DESIGN AND DEVELOPMENT

OF

AN EFFICIENT SIMULATION SOFTWARE

FOR

POWER SUPPLY DISTRIBUTION SYSTEM IN THE

MTR CORPORATION

CHAN KWOK LEUNG

M. PHIL.

THE HONG KONG POLYTECHNIC UNIVERSITY

2000

Abstract of thesis entitled

gn and Development of an Efficient Simulation Software for Power Supply Distribution System in the MTR Corporation

submitted by

Chan Kwok Leung

for the degree of

Master of Philosophy in Engineering

at The Hong Kong Polytechnic University

in

August 2000

ABSTRACT

This study aims to design an efficient simulation software for the study of power flow in the AC power distribution system of MTR Corporation. The software is expected for the use of design engineers in the High Voltage Electrical System team for their power system long/short term power system design and analysis. The development of the software is based on the object-oriented technology. The use of the technology enhances the development of the input system by a Windows-based graphical user interface (GUI) tailor-made for the Corporation. By making use of the technology, a Distribution Circuit Object Model (DCOM) is designed which allows users to build their own designed power networks on the GUI. A Generalised Network Access Method (GNAM) is developed to update and control the object based power networks for various system calculations and graphical displays. The developed GNAM also simply the programming structure and at the same time provide a high degree of modularity suitable for future software enhancement. Because of the single-source characterises of the power network in the Corporation, the Nodal Admittance method is proposed for power flow studies, and a AC Norton-equivalent model is developed to support studies of the system with voltage dependent loads. The model is an extension study on the DC Norton-equivalent model applicable on power flow calculations for DC traction system.

ACKNOWLEDGEMENTS

The work presented in this thesis was carried out under the supervisions of Dr. C.T. Tse and Dr. S.L. Ho of Hong Kong Polytechnic University and Mr. Charlie Chow of MTR Corporation. It is with great pleasure and deep gratitude that I wish to thank Dr. C.T. Tse and Dr. S.L. Ho for providing me with invaluable guidance and criticism throughout the contract period. I wish to thank Mr. Charlie Chow for his guidance throughout the project. I also wish to express my gratitude to Mr. S.C. Chow of MTR Corporation for his continuous encouragement and support on my writing of this dissertation. I am indebted to the support of the Hong Kong MTR Corporation and the Hong Kong Polytechnic University for their funding in setting up this Teaching Company Scheme to carry out an industrial applicable research programme. I would also like to acknowledge the valuable advice from Dr. T.K. Ho and the generous support from the Rolling Stock Section and the Electrical System team of Electrical & Mechanical Services Design Section of the MTR Corporation for giving their internal technical information. I am also grateful for the MTR Corporation to give permission to publish all information related to the software. Last, but not least, every effort substantiated from the Department of Electrical Engineering at the Hong Kong Polytechnic University is also appreciated.

BIOGRAPHICAL SKETCH

Mr. Chan received his sandwich training in Wong & Oyang Building Services Ltd. In 1994-95 and received his first Bachelors Degree in Engineering with 2nd Upper Class Honours in 1996. He then decided to stay in the University as a Teaching Company Associate and work on the Teaching Company Scheme as held between the MTR Corporation and the University.

CONTENTS

Chapter 1 Introduction

1.1 Background

A DC Multi-Train Simulator (MTS) has already been developed in 1991 in University of Birmingham to study the power demand in the traction system of MTR Corporation [1]. It is then desirable to develop a software interfacing with the MTS for the analysis of the AC distribution system. Thus, in 1996, a Teaching Company Scheme was jointly set up by the MTR Corporation and the Hong Kong Polytechnic University for the development of the software. The present thesis will report the findings developed under the Scheme.

1.2 The MTR AC Power Distribution System

The Hong Kong Mass Transit Railway (MTR) is the largest railway company in Hong Kong and has the dominating transportation role in the urban area. Therefore, a reliable power supply system is important for the Corporation. The MTR Corporation (MTRC) has its own AC power distribution system (starts at 33 kV level) supplied from two local power supply companies. Each electric company supplies several infeed points, each of which is electrically separated from each other [2-3]. Station services are operated by AC whereas trains services are supplied through DC power network. The DC power is obtained by transforming the 33kV AC to DC 1500V DC at different traction substation.

1.3 The Areas of Research in Concern

Power flow simulation by digital computer is not a new topic in the literature [4-9]. With the rapid development in the computational speed available in digital computer, research on computational speed for different power flow simulation algorithm becomes less critical. Instead, the interactive capabilities of power flow simulation software are receiving more interest [10-15]. To provide interactive features, the size of the software to be developed will increase considerably. Effective Software Project Management is thus a critical factor in the success of delivery a usable software product [16-20]. One of the main concerns of interactive software is the design on a Graphical User Interface (GUI). The GUI provides a platform for user to operate the software and to develop any new power system visually for power system simulations. To develop a GUI, the use of an object-oriented technology is nevertheless the required tools to achieve this. By using the technology, power network elements are designed as objects. Various object models have been proposed for power system simulation [21-32] such that a power network is modelled with objects with connections that are modelled by object pointers. However, there is a lack of discussions in the literature on developing visual object model that is suitable in developing a Windows-based GUI for power system simulation and how to track the modifications of object-based network by using the technology. There exist very few discussions on topic of controlling modifications in power network configurations by tracking network connectivity with network topology [33].

The degree of user-friendliness, in a great extent, shall contribute to the efficiency in using the software. As the system design engineers in MTRC are the end-user of the software, a Windows-based application software is most suitable. The software development under Window environment will be discussed in more detail in the Chapter 2 and 3.

The main concern of the system design engineer of MTRC is the power flow of the system in different operating conditions (normal and contingency). Although various digital solutions for AC power flow calculation have well been developed in the literature [34-40], there is a need to determine the most efficient algorithm especially for the AC power distribution system in MTRC. In view of the unique single-source characteristic of the AC power distribution system in MTRC, iterative nodal admittance matrix method is proposed [41-42]. This method can reduce the computational efforts and computer storage as compared to the methods that are used in multi-source AC power system [43].

1.4 Design in Modularity

The overall design concept of this software can be divided basically into five stages (Fig. 1.1). The basic framework (A) includes project management, software organization, Graphical User Interface (GUI), the programming environment and its language. Process (A) represents the starting point of the design, the long-term maintenance structure and the communication effort needed and it also contributes one type of flexibility. As the software deals with the MTR power distribution system comprising of different networks, the latest object-oriented programming technology is employed to build dynamic networks (network that is frequently changed due to data editing or network restructuring). Process (B) is to support graphical user interface and all levels of network modifications in order to provide network simulation flexibility for user. An object-oriented method should be developed to access and convert the static network (the network that has been modified) to specific input data set (C). The input data set, which is representing the network, will then be used by suitable calculation methods for actual engineering simulation (D). The format of the input data set depends completely on which calculation method is being adopted. Finally, the results (output data set) are translated into display or file format (E).

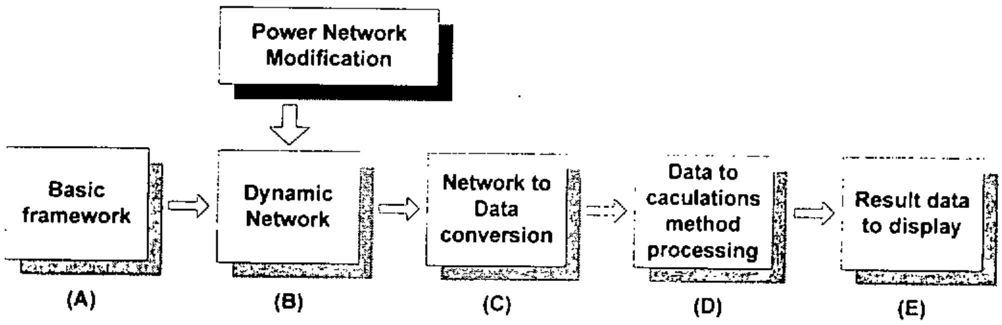


Fig. 1.1 General Design Concept

1.5 Network Model and Network Access Method

By using object-oriented technology, a modified Distribution Circuit Object Model (DCOM) based on [21] is proposed to model the elements in the MTR AC power distribution systems. A power network can then be established based upon this object model. To handle the changes in network configuration of the power system, a Generalised Network Access Method (GNAM) has been developed which can accurately track the currently activated network. This approach can handle the process of network formulations, network modifications. By comparing with the approach of network topology [33], this method can much reduce the computational efforts and the implementation complexity. Moreover, this method is independent of the size of the network in concern and its programming structure is reusable for other applications. The DCOM and the GNAM thus form the kernel part of this software.

1.6 Power flow Simulation and Load Modelling

For a large and complex system such as the MTRC system, load modelling is vital for reliable results. In the DC traction load, a constant power characteristic is usually assumed. The conventional methods, however, adopts constant conductance model of constant current injection model for convenience [52]. Therefore, numerical techniques are required since these load-modelling techniques cannot reflect the 'actual' load characteristic. Thus, the modelling parameters have to be updated in every iteration. By exploring the tracking characteristics of different DC loadflow computational techniques, the present work introduces a 'modified Norton equivalent method', which tracks the load characteristic closely and eliminates the parameter updating. Simulations show that the method can increase the convergence characteristic and drastically reduces the computation time. The technique is then applied to the MTR AC power distribution system connected with active/reactive loads. The voltage dependent load characteristics can be tackled again by another Norton equivalent: a current vector plus a fictitious load represented by a 2x2 matrix. The distinguished features with respect to the well-known loadflow methods will be highlighted.

1.7 The features of the developed software

In this software a name-based one-line diagram is developed. Cables are represented graphically by three line segments that are of the same graphical format as the one-line diagram that the engineers in MTRC are using [44]. Graphical inputs and editing of network elements are done by input dialog box in Windows format. A visual relay database is developed for on-screen relay setting. The GNAM is developed as a single programming structure for controlling object based power network such as retrieving the technical data of an power element of a chosen power network, saving a power network and display the result data onto the one-line diagram graphically. Colour representation of different power networks is developed. Changeover switch is available to allow network changeover. Power flow is indicated by colour. The colour that represents the power flow will automatically be adjusted whenever switching is activated. Large scale of power distribution network (up to 100 bus per network) is allowed. Power flow results are displayed on the same one-line diagram. (Appendix A).

1.8 Brief Outline of the Thesis

In Chapter 2, the Interface Design and the interactive design of the GUI are discussed and user-friendliness will be defined. The choice of a programming language will be examined. The design on the visualization of the GUI will be explained in details. The use of implementation modules and the operation pie are introduced. Then, the proposed GNAM and its applications are discussed in Chapter 3 where the designed distribution software architecture will be introduced. Besides, a network model is defined in the software domain. The use of object and object pointers to simulate various power system elements and the direct connections between elements will be fully explained. The implementation details of the proposed GNAM will be presented. Applications of the GNAM will then be classified. The DC Norton equivalent model will be introduced in Chapter 4. Different studies on the MTR DC traction system will be presented to justify the effectiveness of the Norton-equivalent model. Application of Norton equivalent of AC load modelling will be revealed in Chapter 5. Its effectiveness in the application on the digital power flow calculation for MTRC AC system will be justified by different cases studies on the system in comparison with conventional algorithm. Fault analysis together with an effective method to account for transformer phase shift will be presented in Chapter 6, in which the design for simulation of the operations of relays and the 33kV changeover system in MTRC will be discussed. Chapter 7 will then summarizes the present research and development work. The contributions to the MTRC will be concluded as well.

1.9 Publications

Three conference papers associated to the present work have been published. The first paper describes the features of the developing software. The second paper proposes a new effective DC loadflow algorithm using Norton-Equivalent for load model. The third paper extends the Norton Equivalent to represent the voltage-dependent load of the MTRC AC power distribution system.

1. K.L. Chan, C.T. Tse, S.L. Ho, S.C. Chow, W.Y. Lo, K.K. Chow, 1998, "An Interactive Graphical Simulation Package for the AC Supply System of Hong Kong Mass Transit Railway", International Conference on Mass Transit Railways, London U.K., Conference Publication No. 453, pp18-24.

2. C.T. Tse, K.L. Chan, S.L. Ho, S.C. Chow, W.Y. Lo, 1998, "Tracking Techniques for DC Traction Loadflow", International Conference on Mass Transit Railways, London, Conference Publication No. 453, pp286-290

3. C.T. Tse, K.L. Chan, S.L. Ho, C.Y. Chung, S.C. Chow, W.Y. Lo, 1997, "Effective Loadflow Technique with Non-Constant MVA Load for the Hong Kong Mass Transit Railway Urban Lines Power Distribution System", APSCOM-97, H.K., proceedings Vol. 2, pp753-757.

Chapter 2 Interface and Interactive Design

2.1 Introduction

It is intended to combine graphics and calculation into one single programme so that true representation and simulation can be achieved. Objects[1] are used to simulate what exist in the power distribution system. The objects include drawing methods to produce visual effects on the user interface. The connections between objects can be simulated. It works as if it is actually a network, not a graphic because what the user sees is just the visual part (the graphical image) of the objects. The objects are actually connected to form the network. This is an important application in power system simulation provided by the object-oriented methodology. The following two chapters will discuss the various design concepts with the visual and non-visual object-oriented programming techniques.

2.2 Defining User-friendly

The word user-friendly is well known as the key point in selling new software. Other than providing graphics, user-friendly can be classified as the following groups:

- **Flexibility** is the availability for making changes and the time-effort needed.
- **Interaction** concerns bi-directional communication. The computer communicates

---

[1] The word "objects' refers to power network elements that modelled by object-oriented programming.

with user in all kind of innovative channels.

- **Reality** is the capability to provide reality in operating the power system on screen.
- **Attractive** focus on graphical and colorful backgrounds and graphical design.
- **Reliability** is the ability to prevent user's typing-mistake or mal-operation.

The user-friendliness of software depends on the degree of the above capabilities that the software can be provided. Sometimes, they are integrated into one single design.

