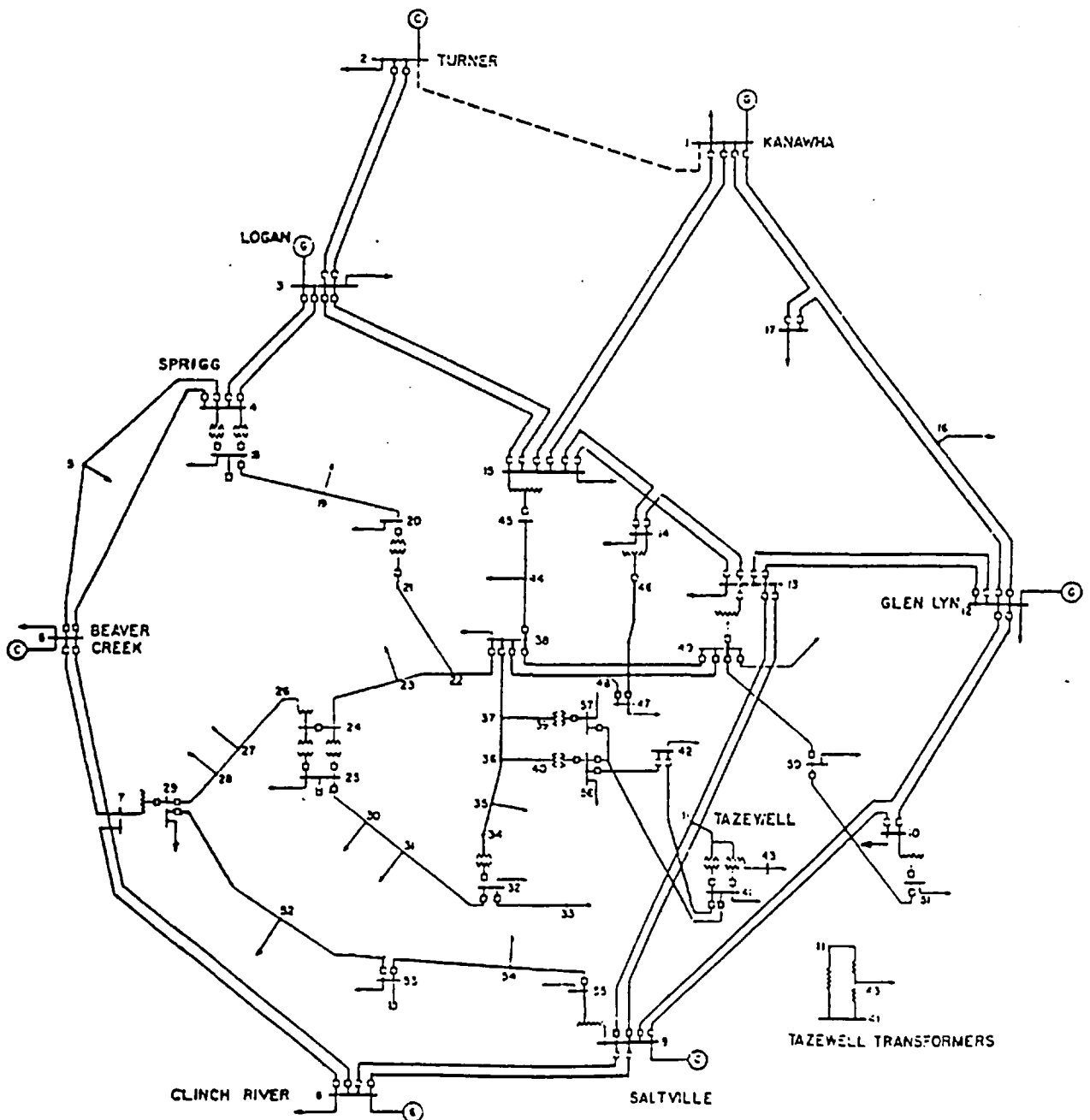


Fig. C.7: IEEE 57-bus Test System

TABLE C.13

TRANSMISSION NETWORK DATA
57-BUS TEST SYSTEM

BUS	TO	BUS	R	X	B/2
1	2		0.0083	0.0280	0.0645
2	3		0.0298	0.0850	0.0409
3	4		0.0112	0.0366	0.0190
4	5		0.0625	0.1320	0.0129
4	6		0.0430	0.1480	0.0174
6	7		0.0200	0.1020	0.0138
6	8		0.0339	0.1730	0.0235
8	9		0.0099	0.0505	0.0274
9	10		0.0369	0.1679	0.0220
9	11		0.0258	0.0848	0.0109
9	12		0.0648	0.2950	0.0386
9	13		0.0481	0.1580	0.0203
13	14		0.0132	0.0434	0.0055
13	15		0.0269	0.0869	0.0115
1	15		0.0178	0.0910	0.0494
1	16		0.0454	0.2060	0.0273
1	17		0.0238	0.1080	0.0143
3	15		0.0162	0.0530	0.0272
4	18		0.0000	0.5550	0.0000
5	6		0.0302	0.0641	0.0062
7	8		0.0139	0.0712	0.0097
10	12		0.0277	0.1262	0.0164
11	13		0.0223	0.0732	0.0094
12	13		0.0178	0.0580	0.0302
12	16		0.0180	0.0813	0.0108
12	17		0.0397	0.1790	0.0238
14	15		0.0171	0.0547	0.0074
18	19		0.4610	0.6850	0.0000
19	20		0.2830	0.4340	0.0000
20	21		0.0000	0.7767	0.0000
21	22		0.0736	0.1170	0.0000
22	23		0.0099	0.0152	0.0000
23	24		0.1660	0.2560	0.0042
24	25		0.0000	1.1820	0.0000
24	26		0.0000	0.0473	0.0000
26	27		0.1650	0.2540	0.0000
27	28		0.0618	0.0954	0.0000
28	29		0.0418	0.0587	0.0000
7	29		0.0000	0.0648	0.0000
25	30		0.1350	0.2020	0.0000
30	31		0.3260	0.4970	0.0000
31	32		0.5070	0.7550	0.0000
32	33		0.0392	0.0360	0.0000
32	34		0.0000	0.9530	0.0000
34	35		0.0520	0.0780	0.0016
35	36		0.0430	0.0537	0.0008
36	37		0.0290	0.0366	0.0000
37	38		0.0651	0.1009	0.0010
37	39		0.0239	0.0379	0.0000
36	40		0.0300	0.0466	0.0000
22	38		0.0192	0.0295	0.0000
11	41		0.0000	0.7490	0.0000
41	42		0.2070	0.3520	0.0000
41	43		0.0000	0.4120	0.0000
38	44		0.0289	0.0585	0.0010
15	45		0.0000	0.1042	0.0000
14	46		0.0000	0.0735	0.0000
46	47		0.0230	0.0680	0.0016
47	48		0.0182	0.0233	0.0000

