A Graphical Interface for the BPA Load Flow and Transient Stability Programs

A Thesis Presented to

The Department of Electrical and Computer Engineering

The Faculty of Engineering

The University of Manitoba

In partial fulfilment of the requirements
for the degree of Master of Science
in Electrical and Computer Engineering

by Hong Jiang

September 1997

0-612-23355-3

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION PAGE

A GRAPHICAL INTERFACE FOR THE BPA LOAD FLOW AND TRANSIENT STABILITY PROGRAMS

BY

HONG JIANG

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University

of Manitoba in partial fulfillment of the requirements of the degree

of

MASTER OF SCIENCE

Abstract

In the power system industry, the load flow (or power flow) program is a key tool for power system network planning, design and operating engineers while a transient stability program provides an effective tool for performing the dynamic simulation of a power system when disturbed from its steady-state condition under various perturbations.

Power system load flow and transient stability programs have been around world for a long time, but it takes considerable effort and time to be proficient in their use. In addition, data entry for these programs is through the use of data files and the "look and feel" is more like that of a mainframe-era program, with associated entry of data into files and batch-oriented execution[1].

To describe a power system network (circuit) for simulation and analysis, both load flow and transient stability programs require that the system network information be mapped into data files. Traditionally, this mapping procedure is usually done by hand-coding, line by line. For some analysis programs such as the BPA Load Flow and Transient Stability Programs, data entry is strict - requiring data to be placed in individual columns. This mapping procedure is tedious, time-consuming and prone to error.

Obviously, a graphical interface for both load flow and transient stability programs would be a much more efficient means for data entry and result visualization. Using a graphical interface, power system engineers do not need to hand-code the input data files as

previously required. Instead, now, they are able to define the power system network by using an one-line diagram and the computer takes care of the rest of the work of data mapping from the one-line diagram into data files. Also, the visualization of simulation results provides an easy way for output analysis and the BPA Package Manager program developed here gives users more freedom from typing and remembering various commands and data file names in each procedure.

In addition, it is important to note that the methods used in this thesis could be applied in other load flow and transient stability simulation programs or even other fields of power system industry easily.

Acknowledgements

The author wishes to express his sincere and deep gratitude to his advisor Professor A.M.Gole for his valuable guidance and encouragement during the course of study and investigation.

The author would also like to thank Dennis Woodford, director of Manitoba HVDC Research Centre for his suggestion and Dr. Alan Wang at Manitoba HVDC Research Centre for his helpful guidance.

Research grant (E4.202) support from the Manitoba HVDC Research Centre which made this work possible is greatly appreciated. The excellent working environment provided by Manitoba HVDC Research Centre are gratefully acknowledged.

Contents

List of Figures

List of Tables

List of Lists

Chapter 1:

Introduction

1.1 Overview

Today's power system engineers face new and increasingly complex problems associated with planning and operating power system. This myriad of new problems place additional burdens on engineers, at a time when many utilities across North America are reducing staff levels. This has increased the demand for faster, easier to use, more sophisticated tools for planning and operating power system. Advances in computer hardware have increased expectations of software users and many new software techniques and tools are migrating to the power system industry to improve the existing software environment[2]. A Graphical User Interface (GUI) is one such technique that uses the latest in man-machine interface technique. On the other hand, the BPA Load Flow and Transient Stability Programs represent proven software with two decades of history.

In power system analysis, the load flow study is of paramount importance. The study reveals the electrical performance and load flows (real and reactive) for specified conditions when the power system is operating under steady state. The load flow study also provides information about the line and transformer loads (as well as losses) throughout the system and voltage (magnitude and phase) at different points in the system for evaluation and regulation of the performance of the power system under prior known conditions.

Further alternative plans for future expansion to meet new load demands can be analysed.

The dynamic nature of load growth, load distribution and generation patterns makes the network planning, design and operating more difficult. To forecast far into the future, any design must look at the past and present. This makes the load flow program a key tool for the network design engineer.

The Bonneville Power Administration (BPA) Load Flow program is a collection of FORTRAN-coded computer programs permitting analysis of steady-state operation of an electric power system network. This program is intended to help electric power system planning and design engineers investigate a given electric power system network with respect to bus voltage distribution, line real and reactive load flows, line overloads, system reactive requirements, area interchange control, dropping, line outages, etc.

The transient stability of an electric power system is referred as its capability to return to the original equilibrium position on the occurrence of a disturbance or to another equilibrium state which is generally in the proximity of the initial equilibrium point.

The BPA Transient Stability Program (TSP) was developed in accordance with the Western Systems Coordinating Council (WSCC) specification (see Appendix A) and provides an effective tool for performing the dynamic simulation of the power system when disturbed from its steady-state condition under various perturbations.This program is designed to run in conjunction with the BPA Load Flow Program.

1.2 Benefits

In describing a power system network for simulation and analysis on the BPA Load Flow and Transient Stability Programs, all the network information about the system must be hand-coded in input data files which are required for running the BPA Load Flow and Transient Stability Programs respectively. To power system engineers, the entire work of simulation and output analysis is usually done using the following procedures: draw out the system diagram on paper, number or name all nodes (buses), then enter all necessary information in strict format for the variety of system components, run BPA Load Flow and Transient Stability Programs, at last, analyse the output results.

It would be preferable if the data mapping procedure was transparent to users. The users should merely have to construct the system network one-line diagram on computer screen as they would do on paper before, and computer program should extract the necessary system network information from the one-line diagram and generate input data files required by the BPA Load Flow and Transient Stability Programs. This is what the graphical interface developed in this thesis achieves.

The interface is designed to be totally graphical, with no intermediate typing of commands and data entry is through pop-up menus for all system components, there is a likelihood of making a data entry error as often happens with non-graphical user interface load flow and stability programs[1].

A decrease in frequency of keyboard use should also decrease the frequency of typographical errors. Therefore, the keyboard is used only for necessary information entries, such as for entering values of node voltage, string name of bus, branch and transformer parameters and so on. The rest of one-line diagram construction work is carried out using only the mouse.

By developing a graphical representation of components of the power systems on a computer screen, the user may monitor his or her input to make sure that the proper system circuit and information is being described well.

Along with the ability to enter information easily and generate the BPA Load Flow and Transient Stability Programs input data files accurately, the interface also has the ability to save the data in a graphical format in a so called "DRAFT" file. This feature allows the users to reload a previous diagram or exchange system data with another user using this diagrammatic format without any difficulty because what is exchanged is the draft file (one-line diagram) rather than input data file. Thus exchanging circuit information between users is reduced to a matter of exchanging a graphical file and leads to the possibility of co-operative projects between users [1]. However, if the users want to makes modification in the input data files directly, it would be very difficult through hand-coding, unless the users have a hand-drafted system circuit diagram on hand and are very familiar with each component data format of the BPA Load Flow and Transient Stability Programs. With the graphical interface, both the input data files and graphical file are generated, the former are for running BPA Load Flow and Transient Stability Programs, the latter for reloading or exchanging with others later.

Also, the graphical interface makes it possible that the results from running BPA Load Flow Program can be shown on the system circuit diagram visually rather than in the output data file. In the past, simulation results meant stacks of output consisting of data indicating voltage, current, phase angle, real and reactive power and so on. Now, using the graphical interface, that situation has been changed totally. Power system network planning and design engineers use diagrams or graphs to show the load flow results or transient and fault processes evolved dynamically on the screen.

1.3 Objective

The main objective of this thesis is to provide users with a graphical interface to build power system network in (one-line) diagram form and have the computer automatically map input data files required for running both BPA Load Flow and Transient Stability program from the one-line diagram.

Another objective in this thesis is to make the simulation results from the BPA Load Flow Program visualizable, that is, displaying or printing the simulation result directly on the existing system one-line diagram. One added benefit of graphical entry to the power system engineers is automatic documentation.

Also, a BPA Package Manager program is developed to help users organize all the commands and relevant files that are involved when the user constructs the system circuit diagram and simulate on it.

The above approach closely resembles that followed for the PSCAD/EMTDC electromagnetic transient simulation package[3].

It is important to note that the approach used in this thesis demonstrates a general approach for graphical interface methods to be exploited in analogous fields, i.e., in other load flow and stability programs or even in other fields of power system analysis.

Chapter 2:

BPA Load Flow and Transient Stability Programs

2.1 BPA Load Flow Program

Electric power system network planning and design encompasses the following five tasks:

(1) Determination of load centres (points) and generation patterns as well as sizes of loads and generation

(2) Determination of available transmission corridors (rights of way) and assessment of the capacity of these corridors to accommodate transmission lines.

(3) Evaluation of existing or planned networks with regard to adequate power-carrying capacity, voltage regulation, reliability of service and operating economics.

(4) Determination of size and routing to new transmission lines and size and location of terminal equipment for achieving efficient and economical reinforcements when needed.

(5) Evaluation of proposed reinforcements with respect to the ability to withstand transient disturbances, reliability of overall service, economics, impact on regional economy, environment, energy conservation and operational constraints such as construction lead times, coordination of various facility ownership interests, flexibility for future growth and compatibility with other long-range plans.

Load flow program is a key tool for all above tasks. A load flow study is the determination of the voltage, current, real or reactive power, and power factor at various points (nodes) in an electric power system network under existing or contemplated conditions of normal operation. Load flow studies are essential in analysing and planning the future development of a power system because operation of a power system greatly depends on how much we know the effects of interconnections with other power systems, of new loads or change of loads, new power plants, and new transmission lines before they are installed.

A typical load flow program is capable of handling systems of over 2000 buses, 3000 lines, and 500 transformers. Of course, programs can be expanded to even greater size if the available computer has enough memory and higher speeds.

The solution of the load flow problem follows an iterative process by assigning estimated values to the unknown bus voltage and calculating a new value for each bus voltage from the estimated values at the other buses and real and reactive power specified. A new set of values for voltage is thus obtained for each bus and used to calculate another set of bus voltage. Each calculation of a new set of voltage is called an iteration. The iterative process is repeated until the changes at each bus are less than a specified minimum value.

Several different methods can be used for selecting the iteration. The Gauss-Seidel method is one of the most common methods used in load flow studies. Another common method is Newton-Raphson method which is very suitable for load flow studies on large systems. Each of these two methods has its own particular strength. Most programs begin with one Gauss-Seidel iteration to obtain a good initial value for the first Newton-Raphson iteration. This voltage is then used to calculate P (real power) at every bus except the swing bus (the bus at which only voltage magnitude and phase are specified) and Q (reactive power) at those buses where reactive power is specified. The differences between specified and calculated values are used to determine the correction of bus voltage. The process is then repeated until the calculated values of P and Q or |V| at every bus differ from the specified values by less than the chosen precision index.

The BPA Load Flow program was developed by the Bonneville Power Administration (BPA) in accordance with the Western Systems Coordinating Council (WSCC) specifications (see Appendix A) and was originally written to be executed on the VAX 11/780. It was written in FORTRAN language about two decades ago.

In order to make more efficient use of computer memory space and computation time, the BPA Load Flow Program uses advanced techniques of large-system analysis including the Newton-Raphson method of solution of algebraic equations and sparse-matrix computation techniques. The program has a great deal of flexibility which provides an abundance of available features. One of the many significant features is its ability to investigate small regions within a large network. Since its first release, there are several users around the world who are using the BPA Load Flow and Transient Stability programs including Teshmont Consultants in Winnipeg, Canada. One additional favourable feature is that because it is developed by the Bonneville Power Administration (BPA) which is a United States government agency, the programs are in the public domain and is available free of charge.

2.2 BPA Transient Stability Program

The stability of an electric power system is referred as its capability to return to the original equilibrium position on the occurrence of a disturbance or to another equilibrium state which is generally in the proximity of the initial equilibrium point. The disturbance may be small or large and the system may become unstable in either event depending on the operating condition. The study of power system stability in the circumstance of small disturbance constitutes what is known as "static stability" analysis in the literature. The mathematical model for such a static stability study is a set of linear time-invariant differential equations. When the disturbance is large, the non-linearity inherent in the power system can no longer be ignored and the stability study under such large disturbance constitutes what is known as "transient stability" analysis. The mathematical model for such a study is a set of non-linear differential equations coupled with a set of non-linear algebraic equations. The dimension of the mathematical model for both static and transient stability can be very large, even for moderately sized systems.

Like the BPA Load Flow Program, the BPA Transient Stability Program (TSP) was also written in FORTRAN language and run on VAX 11/780 originally. It too was developed using WSCC specifications (see Appendix A). This program provides an effective tool for performing the dynamic simulation of a power system when disturbed from its steady-state condition under various perturbation. There are basically two methods of solution used in the BPA Transient Stability Program. All differential equations are linear and solved by the trapezoidal rule of integration. The network equations are solved iteratively using the triangularized matrix. Both of these methods are described in literature [4][5].

In most transient stability programs, the electrical power system network is solved in the phase domain using algebraic (complex) equations, with the mechanical rotor dynamic equations being represented by non-linear differential equations.

The BPA Transient Stability program is designed to run in conjunction with the BPA Load Flow program, but the load flow program must be run before the transient stability program. This is because transient stability program begins with a disturbance due to a fault or switching and needs initial values for power and other system quantities supplied from the load flow program.

2.3 Data for the BPA Load Flow and Transient Stability Programs

To begin the study, the system diagram is usually drawn out on the paper by using various visual symbols, all nodes(buses) are numbered or named, and each branch, from node to node, is described. Errors often occur during the mapping system diagram into data files required by the BPA Load Flow and Transient Stability Programs. In addition, typographical errors may be found.

Both the BPA Load Flow and the Transient Stability Programs require several files to describe the power system network being studied. The BPA Load Flow program needs a program control file, a base data file, a non-formatted file from a previous load flow case and, optionally, a change file. The BPA Transient Stability program needs a data file which describes the disturbance(s) being simulated. Before using the load flow program, users need to gather related data of network and controls. After that, the first step in a simulation study is to set up one or more of these files in a form suitable to the BPA Load Flow and Transient Stability program. Then, the load flow program is run. The load flow program first reads the data from these files, checks for data consistency and integrity, after which it computes results and generates the output file. After analysis of the results, users may want to insert or remove a bus, line or other component in the input file for subsequent runs. Although the BPA Load Flow offers users with the program control language to handle modifications to the system, the procedure is significantly user-unfriendly. Users often forget the original model after several iterations of changes.