High flexibility software should provide quick methods for user to make changes on those items that require frequent changes because of simulations and corrections. High automation is important. The number of user operating steps (e.g. the number of mouse click required for an operation) should be considered. High automation means fewer users operating steps. Methods to make changes on parameters of the working environment and/or structure should be provided. High interaction provides effective communication, in any possible form, to direct user in operations and prevent wrong operations. Supplies of reality give the 'true' world feeling. It makes use of deep knowledge on the changing appearance with the movements of the object to simulate, as closely as possible, the sense of materialistic and reality for the objects in the application domain. Attractive relies on graphical capability. They include colour, fonts, line, shape and 3D graphics.

2.3 Choice of Programming Language

Most of the power system simulation software developed in University is based on C++

programming language [12-15] that provides a relatively fast computer compiling time. However, because of the high competition in computer industry, the development of computer technology increases dramatically in the last few years. Because of the code release of Windows language, the Application Programming Interface (API) and many new visual programming tools have been developed. Among the programming tools, Delphi [45] was adopted to develop the software.

Generally, Delphi provides a new way to develop applications for Windows in a user-friendly environment. It combines the speed and ease-to-use of a visual development environment with the power, flexibility, and reusability of a fully object-oriented language using the fastest compiling technology. Delphi supports creating stand-alone executable file for developed application. Many of the traditional requirements of programming for Windows are handled for the user within its Class Library.

From the viewpoint of solving the requirements of this project, Delphi highly supports windows application development and the use of object-oriented language and includes a Visual Component Library (VCL) to support development of own Windows based software. The organisations with Delphi project provide systematic modular system and support modular system organisation in object notation using Pascal language. The use of object-oriented programming already supports code maintenance in design stage and implementation stage. New objects and associated implementation details can be designed and added onto the DCOM when required. Changes with the windows interface can be readily implemented with its working environment. Implementation requires programming technique, general understanding of object-oriented language and the windows language provided by Delphi.

In conclusion, with consideration of programming time, Delphi supports a close start in developing stand-alone software. It also provides high maintainability. For example, it support Fortran translation

All of the above supports meet the requirements in this project. Thus, Delphi is chosen for developing the object-oriented user-friendly software.

2.4 Design on Basic Framework

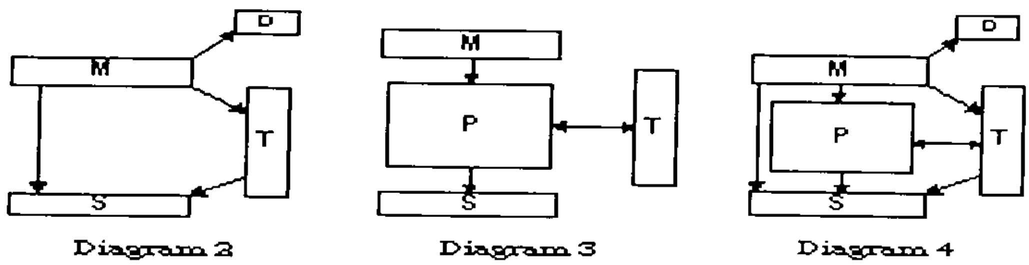
2.4.1 General Interfacing Design

For the users, the software is just a window. The first task in design is to determine how users can operate through the window and how to make co-ordination between various components. A common design contains a toolbar, a main menu, a pop-up menu and a status bar on the window [46].

2.4.2 Inside Window

The tool bars provide first means to operate the software. The tools in the toolbar upply function of direct operation or selection. Main menu provides another choice in ie software operation and at the same time provides keyboard operation. The main ienu has three functions: (1) provides direct operations, (2) provides channels (like Options") for modification of software system parameters and (3) provides selections. spite of this, 'Help' menu can be added to increase the user-friendliness. For direct operations, the items on the menu invoke a procedure itself or it can simply point to the tools in the tool bar that perform direct operation. The pop-up menu is an independent menu that appears (at the mouse point of the screen) when needed (user invoked). The items on the pop-up menu should point to the most popular direct operations to act as a quick operation method. Therefore, usually the items point to the tools in the tools bar. From the user point of view, it is the quickest way, relative to the main menu, to invoke command since the pop-up menu always stays behind the mouse point where the user "exists". However, this type of menu cannot contain all commands that the software can do because of the dimension. Therefore, a pop-up with variable items may be desirable. This involves more complexes programming co-ordination.

The status bar displays the current operational status and drawing status. Both the items in the main menu and the tools in the toolbar use the status bar to communicate with users what will work with the operations. The conceptual interaction among the above visual components of the Windows interface is shown in Fig. 2.1, Diagram 2.



g. 2.1 interaction of interface components: Main menu (M) Edit dialog (D), Tool bar (T), Status bar (S), rawing panel (P)

2.4.3 Power System Formulation and Display

In the toolbar, specific tools allow specific network objects to be created. The objects are created in computer memory. When an object is created, the visual attributes are converted into visual effect by pre-defined drawing method in the object. The drawing panel (Fig. 2.1, diagram 3) provides a visible area for viewing the visual effect. The tools operate the object in the panel and the objects update the status of the tools. Meanwhile, an interface must be defined on the visual effect of each created object to provide channel for user choosing the object and performing operation with the object. In principle, users use mouse to operate the network objects and computer can operate objects by using object pointers.

A system is formed when network objects are created and then connected. The structures of the network should be defined inside the objects and therefore connections among objects must be simulated (refer to section 3.5). The tools in the toolbar can operate the power distribution system directly.

The conceptual interaction among the above visual components of the window is shown in Fig. 2.1 where diagram 4 is the combination of diagram 2 and diagram 3.

2.4.4 Operations with Tools

Most of the tools in the tool bar are designed to perform implementations directly. That is, they perform Command Operations (immediate actions). In design stage, actual implementations are organised in "implementation modules". Thus, the Command Operations are described as directly pointing to implementation module (Fig. 2.2).

Some operations involve more than one type of unknown. In this case, selection(s) is/are required before an actual operation. A selection process is derived into several steps. Each selection step provides one type of information. The full selections complete when all selection steps are correct. The number of step required depends on the complexity. The command operation can be either on another tool or on the object interface that finally be chosen.

For example, some of the tools represent the network elements. When one of them is selected, the user is allowed to create the object represented by the selected tool (Fig. 2.3).

Button: "Auto loadflow"	Actual operation
Implementation steps to perform loadflow in all power networks	Implementation module

Fig. 2.2 Example implementation module – Auto loadflow

Button: "Network 1"	1st Selection
Button: "Create source"	2nd Selection
Operation on drawing panel	Actual operation
Implementation steps in creating a generator	Implementation module

Fig. 2.3 Example implementation module – Create a source.

The conceptual routes in implementation are described in Fig. 2.4 and the designed tool bars are shown in Fig. 2.5.

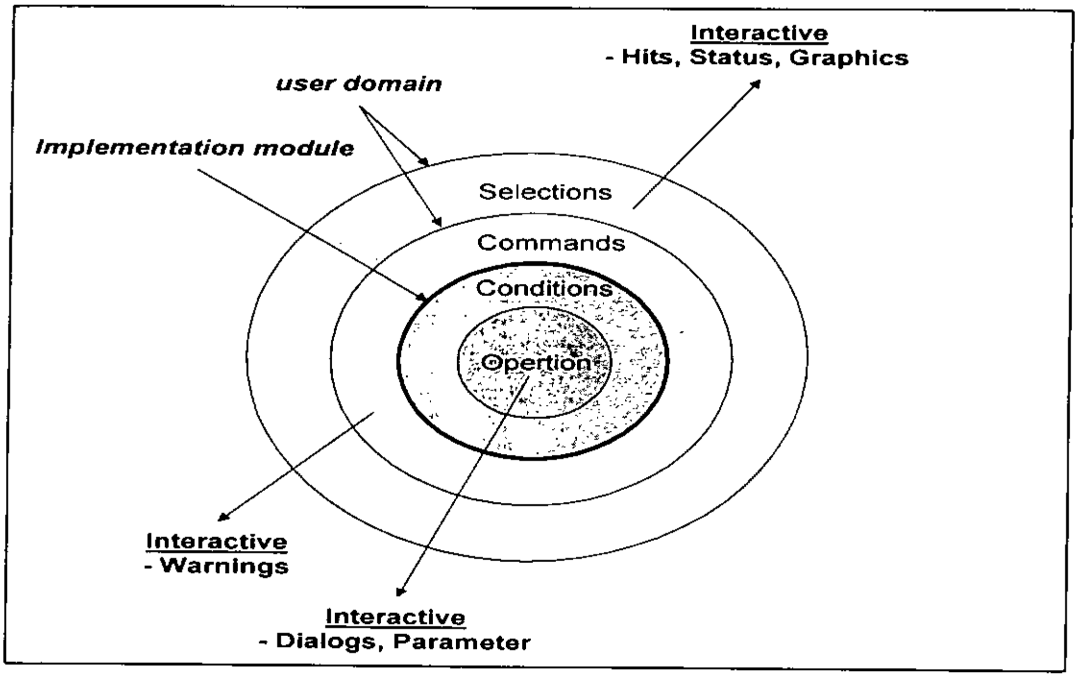


Fig. 2.4 The Operation Pie

2.5 Combination of Interactive Components

This description of interactive part of the software sometimes unavoidably uses the language of programming since the features are more or less well defined in windows programming. Therefore, the first step is to understand the basic construction and the usage in each interactive component that are supplied from Delphi. They have already provided a high flexibility. In fact the provision is more than enough. Careful selections and accuracy are required in the design for own application. There are some complex cases that require new developments. But the starting point is still based on the Visual Component Library (VCL) provided by Delphi.

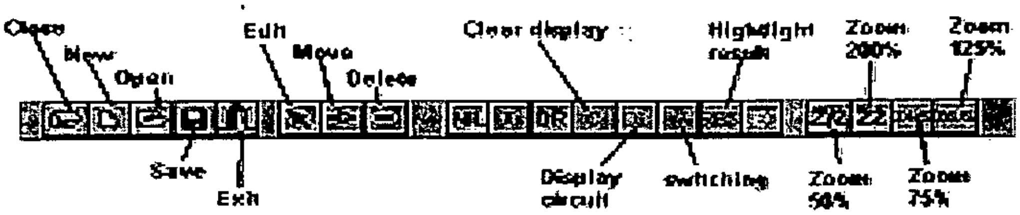


Fig. 2.5 A Operational tool bar.

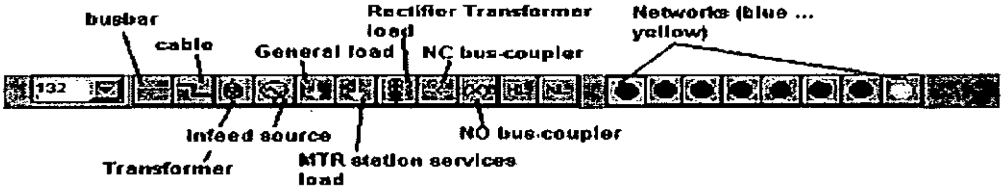


Fig. 2.5 B Network bar.

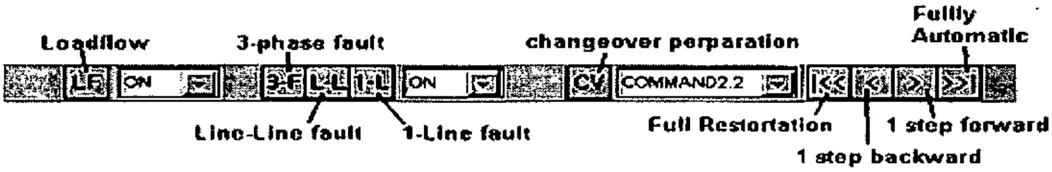


Fig. 2.5 C Simulation bar 1.

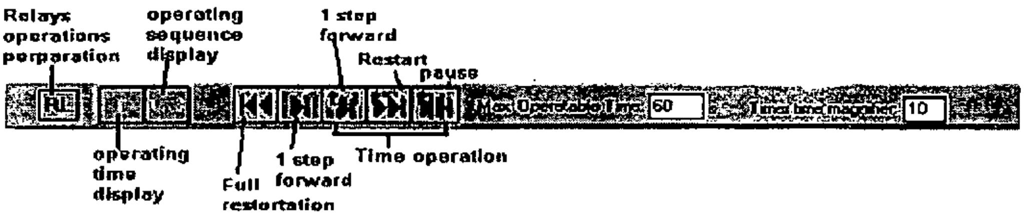


Fig. 2.5 D Simulation bar 2.

2.6 Handling the Power System Graphical Interface

In the interactive domain, "interactive" can be classified into three groups namely Handling dialogs, Handling graphics and Handling mouse event. A common way for software to communicate with user is by using dialog boxes. Some dialogs give a message only. Some require user prompt with choices that change the programme flow.

This should be considered in design stage.

Some interactive techniques are based on the graphics of the designed objects. Different graphical effects represent different status of the object. The effects are handled inside the objects in the design stage.

The only direct method to select the graphical components on screen is by using a mouse. Basically, a stand-alone graphics will not response on a mouse click action if the graphics is just a drawing. To response to the mouse click action from user, the graphics must be something more than ordinary graphics. If the graphics are not objects, the situation become complex to be handled. When visual object-oriented language is used, the graphics becomes the graphical image of the object. A region that belongs to the object can be defined in order to accept the mouse click action.

2.7 Component Modifications

For modification of more than one piece of information, a simple input box will not apply [46]. A common way is to design a new dialog window that suits a particular application to handle the complex entries and data controls. One example is editing objects: Edit the technical data of cable, bus, transformer, etc. In this case, visual components with data-editing capabilities are used instead.

2.8 Visual Indications

By using the technique introduced in section 2.6, e.g., to switch off a cable, users select the mode of 'switching' and click on the cable to be switched off. The appearance of the cable can be changed to dotted line, indicating an off status.

2.9 Conclusion

Delphi provides a quick solution to develop new window-based software. User-friendly and interactive features can also be developed. With the aid of this visual programming tool and the visual object-oriented programming technology, the power system can be displayed and simulated on screen. Simple graphical operations and high degree of user-friendliness can be developed in the design stage to reach one of the objectives of this project.

