OPTIMAL DISPATCH OF POWER GENERATION SOFTWARE PACKAGE USING MATLAB

MUHAMAD FIRDAUS BIN RAMLI

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: **OPTIMAL DISPATCH OF POWER GENERATION SOFTWARE PACKAGE USING MATLAB**

SESI PENGAJIAN: **2007/2008**

Saya **MUHAMAD FIRDAUS BIN RAMLI (850923-07-5467)**
(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

[ ]	**SULIT**	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
[ ]	**TERHAD**	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
[√]	**TIDAK TERHAD**	

Disahkan oleh:

(TANDATANGAN PENULIS)

(TANDATANGAN PENYELIA)

Alamat Tetap:

6273 MUKIM 5 KUBANG SEMANG,
14400 BUKIT MERTAJAM,
PULAU PINANG.

MRS. NORHAFIDZAH BINTI MOHD SAAD
(Nama Penyelia)

Tarikh: **30 NOVEMBER 2007**

Tarikh: : **30 NOVEMBER 2007**

CATATAN:
* Potong yang tidak berkenaan.
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

"I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor Degree of Electrical Engineering (Power Systems)"

Signature: ______________________________

Name: NORHAFIDZAH BINTI MOHD SAAD

Date: 30 NOVEMBER 2007

OPTIMAL DISPATCH OF POWER GENERATION SOFTWARE PACKAGE USING MATLAB

MUHAMAD FIRDAUS BIN RAMLI

This thesis is submitted as partial fulfillment of the requirements for the award of the Bachelor Degree of Electrical Engineering (Power Systems)

Faculty of Electrical & Electronics Engineering

Universiti Malaysia Pahang

NOVEMBER 2007

"All the trademark and copyrights use here in are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author."

Signature : ________________________________

Author : MUHAMAD FIRDAUS BIN RAMLI

Date : 30 NOVEMBER 2007

Specially dedicated to

my beloved family and those people who have guided and inspired me

throughout my journey of education.

ACKNOWLEDGEMENT

First and foremost, I am very grateful to the almighty ALLAH S.W.T for giving me this opportunity to accomplish my Final Year Project.

In preparing this project and thesis I was in contact with many people, they have contributed toward my understanding and thoughts. In particular, I wish to express my sincere appreciation to my project supervisor, Mrs. Norhafidzah binti Mohd Saad, for encouragement, guidance, motivation and friendship. Without her continued support and interest, this project and thesis would not have been same as presented here.

My fellow friends should also be recognized for their support. My sincere appreciation also extends to all lecturer and others who had provided assistance at various occasions. Their views and tips are useful indeed. Unfortunately, it is not possible to list all of them in this limited space.

Most importantly I extend my gratitude to my parents who have encouraged me throughout my education and I will always be grateful for their sacrifice, generosity and love.

ABSTRACT

In the reality practical power system, power plants are not at the same distance from the centre of load and their fuel costs are different. Also, under normal operating condition the generation capacity is more than the total load demand and losses. Thus there is one main option for scheduling generation that is called optimal dispatch. Optimal dispatch of power generation is to find an effective real and reactive power scheduling to power plants to meet load demand as well as to minimize the operating cost. This cost function may present economic cost, system security and others but in this project, the analysis will limited to the economic dispatch of real power generation. This economic dispatch analysis has been studied by many researchers using different method. However this analysis is very difficult and takes much time to be done by hand calculations. The existence of this optimal dispatch of power generation software package will help the consumer to make this analysis done easier. This friendly software package will be a good medium for researcher to obtain optimal dispatch of power generation without really much effort on hand calculation.

ABSTRAK

Di dalam situasi sebenar sistem kuasa, semua penjana selalunya tidak berada pada jarak yang sama dengan titik tengah beban dan kos bahan bakarnya berbeza antara satu sama lain. Tambahan pula, jika ia berada di bawah tahap sebenar operasi jumlah penjanaan adalah berlebihan daripada jumlah permintaan beban. Jadi untuk memastikan semua penjana dapat menjana kuasa yang mencukupi analisis sistem kuasa digunakan. Analisis sistem kuasa adalah bertujuan untuk mendapatkan nilai kuasa yang sepatutnya dijanakan oleh penjana bagi mencapai nilai permintaan kuasa dan dalam masa yang sama mengurangkan kos operasi. Kos operasi ini boleh di kategorikan sebagai kos ekonomi, sistem keselamatan dan lain-lain tetapi didalam projek ini, analisis hanya dihadkan kepada kos ekonomi dalam penjanaan kuasa. Analisis kos ekonomi ini telah dikaji oleh ramai pengkaji dengan menggunakan pelbagai kaedah. Walau bagaimanapun analisis ini amat sukar dan mengambil masa yang panjang untuk disiapkan melalui pengiraan secara insani. Dengan wujudnya perisian Optimal Dispatch of Power Generation ini diharapkan ia dapat membantu pengguna khususnya pengkaji menjalankan analisis ini dengan mudah. Perisian ini akan menjadi sumber yang baik untuk pengkaji menjalan analisis system kuasa tanpa perlu memberikan perhatian yang keterlaluan kepada pengiraan insani.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	TITLE PAGE	i
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGEMENT	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENTS	vii
	LIST OF TABLES	x
	LIST OF FIGURES	xi

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
1	**INTRODUCTION**	
	1.1 Introduction	1
	1.2 Objective	2
	1.3 Scope of Project	2
2	**LITERATURE REVIEW**	
	2.1 Introduction	3
	2.2 Optimal Dispatch of Power Generation	3
	2.3 Problem Formulation	4
	2.3.1 Economic Dispatch Problem	4
	2.4 MATLAB GUI	11
	2.4.1 A Brief Introduction of GUIDE	12
	2.4.2 Two Basic Tasks in Process of Implementing a GUI	12
	2.5 The Similar Software in Market	13
3	**METHODOLOGY**	
	3.1 Introduction	15
	3.2 Flow Chart of Project	16
	3.3 Problem Simulation	17
	3.3.1 5-Bus Power System Network	17
	3.3.2 26-Bus Power System Network	20
	3.5 Develop MATLAB GUI Using MATLAB GUIDE	25

4 RESULT AND DISCUSSION

4.1 Introduction 29
4.2 Optimal Dispatch of Power Generation Software Package 29
4.2.1 Introduction 29
4.2.2 Detail of Software Package 31
4.2.3 Formula of Power Flow Input Data 43
4.2.3.1 Line Data Input Format 43
4.2.3.2 Bus Data Input Format 44
4.3 Optimal Dispatch Simulation Result Using MATLAB 44
4.2.1 Result for One-Line Diagram of 5-Busbar 45
4.2.1.1 Newton-Rahpson Method 45
4.2.1.2 Gauss-Seidel Method 48
4.2.1.3 Fast-Decouple Method 51
4.2.2 Result for One-Line Diagram of 26-Busbar 54
4.2.2.1 Newton-Raphson Method 54
4.2.2.2 Gauss-Seidel Method 59
4.2.2.3 Fast-Decouple Method 64
4.2.3 Discussion & Analysis of Simulation 69
4.2.3.1 Result for 5-Busbar System 69
4.2.3.2 Result for 26-Busbar System 70

5 CONCLUSION AND RECOMMENDATION

5.1 Conclusion 71
5.2 Future Recommendations 72
5.2.1 Commercialization 72

REFERENCES 73

Appendices A-C 74-82

LIST OF TABLES

TABLE NO.	TITLE	PAGE
3.1	Generation Data for 5-Bus Power System Network	18
3.2	Generation Power Limit for 5-Bus Power System Network	18
3.3	Load Data for 5-Bus Power System Network	19
3.4	Line and Transformer Data for 5-Bus Power System Network	19
3.5	Generation Data for 26-Bus Power System Network	21
3.6	Transformer Data for 26-Bus Power System Network	21
3.7	Shunt Capacitor Data for 26-Bus Power System Network	22
3.8	Generation Power Limit for 26-Bus Power System Network	22
3.9	Load Data for 26-Bus Power System Network	23
3.10	Line and Transformer Data	23
3.11	Basic MATLAB GUI Component	26
4.1	Line Data Format	43
4.2	Bus Data Format	44
4.3	Power Flow Solution by Newton-Raphson Method for 5-Bus Power System Network	45
4.4	Result from Gradient Method for 5-Bus Power System Network	46
4.5	New Power Flow Solution by Newton-Raphson Method for 5-Bus Power System Network	47

4.6	Power Flow Solution by Gauss Seidel Method for 5-Bus Power System Network	48
4.7	Result from Gradient Method for 5-Bus Power System Network	49
4.8	New Power Flow Solution by Gauss Seidel Method for 5-Bus Power System Network	50
4.9	Power Flow Solution by Fast Decouple Method for 5-Bus Power System Network	51
4.10	Result from Gradient Method for 5-Bus Power System Network	52
4.11	New Power Flow Solution by Fast-Decouple Method for 5-Bus Power System Network	53
4.12	Power Flow Solution by Newton-Raphson Method for 26-Bus Power System Network	54
4.13	Result from Gradient Method for 26-Bus Power System Network	56
4.14	New Power Flow Solution by Newton-Raphson Method for 26-Bus Power System Network	57
4.15	Power Flow Solution by Gauss Seidel Method for 26-Bus Power System Network	59
4.16	Result from Gradient Method for 26-Bus Power System Network	61
4.17	New Power Flow Solution by Gauss Seidel Method for 26-Bus Power System Network	62
4.18	Power Flow Solution by Fast Decouple Method for 26-Bus Power System Network	64
4.19	Result from Gradient Method for 26-Bus Power System Network	66
4.20	New Power Flow Solution by Fast-Decouple Method for 26-Bus Power System Network	67
4.21	Result from simulation using MATLAB for 5-busbar system with 3 generators	69
4.22	Result from simulation using MATLAB for 26-busbar system with 6 generators	70

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
2.1	Result of Economic Dispatch Analysis	13
3.1	Flowcharts of Project	16
3.2	One-line Diagram of 5-Bus Power System	17
3.3	One-line Diagram of 26-Bus Power System	20
3.4	MATLAB GUIDE Layouts	24
3.5	Property Inspector	25
3.6	Example of Completed GUI	26
3.7	Example of M.file	27
4.1	MATLAB Set Path	29
4.2	Main Page of Software Package	30
4.3	Optimal Dispatch Definition	31
4.4	Example and Tutorial	32
4.5	Data Preparation	33
4.6	Problem Neglecting Losses and No Generator Limit	33
4.7	Problem Neglecting Losses and Including Generator Limit	34
4.8	Problem Including Losses	34
4.9	Problem with Power Flow Solution	35
4.10	Input Data for Problem with Power Flow Solution	36
4.11	Tips on Input Data Preparation	37
4.12	Result Window for Optimal Dispatch with Power Flow Solution	38

4.13	Type of Analysis	39
4.14	Economic Dispatch Neglecting Losses and No Generator Limit	40
4.15	Economic Dispatch Neglecting Losses and Including Generator Limit	40
4.16	Economic Dispatch Including Losses	41
4.17	Economic Dispatch with Power Flow Solution	41
4.18	Credit	42

CHAPTER 1

INTRODUCTION

1.1 Introduction

In the practical power system, power plants are not at the same distance from the centre of load and their fuel costs are different. Also, under normal operating condition the generation capacity is more than the total load demand and losses. Thus there is one main option for scheduling generation that is called optimal dispatch. Optimal dispatch of power generation is to find an effective real and reactive power scheduling to power plants to meet load demand as well as minimize the operating cost. This cost function may present economic cost, system security and others but in this project, the analysis will limited to the economic dispatch of real power generation [5].

This economic dispatch analysis has been studied by many researchers using different method.However this analysis is very difficult and takes much time to be done by hand calculations. The existence of this optimal dispatch of power generation software package will help the users to obtain optimal dispatch of power generation analysis done easier without really much effort on hand calculation [5].

1.2 Objective

The objective of this project is:

i. To study and analyze the real and reactive power scheduling of each power plant in such way as to minimize the operating cost of power generation.
ii. To obtain simulation on optimal dispatch of power generation using MATLAB.
iii. Build a user friendly software package using MATLAB GUI to analyze optimal power flow problem.

1.3 Scope of Project

In this project, there are several scopes that the author needs to cover:

i. Study and analyze the best widely used method between Newton-Raphson, Fast Decouple and Gauss Seidel method to obtain the optimal dispatch power generation.
ii. Simulation and analysis all the methods for Optimal Dispatch in MATLAB. This simulation and analysis had categories by two phases; the first one is based on figure of one line diagram of 5-bus and 26-bus power system with generator. Sample of power system diagram from IEEE will be done in second phase.
iii. Simulation using MATLAB GUI and this stage will be classified to two phases. Development of the GUI gone in two phase, the first phase cover on designing the lay out of GUI and second phase will cover MATLAB GUI programming.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Literature reviews are very important as a reference for making this software package as good as possible. The author has studied many journals and article that had been done by previous researcher especially from IEEE.

2.2 Optimal Dispatch of Power Generation

Optimal dispatch is the operation of generation facilities to produce energy at the lowest cost to reliability serve consumer, recognizing any operational limit of generation and transmission facilities [1]. The power balance constraint for power system demand, transmission loss and total generating power as well as the generating power constraints for all units should be satisfied [6]. This optimization can be done by selected objective functions or cost function while maintaining an acceptable system performance in terms of generator capability limits and the output of compensating device. This cost function may present economic costs, system security, safety and others [5]. In this project, analysis will be limited in economic dispatch or optimal dispatch.

2.3 Problem Formulation

2.3.1 Economic Dispatch Problem [5]

Transmission losses are the major factor and affect the optimum dispatch of generation because usually in a large interconnected network, power have to transmit over long distance with low density areas. One common practice for including the effect of transmission losses is to express the total transmission loss as quadratic function of generator power outputs. The simplest quadratic form is

$$PL - \sum_{i=1}^{ng} . \sum_{j=1}^{ng} \text{Pi Bij Pj} \tag{1}$$

A more general formula containing a linear term and a constant term is Kron's loss formula

$$PL = \sum_{i=1}^{ng} . \sum_{j=1}^{ng} \text{Pi Bij Pj} + \sum_{i=1}^{ng} \text{B0i Pi} + \text{B00} \tag{2}$$

To minimize the overall generating cost Ci, which is the function of plant output

$$Ct = \sum_{i=1}^{ng} \alpha i + \beta iPi + \gamma iPi^2 \tag{3}$$

subject to the constraint the generation should equal total demand plus losses

$$\sum_{i=1}^{ng} Pi = PD + PL \tag{4}$$

satisfying the inequality constraints, expressed as follows;

$$P_{i\,(min)} \leq P_i \leq P_{i\,(max)} = 1, \ldots\ldots, n_g \qquad i = 1, \ldots., n_g \tag{5}$$

Where,

i : Index of dispatchable units

αi, βi, γi : Cost coefficients of units i

Pi : The generated power of unit i

PL : Transmission line losses

B0i, Bij, B00 : Transmission line coefficients

PD : Total load demand

n : Number of all dispatched units

$Pi_{(min)}$: Minimum generation limits of units i

$Pi_{(max)}$: Maximum generation limits of units i

Using the Lagrange multiplier add adding additional terms to include the equality constraints, we obtain

$$\mathcal{L} = Ct + \lambda(PD + PL - \sum_{i=1}^{ng} Pi) + \sum_{i=1}^{ng} \mu i(max)\,(Pi - Pi(\mathrm{max})) + \sum_{i=1}^{ng} \mu i(min)\,(-Pi(\mathrm{min})) \tag{6}$$

The constraints mean

$\mu i(max) = 0$ when $Pi < Pi(max)$

$\mu i(min) = 0$ when $Pi > Pi(min)$

The minimum of unconstrained function is found at the point where the partials of the function to its variables are zero.

$$\frac{\partial \mathcal{L}}{\partial Pi} = 0 \tag{7}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \tag{8}$$

$$\frac{\partial \mathcal{L}}{\partial \mu i\,(\max)} = Pi - Pi(\max) = 0 \tag{9}$$

$$\frac{\partial \mathcal{L}}{\partial \mu i\,(\min)} = Pi - Pi(\min) = 0 \tag{10}$$

Equation (9) and (10) imply that Pi should not be allowed to go beyond its limit, and when Pi is within its limits $\mu i(min) = \mu i(max) = 0$ and Kuhn-Tucker function becomes the same as the Lagrangian one. First condition, given by (7), results in

$$\frac{\partial Ct}{\partial Pi} + \lambda\left(0 + \frac{\partial PL}{\partial Pi} - 1\right) = 0$$

Since

$$Ct = C1 + C2 + ... + Cng$$

then

$$\frac{\partial \mathcal{L}}{\partial Pi} = \frac{\partial Ci}{\partial Pi}$$

and the condition for optimum dispatch is

$$\frac{\partial Ci}{\partial Pi} + \lambda\frac{\partial PL}{\partial Pi} - \lambda \qquad i = 1,\ldots\ldots\ldots,n_g \tag{11}$$

The term $\frac{\partial PL}{\partial Pi}$ is knows as the incremental transmission loss. Second condition, given by (8), results in

$$\sum_{i=1}^{ng} Pi = PD + PL \qquad (12)$$

Equation (12) is precisely the equality constraint that was to be imposed.

Equation (11) is rearranged as

$$\left(\frac{1}{1-\frac{\partial PL}{\partial Pi}}\right)\frac{dCi}{dPi} = \lambda \qquad i = 1,\ldots\ldots,n_g \qquad (13)$$

or

$$Li\frac{dCi}{dPi} = \lambda \qquad i = 1,\ldots,n_g \qquad (14)$$

where Li is knows as the *penalty factor* of plant I and is given by

$$Li = \frac{1}{1-\frac{\partial PL}{\partial Pi}} \qquad (15)$$

The effect of transmission loss is to introduce a penalty factor with a value that depends on the location of the plant. Equation (14) shows that the minimum cost is obtained when the incremental cost of each power plant multiplied by its penalty factor is same for all plants.

The incremental production cost is given by

$$\frac{dCi}{dPi} = 2\gamma iPi + \beta i \qquad (16)$$

The incremental production cost (16), and the incremental transmission loss is obtained from the loss formula (2) which yields

$$\frac{\partial PL}{\partial Pi} = 2\sum_{j=1}^{ng} BijPj + B0i \qquad (17)$$

Substituting the expression for the incremental production cost and the incremental transmission loss in (11) results in

$$\beta i + 2\gamma iPi + 2\lambda \sum_{j=1}^{ng} BijPj + B0i\lambda = \lambda$$

or

$$\left(\frac{\gamma i}{\lambda} + Bii\right)Pi + \sum_{\substack{j=1 \\ j\neq i}}^{ng} BijPj + \frac{1}{2}\left(1 - B0i - \frac{\beta i}{\lambda}\right) \qquad (18)$$

Extending (18) to all plants results in matrix form

$$\begin{bmatrix} \frac{\gamma 1}{\lambda} + B11 & B12 & \cdots & B1ng \\ B21 & \frac{\gamma 2}{\lambda} + B22 & \ldots & B2ng \\ \vdots & \vdots & \ddots & \vdots \\ Bng1 & Bng2 & \cdots & \frac{\gamma ng}{\lambda} + Bngng \end{bmatrix} \begin{bmatrix} P1 \\ P2 \\ \vdots \\ Png \end{bmatrix} = \frac{1}{2} \begin{pmatrix} 1 - B01 - \frac{\beta 1}{\lambda} \\ 1 - B02 - \frac{\beta 2}{\lambda} \\ \vdots \\ 1 - B0ng - \frac{\beta ng}{\lambda} \end{pmatrix} \qquad (19)$$

or in short form

$$EP = D \qquad (20)$$

To find the optimal dispatch for estimated value of $\lambda^{(1)}$, the simultaneous linear equation given by (20) is solved.

