Development of a New Loss Prediction Method in a Deregulated Power Market using Proportional Sharing

By

TEE PING HONG

DISSERTATION

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfilment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

MAY 2011

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Development of a New Loss Prediction Method in a Deregulated Power Market using Proportional Sharing

by

Tee Ping Hong

A Dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
Electrical & Electronics Engineering

Approved by,

Ir. Perumal Nallagownden
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

May 2011

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

TEE PING HONG

ABSTRACT

The advent of worldwide deregulation of power industry and unbundling of transmission services have resulted in the need to measure the flow of power primarily for pricing and tariff purposes. Reliable analysis and prediction of electrical losses is therefore attracting more attention than ever for efficient power system management.

This report first presents the background for deregulation energy power industry, comparing both traditional utility structure and the deregulated market model. Problems arise when new transmission loss allocation method is strictly needed for promoting fair competition in deregulated power market, which lead to the determination of the objectives and scope of study. Literature review is then done to discuss nodal pricing in current deregulated markets, which followed by discussion on several existing loss allocation methods. Then, the response of each individual generator in IEEE 24-bus RTS to change in demand and the corresponding associated losses are also presented in the result and discussion section. Corresponding losses to each bus using different loss allocation methods are then plotted for critical analysis. Eventually, the loss allocation method which provides better indicative measure to promote efficient network usage will be adopted for loss prediction.

To date, a novel approach for the loss prediction has been developed for the deregulated power market with the insight from Euler's Method. The effectiveness of the proposed approach is demonstrated by visual comparison and strict statistical criterion and the results suggest that it proves to be astoundingly accurate for the prediction of system losses and the associated bus losses. MATLAB with matpower4.0b extension was used to present the study on IEEE 24bus RTS and a larger IEEE 57bus test case.

ACKNOWLEDGEMENTS

My profound thanks to my supervisor, Ir. Perumal who consistently provided useful guidelines and advices coupled with his dedication to every advisory session. The project would not have been possible if not for the generous assistance of my supervisor who shared his knowledge and expertise.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background of study

The pricing of electricity has always been a major concern to system participants, even before the introduction of deregulation. The previous monopolistic structure used a simple pricing scheme based on a uniform distribution of the approximated loss of 2% to 5% of generated power. This simple loss allocation, however, is not sufficient for the restructured market as it does not encourage competition between market participants. Given that healthy competition should encourage lower prices, it is important to develop an electricity-pricing scheme that promotes competition.

To promote fair competition, market participants must be charged in a way that reflects their use of the system. A critical part of this is distribution of system losses to the market participant. Presently, some electricity markets such as in mainland Spain and Brazil have adopted a *pro rata* approach to loss sharing [1], while other markets such as in Australia [2] and New Zealand [3] have adopted the incremental method. Yet these present methods are not felt to be completely satisfactory leading some markets, such as Brazil, to consider implementing alternative approaches [4].

These allocation methods have been critically analyzed and tested on various systems. Previous work has showed that *pro rata* method exhibits fairly similar loss distribution characteristics to the incremental method [5].

This result questions the necessity of implementing incremental method when the simple *pro rata* can produce fairly similar results. Besides, *pro rata* and incremental method do not encourage competition in the electricity market because of their simple objective, which depends only on the power injection. Network dependent alternatives such as the proportional sharing would provide a better indicative measure to promote efficient network usage. Hence proportional sharing will be applied in this paper prior to the process of loss prediction.

1.2 Problem Statement

Today, deregulation not only reforms the way business transaction is performed, but also the generation companies' (genco) business strategy by only supplying sufficient power to meet the demand at a given point of time to prevent unrecovered generation cost [6]. This is replicated in retailers' approach by only purchasing sufficient power from these gencos to be sold to end users. Prediction hence plays an even more crucial role in a deregulated market since no genco will be willing to generate power in excess than what is demanded. The question now is how to harness the information provided by any of those tracing methods in performing efficient forecasting in a deregulated energy market.

1.3 Objectives

The objectives of the project are listed as follows:

- To analyze existing loss allocation methods and assess their suitability.
- To determine the type of market structure promoted by each method.
- To recognize the most reasonable loss allocation method based on the results obtained from power flow simulations and loss allocations using Matlab.
- To develop a novel approach for the prediction of system losses using the established loss allocation method.

1.4 Scope of Study

This project involved detailed study on transmission loss allocation in a deregulated power market. Several existing loss allocation methods such as *pro rata* allocation, incremental allocation and proportional sharing method approach are compared and discussed based on a few determining factors like network structures and level of competition promoted. Besides, nodal pricing in several electricity markets will be assessed. Then, these methods are implemented on the IEEE 14-bus test system to compare each losses distribution, before implemented on a larger and more realistic IEEE 24-bus Reliability Test System.

The later stage of the project is geared towards the research on different prediction approaches which will assist the works in developing a feasible loss prediction method. Results obtained will tested over and over again based on a strict statistical criterion and the end results desired is an accurate and feasible loss prediction method which would be deemed beneficial for every power market participant, for it can couple with load demand prediction to yield an electricity pricing prediction scheme..

CHAPTER 2

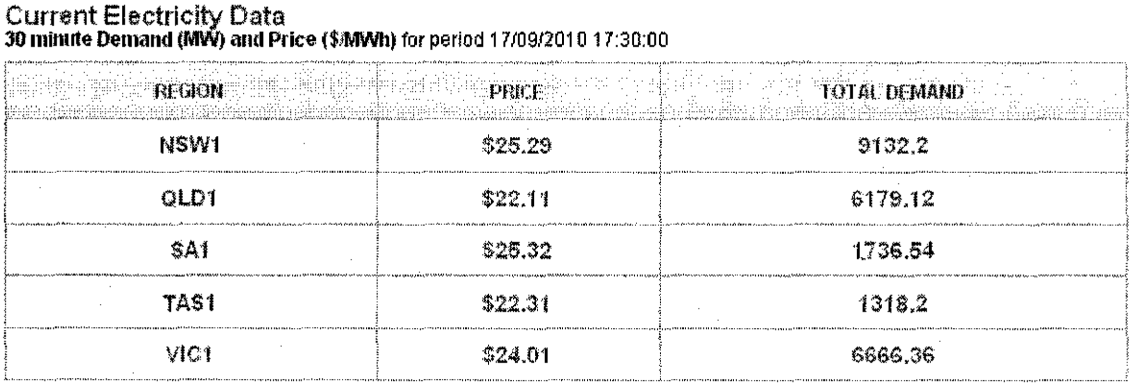
LITERATURE REVIEW

2.1 Deregulation in Australia

The introduction of deregulation in Australia comes into play when National Electricity Market (NEM) commenced the operation on 13 December 1998. The eastern and southern state of Australia has been included in the NEM, including New South Wales, Queensland, South Australia, the Australian Capital Territory and Victoria. Due to geographic reasons, Western and Northern Australia are not included. New interconnections are laid between the NEM regions (states) to promote competition. Theses interconnections also aimed at reducing price volatility as well as accommodating demand [7].

2.2 Nodal Pricing

In the NEM, a set of nodal prices for an electricity network is computed simultaneously for the states aforementioned, as depicted in Figure 1. The set of prices will be updated every 30 minutes by the Australian Energy Market Operator (AEMO) based on the marginal cost of supplying a small increment in demand at each location.



Current Electricity Data
30 minute Demand (MW) and Price ($/MWh) for period 17/09/2010 17:30:00

REGION	PRICE	TOTAL DEMAND
NSW1	$25.29	9132.2
QLD1	$22.11	6179.12
SA1	$25.32	1736.54
TAS1	$22.31	1318.2
VIC1	$24.01	6666.36

Figure 1 *Set of nodal prices*

The prices also account for influential factors such as costs of producing electricity, transmission loss and capacity limitations [8], which directly promote competitions between market participants as they are now charged in a way that reflects their use of the system. Locational pricing is divided into two parts, as shown in figure 2 [9]:

- *Intra-regional pricing:*

There are one regional reference node (RRN) and several transmission network connection points in an inter-regional node. Intra-regional pricing is thus the pricing between a RRN and any transmission network connection point within the node. It is based on historical network flow data from previous 12 months for each load and generation bus relative to the RRN analyzed. Marginal loss factors (MLF) are then calculated and averaged to arrive to a single weighted average MLF for each load and generation connection point is calculated from:

$$MLF = 1 + \frac{\partial\, loss}{\partial\, load\ increment} \tag{1}$$

$$MLF\ at\ RRN = 1 \tag{2}$$

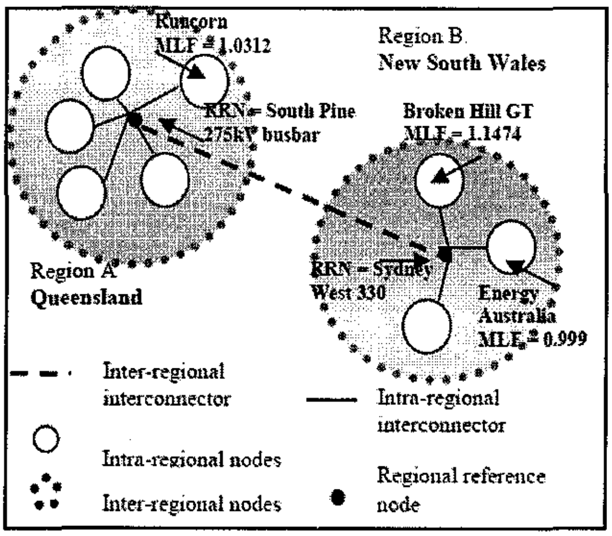


Figure 2 *Nodal pricing concept*

Limitation for this method is that there is a two year delay for a change in load or generation data before it takes the effect on the transmission loss factor. Therefore, a forward looking loss factors methodology is employed at 1 January 2004 throughout NEM. This method is based on the principle of "minimal extrapolation" [10]. This method takes into account of the effect of load growth. Changes in load impact the transmission flows and dispatch of generation. The static MLF for node "i" in region "j" is then defined as:

$$MLF = \frac{\sum_k \left(d_i^k \left(\frac{\alpha_i^k}{\alpha_{rj}^k} \right) \right)}{\sum_k d_i^k} \tag{3}$$

where,

α_i^k is the MLF of node "i" with respect to swing bus for trading period "k"

α_{rj}^k is the MLF of the reference node for region "j" with respect to swing bus for trading period "k".

d_i^k is the demand for node "i" for trading period "k".

- *Inter-regional pricing:*

It is the pricing between two RRNs, which is necessary to accommodate for the large and variable flows between RRNs. Heavily influenced by inter-regional marginal loss equations, it is obtained by applying linear regression to the set of hourly MLFs, based on DC approximations [7]. Nodal spot price at a particular location within the region is then calculated by multiplying spot price at the RRN by appropriate MLF. For example, Broken Hill GT in Figure 2 has the price of electricity 1.1474 times the price enacted at its RRN, Sydney.

2.3 Loss Allocation Methods

Existing loss allocation methods in current electricity market can be categorized as either *pro rata*, incremental, proportional sharing or loss formula methods, which will be outlined as follows.

2.3.1 Pro rata allocation

The *pro rata* allocation method [11] is the simplest loss allocation method. It assigns losses based on a comparison of the level of power or current injected or consumed by a specific generator or load to the total power generated or delivered in the system. Starting from solved load flow solution, losses are systematically distributed based on the real power injected or consumed at each node, as shown in (4) and (5).

$$L_{Gi} = \frac{P_{loss}}{x}\frac{P_{Gi}}{P_G} \tag{4}$$

$$L_{Dj} = \frac{P_{loss}}{x}\frac{P_{Dj}}{P_D} \tag{5}$$

Together equations (4) and (5) represent the *pro rata* allocation of losses to the generator at bus *i* and load at bus *j*. P_G is total real power generated in the system while P_{Gi} is the total MW output of the generators at bus *i*. Alternatively, P_D is total real power consumed and P_{Dj} is the real power consumed by loads of bus *j*. P_{loss} is the system transmission power losses. The multiplying factor x can be used to weight the distribution of system losses towards either of the market participants.