Chapter 3 Object Model and Network Access Method

3.1 Introduction

Among the literature discussing the adoption of object-oriented programming in power system simulations, Dr. Jun Zhu [30] suggested a general Distribution Circuit Object Model (DCOM) as the kernel of the software system for power system simulation. His suggested model aims to support wide range of simulation solutions. The original DCOM is being modified in this project for the MTR system.

As proposed by David Elegy, a conceptual model should be first developed [47]. Software that aims to simulate power distribution system can be described by the distributed system architecture [30] (Fig. 3.1). The architecture contains all basic component modules such as the GUI, Application Programming Interface (API), Distribution Circuit Object Model (DCOM), Database and Applications. The APIs access DCOM to create dynamic power distribution system, access the Database and achieve applications capabilities. Actually, there are a lot of APIs. The dynamic power distribution system should be placed in between API and DCOM because it uses the data structure of DCOM created by an API. The dynamic power distribution system uses memory when created and, with visual attribute, the user can see it on screen.

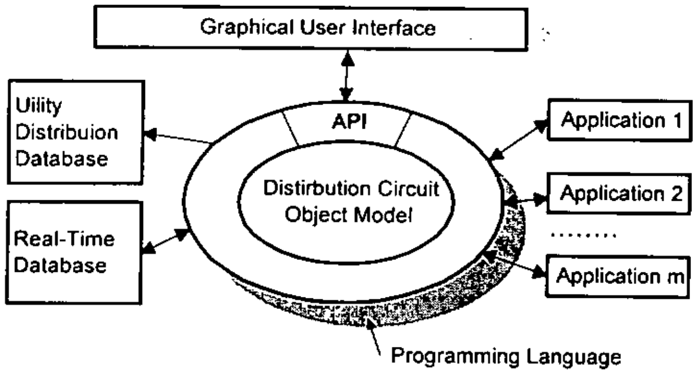


Fig. 3.1 Distributed Software Architecture

The DCOM provides method to organise the objects and establish interaction between objects. Moreover, it is able to provide channels so that user can operate on the objects, visual attributes and visual effect. Interfaces for those objects are already discussed in the previous chapter. An object includes properties, methods and events. A drawing method and a user interface of that object should be provided.

An object is built from an API by accessing the DCOM. A number of objects can be created to represent different network elements. However, engineering simulation always works with the entire network rather than working with individual elements. It is much convenient to derive method to access the network as a whole. A whole-network concept is therefore proposed and this must be achieved based on the proposed power network object model.

The most time-consuming part in using simulation software developed by conventional function-oriented approach is mainly due to the restructuring of the network, which in turn requires restructure of the data for calculation. Switching is one of the main causes of the restructuring. For example, switching off a cable may disconnect the supply in a radial circuit (but not in a ring circuit) and a reduction on the matrix size occurs. Switching a normally closed circuit breaker between two bus-sections separates the data and destroys the original dimension of the data structure. A generalised network access method is to be developed to handle problems arisen from network restructures.

3.2 Generalised Network Linkage Model

The MTR urban-line power distribution system consists of 4 electrical networks. In this thesis, a network is defined as an electrically continuous power network. The 4 networks are normally disconnected. Each network consists of power system elements known as objects. To establish system components, objects are created and connections established during the process of creating the objects. Thus, a network is a collection of objects and connections. This "network = $\sum$ (objects, connections)" is called the "Generalised Network-Linkage Model (GNLM)". Based on this model, a method called "Generalised Network Access Method (GNAM)" is derived to access the whole network. This method can solve many problems ranging from drawing to power system simulation. One example is to produce input data set to represent the current status of the network for engineering calculations (Fig. 1.1, step (C)). The following sections that discussing the DCOM and the GNAM are highly interdependent.

3.3 Object Definition

In object-oriented programming, an object is a data type or a data structure. Thus an object takes up memory only when it has been created. The dimension of this data type is flexible to be extended so that different classes of objects can be implemented while it preserves the original object structure.

Since an object is a data type only, a created object takes up memory and work as what is described in the data type. It is not necessary to have visual appearance. The visual appearance is the visual or drawing effects that is done by explicitly defining the drawing method and drawn, with different visual attributes, at run-time. Thus, work with object-oriented programming can deal only with the functional level. Visual operation level is dealt with only when a GUI is required.

3.4 Network Element as an Object

To define various network elements as objects, a deep knowledge of the various power system components is needed. An object should be the elementary component of the real power system in the application domain. Here the application domain is the power distribution system. The objects could be busbar, source, cable, transformer, circuit breakers and relays. This is called 'Finding object' by using real-world concept [30].

After finding the objects, the objects should be modelled and defined. An overall structure is designed with the class diagram (section 3.5). Then different objects are defined with real-world knowledge so that they have properties and methods to simulate different operations and various operating statuses.

3.5 Designed Object Hierarchical Relationship – The Class Diagram

The DCOM suggested by Dr. Zhu is an overall architecture of the designed objects. It shows the objects Hierarchy, Association and Aggregation [30]. Some changes have to be made to simulate the power distribution system in MTR. The architecture is rearranged as shown (Fig. 3.2).

Most of the design effort is placed on Hierarchy and Association. Hierarchy related to the types, grouping and the parent-child relationship. The general structure of Hierarchy is already shown on Fig. 3.2. Associations simulate the connections among elements. The Aggregation is a direct object pointing action. A power network can be accessed when each object is accompanied with an Aggregation. However, this method cannot reflect the continuity property of the network, which is an important factor in power system simulation. Therefore, instead of using Aggregation, a method based on Association is derived for accessing all elements in a power network. Thus, the Aggregation will not be used and is not shown in Fig. 3.2. Fig. 3.2 clearly shows the Associations among different objects to simulate the connections among network elements. The diagram uses the convention as stated in [30].

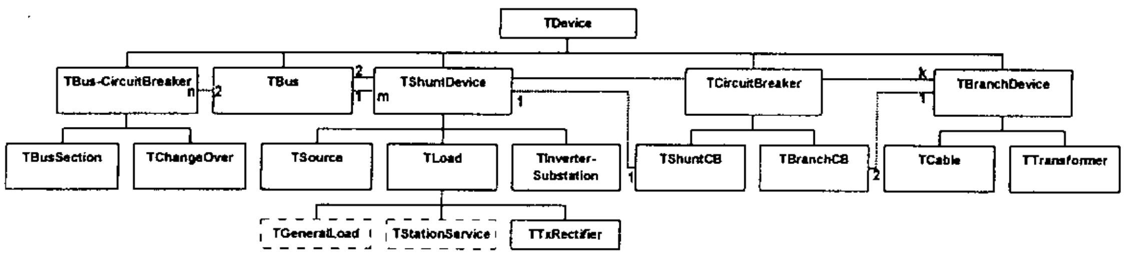


Fig. 3.2 Modified Distribution Circuit Object Model (DCOM)

3.6 Generalised Network Access Method

The dynamical change of network configuration due to various simulation needs can be a complex problem. One may think of the set of data representing the network changes in parallel with the changes in the network. This becomes complicated for a complex system. Since the network structure must be static (fixed) before any one simulation and after many modifications, the methodology aims to direct the computer control passage from one object to another connected object until all objects have been reached, i.e. the whole-connected network is accessed. Thus, the control of simulation is independent of the changes on network. The network access method always collects information from the latest network. In this chapter, the discussion will be concentrated on the development of the network access method.

3.7 Connecting Elements during Network Built-up

As network is built up from objects defined in the preceding chapter, the bi-directional links among the connected objects are established in the process of object creations. This method is known as 'Association'. When a branch device, namely "Branch Device1" connects two buses "Bus1" and "Bus2", Association is established. The object language "Bus1.Branch Device1" means the branch "Branch Device1" of the bus "Bus1". This is a control access by relationship of continuity. Very often, when a simulation requires this connection-type relationship, it cannot be achieved by the method of Aggregation.

3.8 Fundamental Elements of an Network (Structural Consideration)

In view of network topology, Buses are the basic structure of a network; shunt devices are parts of the buses. Branch devices are the only means to link buses. Thus, buses and branch devices are the fundament elements.

$\therefore$ Network = $\sum$ [(Bus, Branch Device) + Associations]

3.9 Implementation of GNAM

Since all elements in a network are connected with the method of Association, there must exist method to reach all elements. We concentrate on the fundamental elements. Starting from a bus, its branch devices can be reached by the bus with language of "Bus.Branch Device". Since branch devices link buses, bus can be reached by object language of "Branch Device.Bus". So, two control statements are adequate for accessing the whole network: one is current bus and the other is control "Bus.Branch Device". By using object language of "Branch Device.Bus", the controls can repeat and eventually reach all elements. But this unlimited repeat will cause overflow even when there are only two buses and one branch device. Identification for the first bus and the next bus is needed. Thus the 'Next' bus must be identified.

One of the main purposes of reaching all elements is to operate ALL elements without repeating. This is an 'unconditional operation' since all objects will be operated. However, it still contains 'conditional passage control'. That means control will only operate those objects that have not been operated before. In this application, a mark of "reached" is equivalent to a mark of "operated". So, only one of the marks "operated" or "reached" is needed. Meanwhile, a method must be found to 'clear' the mark before a new reaching action begins. This becomes one of the applications of the GNAM to be developed.

When only the target elements are concerned, "operated" implies "reached" but "reached" does not imply "operated". Thus, the equality of the previous discussion " 'reached' equivalent to 'operated' " does not apply. Separate marks for these two statuses are needed. Moreover, a second condition for allowing operation is needed (conditional operation). This method applies equally well for situation when targets are distributed discretely; that is when they have no relationship of continuity.

However, by looking at the real situation of application domain, the target objects (network elements) should always contain electrical continuity. For example, in the case of retrieving data of each of the network element for loadflow studies, the control must starts from the source bus to all the branch devices and finally to the loads. This continuity implies that in all cases, target objects must always have paths to the source bus. This is an important concept in applying GNAM to solve problems in the project application domain. The conclusion now becomes "if an object has never been reached, it must be a target". With this conclusion, the implementation of GNAM can be simplified as one having two logic paths only: one is either 'stop' or 'operate' and the other is 'continue'. Fig. 3.3 shows the logic flow of the GNAM.

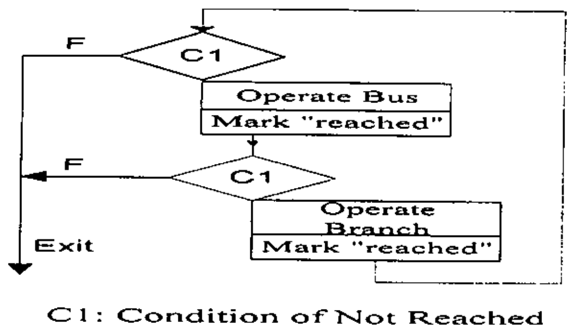


Fig. 3.3 The logic flow of GNAM

3.10 Rules in Applying GNAM.

Two rules must be followed before applying the GNAM. All objects in the network must be inter-connected and continuity exists.

3.11 Applications of GNAM

Object is defined to represent the network elements in the one-line diagram on the schematic. By using this concept, a network can be defined as a sum of network objects and the object links. By viewing the linked network objects as a whole (a generalised network-object linkage model), complex simulation problems regarding to power distribution network restructuring may be resolved into a sum of two or more whole network problems. Each whole-network problem can be solved with the network access method. The method can be implemented by using object-oriented programming technique. In general, the method can be applied to solve any kind of object-oriented simulation problems generated by the single-source MTR power distribution system. In fact, any capabilities of the software related to the whole power distribution system or a single network require the GNAM to access the elements in a whole network. The following is a list of all current applications.

Display Applications

- Display graphical result (GNAM)
- Display circuit (GNAM)
- Clear display (GNAM)

Supporting Applications (Direct use of GNAM)

- Indicate electric flow with colour (GNAM)

- Black out network (GNAM)
- Point to All objects (GNAM)
- Delete All objects (GNAM)
- Save All objects (save one-line diagram) (GNAM)
- Setup network numbering for bus (GNAM)
- Setup network numbering for lines (GNAM)

Operational Application

- Reset the mark "reached" in whole network (GNAM)

Switching Applications (Commands)

- Switching (ON/OFF) generator or
- Switching (ON/OFF) line
- Switching (ON/OFF) change-over bus-coupler (see Chapter 6)

Drawing Applications (Commands)

- Save whole power system
- Close whole power system

Engineering Simulation Applications (Commands)

- Load network data for engineering simulations (Commands)

- Change-over operations (Commands) (see Chapter 6)

- Relays operations (Commands) (see Chapter 6)

3.12 Conclusion

The proposed modified DCOM is tailor-made object model for the MTRC system. The proposed GNAM is a simple but important method for internal operations and engineering simulation of a power distribution system. It provides a fixed logical structure to solve a frequently changed structure (the power system). By using the object-oriented programming language, its implementation is also simple, straightforward and easily understood. With the proposed object model and the Generalised Network-Linkage Model, the developed GNAM are shown to be applicable in various situations.

Chapter 4 Tracking Techniques for DC Traction Loadflow

4.1 Introduction

Train simulation is vital for both long-term and short-term planning of train operation. The development of train simulators could be functionally classified into four major categories [48]: the train performance evaluations, signalling system studies, energy consumption calculation and AC/DC network loadflow solution. The present work is focused on the last one, i.e. AC/DC loadflow. The study was first initiated with the DC traction flow using some tracking techniques, which are found to be much effective than existing methods. The tracking techniques are then extended to the analysis of more complicated AC distribution system of MTRC, to be discussed in the subsequent Chapters. As for the AC loadflow, the very early Gauss-Seidel (GS) method has become obsolete because Newton-Raphson (NR) method is very effective due to the rapid development of *sparsity* technique in the early 70's [49] together with advances of digital computer. However, it is found that some traction loadflow still uses the GS method [50], even though it is well known that the calculation could be speeded up by using the NR technique [48].