Then, we continue the iterative process using the gradient method. To do this, from (18), Pi at the k^{th} iteration is expressed as

$$Pi^{(k)} = \frac{(1 - B0i) - \beta i - 2\lambda^{(k)} \sum_{j \neq i} BijPj^{(k)}}{2(\gamma i + \lambda^{(k)} Bii} \tag{21}$$

Substituting for Pi from (21) in (12) results in

$$\sum_{i=1}^{ng} . \frac{\lambda^{(k)}(1 - B0i) - \beta i - 2\lambda^{(k)} \sum_{j \neq i} BijPj^{(k)}}{2(\gamma i + \lambda^{(k)} Bii} = PD + PL^{(k)} \tag{22}$$

or

$$f(\lambda)^{(k)} = PD + PL^{(k)} \tag{23}$$

Expending the left-hand side of the above equation in Taylor's series about an operating point $\lambda^{(k)}$, and neglecting the higher-order terms results in

$$f(\lambda)^{(k)} + \left(\frac{df\lambda}{d\lambda}\right).^{(k)} \Delta\lambda^{(k)} = PD + PL^{(k)} \tag{24}$$

or

$$\Delta\lambda^{(k)} = \frac{\Delta P^{(k)}}{\left(\frac{df(\lambda)}{d\lambda}\right)^{(k)}}$$

$$= \frac{\Delta P^{(k)}}{\Sigma\left(\frac{dPi}{d\lambda}\right)^{(k)}} \tag{25}$$

where

$$\sum_{i=1}^{ng} \left(\frac{\partial Pi}{\partial \lambda}\right)^{(k)} = \sum_{i=1}^{ng} \frac{\gamma i(1 - B0i) + Bii\beta i - 2\gamma \sum_{j \neq i} BijPj^{(k)}}{2(\gamma i + \lambda^{(k)} Bii)^2} \tag{26}$$

therefore,

$$\lambda^{(k+1)} = \lambda^{(k)} + \Delta\lambda^{(k)} \qquad (27)$$

where

$$\Delta P^{(k)} = PD + PL^{(k)} - \sum_{i=1}^{ng} Pi^{(k)} \qquad (28)$$

The process is continued until $\Delta P^{(k)}$ is less than a specified accuracy.

If an appropriate loss formula expressed by

$$PL = \sum_{i=1}^{ng} BiiPi^{2} \qquad (30)$$

is used, Bij = 0, B00 = 0, and the solution of the simultaneous equation given by (21) reduces to the following simple expression

$$Pi^{(k)} = \frac{\lambda^{(k)} - \beta i}{2(\gamma i + \lambda^{(k)} Bii)} \qquad (31)$$

and (28) reduces to

$$\sum_{i=1}^{ng} \frac{\partial y^{(k)}}{\partial x} = \sum_{i=1}^{ng} \frac{\lambda^{(k)} - \beta i}{2(\gamma i + \lambda^{(k)} Bii)^{2}} \qquad (32)$$

2.4 MATLAB GUI

A graphical user interface (GUI) is a graphical display that contains devices, or components, that enable a user to perform interactive tasks. A good GUI can make programs easier to use by providing them with a friendly appearance and with controls icon like pushbuttons, list boxes, sliders, menus, radio button and so forth (refer to appendix A1). To perform these tasks, the user of the GUI does not have to create a script or type commands at the command line. Often, the user does not have to know the details of the task at hand. The GUI should behave in an understandable and predictable manner, so that a user knows what to expect when he or she performs an action. For example, when a mouse click occurs on a pushbutton, the GUI should initiate the action described on the label of the button [2] [3].

Each component, and the GUI itself, are associated with one or more user-written routines known as callbacks. The execution of each callback is triggered by a particular user action such as a button push, mouse click, selection of a menu item, or the cursor passing over a component. The creator of the GUI will provide these callbacks. MATLAB enables the user to create GUIs programmatically or with GUIDE, an interactive GUI builder. It also provides functions that simplify the creation of standard dialog boxes. The technique had chosen depends on the creator experience, preferences, and the kind of GUI that want to create [2] [3].

GUIDE, the MATLAB graphical user interface development environment, provides a set of tools for creating graphical user interfaces (GUIs). These tools simplify the process of laying out and programming GUIs [2] [3].

2.4.1 A Brief Introduction of GUIDE

GUIDE, the MATLAB graphical user interface development environment, provides a set of tools for creating graphical user interfaces (GUIs). These tools simplify the process of laying out and programming GUIs [1].

- GUIDE is primarily a set of layout tools
- GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI
  - This M-file also provides a framework for the implementation of the callbacks - the functions that execute when users activate a component in the GUI [1].

2.4.2 Two Basic Tasks in Process of Implementing a GUI

The two basic task in Process of implementing a GUI is first, laying out a GUI where MATLAB implement GUIs as figure windows containing various styles of uicontrol (User Interface) objects. The second task is programming the GUI, where each object must be program to perform the intended action when activated by the user of GUI.

2.5 The Similar Software in Market

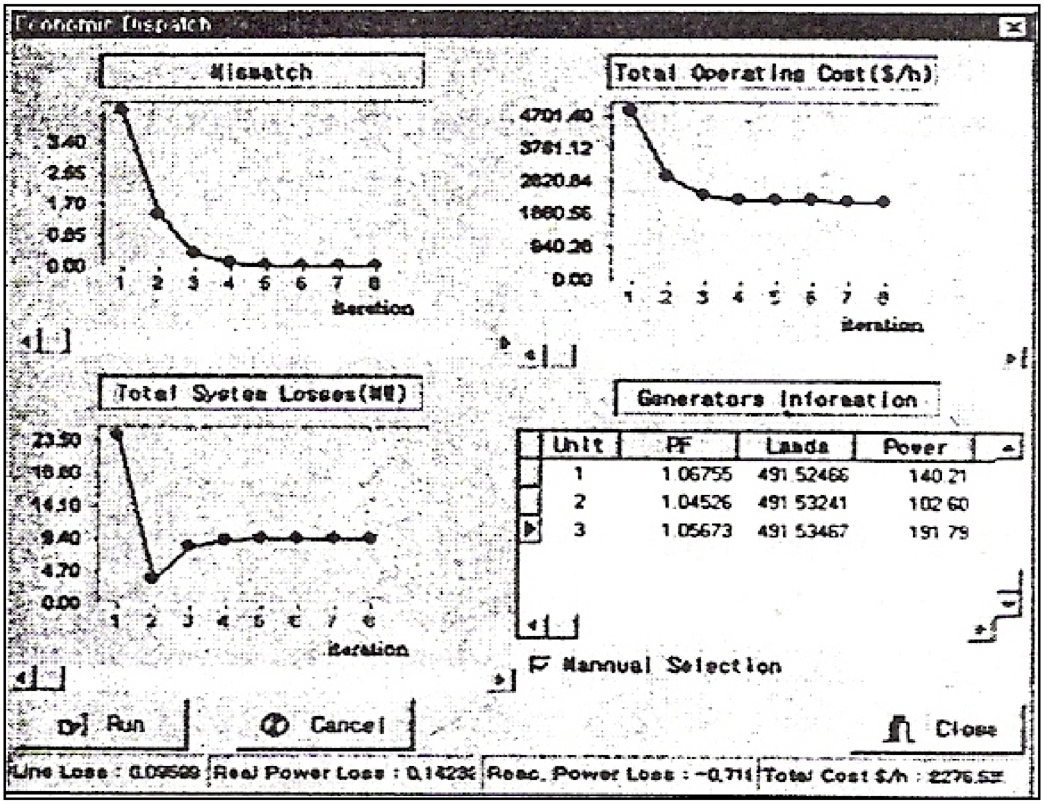


Figure 2.1: Result of Economic Dispatch Analysis.

Figure 2.1 is a software package named Windows-based Interactive and Graphic Package for the Education and Training of Power System Analysis and Operation. This software package is developed by Joong-Rin Shin, Wook-Hwa Lee & Dong-Hae Im from Kon-Kuk University, Seoul, Korea. This software package is developed by GUI and VDBMS using Borland C++.

The application programs in this package include the Power Flow (PF) calculation, the Transient Stability Analysis (TSA), the Fault Analysis (FA), the Economic Dispatch (ED), and the Automatic Load-Frequency Control (ALFC). This application software is designed as independent modules. Each module has a separate graphical and interactive interfacing window. In addition, the user can easily switch from one application module to another.

For the ED problem, this software will show the power mismatch, total operating cost, total system lost and generator information. For the input, the graphic editor has been specially designed to visually edit the one-line diagram of the power system with dialog box on the window [7].

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter presents the methodology of this project. It describes on how the project is organized and the flow of the steps in order to complete this project. The methodology is diverged in two parts, which is the simulation and analysis of Optimal Dispatch in MATLAB and the other is developing the layout and programming for GUI MATLAB.

There are three mains step for software development of this project. Before the project is developing using MATLAB, it is needed to study the method of Optimal Dispatch of Power Generation analysis and how MATLAB GUIDE work. The flowchart in Figure 3.1 illustrated the sequence of steps for this project. The first step is to study about Optimal Dispatch analysis and MATLAB. The second step is to develop the suitable formula of each type of Optimal Dispatch and running the simulation in MATLAB. The last step is developing GUI in MATLAB and programs every GUI component to make sure the software package as friendly as possible to the user.

3.2 Flow Chart of Project

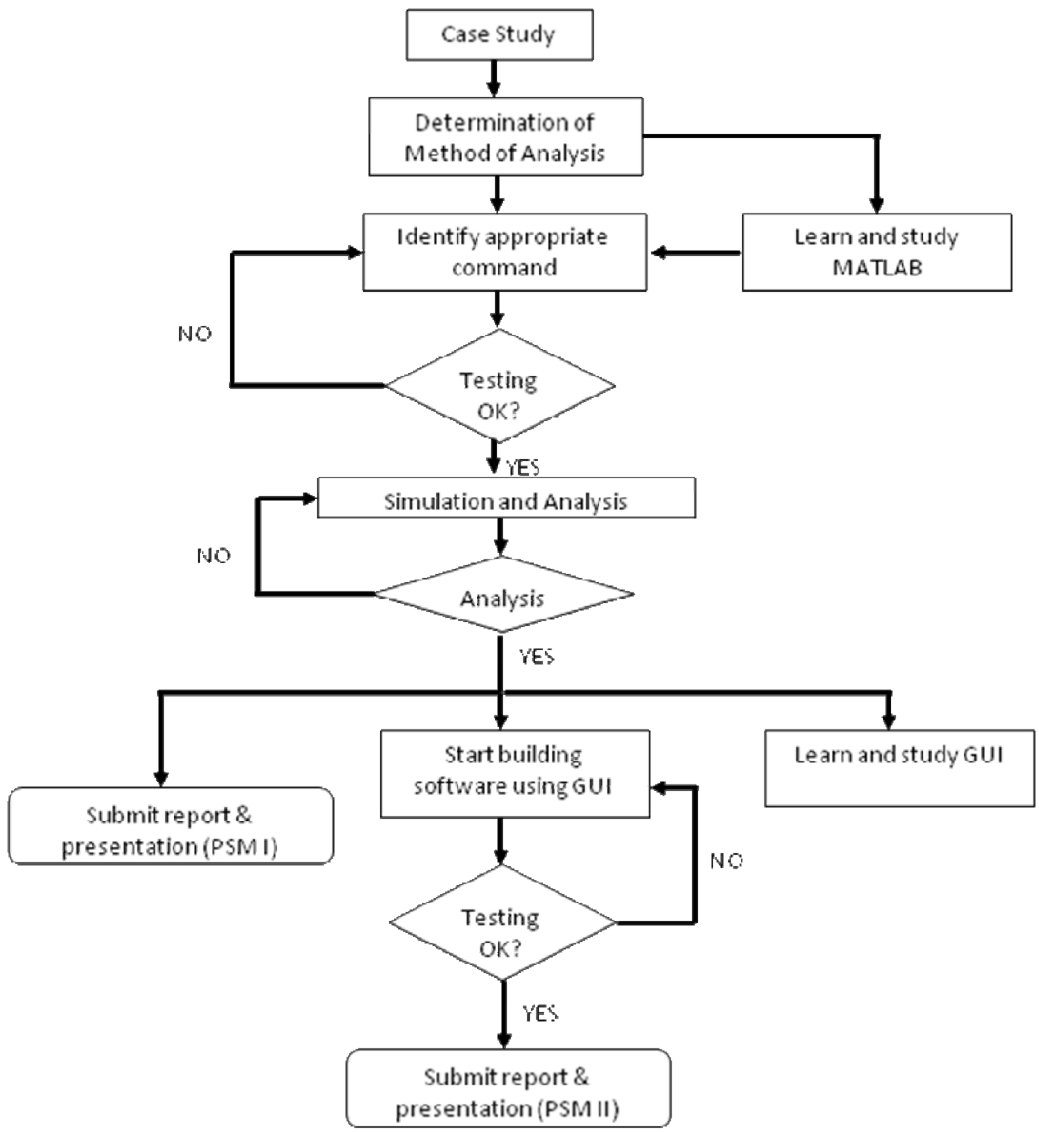


Figure 3.1: Flow Chart of Project

3.3 Problem Simulation

For simulation and analysis using MATLAB, the author has done the simulation for 5 and 26-bus power system network. The simulation had done by using 3 main methods to obtain power flow solution that is Newton-Raphson Method, Gauss-Siedel Method and Fast Decouple Method. The data of the 5 and 26 busbar power system network are described below:

3.3.1 5-Bus Power System Network

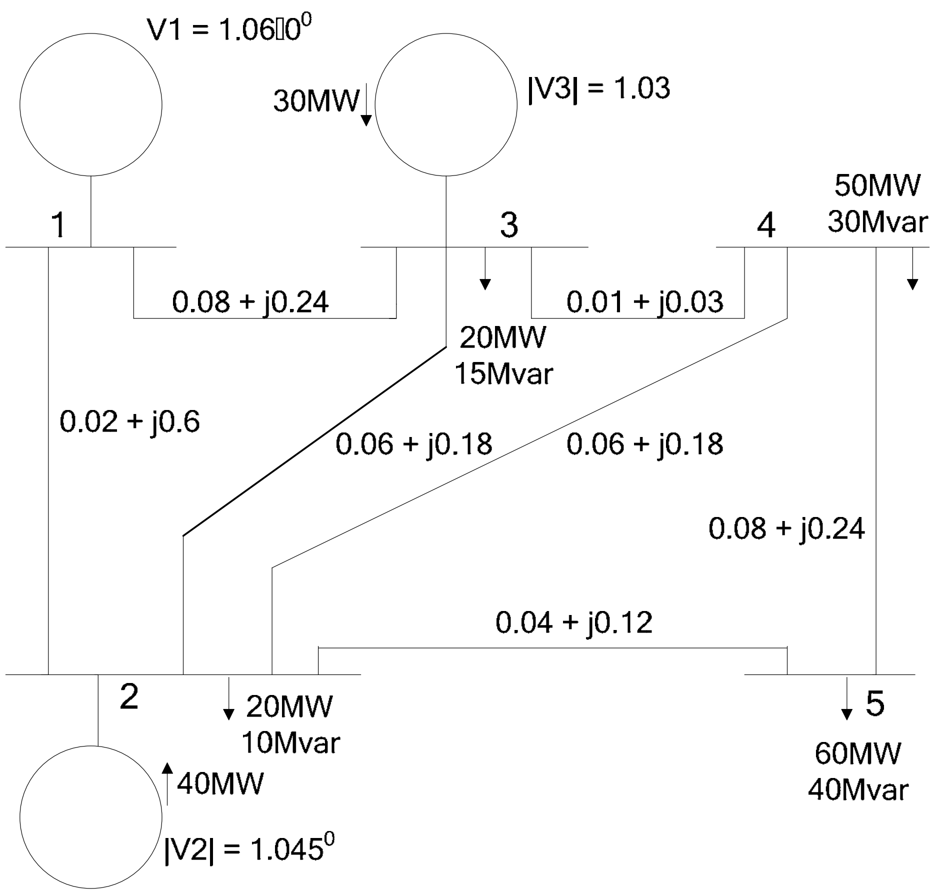


Figure 3.2: One-line diagram of 5-bus power system

Bus 1 taken as the slack bus with its voltage adjusted to $1.06\angle 0^0$

The cost function for P_1, P_2 and P_3 is as follows.

$C_1 = 200 + 7.0P_1 + 0.008P_1^2$

$C_2 = 180 + 6.3P_2 + 0.009P_2^2$

$C_3 = 140 + 6.8P_3 + 0.007P_3^2$

Voltage magnitude, generation schedule and the reactive power limits for the regulated buses are tabulated in Table 3.1.

Table 3.1

GENERATION DATA					
Bus No.	Voltage Magnitude	Generation MW	Generation Mvar	Min. Mvar Capacity	Max. Mvar Capacity
1	1.060	0	0	10	50
2	1.045	40	30	10	50
3	1.030	30	10	10	40
4	1.000	0	0	0	0
5	1.000	0	0	0	0

The generator's real power limits is shown in Table 3.2

Table 3.2

GENERATOR REAL POWER LIMITS		
Gen	Min. MW	Max. MW
1	10	85
2	10	80
3	10	70

The load data is shown in Table 3.3

Table 3.3

LOAD DATA		
Bus No.	Load	
	MW	Mvar
1	0	0
2	20.0	10.0
3	20.0	15.0
4	50.0	30.0
5	60.0	40.0

The line and transformer series resistance, reactance, transformer tap and one-half the total capacitive susceptance in per unit on a 100-MVA base are tabulated below.

Table 3.4

LINE AND TRANSFORMER DATA					
Bus No.	Bus No.	R, pu	X, pu	½ B, pu	Tap Setting pu
1	2	0.02	0.06	0.030	1
1	3	0.08	0.24	0.025	1
2	3	0.06	0.18	0.020	1
2	4	0.06	0.18	0.020	1
2	5	0.04	0.12	0.015	1
3	4	0.01	0.03	0.010	1
4	5	0.08	0.24	0.025	1

3.3.2 26-Bus Power System Network

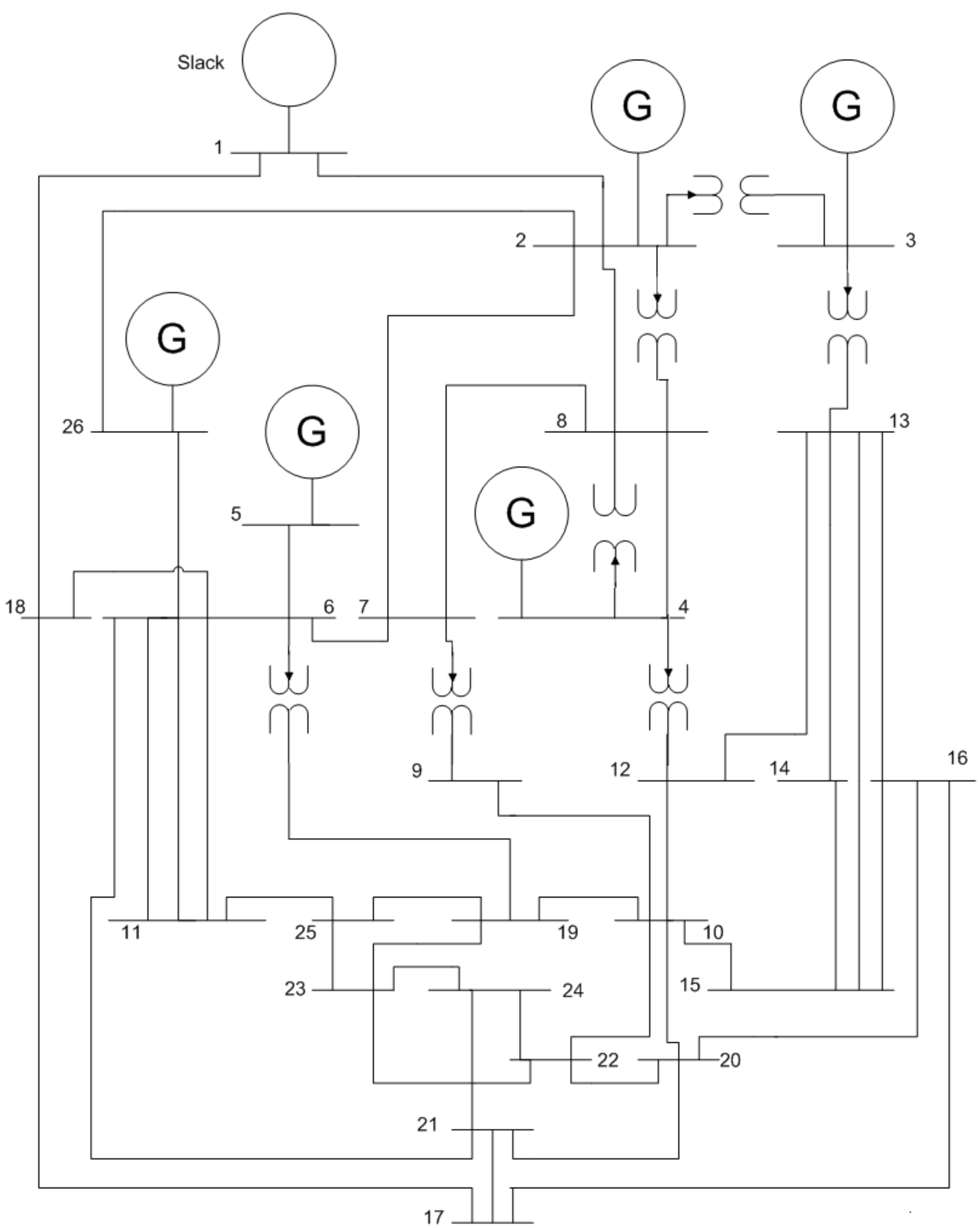


Figure 3.3: One-line diagram of 26-bus power system

Bus 1 taken as the slack bus with its voltage adjusted to $1.06\angle 0^0$

The cost function for P_1, P_2, P_3, P_4, P_5 and P_{26} is as follows.