TABLE C.13

(Cont.)

48	49	0.0834	0.1290	0.0024
49	50	0.0801	0.1280	0.0000
50	51	0.1386	0.2200	0.0000
10	51	0.0000	0.0712	0.0000
13	49	0.0000	0.1910	0.0000
29	52	0.1442	0.1870	0.0000
52	53	0.0762	0.0984	0.0000
53	54	0.1873	0.2320	0.0000
54	55	0.1732	0.2265	0.0000
11	43	0.0000	0.1530	0.0000
44	45	0.0624	0.1242	0.0020
40	56	0.0000	1.1950	0.0000
41	56	0.5530	0.5490	0.0000
42	56	0.2125	0.3540	0.0000
39	57	0.0000	1.3550	0.0000
56	57	0.1740	0.2600	0.0000
38	49	0.1150	0.1770	0.0030
38	48	0.0132	0.0482	0.0000
9	55	0.0000	0.1205	0.0000

TABLE C.13.1

TRANSFORMER		TAP SETTING
4	18	0.970
7	29	0.967
9	55	0.940
10	51	0.930
11	41	0.955
11	43	0.958
13	49	0.895
14	46	0.900
15	45	0.955
21	20	1.043
24	25	1.000
24	26	1.043
34	32	0.975
39	57	0.980
40	56	0.958

TABLE C.13.2

BUS	SHUNT ELEMENT
18	0.100
25	0.059
53	0.063

+ = Capacitor
- = Reactor

TABLE C.14

BASE-CASE LOAD FLOW SOLUTION
57-BUS TEST SYSTEM

BUS	TYPE	VOLTAGE	ANGLE	GEN MW	GEN MVAR	LOAD MW	LOAD MVAR
1	1	1.040	0.000	479.040	130.420	55.000	17.000
2	2	1.010	-1.207	0.000	-0.730	3.000	38.000
3	2	0.980	-5.915	40.000	3.420	41.000	21.000
4	3	0.972	-7.212	0.000	0.000	0.000	0.000
5	3	0.965	-8.410	0.000	0.000	13.000	4.000
6	2	0.968	-8.592	0.000	5.450	75.000	2.000
7	3	0.971	-7.593	0.000	0.000	0.000	0.000
8	2	1.002	-4.613	450.000	67.550	150.000	22.000
9	2	0.966	-9.403	0.000	1.100	121.000	26.000
10	3	0.961	-11.285	0.000	0.000	5.000	2.000
11	3	0.956	-9.837	0.000	0.000	0.000	0.000
12	2	0.998	-10.582	310.000	125.580	377.000	24.000
13	3	0.969	-9.592	0.000	0.000	18.000	2.300
14	3	0.963	-9.034	0.000	0.000	10.500	5.300
15	3	0.979	-7.058	0.000	0.000	22.000	5.000
16	3	1.001	-8.951	0.000	0.000	43.000	3.000
17	3	1.013	-5.487	0.000	0.000	42.000	8.000
18	3	0.921	-11.559	0.000	0.000	27.200	9.800
19	3	0.871	-12.551	0.000	0.000	3.300	0.600
20	3	0.855	-12.599	0.000	0.000	2.300	0.100
21	3	0.816	-11.876	0.000	0.000	0.000	0.000
22	3	0.815	-11.814	0.000	0.000	0.000	0.000
23	3	0.813	-11.849	0.000	0.000	6.300	2.100
24	3	0.801	-11.830	0.000	0.000	0.000	0.000
25	3	0.728	-16.337	0.000	0.000	6.300	3.200
26	3	0.839	-12.030	0.000	0.000	0.000	0.000
27	3	0.877	-11.057	0.000	0.000	9.300	0.500
28	3	0.900	-10.240	0.000	0.000	4.600	2.300
29	3	0.919	-9.687	0.000	0.000	17.000	2.600
30	3	0.699	-16.907	0.000	0.000	3.600	1.800
31	3	0.657	-16.892	0.000	0.000	5.800	2.900
32	3	0.675	-16.346	0.000	0.000	1.600	0.800
33	3	0.672	-16.349	0.000	0.000	3.800	1.900
34	3	0.763	-12.724	0.000	0.000	0.000	0.000
35	3	0.772	-12.610	0.000	0.000	6.000	3.000
36	3	0.785	-12.503	0.000	0.000	0.000	0.000
37	3	0.794	-12.346	0.000	0.000	0.000	0.000
38	3	0.819	-11.730	0.000	0.000	14.000	7.000
39	3	0.792	-12.382	0.000	0.000	0.000	0.000
40	3	0.784	-12.596	0.000	0.000	0.000	0.000
41	3	0.840	-13.657	0.000	0.000	6.300	3.000
42	3	0.783	-14.080	0.000	0.000	7.100	4.400
43	3	0.894	-10.728	0.000	0.000	2.000	0.100
44	3	0.842	-11.172	0.000	0.000	12.000	1.800
45	3	0.904	-9.241	0.000	0.000	0.000	0.000
46	3	0.850	-9.932	0.000	0.000	0.000	0.000
47	3	0.825	-11.303	0.000	0.000	29.700	11.600
48	3	0.823	-11.451	0.000	0.000	0.000	0.000
49	3	0.826	-11.634	0.000	0.000	18.000	8.500
50	3	0.819	-12.128	0.000	0.000	21.000	10.500
51	3	0.874	-11.781	0.000	0.000	18.000	5.300
52	3	0.869	-10.872	0.000	0.000	4.900	2.200
53	3	0.849	-11.355	0.000	0.000	20.000	10.000
54	3	0.867	-10.829	0.000	0.000	4.100	1.400
55	3	0.896	-9.999	0.000	0.000	6.800	3.400
56	3	0.765	-14.067	0.000	0.000	7.600	2.200
57	3	0.750	-14.397	0.000	0.000	6.700	2.000