The NR technique is based on power mismatch equation and the weakness is that the Jacobian matrix changes with updates in the bus voltage, and the most time-consuming procedure of matrix formation/inversion have to be repeated in every iteration step.

Subsequently, this limitation was solved by modifying the NR to the so-called fast-decoupled (FD) method [51]. It was observed that although FD requires more iteration than that of the NR, the CPU time required could be much reduced because the matrix formation/inversion is performed only once in the FD algorithm. However, in some ill-condition cases, such as in the case with large R/X ratio, the FD converges poorly (or sometimes does not converge at all), the required CPU time may be more than that of the NR.

As for the DC traction loadflow, two conventional load models namely Conductance Model and Current Injection Model [52] have been commonly used to model the non-linear load. In this chapter, three tracking techniques are proposed instead and their effectiveness technique is compared with the conventional methods. The criterion of selecting the most appropriate technique is based on the overall CPU requirement, rather than the rate of convergency.

4.2 The MTR AC System

The MTR system obtains the electricity supply from the two power companies in Hong Kong at 33 kV and has its own 33 kV, 11 kV and 415 V distribution network. As for the train supply, the 33 kV is stepped down to 586 V which is then converted into DC by two 6-pulse series rectifier bridges. Because most of the traction power supply substations have two 4 MW rectifiers connected in parallel, a 24-pulse rectified DC can be obtained by arranging a 15 degree phase shift on the rectifier transformers. The rectified output is connected to a 1500 V overhead line network supplying electricity to the trains.

4.3 Loadflow Modelling Techniques

4.3.1 Conductance and Current Injection Methods

In accordance with [52], two methods can be used for train load modelling:

(a) Conductance model: the train load is represented by a conductance $G=P_L/V^2$, (b) Current Injection model: the train load is represented by a current source $I=P_L/V$.

where P_L is the train power drawn through the catenary at a voltage V. (P_L is negative for regenerative braking.) Both G and I will vary with updated V values. Based on a 168-node system [52], it was found that method (a) converges in 16 iterations and method (b) in 11 iterations. The reason that (a) converges more rapidly than (b) was explained with the aid of some complicated quadratic curves [52].

Alternatively, as one examines the actual load characteristics and the load modelling techniques as shown in Fig. 4.1, it can be seen that method (b) is closer to the actual load characteristics and will converge more rapidly than method (a). Therefore, it is desirable to introduce some modelling techniques to 'track' the load variation closely with voltage changes, for example the line (c) as shown in Fig. 4.1, so that the loadflow might converge rapidly to its true solution.

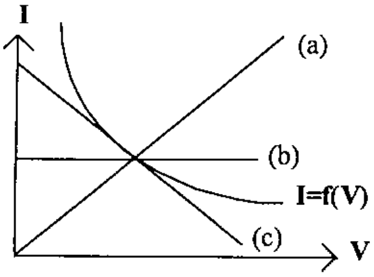
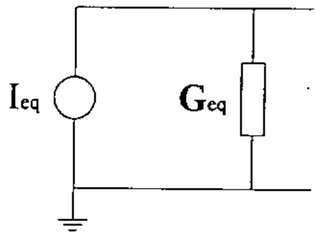


Fig. 4.1 Load characteristic and modeling techniques

(a) Conductance method
(b) Current injection
(c) Norton equivalent

4.3.2 Norton Equivalent Method (c)



Dig. 4.1 Norton equivalent model

A Norton equivalent consists of two elements: an equivalent conductance G_{eq} and an equivalent current I_{eq} (Dig. 4.1). For a load of constant power P_L, the slope $dI/dV = -P_L/V^2$ is negative. Therefore the train load may be represented by a negative equivalent conductance G_{eq} and an equivalent current $I_{eq}=2P_L/V$. Since the source (e.g. transformer rectifier) can also be represented by a Thevenin Equivalent, which can then be transformed to another Norton equivalent, the entire system would consist of circuit conductances and Norton equivalents of sources/loads. With some assumed initial voltages (e.g. V=1pu) for all the load buses, a nodal method [I]=[G][V] is applied and the Norton equivalents of the loads will be adjusted with the updated values of [V]. The process will be repeated until [ΔP] of all the load buses are within a specified tolerance.

4.3.3 Jacobian Method (d)

Similar to the Newton Raphson method, the power mismatch equation at a node i in a

$$\Delta P_i = P^{cal} - P^{spec} = V_i \Sigma(G_{ik} V_k) - P^{spec} \quad (1)$$

where P^{cal} is the calculated power and P^{spec} is the specified power

and the elements of the Jacobian matrix can be obtained by differentiating (1):

$$J_{ii} = \partial P_i / \partial V_i = G_{ii} V_i + P^{cal} / V_i \quad (2a)$$

$$J_{ik} = \partial P_i / \partial V_k = G_{ik} V_i \quad (2b)$$

Thus, the power equations for the DC system can be described by $[\Delta P]=[J][\Delta V]$. With some initial voltage [V], $[\Delta P]$ and $[J]$ are then evaluated and the system voltage is updated by $[\Delta V]=[J]^{-1}[\Delta P]$. The process is repeated until $[\Delta P]$ of all buses (of both sources and loads) are within a specified tolerance. Here, the Jacobian $[J]=[\partial P/\partial V]$ may be regarded as the slope (multi-dimension) of the entire mismatch equations, which tracks the change of power with voltage variation.

Because both the Norton and Jacobian methods are tracking the load characteristic with respect to voltage variation, their similarities are listed in the Table 4-1. It can be seen that the algorithm for the Norton technique is simpler and the CPU time required per iteration is less than that required by the Jacobian method.

		Norton Equivalent	Jacobian
a	Load characteristic	Can be voltage dependant	Constant MW (P^{spec})
b	Criterion used	Slope of I/V characteristic of individual load	Slope of mismatch equations of entire system
c	Number of nodes	M+N (M=source, N=load)	2M*+N
d	Matrix size	(M+N)x(M+N)	(2M+N-1)x(2M+N-1)
e	Matrix sparsity structure	G-structure	G-structure
f	Matrix elements	Simple G value	Function of G and V
g	Elements variation in each iteration	Only diagonal elements	All elements change
h	Nodal mismatch checking	For N load bus only but skipped for nodes with zero load	For all (2M+N-1) nodes
j	Nodal mismatch comparison	Specified load MW versus calculated MW	Specified load MW versus summation of branch MW flow
k	Number of iteration	Relatively less	Slightly more.
m	CPU time per iteration	Relatively less $	Relatively more $

$ Due to d, f, g, h and j of above.

* Since there is an additonal bus for each source.

Table 4-1 Comparison of loadflow methods using tracking techniques.

4.4 Case Study and Modified Technique

4.4.1 Simple 2-bus System

The four aforesaid modelling techniques are firstly applied to the simple 2-bus system (Fig. 4.2) and the result is tabulated in Table 4-2. Because modelling methods (a) and (b) may be regarded as special Norton equivalents each having only one element, the terms G_{eq} and I_{eq} can also be used. From the results, it is observed that method (a) requires 8 iterations to converge, method (b) 6 iterations, method (c) 2 iterations and method (d) 3 iterations.

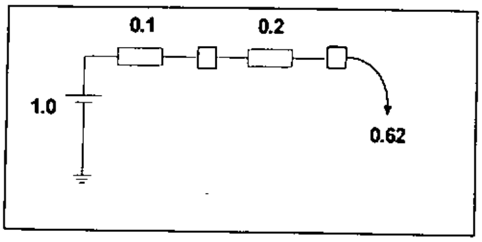
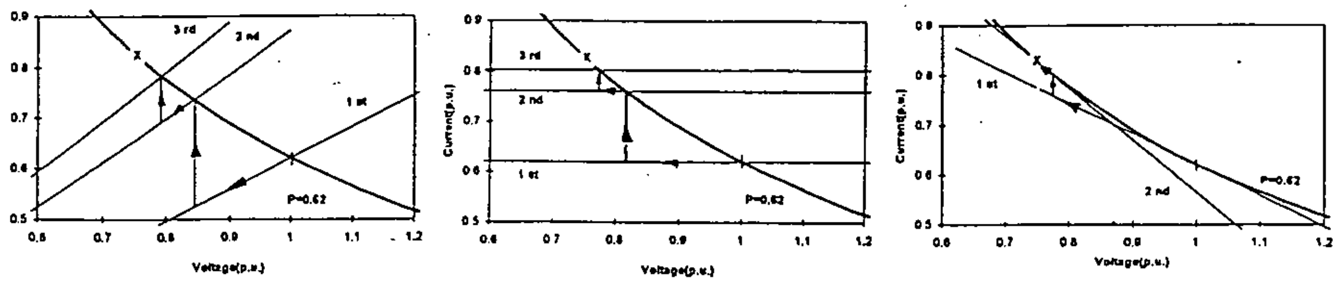


Fig. 4.2 Simple 2-bus system

Methods		Iteration 1	2	3	4	5	6	7	8
(a) Conductance	G_{eq}	0.62	.8721	.9869	1.0415	1.0679	1.0809	1.0873	1.0904
	V	.8432	.7926	.7716	.7619	.7574	.7551	.7540	.7535
	ΔP	.17922	.07210	.03250	.01538	.00744	.00364	.00179	.00088
(b) Current Injection	I_{eq}	0.62	.7617	.8036	.8170	.8213	.8227		
	V	.8140	.7715	.7589	.7549	.7536	.7532		
	ΔP	.11532	.03237	.01012	.00327	.00107	.00035		
(c) Norton Equivalent	G_{eq}	-0.62	-1.042						
	I_{eq}	1.24	1.6073						
	V	.7715	.7532						
	ΔP	.03237	.00035						
(d) Jacobian	V	.8140	.7859	.7531					
	ΔP	.1153	.01011	.00011					
(e) Modified Norton Equivalent	G_{eq}	-0.62	-0.62	-0.62					
	I_{eq}	1.24	1.2820	1.2888					
	V	.7715	.7560	.7535					
	ΔP	.03237	.00518	.00089					

Table 4-2 Results for the 2-bus systems (ε=0.001 pu)

Moreover, the parameter variations for the first three methods may be inferred from Fig. 4.3. In Fig. 4.3(a), for instance, the iteration starts with (V,I)=(1, 0.62) on the load characteristic curve P=0.62. After the 1st iteration, the calculated (V,I)=(0.8432, 0.5228) lies on the first approximation of G=0.62 straight line. Then the 2nd iteration begins with (0.8432, 0.7353) and after the 8th iteration, it finally converges at (0.7535, 0.8228) indicated by a 'x' in Fig. 4.3(a). Since the load model is almost orthogonal to the load characteristic, the path along the line of G=constant will deviate from the load curve, resulting in poor convergency for the conductance method (a). On the other hand, the path in Fig. 4.3(c) is very close to the load characterises and method (c) converges rapidly in two iterations.



(a) Conductance (b) Current Injection (c) Norton Equivalent

Fig. 4.3 Paths in the first two iterations for different methods

4.4.2 The MTR Traction System

The next step is to apply the same techniques to the DC traction system of the MTR system. In this case, the system has a total of 120 nodes (18 rectifier substations plus 102 trains). The result is tabulated in Table 4-3 for different MW tolerances. It is observed that method (c) converges rapidly in three iterations, but the iteration counts for other methods would increase with decrease in tolerances. The two conventional methods (a) & (b) have poor convergent characteristics while the two tracking methods (c) & (d) converge rapidly. In particular, (c) requires less iteration counts than (d), which may be attributed to the fact that the individual load characteristic is more linear than the power mismatch equations.

			Methods		
ε	a	b	c	d	e
.00001	10	7	3	3	4
.000001	12	9	3	4	5
.0000001	14	10	3	4	5

Table 4-3 Iteration count for the MTR system

			Methods		
ε	a	b	c	d	e
.00001	80.2	23.7	26.1	46.7	17.3
.000001	96.3	28.8	26.1	62.0	20.1
.0000001	112.4	31.4	26.1	62.0	20.1

Table 4-4 CPU time (msec) for the iterative with varying MW tolerance ε calculation only

4.4.3 Bi-factorization Sparsity Technique

For a longitudinal network such as the traction system, normally there are two branches connected to a node, this implies that the system will have 97% zero elements for a 100-node system, 99% for a 300-node system. In the present study, sparsity technique of Zollenkopf bi-factorization is applied [49]. The sparsity algorithm is divided into four steps: (i) sparse structure formation, (ii) optimal ordering, (iii) matrix factorisation, and (iv) solution by forward and backward substitutions. The most time consuming process is (ii), followed by (iii), (i) and (iv). In the aforesaid four methods, steps (i) and (ii) are performed once only as the sparse structure remains unchanged. Step (iii) will be repeated only if there is any change in the system matrix elements, such as the Jacobian elements in method (d) and the G_{eq} (diagonal) in methods (a) & (c). Thus, the factorisation in step (iii) has to be repeated for (a), (c) & (d), but it is required *once* only for (b). (Step (iv) must be operated in every iteration for all methods.)

4.4.4 Comparison Using CPU Time

The effectiveness of the four aforesaid methods is once more compared by the total CPU time required as shown in Table 4-4, and it can be observed that

1. Method (a) is the slowest (i.e. largest CPU time), due to the largest number of iteration count.
2. Method (d) is also slow because it requires more CPU time per iteration (see Table 4-4).
3. The CPU for (b) is comparable to that of (c) although it requires more iterations than (c).

4.4.5 Modified Norton Equivalent Method (e)

Although the Norton method (c) has the best convergency characteristics, the main weakness is that factorisation is required in every iteration step as the 'slope' changes with updated V values. From Fig. 3, one can observe that the load characteristic is roughly linear within the *normal* operating voltage range (e.g. V=0.9 to 1.0 p.u.). Thus, the Norton method can be modified by keeping the slope unchanged in all subsequent iterations so that the factorisation is performed once only. From Table 4-2, it can be seen that with the slope fixed at G_{eq}=–0.62, the current I_{eq} has to be adjusted in every iteration to match the load characteristic. Because the subsequent G_{eq}'s are not exactly tangential to the load characteristics, the iteration number is increased by one in this case.