$C_1 = 240 + 7.0_{P1} + 0.0070_{P1}{}^2$

$C_2 = 200 + 10.0_{P2} + 0.0095_{P2}{}^2$

$C_3 = 220 + 8.5_{P3} + 0.0090_{P3}{}^2$

$C_4 = 200 + 11.0_{P4} + 0.0090_{P4}{}^2$

$C_5 = 220 + 10.5_{P2} + 0.0080_{P5}{}^2$

$C_{26} = 190 + 12.0P_{26} + 0.0075P_{26}{}^2$

Voltage magnitude, generation schedule and the reactive power limits for the regulated buses are tabulated in Table 3.5.

Table 3.5

GENERATION DATA				
Bus No.	Voltage Magnitude	Generation MW	Min. Mvar Capacity	Max. Mvar Capacity
1	1.025			
2	1.020	79.0	40	250
3	1.025	20.0	40	150
4	1.050	100.0	40	80
5	1.045	300.0	40	160
26	1.015	60.0	15	50

Transformer tap settings are giver in the Table 3.6

Table 3.6

TRANSFORMER DATA	
Transformer Designation	Tap Setting Per Unit
2 – 3	0.960
2 – 13	0.960
3 – 13	1.017

4 – 8	1.050
4 – 12	1.050
6 – 19	0.950
7 – 9	0.950

The shunt capacitive data is shown in Table 3.7

Table 3.7

SHUNT CAPACITOR DATA	
Bus No.	Mvar
1	4.0
4	2.0
5	5.0
6	2.0
9	3.0
11	1.5
12	2.0
15	0.5
19	5.0

The generator's real power limits is shown in Table 3.8

Table 3.8

GENERATOR REAL POWER LIMITS		
Gen	Min. MW	Max. MW
1	100	500
2	50	200
3	80	300
4	50	150

5	50	200
5	50	120

The load data is as shown in Table 3.9

Table 3.9

LOAD DATA					
Bus No.	Load MW	Load Mvar	Bus No.	Load MW	Load Mvar
1	51.0	41.0	14	24.0	12.0
2	22.0	15.0	15	70.0	31.0
3	64.0	50.0	16	55.0	27.0
4	25.0	10.0	17	78.0	38.0
5	50.0	30.0	18	153.0	67.0
6	76.0	29.0	19	75.0	15.0
7	0.0	0.0	20	48.0	27.0
8	0.0	0.0	21	46.0	23.0
9	89.0	50.0	22	45.0	22.0
10	0.0	0.0	23	25.0	12.0
11	25.0	15.0	24	54.0	27.0
12	89.0	48.0	25	28.0	13.0
13	31.0	15.0	26	40.0	20.0

The line and transformer series resistance, reactance, transformer tap and one-half the total capacitive susceptance in per unit on a 100-MVA base are tabulated in Table 3.10

.

Table 3.10

LINE AND TRANSFORMER DATA									
Bus No.	Bus No.	R, pu	X, Pu	½ B, pu	Bus No.	Bus No.	R, pu	X, pu	½ B, pu
1	2	0.0005	0.0048	0.0300	10	22	0.0069	0.0298	0.0005
1	18	0.0013	0.0110	0.0600	11	25	0.0960	0.2700	0.0010
2	3	0.0014	0.0513	0.0500	11	26	0.0165	0.0970	0.0004
2	7	0.0103	0.0586	0.0180	12	14	0.0327	0.0802	0.0000
2	8	0.0074	0.0321	0.0390	12	15	0.0180	0.0598	0.0000
2	13	0.0035	0.0967	0.0250	13	14	0.0046	0.0271	0.0001
2	26	0.0323	0.1967	0.0000	13	15	0.0116	0.0610	0.0000
3	13	0.0007	0.0054	0.0005	13	16	0.0179	0.0888	0.0001
4	8	0.0008	0.0240	0.0001	14	15	0.0069	0.0382	0.0000
4	12	0.0016	0.0207	0.0150	15	16	0.0209	0.0512	0.0000
5	6	0.0069	0.0300	0.0990	16	17	0.0990	0.0600	0.0000
6	7	0.0053	0.0306	0.0010	16	20	0.0239	0.0585	0.0000
6	11	0.0097	0.0570	0.0001	17	18	0.0032	0.0600	0.0038
6	18	0.0037	0.0222	0.0012	17	21	0.2290	0.4450	0.0000
6	19	0.0035	0.0660	0.0450	19	23	0.0300	0.1310	0.0000
6	21	0.0050	0.0900	0.0226	19	24	0.0300	0.1250	0.0002
7	8	0.0012	0.0069	0.0001	19	25	0.1190	0.2249	0.0004
7	9	0.0009	0.0429	0.0250	20	21	0.0657	0.1570	0.0000
8	12	0.0020	0.0180	0.0200	20	22	0.0150	0.0366	0.0000
9	10	0.0010	0.0493	0.0010	21	24	0.0476	0.1510	0.0000
10	12	0.0024	0.0132	0.0100	22	23	0.0290	0.0990	0.0000
10	19	0.0547	0.2360	0.0000	22	24	0.0310	0.0880	0.0000
10	20	0.0066	0.0160	0.0010	23	25	0.0987	0.1168	0.0000

3.4 Develop MATLAB GUI Using MATLAB GUIDE

GUIDE, the MATLAB graphical user interface development environment, provides a set of tools for creating graphical user interfaces (GUIs). These tools simplify the process of laying out and programming GUIs.

There are 5 steps in build the MATLAB GUI. First Use a MATLAB tool called guide (GUI Development Environment) to layout the components that show in Figure 3.4. This tool allows a programmer to layout the GUI, selecting and aligning the GUI components to be placed in it. The basic component of the MATLAB GUI is shown in Table 3.11.

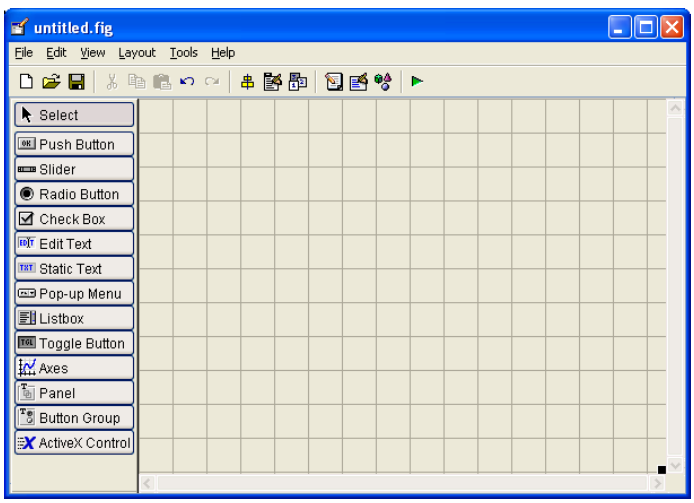


Figure 3.4: MATLAB GUIDE Layouts

Next is Use a MATLAB tool called the Property Inspector (built into guide) to give each component a name (a "tag") and to set the characteristics of each component, such as its color, the text it displays, and so on. After that, save the figure to a file. When the figure is saved, two files will be created on disk with the same name but different extents. The fig file contains the actual GUI that has been created, and the M-file contains the code to load the figure and skeleton call backs for each GUI element.

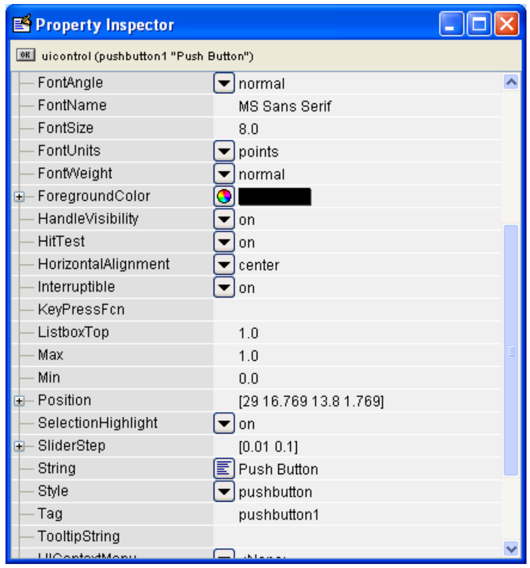


Figure 3.5: Property Inspector

Table 3.11: Basic MATLAB GUI Component

Element	Created By	Description
Graphical Controls		
Pushbutton	uicontrol	A graphical component that implements a pushbutton. It triggers a callback when clicked with a mouse.
Toggle button	uicontrol	A graphical component that implements a toggle button. A toggle button is either "on" or "off," and it changes state each time that it is clicked. Each mouse button click also triggers a callback.
Radio button	uicontrol	A radio button is a type of toggle button that appears as a small circle with a dot in the middle when it is "on." Groups of radio buttons are used to implement mutually exclusive choices. Each mouse click on a radio button triggers a callback.
Check box	uicontrol	A check box is a type of toggle button that appears as a small square with a check mark in it when it is "on." Each mouse click on a check box triggers a callback.
Edit box	uicontrol	An edit box displays a text string and allows the user to modify the information displayed. A callback is triggered when the user presses the Enter key.
List box	uicontrol	A list box is a graphical control that displays a series of text strings. A user can select one of the text strings by single- or double-clicking on it. A callback is triggered when the user selects a string.
Popup menus	uicontrol	A popup menu is a graphical control that displays a series of text strings in response to a mouse click. When the popup menu is not clicked on, only the currently selected string is visible.
Slider	uicontrol	A slider is a graphical control to adjust a value in a smooth, continuous fashion by dragging the control with a mouse. Each slider change triggers a callback.

Static Elements		
Frame	`uicontrol`	Creates a frame, which is a rectangular box within a figure. Frames are used to group sets of controls together. Frames never trigger callbacks.
Text field	`uicontrol`	Creates a label, which is a text string located at a point on the figure. Text fields never trigger callbacks.
Menus and Axes		
Menu items	`uimenu`	Creates a menu item. Menu items trigger a callback when a mouse button is released over them.
Context menus	`uicontextmenu`	Creates a context menu, which is a menu that appears over a graphical object when a user right-clicks the mouse on that object.
Axes	`axes`	Creates a new set of axes to display data on. Axes never trigger callbacks.

After laying out the GUI component and set the property, the GUI will be look like in Figure 3.6.

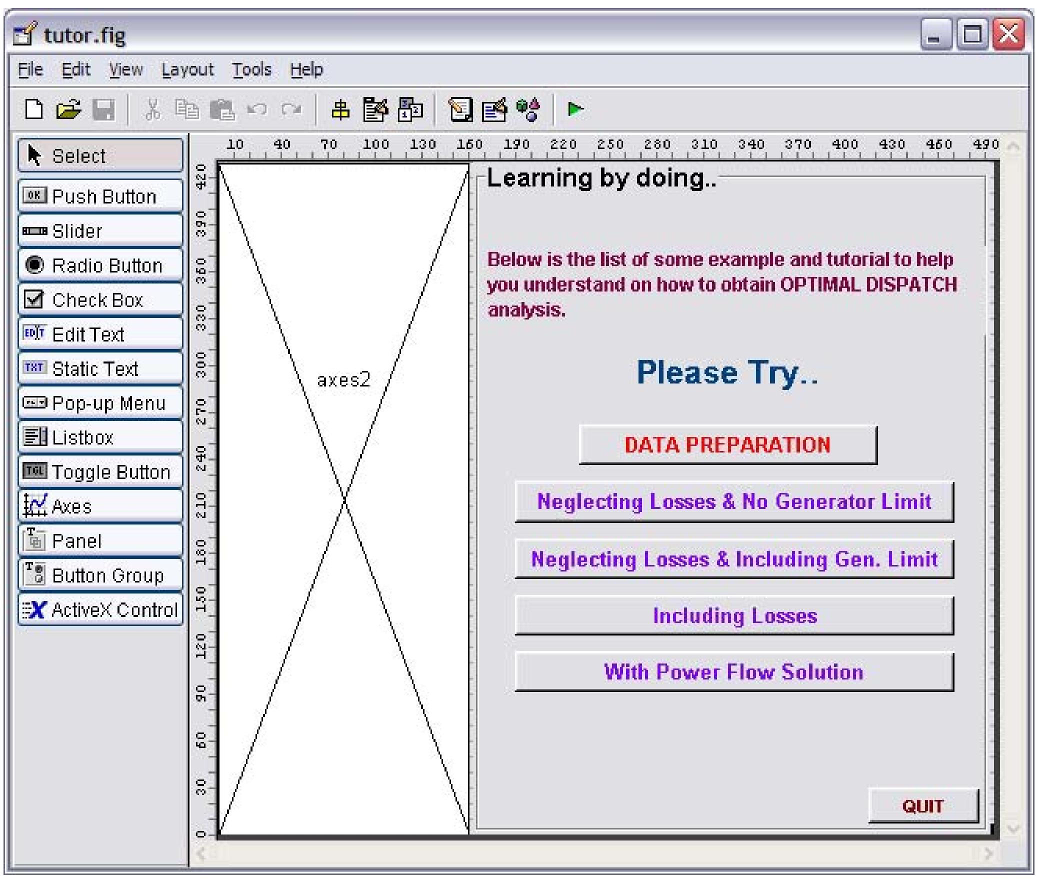


Figure 3.6: Example of Completed GUI

Figure 3.7 show the example of M.file that generated when the completed layout like in Figure 3.6 have been saved. Using the M.file, the behavior of the GUI can be programmed by several code or function. This code or function will controls how the GUI respond to events such as button clicks, slider movement, menu item selection, or the creation and deletion of component. This programming takes the form of set of functions, called callbacks, for each component and for the GUI itself.

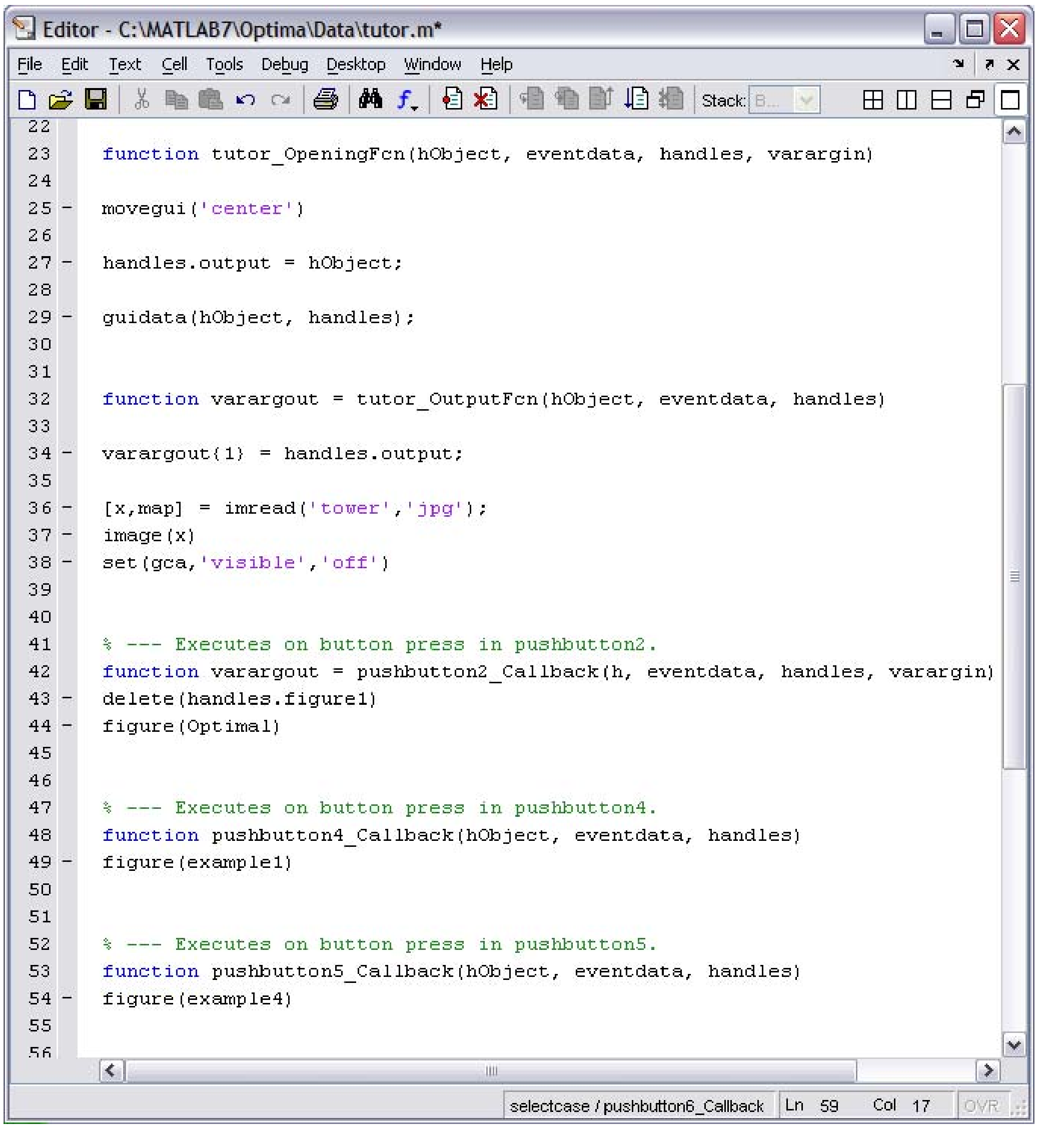


Figure 3.7: Example of M.file

CHAPTER 4

RESULT AND DISCUSSIONS

4.1 Introduction

The discussion of this chapter was categorized by two parts. The first parts consist of the discussions on the software package development using MATLAB Graphical User Interface Development Environment. Meanwhile the second part consist the discussion on the optimal dispatch simulation result with MATLAB based on problem in Section 3.

4.2 Optimal Dispatch of Power Generation Software Package

4.2.1 Introduction

Optimal Dispatch of Power Generation Software Package is a MATLAB GUI file that has been developed to solve the optimal dispatch analysis. The entire software package file was installed in the C:\MATLAB7\Optimal\Optima. Firstly, user must copy the Optima folder to the MATLAB7 folder. This is important because if this step is not followed, the user will face difficulties to load the software. Folder Optima have three subfolders that are Data, Formula, and Figure. The first time user must add the entire folder by click the 'Add with Subfolder...' as shown in Figure 4.1. After the adding path process completed by clicking save button, users just have to type 'Optimal' in the command window to load the software package.

To view the list of folder in Optima folder, user can type 'cd('c:\matlab7\optima')' and followed by 'ls'. The step is same if the users want to view the list of file that have in Data, Formula and Picture folder by typing 'cd('c:\matlab7\optima\data')' and followed by 'ls'. If the users want to learn more about this command, the user can type 'help cd' or 'help ls' in command window in MATLAB.