It is clear from (4) and (5) that this method is totally reliant on the power injections at buses and independent of the network topology. Losses are distributed across all buses, according to their level of generation or consumption only. Two loads in different locations but with identical demands will be allocated the same level of loss. Hence, this method will promote an unhealthy form of competition. Furthermore, no incentive is provided for placing generation closer to load centers, a practice which usually leads to reduced system losses. The *pro rata* method is also unable to trace power flows, making it difficult to justify the different allocations.

2.3.2 Incremental allocation

The incremental allocation [11] of loss sharing addresses how a small change in power injections at a single bus affects the overall system losses. The transmission system here is viewed as a black box with injection points connected to it. Loss coefficients are calculated based on the change in loss due to a change in a bus injection. Losses are then allocated to market participants using the loss coefficients. An incremental method [13] was implemented. It is a simple method that shows the fundamental features of the incremental method. The essence of the method is based on (6), where P_{loss} is the system transmission power losses, and Pi is the power injection at a particular load.

$$P_{loss} = \sum_{i=1}^{n} \frac{\partial P_{loss}}{\partial P_i} \qquad (6)$$

Individual loads are incremented sequentially from zero to full load. The change in losses was determined using a series of load flow calculations rather than solving (6) directly. At each step, losses obtained are allocated to the corresponding load (and generator if contracts are specified). The main limitation of this method is that losses are highly dependent on the incremental steps taken. It is expected then that a loss allocation would be non-unique.

2.3.3 *Proportional sharing allocation*

The proportional sharing method introduced by Bialek [12] is a topological tracing method, treating each node as an ideal mixer, in the way that power flowing out of a node can be considered the proportional sum of the power flowing into the node. This allows the demands of load to be traced "up" to the generators or the output of the generator to be traced "down" to the loads. To understand the allocation method, consider the tracing of power upstream, from the loads to the generating sources. Starting from a solved load flow solution, the power balance equation at node *i* considering the power inflows from "upstream" is defined by (7).

$$P_i^g = \sum_{j \epsilon \alpha_i^u} |P_{ij}^g| + P_{Gi} \qquad for\, i = 1, 2, \dots, n \tag{7}$$

P_i^g is the unknown gross nodal power flow through node *i*, P_{ij}^g is the unknown gross line flow in line *i-j*, whereas α_i^u is the set of nodes supplying node *i*, and $P_{Gi}$is the power generation in node *i*. The line flows, P_{ij}^g also can be expressed as a proportion of the flows into the upstream node j. By continuing this process, the contributions of system's generators to the *i*-th gross nodal power can be expressed according to (8).

$$P_i^g = \sum_{k=1}^{n} [A_u^{-1}]_{ik}\, P_{Gk} \tag{8}$$

$$[A_u^{-1}]_{ij} = \begin{cases} 1 & i = j \\ -\dfrac{|P_{ij}|}{P_j} & j \in \alpha_i^u \\ 0 & otherwise \end{cases}$$

A_u is the upstream distribution matrix and P_{Gk} is the generation at node k. In these cases, the gross nodal and line flows refer to those power flows in a lossless system. The difference between the gross and actual demand gives the loss allocated to a load. Unlike the previous two methods, the proportional sharing method is capable of approximating the contribution of each generator to each load through tracing the flow of power.

The assignment of losses to either generators or loads s could encourage the market participants to take corrective actions that will reduce their share of losses. In the context of competition, this theoretical understanding shows that this method will perhaps promote operational efficiency. The problem with this approach, however, is that the distribution of power flows is built on the proportional sharing principle, which lacks physical and economical justification, meaning that proposed strategies to reduce losses may not be technically satisfactory.

2.4 Prediction using Learning Coefficient by Regression Method

The usage of learning coefficients in determining generators' optimal dispatch has been established in [13], by approximating the heat-rate curve of generator with a generalized quadratic relationship in the form of $H(P_G) = \alpha/P_G + \beta + \gamma P_G$ [13], where α, β and γ will be solved for respectively by simultaneous equation.

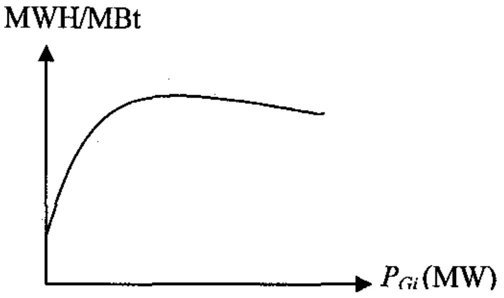


Figure 3 *Generator Heat Rate Curves [13]*

This is extended to the prediction of the (i) share of generations meeting a retailer's demand, and (ii) power loss in a transaction meeting a retailer's demand [14]. Both the relationships assume the form of the aforementioned heat-rate curve of a generator.

Two types of learning coefficients $(\alpha_1,\beta_1,\gamma_1)$ and $(\alpha_2,\beta_2,\gamma_2)$, need to be generated at each demand node of the network. Thus, there will be $(ng \times npg)$ sets of $(\alpha_1,\beta_1,\gamma_1)$ and $(ng \times npg)$ sets of $(\alpha_2,\beta_2,\gamma_2)$ to be generated using real time operating scenarios, during learning exercise. '*ng*' represents number of generations and '*npq*' represents number of retail or demand points .

Transmission losses is tacit within generated power for the reason that power supplied has to meet the demand of retailers, inclusive of losses. It is also clear that the greater the power demanded, the greater the generation level will be and subsequently the losses too, up to an allowable limit when no extra generation can take place, as observed from the heat rate curve.

2.3.1 *Learning relationship between a generator's contribution to a retailers demand at the receiving end*

$$\alpha_1/P_d + \beta_1 + \gamma_1 P_d = P_{gd} \tag{9}$$

where P_d is the total demand at a retailer's point of receipt in per unit (p.u.), P_{gd} is a generator's contribution to a retailers demand at the point of receipt in p.u.

2.3.2 *Learning relationship between a retailer's demand and the associated loss in a transaction*

$$\alpha_2/P_d + \beta_2 + \gamma_2 P_d = Loss_t \tag{10}$$

where P_d is the total demand at a retailer's point of receipt in p.u., $Loss_t$ is loss in a transaction in p.u. (The difference between a generation's contribution to a demand at the generation end and a generation's contribution to a demand at the load bus, which is the result of the Sending Algorithm less the result of Receiving Algorithm for the same generator and load).

2.3.3 *Generation of the coefficients*

Three unknowns require a minimum of three samples for the generation of one set of learning coefficients. The four equations presented above can then be solved respectively, in matrix form:

$$\begin{bmatrix}\alpha_1\\ \beta_1\\ \gamma_1\end{bmatrix} = \begin{bmatrix}\frac{1}{P_{d1}} & 1 & P_{d1}\\ \frac{1}{P_{d2}} & 1 & P_{d2}\\ \frac{1}{P_{d3}} & 1 & P_{d3}\end{bmatrix}^{-1} \begin{bmatrix}P_{gd1}\\ P_{gd2}\\ P_{gd3}\end{bmatrix} \tag{11}$$

$$\begin{bmatrix}\alpha_2\\ \beta_2\\ \gamma_2\end{bmatrix} = \begin{bmatrix}\frac{1}{P_{d1}} & 1 & P_{d1}\\ \frac{1}{P_{d2}} & 1 & P_{d2}\\ \frac{1}{P_{d3}} & 1 & P_{d3}\end{bmatrix}^{-1} \begin{bmatrix}Loss_{t1}\\ Loss_{t2}\\ Loss_{t3}\end{bmatrix} \tag{12}$$

For improved credibility in the learning coefficients, a higher number of samples are to be used [15]. As the number of samples increases, the learning coefficients will be determined using the regression method, where:

$$x = (A^TA)^{-1}A^Tb \tag{13}$$

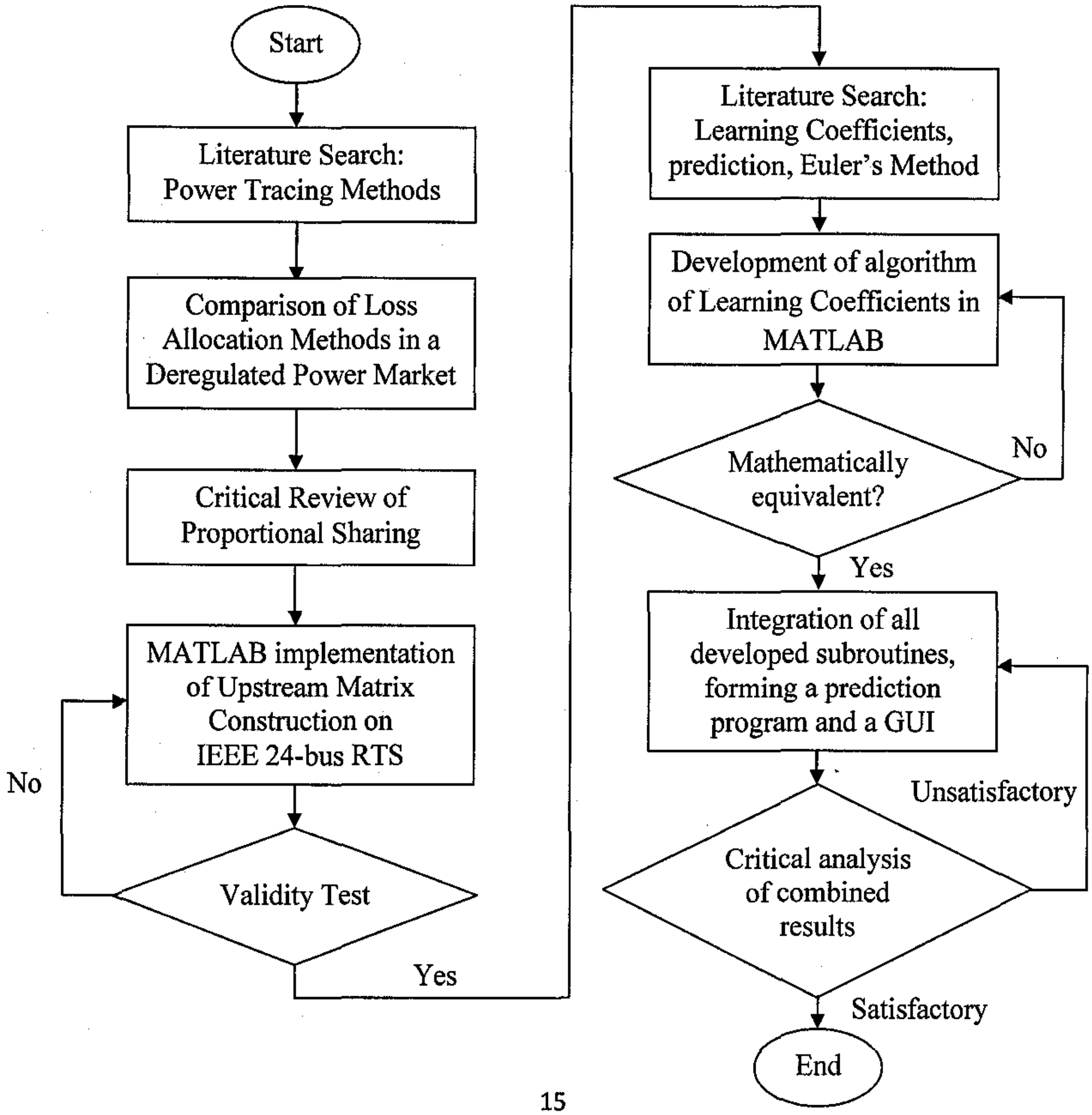
For matrices or the form $Ax = b$, addition of samples will elicit a row addition to A and b matrices, changing the dimensions to a rectangular matrix. Note that in (13), A and b are the load demand matrix and the loss matrix respectively. Meanwhile, x that is obtained using regression method is the desired learning coefficients, i.e. α, β and γ.