The best way to demonstrate the preparation of the data file is to work through an example. The simple nine buses network is shown in Figure 2.1. The corresponding input data files for running the BPA Load Flow and Transient Stability Programs are listed below.

```
(POWERFLOW,CASEID=95KD,PROJECT=HN1995)
/ NEW_BASE,FILE=95KD.bse \
/ P_INPUT_LIST,FULL  \
/ P_INPUT_LIST,NONE  \
/ P_OUTPUT_LIST,FULL \
/ P_OUTPUT_LIST,NONE \
/ AI_LIST=NONE    \
/ OI_LIST=NONE    \
/ RPT_SORT=ZONE   \
/ AI_CONTROL=OFF \
/ P_ANALYSIS_REPORT,LEVEL=4 \
BS BLKBUS1     16.5 1   .0    .0  .0  .0250.71.60100.0-100.1040  .0BUS1     16.5  0
BQ BLKBUS2     18.0 1   .0    .0  .0  .0165.163.0100.0-100.1025   0BUS2     18.0  0
BQ BLKBUS3     13.8 1   .0    .0  .0  .0110. 85.0 60.0-60.01025   0BUS3     13.8  0
B  BLKBUS4     230. 1   .0    .0  .0  .0   .0    .0    .0   .01.02  .0BUS4     230.  0
B  BLKBUS5     230. 1125.0 50.0   .0  .0   .0    .0    .0   .0.996  .0BUS5     230.  0
B  BLKBUS6     230. 1 90.0 30.0   .0  .0   .0    .0    .0   .01013  .0BUS6     230.  0
B  BLKBUS7     230. 1   .0    .0  .0  .0   .0    .0    .0   .01.02  .0BUS7     230.  0
B  BLKBUS8     230. 1100.0 35.0   .0  .0   .0    .0    .0   .01016  .0BUS8     230.  0
B  BLKBUS9     230. 1   .0    .0  .0  .0   .0    .0    .0   .01.02  .0BUS9     230.  0
L  BLKBUS7     230.0BUS5      230.10  .00.0320 .1610 .0     .1530    .0   BLANK  0  0   .0  .0
L  BLKBUS6     230.0BUS9      230.10  .00.0390 .170  .0     .1790    .0   BLANK  0  0   .0  .0
L  BLKBUS4     230.0BUS6      230.10  .00.0170 .0920 .0     .0790    .0   BLANK  0  0   .0  .0
L  BLKBUS4     230.0BUS5      230.B0  .00.010  .0850 .0     .0880    .0   BLANK  0  0   .0  .0
```

```
L  BLKBUS8    230.0BUS9    230.10  .00.01190.10080.0     .10450  .0   BLANK  0  0  .0  .0
L  BLKBUS7    230.0BUS8    230.10  .00.00850.0720 .0     .07450  .0   BLANK  0  0  .0  .0
T  BLKBUS2    18.00BUS7    230.B0  .00     .0.06250    .0     .0 18.0230.0   0  0  .0  .0  .0
T  BLKBUS1    16.50BUS4    230.B0  .00     .0.05760    .0     .016.50230.0   0  0  .0  .0  .0
T  BLKBUS3    13.80BUS9    230.B0  .00     .0.05860    .0     .013.80230.0   0  0  .0  .0  .0
( END )
```

List 2.1 Input data file for BPA Load Flow Program

```
CASE 95KD              1
LS -BUS7    230. -BUS5    230. 1   -2     6.0     .0     .0     .0
LS  BUS7    230.  BUS5    230. 1    2     1.0     .0     .0     .0
LS  BUS7    230.  BUS5    230. 1   -2      .0     .0     .0     .0
M  BUS1     16.51100.0 1.0    1  H BLK    .0    .0   .0   .0
M  BUS3     13.83100.0 .85    1  S BLK    .0    .0   .0   .0
M  BUS2     18.02100.0 .85    1  S BLK    .0    .0   .0   .0
MP BUS1     16.51  2364100100100.  .0.0608.0969.1460.09698.96 .0.0336    .0  .0 .0
MP BUS3     13.83 301.0100100100.  .0.1813 .2501.3131.2585.89.60.0742    .0  .0 .0
MP BUS2     18.02 640.0100100100.  .0.1198.1969.8958.8645 6.0.54.0521    .0  .0 .0
FF   .0 2.0 300.0 2.0  15  .010  25  .10 8. 1.0        60 0      0 0 0 0 0 0 0 0 0
90
MH  4    .0        1.30          .80        60.0  60.0   500   0   .0   .0
BH        1        .0      .0 2        .0      .0
B  BUS8     230.  3  3            0
B  BUS5     230.  3  3            0
B  BUS6     230.  3  3            0
B  BUS1     16.5  3  3            0
```

List 2.2 Input Data file for BPA Transient Stability Program

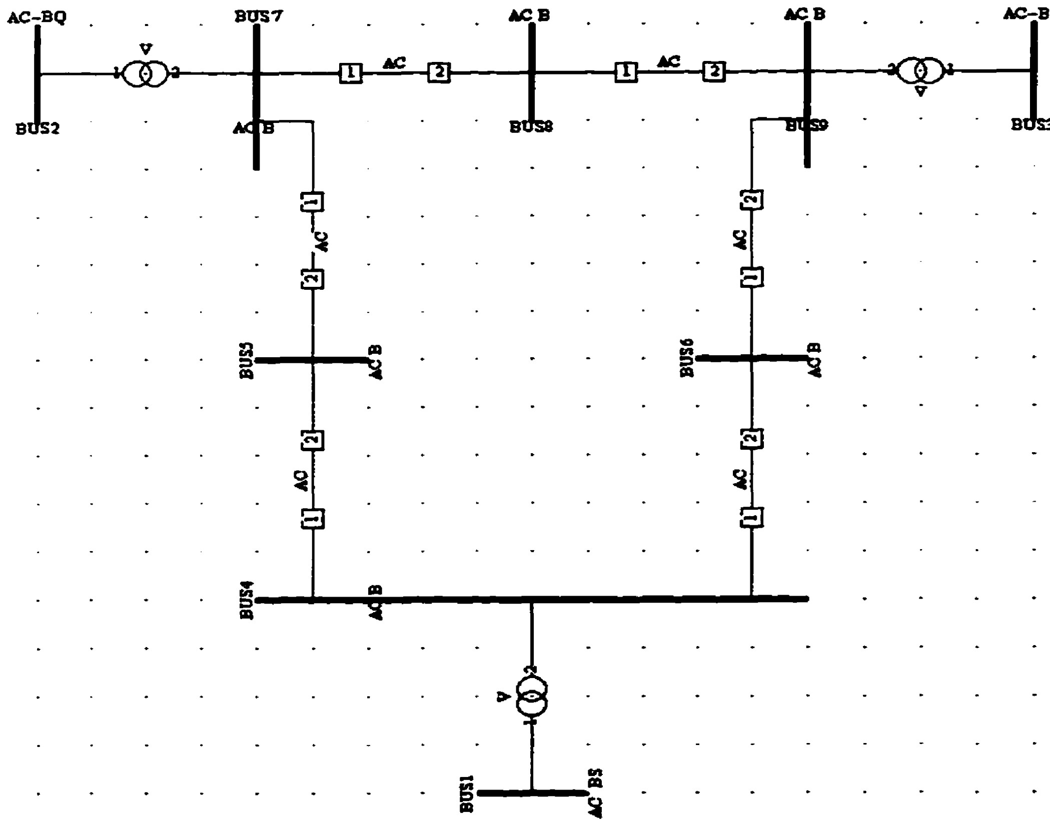


Figure 2.1 Classical Nine Buses Network Diagram

An electric power system is described via input data files. All network components such as bus, ac/dc transmission lines, transformers, ac machines data are entered into the input data file for the load flow and transient stability programs to read. Each record has a unique format which is described in pertinent sections. Figure 2.2 applies to B-blank bus format. Other types of record are similar and described in [8].

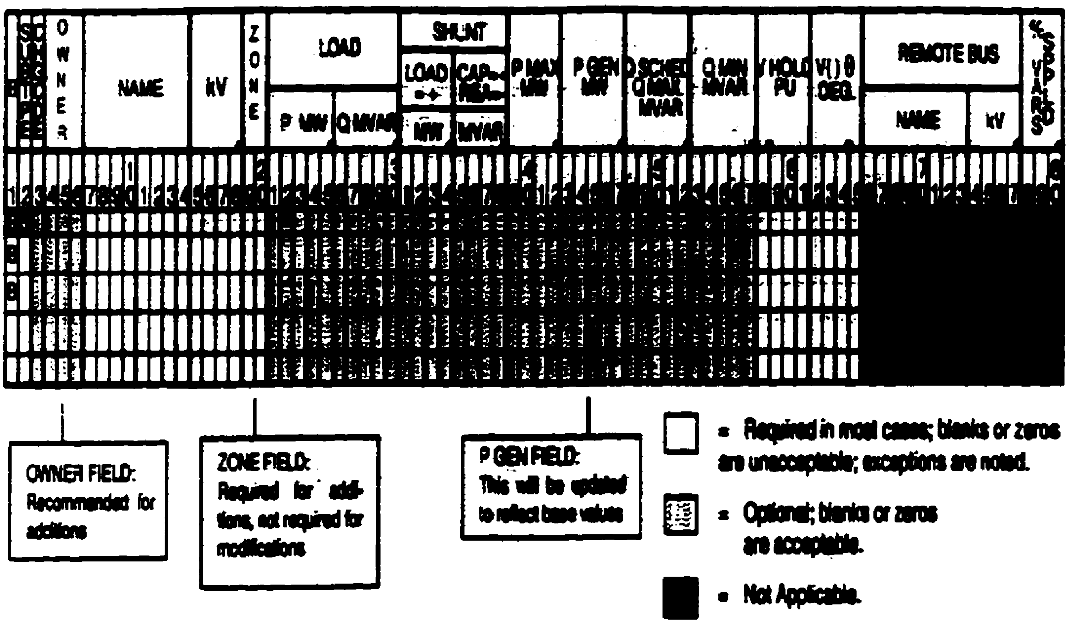


Figure 2.2 B-blank Subtype Format

Column	ID Field	Format	Description
1-2	yes	A2	BS — System swing or slack bus
3	no	A1	Change code
4-6	no	A3	Ownership
7-14	yes	A8	Bus name
15-18	yes	F4.0	Base kV
19-20	no	A2	Zone
21-25	no	F5.0	Load MW
26-30	no	F5.0	Load MVAR
31-34	no	F4.0	Shunt Admittance Load in MW at base kV
35-38	no	F4.0	Shunt Admittance in MVAR
39-42	no	F4.0	P MAX
43-47	no	F5.0	P GEN
48-52	no	F5.0	Q MAX
53-57	no	F5.0	Q MIN
58-61	no	F4.3	V HOLD

Table 2.1 Column Description for B Bus Data

From the description of ac bus above, it can be seen that each character in a column of the record has been assigned special meaning. Any mis-placement of data in the record will lead to a misinterpretation by the program, and will result in the erroneous results. So, while users enter data into the input data file, they must be very careful. In addition, when data is put in the wrong position, it is very difficult for the user to find out where the error is. When the system is very large (the usual case), the input data file becomes very large and complex. We can imagine that to find out and correct error(s) will cost users a lot of time. The user could have spent a lot of time on it instead of concentrating on the simulation exercise and the corresponding analysis and design. Thus there appears to be a strong demand for user-friendly simulation tools.

A convenient way to describe the system circuit for simulation exercise and analysis is to use a graphical representation to construct the system circuit drawing visually on the computer screen using a Graphical User Interface (GUI). In the GUI environment, the user, especially the beginner, does not have to master every detail of the information provided in user's manual prior or starting productive work with the BPA Load Flow and Transient Stability Programs.

In the approach developed by the author, end users assemble all the components of a power system circuit by dragging and dropping (via a mouse) visual representations of the components onto the workspace of the monitor screen from the components library and putting the components together. The users can assemble various buses, ac lines, dc lines, transformers, machine, governor, exciter components in ones system diagram and then compile the system without quitting the GUI screen. The computer program will automatically map the visual graph of the system circuit into the data files required by the load flow and stability simulation programs. As this work liberates users from the drudgery of model building, they are able to concentrate the bulk of their efforts on the simulation exercise and the corresponding analysis and design.

Chapter 3

PSCAD/EMTDC as the Development Environment

EMTDC is a power system electromagnetic transients simulation program. PSCAD is a graphic user interface developed for EMTDC. It has greatly simplified the tasks required to set-up and analyse the results of a simulation. EMTDC merely becomes the engine doing all the calculation and PSCAD aids the user to build model and analyse the results of the simulation.

In this chapter we will introduce PSCAD/EMTDC and its attractive features. We will examine the PSCAD/EMTDC structure and adapt them to supply a graphical interface for the BPA Load Flow and Transient Stability programs.

3.1 Overview

An electric power system is a very large and complex network, requiring many dimension of studies. Correspondingly, there exist many simulation tools for engineers to analyse and study these problems. Usually, these simulations are classified into four ranges:

(1) Load flow study

(2) Electrical network or steady state electromechanical transient study

(3) Transient stability study

(4) A detailed electromagnetic transient simulation study.

Each kind of simulation has its individual feature, application scope and limitation. According to the practical issues, engineers choose suitable tools for simulation and analysis.

Many problems of a power system can be adequately studied on a good load flow and stability program. Transient simulation using the extreme detail available in electromagnetic transients simulation programs should not replace the stability study. Generally speaking, if a problem in the power system can be studied and solved on the load flow and transient stability program, then so be it. Only when it becomes obvious that such a study is inadequate, or requires confirmation from a separate method of study should the electromagnetic transients solution be attempted. For example, a self-excitation study of an ac machine can be done on a good stability program. However, if transformer saturation is a factor, then harmonic effects may become significant particularly if harmonic resonance are possible. In these situation, a better solution should be with an electromagnetic transients simulation study.

EMTDC™ originally developed at the Manitoba Hydro with considerable University of Manitoba input is a power system electromagnetic transients simulation program designed to aid the engineer in the study of complex power systems and is fast becoming the worldwide industry standard. The strength of EMTDC lies with its ability to model virtually any conceivable system. Users are able to make use of an extensive supplied library of power system component models for their transients simulation study. Users may also choose to develop their own unique models for use with EMTDC if it is found that the existing components are not suitable to their study.