SYSTEM GENERATION: 1279.039 MW 332.850 MVAR
SYSTEM LOAD: 1250.795 MW 336.397 MVAR

C.8 - IEEE 118-BUS TEST SYSTEM

The IEEE 118-bus test system [100] consists of:

- . 33 Generators
- . 21 Synchronous compensators
- . 91 Loads
- . 179 Lines of 345 & 138 kV
- . 9 Tap changing transformers
- . 12 Shunt capacitors
- . 2 Shunt reactors

Fig. C.8: IEEE 118-bus Test System

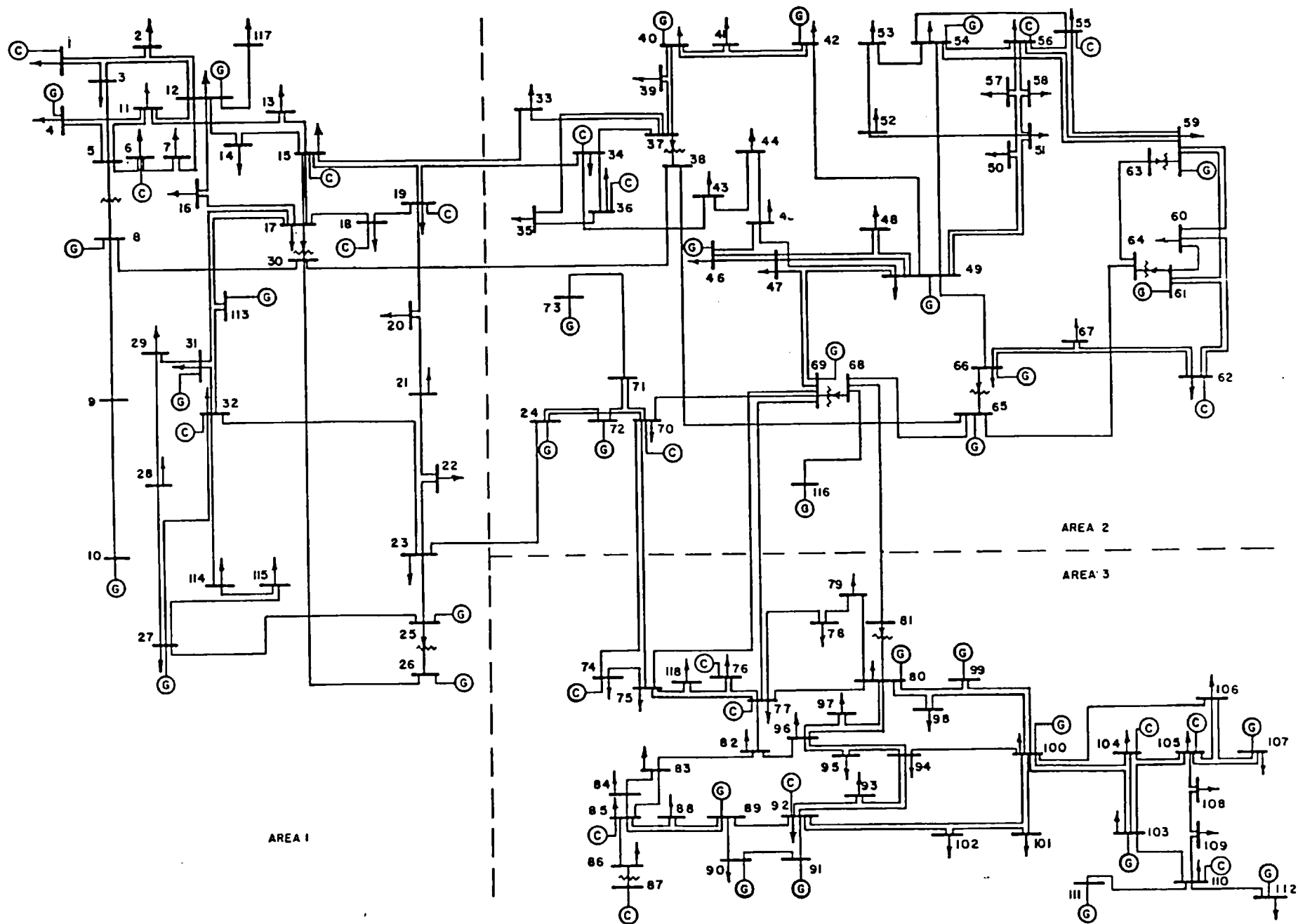


TABLE C.15

TRANSMISSION NETWORK DATA
118-BUS TEST SYSTEM

BUS	TO	BUS	R	X	B/2
1	2		0.0303	0.0999	0.0127
1	3		0.0129	0.0424	0.0054
2	12		0.0187	0.0616	0.0078
3	5		0.0241	0.1080	0.0142
3	12		0.0484	0.1600	0.0203
4	11		0.0209	0.0688	0.0087
4	5		0.0018	0.0080	0.0010
5	11		0.0203	0.0682	0.0087
5	6		0.0119	0.0540	0.0071
6	7		0.0045	0.0208	0.0027
7	12		0.0086	0.0340	0.0043
8	30		0.0043	0.0504	0.2570
8	8		0.0000	0.0267	0.0000
8	9		0.0024	0.0305	0.5810
9	10		0.0026	0.0322	0.6150
11	12		0.0059	0.0196	0.0025
11	13		0.0222	0.0731	0.0094
12	16		0.0212	0.0834	0.0107
12	117		0.0329	0.1400	0.0179
12	14		0.0215	0.0707	0.0091
13	15		0.0744	0.2444	0.0313
14	15		0.0595	0.1950	0.0251
15	17		0.0132	0.0437	0.0222
15	19		0.0120	0.0394	0.0050
15	33		0.0380	0.1244	0.0160
16	17		0.0454	0.1801	0.0233
17	113		0.0091	0.0301	0.0038
17	18		0.0123	0.0505	0.0064
17	31		0.0474	0.1563	0.0200
18	19		0.0111	0.0493	0.0057
19	20		0.0252	0.1170	0.0149
19	34		0.0752	0.2470	0.0316
20	21		0.0183	0.0849	0.0108
21	22		0.0209	0.0970	0.0123
22	23		0.0342	0.1590	0.0202
23	32		0.0317	0.1153	0.0586
23	24		0.0135	0.0492	0.0249
23	25		0.0156	0.0800	0.0432
24	70		0.1022	0.4115	0.0510
24	72		0.0488	0.1960	0.0244
25	27		0.0318	0.1630	0.0882
25	26		0.0000	0.0382	0.0000
26	30		0.0079	0.0860	0.4540
27	32		0.0229	0.0755	0.0096
27	115		0.0164	0.0741	0.0099
27	28		0.0191	0.0855	0.0108
28	29		0.0237	0.0943	0.0119
29	31		0.0108	0.0331	0.0041
30	38		0.0046	0.0540	0.2110
17	30		0.0000	0.0388	0.0000
31	32		0.0298	0.0985	0.0125
32	113		0.0615	0.2030	0.0259
32	114		0.0135	0.0612	0.0081
33	37		0.0415	0.1420	0.0183
34	36		0.0087	0.0268	0.0028
34	37		0.0026	0.0094	0.0049
34	43		0.0413	0.1681	0.0211
35	36		0.0022	0.0102	0.0013
35	37		0.0110	0.0497	0.0066
37	39		0.0321	0.1060	0.0135