As the compared to the famed artificial neural network (ANN) that is commonly used to model complex relationships between inputs and outputs or to find patterns in data, learning coefficients required comparatively lesser data samples. It uses merely 10 samples for generation of learning coefficients to be used for loss prediction, which greatly reduce computational time in a complex power system.

CHAPTER 3

METHODOLOGY

3.1 Procedure Identification



The first part of the project is geared towards the familiarization of the deregulated energy market, power tracing concept, methods and applications. Application of different loss allocation methods on two distinct test system, i.e. IEEE 14-bus network and IEEE 24-bus reliability test system were then put under inspection to analyze the efficiency and effectiveness of respective methods in encouraging the competition between market participants, leading to some comments on the comparative behavior of the different approaches.

Building on the application of the tracing methodology in the deregulated energy market, issues such as congestion management, ATC and prediction are studied. The prediction methodology proposed in [14] was analyzed, with several areas identified for refinements. MATLAB implementation of the proportional sharing method is first implemented. The result is then verified manually to be mathematically correct.

For simplicity purposes, IEEE 14 bus test system is used as input data to perform power flow tracing. Upon verification, the programme is then tested on the IEEE 24 bus Reliability Test System and bigger IEEE 57 bus test system. Only upon ascertaining the credibility of the M-File, generation of learning coefficients is then realized through another M-File prior to the implementation of prediction method through the established algorithm which computes the required learning coefficients.

The challenge in the implementation of the learning coefficient method is the data handling and addressing in M-File, due to the sheer amount of learning coefficients to be dealt with, as described in the Literature Review section. Hence for purpose of clarity and simplicity only one relationship was picked to be examined, which is the relationship between a retailer's demand and the associated loss (in every buses) in a transaction. Prior to implementation in MATLAB, careful attention is given to efficient manipulation and handling of data using matrices to produce a minimal execution time M-File coding. Furthermore, load hourly demand from week 1 to week 52 is expressed as the percentage of full load, as the samples of load demand throughout the year are now scattered on the graph, as shown in Figure 9.

Next, the short term prediction of loss given any oncoming demand using the learning coefficients is done. This section involves in depth analytical and critical review of the learning coefficients and the regression method to devise a prediction algorithm, where no previous reference exist. For every 10% increment in load demand (refer to Figure 9) e.g. section of 30- 40%, 40- 50%, 50-60% etc., power flow simulation will be performed repeatedly on ten different samples of load demand falling in each respective section. For the purpose of simplicity, each section is now equally divided into 10 smaller sections, corresponding to 1% of full load demand before these 10 samples were taken for power flow simulation coupled with loss allocation using proportional sharing.

For the instance of section 50- 60%, the sample of 51% of full load is the first sample taken for power flow simulation and loss allocation. The end result using the first sample is a set of losses assigned to each buses and their corresponding load demand, providing us with one relationship for these two variables. The same process, i.e. power flow and loss allocation is then performed on the following nine samples: 52%, 53%, 54%... 60% of full load, yielding another nine sets of load demand and respective losses in each bus.

Next, 10 sets of load demand and bus losses will elicit one set of learning coefficients (for a particular bus in the system of interest) each for all 24 buses in the section 50- 60% of load demand, using regression method. Noted that the set of learning coefficients obtained is only for a particular bus, and if every bus in the system is interested for loss prediction, the same process will be performed on the bus of interest.

With the sets of learning coefficients obtained, system losses and associated bus losses can be predicted up to a certain accuracy, for any oncoming load hourly demand which falls under the section 50- 60% of full load. The same process shall be performed on other sections (60- 70%, 70-80%, etc.) as well in order to predict the losses for any load hourly demand which falls in the corresponding sections.

The performance of the devised prediction method is gauged in terms of the mean absolute percentage error (MAPE) define below:

$$\sum MEAN = \frac{1}{N}\sum_{i=1}^{N}\frac{|Actual\ (i) - \ Forecast\ (i)|}{Actual\ (i)} \times 100 \qquad (14)$$

In general, a MAPE of 10% is considered good, while a MAPE in the range 20-30% or even higher is quite common. Although a MAPE of 10% is quite common, but it is always desirable to acquire an accuracy of at least 97%. Thus, further prospect on reducing the MAPE to a more stringent 3% is envisioned to be greatly beneficial. Trial and error was done continuously until the criterion is satisfied.

Lastly, all the individual subroutines are integrated into a fully automated program as well as a MATLAB graphical user interface (GUI) which would prompt the user for the week and hour of the day for loss prediction. Upon receiving the required input, it will display the estimated breakdown of total losses for the predicted power of the particular load. The flow chart of the prediction program is as shown in the following figure.

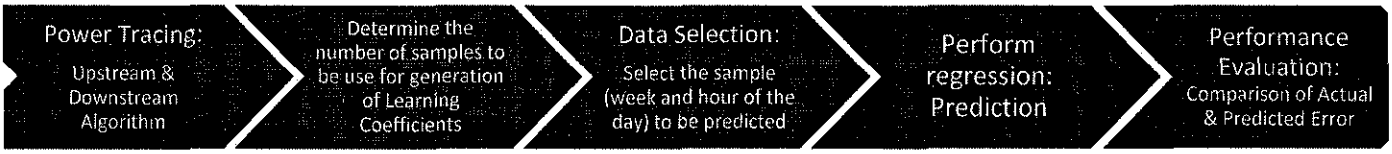


Figure 4 *Flow Chart of the Prediction Program*

3.2 Tools and equipments used

MATLAB is the sole software required. All simulations are carried out by first writing the codes in M-Files. The MatPower extension to MATLAB, develped by the Power Systems Engineering Research Centre (PSERC) is used to perform loads flow in written M-Files. The MatPower extension is free for download from PSERC's homepage.

With the detailed line, load and generator data, the power flow simulations on IEEE 14-bus, 24-bus RTS and 57-bus had been successful, providing an important information about their respective bus data and branch data. This is important for obtaining an accurate loss allocation using proportional sharing, which is useful and beneficial for the works on developing a loss prediction method.

Besides, Valerie Lim's papers on different transmission loss allocations methods had been useful throughout the research period. Other indispensible resources are Bialek's conference papers on the Proportional Tracing methodology and the book entitled FACTS: Modelling and Simulation in Power Networks by Enrique Archa et al.

3.3 IEEE 14-bus test system

The IEEE 14-bus system contains two generating sources, 14 loads and synchronous condensers and 12 shunt elements representing line capacitance and off-nominal transformers. It has been used regularly in other works to confirm the effectiveness of different loss allocation procedures. Previous work on loss allocation using different methods [5] had been conducted on this system.

Therefore it is desirable if similar result of loss allocation could be achieved. In other words, results from [5] could act as a validity test on the results of simulation which eventually lead to the allocation of system losses, especially the MATLAB power flow simulations as well as the algorithms of loss allocation methods. This is utmost important to serve as an assurance when all the research works are conducted on a larger and more realistic network, i.e. the IEEE 24-bus RTS and 56-bus test system which will be discussed in the following section.

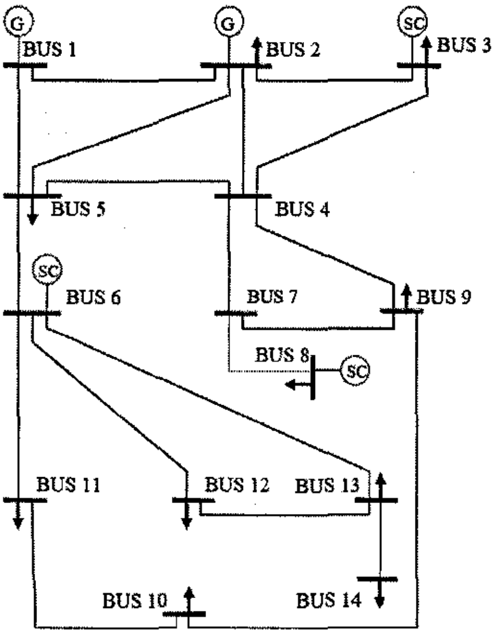


Figure 5 *IEEE 14-bus system*

3.4 IEEE 24-bus Reliability Test System

IEEE 24-bus RTS system [16] will be used to do power flows tracing and transmission loss allocation. The following figure shows the diagram of the system that will be used in the project.

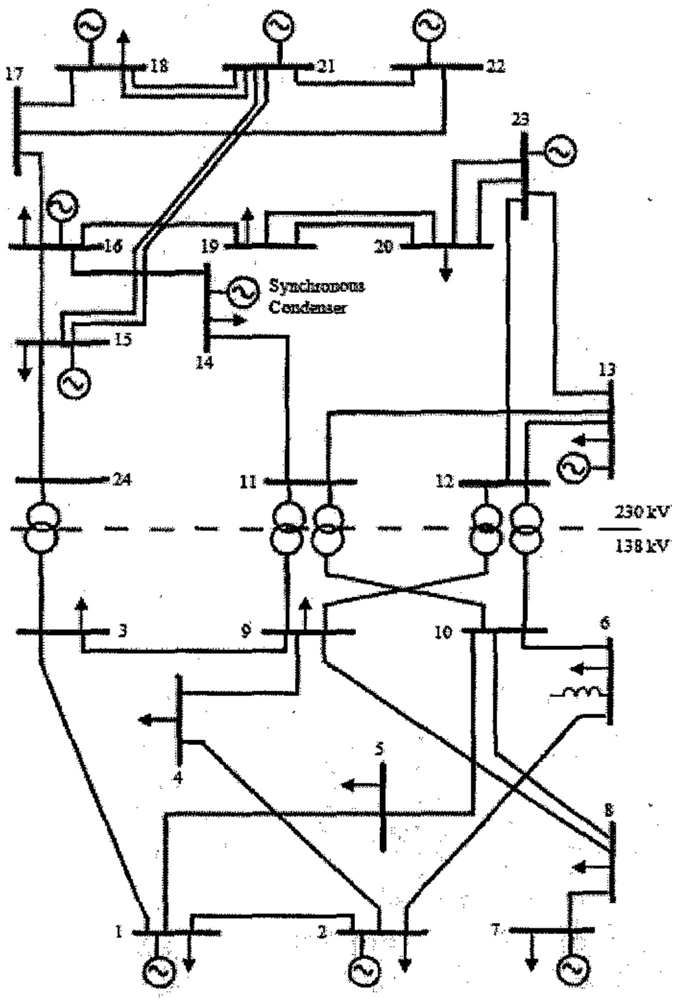


Figure 6 *IEEE 24-bus RTS*

3.5 IEEE 57-bus test system

The IEEE 57 Bus Test Case represents a portion of the American Electric Power System (in the Midwestern US) as it was in the early 1960's. The data was kindly provided by Iraj Dabbagchi of AEP and entered IEEE Common Data Format by Rich Christie at the University of Washington in August 1993.