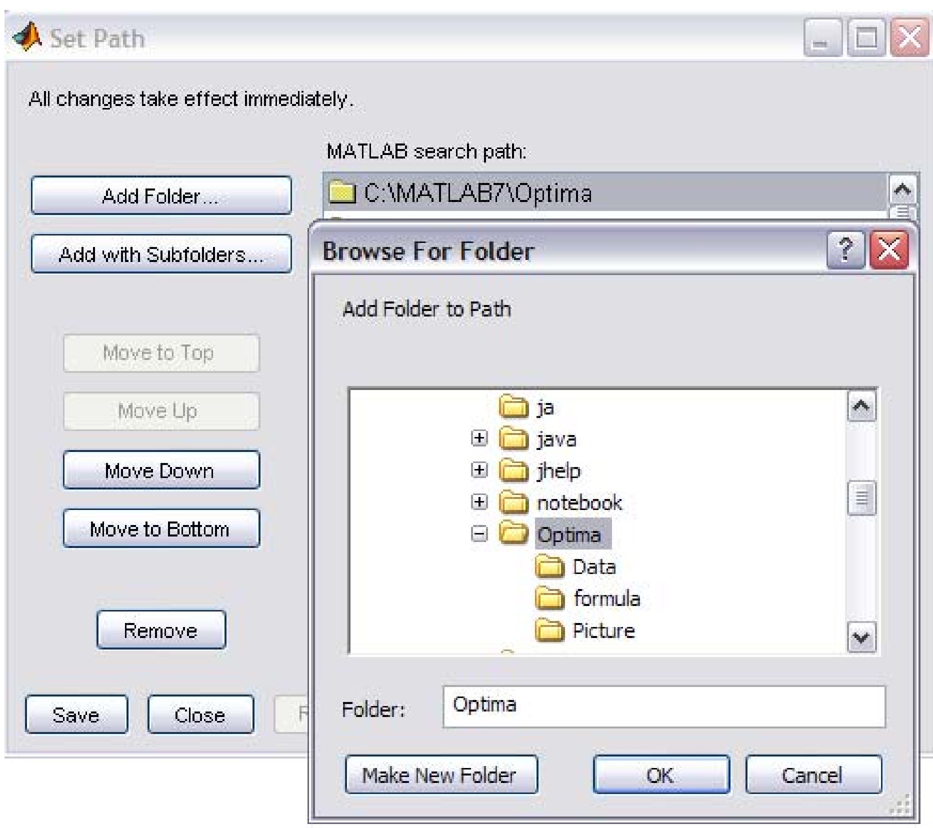


Figure 4.1: MATLAB Set Path

4.2.2 Detail of Software Package

After entered the MATLAB window by clicking twice at MATLAB icon, user have to load the software package. To load the Optimal Dispatch of Power Generation Software Package, user must type ‘Optimal’ in MATLAB command window. The main page of the software package will appear as shown in Figure 4.2.

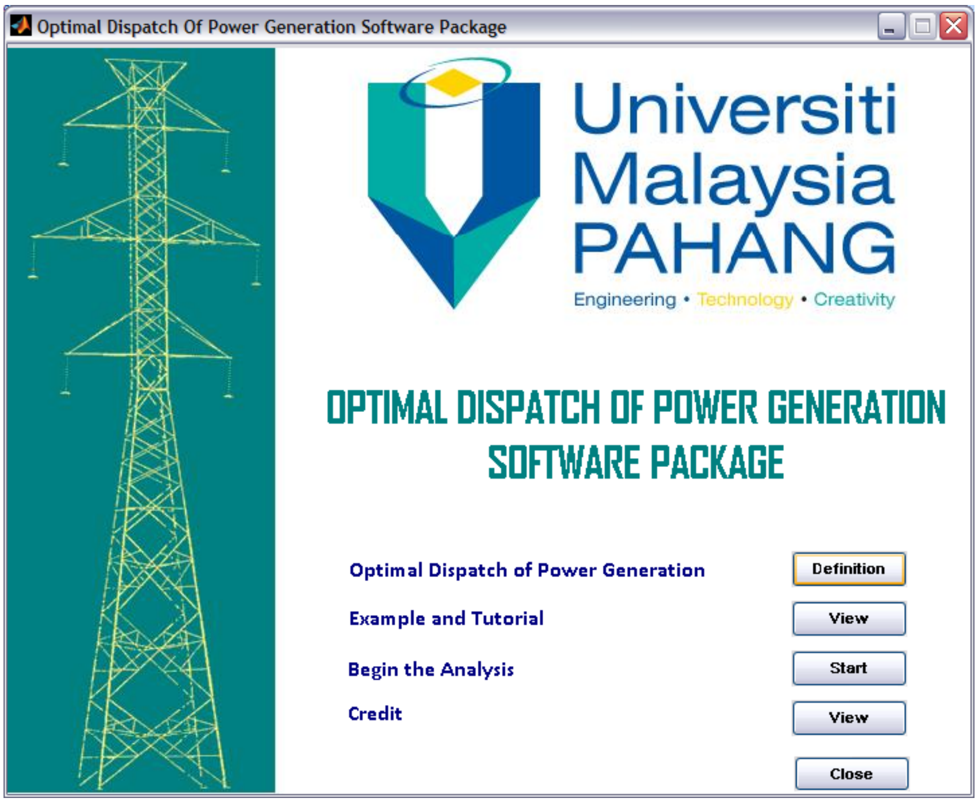


Figure 4.2: Main Page of software package

This main page has five push button that are Definition, View, Start, View and Close. The pushbutton ‘Definition’ will open the new window named Optimal Info that gives the user information about the definition of optimal dispatch of power generation analysis. This can be seen in like in Figure 4.3. The ‘View’ button will open the new window named ‘Example & Tutorial’ that give the user list of example and tutorial on optimal dispatch analysis as shown in Figure 4.4.

The 'Start' button will open the new window named 'Type of Analysis' which give the user list of option on type of analysis to begin analysis. This window is shown in Figure 4.5. The 'View' button will open the new window named 'Credit' that give the information about the designer and the supervisor of this software package like in figure 4.6. The last button 'Close' will close the entire windows that are opened, this function are applied for this main page only. Another 'Close' button that use on other window can just close that window only.

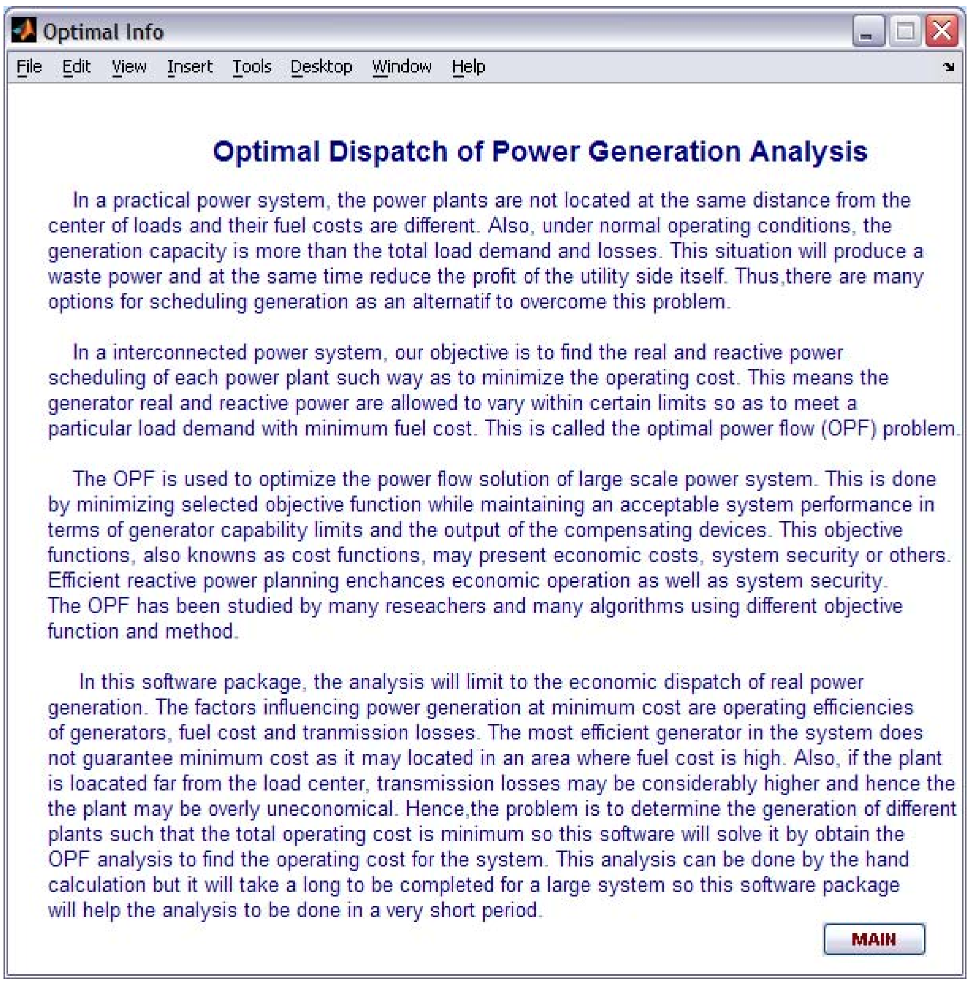


Figure 4.3: Optimal Dispatch Definition

This figure is generated by OptimalInfo.fig, OptimalInfo.asv and OptimalInfo.m file where the first two file generate the window and the last file generated the content that appear in the window. This window briefly gives the information to the user about optimal dispatch of power generation analysis. The 'MAIN' pushbutton will close this window and call back the main page of software package window.

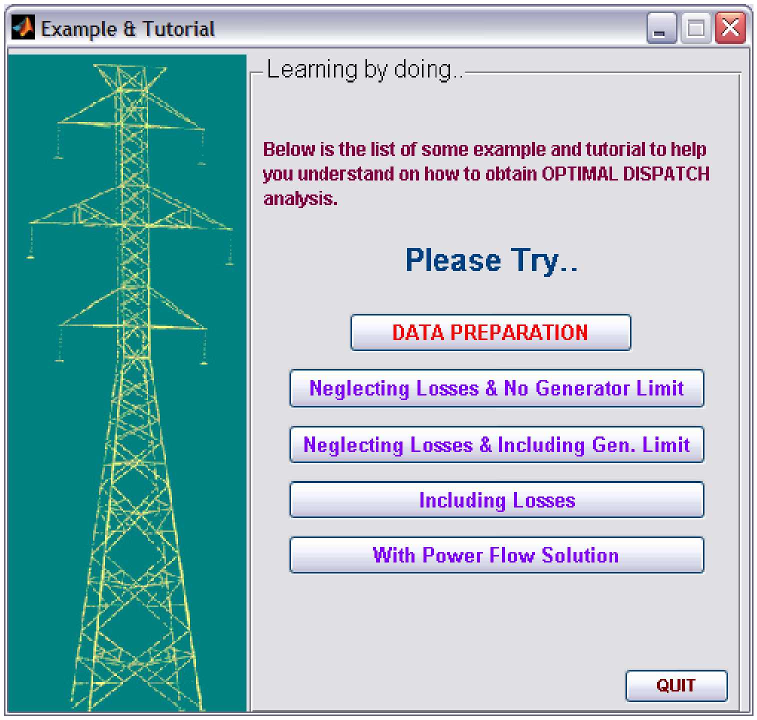


Figure 4.4: Example and Tutorial

This window gives the list of example and tip to user especially the beginner on how this software package running the analysis. 'Data Preparation' button will open the window named 'Data Preparation' as shown in Figure 4.5 which give the information on data that required before starting the analysis. Another four pushbutton represent example of problem on optimal dispatch analysis that provided in this software package. There are four types of analysis; optimal dispatch neglecting losses and no generator limit, optimal dispatch neglecting losses and including generator limit, optimal dispatch including losses and optimal dispatch with power flow solution. These four buttons will open new windows as shown in Figure 4.6, 4.7, 4.8 and 4.9.

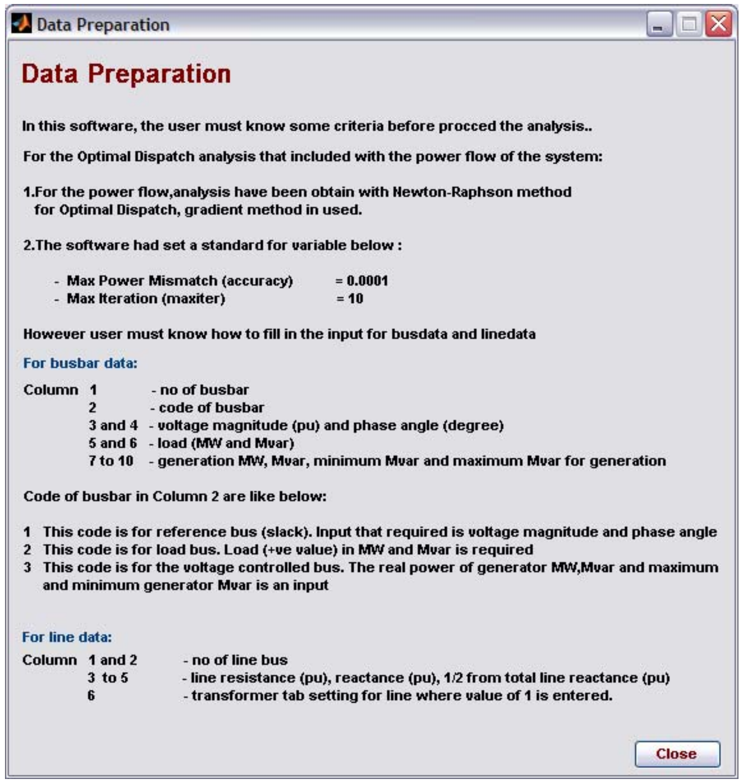


Figure 4.5: Data Preparation

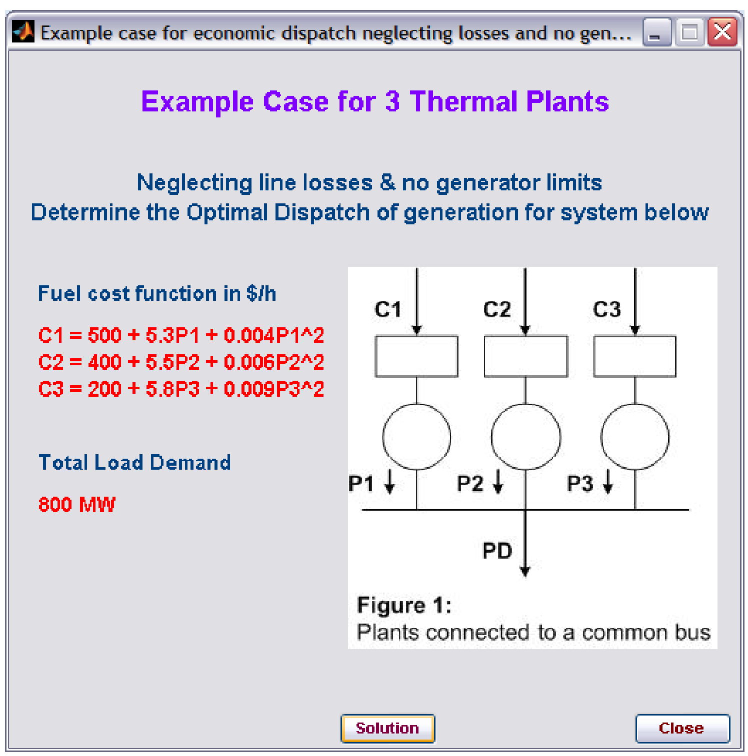


Figure 4.6: Problem Neglecting Losses and No Generator Limit

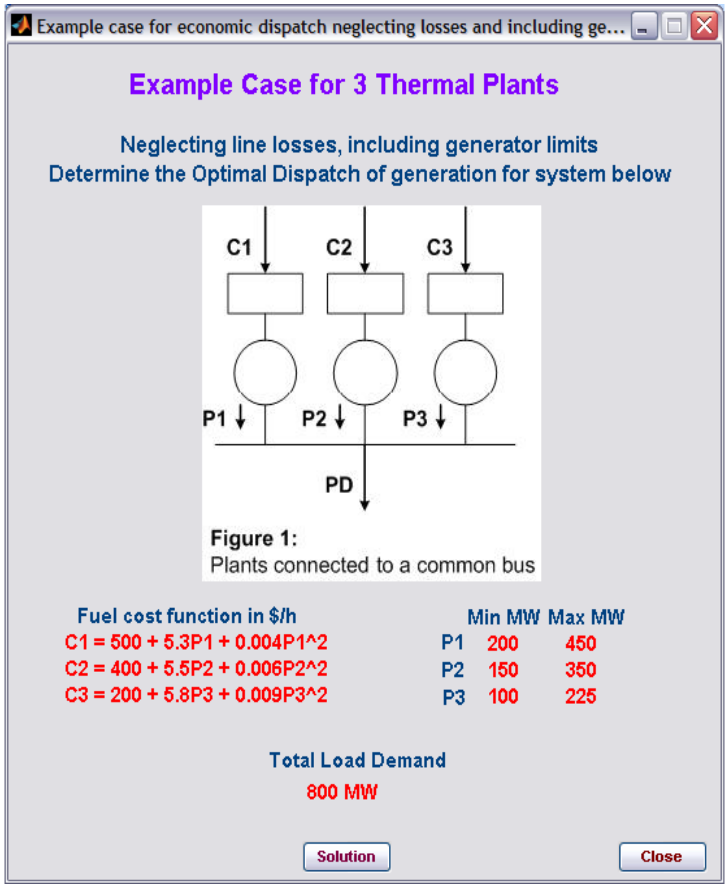


Figure 4.7: Problem Neglecting Losses and Including Generator Limit

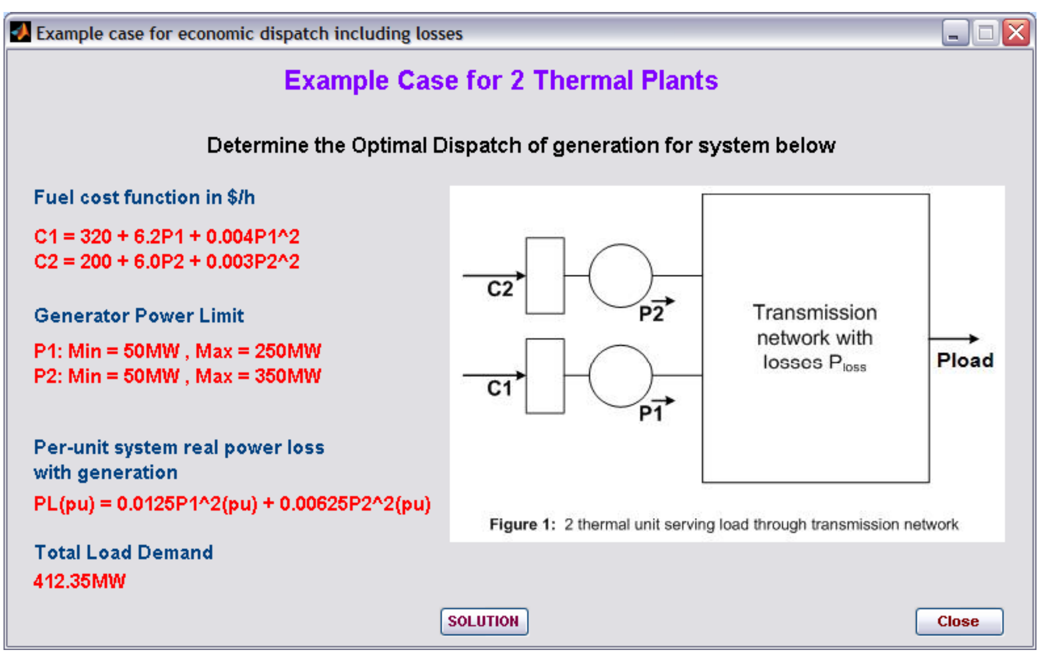


Figure 4.8: Problem Including Losses

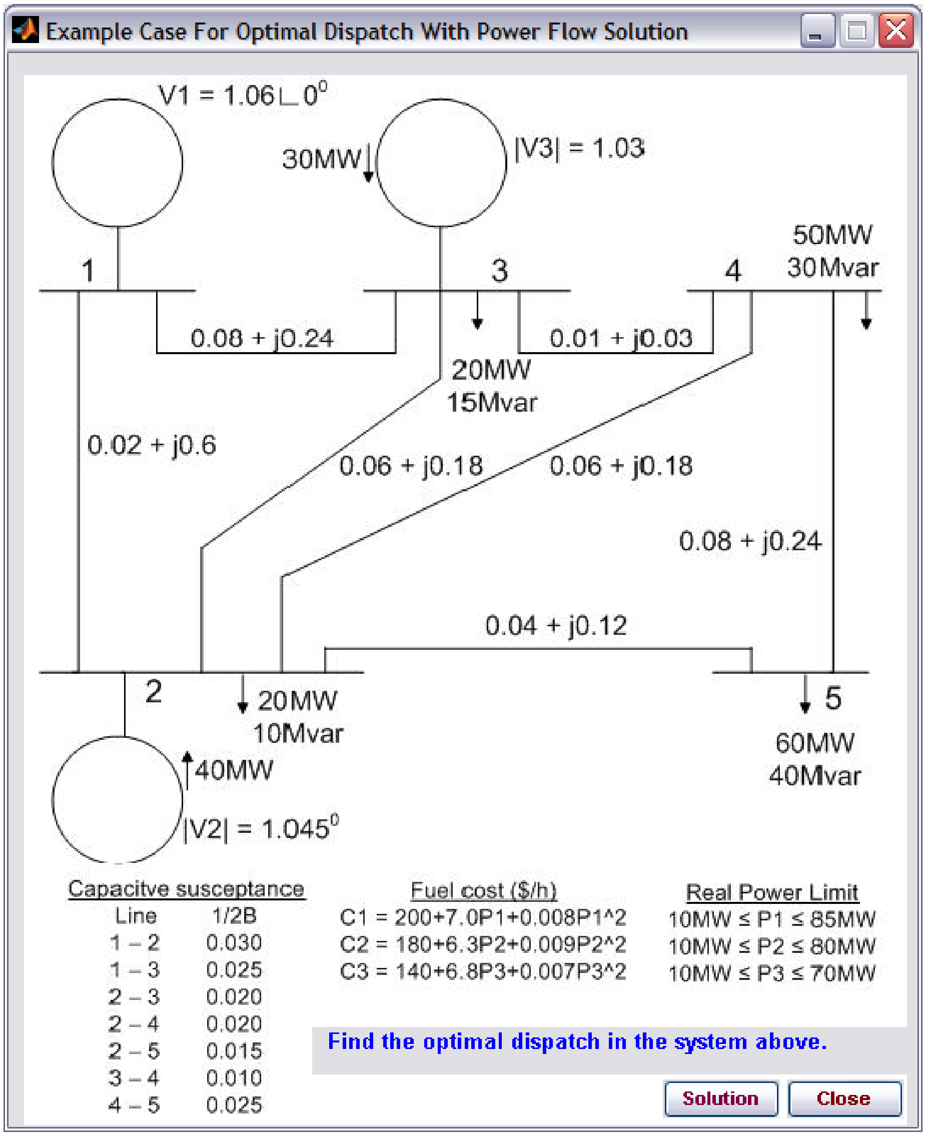


Figure 4.9: Problem with Power Flow Solution

For all the example of problem on optimal dispatch analysis as shown in Figure 4.6, 4.7, 4.8 and 4.9, the solution and important tips are provided to help the user understand on how to fill in the input data. Window of solution, tips and result for problem with power flow solution are shown in Figure 4.10, 4.11 and 4.12.