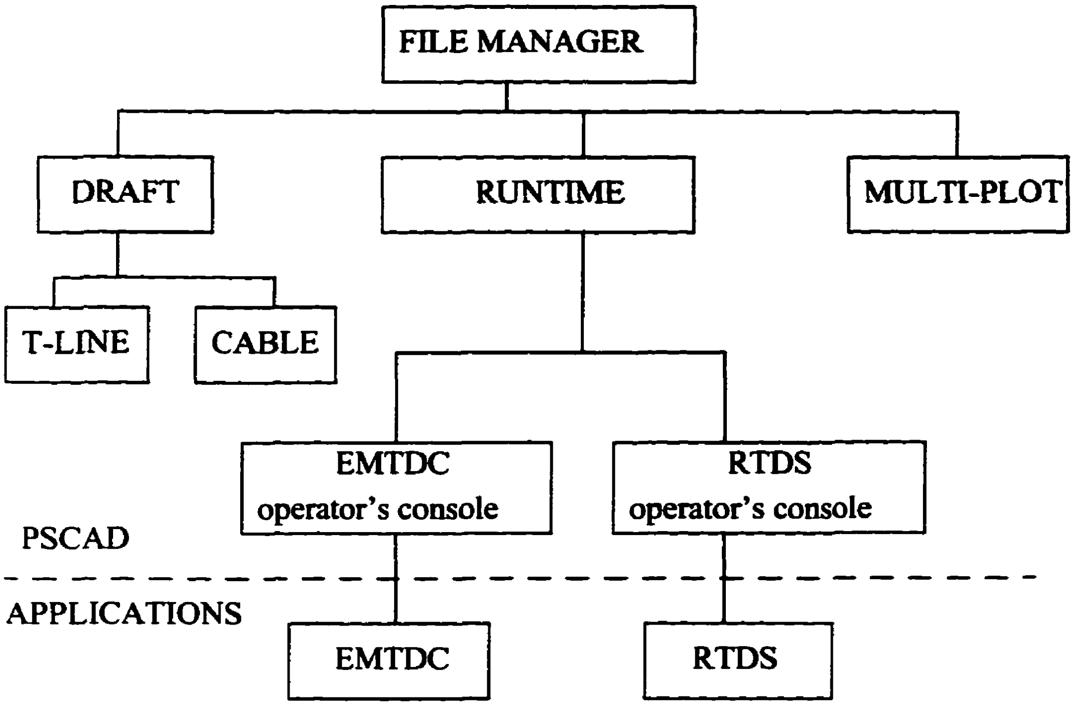


Figure 3.1 Interface between PSCAD and APPLICATION

PSCAD™ developed by the Manitoba HVDC Research Centre is a powerful graphical user interface which substantially increases productivity in undertaking simulation of electromagnetic transients studies of electric power systems. With the PSCAD family of software tools the power system engineers are able to use the full power of modern computer workstations to run the EMTDC cases. PSCAD also serves as the front end for the RTDS™ (Real-Time Digital Simulator). The development of PSCAD, a graphical user interface for EMTDC, has greatly simplified the tasks required to set-up, run, and analyse the result of a simulation with EMTDC. EMTDC merely becomes the engine doing all the calculations. The interface between PSCAD and APPLICATIONS (EMTDC, RTDS) are shown in Figure 3.1.

There is a requirement for incorporating the power system stability program into PSCAD. The objective is to represent a portion of the electric power system critical for design and study, as an electromagnetic transient model within the stability program which allows for representations of a large network with reduced details.

The PSCAD front end for both EMTDC and RTDS has made it a powerful simulation tool. This is because unlike the traditional method for entering data via a data file, PSCAD/ EMTDC uses a totally graphical front-end, which is a collection of well-integrated modules (DRAFT, RUNTIME, and MULTIPLOT) and forms a complete graphical environment for power systems simulation. Also, via PSCAD's RUNTIME module, simulation cases may be run in an interactive manner, giving the engineer a system operator type interface, complete with slider, push-button, dials, meters and an oscilloscope type plot.

Another advantage of a seamless connection between the load flow/stability and EMTDC modules is that initial conditions may be first calculated with the phase solution inherent in the load flow engine of a stability program. Initializing the network and control system to steady-state conditions is a major effort in many simulation programs. There is no initialization process evident in EMTDC. The problem of initialization is solved by running the model to a steady state condition, and freezing all states and variables in the snapshot file which is really a kind of data file with everything initialized. However, it may be considered a disadvantage to do this because if the model contains instabilities, it may never reach the steady state, or may require excessive computer time to do so. EMTDC allows initial conditions to be entered, but at present they have to be externally computed. This has always been a moot point with EMTDC.

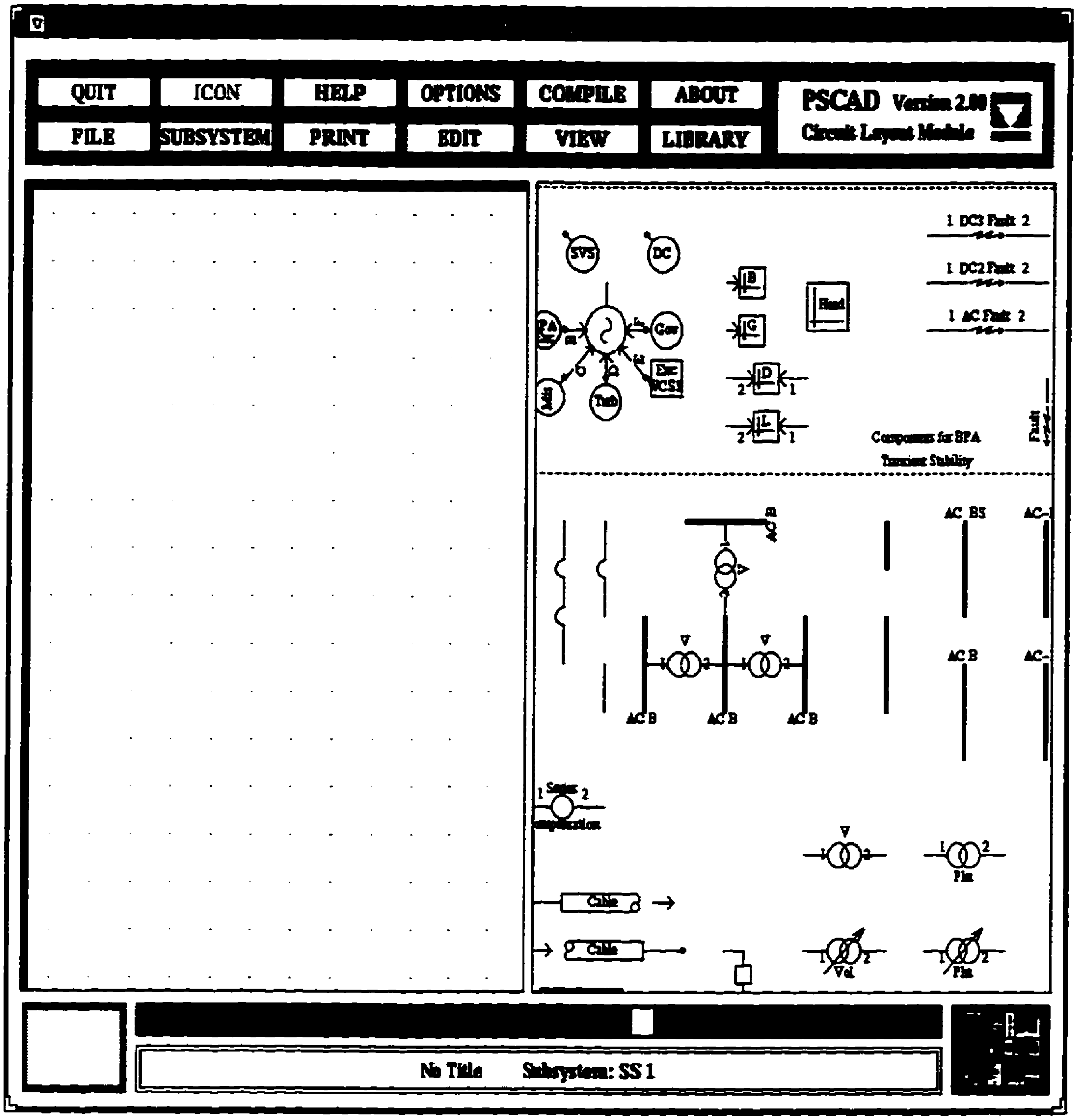
The above discussion entails a significant amount of work. However, this thesis only covers a portion of the whole exercise, in particular adapting the DRAFT module in PSCAD to generate BPA input data files and render the output results, and the combination of an additional GUI component for run management

3.2 Structure of PSCAD

PSCAD's general structure is illustrated in Figure 3.1. In the material to follow, we will go in detail to see each part of PSCAD.

DRAFT:

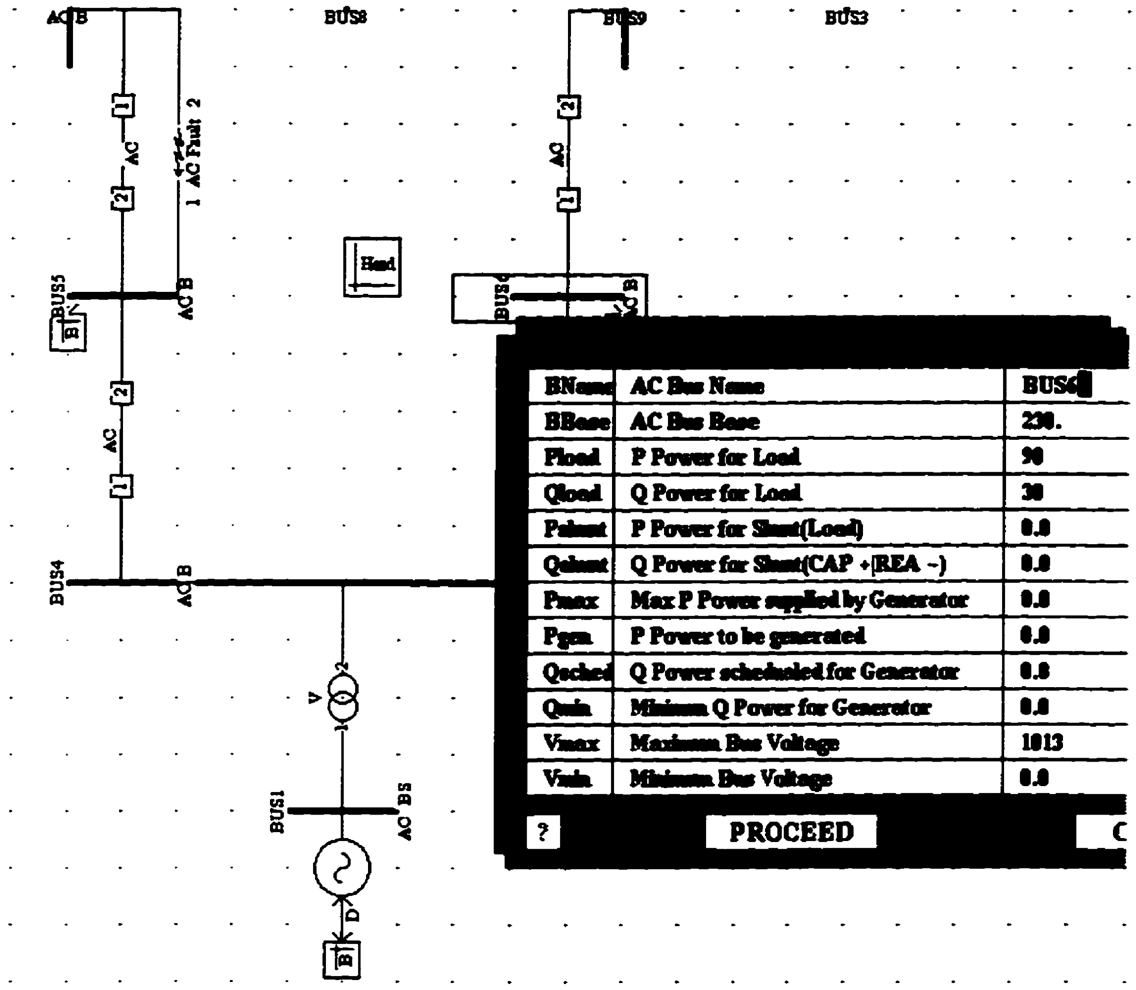
The DRAFT package shown in Figure 3.2 is perhaps the most powerful of the PSCAD family. With the DRAFT package of PSCAD, the user is able to graphically construct the power system models which are to be simulated.



No Title Subsystem: SS 1

Figure 3.2 The DRAFT Window in PSCAD

Power system component icons are located on the palette (right hand side of the DRAFT window) and placed onto the canvas (left hand side of the DRAFT window). Interconnection of the components completes the power system model. Parameters required by the various components are entered directly onto menus which appear when a new item has been placed (Figure 3.3). Power system models which contain a large number of components and interconnections can easily be handles since there are many layers of canvas, each of which can be scrolled on the screen.



BName	AC Bus Name	BUS6
BBase	AC Bus Base	230.
Pload	P Power for Load	90
Qload	Q Power for Load	30
Pshunt	P Power for Shunt(Load)	0.0
Qshunt	Q Power for Shunt(CAP +|REA -)	0.0
Pmax	Max P Power supplied by Generator	0.0
Pgen	P Power to be generated	0.0
Qsched	Q Power scheduled for Generator	0.0
Qmin	Minimum Q Power for Generator	0.0
Vmax	Maximum Bus Voltage	1013
Vmin	Minimum Bus Voltage	0.0

Figure 3.3 Pop-up menu for parameters entry of components

Once the user has completed the layout, hardcopy output can be obtained from either a PostScript based laser printer or plotter capable of accepting HP-GL commands.

RUNTIME:

The RUNTIME module provides the control panel and data acquisition system features for EMTDC in a manner akin to an operator's console. The software provides the interface to allow the user to load, start, stop and communicate with the simulation while it is in progress. Icons of various meters and on-line plotting of data as it is generated allow immediate feedback of the simulation case. Artifacts such as sliders, potentiometers, switches, push buttons are available for the user to interact with the program.