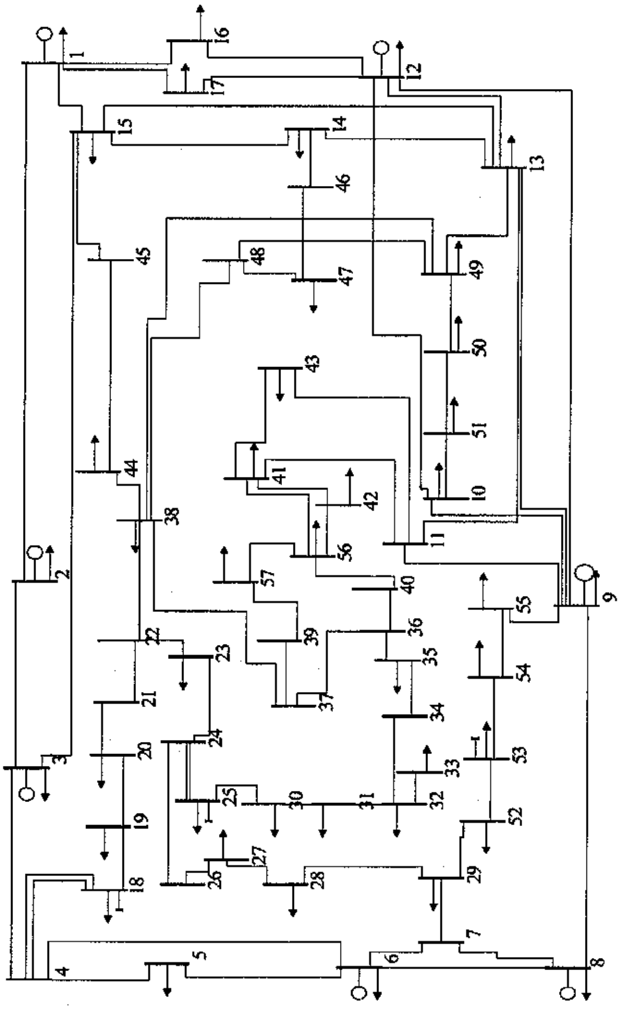


Figure 7 *IEEE 57 bus test system*

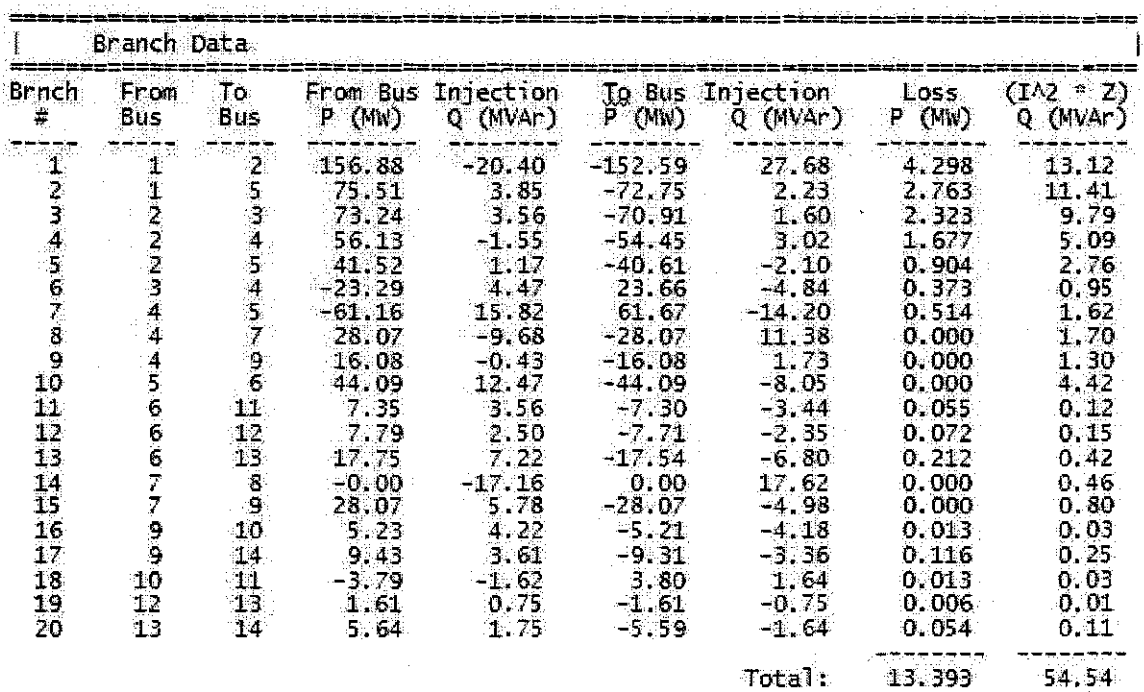
CHAPTER 4

RESULT AND DISCUSSION

4.1 Power Flow Simulation

The power flow simulation is first performed by using the power flow programs in Matpower to run the bus data and generation data of IEEE 14-bus test system in m-file. Then, the result of power flow, the load flow among the buses and losses will be used in the algorithms to perform allocation of transmission losses. The result of the power flow in IEEE 14-bus test system is as shown below in Table 1.

Table 1 *Power flow results of IEEE 14-bus test system*



```
================================================================================
|     Branch Data                                                              |
================================================================================
Brnch   From   To    From Bus Injection   To Bus Injection     Loss   (I^2 * Z)
  #     Bus    Bus    P (MW)   Q (MVAr)   P (MW)   Q (MVAr)   P (MW)   Q (MVAr)
-----  -----  -----  --------  --------  --------  --------  --------  --------
   1      1      2    156.88    -20.40   -152.59     27.68     4.298     13.12
   2      1      5     75.51      3.85    -72.75      2.23     2.763     11.41
   3      2      3     73.24      3.56    -70.91      1.60     2.323      9.79
   4      2      4     56.13     -1.55    -54.45      3.02     1.677      5.09
   5      2      5     41.52      1.17    -40.61     -2.10     0.904      2.76
   6      3      4    -23.29      4.47     23.66     -4.84     0.373      0.95
   7      4      5    -61.16     15.82     61.67    -14.20     0.514      1.62
   8      4      7     28.07     -9.68    -28.07     11.38     0.000      1.70
   9      4      9     16.08     -0.43    -16.08      1.73     0.000      1.30
  10      5      6     44.09     12.47    -44.09     -8.05     0.000      4.42
  11      6     11      7.35      3.56     -7.30     -3.44     0.055      0.12
  12      6     12      7.79      2.50     -7.71     -2.35     0.072      0.15
  13      6     13     17.75      7.22    -17.54     -6.80     0.212      0.42
  14      7      8     -0.00    -17.16      0.00     17.62     0.000      0.46
  15      7      9     28.07      5.78    -28.07     -4.98     0.000      0.80
  16      9     10      5.23      4.22     -5.21     -4.18     0.013      0.03
  17      9     14      9.43      3.61     -9.31     -3.36     0.116      0.25
  18     10     11     -3.79     -1.62      3.80      1.64     0.013      0.03
  19     12     13      1.61      0.75     -1.61     -0.75     0.006      0.01
  20     13     14      5.64      1.75     -5.59     -1.64     0.054      0.11
                                                             --------  --------
                                                    Total:    13.393     54.54
```

4.2 Transmission Loss Allocation Algorithms

After the power flow simulations done, the bus and branch data will then be taken to solve the loss allocation methods. The methods presented here are *pro rata* to load, *pro rata* 50:50, incremental allocation and proportional sharing allocation. The formulas of the loss allocation methods had been presented in the literature review part.

Previous analysis of different loss allocation methods on IEEE 14-bus test system done by Valerie Lim [5] is as shown in Table 2. The result had been taken as a reference to validate the the power flow program and loss algorithms. It is desired that the result will be identical with data in Table 2 to ensure that the accuracy of further analysis on bigger network, i.e. the IEEE 24-bus RTS and 57-bus test system will be safeguarded. The result using MATLAB had been shown in Table 3, as compared to the data in Table 2.

With closer look, it is well assured that the losses allocated to every buses using each of the loss allocations methods is almost identical. This shows that the power flow simulations in MATLAB and algorithms constructed had yielded a desired result. Thus, the same procedure of work implementation can now be done to bigger network.

Table 2

Previous analysis done with applications of different methods onto IEEE 14-Bus

Bus no.	**Real Power Loss (MW)**			
	Pro rata to loads	**Pro rata 50:50**	**IM**	**PS**
1	0	5.71	0	0
2	1.12	1.54	0.51	0.48
3	4.87	2.44	5.52	5.55
4	2.47	1.24	2.49	2.45
5	0.39	0.2	0.29	0.31
6	0.58	0.29	0.43	0.45
9	1.53	0.76	1.52	1.51
10	0.47	0.23	0.51	0.48
11	0.18	0.09	0.15	0.17
12	0.32	0.16	0.27	0.3
13	0.7	0.35	0.69	0.72
14	0.77	0.39	1.01	0.95
Total	**13.4**	**13.4**	**13.39**	**13.37**

Table 3

Results obtained with applications of the different methods using MATLAB

Bus no.	**Real Power Loss (MW)**			
	Pro rata to loads	**Pro rata 50:50**	**IM**	**PS**
1	0.0000	5.7119	0	0.0000
2	1.1219	1.5441	0.51	0.4834
3	4.8700	2.4350	5.52	5.5471
4	2.4712	1.2356	2.49	2.4560
5	0.3929	0.1965	0.29	0.3081
6	0.5790	0.2895	0.43	0.4540
9	1.5251	0.7626	1.52	1.5157
10	0.4653	0.2326	0.51	0.4798
11	0.1809	0.0905	0.15	0.1668
12	0.3154	0.1577	0.27	0.3131
13	0.6979	0.3490	0.69	0.7135
14	0.7703	0.3852	1.01	0.9526
Total	**13.3899**	**13.3902**	**13.39**	**13.3901**

4.3 A Novel Approach to Loss Prediction

It is impossible to solve most differential equations in the sense of obtaining an explicit formula for the solution. Despite the absence of an explicit solution, we can still learn a lot about the solution through a graphical approach (direction fields) or a numerical approach (Euler's method). Thus, consider the following equation:

$$y' = y \qquad y(0) = 1 \tag{15}$$

The differential equation tells us that $y'(0) = 1$, so the solution curve has slope 1 at point (0, 1). As a first approximation the tangent line at (0, 1) could be used as a rough approximation to the solution curve. Euler's idea was to improve on the approximation by proceeding only a short distance along this tangent line and then making a midcourse correction by changing direction. If the step size decreased from 0.5 to 0.05, better Euler approximation could be obtained, as shown in Figure 8.

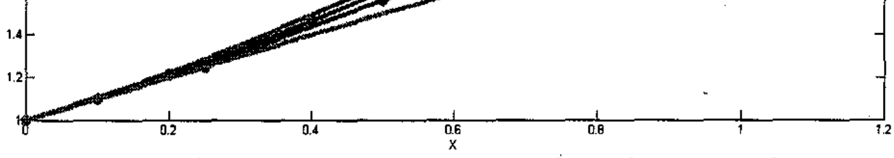


Figure 8 *Euler approximations approaching the exact solution*

This is a valuable insight which would prove useful for the improvement of the accuracy of the devised prediction method. Instead of selecting data samples randomly from past history (in previous section) prior to power flow simulation, only data samples from a certain range of power demand percentage will be chosen.

The differential equation tells us that $y'(0) = 1$, so the solution curve has slope 1 at point (0, 1). As a first approximation the tangent line at (0, 1) could be used as a rough approximation to the solution curve. Euler's idea was to improve on the approximation by proceeding only a short distance along this tangent line and then making a midcourse correction by changing direction. If the step size decreased from 0.5 to 0.05, better Euler approximation could be obtained, as shown in Figure 8.

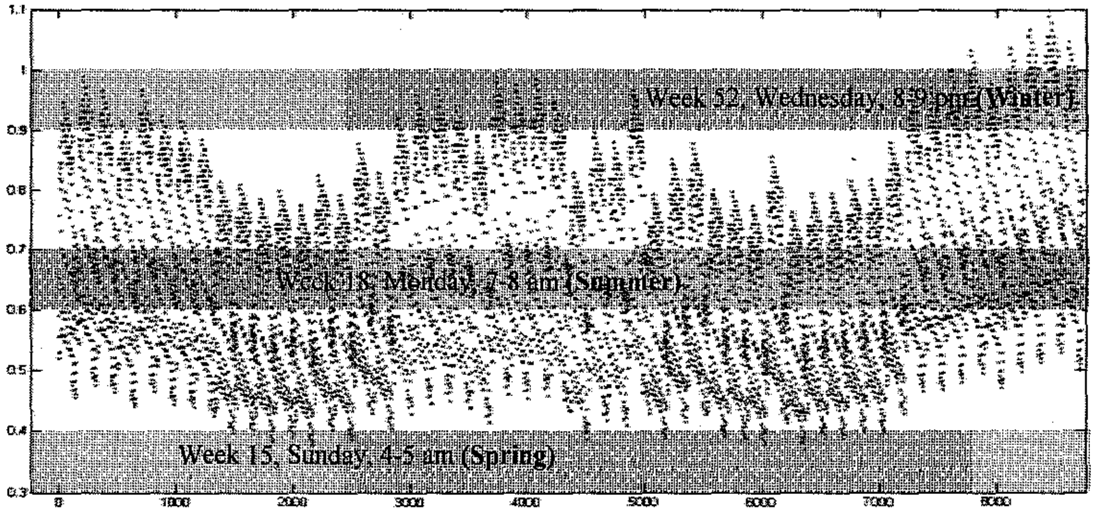


Figure 9 *Data samples vs. load demand (in proportion of full load)*

Figure 9 shows the distribution of system load demand throughout the year in 24-bus RTS. The y-coordinates of the graph are obtained in such a way that all load demand is taken as the proportion of full load, which is referred to as 1 in the graph whereas the peak load demand in a year is 10% higher than the full load, thus having value of 1.1 in the graph. Learning coefficients come into play when the y-axis is now divided into different sections prior to power flow simulation.