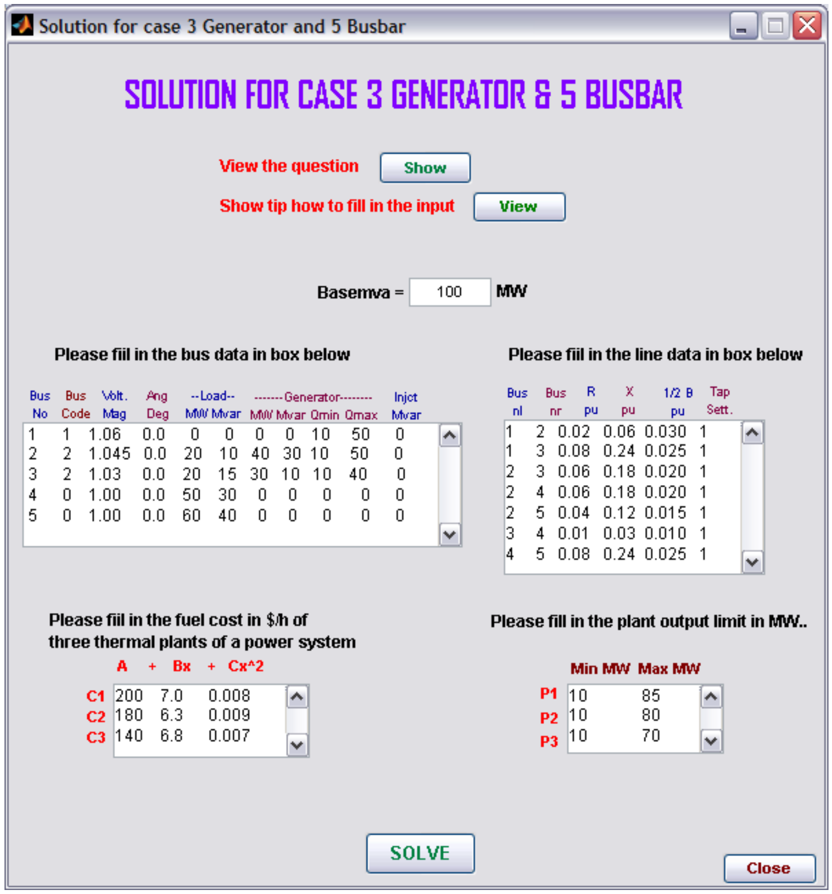


Figure 4.10: Input data for problem with power flow solution

Figure 4.10 above show the window where user must fill in the input data for the problem with power flow solution. For this input data window, user must fill in the bus data, line data, fuel cost, generator limit and base MVA. The input data required are different depends on type of analysis. To view back the question for this problem as shown in Figure 4.9, users need to click the 'Show' button at the top of this window. In this example the input data are left in blank, it is because user are given a chance to fill in the input data based on their own knowledge. The 'View' button will open the tips window as shown in Figure 4.11 and the 'SOLVE' button will simulate the result of this analysis as shown in Figure 4.12.

For all the example of problems that are given in this software, important tips are given to the user as in Figure 4.11. These tips are given because users are encouraging to do self-learning process. With this tips, users can self recheck their input data by open this tips window by clicking the button view in the Figure 4.10.

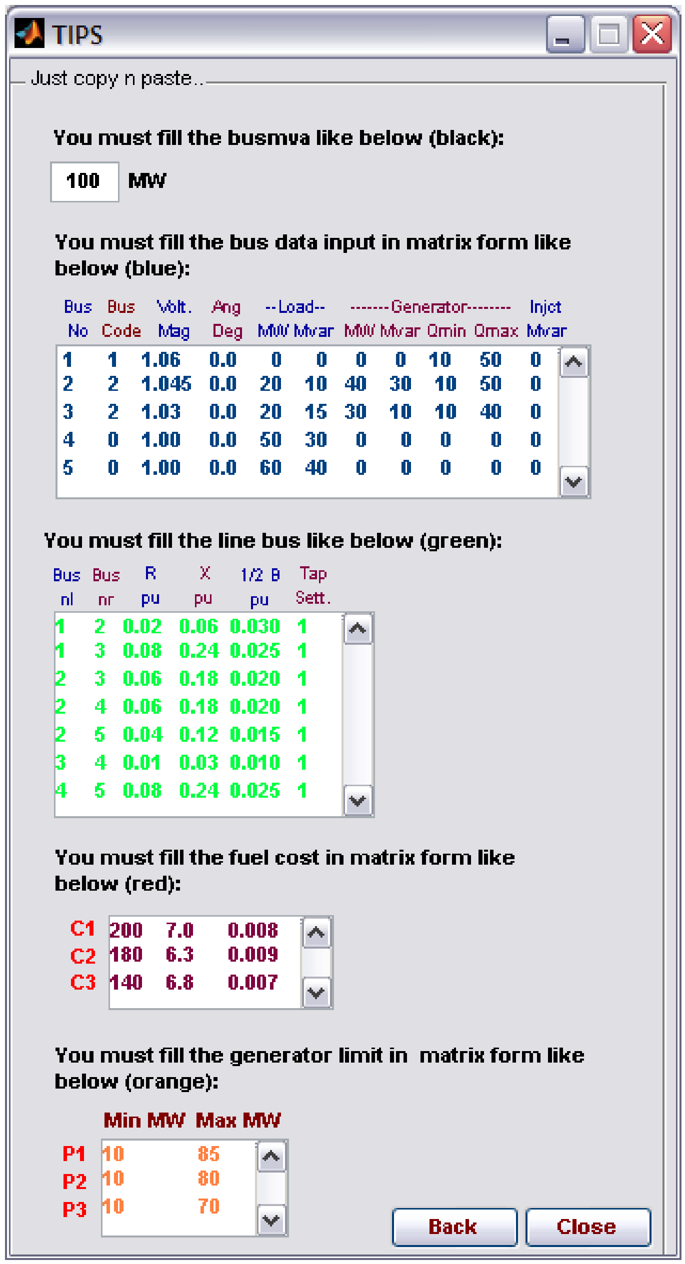


Figure 4.11: Tips on Input Data Preparation.

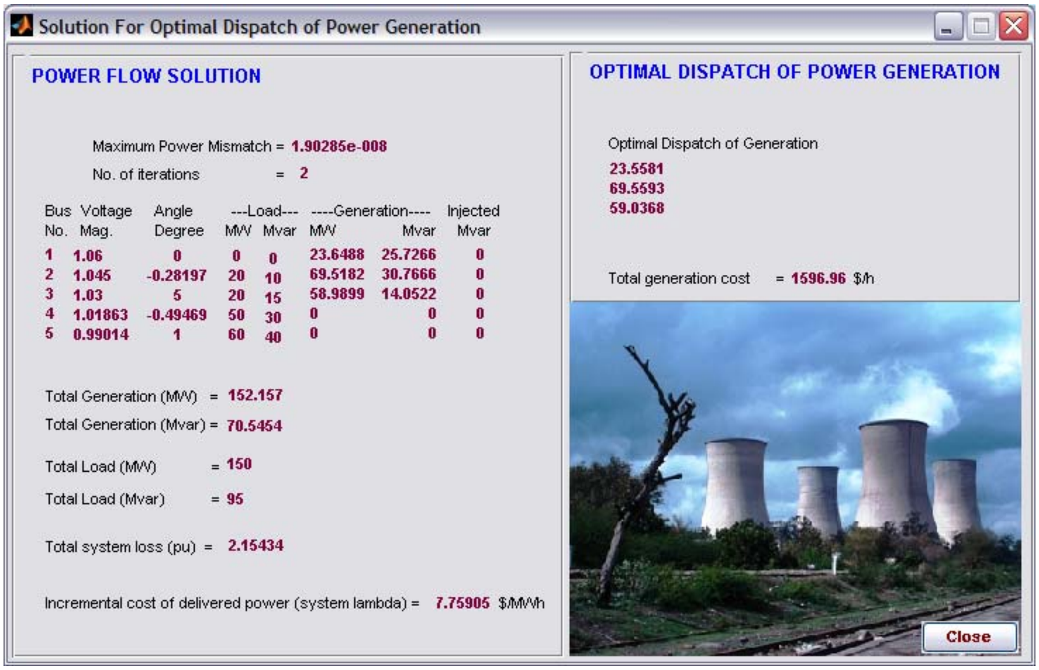


Figure 4.12: Result Window for Optimal Dispatch with Power Flow Solution

The results are different based on the type of analysis chosen. For optimal dispatch with power flow problem, the result is shown in Figure 4.12 where the results are separated by two parts. The first part is shown in the result on power flow solution such as maximum power mismatch, no. of iteration, power flow of all buses, total generation(MW and Mvar), total load(MW and Mvar), total system loss(pu) and incremental cost delivered power(system lambda). For the optimal dispatch part, the result is shown the optimal dispatch of generation for each power plant and the generation cost.

Before users start with the optimal dispatch of power generation analysis, they have to choose types of analysis that are given in Figure 4.13 by clicking the ‘Start’ button in the main menu. Type of analysis offered by this software is economic dispatch neglecting losses and no generator limit, economic dispatch neglecting losses and including generator limit, economic dispatch including losses and economic dispatch with power flow solution.

All the 'Proceed' button on this window will open new window as shown in Figure 4.14, 4.15, 4.16 and 4.17 based on type of analysis that chosen by the user.

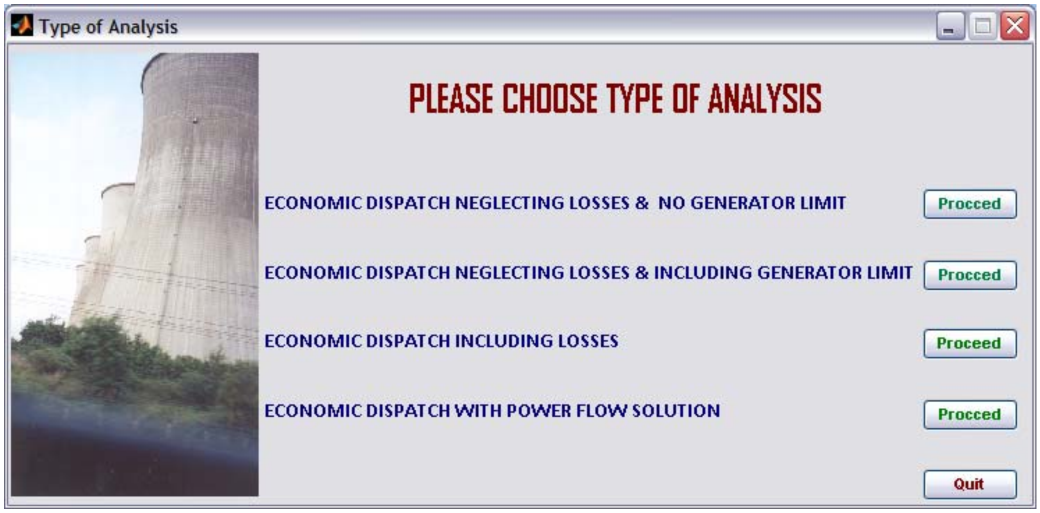


Figure 4.13: Type of Analysis

in inputs data is exactly same with the process in the example and tutorial given in this software package. It is very important for user especially the beginner to view all the example and input data preparation tips given to make sure there have no problem to start the analysis. The input data that have to fill in are different base on the type of analysis chosen.

Before users click the 'SOLVE' button to get the result, user must check all the input data that have been entered. The input data edit boxes have the slider in case there have a long or many input data to enter. For the result of the analysis, the result window will appear exactly same with the result window for the problem in the example and tutorial as shown in Figure 4.12. The results of the analysis are different based on the type of analysis.

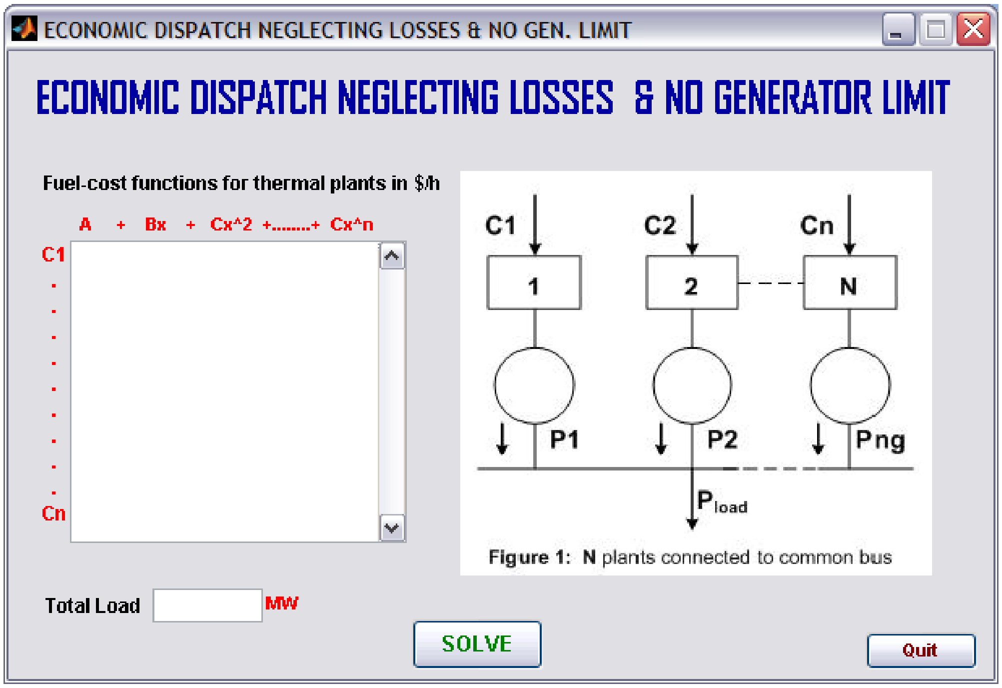


Figure 4.14: Economic Dispatch Neglecting Losses and No Generator Limit

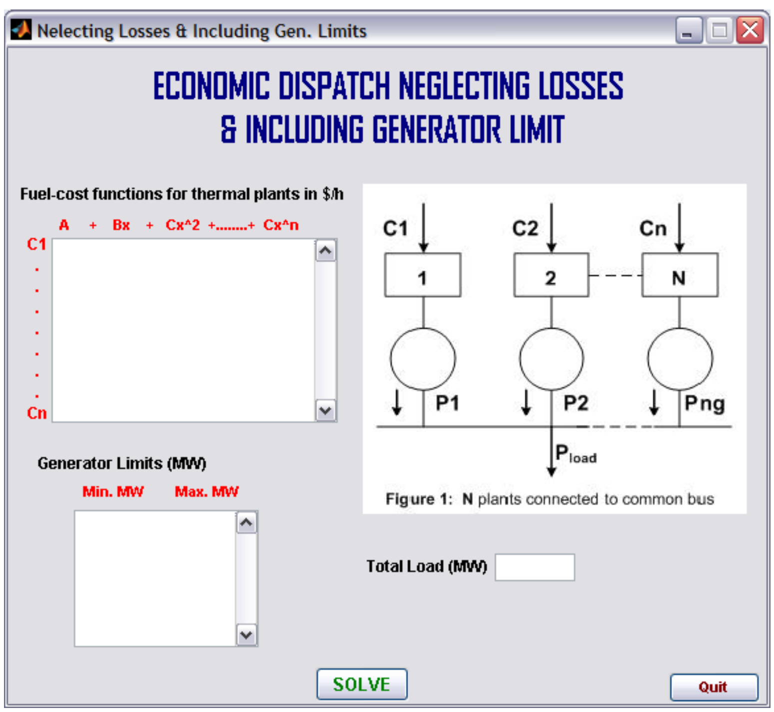


Figure 4.15: Economic Dispatch Neglecting Losses and Including Generator Limit

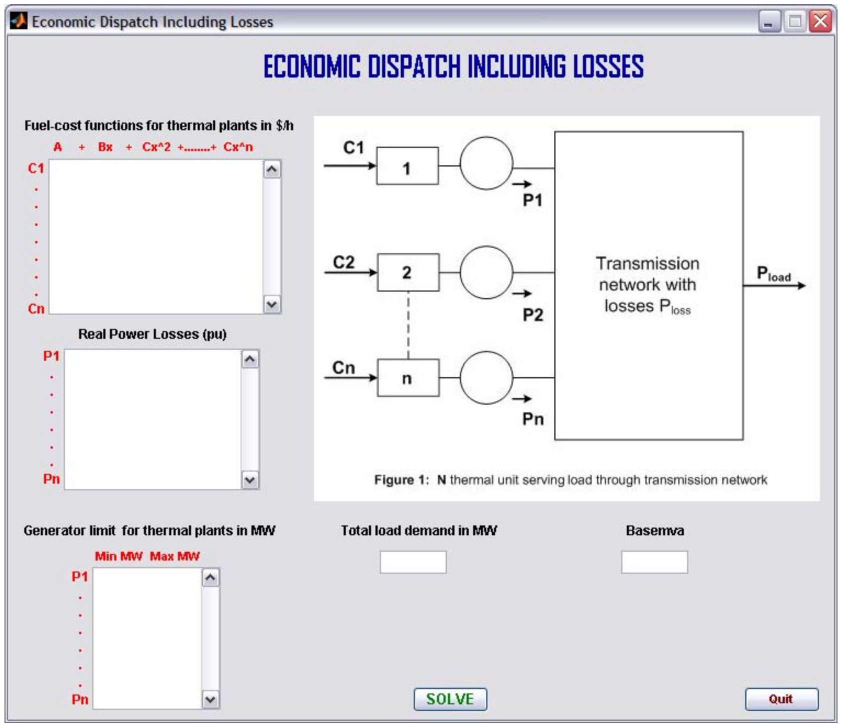


Figure 4.16: Economic Dispatch Including Losses

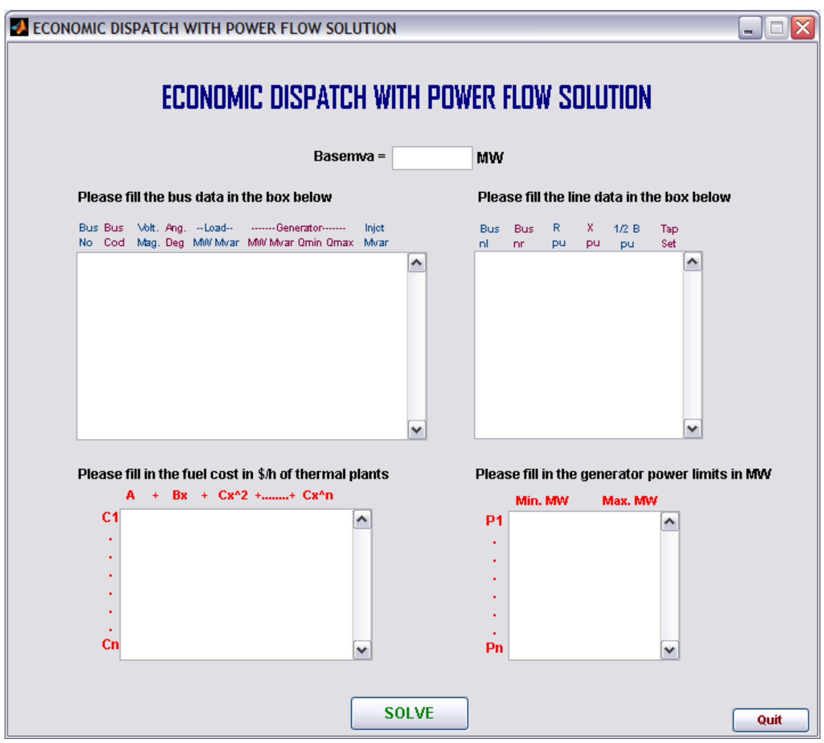


Figure 4.17: Economic Dispatch with Power Flow Solution

Figure 4.18: Credit

4.2.3 Formula of Power Flow Input Data

4.2.3.1 Line Data Input Format

Table 4.1: Line Data Format

No. Column	Detail	Additional
1	From bus	n, interger
2	To Bus	n, interger
3	Line Resistance	Per unit (p.u)
4	Reactance	Per unit (p.u)
5	1/2 Total Line Charging Susceptance	Per unit (p.u)
6	Transformer Tap	1 is entered