MULTIPLOT:

Plotting and analysis of data generated by EMTDC can be done using the MULTIPLOT module. Scaling and general formatting of the data is possible. Fourier analysis of plotted data can be performed directly. automated sequences may be programmed so as to simplify the handling of large amounts of data.

The MULTIPLOT module allows the user to prepare report ready pages of plots. Many plots may be combined and manipulated onto a single page.

In the above modules, DRAFT is used to draw the graphical representation of the power systems and generate two FORTRAN subroutine files as input to PSCAD's RUNTIME module. Then, in the RUNTIME module, they are compiled and linked to the EMTDC main program; the case is run; and output files are generated for further analysis. Finally, the MULTIPLOT module reads the output file, analyses data and plots output result.

The user is not confined to using the default library of models that comes with the program. If the user wants to design a new model of his or her own choosing, the FORTRAN interface provided through DRAFT and the DSDYN subroutine (described below) can be used. To enable flexible programming, EMTDC is structured with a main program coordinating all activities of input and output, network solution, and interfacing to the users model. The main program of EMTDC calls the following subroutines.

.DSDYN User defined dynamics (stored in xxx.dsd.f file)
.DSOUT User defined output (stored in xxx.dso.f file)

DSDYN and DSOUT are FORTRAN 77 subroutines which will be compiled for each simulation and linked to the EMTDC main program in the RUNTIME stage. Subroutines DSDYN and DSOUT can be formed through their automatic assembly in PSCAD. In facts, PSCAD supplies a common user interface in its component definition file to help users code their Fortran subroutine program.

We soon realised that the above FORTRAN subroutine code generation feature in PSCAD could be expanded so that instead of generating FORTRAN subroutine code, this code generation feature can be used to generate two plain text files in any desired format. These two plain text files, for example, *could become* input data files for the BPA programs!. These two plain text files would have the same names (xxx.dsd.f and xxx.dso.f) as in PSCAD, but would no longer be eligible for use with RUNTIME. In addition, since we have drawn the one-line diagram on the screen to represent the system, furthermore, we have also made an adaptation to display the load flow result on this one-line diagram.

3.3 Construction of Diagram and Generation of Input Data Files

As we know, components existing in the PSCAD master library generate FORTRAN subroutines in the xxx.dsd.f and xxx.dso.f two files on compiling(mapping). These FORTRAN subroutines will be compiled and linked to the EMTDC main program in the RUNTIME module and executed. The BPA input data files contain components information in the form of special format records. In order to trick DRAFT into generating BPA input data files, we must re-design all possible components which will be used in the BPA input data file. Thus, each type or subtype of record in the BPA input data files has a corresponding user-defined component in the PSCAD user-defined library. Users can only use these special user-defined components to build the system network model and enter all necessary information for the BPA programs during the construction. Figure 3.4 shows all the user-defined components for both the BPA Load Flow and Transient Stability programs.

All components in the master library for EMTDC can not be used for system one-line diagram building. While compiling (mapping), the entered data information in each component and other physical connection information between components will be written into the xxx.dsd.f and xxx.dso.f two files which are originally used for storing FORTRAN subroutines code.

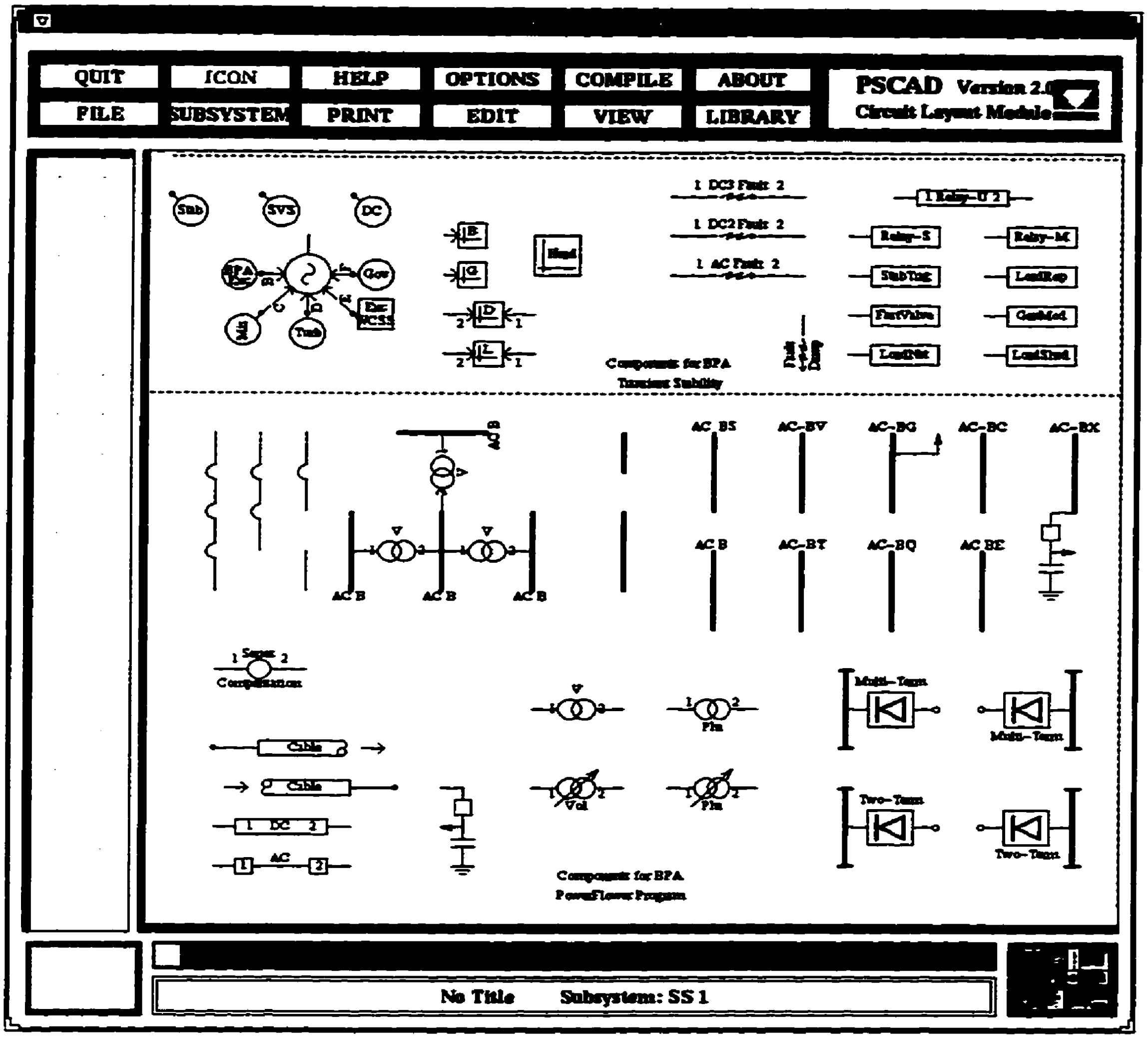


Figure 3.4 All components designed by the author for both the BPA Load Flow and Transient Stability on the PSCAD DRAFT palette

Although all possible components are redefined and the required information is generated, there are still some problems to be solved. The biggest issue is that even if the needed information data are generated, they are not placed in the required format in the data files. As we mentioned before, the BPA Load Flow and Transient Stability Programs require all the information to be placed in the exact position in each line record of their input data files.

Any shift of data in record columns will lead to error. Another issue is that the DRAFT compiling (or mapping) generates some information in the xxx.dsd.f and xxx.dso.f two files which is not required by the BPA Load Flow and Transient Stability Programs. To solve these two issues, an added conversion procedure is needed to erase unnecessary information and reorganize the data in BPA required format.

Due to the fact that DRAFT compiling was originally developed to deal with the EMTDC program, we can not solve these essential issues directly. Thus, we are required to develop a new "compiler" program. It is possible to entirely rewrite a new DRAFT compiler program for the BPA program, but, in this thesis, an alternative method is developed. In our approach, the Draft Compiler is still made use of to generate 90% of the required information although they are not all in the BPA required format. The Draft Compiler is considered to be a "pre-processor" and its output is piped into an additional compiler stage developed by us, which converts the raw DRAFT compiled output to the final BPA required format.

3.4 Running the BPA Load Flow and Transient Stability Programs

The BPA Load Flow and Transient Stability Programs are two separate analysis programs. They need to be run separately. But the BPA Transient Stability Program can not be run until after running the BPA Load Flow Program. The reason is that BPA Transient Stability Program needs a data file generated by the BPA Load Flow Program with the initial steady state network load flow which is used to initialise the state variables in the stability simulation.

In addition, there is another important issue to be mentioned here. If both the PSCAD/DRAFT and BPA Load Flow and Transient Stability Programs are installed on the user's machine, running them is simple. However, at this time at Manitoba HVDC Research Centre, the PSCAD/DRAFT can run on both SUN and HP workstation while the BPA Load Flow and Transient Stability Programs can only run on the HP workstation (although theoretically speaking, they can be re-compiled and run on any other platform workstation). So, there is additional complexity for the users to run the BPA Load Flow and Transient Stability programs if they are not installed on the same machine as the PSCAD/DRAFT program.

There are two way to handle this problem. The crude method is to use the unix command "rlogin" to establish a remote login session from local terminal to the remote machine on which the BPA Load Flow and Transient Stability Programs are installed, then the users run the BPA Load Flow and Transient Stability Programs on the remote machine same as on their local machine. But, this is inconvenient for the users because the command "rlogin" opens a new shell window and the users need to type a command in this new shell window to run the BPA Load Flow and Transient Stability Programs. It is a way which needs several steps to run the BPA Load Flow and Transient Stability Programs, however, for the users who are familiar with using mouse "point and click" to activate an event, this is not the best solution. In order to circumvent this problem, we designed a totally graphical environment which we call the "BPA Package Manager" program. we will discuss it in 3.6

A better way to handle above problem is to execute several commands in one mission. This can avoid interaction with the shell. Fortunately, there is an alternative which allows for execution of several commands in one mission on a remote machine without any interaction with the shell. The unix command “rsh” servers to connect to a specified host name (here, it was “bobbit”, the resident server for the BPA programs) and execute the specified command (here, they are “bpf” for the BPA Load Flow and “TSP” for the BPA Transient Stability Program).

Although the approach was developed because the BPA suite run only on the HP machine while the users could be dogged on another machine, it has the added advantage that even different components of the simulation process can be run on different platform in a networked computing environment.

There are two examples showing how the unix command “rsh” works. The following command appends the remote file lizard.file from a machine called lizard to the file called example.file on a machine called example.

```
example% rsh lizard cat lizard.file >> example.file
```

The next example appends the file lizard.file on a machine called lizard to the file another.lizard.file which also resides on a machine called lizard.

```
example% rsh lizard cat lizard.file ">>" another.lizard.file
```

For our case, the BPA Load Flow and Transient Stability Programs are installed on the "bobbit" machine and input data files are located in the directory ~/BPA/examples so the command looks like:

```
ace% rsh bobbit 'cd ~/BPA/examples;pwd;bpf xxx.load.dat'
ace% rsh bobbit 'cd ~/BPA/examples;pwd;TSP xxx.stab.dat'
```

3.5 Visualization of the Load Flow Result

Running the BPA Load Flow and Transient Stability Programs will generate output data files. Some of them exist in the ASCII format. These output data files contain the simulation results that are used by power system engineers for system analysis, planning and operation studies. Usually, in all data files, the user is mainly interested in voltage amplitude and phase angle at buses, real and reactive power at the two ends of transmission lines, and real and reactive power at the two ends of transformers. Although, the user can obtain all these result in the output data file, they are not gathered in one place. In contrast, they are scattered throughout the entire data output file. Reading the result output data is not easy for the users.

So, how to represent the result data visually and effectively is an important question. The result output from the BPA Load Flow Program is the steady state network solution; and unlike the EMTDC program solution, it is time independent. Thus, a natural way to represent the network solution is to write it back to the one-line diagram of the network available in PSCAD/DRAFT.

In order to achieve this, component definition in DRAFT is extended to include extra fields. These fields can not be accessed by the users, but the contents of these fields are displayed in the system one-line diagram. Thus, during the circuit construction stage, these fields are blank and no any results are displayed in the graphic rendering. The case is compiled and run on the BPA Load Flow; and the result output files are generated. A special post-processing program was developed by us to extract the relevant information from the output file and write it to these specially designed fields mentioned above. When the file is reloaded back into the DRAFT window, the original one-line diagram of the circuit re-appears with the network solution superposed on the various component.

Up to here, we have discussed all stages and related problems that the users will encounter in the procedures for model building, simulation and visualization of load flow result. Now, for the users, the main work is concentrated on the stage of model-building. The rest of the work, that is, pre-processing, running the BPA Load Flow and Transient Stability Programs, post-processing, displaying result data accompanied with system one-line diagram, are all taken care of by the computer. The only thing the users are required to do is to type a few simple commands for starting the various processes. These procedures greatly reduce the time and make the simulation exercise simple. If the users think the output result is not reasonable or they want to change the layout of components in network, they can return to the beginning, graphically modify the system model and repeat all stages. Compared with the traditional way, it saves a great amount of time, especially when frequent modifications are necessary.

3.6 BPA Package Manager

From the above description, it can be seen that several procedures are involved in the model-building, simulation exercise and result visualization. First of all, PSCAD/DRAFT is used to construct a system one-line diagram. Secondly, the one-line diagram must be converted to BPA input data files. Thirdly, the BPA Load Flow and Transient Stability Program must be run, Fourthly, the original draft file must be extended to include the result. Finally the one-line diagram must be rendered. In each stage, the users need to remember all the command names and type them for executing a particular step. However, the users are also required to enter in each step the name of the case file they are working with e.g. "preprocess case file". This could get cumbersome and boring if it has to be carried out repeatedly and hence can precipitate errors. A workaround for this problem is to design an interface panel, so that even these processes can be carried out by mouse "point and click". As a consistent GUI concept, the users tend to use the mouse rather than the keyboard to realize all events and actions. The philosophy in the writing of this program was to keep the user's hands off the keyboard but on the mouse. The only function of the keyboard is to enter numerical data when necessary, and little else.