37	40		0.0593	0.1680	0.0210
37	38		0.0000	0.0375	0.0000
38	65		0.0090	0.0986	0.5230
39	40		0.0184	0.0605	0.0077
40	41		0.0145	0.0487	0.0061

TABLE C.15

(Cont.)

40	42	0.0555	0.1830	0.0233
41	42	0.0410	0.1350	0.0172
42	49	0.0358	0.1610	0.0860
43	44	0.0608	0.2454	0.0303
44	45	0.0224	0.0901	0.0112
45	46	0.0400	0.1356	0.0166
45	49	0.0684	0.1860	0.0222
46	47	0.0380	0.1270	0.0158
46	48	0.0601	0.1890	0.0236
47	49	0.0191	0.0625	0.0080
47	69	0.0844	0.2778	0.0355
48	49	0.0179	0.0505	0.0063
49	50	0.0267	0.0752	0.0093
49	51	0.0486	0.1370	0.0171
49	56	0.0090	0.0459	0.0248
49	69	0.0985	0.3240	0.0414
49	54	0.0398	0.1450	0.0734
50	57	0.0474	0.1340	0.0166
51	52	0.0203	0.0588	0.0070
51	58	0.0255	0.0719	0.0089
52	53	0.0405	0.1635	0.0202
53	54	0.0263	0.1220	0.0155
54	55	0.0169	0.0707	0.0101
54	56	0.0027	0.0095	0.0036
54	59	0.0503	0.2293	0.0299
55	56	0.0048	0.0151	0.0019
55	59	0.0473	0.2158	0.0282
56	57	0.0343	0.0966	0.0121
56	58	0.0343	0.0966	0.0121
56	59	0.0407	0.1200	0.0552
59	60	0.0317	0.1450	0.0188
59	61	0.0328	0.1500	0.0194
60	61	0.0026	0.0135	0.0073
60	62	0.0123	0.0561	0.0073
61	62	0.0082	0.0376	0.0049
62	66	0.0482	0.2180	0.0289
62	67	0.0258	0.1170	0.0155
63	64	0.0017	0.0200	0.1080
59	63	0.0000	0.0386	0.0000
64	65	0.0027	0.0302	0.1900
61	64	0.0000	0.0268	0.0000
65	66	0.0000	0.0370	0.0000
65	68	0.0014	0.0160	0.3190
66	67	0.0224	0.1015	0.0134
68	116	0.0003	0.0040	0.0820
68	69	0.0000	0.0370	0.0000
68	81	0.0017	0.0202	0.4040
69	75	0.0405	0.1220	0.0620
69	77	0.0309	0.1010	0.0519
69	70	0.0300	0.1270	0.0610
70	71	0.0088	0.0355	0.0043
70	74	0.0401	0.1323	0.0168
70	75	0.0428	0.1410	0.0180
71	72	0.0446	0.1800	0.0222
71	73	0.0087	0.0454	0.0059
74	75	0.0123	0.0406	0.0051
75	118	0.0145	0.0481	0.0059
75	77	0.0601	0.1999	0.0249
76	118	0.0164	0.0544	0.0068
76	77	0.0444	0.0148	0.0184
77	78	0.0037	0.0124	0.0063
77	80	0.0108	0.0331	0.0350
77	82	0.0298	0.0853	0.0409
78	79	0.0054	0.0244	0.0032
79	80	0.0156	0.0704	0.0093
80	96	0.0356	0.1820	0.0247
80	97	0.0183	0.0934	0.0127
80	98	0.0238	0.1080	0.0143
80	99	0.0454	0.2060	0.0273
80	81	0.0000	0.0370	0.0000
82	96	0.0162	0.0530	0.0272
82	83	0.0112	0.0366	0.0190
83	84	0.0625	0.1320	0.0129