The study is repeated on the MTR system and the result is tabulated in Table 4-3 and Table 4-4 as well. It is observed that, although method (e) requires more iteration counts, it is faster than (c) by 34% for ε=0.00001 MW and 23% for smaller ε values.

4.5 Selection of Tolerance

The criterion of loadflow study is to determine the voltage such that the power mismatch at any bus is within a specified MVA tolerance. Because the rectifier rating in the MTR system is 4 MW and the maximum train loading would not exceed 1 MW, an appropriate tolerance would be ε=0.00001 MW.

However in some DC loadflow [52], the voltage difference ΔV between voltage obtained in successive iterations is used as the tolerance instead. This approach has the following discrepancies:

1. The necessity to store the previous [V] vector.

2. Since $P=VI$ and hence $\Delta P=V\Delta I+I\Delta V$, convergency in voltage may not imply power convergency, or vice versa. For example, from Table 4-2, the power converges for ΔV=0.7715– 0.7532=0.0183 in the 2nd iteration for method (c), whilst in the 7th iteration for method (a) the power is not yet convergent even with ΔV=0.7551–0.7540=0.0011.

3. It is difficult to co-relate individual bus V and ΔP in a large network. In order to ensure that ΔP is within an acceptable value, a very small ΔV is usually used, e.g. 0.000013 pu in [52]. This in turn requires more iteration counts e.g. 16 and 11 respectively [52] for methods (a) and (b).

4.6 Conclusion

Traditionally, two load modelling methods were applied to model the train load for DC traction loadflow, but they converge slowly. In order to increase the rate of convergency, three load tracking techniques are proposed. It is found that if the load is modelled by an exact slope to the load characteristics, the system converges rapidly. However, if the CPU time is considered, the fastest method is the one that uses an 'approximate' slope. Therefore, the modified Norton Equivalent load modelling method is most effective for DC traction loadflow. The 'slope' concept is further explored in the AC loadflow analysis of the next chapter.

Chapter 5 Effective Loadflow Technique with Voltage-dependent MVA Load

5.1 Introduction

Conventional loadflow technique for AC transmission system (e.g. 132kV) has three different types of buses, namely: slack, PV and PQ, in which the PQ (load) buses are assumed to have a constant MVA of $S=P+jQ$. This assumption is justified because the automatic on-load tap changer in the transformer (e.g. 132/11 kV) functions to maintain the LV side voltage constant, and hence the MVA load on the LV side becomes constant irrespective of voltage changes on the HV side. As a result, the refused MVA load on the HV side is approximately constant because the transformer loss is relatively small and the loss variation due to load changes is even smaller.

However, this constant MVA load assumption may not be valid for distribution systems. For instance, the power distribution system of the Mass Transit Railway (MTR) Corporation consists of 33, 11, and 0.415 kV buses connecting to the various loads including chillier plant, passenger-station, traction and drive etc., and each of these buses has different voltage dependant characteristics described by [53] and [54] as $S=P_0V^a+jQ_0V^b$. Moreover, the Railway is one of the busiest urban transit systems in the world as it provides the people of Hong Kong a safe and reliable train service, 19 hours a day, 365 days a year to carry more than 5 millions passengers everyday. When delays such as that due to problems on the supply system occur, the Railway must endeavour to achieve the quickest service recovery in a safest manner. Hence, it is vital to introduce a loadflow technique which is:

1. capable to look after the different load characteristics exhibited in the system, and

2. fast enough for the operation engineer to make quick decisions in emergency cases.

5.2 The MTR AC Supply System

The two local power companies supply electricity to the MTR Urban Lines system through 132/33kV transformers at four infeed points: two from China Light & Power Co. Ltd at Kowloon Bay and Kwai Fong and two from Hong Kong Electric Co. Ltd at Admiralty and Chai Wan. Power will then be converted to a 11kV distribution system at strategic locations and a 1500V DC at traction substations.

Two automatic changeover schemes have been incorporated in the power supply system to achieve a high continuity of power supply to ensure safe operation of the railway. When the power supply from a power company fails, the 33kV changeover scheme will operate to maintain the railway in full service. Each passenger station is supplied with 11kV cables connecting to different sources. When a fault occurs on the 33kV or the 11kV network, changeover on the LV circuits will maintain the station essential loads to keep the station in operation.

5.3 Loadflow Modelling Techniques

5.3.1 Modelling in Nodal Method

Although the MTR distribution system has several sub-systems, the primitive nodal method of [I]=[Y][V] can be applied for loadflow studies, because in each sub-system there is only one single source having a reference angle of zero degree. Thus this source can be represented by a Thevenin model, which is then transformed to a Norton equivalent. As for the AC load, the two modelling methods as mentioned in Chapter 4 for DC are applicable with sight modification:

(i) Admittance model: a load is represented by an admittance $Y=S^{*}/V^{2}$ (S itself is voltage dependent) and the voltage V is to be adjusted in each iteration.

(ii) Current Injection model: a load is represented by a current source $I=S^{*}/V^{*}$ which is being varied with adjustments in V.

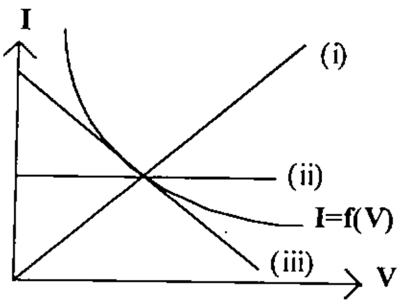


Fig. 5.1 Load modelling in DC system

As already pointed out in Chapter 4, (ii) converges much faster than (i) (Fig. 5.1). Interpretation of the above two models can be illustrated by means of the *I-V* relationship of a DC system where $I=f(V)=P/V=P_oV^{a-1}$ (where a is an exponential constanct) as shown in Fig. 5.1. The reason that (ii) has a better convergency is attributed to that the fact that it is closer to the actual load characteristics than (i). Therefore, it is envisaged that if the load is so modelled that it is *always* tangential to the load characteristics, for example the line (iii) as shown in Fig. 5.1, the loadflow might rapidly converge to its true solution.

5.3.2 Norton Equivalent Load Modelling

A Norton equivalent consists of two elements: Y_{eq} and I_{eq}. In a DC system, the tangential Y_{eq} characteristic can be obtained by simple differentiation dI/dV. In AC system, however, the dI/dV has to be determined by splitting the real and imaginary parts as shown below. Consequently, $[\mathbf{Y}_{eq}]=\Delta\mathbf{I}/\Delta\mathbf{V}$ is a 2x2 square matrix defined by the four parameters.

Consider a bus with a voltage-dependent load characteristic given by

$$\bar{S} = P + jQ = P_oV^a + Q_oV^b$$

$$\bar{I} = \frac{\bar{S}^*}{V^*} = \frac{\bar{S}^*\bar{V}}{V^2} = \frac{1}{V^2}(P - jQ)(V_R + jV_J)$$

$$\begin{bmatrix} I_R \\ I_J \end{bmatrix} = \frac{1}{V^2}\begin{bmatrix} P & Q \\ -Q & P \end{bmatrix}\begin{bmatrix} V_R \\ V_J \end{bmatrix}$$

By differentiation,

$$\begin{bmatrix}\Delta I_R\\ \Delta I_J\end{bmatrix}=\frac{1}{V^2}\begin{bmatrix}P & Q\\ -Q & P\end{bmatrix}\begin{bmatrix}\Delta V_R\\ \Delta V_J\end{bmatrix}+\frac{1}{V^2}\begin{bmatrix}\Delta P & \Delta Q\\ -\Delta Q & \Delta P\end{bmatrix}\begin{bmatrix}V_R\\ V_J\end{bmatrix}-\frac{2\Delta V}{V^3}\begin{bmatrix}P & Q\\ -Q & P\end{bmatrix}\begin{bmatrix}V_R\\ V_J\end{bmatrix} \quad (5.1)$$

where $$\begin{bmatrix}\Delta P\\ \Delta Q\end{bmatrix}=\frac{aP_oV^{a-1}}{bQ_oV^{b-1}}\Delta V=\frac{aP}{bQ}\frac{\Delta V}{V} \quad (5.2)$$

Since $V^2=V_R{}^2+V_J{}^2$ by differentiation,

$$\Delta V=\frac{V_R}{V}\Delta V_R+\frac{V_J}{V}\Delta V_J \quad (5.3)$$

Put (5.3) and (5.2) into (5.1), (5.1) becomes

$$\begin{bmatrix}\Delta I_R\\ \Delta I_J\end{bmatrix}=\begin{bmatrix}Y_{RR} & Y_{RJ}\\ Y_{JR} & Y_{JJ}\end{bmatrix}\begin{bmatrix}\Delta V_R\\ \Delta V_J\end{bmatrix}$$

Where $$Y_{RR}=\frac{(a-2)PV_R^2}{V^4}+\frac{(b-2)QV_RV_J}{V^4}+\frac{P}{V^2} \quad (5.4a)$$

$$Y_{RJ}=\frac{(a-2)PV_RV_J}{V^4}+\frac{(b-2)QV_J^2}{V^4}+\frac{Q}{V^2} \quad (5.4b)$$

$$Y_{JR}=\frac{(a-2)PV_RV_J}{V^4}-\frac{(b-2)QV_R^2}{V^4}-\frac{Q}{V^2} \quad (5.4c)$$

$$Y_{JJ}=\frac{(a-2)PV_J^2}{V^4}-\frac{(b-2)QV_RV_J}{V^4}+\frac{P}{V^2} \quad (5.4d)$$

Therefore this $\mathbf{Y}_{eq}$ is only a fictitious admittance since it cannot be expressed in the form of G+jB.

Once the load equivalent admittance [Y_{eq}] is determined, the equivalent current vector [I_{eq}] can be calculated by [I_{eq}]=[I]-[Y_{eq}][V] (see Fig. 5.2), where [I] is the load current. Finally, the entire AC system would consist of circuit admittances and Norton equivalents of sources/loads. With some assumed initial voltages (e.g. $1\angle 0°$) for all load buses, a nodal method is applied and the Norton equivalents of the loads will be adjusted with the updated [V]. The process will be repeated until [ΔP] and [ΔQ] of all the load buses are within a specified tolerance.

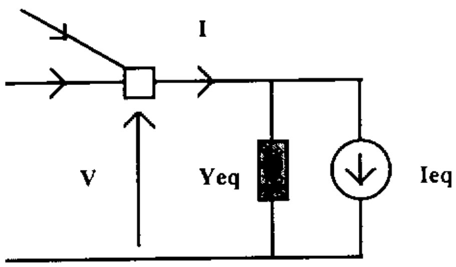


Fig. 5.2 Load represented by Norton equivalent

5.4 Case Studies

5.4.1 Simple 2-bus System

The three aforesaid modelling techniques are firstly applied to the simple 2-bus system (Fig. 5.3) and the result is tabulated in Table 5-1. Because modelling methods (i) and (ii) may be regarded as special Norton equivalents each having only one element, the terms **Y_{eq}** and **I_{eq}** can also be used. From the results, it is observed that:

a. Methods (i) and (ii) both require 7 iterations,

b. The proposed Norton equivalent technique (iii) converges after 2 iterations.

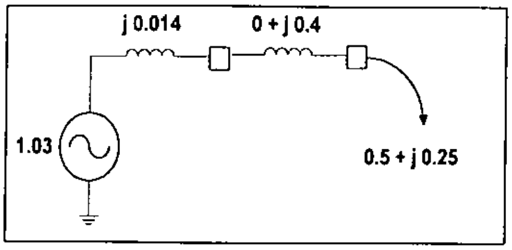


Fig. 5.3 Simple 2-bus system

5.4.2 The MTR 139-bus Subsystem

The next step is to apply these three techniques to the MTR urban line AC distribution system. In order to increase the size of individual sub-system, it is assumed that a supply failure occurs in one of the subsystems, the changeover scheme operates and two of the subsystems are connected to form one large subsystem, having 139 buses. Only the result of this combined subsystem is then tabulated in Table 5-2 for different MVA tolerance and with constant MVA loads assumed. It can be seen that the proposed technique (iii) converges rapidly in three iterations. As for non-constant MVA loads and with the approximate assumption that the exponential constants a and b are identical for all buses (in actual case they are not), the results are computed as shown in Table 5-3: method (iii) converges rapidly by not more than three iterations. Thus, the fast convergency of the proposed technique is confirmed.

5.4.3 Conventional Newton-Raphson Method

A well-known iterative method based on tangential approach is the Newton-Raphson

					Iteration			
Methods		1	2	3	4	5	6	7
(i) Admittance Method	$\mathbf{Y}_{eq}$	0.5	0.5941	0.6229	0.6320	0.6349	0.6359	0.6362
		-0.25	-.2970	-.3114	-.3160	-.3175	-.3180	-.3181
	VR	0.9017	0.8752	0.8671	0.8645	0.8637	0.8635	0.8634
	V_J	-.1691	-.1917	-.1981	-.2000	-.2007	-.2009	-.2009
	ΔP	.07920	.02308	.00723	.00232	.00075	.00024	.00008
	ΔQ	.03960	.01542	.00362	.00116	.00037	.00012	.00004
(ii) Current Injection	$\mathbf{I}_{eq}$	0.5	0.4566	0.4865	.4830	0.4855	0.4852	0.4854
		-.25	-.3718	-.3903	-.4000	-.4015	-.4023	-.4025
	V_R	0.9265	0.8761	0.8684	0.8644	.8638	.8634	0.8634
	V_J	-.2076	-.1890	-.2014	-.2000	-.2010	-.2009	-.2010
	ΔP	-.01500	.02971	-.00111	.00249	-.00010	.00022	.00000
	ΔQ	.12188	.01055	.00902	.00089	.00077	.00008	.00007
(iii) Norton Equivalent	Y_{RR}	-0.5	-.4106					
	Y_{RJ}	0.25	0.5652					
	Y_{JR}	0.25	0.5652					
	Y_{JJ}	0.5	0.4106					
	Ieq	1.0	.9534					
		-0.5	-.8081					
	V_R	0.8690	.8633					
	V_J	-.2126	-.2010					
	ΔP	-.02782	.00003					
	ΔQ	.02082	-.00009					
(iv) Newton Raphson	V	0.8690	0.8681	0.8664				
	δ	-.2122	-.2290	-.2287				
	ΔP	-.04692	.00044	.00000				
	ΔQ	0.04130	.00037	.00000				
(v) Modified Norton Equivalent	Y_{RR}	-0.5	-0.5	-0.5	-0.5			
	Y_{RJ}	0.25	0.25	0.25	0.25			
	Y_{JR}	0.25	0.25	0.25	0.25			
	Y_{JJ}	0.5	0.5	0.5	0.5			
	Ieq	1.0	.9643	.9679	.9673			
		-0.5	-.5152	-.5175	-.5179			
	V_R	0.8690	.8649	.8635	.8634			
	V_J	-.2122	-.1995	-.2012	-.2009			
	ΔP	-.02782	-.00361	-.00046	-.00006			
	ΔQ	.02082	.00126	.00042	.00003			

Note: 1. Real and imaginary parts are shown for $\mathbf{Y}_{eq}$ and $\mathbf{I}_{eq}$.