This higher level interface has been christened by us as the BPA Package Manager. It helps the users to organize various commands and files in all stages of the model-building and simulation exercise. It uses image buttons to represent procedures, different image button for individual procedure. The advantages of using an image button instead of typing a command and file are easy to remember and fast to use. Clicking the image button means typing a command and executing it. For example, when the user clicks the draft image button, the PSCAD/DRAFT window appears on screen.

Chapter 4:

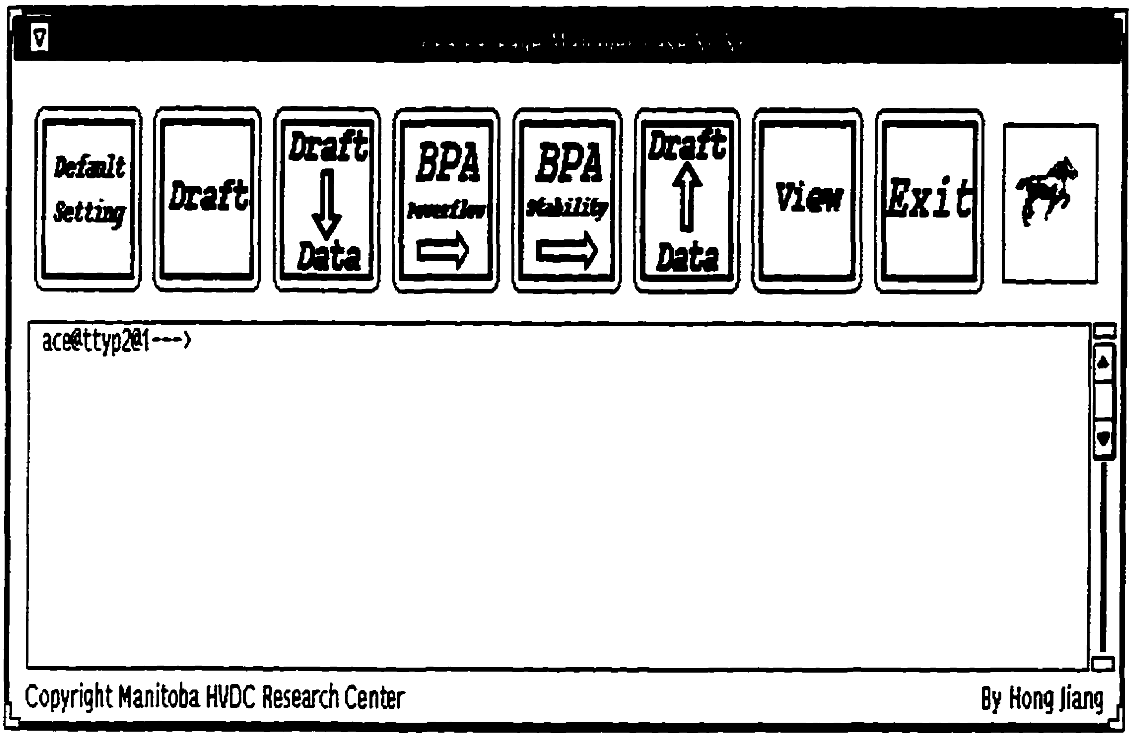
Program Operation and Example Implementation

In the previous chapter we introduce various procedures and discuss all potential problems that the users will meet during the model-building and simulation exercise. In this chapter, we will go in detail to describe the command and operation in each procedure. The best way to understand how the GUI for load flow and transient stability works is to use it to construct a simple model, compile and generate input data files required by the BPA Load Flow and Transient Stability Programs; run the BPA Load Flow and Transient Stability Programs and generate an output data file; process the output data file and finally show or print out the result data superposed with the system one-line diagram.

For the users who are familiar with PSCAD/EMTDC, the following steps are analogous.

4.1 Starting up and Initialization

To start the BPA Package Manager program designed in this thesis, the users can be in any directory and execute the command called BPAPM. After initialization, a new window entitled "BPA Package Manager" appears on the screen as Figure 4.1 shown below.



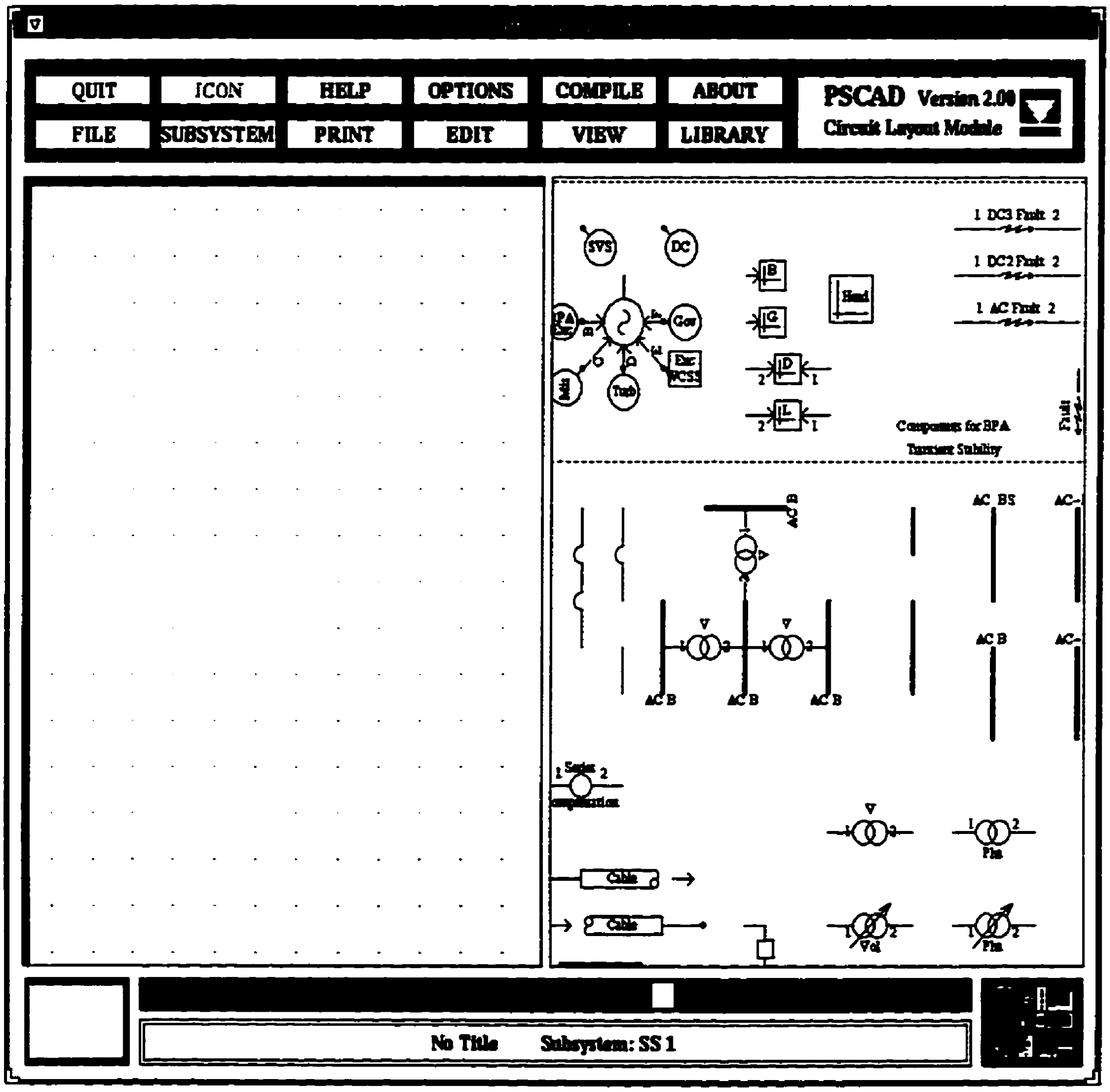
```
ace@ttyp2@1--->
```

Figure 4.1 BPA Package Manager

The window is composed of two areas, the upper is the command button area, the rest is a terminal command subwindow.The entire window can be resized according to the user's demand. Before starting the circuit diagram construction, the users need do some simple tasks such as setting default values, i.e., working directory, case name and machine name on which the BPA Load Flow and Transient Stability Programs are installed. To do this, the users move the mouse to point to the "Default Configuration" button and click the right key of the mouse. A subwindow titled "Default Configuration" appears to ask the users to enter the default working directory, case name and machine name and so on. After finishing these default pre-setting, the users can either close the subwindow or keep it alive.

4.2 Constructing Circuit

After finishing the default configuration, the circuit diagram construction begins. the users move the cursor to the “Draft” button and click the left key of the mouse. A new DRAFT window appears as in Figure 4.2. The window is divided into several different areas.



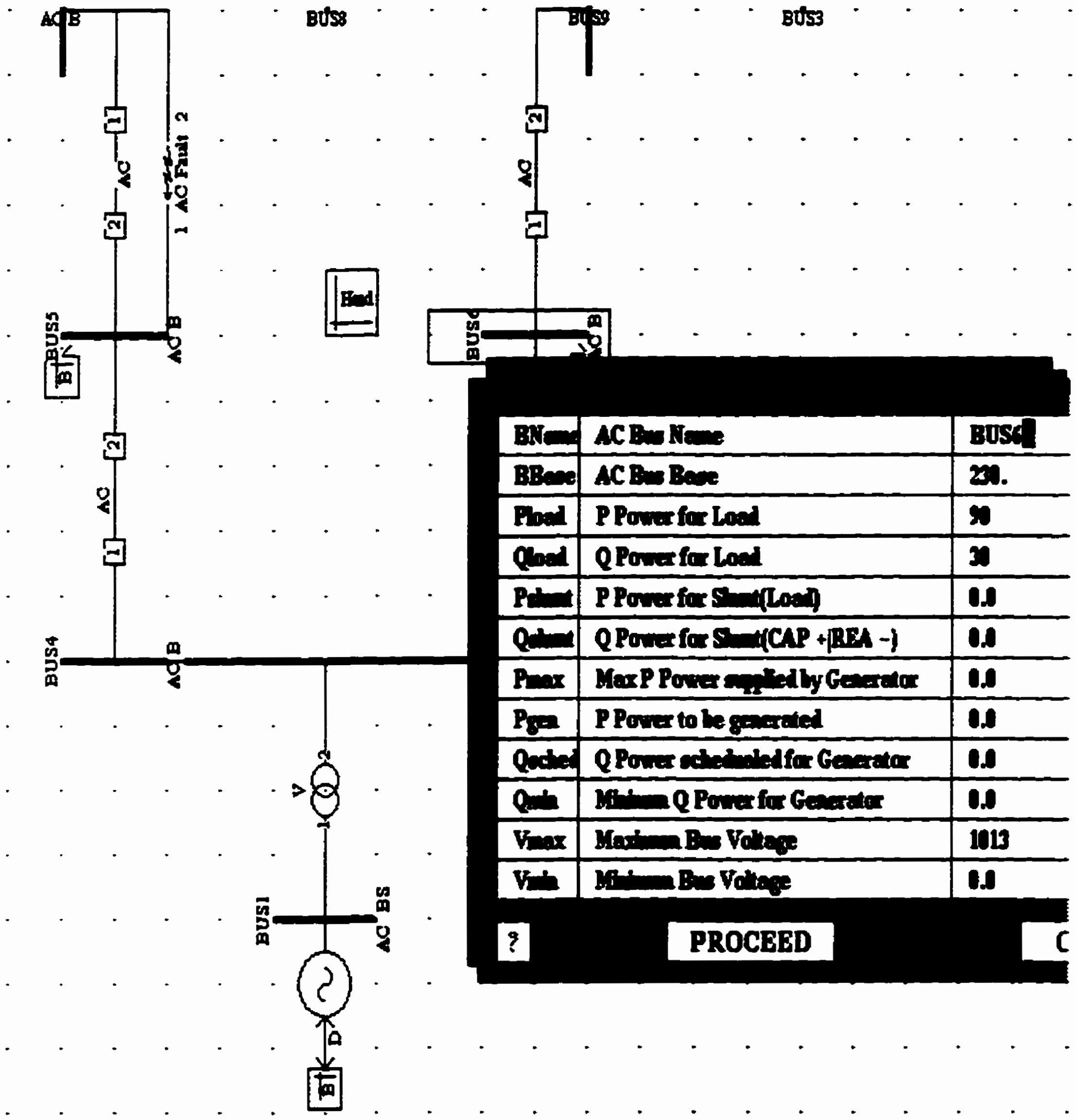
No Title Subsystem: SS 1

Figure 4.2 Draft Window for One-Line Diagram Construction

If there exists a circuit one-line diagram file and the users want to modify or use it directly for simulation, the FILE->LOAD (click on the FILE button and then on the LOAD button that appears on the pop-up submenu) command loads a circuit drawing into DRAFT. If the current circuit drawing has not been saved when a new circuit is loaded, a toggle dialogue window appears allowing user to save the current circuit. It is a good idea for the user to save the circuit drawing in a specified file name before entering the next step.

Certainly, users can save a circuit drawing under a new name if they want. The "FILE->SAVE AS.." command is used to save a circuit drawing under a new name. There exist two methods of saving: “Quick” and “Transfer”. The default saving method is “Quick”. The “Transfer” option is used when transferring a circuit drawing from one user, group, or site to another, where it is possible that some custom designed components may not exist. In these cases, the component definition information is included inside the file to assist in the transfer process. In our circumstance, except single line component, all other components are “custom” as far as PSCAD/EMTDC is concerned, so if users want to transfer their draft file from one user, group, or site to another, they need to save the circuit drawing file using the “Transfer” method.

The process of circuit construction is similar to that of drawing a schematic diagram of the power system circuit under study. The components needed to construct the system circuit are picked up from the component library on the right side and placed on the drawing canvas. The components are connected using the stretchable wire component. The parameters for components are entered by using a pop-up menu as seen in Figure 4.3 and a constructed diagram for BPA Load Flow and Transient Stability programs is shown in Figure 4.4.