TABLE C.15

(Cont.)

83	85	0.0430	0.1480	0.0174
84	85	0.0302	0.0641	0.0061
85	86	0.0350	0.1230	0.0138
85	88	0.0200	0.1020	0.0138
85	89	0.0239	0.1730	0.0235
86	87	0.0282	0.2074	0.0222
88	89	0.0139	0.0712	0.0096
89	90	0.0158	0.0653	0.0794
89	92	0.0079	0.0380	0.0481
90	91	0.0254	0.0836	0.0107
91	92	0.0387	0.1272	0.0163
92	93	0.0258	0.0848	0.0109
92	94	0.0481	0.1580	0.0203
92	100	0.0648	0.2950	0.0386
92	102	0.0123	0.0559	0.0073
93	94	0.0223	0.0732	0.0094
94	95	0.0132	0.0434	0.0055
94	96	0.0269	0.0869	0.0115
94	100	0.0178	0.0580	0.0302
95	96	0.0171	0.0547	0.0074
96	97	0.0173	0.0865	0.0120
98	100	0.0397	0.1790	0.0238
99	100	0.0180	0.0813	0.0108
100	101	0.0277	0.1262	0.0164
100	103	0.0160	0.0525	0.0268
100	104	0.0451	0.2040	0.0270
100	106	0.0605	0.2290	0.0310
101	102	0.0246	0.1120	0.0147
103	110	0.0391	0.1913	0.0230
103	104	0.0466	0.1584	0.0203
103	105	0.0535	0.1625	0.0204
104	105	0.0099	0.0378	0.0049
105	106	0.0140	0.0547	0.0072
105	107	0.0530	0.1830	0.0236
105	108	0.0261	0.0703	0.0092
106	107	0.0530	0.1830	0.0236
108	109	0.0105	0.0288	0.0038
109	110	0.0278	0.0762	0.0101
110	111	0.0220	0.0755	0.0100
110	112	0.0247	0.0640	0.0310
114	115	0.0023	0.0104	0.0014

TABLE C.15.1

TRANSFORMER	TAP	SETTING
8	5	0.985
26	25	0.960
30	17	0.960
38	37	0.938
63	59	0.960
64	61	0.985
65	66	0.935
68	69	0.935
81	80	0.935

TABLE C.15.2

BUS	SHUNT ELEMENT
5	-0.400
34	0.140
37	-0.250
44	0.100
45	0.100
46	0.100
48	0.150
74	0.120
79	0.200
82	0.200
83	0.100
105	0.200
107	0.060
110	0.060