2. When a=b=0, Y_{RR}=-Y_{JJ}, and Y_{RJ}=Y_{JR}.

Table 5-1 Results for the 2-bus systems (a=b=0, MVA tolerance ε=0.0002 pu)

(NR) method in which the 'Jacobian' matrix may be regarded as the 'slope' of the entire mismatch equations. The above studies (except the case for non-constant MVA load) are then repeated using NR and the results are tabulated also in Table 5-2a and Table 5-2b for easy comparison. It is observed that the NR in general requires more iteration than the proposed Norton technique. Moreover, since both the NR and the Norton methods are based on 'tangential' approach, their similarities are listed in Table 5-3. It can be seen that the algorithm for the Norton technique is simpler and the CPU time required per iteration is less than the NR.

5.5 Modified Norton Equivalent

The number of iteration counts for the four methods has been compared in Table 5-2a and Table 5-2b. As already discussed in Chapter 4, the choice of method should be based on the computer time, rather than the iteration count. Thus, the result on computation time required for the four methods is listed in Table 5-2c. Table 5-2c shows that, although method (iii) converges fast (in 3 iteration), method (ii) is more desirable since it requires less CPU time. This is because the Y matrix in (ii) remains unchanged during the iteration.

However, if the Norton Equivalent method (iii) is so modified that the 2x2 matrix calculated in the first iteration is kept constant for the each load, and only the I_{eq} is adjusted (e.g. see (v) in Table 5-1), the system Y-matrix would be constant, the CPU time is much reduced, as seen from the last column of Table 5-2c. Therefore, it can be concluded that, the Modified Norton Equivalent technique (v), although, converges slower than the two 'exactly-tangential' methods (iii) and (iv), it the fastest and most suitable for the loadflow calculation of AC distribution system of MTRC.

	Methods			
ε	(i)	(ii)	(iii)	(iv)
.0002	9	8	3	4
.0001	10	9	3	4
.00002	13	11	3	4
.00001	15	12	3	4
.000002	18	15	3	5

Table 5-2a Iteration count for the MTR system. (a=b=0)

		a				
		0.0	0.5	1.0	1.5	2.0
	0.0	15,12,3	11,8,3	9,6,3	9,5,3	8,5,3
	0.5	10,11,3	8,8,3	7,6,3	6,5,3	6,6,2
b	1.0	8,11,3	7,8,3	6,6,3	5,6,2	5,7,2
	1.5	7,11,3	5,8,3	5,6,3	4,6,2	3,7,2
	2.0	6,11,3	5,9,3	4,6,2	3,7,2	1,7,1

Note: 1. The shaded study is the same as that of Table 5-2a.

2. When a=b=2, $\mathbf{Y}_{eq}$ is independent of V, and one iteration is necessary for (i) and (iii).

Table 5-2b Iteration count for the MTR system using with varying p.u. MVA tolerance ε methods (i), (ii) and (iii) with different a & b values. (ε=0.00001 pu)

	Methods				
ε	(i)	(ii)	(iii)	(iv)	(v)
.0002	4850 (9)	834 (8)	1609 (3)	2151 (4)	676 (5)
.0001	5382(10)	875 (9)	1609 (3)	2151 (4)	800 (6)
.00002	6965(13)	958 (11)	1609 (3)	2151 (4)	800 (7)
.00001	8052(15)	1000(12)	1609 (3)	2151 (4)	800 (7)
.000002	9630(18)	1124 (15)	1609 (3)	3128 (5)	842 (8)

Table 5-2c computation time (msec) and iteration count (bracketed) for the MTR system with varying p.u. MVA tolerance ε. (a=b=0)

		(iii) Norton Equivalent	(iv) Newton Raphson
a	Load characteristic	Can be voltage dependant	Constant P & Q
b	Criterion used	Slope of I/V characteristic of individual load	Slope of mismatch equations of entire system
c	Number of nodes	N nodes	N nodes + one slack
d	Matrix size	2Nx2N real matrix	Same
e	Matrix sparsity structure	Y-structure	Same
f	Matrix elements	Simple G or B	Function of G, B, V, sinδ, and cosδ
g	Elements variation in each iteration	Only diagonal block elements change	All elements change
h	Nodal mismatch checking	Skipped for nodes without load	For all the N nodes
j	Nodal mismatch comparison	Specified load MVA vs calculated MVA	Specified load MVA vs summation of branch MVA flow
k	Number of iteration	Relatively less	Slightly more.
m	CPU time per iteration	Relatively less $	Relatively more $

$ Due to f, g, h and j of above.

Table 5-3 Comparison of loadflow methods using tangential approach

5.5.1 Conventional Fast-decoupled Method

Another method using tangential (approximate) approach, the fast-decoupled (FD) method, is also employed to this study. It is found that this FD method does not converge for the present system, because the MTR distribution network mainly consists of cables of large R/X ratio.

5.6 Conclusion

AC loadflow techniques have been well established. However, as for the MTR AC distribution system having voltage dependent load characteristics, the primitive nodal techniques of Norton Equivalent can be applied instead.

When compared with the conventional methods by assuming constant MVA loads, it is found that the fast-decoupled method could not converge because the cables in the MTR system have high R/X ratio. As compared to the technique based on Newton Raphson (NR) approach, the proposed Norton technique is slightly faster. In fact, both methods are both using *slopes*: in the NR technique, the 'Jacobian' is the *slope* of the entire mismatch equations, whilst the Norton technique is the *slope* of individual load. The advantage of the Norton technique may be attributed to the fact that the individual load characteristic is more linear than the combined mismatch equations.

Moreover, if the Norton Equivalent is further modified by keeping the 'slope' constant, the overall computation time is much reduced (despite the increase of iteration count).

The major advantage of the proposed Modified Norton Equivalent AC loadflow technique, as compared to the conventional loadflow technique, is that it can be applied to any type of load with well-defined characteristics (i.e. the expontenial constants a and b well-defined). However, the restriction is the requirement of having only single source in each sub-system. Nevertheless, the proposed Modified Norton Equivalent is most effective in the study of the MTR distribution system.

Chapter 6 Fault Analysis, Changeover and Relay System

6.1 Fault Analysis

One of the major objectives of the developed software is to ensure the fault levels in the entire MTR AC system are within the capacity of the circuit breakers, under all foreseeable operating conditions. These include normal operating condition and in cases of system contingency (e.g. subsequent to a fault and with system rearrangement, or the so-called MTR changeover system).

6.2 Effect of Transformer Phase Shift

The digital solution of fault analysis is described in [55]. Based on conventional fault analysis, the fault current (at fault point), bus fault voltage and branch fault flow can be calculated. Because the transformers of MTR system are all of delta-wye-11 type with solidly earthed wye, 30E phase shift will occur. Of course, the phase shift will not affect the loadflow nor the balanced 3-phase fault as the positive quantities (current and voltage) are all shifted by the same amount of 30E. However, this would significantly affect the unbalanced fault because the negative sequence quantities will be at the same time shifted by -30E. For instance, if a single line-to-ground fault occurs in the *a*-phase of the transformer wye side, the fault flow on the *a*-*b*-*c* phases will be 2, -0.4 and -0.4 pu respectively for transformer **A**, and 0.4, 0.4 and 0.4 pu for the transformer **B**, as shown in Fig. 6.1.

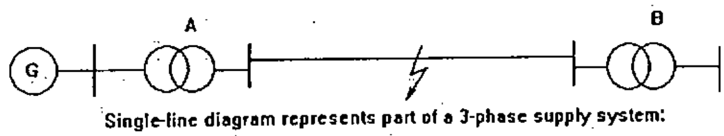


Fig. 6.1a L-G fault at the middle of the line.

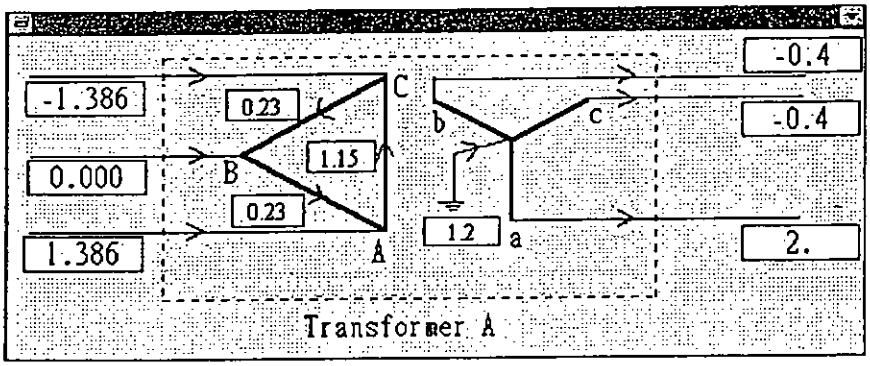


Fig. 6.1b Winding and phase current flows of Transformer A

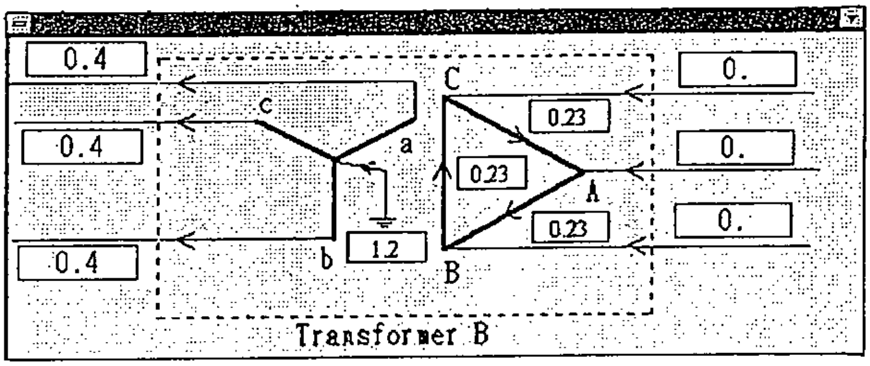


Fig. 6.1c Winding and phase current flows of Transformer B

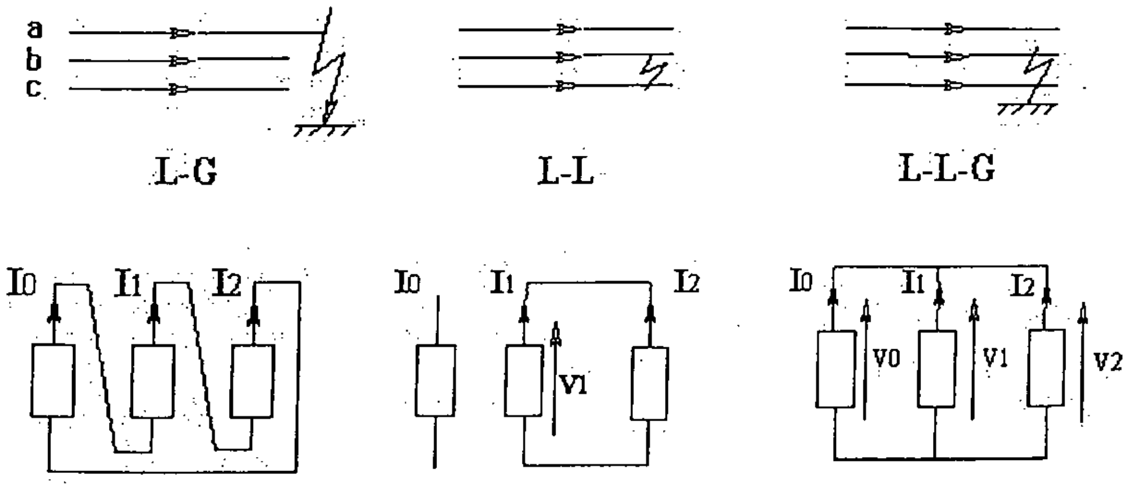


Fig. 6.2 Determination of sequence fault current at fault point

However, the fault flows on the delta side of **A** becomes -1.39, 0 and 1.39 pu, but zero for **B**. Indeed, the whole fault flow pattern is significantly affected by the transformer phase shift, and it is vital to consider the transformer phase shift in the present software.

6.3 Determination of Effect of Phase Shift

In the present software, the fault calculation with the effect of transformer 30E phase shifts can be summarised in the following steps:

S1. The three-sequence fault currents I_f are determined by solving the network equation of Fig. 6.2, in accordance to the type of fault.

S2. The sequence voltage of bus j is determined by

$$V_j=E_j-I_fZ_{jk} \quad (6.1)$$

where

V_j is sequence fault voltage,

E_j is sequence prefault voltage (E_j=0 for negative and zero sequences),

I_f is sequence fault current calculated in step S1, and

Z_{jf} is an element of system impedance matrix.

S3. The branch sequence current flow between bus **m** and **n** is calculated by

$$I_{mn}=(V_m-V_n)Y_{mn} \quad (6.2)$$

where

V_m and V_n are sequence voltages and

Y_{mn} is branch admittance.