BName	AC Bus Name	BUS4
BBase	AC Bus Base	230.
Pload	P Power for Load	90
Qload	Q Power for Load	30
Pshunt	P Power for Shunt(Load)	0.0
Qshunt	Q Power for Shunt(CAP +|REA -)	0.0
Pmax	Max P Power supplied by Generator	0.0
Pgen	P Power to be generated	0.0
Qsched	Q Power scheduled for Generator	0.0
Qmin	Minimum Q Power for Generator	0.0
Vmax	Maximum Bus Voltage	1013
Vmin	Minimum Bus Voltage	0.0

Figure 4.3 Pop-up menu for parameters entry of components

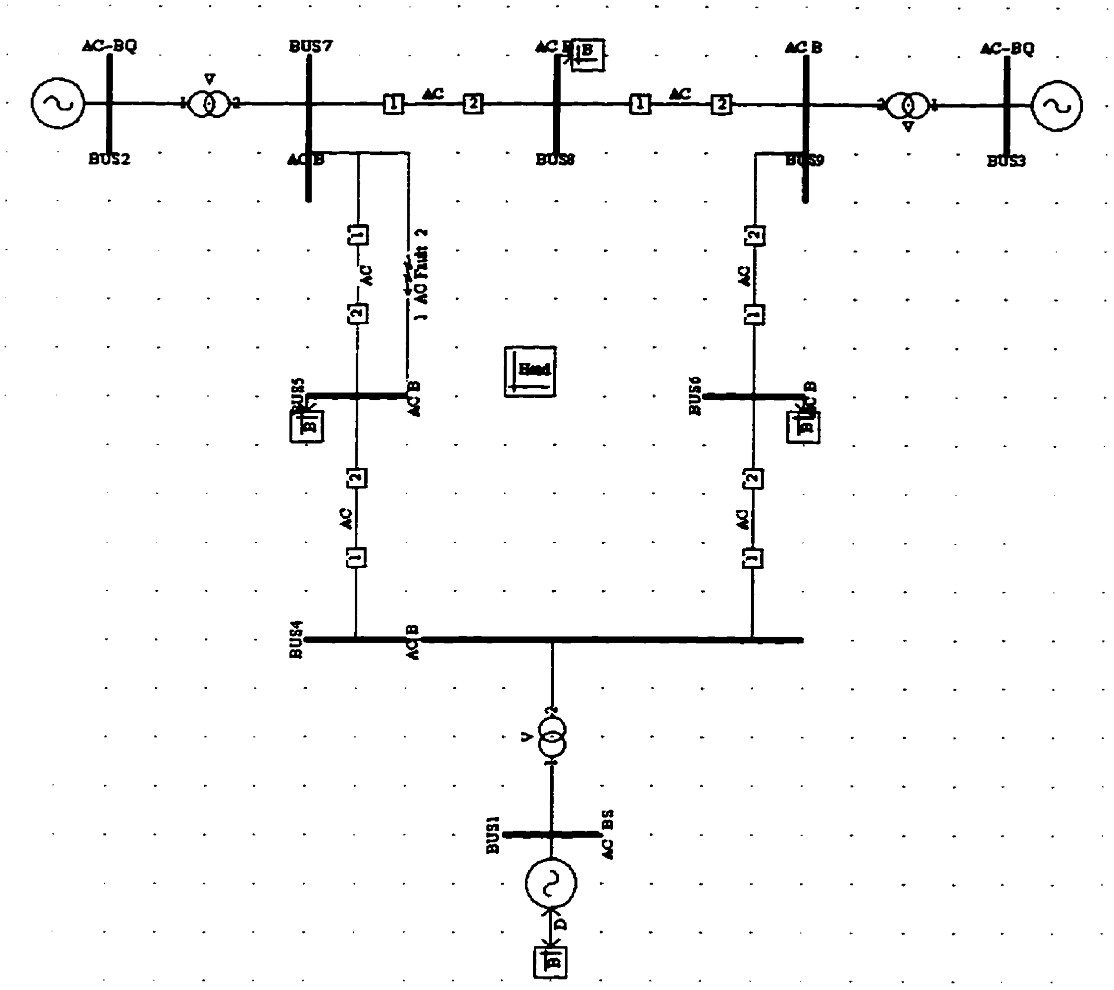


Figure 4.4 Nine Buses Network One-Line Diagram

4.3 Compiling (Mapping) and Pre-Processing

After completing the model-building, the next step is to "compile" (or map) the one-line diagram, which means to convert the one-line diagram to an intermediate data file which is subsequently processed into the BPA input data format. A special pre-processing program was developed by us to handle the final data conversion. In the next step, this special pre-processing program is applied to finish the conversion from the intermediate file to the final data format file used by the BPA Load Flow and Transient Stability Programs. This is a key step in the entire procedure.

If some of the circuit parameters have not been specified when a compile is attempted, DRAFT will request these parameters to be entered. If the compile action results in any error messages or warning messages, an error/warning window will open showing the location of the first error. From this window, the other errors and warnings may be identified. This is a standard feature of PSCAD/DRAFT.

After compiling, two new intermediate files are generated by the compiler: one with extension name ".dsd.f", another with extension name ".dso.f". List 4.1 and List 4.2 are corresponding two intermediate files for nine buses network modal in Figure 4.4. It is these two files that are used to record the information required by the BPA Load Flow and Transient Stability programs. The extension name of these files have special significance in EMTDC, but we need not go into that here.

```
C ==================================================================
      SUBROUTINE DSDYN
C ------------------------------------------------------------------
C   Title:s
C
C   *** FORTRAN 77 ***
C
C   Associated files:
C      Bus9.dat
C      Bus9.dso.f
C
C ==================================================================
C   Include and Common Block Declarations
C ------------------------------------------------------------------
      INCLUDE 'emt.d'
      INCLUDE 'emt.e'
      COMMON  /S1/TIME,DELT,ICH,PRINT,FINTIM
      COMMON  /S2/STOR(ND10),NEXC
      COMMON  /S4/VAR(ND11),CON(ND12),PGB(ND13)
      REAL    PI_, TWO_PI, PI_BY3, PI2_BY3
```

```
      DATA    PI_/3.141592654/,    TWO_PI/6.283185307/
      DATA    PI_BY3/1.047197551/, PI2_BY3/2.094395102/
C
C ==================================================================
C   Function Declarations
C ------------------------------------------------------------------
C
C ==================================================================
C   Variable Declarations
C ------------------------------------------------------------------
      REAL    TIME,DELT,PRINT,FINTIM,STOR,VAR,COM,PGB
      INTEGER ICH,NEXC
C
C ==================================================================
C   Subsystem: "SS 1"
C ------------------------------------------------------------------
L 1 1 1 2 BLK 0 1 0 0.0 0 0.00850 0.0720 0.0 0.07450
                    0.0 BLANK 0 0 0 0 0.0 0.0
DSP 1 1 1 2 ACLN3 1
L 1 2 1 3 BLK 0 1 0 0.0 0 0.01190 0.10080 0.0 0.10450
                    0.0 BLANK 0 0 0 0 0.0 0.0
DSP 1 2 1 3 ACLN4 1
T  1 4 1 3 BLK 0 B 0 0.0 0 0.0 0.05860 0.0 0.0
                    13.80 230.0 0 0 0 0 0.0 0.0 0.0
DSP 1 4 1 3 TRAN3 B
B  1 3 1 3 BLK BUS9 230.0 1 0.0 0.0
                    0.0 0.0 0.0 0.0 0.0 0.0 1.020 0.0 0
B  1 2 1 2 BLK BUS8 230.0 1 100.0 35.0
                    0.0 0.0 0.0 0.0 0.0 0.0 1.016e+03 0.0 0
B  1 5 1 5 BLK BUS5 230.0 1 125.0 50.0
                    0.0 0.0 0.0 0.0 0.0 0.0 0.9960 0.0 0
B  1 6 1 6 BLK BUS6 230.0 1 90.0 30.0
                    0.0 0.0 0.0 0.0 0.0 0.0 1.013e+03 0.0 0
B  1 7 1 7 BLK BUS4 230.0 1 0.0 0.0
                    0.0 0.0 0.0 0.0 0.0 0.0 1.020 0.0 0
T  1 8 1 7 BLK 0 B 0 0.0 0 0.0 0.05760 0.0 0.0
                    16.50 230.0 0 0 0 0 0.0 0.0 0.0
DSP 1 8 1 7 TRAN2 B
BS 1 8 1 8 BLK BUS1 16.50 1 0.0 0.0 0.0 0.0
                    250.0 71.60 100.0 -100.0 1.04e+03 0.0 0
B  1 1 1 1 BLK BUS7 230.0 1 0.0 0.0
                    0.0 0.0 0.0 0.0 0.0 0.0 1.020 0.0 0
BQ 1 4 1 4 BLK BUS3 13.80 1 0.0 0.0 0.0 0.0
                    110.0 85.0 60.0 -60.0 1.025e+03 0 0
BQ 1 9 1 9 BLK BUS2 18.0 1 0.0 0.0 0.0 0.0
                    165.0 163.0 100.0 -100.0 1.025e+03 0 0
T  1 9 1 1 BLK 0 B 0 0.0 0 0.0 0.06250 0.0 0.0
                    18.0 230.0 0 0 0 0 0.0 0.0 0.0
DSP 1 9 1 1 TRAN1 B
L 1 7 1 5 BLK 0 B 0 0.0 0 0.010 0.0850 0.0 0.0880
                    0.0 BLANK 0 0 0 0 0.0 0.0
DSP 1 7 1 5 ACLN1 B
L 1 7 1 6 BLK 0 1 0 0.0 0 0.0170 0.0920 0.0 0.0790
                    0.0 BLANK 0 0 0 0 0.0 0.0
DSP 1 7 1 6 ACLN5 1
L 1 6 1 3 BLK 0 1 0 0.0 0 0.0390 0.170 0.0 0.1790
                    0.0 BLANK 0 0 0 0 0.0 0.0
DSP 1 6 1 3 ACLN6 1
L 1 1 1 5 BLK 0 1 0 0.0 0 0.0320 0.1610 0.0 0.1530
                    0.0 BLANK 0 0 0 0 0.0 0.0
DSP 1 1 1 5 ACLN2 1
C ==================================================================
      RETURN
      END
```

List 4.1 Intermediate data file (.dsd.f) after mapping

```
C ==================================================================
      SUBROUTINE DSOUT
C ------------------------------------------------------------------
C   Title:s
C
C   *** FORTRAN 77 ***
```

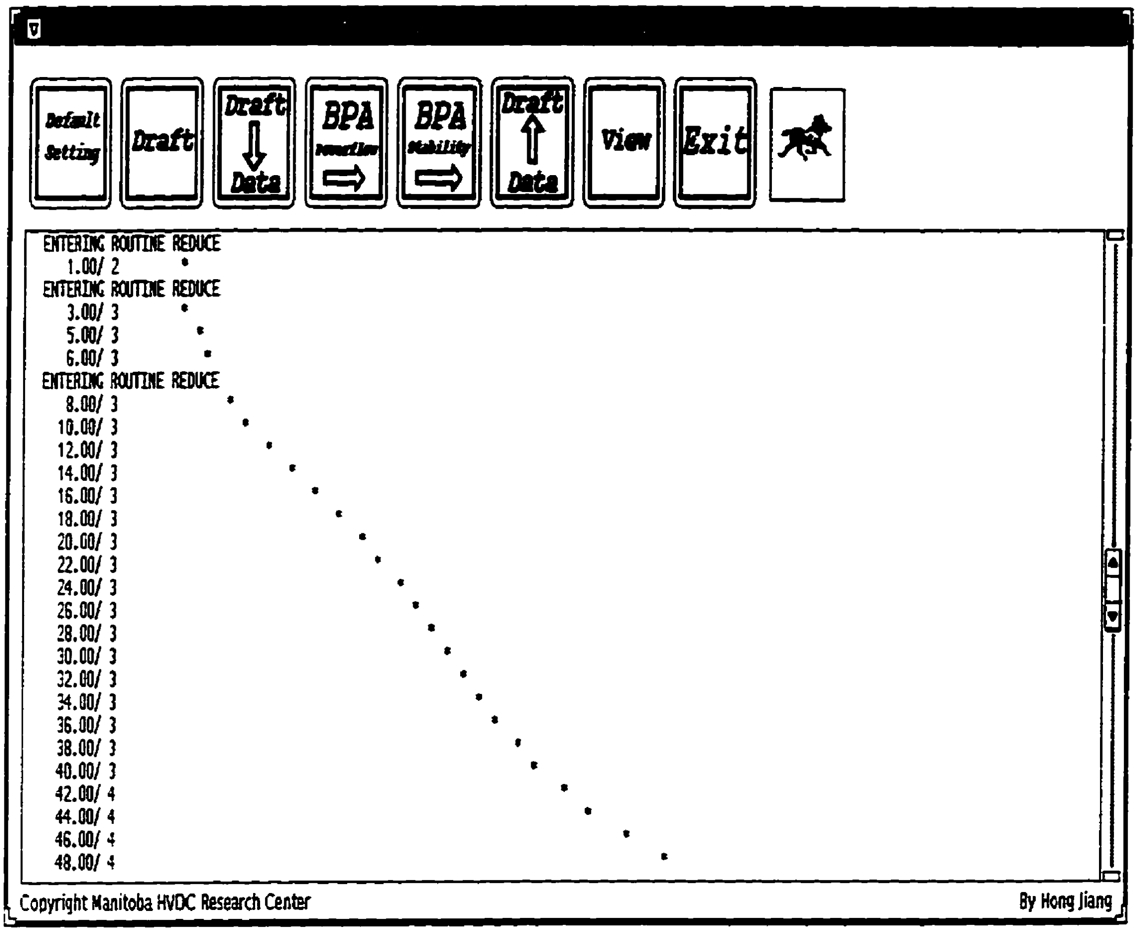
```
C
C   Associated files:
C      Bus9.dat
C      Bus9.dsd.f
C
C ==================================================================
C   Include and Common Block Declarations
C ------------------------------------------------------------------
      INCLUDE 'emt.d'
      INCLUDE 'emt.e'
      COMMON  /S1/TIME,DELT,ICH,PRINT,FINTIM
      COMMON  /S2/STOR(ND10),NEXC
      COMMON  /S4/VAR(ND11),CON(ND12),PGB(ND13)
      REAL    PI_, TWO_PI, PI_BY3, PI2_BY3
      DATA    PI_/3.141592654/,    TWO_PI/6.283185307/
      DATA    PI_BY3/1.047197551/, PI2_BY3/2.094395102/
C
C ==================================================================
C   Function Declarations
C ------------------------------------------------------------------
C
C ==================================================================
C   Variable Declarations
C ------------------------------------------------------------------
      REAL    TIME,DELT,PRINT,FINTIM,STOR,VAR,CON,PGB
      INTEGER ICH,NEXC
C
C ==================================================================
C
C ==================================================================
C   Subsystem: "SS 1"
C ------------------------------------------------------------------
M  1 9 2 100.0 0.850 1 1 BLK
                     0.0 0.0 0.0 0.0
MF 1 9 2 640.0 100.0 100.0 100.0 0.0
                     0.11980 0.19690 0.89580 0.86450 6.0
                     0.540 0.05210 0.0 0.0 0.0
M  1 4 3 100.0 0.850 1 1 BLK
                     0.0 0.0 0.0 0.0
MF 1 4 3 301.0 100.0 100.0 100.0 0.0
                     0.18130 0.250 1.3130 1.2580 5.890
                     0.60 0.07420 0.0 0.0 0.0
B1 1 8 3 3 0
M  1 8 1 100.0 1.0 1 0 BLK
                     0.0 0.0 0.0 0.0
MF 1 8 1 2.364e+03 100.0 100.0 100.0 0.0
                     0.06080 0.09690 0.1460 0.09690 8.960
                     0.0 0.03360 0.0 0.0 0.0
B1 1 6 3 3 0
B1 1 5 3 3 0
B1 1 2 3 3 0
LS2 2 1 1 1 5 0 0 1 1 0.0
                     0.0 0.0 0.0 888888 888888
LS2 2 1 1 1 5 0 0 1 0 1.0
                     0.0 0.0 0.0 888888 888888
LS2 2 1 1 1 5 3 0 1 1 6.0
                     0.0 0.0 0.0 888888 888888
F1F 0.0 2.0 300.0 2.0 15 0.010 25 0.10 8.0 1.0
                     60.0 0 0 0 0 0 0 0 0 0 0
M1H 4 0.0 1.30 0.80 60.0 60.0 500 0 0.0 0.0
B1H 1 0.0 0.0 2 0.0 0.0
L1H 1 0.0 0.0 2 0.0 0.0 3 0.0 0.0 4 0 0.010 0.050 0.050
D1H 1 0.0 0.0 2 0.0 0.0 3 0.0 0.0
                     4 0.0 0.0 5 0.0 0.0 6 0.0 0.0
                     7 0.0 0.0 8 0.0 0.0 9 0.0 0.0
G1H NULL 0.0
                     1 0.0 0.0 2 0.0 0.0 3 0.0 0.0
                     4 0.0 0.0 5 0.0 0.0 6 0.0 0.0
                     7 0.0 0.0 8 0.0 0.0 9 0.0 0.0
                     10 0.0 0.0 11 0.0 0.0 12 0.0 0.0
                     13 0.0 0.0 14 0.0 0.0 15 0.0 0.0
E1H 0 0
C ==================================================================
      RETURN
      END
```