+ = Capacitor
- = Reactor

TABLE C.16

BASE-CASE LOAD FLOW SOLUTION
118-BUS TEST SYSTEM

BUS	TYPE	VOLTAGE	ANGLE	GEN MW	GEN MVAR	LOAD MW	LOAD MVAR
1	2	0.903	-17.872	0.000	0.050	51.000	27.000
2	3	0.921	-17.686	0.000	0.000	20.000	9.000
3	3	0.918	-17.250	0.000	0.000	39.000	10.000
4	2	0.967	-14.242	-9.000	58.740	30.000	12.000
5	3	0.967	-13.707	0.000	0.000	0.000	0.000
6	2	0.948	-16.329	0.000	26.680	52.000	22.000
7	3	0.945	-16.745	0.000	0.000	19.000	2.000
8	2	1.003	-8.760	-28.000	-123.490	0.000	0.000
9	3	1.042	-1.467	0.000	0.000	0.000	0.000
10	2	1.044	5.470	450.000	-50.880	0.000	0.000
11	3	0.941	-16.497	0.000	0.000	70.000	23.000
12	2	0.944	-17.106	85.000	100.650	47.000	10.000
13	3	0.918	-17.515	0.000	0.000	34.000	16.000
14	3	0.934	-17.691	0.000	0.000	14.000	0.100
15	2	0.919	-17.785	0.000	30.230	90.000	30.000
16	3	0.934	-17.222	0.000	0.000	25.000	10.000
17	3	0.948	-15.561	0.000	0.000	11.000	3.000
18	2	0.924	-17.547	0.000	45.350	60.000	34.000
19	2	0.911	-17.780	0.000	-14.280	45.000	25.000
20	3	0.911	-16.854	0.000	0.000	18.000	3.000
21	3	0.919	-15.388	0.000	0.000	14.000	8.000
22	3	0.941	-13.149	0.000	0.000	10.000	5.000
23	3	0.988	-8.694	0.000	0.000	7.000	3.000
24	2	0.980	-8.663	-13.000	-13.330	0.000	0.000
25	2	1.049	-1.976	220.000	278.130	0.000	0.000
26	2	1.015	0.011	314.000	-255.690	0.000	0.000
27	2	0.937	-13.848	-9.000	4.000	62.000	13.000
28	3	0.923	-15.364	0.000	0.000	17.000	7.000
29	3	0.920	-16.317	0.000	0.000	24.000	4.000
30	3	1.012	-11.014	0.000	0.000	0.000	0.000
31	2	0.924	-16.274	7.000	37.970	43.000	27.000
32	2	0.930	-14.279	0.000	-16.120	59.000	23.000
33	3	0.913	-13.277	0.000	0.000	23.000	9.000
34	2	0.932	-18.089	0.000	71.930	59.000	26.000
35	3	0.925	-18.394	0.000	0.000	33.000	9.000
36	2	0.926	-18.433	0.000	22.160	31.000	17.000
37	3	0.933	-17.495	0.000	0.000	0.000	0.000
38	3	1.013	-13.043	0.000	0.000	0.000	0.000
39	3	0.898	-20.437	0.000	0.000	27.000	11.000
40	2	0.894	-21.534	-46.000	38.980	20.000	23.000
41	3	0.888	-21.858	0.000	0.000	37.000	10.000
42	2	0.916	-20.804	-59.000	41.070	37.000	23.000
43	3	0.925	-18.027	0.000	0.000	18.000	7.000
44	3	0.945	-15.857	0.000	0.000	16.000	8.000
45	3	0.955	-14.158	0.000	0.000	53.000	22.000
46	2	0.984	-11.677	19.000	-4.980	28.000	10.000
47	3	1.003	-9.571	0.000	0.000	34.000	0.000
48	3	1.004	-10.429	0.000	0.000	20.000	11.000
49	2	1.012	-9.473	204.000	116.360	87.000	30.000
50	3	0.982	-11.270	0.000	0.000	17.000	4.000
51	3	0.938	-13.368	0.000	0.000	17.000	8.000
52	3	0.925	-14.114	0.000	0.000	18.000	5.000
53	3	0.910	-14.851	0.000	0.000	23.000	11.000
54	2	0.922	-14.161	48.000	4.000	113.000	32.000
55	2	0.918	-14.386	0.000	4.540	63.000	22.000
56	2	0.921	-14.240	0.000	-2.370	84.000	18.000
57	3	0.942	-13.347	0.000	0.000	12.000	3.000
58	3	0.927	-13.985	0.000	0.000	12.000	3.000
59	2	0.967	-10.745	155.000	163.490	277.000	113.000

TABLE C.16

(Cont.)

60	3	0.985	-7.105	0.000	0.000	78.000	3.000
61	2	0.989	-6.237	160.000	-36.020	0.000	0.000
62	2	0.991	-6.860	0.000	1.210	77.000	14.000
63	3	1.007	-7.611	0.000	0.000	0.000	0.000
64	3	1.008	-5.873	0.000	0.000	0.000	0.000
65	2	1.004	-2.666	391.000	-552.010	0.000	0.000
66	2	1.049	-2.943	392.000	381.770	39.000	18.000
67	3	1.015	-5.557	0.000	0.000	28.000	7.000
68	3	1.019	-2.974	0.000	0.000	0.000	0.000
69	1	1.035	0.000	524.270	250.220	0.000	0.000
70	2	0.977	-6.606	0.000	7.500	66.000	20.000
71	3	0.979	-7.073	0.000	0.000	0.000	0.000
72	2	0.969	-8.318	-12.000	-11.070	0.000	0.000
73	2	0.983	-7.311	-6.000	9.680	0.000	0.000
74	2	0.950	-6.896	0.000	-6.990	68.000	27.000
75	3	0.963	-5.629	0.000	0.000	47.000	11.000
76	2	0.941	-3.387	0.000	-63.340	68.000	36.000
77	2	1.003	-4.628	0.000	79.490	61.000	28.000
78	3	1.000	-4.899	0.000	0.000	71.000	26.000
79	3	1.006	-4.585	0.000	0.000	39.000	32.000
80	2	1.039	-2.277	477.000	323.320	130.000	26.000
81	3	1.055	-2.932	0.000	0.000	0.000	0.000
82	3	0.986	-4.043	0.000	0.000	54.000	27.000
83	3	0.983	-2.904	0.000	0.000	20.000	10.000
84	3	0.980	-0.497	0.000	0.000	11.000	7.000
85	2	0.985	1.010	0.000	-5.230	24.000	15.000
86	3	0.986	-0.643	0.000	0.000	21.000	10.000
87	2	1.015	-0.893	0.000	11.850	0.000	0.000
88	3	0.985	4.163	0.000	0.000	48.000	10.000
89	2	0.994	8.332	607.000	-3.880	0.000	0.000
90	2	0.984	1.905	-85.000	57.760	78.000	42.000
91	2	0.979	1.928	-10.000	-13.080	0.000	0.000
92	2	0.989	2.504	0.000	-16.340	65.000	10.000
93	3	0.985	-0.472	0.000	0.000	12.000	7.000
94	3	0.989	-2.597	0.000	0.000	30.000	16.000
95	3	0.978	-3.521	0.000	0.000	42.000	31.000
96	3	0.990	-3.724	0.000	0.000	38.000	15.000
97	3	1.010	-3.358	0.000	0.000	15.000	9.000
98	3	1.021	-3.870	0.000	0.000	34.000	8.000
99	2	1.007	-4.226	-42.000	-17.530	0.000	0.000
100	2	1.015	-3.279	252.000	93.730	37.000	18.000
101	3	0.991	-1.651	0.000	0.000	22.000	15.000
102	3	0.989	1.016	0.000	0.000	5.000	3.000
103	2	1.003	-7.019	40.000	75.410	23.000	16.000
104	2	0.958	-9.217	0.000	2.450	38.000	25.000
105	2	0.948	-10.209	0.000	-18.390	31.000	26.000
106	3	0.944	-10.414	0.000	0.000	43.000	16.000
107	2	0.925	-12.919	-22.000	6.550	28.000	12.000
108	3	0.946	-11.365	0.000	0.000	2.000	0.100
109	3	0.945	-11.802	0.000	0.000	8.000	3.000
110	2	0.948	-12.673	0.000	0.290	39.000	30.000
111	2	0.960	-11.189	36.000	-1.840	0.000	0.000
112	2	0.936	-15.622	-43.000	41.510	25.000	13.000
113	2	0.954	-15.803	-6.000	35.340	0.000	0.000
114	3	0.926	-14.547	0.000	0.000	8.000	3.000
115	3	0.926	-14.554	0.000	0.000	22.000	7.000
116	2	1.003	-3.281	-184.000	-372.540	0.000	0.000
117	3	0.920	-18.251	0.000	0.000	20.000	8.000
118	3	0.946	-4.985	0.000	0.000	33.000	15.000