The range of each section in this case will be taken as 0.1 and as a result, the range from 0.3 up to 1.1 will give us 8 different sections for analysis, specifically for the generation of learning coefficients. As referring to equation 12, 10 data samples from each section will be needed for the regression method. That means that 10 samples with each equivalent to a 0.1 step size will be taken for the improvement of the credibility of Learning Coefficients.

Table 4

Sets of Learning Coefficients for Different Load Demand in IEEE 24-bus RTS

Bus	Pd: 30-40 %			Pd: 60-70 %			Pd: 90-100 %		
#	α	β	γ	α	β	γ	α	β	γ
1	0.0028	-0.0036	0.222	0.1195	-0.3977	0.562	1.1735	-2.7082	1.8398
2	-0.0007	0.0077	0.0146	0.0012	0.0014	0.02	0.0264	-0.0601	0.0578
3	-0.0023	0.0064	3.1821	0.0868	-0.1618	3.2619	1.6665	-2.2269	3.9425
4	-0.0486	0.6075	1.05	-0.1805	1.3412	0.0049	-9.9001	26.6752	-14.9711
5	-0.0373	0.4802	1.3382	-0.2068	1.4082	0.0328	-2.9318	6.5763	-1.4751
6	-0.1909	1.2512	2.1787	0.262	-1.2488	4.8162	-52.6152	84.6778	-29.9454
7	-	-	-	-	-	-	-	-	-
8	-0.2535	1.3214	1.6075	-1.4141	3.9716	0.053	-27.6622	30.4211	-5.3707
9	-0.5834	3.1702	2.8254	-17.1293	29.78	-7.5291	-121.497	160.0412	-48.3405
10	-0.7433	3.5729	1.8765	-3.2743	7.3928	0.9246	-167.9041	196.2115	-53.0172
11	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-
13	39.8568	227.6785	-117.9482	233.2604	-42.9982	-20.4036	1173.7782	-859.8671	158.6823
14	-0.4246	1.8796	3.1259	-2.521	6.0761	0.9673	-6.528	11.2314	-0.7087
15	-0.071	0.1893	1.1359	-0.4521	0.6546	0.9899	-1.2581	1.2856	0.8651
16	-0.043	0.3719	1.2976	-0.2415	1.1464	0.5213	-0.5214	1.8616	0.06
17	-	-	-	-	-	-	-	-	-
18	0.0486	-0.1215	0.1269	0.3529	-0.4722	0.2306	1.1624	-1.0699	0.3421
19	-0.3306	1.5801	1.59	-1.9933	5.1364	-0.3645	37.9551	-32.8383	7.9211
20	-0.2699	1.8349	1.4838	-0.9609	4.1791	-0.5311	3.824	-6.9321	3.3812
21	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-	-
24	-	-	-	-	-	-	-	-	-

Critical analysis has been done on lower, medium and higher value of power demand, i.e. 30-40%, 60-70% and 90-100% of full load. The corresponding set of learning coefficients is as shown in Table 4 as the system characteristics is now represented by a few sets of learning coefficients, instead of one in previous studies.

Performance of the new method has proved to be highly accurate, as shown in Table 5 when a date sample from each section were taken for loss prediction. The predicted losses and the actual losses will then be compared by means of the MAPE. The 3 data samples were each selected from sets of 8736 data samples in Figure 9.

Table 5 *Comparison between Actual and Predicted Losses on IEEE 24-bus RTS*

Bus	Week 15, Sunday, 4-5 am			Week 18, Monday, 7-8 am			Week 52, Wednesday, 8-9 pm		
#	Actual losses	Predicted losses	MAPE [%]	Actual losses	Predicted losses	MAPE [%]	Actual losses	Predicted losses	MAPE [%]
1	0.0959	0.0959	0.0072	0.1674	0.1674	0.0035	0.311	0.311	0.0202
2	0.0114	0.0114	0.0013	0.0159	0.0159	0.0009	0.0214	0.0214	0.0081
3	2.2177	2.2177	0	3.7331	3.7331	0.0001	5.3919	5.3919	0.0011
4	0.7383	0.7383	0.0061	0.9687	0.9687	0.0001	2.0082	2.0034	0.2406
5	0.7118	0.7117	0.0055	0.9759	0.9759	0.0006	1.1772	1.1767	0.0447
6	2.034	2.0339	0.0056	3.3103	3.3098	0.015	5.236	5.218	0.3425
7	-	-	-	-	-	-	-	-	-
8	2.0008	2.0007	0.0057	2.7597	2.7597	0.0006	4.5293	4.5224	0.151
9	4.2198	4.2196	0.0035	6.1505	6.1645	0.2271	6.729	6.6964	0.4839
10	4.002	4.0018	0.0047	5.9878	5.9857	0.0353	7.5065	7.4625	0.5857
11	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-
13	145.72	145.7309	0.0075	57.0926	57.0794	0.0231	7.0313	7.1315	1.4242
14	3.6584	3.6582	0.0048	5.3003	5.3003	0.0006	6.3442	6.3443	0.0009
15	1.5235	1.5235	0.0012	2.4775	2.4775	0.0001	3.4216	3.4216	0.0002
16	0.7625	0.7625	0.0045	1.1145	1.1145	0.0005	1.3593	1.3593	0.0002
17	-	-	-	-	-	-	-	-	-
18	0.0795	0.0796	0.0163	0.1904	0.1904	0.0015	0.3679	0.3679	0.0025
19	2.2205	2.2204	0.0066	3.0146	3.0146	0.0009	3.006	3.0052	0.0265
20	2.024	2.0238	0.0081	2.5831	2.5831	0.0017	0.3086	0.3088	0.0448
21	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-	-
24	-	-	-	-	-	-	-	-	-
Total	172.0201	172.0298	0.0056	95.8423	95.8405	0.0019	54.7495	54.7423	0.0132

As the loss prediction done on IEEE 24-bus RTS has yielded a MAPE percentage less than 0.1%, similar condition and analysis has been applied to a larger and more practical system, i.e. IEEE 57-bus system to determine the effectiveness and validity of the method. Again, the results obtained from the system has shown that the newly devised method capable of predicting the oncoming losses in every buses accurately, regardless of the size of the system. This can be shown from Table 6 that the MAPE for each condition is far lower than 1%.

Table 6 *Comparison between Actual and Predicted Losses on IEEE 57-bus Test Case*

Bus	Week 15, Sunday, 4-5 am			Week 18, Monday, 7-8 am			Week 52, Wednesday, 8-9 pm		
#	Actual losses	Predicted losses	MAPE [%]	Actual losses	Predicted losses	MAPE [%]	Actual losses	Predicted losses	MAPE [%]
1	25.1687	25.1805	0.0471	1.7745	1.6351	7.8573	0	0	0
2	0.1023	0.1023	0.0199	0.0977	0.2252	130.3771	0.0386	0.0386	0.0394
3	0.7569	0.7567	0.0253	0.3679	0.3283	10.7638	0.9217	0.9216	0.0135
4	-	-	-	-	-	-	-	-	-
5	0.2036	0.2036	0.01	0.2455	0.2455	0.0013	0.3656	0.3684	0.7516
6	0.7344	0.7344	0.0061	1.0172	1.0172	0.0008	1.2333	1.2289	0.3568
7	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-
9	1.0491	1.049	0.003	1.6178	1.6178	0.0004	2.0927	2.0928	0.0008
10	0.0722	0.0721	0.0047	0.1074	0.1074	0.0003	0.1045	0.1039	0.5181
11	-	-	-	-	-	-	-	-	-
12	0.6012	0.6011	0.0174	0.4168	0.4168	0.0006	1.8879	1.8119	4.0243
13	0.2931	0.2931	0.0064	0.3943	0.3943	0.0096	0.7422	0.7633	2.8398
14	0.2057	0.2057	0.0081	0.2755	0.2791	1.2832	0.3445	0.3455	0.302
15	0.5394	0.5396	0.0517	0.3909	0.4256	8.8947	0.4963	0.4963	0.0029
16	0.3081	0.3081	0.0203	0.2157	0.2426	12.4395	1.0966	1.0964	0.0137
17	0.5476	0.5475	0.0233	0.6427	0.5848	9.0153	0.6743	0.6742	0.0111
18	0.6374	0.6373	0.0134	0.6605	0.6605	0.0022	0.7464	0.7463	0.0136
19	0.1098	0.1098	0.0033	0.1143	0.1143	0.0009	0.1601	0.16	0.0108
20	0.0896	0.0896	0.0063	0.1156	0.1135	1.7724	0.1276	0.1279	0.1974
21	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-
23	0.2389	0.2389	0.0068	0.3236	0.3246	0.2986	0.3666	0.3681	0.4095
24	-	-	-	-	-	-	-	-	-
25	0.1759	0.1759	0.0043	0.2597	0.2597	0.0001	0.3428	0.3431	0.0729
26	-	-	-	-	-	-	-	-	-
27	0.1362	0.1362	0.0013	0.2194	0.2194	0.0003	0.3096	0.3096	0.0029
28	0.0419	0.0419	0.0009	0.068	0.068	0.0002	0.0963	0.0963	0.0019
29	0.089	0.089	0.0024	0.1388	0.1388	0.0002	0.1844	0.1844	0.0001
30	0.1167	0.1167	0.0028	0.1782	0.1782	0.0003	0.2461	0.2463	0.0579
31	0.2244	0.2244	0.0015	0.359	0.3596	0.1737	0.5418	0.5425	0.1195
32	0.0828	0.0833	0.5786	0.0965	0.0974	0.9879	0.1496	0.15	0.2619
33	0.1993	0.2005	0.572	0.2335	0.2358	0.971	0.363	0.364	0.2568
34	-	-	-	-	-	-	-	-	-
35	0.3105	0.3126	0.6853	0.3459	0.3495	1.0295	0.5313	0.5327	0.2756
36	-	-	-	-	-	-	-	-	-
37	-	-	-	-	-	-	-	-	-
38	0.4452	0.4494	0.935	0.5149	0.523	1.5668	0.7742	0.7775	0.4293
39	-	-	-	-	-	-	-	-	-
40	-	-	-	-	-	-	-	-	-
41	0.0958	0.0958	0.0072	0.1299	0.1299	0.001	0.1629	0.1661	1.9108
42	0.1428	0.1428	0.0027	0.2217	0.2217	0.0002	0.322	0.3256	1.1107
43	0.0304	0.0304	0.0072	0.0412	0.0412	0.001	0.0517	0.0527	1.9108
44	0.4003	0.4039	0.8927	0.2684	0.2875	7.1156	0.4906	0.4906	0.0058
45	-	-	-	-	-	-	-	-	-
46	-	-	-	-	-	-	-	-	-
47	0.8373	0.8209	1.958	1.0045	1.0146	1.0065	1.3259	1.3289	0.2235
48	-	-	-	-	-	-	-	-	-
49	0.3614	0.3613	0.0098	0.3943	0.398	0.9363	0.7422	0.7608	2.5025
50	0.4652	0.4651	0.0036	0.7453	0.7381	0.9725	0.9046	0.9133	0.9622
51	0.2598	0.2597	0.0047	0.3868	0.3868	0.0003	0.3761	0.3742	0.5181
52	0.0489	0.0489	0.0074	0.0986	0.0986	0.0009	0.1684	0.1684	0.002
53	0.3015	0.3015	0.0109	0.5793	0.5793	0.0011	0.9888	0.9888	0.0024
54	0.073	0.073	0.007	0.1097	0.1097	0.0006	0.167	0.167	0.0016
55	0.059	0.059	0.003	0.0909	0.0909	0.0004	0.1176	0.1176	0.0008
56	0.1815	0.1815	0.0023	0.3139	0.3148	0.2654	0.4859	0.4891	0.663
57	0.1873	0.187	0.1569	0.3159	0.3179	0.6525	0.4886	0.4908	0.4419
Total	36.9239	36.93	0.0165	15.8923	15.8908	0.0093	21.7307	21.7244	0.029

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

Previous work has showed that converging trend of the learning coefficients underpins the learning coefficient method in performing prediction, where given sufficient samples and spread of samples, prediction done for an oncoming demand could be done to a highly accurate degree, with a 5% MAPE. Still, it could be improved further to a degree of 0.1% if different set of learning coefficients were generated under different ranges of load demand. This could be done through a partition of possible range of load demand into different sections. In this paper, 3 sections have been selected: 30-40%, 60-70% and 90-100% of full load which signifies the lower, medium and higher load demand.