List 4.2 intermediate data file (.dso.f) after mapping

Originally, the PSCAD's compiler (mapping) forms xxx.dsd.f and xxx.dso.f two files in FORTRAN 77 format. Then, these two files are compiled and linked to the EMTDC main simulation program in the RUNTIME stage. Since we use user-defined components totally, we do not want FORTRAN subroutine to be generated but a plain text file which is converted into the BPA input data format and used in the next step. The List 2.1 and List 2.2 are the data files converted (pre-processed) from List 4.1 and List 4.2. They are in the required format for the BPA Load Flow and Transient Stability Programs.

4.4 Running the BPA Load Flow and Transient Stability programs

The BPA Load Flow and Transient Stability programs are two separate analysis programs. They need to be run separately. The BPA Transient Stability program can not run until after running BPA Load Flow program. The reason is that running the BPA Transient Stability program starts from the steady state generated by the BPA Load Flow program. On the BPA Package Manager, I have designed two image buttons, one each for the BPA Load Flow and Transient Stability programs respectively. Running the BPA Load Flow and Transient Stability program is very simple, just "point and click" the image buttons respectively. But keep in mind the running order. Figure 4.5 shows a typical run of the BPA Transient Stability program, which produces a crude "print plot" of stars as the run progresses. The vertical axis represents time and the horizontal painter of the star represents the magnitude of a typical variable being plotted



```
ENTERING ROUTINE REDUCE
   1.00/ 2        *
ENTERING ROUTINE REDUCE
   3.00/ 3        *
   5.00/ 3          *
   6.00/ 3           *
ENTERING ROUTINE REDUCE
   8.00/ 3              *
  10.00/ 3                *
  12.00/ 3                   *
  14.00/ 3                      *
  16.00/ 3                         *
  18.00/ 3                            *
  20.00/ 3                               *
  22.00/ 3                                 *
  24.00/ 3                                    *
  26.00/ 3                                      *
  28.00/ 3                                        *
  30.00/ 3                                          *
  32.00/ 3                                            *
  34.00/ 3                                              *
  36.00/ 3                                                *
  38.00/ 3                                                   *
  40.00/ 3                                                     *
  42.00/ 4                                                         *
  44.00/ 4                                                            *
  46.00/ 4                                                                 *
  48.00/ 4                                                                      *
```

Copyright Manitoba HVDC Research Center   By Hong Jiang

Figure 4.5 Running of the BPA Transient Stability program

4.5 Post-Processing and Output Rendering

As described earlier, a feature was developed for graphical rendering of the solution information in DRAFT. To implement this function, each component needs to have a pre-defined text parameter in its graphics section of the component definition. Then, with the post-processing program developed by us, those data are added to the circuit drawing file (.dft). Of course, before running the post-processing program to add the result data to the

*.dft file, the user should have been saved the *.dft. The post-processing program does not destroy the rest of the circuit drawing file, it only reads the data from the output file of running BPA Load Flow and then writes them in the correct location in the circuit drawing file.

The user could over-write the original circuit diagram file with the result at output or to save the output and the drawing in a new file. I suggest users form a new circuit diagram file because some times, users do want their original circuit drawing file without the BPA Load Flow result.

The procedure to achieve the above objective is simple. Just point the "View" button and click the mouse. A new "Draft" window appears and displays the one-line diagram superposed with the load flow result data. Of course, the FILE->LOAD command is also used to load the circuit drawing file and show it on the left drawing canvas. The resultant diagram can be printed for a hardcopy output on a PostScript or HP-GL graphics. Figure 4.6 shows the rendering of the load flow result on the diagram.

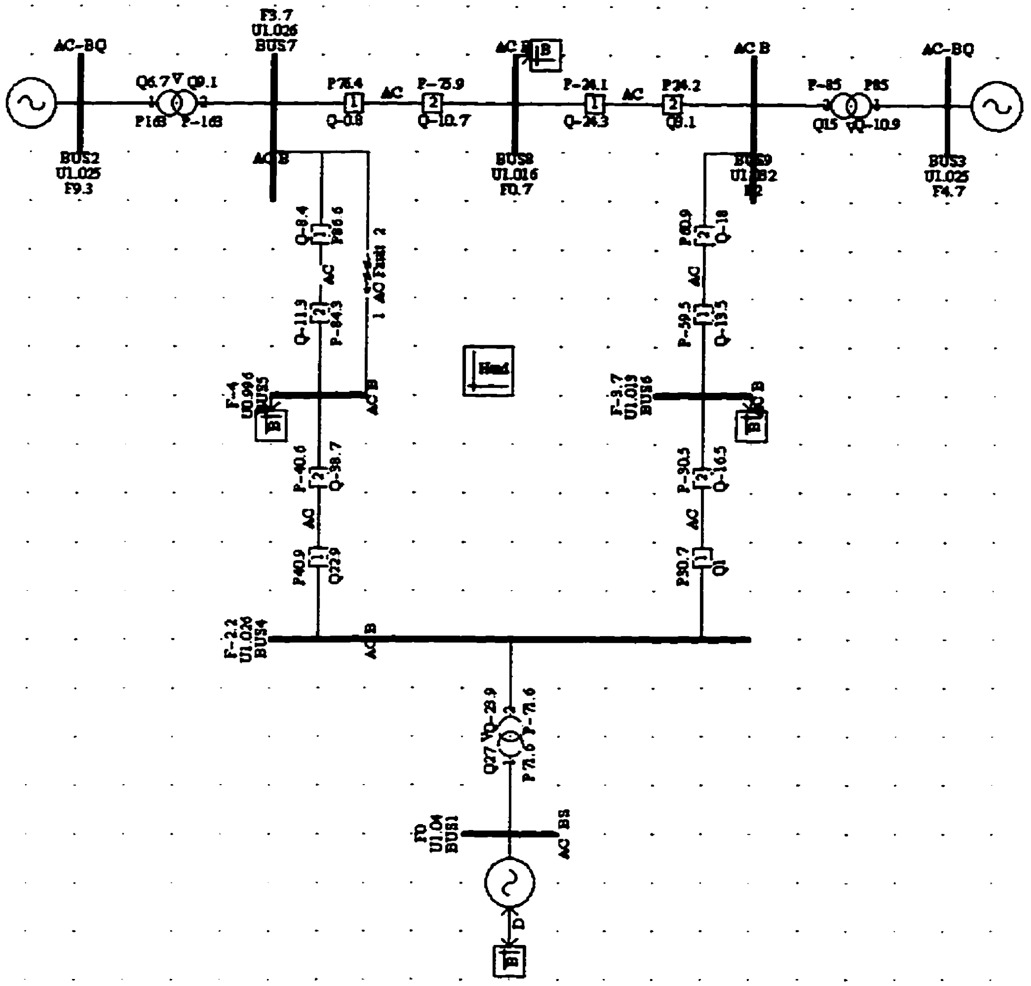


Figure 4.6 Classical nine buses one-line diagram superposed

with the BPA Load Flow results

Chapter 5:

Conclusion and Further Works

As mentioned in Chapter 1, the objective of this thesis was to provide a more efficient method of data entry for running the BPA Load Flow and Transient Stability program and visualization of the results. These have been realized through the implementation of a graphical interface.

A graphical interface has been proved to be a better solution for such problem as BPA Load Flow and Transient Stability programs since it mimics the way power system engineers draw a system circuit diagram on the paper. The information in the diagram are extracted by a pre-processing program, which is transparent to the user. When users complete constructing the system circuit, a data file is already mapped in required format at the same time.

Similarly, when the load flow finish running, its simulation results are extracted and mixed with original diagram file using a post-processing program which is transparent to the user and a visual rendering of the output is provided.

Also, to avoid remembering and typing various commands and file names during the entire simulation exercise, the BPA Package Manager was developed in this thesis to give the user an integrated environment.

In conclusion, a graphical interface is a very easy, friendly and efficient method for mapping data for the BPA Load Flow and Transient Stability program. With proper system circuit construction, all errors in the mapping information to a data file are eliminated. Typographical errors are much easier detected because the information entered by the user is replaced by its respective circuit component. Also, with the ability to store a drawing of the circuit, data files may easily be updated.

One of the benefits that could be interest to users is that the methods used in this thesis can be applied to other load flow and stability programs. They may also be applied in other fields of power system engineering.

References

[1] A.M. Gole, O.B.Nayak, T.S. Sidhu, M.S.Sachdev, *"A Graphical Electromagnetic Simulation Laboratory for Power Systems Engineering Programs"*, IEEE PES Summer Meeting, Portland, OR, July 1995.

[2] Prabha Kundur, Kip Morison, *"Developments in Power system Analysis"*, IEEE Canadian Review, No.21.,Winter 1995.

[3] Om Nayak, Garth Irwin and Arthur Neufeld, *"A Totally Graphical User Interface for Electromagnetic Transients Simulation Tools"*, IEEE Computer Application in Power, Vol. 8, No.1, January 1995.

[4] Hermann Dommel and Nobuo Sato, *"Fast Transient Stability Solution"*, IEEE Transactions on Power Apparatus and System, July/August 1972.

[5] Herman Dommel, *"Digital Computer Solution of Electromagnetic Transients in Single- and Multi-phase Network"*, IEEE Trans. PAS, Vol. PAS-88, No.4, APril 1969, pp 388-399

[6] W.D. Stevenson, *"Elements of Power System Analysis"*, McGraw-Hill, 1975.

[7] BR Gupta, *"Power System Analysis and Design"*, Wheeler Publishing, 1993.

[8] *"BPA Load Flow Program User's Guide",*

[9] *"BPA Transient Stability Program User's Guide",*

[10] Valerie Quercia and Tim O'Reilly, *"X Window System User's Guide"*, O'Reilly & Associates, Inc., 1993

[11] Dan Heller, *"XView Programming Manual*", O'Reilly & Associates, Inc. 1990

[12] Dan Heller, *"XView Reference Manual"*, O'Reilly & Associates, Inc. 1990

[13] Adrian Nye, *"Xlib Programming Manual"*, O'Reilly & Associates, Inc. 1990

[14] Adrian Nye, *"Xlid Reference Manual"*, O'Reilly & Associates, Inc. 1990

[15] *"PSCAD Circuit Design FileManager Manual"*, Manitoba HVDC Research Center

[16] *"PSCAD Draft Manual"*, Manitoba HVDC Research Center

[17] *"PSCAD Tline & Cable Manual"*, Manitoba HVDC Research Center

[18] Brian W.Kernighan & Dennis M.Ritchie, *"The C Programming Language"*, Prentice-Hall, Inc. 1978

[19] Samuel P.Harbison & Guy L.Steels Jr., *"C: A Reference Manual"*, Prentice-Hall, Inc. 1987.

[20] S.R.Bourne, *"The UNIX System"*, Addison-Wesley Publishing Company, 1983

APPENDIX A: Functional Specifications for Western Systems Coordinating Council Load Flow and Transient Stability Computer

Part I: General, Load Flow and Transient Stability

A. WSCC will consider any deviations and alternates of these specifications, provided the details of such deviations and alternates are fully explained by the supplier.