SYSTEM GENERATION: 3807.265 MW 823.011 MVAR
SYSTEM LOAD: 3667.979 MW 1437.983 MVAR

APPENDIX D

DATA FOR THE DYNAMIC SIMULATION OF TEST SYSTEMS

D.0 - INTRODUCTION

The generating plant data used for the dynamic simulation of test systems are summarised in this appendix.

Data are given for the different parameters of the detailed representation of generating plants shown in Fig. D.1. The boiler was not represented in the simulation studies carried out.

The data for the generation units of the WSCC 9-bus test system were taken from Anderson and Fouad [27] and for the New England 39-bus system from Athay et al.[100].

Identical data were used for the power system stabiliser and governor-turbine models in all the generating plants of the test systems.

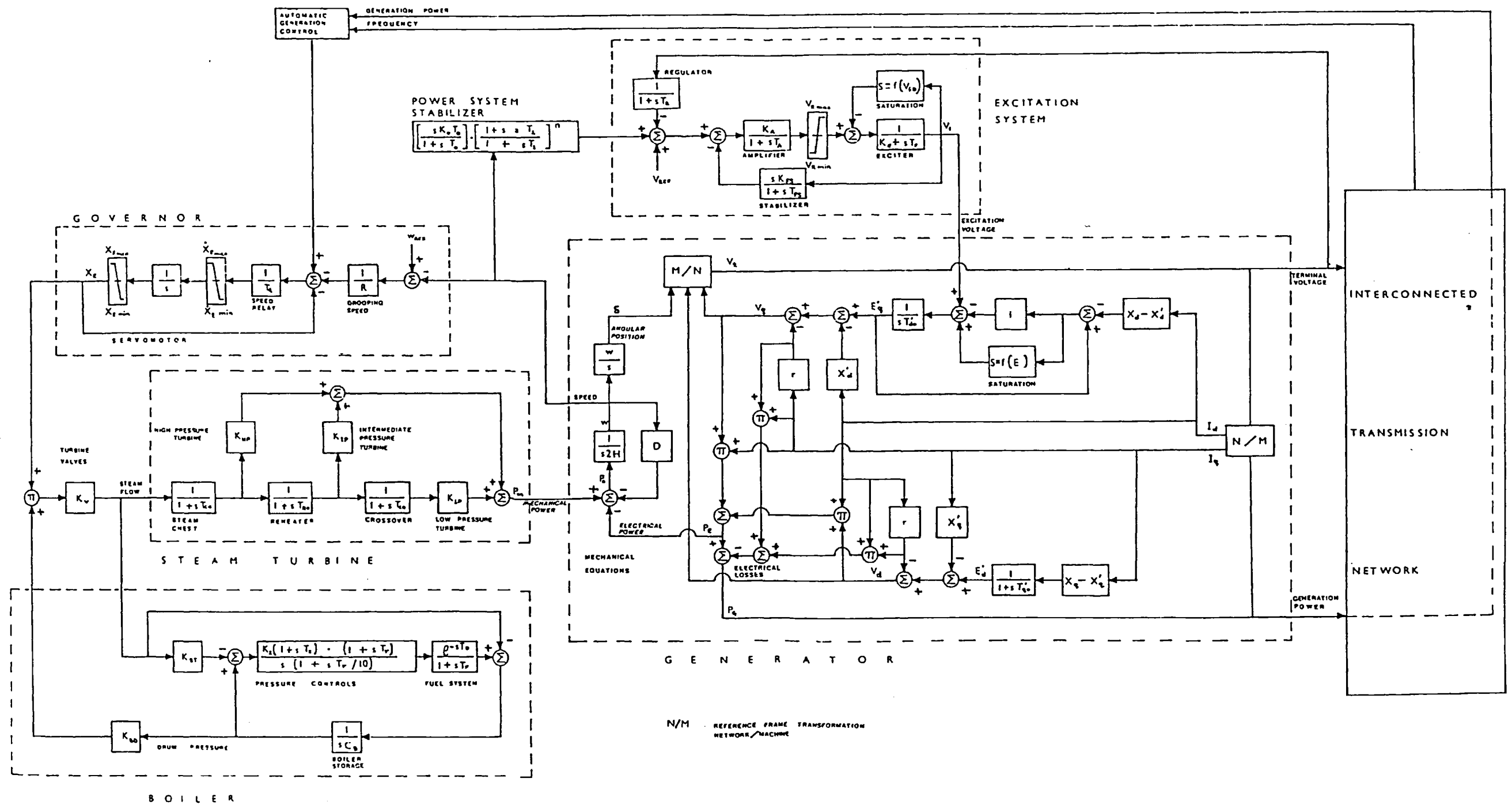


Fig. D.1: Generating Plant Model

D.1 - WSCC 9-BUS TEST SYSTEM

TABLE D.1

Generator Data
9-Bus Test System

UNIT	Ra	Xd	Xq	X'd	X'q	H	D	T'do	T'qo
1	0.0000	0.1416	0.0969	0.0608	0.0608	23.640	0.0000	8.9600	0.4000
2	0.0000	0.8958	0.8645	0.1198	0.1198	6.400	0.0000	6.0000	0.5350
3	0.0000	1.3125	1.2578	0.1813	0.1813	3.010	0.0000	5.8900	0.6000

TABLE D.2

Excitation System Data
9-Bus Test System

UNIT	Ka	Ta	Ke	Te	Kf	Tf	Vrmax	Vrmin
1	25.0000	0.2000	-0.0601	0.6758	0.1080	0.3500	1.0000	-1.0000
2	25.0000	0.2000	-0.0601	0.6758	0.1080	0.3500	1.0000	-1.0000
3	25.0000	0.2000	-0.0601	0.6758	0.1080	0.3500	1.0000	-1.0000

TABLE D.3

Power System Stabiliser Data
9-Bus Test System

UNIT	Ko	To	As	Ts	Vsmax	Vsmin
1	1.0000	10.0000	25.0000	0.0227	0.0500	-0.0500
2	1.0000	10.0000	25.0000	0.0227	0.0500	-0.0500
3	1.0000	10.0000	25.0000	0.0227	0.0500	-0.0500

TABLE D.4

Speed Governing System Data
9-Bus Test System

UNIT	Tg	R	Xup	Xdown	Xvmax	Xvmin
1	0.1000	0.0500	0.1000	-0.1000	1.0000	0.0000