Each section will then undergo load flow solution with a step size of 1% before a set of learning coefficients was generated using regression method. Euler's method has indeed provides an insight that better approximation of relationships between power loss and retailer's demand, in the form of heat-rate curve could be achieved with smaller step size. This corroborates that a power system cannot be characterised solely by a set of learning coefficients. Instead, each regions of load demand should have different set of learning coefficients for best prediction of system losses.

In conclusion, the work had shown major breakthrough as prediction error is now reduced to less than 0.1%. No previous work, reference or research had been published in the area of loss prediction. Last but not least, this paper had been accepted for oral presentation at the 5th International Power Engineering and Optimization Conference (PEOCO2011), Shah Alam, coming this June 2011.

5.2 Recommendations

With the development of the new loss prediction method, further research can be directed on the prediction of daily load demand (using learning coefficients) which will be beneficial for any deregulated market participants as the two: system losses and load demand prediction will combine to yield a satisfactory prediction tool for oncoming electricity prices.

Besides, loss prediction using other loss allocation methods, e.g. *pro rata* and incremental method will also be greatly beneficial to other power markets such as Spain, Brazil, Australia and New Zealand. Analysis can be done on these methods by evaluating the performance of loss prediction and comparing the percentage of error with that of proportional sharing.

PUBLICATION

[1] Perumal. N., Tee Ping Hong, (2011). Development of a New Loss Prediction Method in a Deregulated Power Market using Proportional Sharing. PEOCO 2011. IEEE.

REFERENCES

[1] J. Bialek, "Tracing the flow of electricity," IEE Proceedings Generation, Transmission and Distribution, vol. 143, pp. 313-20, 1996.

[2] A. J. Conejo, F. D. Galiana, and I. Kockar, "Z-Bus Loss Allocation," IEEE Transactions on Power Systems, vol. 16, pp. 105-10, 2001.

[3] A. Bhuiya and N. Chowdhury, "Allocation of Transmission Losses in a Deregulated Power System Network," presented at IEEE Canadian Conference on Electrical and Computer Engineering, Shaw Conference Center, Edmonton, Alberta, Canada, 1999.

[4] A. J. Conejo, J. M. Arroyo, N. Alguacil, and A. L. Guijarro, "Transmission loss allocation: a comparison of different practical algorithms," Power Systems, IEEE Transactions on, vol. 17, pp. 571-576, 2002.

[5] V. S. C. Lim, J. D. F. McDonald, and T. K. Saha, "Comparative Distribution of System Losses to Market Participants using Different Loss Allocation Methods," Proceedings of IEEE/PES T&D 2005, Asia Pacific, Dalian, China, 14 -17 August 2005.

[6] Bialek, J., Kattuman, P. (2004). Proportional sharing assumption in tracing methodology. Generation, Transmission and Distribution, IEE Proceedings-.

[7] V. Lim, T. K. Saha, T. Downs, “Preliminary Findings on Usage Allocation and Loss Allocation of Electricity in a Deregulated Market”, presented at Australasian Universities Power Engineering Conference, Christchurch, New Zealand 2003.

[8] National Electricity Market Management Company Ltd, NEMMCO homepage, http://www.nemmco.com.au, 2010.

[9] V. Lim, T. K. Saha, T. Downs, "Loss Allocation Based on Network Reduction in Deregulated Electricity Market", Proceeding of Australasian Universities Power Engineering Conference (AUPEC 2004), Brisbane, Australia, 26-29 September 2004.

[10] NEMMCO, "List of Regional Boundaries and Marginal Loss Factors for the 2004/05 Financial Year", 20 May 2004.

[11] A. J. Conejo, J. M. Arroyo, N. Alguacil, and A. L. Guijarro, "Transmission loss allocation: a comparison of different practical algorithms," Power Systems, IEEE Transactions on, vol. 17, pp. 571-576, 2002.

[12] J. Bialek, "Tracing the flow of electricity," IEE Proceedings Generation, Transmission and Distribution, vol. 143, pp. 313-20, 1996.

[13] Bergen, A. R., & Vittal, V. (1999). *Power Systems Analysis* (2nd Edition ed.). Pearson.

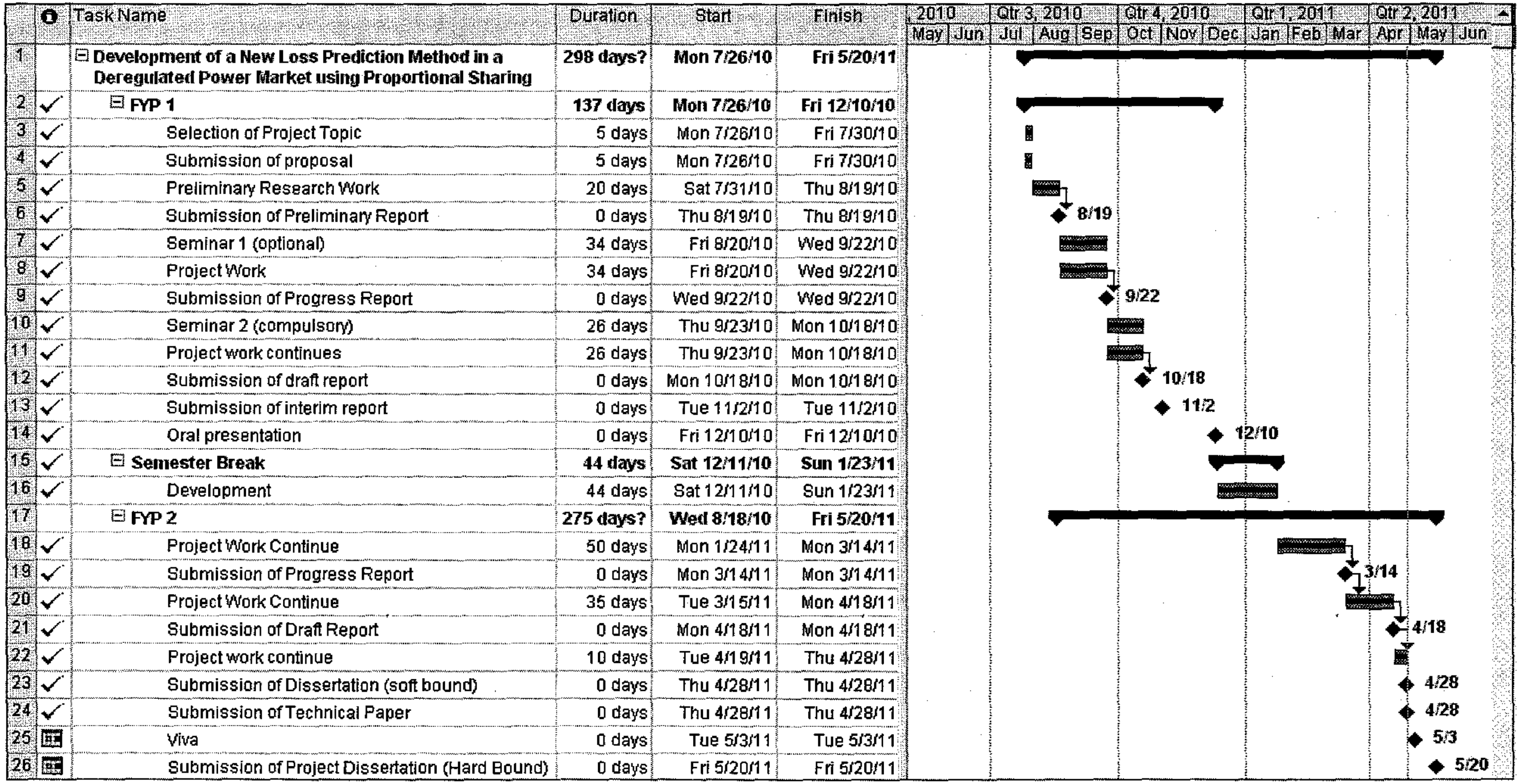
[14] Perumal.N., Mukerjee.R.N., & Syafruddin, M. (2010). System Reliability by Prediction of Generator Output and Losses in a Competitive Energy Market. World Academy of Science, Engineering and Technology 2010, Volume 62, Feb 2010., (pp. 665-669). Penang.

[15] Perumal. N., Lee Tze Yang, (2010). Economic Analysis of a Deregulated Power System using the Proportional Tracing Mehtod. PEOCO 2010. IEEE.

[16] Grigg, Wong, C., Albrecht, P., Allan, P., Bhavaraju, R., Billinton, M., et al. (1996). The IEEE Reliability Test System-1996. A report prepared by the Reliability Test System Task Force of the Application of Probability Methods Subcommitte. *IEEE Transactions on Power Systems, Volume 14* (Issue 12), 1010 – 1020.

APPENDIX A

PROJECT GANTT CHART



	ⓘ	Task Name	Duration	Start	Finish
1		**⊟ Development of a New Loss Prediction Method in a Deregulated Power Market using Proportional Sharing**	**298 days?**	**Mon 7/26/10**	**Fri 5/20/11**
2	✓	**⊟ FYP 1**	**137 days**	**Mon 7/26/10**	**Fri 12/10/10**
3	✓	Selection of Project Topic	5 days	Mon 7/26/10	Fri 7/30/10
4	✓	Submission of proposal	5 days	Mon 7/26/10	Fri 7/30/10
5	✓	Preliminary Research Work	20 days	Sat 7/31/10	Thu 8/19/10
6	✓	Submission of Preliminary Report	0 days	Thu 8/19/10	Thu 8/19/10
7	✓	Seminar 1 (optional)	34 days	Fri 8/20/10	Wed 9/22/10
8	✓	Project Work	34 days	Fri 8/20/10	Wed 9/22/10
9	✓	Submission of Progress Report	0 days	Wed 9/22/10	Wed 9/22/10
10	✓	Seminar 2 (compulsory)	26 days	Thu 9/23/10	Mon 10/18/10
11	✓	Project work continues	26 days	Thu 9/23/10	Mon 10/18/10
12	✓	Submission of draft report	0 days	Mon 10/18/10	Mon 10/18/10
13	✓	Submission of interim report	0 days	Tue 11/2/10	Tue 11/2/10
14	✓	Oral presentation	0 days	Fri 12/10/10	Fri 12/10/10
15	✓	**⊟ Semester Break**	**44 days**	**Sat 12/11/10**	**Sun 1/23/11**
16	✓	Development	44 days	Sat 12/11/10	Sun 1/23/11
17		**⊟ FYP 2**	**275 days?**	**Wed 8/18/10**	**Fri 5/20/11**
18	✓	Project Work Continue	50 days	Mon 1/24/11	Mon 3/14/11
19	✓	Submission of Progress Report	0 days	Mon 3/14/11	Mon 3/14/11
20	✓	Project Work Continue	35 days	Tue 3/15/11	Mon 4/18/11
21	✓	Submission of Draft Report	0 days	Mon 4/18/11	Mon 4/18/11
22	✓	Project work continue	10 days	Tue 4/19/11	Thu 4/28/11
23	✓	Submission of Dissertation (soft bound)	0 days	Thu 4/28/11	Thu 4/28/11
24	✓	Submission of Technical Paper	0 days	Thu 4/28/11	Thu 4/28/11
25	▦	Viva	0 days	Tue 5/3/11	Tue 5/3/11
26	▦	Submission of Project Dissertation (Hard Bound)	0 days	Fri 5/20/11	Fri 5/20/11