S4. Based on V_j calculated in S2, the phase shift ϕ (in degree) of V_j (already calculated in step 2) is determined by

$$\phi=(-1)^m(30°)f(V_j,V_f) \tag{6.3}$$

where

m=2 for positive sequence, m=1 for negative sequence (zero sequence has no phase shift).

f is a function of nominal voltage level of bus j and that of fault point f, given by Table 6-1.

S5. The I_{mn} calculated in step S3 is likewise adjusted by (6.3) to determine the phase-shift.

S6. Finally, the sequence quantities (voltage and current) obtained in steps S4 and S5 are converted to phase quantities by multiplying a matrix **T** where

$$\mathbf{T}=\begin{bmatrix}1 & 1 & 1\\ 1 & h^2 & h\\ 1 & h & h^2\end{bmatrix}$$

where $h=\angle 120^0$ operator

		Fault bus voltage in kV, V_f			
		132	33	11(or 3.3)	0.43
Bus	132	0	-1	-2	-3
Voltage	33	+1	0	-1	-2
in kV,	11(or 3.3)	+2	+1	0	-1
V_j	0.43	+3	+2	+1	0

Table 6-1 Values of $f(V_j, V_f)$.

6.4 MTR 33kV Changeover System

The MTR power system consists of four electrically separated networks. Each network is connected with the other two adjacent networks by two normally opened busbar couplers. One infeed source supplies electricity to one network only. Electricity is stepped down to 33kV from the infeed 132kV level. Therefore, the 33kV network is the backbone network of the power system (Fig. 6.3). Contingency occurs when one or more infeed sources fail. The MTR adopts a 33kV Changeover System to maintain power supply as far as possible during system contingency. Depending on the number of infeed source failure, contingencies are classified into 3 levels named as 1st, 2nd and 3rd order contingency. Depending on the number of infeed source in failure, the 1st and 3rd order contingencies are further divided into 4 divisions while the 2nd order contingency is divided into two divisions [56].

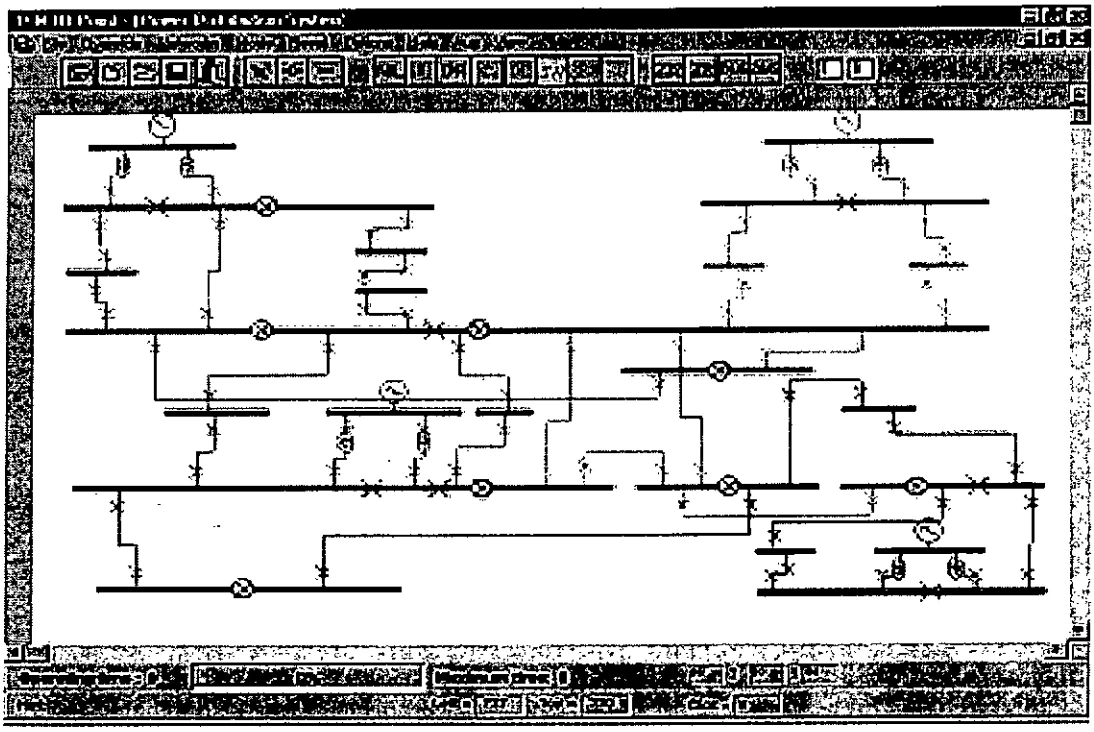


Fig. 6.3 The MTR 33kV networks (for urban lines only).

The simulation of the 33kV changeover system follows the rules and divisions. Each division of a contingency is implemented by a command name. The command will operate a pre-set list of normally opened/closed busbar coupler(s) and cable circuit breakers for electricity transfer for if necessary. By making use of object-oriented programming, when a closing operation occur on a busbar coupler, the two adjacent busbars of the busbar coupler will be merged into one single busbar; whereas when a opening operation occur on a busbar coupler, the single busbar will split into two separate busbars. The rearranged network will then be recognised by the GNAM as described in Chapter 3.

The circuit name of each busbar coupler and each line circuit breaker is edited with the corresponding edit bar. The pre-set list can be established by selecting from the list of available busbar couplers and line circuit breakers in the changeover system dialog (Fig. 6.4). The available list can be edited independently. The pre-set list can be modified with the controls in the middle column (Fig. 6.4).

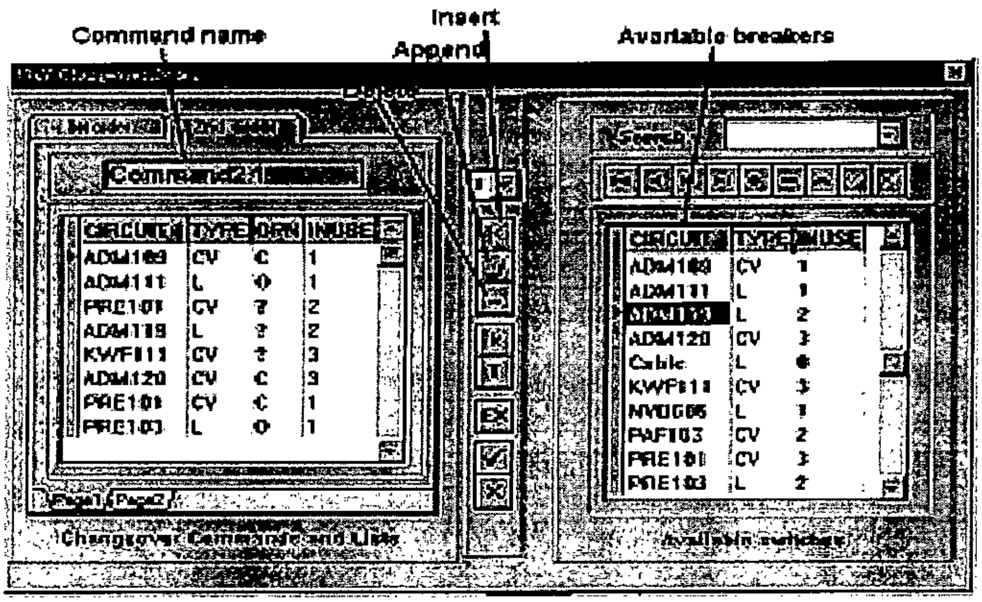


Fig. 6.4 Changeover list dialog box.

A communication dialog displays the status of the couplers and circuit breaker when they are switched by the changeover command. Single step or full automatic changeover is available for choosing (Fig. 6.5). Restorations are implemented by reverse operations (Fig. 6.5).

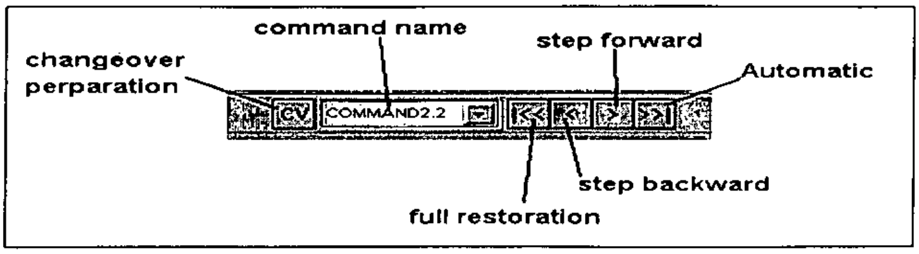


Fig. 6.5 Changeover command and operations tools located in the simulation bar.

6.5 Relay Operation

Based on the current flow or the fault current flow, the operating time of each relay is calculated from the common relay equation for O/C or E/F IDMT relay. Relay types include IDMT (VI, SI, XI) and DTL. Relay operation based on a circuit search with respect to the Relay Database (Fig. 6.1). The graphics display the breaking sequences or the operating time for the case when all relays fail to work by pressing the tools in Fig. 6.7. The correct operating time is simulated by the other tools in Fig. 6.7. In this case, all relays are assumed to be work normally. When a relay operates a C.B., the power system update the configuration automatically and the current flow or the fault flow will be recalculated if the fault cannot be cleared by this C.B. operation. The relays operating time will then be recalculated too. These processes repeat until all faults are cleared or in the case when no relay should operate.

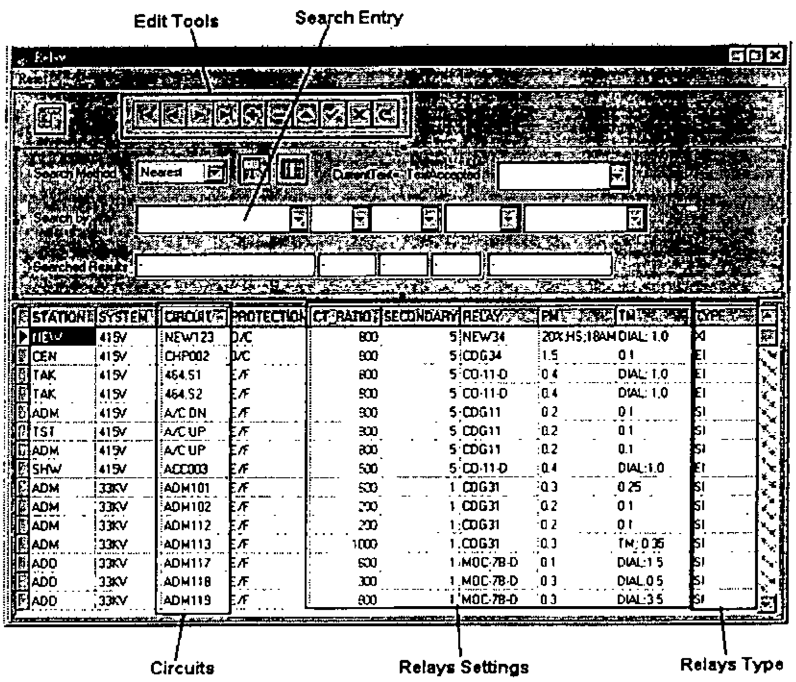


Fig. 6.6 Relay list dialog box.

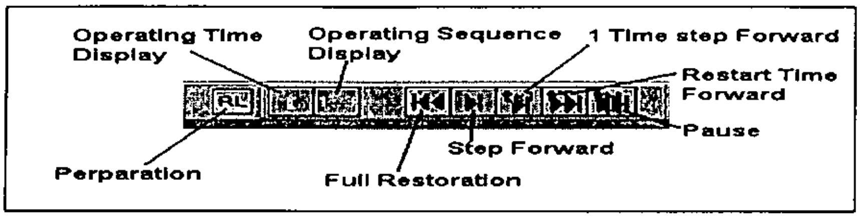


Fig. 6.7 Relay simulation bar.

6.6 Conclusion

Fault analysis, the MTR 33kV changeover system and relay operations are successfully implemented. The fault analysis is generally based on conventional computational method. However, because of the MTR special characteristic, treatment of transformer phase-shift correction can be much simplified. The simulation of the MTR 33kV changeover system is mainly based on the GNAM to get the correct network configuration after system changeover. It again validates the importance of the proposed GNAM.

Chapter 7 Summary, Conclusion, Application and Further Work

7.1 Summary of the Research and Development work

The work for this research project can be summarised into three areas namely the research contents, the software development and the capability in engineering simulation.

7.1.1 Research

- A systematic programme with high degree of modularity on the designed distributed software architecture is proposed.

- The proposed Generalised Network Access Method (GNAM) has successfully been implemented with available object-oriented technology to solve engineering problem of accessing time-changing network.

- The proposed AC and DC Norton equivalent model are suitable and efficient for loadflow studies on voltage dependent loads in AC power distribution system and in DC railway traction system respectively.

7.1.2 Development

- The software combines graphic capability and engineering calculations.
- The software incorporated relay database system as well.
- The software was designed especially for the MTR AC distribution System, e.g. tailor-made graphics and tailor-made input/output data display format.
- The software incorporated Windows operating system environment, which is suitable in running on the computers of the engineers in MTRC.

7.1.3 Capability

The software provides capabilities in

- performing simulation of AC loadflow
- performing simulation of fault analysis.
- calculation of the relay operating sequence and operating time during faults.
- performing simulation of auto-relay operations for fault clearance.
- performing simulation of the MTR 33kV changeover system

7.2 Contribution to the MTR Corporation

The contributions of this software to the MTR Corporation are as follows

- The display of AC power distribution system is of 95% same as the one-line diagram in the MTRC.
- The software provides different colours to identify different MTR AC power networks.
- The software provides high automation and fast simulation.
- The software provides unique features of steps or automatic simulation of the 33kV changeover and restoration system.
- The software provides a user-friendly environment for the engineers in MTRC.
- The software is fully interactive in Windows 95 platform.
- The software provides indications of flow of current, network blackout and line overloading.
- The software supports loadflow, fault analysis, switching, relay operations, and 33kV/11kV changeover.
- The software supports fast building of a new on-screen AC power distribution system.

- The software can be installed on any PC equipped with Windows 95 or above.