2	0.1000	0.0500	0.1000	-0.1000	1.0000	0.0000
3	0.1000	0.0500	0.1000	-0.1000	1.0000	0.0000

TABLE D.5

Turbine Data
9-Bus Test System

UNIT	Kv	Tch	Trh	Tco	Khp	Kip	Klp
1	1.0200	0.2500	7.0000	0.4000	0.3000	0.4000	0.3000
2	1.0200	0.2500	7.0000	0.4000	0.3000	0.4000	0.3000
3	1.0200	0.2500	7.0000	0.4000	0.3000	0.4000	0.3000

D.2 - NEW ENGLAND 39-BUS TEST SYSTEM

TABLE D.6

GENERATOR DATA
39-BUS TEST SYSTEM

UNIT	Ra	Xd	Xq	X'd	X'q	H	D	T'do	T'qo
1	0.0000	0.0200	0.0190	0.0060	0.0060	500.000	0.0000	7.0000	0.7000
2	0.0000	0.2950	0.2820	0.0647	0.0647	30.300	0.0000	6.5600	1.5000
3	0.0000	0.2495	0.2370	0.0531	0.0531	35.800	0.0000	5.7000	1.5000
4	0.0000	0.2620	0.2580	0.0436	0.0436	28.600	0.0000	5.6900	1.5000
5	0.0000	0.6700	0.6200	0.1320	0.1320	26.000	0.0000	5.4000	0.4400
6	0.0000	0.2540	0.2410	0.0500	0.0500	34.800	0.0000	7.3000	0.4000
7	0.0000	0.2950	0.2920	0.0490	0.0490	26.400	0.0000	5.6600	1.5000
8	0.0000	0.2900	0.2800	0.0570	0.0570	24.300	0.0000	6.7000	0.4100
9	0.0000	0.2106	0.2050	0.0570	0.0570	34.500	0.0000	4.7900	1.9600
10	0.0000	0.1000	0.0690	0.0310	0.0310	42.000	0.0000	10.2000	0.5000

TABLE D.7

EXCITATION SYSTEM DATA
39-BUS TEST SYSTEM

UNIT	Ka	Ta	Ke	Te	Kf	Tf	Vrmax	Vrmin
1	5.0000	0.0600	-0.0485	0.2500	0.0400	1.0000	1.000	-1.000
2	6.2000	0.0500	-0.0633	0.4050	0.0570	0.5000	1.000	-1.000
3	5.0000	0.0600	-0.0198	0.5000	0.0800	1.0000	1.000	-1.000
4	5.0000	0.0600	-0.0525	0.5000	0.0800	1.0000	1.000	-1.000
5	40.0000	0.0200	1.0000	0.7850	0.0300	1.0000	10.000	-10.000
6	5.0000	0.0200	-0.0419	0.4710	0.0754	1.2460	1.000	-1.000
7	40.0000	0.0200	1.0000	0.7300	0.0300	1.0000	6.500	-6.500
8	5.0000	0.0200	-0.0470	0.5280	0.0854	1.2600	1.000	-1.000
9	40.0000	0.0200	1.0000	1.4000	0.0300	1.0000	10.500	-10.500
10	5.0000	0.0600	-0.0485	0.2500	0.0400	1.0000	1.000	-1.000