APPENDIX B

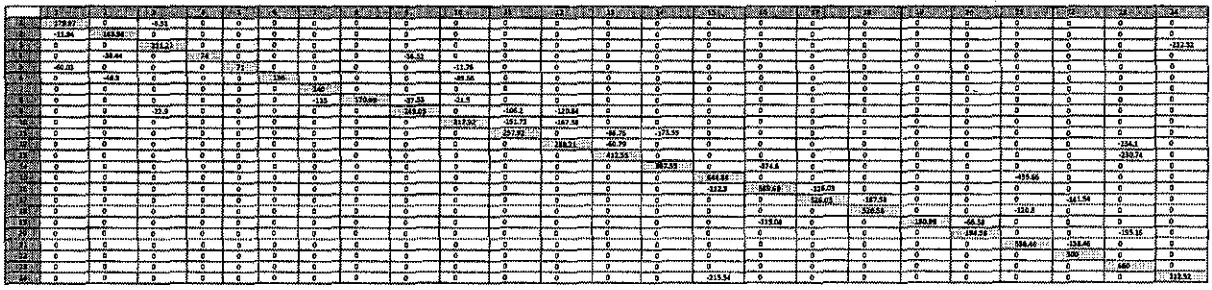
CODING DOCUMENTATION

1. Construction of Upstream Matrix for full load condition

```
================================================================================
|     System Summary                                                           |
================================================================================

How many?                How much?              P (MW)            Q (MVAr)
---------------------    -------------------  -------------  -----------------
Buses             24     Total Gen Capacity    3405.0        -535.0 to 1776.0
Generators        33     On-line Capacity      3405.0        -535.0 to 1776.0
Committed Gens    33     Generation (actual)   2901.2             587.4
Loads             17     Load                  2850.0             580.0
  Fixed           17       Fixed               2850.0             580.0
  Dispatchable     0       Dispatchable          -0.0 of -0.0      -0.0
Shunts             1     Shunt (inj)             -0.0            -102.5
Branches          38     Losses (I^2 * Z)        51.25            454.77
Transformers       5     Branch Charging (inj)     -              549.9
Inter-ties        10     Total Inter-tie Flow  1339.8             204.9
Areas              4

                          Minimum                      Maximum
                 -------------------------  --------------------------------
Voltage Magnitude   0.978 p.u. @ bus 24         1.050 p.u. @ bus 18
Voltage Angle     -12.42 deg   @ bus 6         22.77 deg   @ bus 22
P Losses (I^2*R)             -                  7.05 MW    @ line 14-16
Q Losses (I^2*X)             -                 54.88 MVAr  @ line 14-16
================================================================================
|     Bus Data                                                                 |
================================================================================
 Bus      Voltage          Generation             Load
  #   Mag(pu) Ang(deg)   P (MW)   Q (MVAr)   P (MW)   Q (MVAr)
----- ------- --------  --------  --------  --------  --------
    1  1.035   -7.278    172.00     21.47    108.00     22.00
    2  1.035   -7.370    172.00     15.66     97.00     20.00
    3  0.989   -5.584       -         -      180.00     37.00
    4  0.998   -9.690       -         -       74.00     15.00
    5  1.019   -9.964       -         -       71.00     14.00
    6  1.012  -12.421       -         -      136.00     28.00
    7  1.025   -7.357    240.00     51.84    125.00     25.00
    8  0.993  -11.088       -         -      171.00     35.00
    9  1.001   -7.435       -         -      175.00     36.00
   10  1.028   -9.503       -         -      195.00     40.00
   11  0.990   -2.154       -         -         -         -
   12  1.003   -1.517       -         -         -         -
   13  1.020    0.000    187.25    133.99    265.00     54.00
   14  0.980    2.258      0.00    -27.72    194.00     39.00
   15  1.014   11.566    215.00     -3.95    317.00     64.00
   16  1.017   10.449    155.00     44.40    100.00     20.00
   17  1.039   14.931       -         -         -         -
   18  1.050   16.292    400.00    138.73    333.00     68.00
   19  1.023    8.917       -         -      181.00     37.00
   20  1.038    9.530       -         -      128.00     26.00
   21  1.050   17.117    400.00    106.91       -         -
   22  1.050   22.766    300.00    -29.55       -         -
   23  1.050   10.572    660.00    135.59       -         -
   24  0.978    5.299       -         -         -         -
                        --------  --------  --------  --------
               Total:   2901.25    587.36   2850.00    580.00
================================================================================
|     Branch Data                                                              |
================================================================================
Brnch   From   To    From Bus Injection   To Bus Injection     Loss (I^2 * Z)
  #     Bus    Bus    P (MW)   Q (MVAr)   P (MW)   Q (MVAr)   P (MW)   Q (MVAr)
-----  -----  -----  --------  --------  --------  --------  --------  --------
   1      1      2     11.94    -26.92    -11.94    -22.45     0.004      0.02
   2      1      3     -7.97     21.57      8.31    -26.11     0.342      1.32
   3      1      5     60.03      4.83    -59.29     -4.37     0.741      2.87
   4      2      4     38.44     19.15    -37.85    -20.43     0.587      2.27
   5      2      6     48.50     -1.04    -47.41     -0.19     1.093      4.22
   6      3      9     22.90    -17.01    -22.66     14.75     0.240      0.93
   7      3     24   -211.21      6.12    212.32     34.48     1.113     40.60
   8      4      9    -36.15      5.43     36.52     -6.83     0.364      1.41
   9      5     10    -11.71     -9.63     11.76      7.30     0.046      0.18
  10      6     10    -88.59   -130.31     89.66   -121.12     1.067      4.64
  11      7      8    115.00     26.84   -112.88    -20.35     2.118      8.18
  12      8      9    -36.92      3.36     37.53     -5.46     0.604      2.34
  13      8     10    -21.19    -18.01     21.50     14.61     0.303      1.17
  14      9     11   -105.92    -12.77    106.20     22.87     0.277     10.10
  15      9     12   -120.47    -25.69    120.84     39.16     0.369     13.47
  16     10     11   -151.18     36.03    151.72    -16.10     0.546     19.93
  17     10     12   -166.74     23.18    167.38      0.21     0.641     23.39
  18     11     13    -86.15    -54.97     86.76     49.70     0.618      4.82
  19     11     14   -171.77     48.19    173.55    -42.96     1.778     13.76
  20     12     13    -60.51    -33.30     60.79     25.20     0.271      2.11
  21     12     23   -227.70     -6.07    234.10     34.52     6.399     49.85
  22     13     23   -225.30      5.10    230.74     17.80     5.438     42.38
  23     14     16   -367.55    -23.77    374.60     70.49     7.054     54.88
  24     15     16    112.30    -32.60   -112.01     31.13     0.290      2.28
  25     15     21   -214.92    -41.97    217.83     53.65     2.913     22.65
  26     15     21   -214.92    -41.97    217.83     53.65     2.913     22.65
  27     15     24    215.54     48.59   -212.32    -34.48     3.219     24.93
  28     16     17   -322.68    -33.86    326.03     54.42     3.353     26.31
  29     16     19    115.08    -43.35   -114.65     41.64     0.433      3.33
  30     17     18   -186.94    -58.69    187.58     60.49     0.638      5.10
  31     17     22   -139.09      4.28    141.54     -9.26     2.454     19.14
  32     18     21    -60.29      5.12     60.40    -10.26     0.111      0.87
  33     18     21    -60.29      5.12     60.40    -10.26     0.111      0.87
  34     19     20    -33.17    -39.32     33.29     31.34     0.113      0.88
  35     19     20    -33.17    -39.32     33.29     31.34     0.113      0.88
  36     20     23    -97.29    -44.34     97.58     41.63     0.291      2.25
  37     20     23    -97.29    -44.34     97.58     41.63     0.291      2.25
  38     21     22   -156.46     20.12    158.46    -20.29     1.994     15.54
                                                             --------  --------
                                                    Total:    51.246    454.77
```



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	179.97	0	-8.31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	-11.94	183.58	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	311.21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-212.52
4	0	-38.44	0	74	0	0	0	0	-34.52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	-60.03	0	0	0	71	0	0	0	0	-11.76	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	-48.3	0	0	0	136	0	0	0	-89.58	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	140	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	-135	170.09	-37.53	-21.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	-22.9	0	0	0	0	0	249.03	0	-106.2	-120.84	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	317.92	-151.73	-167.58	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	257.92	0	-86.76	-171.55	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	288.21	-60.79	0	0	0	0	0	0	0	0	0	-234.1	0
13	0	0	0	0	0	0	0	0	0	0	0	0	412.55	0	0	0	0	0	0	0	0	0	-230.74	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	387.55	0	-374.6	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	644.88	0	0	0	0	0	-435.66	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-112.3	569.60	-326.03	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	326.03	-187.58	0	0	0	-141.54	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	320.58	0	0	-120.8	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-119.04	0	0	190.09	-66.58	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	194.98	0	0	-195.16	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	534.46	-138.46	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	300	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	660	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-215.34	0	0	0	0	0	0	0	0	312.52

2. MATLAB Implementation of Upstream Matrix Construction

```
        A=[];
        B=[];
        for i = 1:nb
        fprintf(fd, '\n%5d%7.3f%9.3f', bus(i, [BUS_I, VM, VA]));
        g  = find(gen(:, GEN_STATUS) > 0 & gen(:, GEN_BUS) == bus(i, BUS_I) & ...
                    ~isload(gen));
        ld = find(gen(:, GEN_STATUS) > 0 & gen(:, GEN_BUS) == bus(i, BUS_I) & ...
                    isload(gen));
        if ~isempty(g)
            fprintf(fd, '%10.2f%10.2f', sum(gen(g, PG)), sum(gen(g, QG)));
            A(i,1) = sum(gen(g, PG
        else
            fprintf(fd, '       -         -  ');
            A(i,1) = 0;
        end
        if bus(i, PD) || bus(i, QD) || ~isempty(ld)
            if ~isempty(ld)
                fprintf(fd, '%10.2f*%9.2f*', bus(i, PD) - sum(gen(ld, PG)), ...
                                             bus(i, QD) - sum(gen(ld, QG)));
                B(i,1)=bus(i, PD) - sum(gen(ld, PG));
            else
                fprintf(fd, '%10.2f%10.2f ', bus(i, [PD, QD]));
                B(i,1)=bus(i, PD) - sum(gen(ld, PG));
            end
        else
            fprintf(fd, '       -         -   ');
            B(i,1)=0;
        end
        if isOPF
            fprintf(fd, '%9.3f', bus(i, LAM_P));
            if abs(bus(i, LAM_Q)) > ptol
                fprintf(fd, '%8.3f', bus(i, LAM_Q));
            else
                fprintf(fd, '      -');
            end
        end
    end
    fprintf(fd, '\n                          --------  --------  --------  --------');
    fprintf(fd, '\n                 Total: %9.2f %9.2f %9.2f %9.2f', ...
        sum(gen(ong, PG)), sum(gen(ong, QG)), ...
        sum(bus(nzld, PD)) - sum(gen(onld, PG)), ...
        sum(bus(nzld, QD)) - sum(gen(onld, QG)));
    fprintf(fd, '\n');
    teeg=A;
    teeload=B;
end
    C = [(1:nl)', branch(:,[F_BUS, T_BUS]),branch(:, [PF, QF]), branch(:, [PT, QT]),real(loss), imag(loss)];
    teestart = diag(teeg);
    for i=1:nl
    if C(i,4) < 0
        teestart(C(i,2),C(i,2))=teestart(C(i,2),C(i,2))-C(i,4);
        teestart(C(i,2),C(i,3))=teestart(C(i,2),C(i,3))-C(i,6);
    else
        teestart(C(i,3),C(i,3))=teestart(C(i,3),C(i,3))-C(i,6);
        teestart(C(i,3),C(i,2))=teestart(C(i,3),C(i,2))-C(i,4);
    end
    end
    invteestart=inv(teestart);
    Pfactor=invteestart*teeg;
    Loss=teeload.*(Pfactor-1);
    Tgen=sum(teeg);
    Tload=sum(teeload);
    Tloss=sum(Loss);
```