4.2.3.2 Bus Data Input Format

Table 4.2: Bus Data Format

No. Column	Detail	Additional
1	Bus number	n, interger
2	Code of bus	Slack bus = 1 Load bus = 0 Voltage controlled bus = 2
3	Voltage Magnitude	Per unit (p.u)
4	Angle Degree	Degree
5	Load Power (MW)	Active Power
6	Load Power (MVar)	Reactive Power
7	Generation Power (MW)	Active Power
8	Generation Power (MVar)	Reactive Power
9	Minimum Mvar of Generation	Qmin
10	Maximum Mvar of Generation	Qmax
11	Injected of Shunt Capacitor	Mvar

4.3 Optimal Dispatch Simulation Result Using MATLAB

For simulation using MATLAB, the power flow analysis has been done using three methods that are Newton Raphson, Fast Decouple and Gauss Seidel Method meanwhile the optimal dispatch analysis was been done by gradient method. The 5 and 26 busbar of one line diagram was chosen for the simulation.

4.3.1 Result for One-line Diagram of 5-Busbar

4.3.1.1 Newton-Raphson Method

Firstly, the power flow solution is obtained as shown in Table 4.3:

Power Flow Solution by Newton-Raphson Method

Maximum Power Mismatch = 1.43025e-005

No. of Iterations = 3

Table 4.3: Power Flow Solution by Newton-Raphson Method

Bus No.	Voltage Magnitude	Angle Degree	Load		Generation		Injected Mvar
			MW	Mvar	MW	Mvar	
1	1.060	0.000	0.000	0.000	83.051	7.271	0.000
2	1.045	-1.782	20.000	10.000	40.000	41.811	0.000
3	1.030	-2.664	20.000	15.000	30.000	24.148	0.000
4	1.019	-3.243	50.000	30.000	0.000	0.000	0.000
5	0.990	-4.405	60.000	40.000	0.000	0.000	0.000

Total Load (MW) = 150.000

Total Load (Mvar) = 95.000

Total Generation (MW) = 153.051

Total Generation (Mvar) = 73.230

Total System Loss = 3.05248 MW

Total Generation Cost = 1633.24 $/h

Then, the simulation continues by using gradient method to find the Optimal Dispatch of Generation.

The analysis will repeat until Dpslack is near to 0.001. The Table 4.4 is shown the result of analysis.

Dpslack is the different (absolute value) between the scheduled slack generation determined from the coordination equation and the slack generation obtains from power flow solution [5].

Table 4.4: Result from gradient method

Dpslack (pu)	System Lambda ($/MWh)	P1 (MW)	P2 (MW)	P3 (MW)	Total Generation (MW)	Total Loss (MW)
0.4960	7.767608	33.4558	64.1101	55.1005	152.6664	2.6664
0.0626	7.736667	26.7427	67.9440	57.4127	152.0994	2.0994
0.0189	7.748796	24.9190	68.9023	58.3061	152.1274	2.1274
0.0079	7.754395	24.1668	69.2685	58.7069	152.1422	2.1422
0.0037	7.757066	23.8165	69.4362	58.8966	152.1493	2.1493
0.0018	7.758386	23.6446	69.5182	58.9899	152.1527	2.1527
0.0009	7.759051	23.5581	69.5593	59.0368	152.1542	2.1542

After the Dpslack is reach to 0.001, the new power flow solution as shown in Table 4.5 and optimal dispatch of generation is obtained:

Optimal Dispatch of Generation:

P1 = 23.5581MW
P2 = 69.5593MW
P2 = 59.0368MW

Total system loss = 2.15434 MW

Power Flow Solution by Newton-Raphson Method

Maximum Power Mismatch = 1.90286e-008

No. of Iterations = 2

Table 4.5: New Power Flow Solution by Newton-Raphson Method

Bus Number	Voltage Magnitude	Angle Degree	Load		Generation		Injected Mvar
			MW	Mvar	MW	Mvar	
1	1.060	0.000	0.000	0.000	23.649	25.727	0.000
2	1.045	-0.282	20.000	10.000	69.518	30.767	0.000
3	1.030	-0.495	20.000	15.000	58.990	14.052	0.000
4	1.019	-1.208	50.000	30.000	0.000	0.000	0.000
5	0.990	-2.729	60.000	40.000	0.000	0.000	0.000

Total Load (MW) = 150.000

Total Load (Mvar) = 95.000

Total Generation (MW) = 152.157

Total Generation (Mvar) = 70.545

Injected Mvar = 0.000

Total generation cost = 1596.96 $/h

4.3.1.2 Gauss Seidel Method

Firstly, the power flow solution is obtained as shown in Table 4.6:

Power Flow Solution by Gauss-Seidel Method

Maximum Power Mismatch = 9.76332e-005

No. of Iterations = 22

Table 4.6: Power Flow Solution by Gauss-Seidel Method

Bus No.	Voltage Magnitude	Angle Degree	Load		Generation		Injected Mvar
			MW	Mvar	MW	Mvar	
1	1.060	0.000	0.000	0.000	83.042	7.274	0.000
2	1.045	-1.782	20.000	10.000	40.000	41.811	0.000
3	1.030	-2.664	20.000	15.000	30.000	24.148	0.000
4	1.019	-3.243	50.000	30.000	0.000	0.000	0.000
5	0.990	-4.405	60.000	40.000	0.000	0.000	0.000

Total Load (MW) = 150.000

Total Load (Mvar) = 95.000

Total Generation (MW) = 153.042

Total Generation (Mvar) = 73.230

Total System Loss = 3.05208 MW

Total Generation Cost = 1633.16 $/h

Then, the simulation continues by using gradient method to find the Optimal Dispatch of Generation.

The analysis will repeat until Dpslack is near to 0.001. The Table 4.7 is shown the result of analysis.

Dpslack is the different (absolute value) between the scheduled slack generation determined from the coordination equation and the slack generation obtains from power flow solution [5].

Table 4.7: Result from gradient method

Dpslack (pu)	Lambda ($/MWh)	P1 (MW)	P2 (MW)	P3 (MW)	Total Generation (MW)	Total Loss (MW)
0.4959	7.767606	33.4553	64.1101	55.1009	152.6663	2.6663
0.0627	7.736650	26.7449	67.9430	57.4114	152.0993	2.0993
0.0190	7.748746	24.9251	68.8995	58.3026	152.1272	2.1272
0.0080	7.754342	24.1735	69.2654	58.7031	152.1420	2.1420
0.0038	7.757006	23.8240	69.4327	58.8924	152.1491	2.1491
0.0019	7.758315	23.6535	69.5141	58.9849	152.1525	2.1525
0.0010	7.758983	23.5667	69.5553	59.0320	152.1540	2.1540

After the Dpslack is reach to 0.001, the new power flow solution as shown in Table 4.8 and optimal dispatch of generation is obtained:

Optimal Dispatch of Generation:

P1 = 23.5667
P2 = 69.5553
P3 = 59.0320

Total system loss = 2.15415 MW

Power Flow Solution by Gauss-Seidel Method

Maximum Power Mismatch = 7.5229e-005

No. of Iterations = 8

Table 4.8: New Power Flow Solution by Gauss-Seidel Method

Bus Number	Voltage Magnitude	Angle Degree	Load		Generation		Injected Mvar
			MW	Mvar	MW	Mvar	
1	1.060	0.000	0.000	0.000	23.666	25.721	0.000
2	1.045	-0.282	20.000	10.000	69.514	30.769	0.000
3	1.030	-0.495	20.000	15.000	58.985	14.056	0.000
4	1.019	-1.208	50.000	30.000	0.000	0.000	0.000
5	0.990	-2.729	60.000	40.000	0.000	0.000	0.000

Total Load (MW) = 150.000

Total Load (Mvar) = 95.000

Total Generation (MW) = 152.165

Total Generation (Mvar) = 70.547

Injected Mvar = 0.000

Total generation cost = 1596.96 $/h

4.3.1.3 Fast Decouple Method

Firstly, the power flow solution is obtained as shown in Table 4. 9:

Power Flow Solution by Fast Decoupled Method

Maximum Power Mismatch = 4.03558e-005

No. of Iterations = 10

Table 4.9: Power Flow Solution by Fast Decoupled Method

Bus No.	Voltage Magnitude	Angle Degree	Load		Generation		Injected Mvar
			MW	Mvar	MW	Mvar	
1	1.060	0.000	0.000	0.000	83.053	7.271	0.000
2	1.045	-1.782	20.000	10.000	40.000	41.810	0.000
3	1.030	-2.664	20.000	15.000	30.000	24.147	0.000
4	1.019	-3.243	50.000	30.000	0.000	0.000	0.000
5	0.990	-4.405	60.000	40.000	0.000	0.000	0.000

Total Load (MW) = 150.000

Total Load (Mvar) = 95.000

Total Generation (MW) = 153.053

Total Generation (Mvar) = 73.228

Total System Loss = 3.05252 MW

Total Generation Cost = 1633.25 $/h

Then, the simulation continues by using gradient method to find the Optimal Dispatch of Generation.

The analysis will repeat until Dpslack is near to 0.001. The Table 4.10 is shown the result of analysis

Dpslack is the different (absolute value) between the scheduled slack generation determined from the coordination equation and the slack generation obtains from power flow solution [5].

.

Table 4.10: Result from gradient method

Dpslack (pu)	Lambda ($/MWh)	P1 (MW)	P2 (MW)	P3 (MW)	Total Generation (MW)	Total Loss (MW)
0.496	7.767604	33.4557	64.1102	55.1005	152.6664	2.6664
0.0626	7.736679	26.7431	67.9437	57.4127	152.0995	2.0995
0.0189	7.748793	24.9193	68.9021	58.3059	152.1273	2.1273
0.0079	7.754348	24.1718	69.2662	58.704	152.142	2.142
0.0037	7.757027	23.821	69.4341	58.8941	152.1492	2.1492
0.0018	7.758359	23.6479	69.5166	58.9881	152.1526	2.1526
0.0009	7.759095	23.5531	69.5616	59.0397	152.1544	2.1544

After the Dpslack is reach to 0.001, the new power flow solution as shown in Table 4.11 and optimal dispatch of generation is obtained:

Optimal Dispatch of Generation:

P1 = 23.5531 MW

P2 = 69.5616 MW

P3 = 59.0397 MW

Total system loss = 2.15449 MW

Power Flow Solution by Fast Decoupled Method

Maximum Power Mismatch = 7.27225e-005

No. of Iterations = 2

Table 4.11: New Power Flow Solution by Fast Decoupled Method

Bus	Voltage	Angle	Load		Generation		Injected
Number	Magnitude	Degree	MW	Mvar	MW	Mvar	Mvar
1	1.060	0.000	0.000	0.000	23.639	25.730	0.000
2	1.045	-0.282	20.000	10.000	69.517	30.765	0.000
3	1.030	-0.495	20.000	15.000	58.988	14.051	0.000
4	1.019	-1.208	50.000	30.000	0.000	0.000	0.000
5	0.990	-2.729	60.000	40.000	0.000	0.000	0.000

Total Load (MW) = 150.000

Total Load (Mvar) = 95.000

Total Generation (MW) = 152.143

Total Generation (Mvar) = 70.546

Total generation cost = 1596.97 $/h

4.3.2 Result for One-line Diagram of 26-Busbar

4.3.2.1 Newton-Raphson Method

Firstly, the power flow solution is obtained as shown in Table 4.12:

Power Flow Solution by Newton-Raphson Method

Maximum Power Mismatch = 3.18289e-010

No. of Iterations = 6

Table 4.12: Power Flow Solution by Newton-Raphson Method

Bus	Voltage	Angle	Load		Generation		Injected
No.	Magnitude	Degree	MW	Mvar	MW	Mvar	Mvar
1	1.025	0.000	51.000	41.000	719.534	224.011	4.000
2	1.020	-0.931	22.000	15.000	79.000	125.354	0.000
3	1.035	-4.213	64.000	50.000	20.000	63.030	0.000
4	1.050	-3.582	25.000	10.000	100.000	49.223	2.000
5	1.045	1.129	50.000	30.000	300.000	124.466	5.000
6	0.999	-2.573	76.000	29.000	0.000	0.000	2.000
7	0.994	-3.204	0.000	0.000	0.000	0.000	0.000
8	0.997	-3.299	0.000	0.000	0.000	0.000	0.000
9	1.009	-5.393	89.000	50.000	0.000	0.000	3.000
10	0.989	-5.561	0.000	0.000	0.000	0.000	0.000
11	0.997	-3.218	25.000	15.000	0.000	0.000	1.500
12	0.993	-4.692	89.000	48.000	0.000	0.000	2.000
13	1.014	-4.430	31.000	15.000	0.000	0.000	0.000
14	1.000	-5.040	24.000	12.000	0.000	0.000	0.000
15	0.991	-5.538	70.000	31.000	0.000	0.000	0.000
16	0.983	-5.882	55.000	27.000	0.000	0.000	0.000
17	0.987	-4.985	78.000	38.000	0.000	0.000	0.000

18	1.007	-1.866	153.000	67.000	0.000	0.000	0.000
19	1.004	-6.397	75.000	15.000	0.000	0.000	5.000
20	0.980	-6.025	48.000	27.000	0.000	0.000	0.000
21	0.997	-5.778	46.000	23.000	0.000	0.000	0.000
22	0.978	-6.437	45.000	22.000	0.000	0.000	0.000
23	0.976	-7.087	25.000	12.000	0.000	0.000	0.000
24	0.968	-7.347	54.000	27.000	0.000	0.000	0.000
25	0.974	-6.775	28.000	13.000	0.000	0.000	0.000
26	1.015	-1.803	40.000	20.000	60.00	32.706	0.000

Total Load (MW) = 1263.000
Total Load (Mvar) = 637.000

Total Generation (MW) = 1278.534
Total Generation (Mvar) = 618.791

Injected Mvar = 25.000

Total System Loss = 3.05252 MW

Total Generation Cost = 1633.25 $/h

Then, the simulation continues by using gradient method to find the Optimal Dispatch of Generation.

The analysis will repeat until Dpslack is near to 0.001. The Table 4.13 is shown the result of analysis.

Dpslack is the different (absolute value) between the scheduled slack generation determined from the coordination equation and the slack generation obtains from power flow solution [5].

Table 4.13: Result from gradient method

Dpslack (pu)	2.454100	0.135500	0.004600	0.000800
Lambda ($/MWh)	13.911780	13.527919	13.536333	13.538113
P1	474.119600	446.966900	447.547900	447.691900
P2	173.788600	172.444700	173.086900	173.193800
P3	190.951500	261.368700	263.363100	263.485900
P4	150.000000	138.280100	138.716100	138.814200
P5	196.719600	169.555800	166.099200	165.588400
P26	103.577200	87.057400	86.938800	87.026000
Total Generation (MW)	1289.156500	1275.673600	1275.752000	1275.800200
Total Load (MW)	1263.000000	1263.000000	1263.000000	1263.000000
Total Loss (MW)	26.156500	12.673600	12.752000	12.800200

After the slack mismatch is reach to 0.001, the new power flow solution as shown in Table 4.14 and optimal dispatch of generation is obtained:

Optimal Dispatch of Generation:

P1 = 447.6919

P2 = 173.1938

P3 = 263.4859

P4 = 138.8142

P5 = 165.5884

P26 = 87.0260

Total system loss = 12.8003 MW

Power Flow Solution by Newton-Raphson Method

Maximum Power Mismatch = 2.33783e-005

No. of Iterations = 2

Table 4.14: New Power Flow Solution by Newton-Raphson Method

Bus No.	Voltage Magnitude	Angle Degree	Load		Generation		Injected Mvar
			MW	Mvar	MW	Mvar	
1	1.025	0.000	51.000	41.000	447.611	250.582	4.000
2	1.020	-0.200	22.000	15.000	173.087	57.303	0.000
3	1.045	-0.639	64.000	50.000	263.363	78.280	0.000
4	1.050	-2.101	25.000	10.000	138.716	33.449	2.000
5	1.045	-1.453	50.000	30.000	166.099	142.890	5.000
6	1.001	-2.874	76.000	29.000	0.000	0.000	2.000
7	0.995	-2.406	0.000	0.000	0.000	0.000	0.000
8	0.998	-2.278	0.000	0.000	0.000	0.000	0.000
9	1.010	-4.387	89.000	50.000	0.000	0.000	3.000
10	0.991	-4.311	0.000	0.000	0.000	0.000	0.000
11	0.998	-2.824	25.000	15.000	0.000	0.000	1.500
12	0.994	-3.282	89.000	48.000	0.000	0.000	2.000
13	1.022	-1.261	31.000	15.000	0.000	0.000	0.000
14	1.008	-2.445	24.000	12.000	0.000	0.000	0.000
15	0.999	-3.229	70.000	31.000	0.000	0.000	0.500
16	0.990	-3.990	55.000	27.000	0.000	0.000	0.000
17	0.983	-4.366	78.000	38.000	0.000	0.000	0.000
18	1.007	-1.884	153.000	67.000	0.000	0.000	0.000
19	1.005	-6.074	75.000	15.000	0.000	0.000	5.000

20	0.983	-4.759	48.000	27.000	0.000	0.000	0.000
21	0.977	-5.411	46.000	23.000	0.000	0.000	0.000
22	0.980	-5.325	45.000	22.000	0.000	0.000	0.000
23	0.978	-6.388	25.000	12.000	0.000	0.000	0.000
24	0.969	-6.672	54.000	27.000	0.000	0.000	0.000
25	0.975	-6.256	28.000	13.000	0.000	0.000	0.000
26	1.015	-0.284	40.000	20.000	86.939	27.892	0.000