B. Load Flow and Transient Stability programs shall be compatible. Power flow shall accept card input or tape input from previous case. Stability program shall accept load flow output and system data directly from load flow history tape of any case. Load flow and stability shall also run "back to back" if desired, that is, stability program run as a continuation of load flow without operator intervention to initiate stability run.

C. Program shall be written in a standard FORTRAN language for ease of modification except for relocatable FORTRAN subroutines.

D. Program shall be completely documented including full explanation of theory and equations upon which calculations are based as well as the FORTRAN listings. Complete users manual and program documentation shall be supplied with delivery of programs. A glossary of terms shall be included in the documentation.

E. Programs shall be written for computer acceptable to the Council.

F. Reduced computer storage capacity shall result in reduced program limits rather than elimination of program features.

G. Program features shall not be at expense of excessive input manipulation or machine time.

H. Diagnostics shall be provided in program operation to reveal cause of program stoppage or failure to converge. Included in diagnostic shall be maximum real and quadrature components of power and voltage change and names of buses on which they occur. Bus mismatch shall be available as diagnostic at users option. Diagnostics shall be available at each iteration at users option.

Part II: Load Flow Program

A. System limits

1. 2000 buses
2. 3000 branches, 1000 of which may be transformers with fixed or LTC taps.
3. 50 phase shiftier
4. 60 interchange control areas
5. Voltage at 400 buses controlled remotely by generation or transformer LTC
6. 90 loss accumulation areas other than interchange areas
7. 10 two-terminal and 5 three-terminal DC lines

B. Buses

1. Machines, loads, shunt capacitors, and shunt reactors shall all be representable separately at each bus
2. Loads shall be representable as scheduled mw and mvar.

3. Shunt admittances shall be representable in per-unit as follows:

(a) Fixed admittance

(b) Switched capacitor in steps with on and off voltage

(c) Switched reactor in steps with on and off voltage

4. Machines shall be representable in per-unit as follows:

(a) Scheduled voltage magnitude and angle (swing).

(b) Scheduled mw and voltage magnitude, with or without mvar limits. In event mvar limit is reached, voltage schedule is no longer held.

(c) Scheduled mw and mvar, with or without voltage limits. In event voltage limit is reached, mvar schedule is no longer held.

(d) Scheduled mw and variable mvar to hold voltage on remote bus. Provision shall be made at 50 voltage controlled buses for the controlling action to be shared in scheduled proportion among the reactive outputs of up to 5 remote machines.

5. Buses shall be identified by name rather than number. Bus name shall include bus voltage base.

C. Branches

1. Branch impedance (R+jX) and admittance (G+jB) are to be entered in per-unit on mva bases or bases as specified by user.

2. Program shall provide means for series-parallel combinations of passive elements, (lines, transformers, capacitors, etc.) without intervening active elements (generators, mw scheduled loads) to be combined into equivalents for calculation purposes and restored in

detail in the output.

3. There shall be no restrictions on the magnitude or sign of branch impedances. Zero, low value or negative impedances shall be acceptable.

4. Pi elements having unequal legs shall be acceptable.

5 Parallel lines shall be permitted with identity retained.

6. Branches shall be identified by terminal bus names. Branch numbers shall not be required.

7. Provision shall be made for line current ratings and transformer mva ratings for overload checking.

8. Provision shall be made for calculating line currents at terminals on selected lines.

D. Transformers and Phase Shiftier

1. Fixed tap transformer ratio shall be entered in kv.

2. LTC transformer voltage range and step size shall be specified for each transformer. Provision for automatic impedance adjustment shall be made.

3. LTC transformer shall provide voltage or var control at user option.

4. Phase shifters shall have power or angle subject to user schedule. Phase shiftier impedance adjustment with step change shall be specified by user.

E. Area Interchange Control

1. Each area to have individual swing machine.

2. Area tie lines shall be identified in the branch input data by the user. Either end of the tie branch shall be the metering point as specified for the particular branch by the user. It shall be possible to assign any type branch (line, series capacitor, transformer. etc.) as a tie line.
3. It shall be possible to combine individual interchange areas into larger interchange areas without changing bus or branch cards.
4. Area interchange tolerance shall be specified by user.

F. DC Lines

Representation of direct current lines in the load flow program shall be compatible with that of the stability program.

G. Solution Techniques

1. The basic solution method shall provide essentially exact results with reasonable amount of machine time for size of system. Essentially exact results shall mean less than 0.0001 pu power error per bus and less than 0.01 pu power error for system.
2. Solution method shall provide a tough tie line check prior to first voltage balance in order to prevent possible solution failure or abnormal machine time due to concentration of initial area interchanged errors (due to inaccurate load, generation, or loss estimates) on main swing machine.
3. Any net bus mismatch in an interchange area shall be balanced out by the swing machine for that area.

4. Area interchange shall be measured by summing tie lines at specified metering points rather than by any other method involving loads, losses and generation.

5. Transformer LTC solution shall permit multiple step tap corrections in a single tap change when large tap changes are required.

H. Output

1. Output shall include bus names, voltage magnitude in per unit and kv, angle in degrees, mw and mvar of all machines, loads, shunt devices, and flows at each end of all branches, and terminal current in amperes of selected lines.

2. All transformer LTC tap values and phase shiftier angle settings shall appear with the particular branch readings.

3. Whenever line or transformer ratings are exceeded, the ratings shall be flagged.

4. Whenever bus voltage or machine mvar limits have been reached or exceeded, the appropriate readings shall be flagged.

5. Branch mw and mvar losses shall be listed with the particular branch flow.

6. Bus and branch output data shall be listed in alphabetical bus name order within interchange area groupings.

7. Area interchange summary shall be provided showing desired and actual net interchange, area total losses generation, and load, area swing generation, tie line flows, net area bus mismatch, maximum bus mismatch, spinning reserve.

8. Provision shall be made to summarize losses by a code other than interchange area.

9. A generator summary shall be provided listing individual bus voltage, mw and mvar,

desired voltage, mvar limits and remotely controlled bus data.

10. A transformer LTC summary shall be provided listing final tap, voltage held, desired voltage, maximum and minimum ratios.

11. A total summary shall be made listing certain appropriate solution diagnostics, tolerances, total iterations, if any, number of tap changes, number of area interchange operations, total losses, total generation, total load, total number of buses, total number of branches.

12. Over and under-voltage buses shall be summarized.

13. Overloaded branches shall be summarized.

14. Provision shall be made to list all input data alphabetically by interchange area.

15. Date and case number shall appear on each page.

16. Unlimited title remarks shall be permitted at beginning of case.

17. All summaries shall list data in alphabetic order within interchange area grouping. All summaries should precede bus and branch output data.

18. A limited output option shall be provided in which specified areas or individual specified items are included.

I. Map Print Out

A load flow diagrammatic output provision is to be included in the program with the following features:

1. Items to be available for call-out shall include bus voltage and angle, mw and mvar of

any branch, machine, load, or shunt device, transformer LTC or phase shiftier angle.

2. Location of readings shall have complete coordinate freedom and not be restricted to word length modules.

3. It shall not be necessary to call out readings in pairs if not desired, such as mw and mvar or voltage and angle.

4. Provision shall be made for rounding of readings to nearest specified unit.

J. Change Cases

1. Change cases shall be effected by entering changes only. Unchanged data shall not require repeating such as when only part of the data on a bus card is changed.

2. Provision shall be made to inactivate certain portions of the total system in order to study one portion in detail. The inactivation shall be by a short-cut method such as by simply opening certain tie lines rather than by entering deletions for all buses and branches to be inactivated.

3. A provision shall be made for changing load magnitude by scheduled percentage. It shall be possible to have separate schedules for watts and vars and separate schedules for each interchange area. Provision shall also be made to omit certain specified buses from the old change.

K. Associated Subroutine Options

1. Provision shall be made to form mathematical equivalents for areas not required in detail

Part III: Stability Program

A. System Limits

1. 2000 buses from the load flow plus those necessary for machine representation.
2. 3000 branches from the load flow plus those necessary for machine representation.
3. 600 machines.
4. 10 two-terminal DC lines.
5. 5 three-terminal DC lines.
6. Unlimited switching of branches, loads, generation and faults.
7. Transformers and phase shifters from the load flow.
8. 200 line relays.
9. 100 series capacitor gaps.

B. Load Representation

1. Each bus load (mw & mvar) shall be representable as any desired combination of constant current, constant mva, constant impedance and frequency sensitivity.
2. Load shedding at any 500 buses in at least five frequency and five voltage or frequency step and the voltage and frequency step size shall be an input option variable at each load shedding bus.
3. Constant power loads near fault points shall be adjusted during fault periods to prevent

solution failure.

C. Machine Representation

1. SYnchronous Machines

Each synchronous machine may be representable in any one of the following degrees of detail at the option of the user:

(a) Constant voltage behind reactance, with or without inertia.

(b)Two-axis saliency with or without main-field saturation, i.e., with $X'_{d'}, X'_{q'}, X_{d'}, X_{q'}, T'_{do}$ and inertia.

(c) Up to 10 machines may be represented as having wow-axis saliency with or without main-field saturation, and amortisseur damping and using the following differential equations to determine synchronizing and damping electrical torques directly:

Direct and quadrature axis components of terminal voltage,

$$V_d = \frac{d\lambda_d}{dt} - w\lambda_q - i_d r_a$$

$$V_q = \frac{d\lambda_q}{dt} + w\lambda_d - i_q r_a$$ where $\lambda_{d'}, \lambda_{q'}$ and r_a are armature quantities.

Field Voltage,

$$V_f = \frac{d\lambda_f}{dt} - i_f r_f$$ where λ_f, i_f and r_f are field quantities.

Amortisseur damping.

$$\frac{d\lambda_q}{dt} = i_q r$$

$$\frac{d\lambda_d}{dt} = i_d r$$ where λ_q, λ_q and r are amortisseur quantities.

(d) Shaft damping shall be representable as $K\frac{d}{dt}$, where K is the shaft damping coefficient.

2. Provision shall be included for the accurate representation of induction motors and mechanical load and inertia.

3. Excitation Systems

(a) Each synchronous machine shall be represented with constant field voltage or with an automatic excitation system as specified by input.

(b) The excitation system models to be used shall be formed by digitally simulating a desired analog block diagram for each type of regulator-exciter system to be represented. The number and make-up of the block diagrams shall be easily changeable to provide for the addition of new systems and modification of block diagrams. Initial models shall include those of IEEE Paper No. 31T67-424 except that continuously-acting regulator blocks shall provide for two time lag functions. Also, supplemental control from the sum of two designated quantities through up to four lead-lag functions and a limiter shall be provided for continuously-acting systems.

4. Prime Mover Systems

All governor systems shall be representable in the following degrees of detail at the option of the user:

(a) No governor, i.e., constant power input.

(b) Hydro governor with control, reset, serve and water hammer transfer functions.

(c) Steam-turbine governor with control, serve, valve and transfer functions for tandem, tandem-compound, and cross-compound reheat and non-reheat turbines.

(d) Boiler control systems.

(e) The governor representation shall include provision for fast turbine valving to allow steam-valve manipulation as defined by the user.

(f) The governor representation shall include supplemental control similar in from to the supplemental excitation control described in 3 (b).

D. Direct Current Line Representation

1. Operating modes

(a) Constant power

(b) COnstant current

(c) Power modulation by function of AC system angles

2. Terminal control equipment shall be simulated accurately for dynamic conditions. Approximate representation also shall be available.

E. Automatic System Actions

1. Tie-line frequency biased "Load Control" shall be available for 60 control areas with access to any machines designated.

2. Automatic line tripping on 200 lines with or without single shot reclusion shall be provided utilizing the following optional relay systems:

(a) Impedance characteristic with circle passing through origin, with or without swing blocking zones.

(b) Directional time-overcurrent characteristic.

(c) Out-of step characteristic`

(d) Over-power with time delay characteristic.

(e) Power biased rate-of-change of load flow characteristic.

3. On each line having one of the above relay systems, it shall be possible to initiate a switching operation such as a transfer trip to another location.

4. On each of 20 lines having the above relay systems, it shall be possible to initiate transfer trip operations to 5 other lines, loads or generators.

5. It shall be possible to simulate series capacitor gap operation at 100 locations. Gap operation and capacitor re-insertion each shall be controllable as a function of current.

6. Provision shall be made for generator braking resistors at 20 locations. Control of the braking resistor shall be based upon machine accelerating power or frequency deviation, at the option of the user. It shall be possible to apply the resistor in steps, based upon increments of the controlling factor. Time delay in application and length of application time shall be designated by user.

F. Output

1. Automatic plotting data for at least 100 axes containing up to 15 curves per axis for the following variables shall be available: Selected items shall be available for on-line machine plotting.

Bus voltage and angle
Bus frequency deviation Hertz
Field voltage
Field current
Generator Internal Voltage, e'q
Main-field saturation
Generator mechanical power
Generator electrical power
Quadrature-axis angle
Damping torque
Generator Rotor Slip, Hertz
Machine var output
Summation of P & Q for any number of specified loads, generators, and lines in selected areas
Excitation or governor supplemental control signal output
Regulator output
Governor torque change
Accelerating power
Apparent line impedance
AC line load flow mw & mvar
Relative generator angle
Tie line frequency control
DC line load flow
DC line firing angle
DC line current
Rate of change of power on branches
Exciter saturation

2. Printed output shall include all variables listed in F.1. above for each calculation time interval, or any selected integral value of it, with no restrictive limits on the number of

variables chosen for output. Output form of AC quantities shall be mw and mvar on flows, polar form on currents, validates and apparent impedances. Provision shall be made to tabulate from the output tape any variable or variables as functions of time.

3. All automatic operations shall be summarized in the printed output.

G. Program Operation

1. The program shall be capable of accepting either punched-card or magnetic-tape input at the option of the user. The load-flow history-tape also shall be compatible as input to the stability program.
2. The stability program logic shall check for zero-minus balance errors and print diagnostics.
3. Results of stability calculations shall be stored on magnetic tape in such form that it may be used for later analysis.
4. The calculation time interval for solution of control systems can be different from the time step used in solution of the swing equation.
5. Provisions shall be included to output system conditions at pre-selected time intervals on magnetic tape for future continuation of the study.
6. At users option, loss of synchronism of individual machines shall not automatically terminate a run. Such machines should be shed according to applicable relaying techniques or by area separation and the run continued if desired.