7.3 Application

The main objective for setting up the current Teaching Company Scheme is to develop a software system facilitating the MTRC engineer to analyze any problem pertaining to MTRC power supply system. Based on this software, one of the MTRC staff has successfully completed the studies on the following aspects [60]:

- Normal system operation
- Loss of a grid infeed substation
- Loss of east/west supplies
- Loss of a 132/33kV distribution transformer
- Loss of 33/11kV distribution transformer
- 33kV & 11kV circuit outages
- System operation under congestion & emergency modes
- In-feed substation spare capacity

7.4 Conclusion

The software runs on the Windows paltform which is the most flexible way of using and installing the software by the engineers. The efficiency of the software is justified by its high speed of power flow calculation and the graphical display for any modifications in the power network configuration including data editing, cable switching and busbar changeover with the 33kV changeover system of the MTRC. The software can calculate and display the current flow in the power networks under the cases of normal, fault and infeed failure. An AC Norton-equivalent model is developed to support studies of the system with voltage dependent loads. For fault analysis, users are allowed to choose any one bus as the faulted bus, the software will then calculate and display the fault current flow in the network immediately. The power system is represented by one-line diagram with distinct colors for normally seperated networks. The infeed points, busbars, cables, transformers, loads and circuit breakers are referred to by its own name and no manual numbering is needed. The software is incoporated with the MTR relay database. Each of the named circuit breaker is automatically connected to the databse. The operating sequences/time of the presetted relays can thus be simulated during fault analysis. The software provides a high degree of flexibility and user-friendliness in terms of software operation, modification, maintenance and future enhancement which is suitable for power system design and anlysis by the engineers in MTR Corporation.

7.5 Suggested Further Work

- **DC Power System and Train simulations** – At present, the traction load is retrieved from the sophisticated Multi-Train Simulator in MTRC, it is desirable to have a self-contained software for the entire AC/DC system. Goodman have given a good review of simulation models for railway systems [57] while Dr. Siu gave the latest work in applying OO concept on train movement, signaling and power network simulations [58, 59]. The DC loadflow algorithm presented in this thesis can be integrated with Dr. Siu's Train model. Thus a compenhensive AC-DC network and train simulation software is foreseeable.

- **Optimum loadflow** – Simulation of optimum loadflow can be added to calculate the reduction of total distribution loss inside the power system.

List of References

[1] Dr. C.J. Goodman and Mr. T. K. Ho, "Documentation for Multi-train Simulator, MTRC, H.K. version. Volume I – software description", Traction Research Group, School of Electronic and Electrical Engineering, The University of Birmingham, Nov. 1991.

[2] P. Lawton and F.J. Murphy, "Hong Kong Mass Transit Railway Power Supply System", IEE Proc.-C, 1986, Vol.133 No.7, pp. 462-469.

[3] R. Edgley, "Design for Safety for the Hong Kong Mass Transit Railway", IEE Power Engineering Journal, May 1989, pp. 129-140.

[4] Bose, A., "Digital Simulation of Power Systems", 1998 IEEE Region 10 International Conference on Global Connectivity in Energy, Computer, Communication and Control, 1998, Vol. 2, pp. 444 –447.

[5] Proceedings of the 20th International Conference on Power Industry Computer Applications Power Industry Computer Applications, 1997.

[6] Overbye, T.J., Klump, R.P. and Weber, J.D., "Development and Application of a Power System Simulation Environment", 1996., IEEE 39th Midwest Symposium on Circuits and Systems Volume: 3 , 1996 , Page(s): 1097 –1100.

[7] IEE Colloquium on "Simulation of Power Systems" (Digest No.221) Simulation of Power Systems, 1992.

[8] Foley, M.; Chen, Y., Bose, A., "A Modern Digital Simulation Laboratory for Power Systems", IEEE Computer Applications in Power Volume: 3 2 , April 1990 , Page(s): 16 –19.

[9] IEE Colloquium on "Power System Simulation" (Digest No.79) 1989.

[10] Chan, S., "Interactive Graphics Interface for Power System Network Analysis", IEEE Computer Applications in Power Volume: 3 1, Jan. 1990 , Page(s): 34 –38.

[11] Joong-Rin Shin; Wook-Hwa Lee; Dong-Hae Im, "A Windows-based Interactive and Graphic Package for the Education and Training of Power System Analysis and Operation, IEEE Transactions on Power Systems, Volume: 14, 4 , Nov. 1999 , Page(s): 1193 –1199.

[12] Yuan-Yih Hsu; Chien-Chuen Yang; Chung-Ching Su , "A Personal Computer Based Interactive Software for Power System Operation Education", IEEE Transactions on Power Systems", Volume: 7 4 , Nov. 1992 , Page(s): 1591 –1597.

[13] Daneshdoost, M., Shaat, R., "A PC Based Integrated Software for Power System Education", IEEE Transactions on Power Systems, Volume: 4 3 , Aug. 1989 , Page(s): 1285 –1292.

[14] David C. Yu, Shin-Tzo Chen, "A PC-Based Interactive Graphical Simulation and Analysis Package for a Power Plant Electrical auxiliary System", University of Wisconsin-Milwaukee & Wisconsin Electric Power Company, Milwaukee, IEEE Transactions on Power Systems, 1989, 112-118.

[15] Thomas J. Overbye, Caroline M. Marzinzik, , "A User-friendly Simulation Program for Teaching Power System Operations", University of Illinois, Urbana, IEEE Transactions on Power Systems, Vol. 10, No. 4, Nov 1995, 1725-1733.

[16] "Software Engineering", IEEE Spectrum, January 1994.

[17] Khan, E.H., Girgis, M.R., "The effect of OO life cycle on software project

management", International Conference on Engineering and Technology Management, 1996 , Conference Proceedings. Page(s): 233 –240.

[18] Khan, E.N., "Software project management: the effect of development methodologies and environment" Technology Management: the New International Language, 1991, Page(s): 585.

[19] Hughes, R.T., "The 'step wise' Planning Approach to Software Projects", IEE Colloquium on Project Management for Software Engineers, 1995 , Page(s): 3/1 -3/3.

[20] Rowen, R.B., "Software Project Management under Incomplete and Ambiguous Specifications", IEEE Transactions on Engineering Management, Volume: 37 1 , Feb. 1990 , Page(s): 10 –21.

[21] Jun Zhu, Lubkeman, D.L., "Object-oriented Development of Software Systems for Power System Simulations", IEEE Proceedings on Transmission and Distribution Conference, 1996, Page(s): 596 –601.

[22] Zhou, E.Z., "Object-oriented Programming, C++ and Power System Simulation", IEEE Transactions on Power Systems, Volume: 11 1, Feb. 1996 , Page(s): 206-215.

[23] Bosshart, P.; Bacher, R., "A domain Architecture for Solving Simultaneous Nonlinear Network Equations [power system simulation], The 20th International Conference on Power Industry Computer Applications, 1997 , Page(s): 142 –148.

[24] Flinn, D.G, Dugan, R.C., "A database for Diverse Power System Simulation Applications", Proceedings on Power Industry Computer Application, 1991,

Page(s): 186 –192.

[25] Li, S., Shahidehpour, S.M., “An Object-oriented Power System Graphics Package for Personal Computer Environment”, IEEE Transactions on Power Systems, Volume: 8 3 , Aug. 1993, Page(s): 1054-1060.

[26] Flinn, D.G, Dugan, R.C., “A Database for Diverse Power System Simulation Applications”, IEEE Transactions on Power Systems, Volume: 7 2 , May 1992, Page(s): 784 –790.

[27] Marti, J.R.; Linares, L.R.; Calvino, J.; Dommel, H.W.; Lin, J., “OVNI: an Object Approach to Real-time Power System Simulators”, 1998 International Conference on Power System Technology, POWERCON '98. Volume: 2 , 1998, Page(s): 977 -981.

[28] Zhu, J.; Jossman, P., “Application of Design Patterns for Object-Oriented Modeling of Power Systems”, IEEE Transactions on Power Systems, Volume: 14 2 , May 1999, Page(s): 532-537.

[29] Manzoni, A.; e Silva, A.S.; Decker, I.C., “Power Systems Dynamics Simulation using Object-oriented Programming”, IEEE Transactions on Power Systems, Volume: 14 1 , Feb. 1999 , Page(s): 249-255.

[30] Jun Zhu, Operation Technology Inc., California, “Object-Oriented Development of Software Systems for Power System Simulation”, IEEE Transactions on Power Systems, 1996, 596-601,

[31] Elegy, D., Gilbert, R., Battelle Memorial Institute, Ricland, “An Object Model for Embedded System Development”, Northcon/93 Conference Record, 1993,

Page(s): 234 -238.

[32] S. Li, S.M. Shahidehpour, Illinois Institute of Technology, Chicago, “An Object Oriented Power System Graphics Package for Personal Computer Environment”, IEEE Transactions on Power Systems, Vol. 8, No. 3, Aug 1993, 1054-1060.

[33] Yehsakul, P.D.; Dabbaghchi, I., “A Topology-based Algorithm for Tracking Network Connectivity”, IEEE Transactions on Power Systems, Volume: 10 1 , Feb. 1995 , Page(s): 339-346.

[34] J.B. Ward and H. W. Hale, “Digital Computer Solution of Power-flow Problems”, AIEE Trans. (Power App. Syst.), vol. 75, June 1956, pp. 398-404.

[35] A.F. Glimn and G. W. Stagg, “Automatic Calculation of Load Flows, AIEE Trans. (Power App. Syst.), vol. 76, Oct. 1957, pp. 817-828.

[36] R. J. Brown and W. F. Tinney, “Digital Solutions for Large Power Networks, AIEE Trans. (Power App. Syst.), vol. 76, June 1957, pp. 347-355.

[37] A. M. Sasson and F. J. Jaimes, “Digital Methods Applied to Power Flow Studies", IEEE Trans. Power App. Syst., vol. PAS-86, pp.860-867, July 1967.

[38] F. J. Hubert and D. R. Hayes, “A Rapid Digital Computer Solution for Power System Network Load-flow”, IEEE Trans. Power App. Syst., vol. PAS-90, pp. 934-940, May/June 1971.

[39] P. P. Gupta and M. W. Humphrey Davies, “Digital Computers in Power System Analysis”, Proc. Inst. Elec. Eng., vol. 108A, pp. 383-404, Jan. 1961.

[40] J. E. Van Ness, “Iteration Methods for Digital Load Flow Studies”, AIEE Trans. (Power App. Syst.), vol. 78, pp. 583-588, Aug. 1959.

[41] M. R. G. Al-Shakarchi, "Nodal Iterative Load Flow", M.Sc. thesis, University of Manchester Institute of Science and Technology, Manchester, U. K., 1973.

[42] W. F. Tinney and J. W. Walker, "Direct Solutions of Sparse Network Equations by Optimally Ordered Triangular Factorization", Proc. IEEE, vol. 55, pp. 1801-1809, Nov. 1967.

[43] Stott, B, "Review of Load-Flow Calculation Methods", Proc. IEEE, July, 1974, 916.

[44] "A.C. Power System for TWL, KTL & ISL (Including EHC Extension)", MTR Corporation E&MS, drawing number 0D4/201/00001/005

[45] "Borland Delphi for Windows", User's Guide, 1995 Borland International.

[46] Cantlu, Marco., "Mastering Delphi", San Francisco : Sybex, c1996.

[47] Elegy, D., Gilbert, R., "An Object Model for Embedded System Development", Northcon/93. Conference Record , 1993 , Page(s): 234 –238.

[48] Tzeng Y.S., Wu R.N. and Chen N., 1995, "Unified AC/DC power flow for system simulation in DC electrified Transit Railways", IEE Proc, Electr. Power. Appl. Vol 142, No.6, pp345-354.

[49] Zollenkopf K., "Bi-factorisation algorithm and programming techniques", Conference on Large Sets of linear equations, 1970, Oxford.

[50] Cornel H.C.J., "QENS: An enhanced version of the electric network simulator program", MSc thesis, Queen's University, Canada, Oct. 1987.

[51] Scott B. and Alsac. O., "Fast Decoupled Load Flow", IEEE Trans. Vol. PAS-93, pp 859-969.

[52] Cai Y., Irving M.R. and Case S.H., "Iterative techniques for the solution of complex DC-rail-traction systems including regenerative braking", IEE Proc. C., vol 142, Sept. 1995, pp445-452.

[53] William W. Price, et al., "Load Modelling for Power Flow and Transient Stability Computer Studies", IEEE Trans. on Power Systems, Vol 3, No.1, pp.180-187, February, 1988.

[54] El-Hawary, M.E. and Dias, L.G., "Incorporation of Load Models in Load Flow Studies: Form of Model Effects,", IEE Proc. vol 134, Pt.C, pp.27-30, January 1987.

[55] J. Arrillaga and C.P. Arnold, "Computer Analysis of Power Systems", 1995, John Wiley & Sons Ltd.

[56] "33kV Changeover & Restoration (basic)", Documentation No. T04E11, E&MS, MTRC, 1996.

[57] Goodman, C.J. Siu, L.K., Ho, T.K., "A Review of Simulation Models for Railway Systems", Proceedings of the IEE International Conference on Developments in Mass Transit Systems. Pub. No. 453. London, U.K. April 1996.

[58] Siu, L.K., Goodman, C.J., "An Object-Oriented Concept for Simulation of Railway Signaling and Train Movements", Proceedings of the 3rd International Conference on Computer Aided Design, Manufacture and Operation in the Railway and other Advanced Mass Transit Systems CompRail 92. Washington D.C. U.S.A. Aug. 1992, pp545-556.

[59] Siu, L.K., Goodman, C.J., "An Object-Oriented Power Network Simulator for

Multi-train Simulations", Proceedings of the 4th International Conference on Computer Aided Design, Manufacture and Operation in the Railway and other Advanced Mass Transit Systems CompRail 94. Madrid, Spain. Sept. 1994, pp483-490.

[60] Chan Yin Chin, "Microcmputer Based Power Distribution System for MTR Corporation", Part-time Degree Project Report, January 2000.