Total Load (MW) = 1263.000

Total Load (Mvar) = 637.000

Total Generation (MW) = 1275.815

Total Generation (Mvar) = 590.396

Total Injected (Mvar) = 25.000

Total generation cost = 15447.72 $/h

4.3.2.2 Gauss Seidel Method

Firstly, the power flow solution is obtained as shown in Table 4.15:

Power Flow Solution by Gauss-Seidel Method

Maximum Power Mismatch = 9.67136e-005

No. of Iterations = 74

Table 4.15: Power Flow Solution by Gauss-Seidel Method

Bus No.	Voltage Magnitude	Angle Degree	Load		Generation		Injected Mvar
			MW	Mvar	MW	Mvar	
1	1.025	0.000	51.000	41.000	719.501	224.014	4.000
2	1.020	-0.931	22.000	15.000	79.000	125.352	0.000
3	1.035	-4.212	64.000	50.000	20.000	63.030	0.000
4	1.050	-3.582	25.000	10.000	100.000	49.222	2.000
5	1.045	1.129	50.000	30.000	300.000	124.466	5.000
6	0.999	-2.573	76.000	29.000	0.000	0.000	2.000
7	0.994	-3.204	0.000	0.000	0.000	0.000	0.000
8	0.997	-3.298	0.000	0.000	0.000	0.000	0.000
9	1.009	-5.393	89.000	50.000	0.000	0.000	3.000
10	0.989	-5.561	0.000	0.000	0.000	0.000	0.000
11	0.997	-3.218	25.000	15.000	0.000	0.000	1.500
12	0.993	-4.691	89.000	48.000	0.000	0.000	2.000
13	1.014	-4.430	31.000	15.000	0.000	0.000	0.000
14	1.000	-5.040	24.000	12.000	0.000	0.000	0.000
15	0.991	-5.538	70.000	31.000	0.000	0.000	0.000
16	0.983	-5.882	55.000	27.000	0.000	0.000	0.000
17	0.987	-4.985	78.000	38.000	0.000	0.000	0.000
18	1.007	-1.866	153.000	67.000	0.000	0.000	0.000
19	1.004	-6.397	75.000	15.000	0.000	0.000	5.000

20	0.980	-6.025	48.000	27.000	0.000	0.000	0.000
21	0.997	-5.778	46.000	23.000	0.000	0.000	0.000
22	0.978	-6.437	45.000	22.000	0.000	0.000	0.000
23	0.976	-7.087	25.000	12.000	0.000	0.000	0.000
24	0.968	-7.347	54.000	27.000	0.000	0.000	0.000
25	0.974	-6.774	28.000	13.000	0.000	0.000	0.000
26	1.015	-1.803	40.000	20.000	60.00	32.706	0.000

Total Load (MW) = 1263.000
Total Load (Mvar) = 637.000

Total Generation (MW) = 1278.501
Total Generation (Mvar) = 618.790

Injected Mvar = 25.000

Total System Loss = 3.05252 MW

Total Generation Cost = 1633.25 $/h

Then, the simulation continues by using gradient method to find the Optimal Dispatch of Generation.

The analysis will repeat until Dpslack is near to 0.001. The Table 4.16 is shown the result of analysis.

Dpslack is the different (absolute value) between the scheduled slack generation determined from the coordination equation and the slack generation obtains from power flow solution [5].

Table 4.16: Result from gradient method

Dpslack (pu)	0.1369	0.0042	0.0008
Lambda ($/MWh)	13.526774	13.53749	13.539276
P1	446.8365	447.539	447.6823
P2	171.9396	172.712	172.8251
P3	262.4054	263.9503	264.0564
P4	138.1795	138.7387	138.8389
P5	169.2398	165.8622	165.3565
P26	86.941	86.9393	87.0322
Total Generation (MW)	1275.5418	1275.7415	1275.7914
Total Load (MW)	1263	1263	1263
Total Loss (MW)	12.5418	12.7415	12.7914

After the slack mismatch is reach to 0.001, the new power flow solution as shown in Table 4.17 and optimal dispatch of generation is obtained:

Optimal Dispatch of Generation:

P1 = 447.6823

P2 = 172.8251

P3 = 264.0564

P4 = 138.8389

P5 = 165.3565

P6 = 87.0322

Total system loss = 12.7915 MW

Power Flow Solution by Gauss-Seidel Method

Maximum Power Mismatch = 9.08821e-005

No. of Iterations = 11

Table 4.17: New Power Flow Solution by Gauss-Seidel Method

Bus No.	Voltage Magnitude	Angle Degree	Load		Generation		Injected Mvar
			MW	Mvar	MW	Mvar	
1	1.025	0.000	51.000	41.000	447.606	252.219	4.000
2	1.020	-0.198	22.000	15.000	172.712	73.728	0.000
3	1.040	-0.593	64.000	50.000	263.950	54.808	0.000
4	1.050	-2.107	25.000	10.000	138.739	38.761	2.000
5	1.045	-1.467	50.000	30.000	165.862	143.782	5.000
6	1.000	-2.882	76.000	29.000	0.000	0.000	2.000
7	0.995	-2.410	0.000	0.000	0.000	0.000	0.000
8	0.998	-2.282	0.000	0.000	0.000	0.000	0.000
9	1.010	-4.393	89.000	50.000	0.000	0.000	3.000
10	0.990	-4.318	0.000	0.000	0.000	0.000	0.000
11	0.998	-2.832	25.000	15.000	0.000	0.000	1.500
12	0.994	-3.288	89.000	48.000	0.000	0.000	2.000
13	1.018	-1.224	31.000	15.000	0.000	0.000	0.000
14	1.005	-2.417	24.000	12.000	0.000	0.000	0.000
15	0.996	-3.208	70.000	31.000	0.000	0.000	0.500
16	0.988	-3.971	55.000	27.000	0.000	0.000	0.000
17	0.982	-4.391	78.000	38.000	0.000	0.000	0.000
18	1.007	-1.888	153.000	67.000	0.000	0.000	0.000
19	1.004	-6.086	75.000	15.000	0.000	0.000	5.000
20	0.982	-4.763	46.000	27.000	0.000	0.000	0.000
21	0.977	-5.424	46.000	23.000	0.000	0.000	0.000
22	0.979	-5.334	45.000	22.000	0.000	0.000	0.000
23	0.977	-6.398	25.000	12.000	0.000	0.000	0.000
24	0.968	-6.685	54.000	27.000	0.000	0.000	0.000

25	0.975	-6.268	28.000	13.000	0.000	0.000	0.000
26	1.015	-0.289	40.000	20.000	86.939	27.100	0.000

Total Load (MW) = 1263.000

Total Load (Mvar) = 637.000

Total Generation (MW) = 1275.808

Total Generation (Mvar) = 591.397

Total Injected Mvar = 25.000

Total Generation Cost = 15447.62 $/h

4.3.2.3 Fast Decouple Method

Firstly, the power flow solution is obtained as shown in Table 4.18:

Power Flow Solution by Fast Decoupled Method

Maximum Power Mismatch = 8.40013e-005

No. of Iterations = 26

Table 4.18: Power Flow Solution by Fast Decoupled Method

Bus No.	Voltage Magnitude	Angle Degree	Load		Generation		Injected Mvar
			MW	Mvar	MW	Mvar	
1	1.025	0.000	51.000	41.000	719.501	224.014	4.000
2	1.020	-0.931	22.000	15.000	79.000	125.352	0.000
3	1.035	-4.212	64.000	50.000	20.000	63.030	0.000
4	1.050	-3.582	25.000	10.000	100.000	49.222	2.000
5	1.045	1.129	50.000	30.000	300.000	124.466	5.000
6	0.999	-2.573	76.000	29.000	0.000	0.000	2.000
7	0.994	-3.204	0.000	0.000	0.000	0.000	0.000
8	0.997	-3.298	0.000	0.000	0.000	0.000	0.000
9	1.009	-5.393	89.000	50.000	0.000	0.000	3.000
10	0.989	-5.561	0.000	0.000	0.000	0.000	0.000
11	0.997	-3.218	25.000	15.000	0.000	0.000	1.500
12	0.993	-4.691	89.000	48.000	0.000	0.000	2.000
13	1.014	-4.430	31.000	15.000	0.000	0.000	0.000
14	1.000	-5.040	24.000	12.000	0.000	0.000	0.000
15	0.991	-5.538	70.000	31.000	0.000	0.000	0.000
16	0.983	-5.882	55.000	27.000	0.000	0.000	0.000
17	0.987	-4.985	78.000	38.000	0.000	0.000	0.000
18	1.007	-1.866	153.000	67.000	0.000	0.000	0.000
19	1.004	-6.397	75.000	15.000	0.000	0.000	5.000

20	0.980	-6.025	48.000	27.000	0.000	0.000	0.000
21	0.997	-5.778	46.000	23.000	0.000	0.000	0.000
22	0.978	-6.437	45.000	22.000	0.000	0.000	0.000
23	0.976	-7.087	25.000	12.000	0.000	0.000	0.000
24	0.968	-7.347	54.000	27.000	0.000	0.000	0.000
25	0.974	-6.774	28.000	13.000	0.000	0.000	0.000
26	1.015	-1.803	40.000	20.000	60.00	32.706	0.000

Total Load (MW) = 1263.000
Total Load (Mvar) = 637.000

Total Generation (MW) = 1278.534
Total Generation (Mvar) = 618.791

Total Injected Mvar = 25.000

Total Generation Cost = 16760.73 $/h

Then, the simulation continues by using gradient method to find the Optimal Dispatch of Generation.

The analysis will repeat until Dpslack is near to 0.001. The Table 4.19 is shown the result of analysis.

Dpslack is the different (absolute value) between the scheduled slack generation determined from the coordination equation and the slack generation obtains from power flow solution [5].

Table 4.19: Result from gradient method

Dpslack (pu)	2.4541	0.1366	0.0045	0.0008
Lambda ($/MWh)	13.91179	13.526776	13.537491	13.539276
P1	474.1203	446.8361	447.5385	447.6823
P2	173.7889	171.9396	172.712	172.8251
P3	190.9497	262.4054	263.9503	264.0564
P4	150	138.1796	138.7388	138.8389
P5	196.7201	169.2401	165.8635	165.3564
P26	103.5777	86.9411	86.9394	87.0322
Total Generation (MW)	1289.1567	1275.5419	1275.7425	1275.7913
Total Load (MW)	1263	1263	1263	1263
Total Loss (MW)	26.1567	12.5419	12.7425	12.7913

After the slack mismatch is reach to 0.001, the new power flow solution as shown in Table 4.20 and optimal dispatch of generation is obtained:

Optimal Dispatch of Generation:

P1 = 447.6823 MW
P2 = 172.8251 MW
P3 = 264.0564 MW
P4 = 138.8389 MW
P5 = 165.3564 MW
P26 = 87.0322 MW

Total system loss = 12.7915 MW

Power Flow Solution by Fast Decoupled Method

Maximum Power Mismatch = 6.83751e-005

No. of Iterations = 9

Table 4.20: New Power Flow Solution by Fast Decoupled Method

Bus No.	Voltage Magnitude	Angle Degree	Load		Generation		Injected Mvar
			MW	Mvar	MW	Mvar	
1	1.025	0.000	51.000	41.000	447.606	252.218	4.000
2	1.020	-0.198	22.000	15.000	172.712	73.728	0.000
3	1.040	-0.593	64.000	50.000	263.950	54.808	0.000
4	1.050	-2.107	25.000	10.000	138.739	38.761	2.000
5	1.045	-1.467	50.000	30.000	165.862	143.782	5.000
6	1.000	-2.882	76.000	29.000	0.000	0.000	2.000
7	0.995	-2.410	0.000	0.000	0.000	0.000	0.000
8	0.998	-2.282	0.000	0.000	0.000	0.000	0.000
9	1.010	-4.393	89.000	50.000	0.000	0.000	3.000
10	0.990	-4.318	0.000	0.000	0.000	0.000	0.000
11	0.998	-2.832	25.000	15.000	0.000	0.000	1.500
12	0.994	-3.289	89.000	48.000	0.000	0.000	2.000
13	1.018	-1.224	31.000	15.000	0.000	0.000	0.000
14	1.005	-2.417	24.000	12.000	0.000	0.000	0.000
15	0.996	-3.208	70.000	31.000	0.000	0.000	0.500
16	0.988	-3.971	55.000	27.000	0.000	0.000	0.000
17	0.982	-4.391	78.000	38.000	0.000	0.000	0.000
18	1.007	-1.888	153.000	67.000	0.000	0.000	0.000
19	1.004	-6.086	75.000	15.000	0.000	0.000	5.000
20	0.982	-4.763	46.000	27.000	0.000	0.000	0.000
21	0.977	-5.424	46.000	23.000	0.000	0.000	0.000
22	0.979	-5.334	45.000	22.000	0.000	0.000	0.000
23	0.977	-6.398	25.000	12.000	0.000	0.000	0.000
24	0.968	-6.685	54.000	27.000	0.000	0.000	0.000

25	0.975	-6.268	28.000	13.000	0.000	0.000	0.000
26	1.015	-0.289	40.000	20.000	86.939	27.100	0.000

Total Load (MW) = 1263.000

Total Load (Mvar) = 637.000

Total Generation (MW) = 1275.809

Total Generation (Mvar) = 591.397

Total Injected Mvar = 25.000

Total Generation Cost = 15447.62 $/h

4.3.3 Discussion & Analysis of Simulation

Table 1 and Table 2 show the result for economic dispatch for 5-busbar and 26-busbar system using Newton-Raphson Method, Gauss Siedel Method and Fast Decouple Method.

4.3.3.1 Result for 5-Busbar System

Table 4.21: Result from simulation using MATLAB for 5-busbar system with 3 generators

UNIT	NEWTON-RAPHSON METHOD	GAUSS-SIEDEL METHOD	FAST DECOUPLE METHOD
P1 (MW)	23.5581	23.5667	23.5531
P2 (MW)	69.5593	69.5553	69.5615
P3 (MW)	59.0368	59.0320	59.0397
Loss (MW)	2.25434	2.15415	2.15449
Total of Iteration	2	8	2
Cost ($/h)	1596.96	1596.96	1596.97
Power Mismatch	$1.90285e^{-08}$	$7.5229e^{-05}$	$7.27225e^{-05}$

As shown at Table 1, all the method had obtain the quite similar result except for total of iteration for Gauss Seidel Method. Gauss Seidel Method reached the optimum solution in 8 iteration compared with the others

4.3.2.2 Result for 26-Busbar System

Table 4.22: Result from simulation using MATLAB for 26-busbar system with 6 generators

UNIT	NEWTON-RAPHSON METHOD	GAUSS-SIEDEL METHOD	FAST DECOUPLE METHOD
P1 (MW)	447.6919	447.6823	447.6823
P2 (MW)	173.1938	172.8251	172.8251
P3 (MW)	263.4859	264.0564	264.0564
P4 (MW)	138.8142	138.8389	138.8389
P5 (MW)	165.5884	165.3565	165.3564
P6 (MW)	87.0260	87.0322	87.0322
Loss (MW)	12.8003	12.7915	12.7915
Total of Iteration	8	85	35

Cost ($/h)	15447.72	15447.62	15447.62
Power Mismatch	2.33783e^{-05}	9.0.8821e^{-05}	6.83751e^{-05}

From the result obtain form Table 2, it can be proved that Newton-Raphson Method is best effective method for solving optimal power flow problem. Data in the Table 2 show that only Newton-Raphson Method reached the optimum solution in just 8 iteration compared with the others. For large power systems, the Newton-Raphson Method is found to be more efficient and practical. [5]

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The simulation and development of Optimal Dispatch of Power Generation Software Package has been presented in this project. The development of the user-friendly MATLAB GUI using MATLAB GUIDE was done after a detail of study and analysis. Through the development of this project it has conclude that the MATLAB GUI is a good medium to develop the friendly software package to analyze optimal power flow problem. The GUI seems easy to develop using simple pushbutton but it needs more knowledge and effort to do advanced programming on MATLAB GUI.

The objective of this project is to build a user friendly software package using MATLAB GUI to analyze optimal power flow problem was successfully achieved.

5.2 Future Recommendations

For future recommendations to improve this project, other features on GUI could be done like adding some music during the analysis to make it more interesting. The appearance and the shape of GUI could be improved to make it more users friendly and look matured. In part of analysis, the various method of optimal power flow analysis also could be added like lambda iteration method, linear programming method (LPOPF) or interior point method to give it more scope of study to the researcher.

5.2.1 Commercialization

This project and thesis is meant for the academic motivation of all electrical engineering students especially in teaching and learning session in UMP. All the lecturers of UMP are allowed to use this project and thesis for teaching session. I give full of authorities to my supervisor Mrs. Norhafidzah binti Mohd Saad to handle the matters that come further about the usefulness of my project.

REFERENCES

[1] X. S. Han, H. B. Gooi., Dynamic Economic Dispatch: Feasible and Optimal Solutions, 2001

[2] Chapman, Stephen J., MATLAB Programming for Engineers, Brooks Cole, 2001.

[3] Creating Graphical User Interfaces, Version 7: The Math Work

[4] Hadi Saadat Website. Power System Analysis Published Book. Available at: http://people.msoe.edu/~saadat/psabook.htm.

[5] Hadi Saadat, Power System Analysis(2nd Edition): Mc Graw Hill

[6] C.E Lin, S.T Chen., A Direct Newton Raphson Economic Dispatch, 1992

[7] Joong-Rin Shin, Wook-Hwa Lee., A Window-based Interactive and Graphic Package for Education and Training of Power System Analysis and Operation, 1999

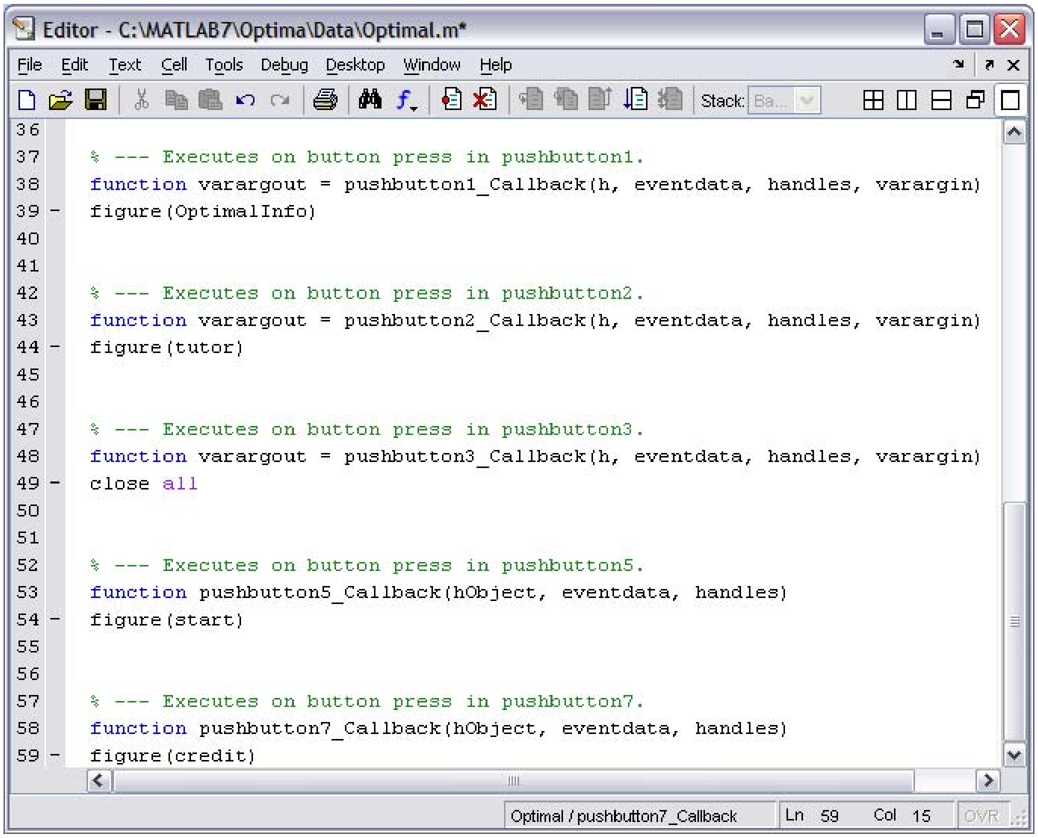
[8] Duane Hanselman, Bruce Littefield, MASTERING MATLAB 7: Pearson/ Prentice Hall, 2005

Appendix A

To generate the main page of this software package, two file was needed. The first file is Optimal.m and the second file is Optimal.fig. The Optimal.fig is the layout of the GUI and Optimal.m is a programming of the layout.

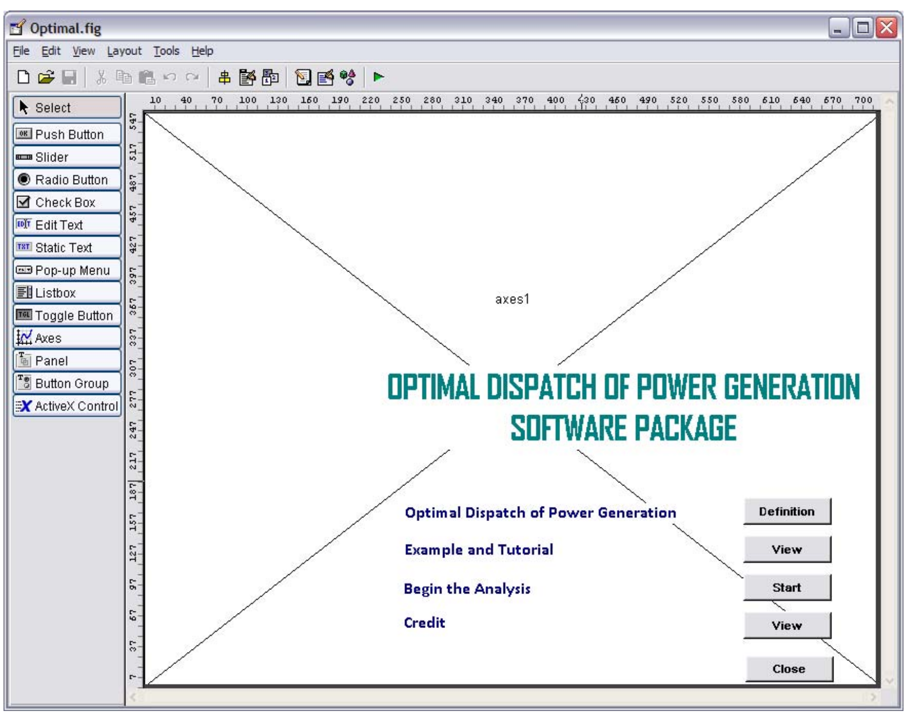
<u>Optimal.m</u>

```
function varargout = Optimal(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Optimal_OpeningFcn, ...
                   'gui_OutputFcn',  @Optimal_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end


function Optimal_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

guidata(hObject, handles);

movegui ('center')

function varargout = Optimal_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

[x,map] = imread('front','jpg');
image(x)
set(gca,'visible','off')
```



Editor - C:\MATLAB7\Optima\Data\Optimal.m*
File Edit Text Cell Tools Debug Desktop Window Help
Stack: Ba...
% --- Executes on button press in pushbutton1.
function varargout = pushbutton1_Callback(h, eventdata, handles, varargin)
figure(OptimalInfo)
% --- Executes on button press in pushbutton2.
function varargout = pushbutton2_Callback(h, eventdata, handles, varargin)
figure(tutor)
% --- Executes on button press in pushbutton3.
function varargout = pushbutton3_Callback(h, eventdata, handles, varargin)
close all
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
figure(start)
% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
figure(credit)
Optimal / pushbutton7_Callback
Ln 59
Col 15
OVR

Optimal.fig



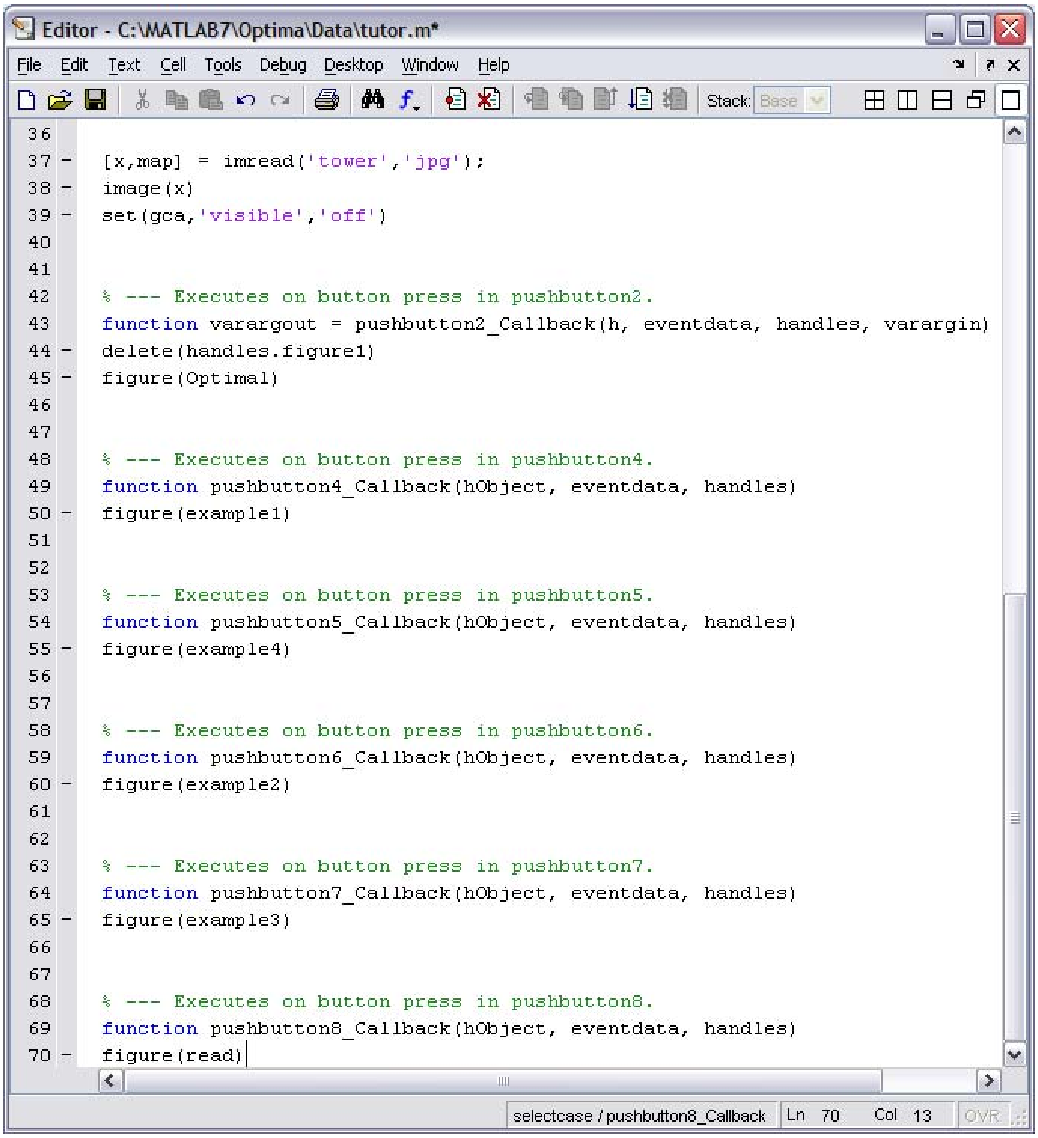
Optimal.fig
File Edit View Layout Tools Help
Select
Push Button
Slider
Radio Button
Check Box
Edit Text
Static Text
Pop-up Menu
Listbox
Toggle Button
Axes
Panel
Button Group
ActiveX Control
axes1
OPTIMAL DISPATCH OF POWER GENERATION
SOFTWARE PACKAGE
Optimal Dispatch of Power Generation
Definition
Example and Tutorial
View
Begin the Analysis
Start
Credit
View
Close

Appendix B

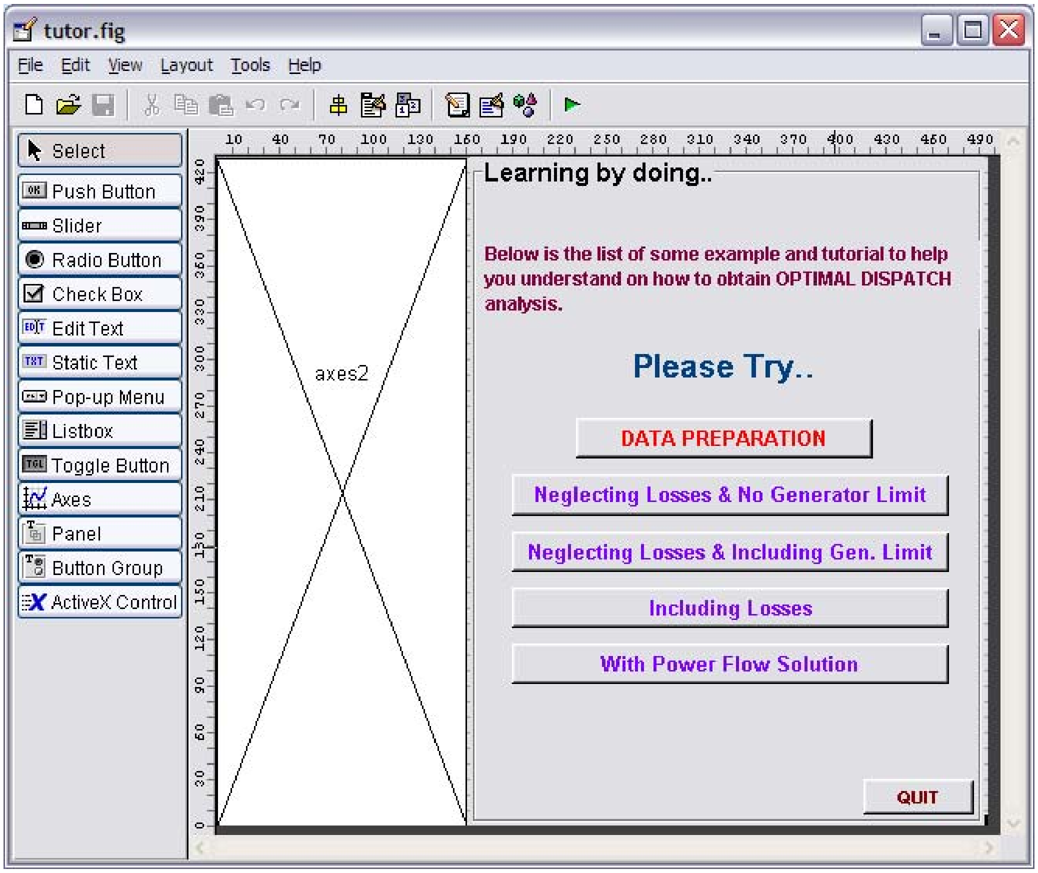
This two file is needed to generate the Example & Tutorial window in the software package. The first file is tutor.m and the second file is tutor.fig. The tutor.fig is the layout of the GUI and tutor.m is a programming of the layout.

tutor.m

```
function varargout = selectcase(varargin)

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @tutor_OpeningFcn, ...
                   'gui_OutputFcn',  @tutor_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before tutor is made visible.
function tutor_OpeningFcn(hObject, eventdata, handles, varargin)

movegui('center')

handles.output = hObject;

guidata(hObject, handles);


function varargout = tutor_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;
```


Editor - C:\MATLAB7\Optima\Data\tutor.m*
File Edit Text Cell Tools Debug Desktop Window Help
Stack: Base
[x,map] = imread('tower','jpg');
image(x)
set(gca,'visible','off')
% --- Executes on button press in pushbutton2.
function varargout = pushbutton2_Callback(h, eventdata, handles, varargin)
delete(handles.figure1)
figure(Optimal)
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
figure(example1)
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
figure(example4)
% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
figure(example2)
% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
figure(example3)
% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
figure(read)
selectcase / pushbutton8_Callback
Ln 70
Col 13
OVR

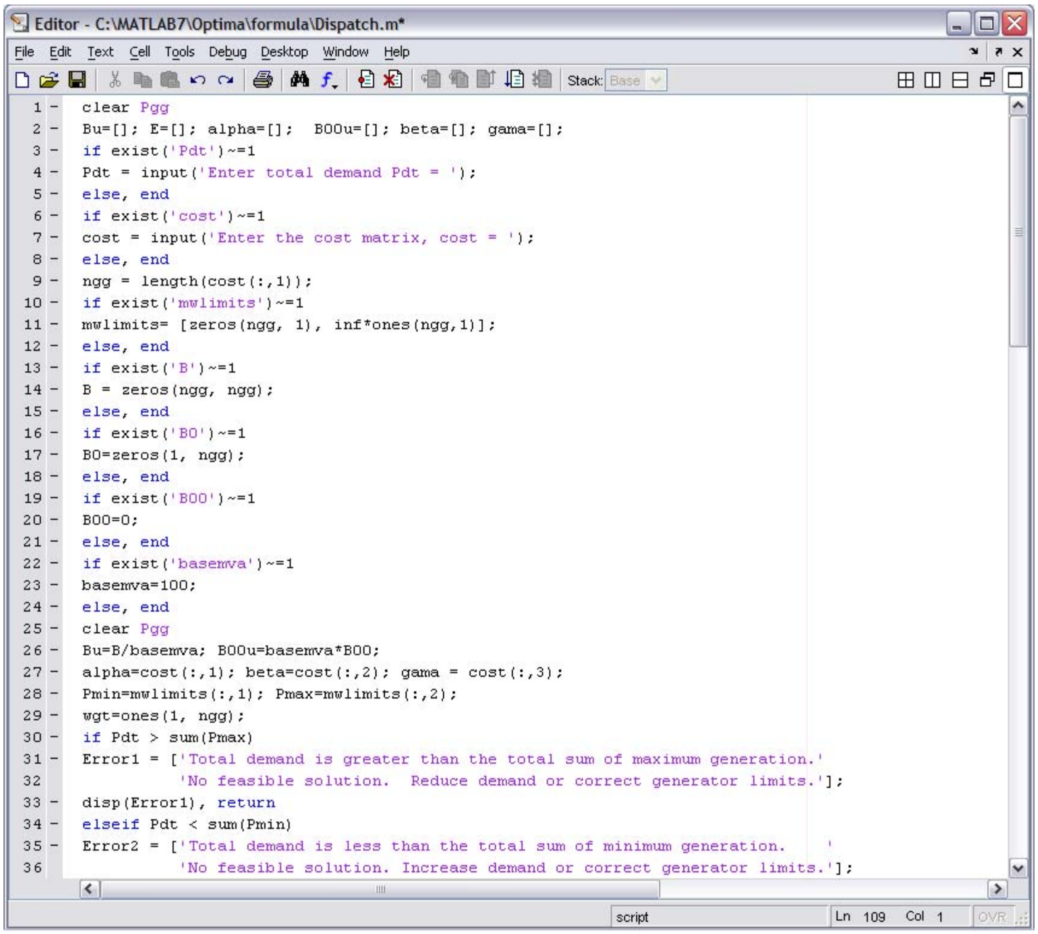
tutor.fig


tutor.fig
File Edit View Layout Tools Help
Select
Push Button
Slider
Radio Button
Check Box
Edit Text
Static Text
Pop-up Menu
Listbox
Toggle Button
Axes
Panel
Button Group
ActiveX Control
axes2
Learning by doing..
Below is the list of some example and tutorial to help you understand on how to obtain OPTIMAL DISPATCH analysis.
Please Try..
DATA PREPARATION
Neglecting Losses & No Generator Limit
Neglecting Losses & Including Gen. Limit
Including Losses
With Power Flow Solution
QUIT

Appendix C

M.file below is the example of formula that generated the optimal dispatch of power generation analysis result. This m.file have to combined with others m.file formula and GUI programming to get the result like in sofware package.

<u>Dispatch.m</u>



```
clear Pgg
Bu=[]; E=[]; alpha=[];  B00u=[]; beta=[]; gama=[];
if exist('Pdt')~=1
Pdt = input('Enter total demand Pdt = ');
else, end
if exist('cost')~=1
cost = input('Enter the cost matrix, cost = ');
else, end
ngg = length(cost(:,1));
if exist('mwlimits')~=1
mwlimits= [zeros(ngg, 1), inf*ones(ngg,1)];
else, end
if exist('B')~=1
B = zeros(ngg, ngg);
else, end
if exist('B0')~=1
B0=zeros(1, ngg);
else, end
if exist('B00')~=1
B00=0;
else, end
if exist('basemva')~=1
basemva=100;
else, end
clear Pgg
Bu=B/basemva; B00u=basemva*B00;
alpha=cost(:,1); beta=cost(:,2); gama = cost(:,3);
Pmin=mwlimits(:,1); Pmax=mwlimits(:,2);
wgt=ones(1, ngg);
if Pdt > sum(Pmax)
Error1 = ['Total demand is greater than the total sum of maximum generation.'
          'No feasible solution.  Reduce demand or correct generator limits.'];
disp(Error1), return
elseif Pdt < sum(Pmin)
Error2 = ['Total demand is less than the total sum of minimum generation.    '
          'No feasible solution. Increase demand or correct generator limits.'];
```

```
disp(Error2), return
else, end
iterp = 0;                                  % Iteration counter
DelP = 10;                  % Error in DelP is set to a high value

E=Bu;
if exist('lambda')~=1
lambda=max(beta);
end
while abs(DelP)  >= 0.0001  & iterp < 200     % Test for convergence
iterp = iterp + 1;                           % No. of iterations
for k=1:ngg
    if wgt(k) == 1
       E(k,k) = gama(k)/lambda + Bu(k,k);
       Dx(k) = 1/2*(1 - B0(k)- beta(k)/lambda);
       else, E(k,k)=1;  Dx(k) = 0;
            for m=1:ngg
               if m~=k
               E(k,m)=0;
               else,end
             end
       end
end
PP=E\Dx';
for k=1:ngg
if wgt(k)==1
   Pgg(k) = PP(k);
   else,end
end
Pgtt = sum(Pgg);
PL=Pgg*Bu*Pgg'+B0*Pgg'+B00u;
DelP =Pdt+PL -Pgtt ;                                  %Residual
for k = 1:ngg
   if Pgg(k) > Pmax(k) & abs(DelP) <=0.001,
   Pgg(k) = Pmax(k); wgt(k) = 0;
   elseif Pgg(k) < Pmin(k) & abs(DelP) <= 0.001
   Pgg(k) = Pmin(k); wgt(k) = 0;
   else, end
end
PL=Pgg*Bu*Pgg'+B0*Pgg'+B00u;
DelP =Pdt +PL - sum(Pgg);                              %Residual
  for k=1:ngg
  BP = 0;
     for m=1:ngg
         if m~=k
            BP = BP + Bu(k,m)*Pgg(m);
            else, end
     end
  grad(k)=(gama(k)*(1-B0(k))+Bu(k,k)*beta(k)-2*gama(k)*BP)/(2*(gama(k)+lambda*Bu(k,k))^2);
  end
sumgrad=wgt*grad';
Delambda = DelP/sumgrad;                 % Change in variable
```

Editor - C:\MATLAB7\Optima\formula\Dispatch.m*

```
lambda = lambda + Delambda;              % Successive solution
end
fprintf('Incremental cost of delivered power (system lambda) = %9.6f $/MWh \n', lambda);
fprintf('Optimal Dispatch of Generation:\n\n')
disp(Pgg')
fprintf('Total system loss = %g MW \n\n', PL)
ng=length(Pgg);
n=0;
if exist('nbus')==1 | exist('busdata')==1
  for k=1:nbus
     if kb(k)~=0
        n=n+1;
        if n <= ng
        busdata(k,7)=Pgg(n); else, end
     else , end
  end
  if n == ng
     for k=1:nbus
         if kb(k)==1
         dpslack = abs(Pg(k)-busdata(k,7))/basemva;
fprintf('Absolute value of the slack bus real power mismatch, dpslack = %8.4f pu \n', dpslack)
         else, end
     end
   else, end
 else, end
clear BP Dx DelP Delambda E PP grad sumgrad wgt Bu B00u B B0 B00
```

script   Ln 109   Col 1   OVR