2.1 Upstream Matrix Implementation in MATLAB

```
teestart =

  Columns 1 through 12

    179.9667         0   -8.3082         0         0         0         0         0         0         0         0         0
    -11.9399  183.9363         0         0         0         0         0         0         0         0         0         0
           0         0  211.2063         0         0         0         0         0         0         0         0         0
           0  -38.4358         0   74.0000         0         0         0         0  -36.5156         0         0         0
    -60.0268         0         0         0   71.0000         0         0         0         0  -11.7595         0         0
           0  -48.5005         0         0         0  136.0000         0         0         0  -89.6592         0         0
           0         0         0         0         0         0  240.0000         0         0         0         0         0
           0         0         0         0         0         0 -115.0000  171.0000  -37.5274  -21.4973         0         0
           0         0  -22.8980         0         0         0         0         0  249.0431         0 -106.1956 -120.8356
           0         0         0         0         0         0         0         0         0  317.9160 -151.7233 -167.3802
           0         0         0         0         0         0         0         0         0         0  257.9190         0
           0         0         0         0         0         0         0         0         0         0         0  288.2158
           0         0         0         0         0         0         0         0         0         0         0         0
           0         0         0         0         0         0         0         0         0         0         0         0
           0         0         0         0         0         0         0         0         0         0         0         0
           0         0         0         0         0         0         0         0         0         0         0         0
           0         0         0         0         0         0         0         0         0         0         0         0
           0         0         0         0         0         0         0         0         0         0         0         0
           0         0         0         0         0         0         0         0         0         0         0         0
           0         0         0         0         0         0         0         0         0         0         0         0
           0         0         0         0         0         0         0         0         0         0         0         0
           0         0         0         0         0         0         0         0         0         0         0         0
           0         0         0         0         0         0         0         0         0         0         0         0
           0         0         0         0         0         0         0         0         0         0         0         0

  Columns 13 through 24

           0         0         0         0         0         0         0         0         0         0         0         0
           0         0         0         0         0         0         0         0         0         0         0         0
           0         0         0         0         0         0         0         0         0         0         0 -212.3191
           0         0         0         0         0         0         0         0         0         0         0         0
           0         0         0         0         0         0         0         0         0         0         0         0
           0         0         0         0         0         0         0         0         0         0         0         0
           0         0         0         0         0         0         0         0         0         0         0         0
           0         0         0         0         0         0         0         0         0         0         0         0
           0         0         0         0         0         0         0         0         0         0         0         0
           0         0         0         0         0         0         0         0         0         0         0         0
    -86.7639 -173.5510         0         0         0         0         0         0         0         0         0         0
    -60.7851         0         0         0         0         0         0         0         0         0 -234.1003         0
    412.5490         0         0         0         0         0         0         0         0         0 -230.7409         0
           0  367.5510         0 -374.6047         0         0         0         0         0         0         0         0
           0         0  644.8386         0         0         0         0         0 -435.6638         0         0         0
           0         0 -112.3009  589.6873 -326.0293         0         0         0         0         0         0         0
           0         0         0         0  326.0293 -187.5785         0         0         0 -141.5426         0         0
           0         0         0         0         0  520.5785         0         0 -120.8000         0         0         0
           0         0         0 -115.0826         0         0  181.0000  -66.5762         0         0         0         0
           0         0         0         0         0         0         0  194.5762         0         0 -195.1588         0
           0         0         0         0         0         0         0         0  556.4638 -158.4574         0         0
           0         0         0         0         0         0         0         0         0  300.0000         0         0
           0         0         0         0         0         0         0         0         0         0  660.0000         0
           0         0 -215.5378         0         0         0         0         0         0         0         0  212.3191
```

```
=================================================
|     Using Proportional Sharing allocation      |
=================================================
         Bus    Generation    Load      Losses
          #       P(MW)       P(MW)      P(MW)
        -----   ----------   -------   --------
          1       172.00     108.00     0.3655
          2       172.00      97.00     0.0232
          3         0.00     180.00     5.7955
          4         0.00      74.00     2.2222
          5         0.00      71.00     1.3973
          6         0.00     136.00     5.2805
          7       240.00     125.00     0.0000
          8         0.00     171.00     5.0667
          9         0.00     175.00     6.0460
         10         0.00     195.00     6.7622
         11         0.00       0.00     0.0000
         12         0.00       0.00     0.0000
         13       187.25     265.00     3.4933
         14         0.00     194.00     6.4915
         15       215.00     317.00     3.6309
         16       155.00     100.00     1.4002
         17         0.00       0.00     0.0000
         18       400.00     333.00     0.4186
         19         0.00     181.00     2.4695
         20         0.00     128.00     0.3833
         21       400.00       0.00     0.0000
         22       300.00       0.00     0.0000
         23       660.00       0.00     0.0000
         24         0.00       0.00     0.0000
        -----   ----------   -------   --------
        Total    2901.25     2850.00    51.2464
```

3. Development of Algorithm of Learning Coefficients in MATLAB

```
showmeimp1=imp1;
after100=xload1/100;
for i=1:nb
    load=after100(i,:);
    loadt=load';
    loss=showmeimp1(i,:);
    losst=loss';
    oneoverload=[];
    for j=1:10
    if loadt(j,1)>0
    oneoverload(j,1)=1/loadt(j,1);
    else
        oneoverload(j,1)=0;
    end
    end
    yi=ones(10,1);
    combined=[oneoverload yi loadt];
    if after100(i,1)>0
    x = inv(combined' * combined) * combined' *losst;
    x=x';
    else
        x=[0 0 0];
    end
    xo=[xo; x];
end

    fprintf(1, '\n===============================================');
    fprintf(1, '\n|           Learning Coefficients             |');
    fprintf(1, '\n===============================================');
    fprintf(1, '\n    Bus #      ALPHA        BETA        GAMMA ');
    fprintf(1, '\n   -------  -----------  ----------  -----------');
    busno=1:nb;
    busno=busno';
    fprintf(1, '\n   %4d    %10.4f  %10.4f   %10.4f',[busno xo]')
    fprintf(1, '\n');
    fprintf(1, '\n');
```

3.1 Learning Coefficients Algorithm Implementation in MATLAB

```
===============================================
|           Learning Coefficients             |
===============================================
    Bus #      ALPHA        BETA        GAMMA
   -------  -----------  ----------  -----------
      1        0.0028     -0.0036       0.2220
      2       -0.0007      0.0077       0.0146
      3       -0.0023      0.0064       3.1821
      4       -0.0486      0.6075       1.0500
      5       -0.0373      0.4802       1.3382
      6       -0.1909      1.2512       2.1787
      7        0.0000      0.0000       0.0000
      8       -0.2535      1.3214       1.6075
      9       -0.5834      3.1702       2.8254
     10       -0.7433      3.5729       1.8765
     11        0.0000      0.0000       0.0000
     12        0.0000      0.0000       0.0000
     13       39.8568    227.6785    -117.9482
     14       -0.4246      1.8796       3.1259
     15       -0.0710      0.1893       1.1359
     16       -0.0430      0.3719       1.2976
     17        0.0000      0.0000       0.0000
     18        0.0486     -0.1215       0.1269
     19       -0.3306      1.5801       1.5900
     20       -0.2699      1.8349       1.4838
     21        0.0000      0.0000       0.0000
     22        0.0000      0.0000       0.0000
     23        0.0000      0.0000       0.0000
     24        0.0000      0.0000       0.0000
```

4. Application of Learning Coefficients on Loss Prediction

```
abba=casestudy
runpf
xload2=xload2;
uload3=xload2/100;
uload1=[];
for i=1:nb
    if uload3(i,1)>0
    uload1(i,1)=1/uload3(i,1);
    else
        uload1(i,1)=0;
    end
end
uload2=ones(nb,1);
uload=[uload1 uload2 uload3];
preloss=uload.*xo;
tpreloss=preloss(:,1)+preloss(:,2)+preloss(:,3);
ttpreloss=sum(tpreloss);
tactloss=imp2;
ttactloss=sum(tactloss);
MAPE=[];
for i=1:nb
    if tactloss(i,1)>0
MAPE(i,1)=abs(tactloss(i,1)-tpreloss(i,1))/tactloss(i,1)*100;
    else
        MAPE(i,1)=0;
    end
end
TMAPE=abs(ttactloss-ttpreloss)/ttactloss*100;

fprintf(1, '\n');
fprintf(1, '\n============================================');
fprintf(1, '\n|                 Forecasting Result        |');
fprintf(1, '\n============================================');
fprintf(1, '\n      Bus     Actual    Predicted     MAPE ');
fprintf(1, '\n       #      Losses     Losses      errors ');
fprintf(1, '\n     -----  ----------  ----------  ----------');
for i=1:nb
fprintf(1, '\n     %4d  %10.4f  %10.4f  %10.4f'    ,[i, tactloss(i,1), tpreloss(i,1), MAPE(i,1)]'  );
end
fprintf(1, '\n      -----  ----------  ----------  ----------');
fprintf(1, '\n      Total%10.4f  %10.4f  %10.4f'    ,[ttactloss, ttpreloss, TMAPE]'  );
fprintf(1, '\n');
fprintf(1, '\n');
```

4.1 Comparison of actual losses and predicted losses

```
============================================
|                 Forecasting Result        |
============================================
      Bus     Actual    Predicted     MAPE
       #      Losses     Losses      errors
     -----  ----------  ----------  ----------
        1      0.0959      0.0959      0.0073
        2      0.0114      0.0114      0.0013
        3      2.2175      2.2175      0.0000
        4      0.7383      0.7382      0.0061
        5      0.7117      0.7117      0.0055
        6      2.0338      2.0337      0.0056
        7      0.0000      0.0000      0.0000
        8      2.0007      2.0006      0.0057
        9      4.2195      4.2193      0.0035
       10      4.0018      4.0016      0.0047
       11      0.0000      0.0000      0.0000
       12      0.0000      0.0000      0.0000
       13    145.7350    145.7459      0.0075
       14      3.6581      3.6579      0.0049
       15      1.5234      1.5233      0.0012
       16      0.7625      0.7624      0.0045
       17      0.0000      0.0000      0.0000
       18      0.0795      0.0795      0.0164
       19      2.2204      2.2202      0.0066
       20      2.0239      2.0237      0.0081
       21      0.0000      0.0000      0.0000
       22      0.0000      0.0000      0.0000
       23      0.0000      0.0000      0.0000
       24      0.0000      0.0000      0.0000
     -----  ----------  ----------  ----------
     Total  172.0332    172.0429      0.0057
```

5. Integration of all developed subroutines, forming a prediction program

```
%==============================================================
%|                          Main Program                       |
%==============================================================

global total abba imp2 xload2 nb extrao   %global variables imp2=Loss, xload2=teeload

abba=total
%\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
%total=0.93389296    %<-------- Subject to different case studies  )
%///////////////////////////////////////////////////////////////

%=============================================================
%|                   BOX of Learning Coefficients             |
%=============================================================

LC3_4=[0.002809744412716,-0.003550765468696,0.221977115622708;-
0.000657719328608,0.007707451204133,0.014582690812547;-0.002266962775632,0.006374451500393,3.18206528751515;-
0.048550624016809,0.607531573063866,1.049988604463403;-0.037276026810479,0.480194065840019,1.338172877488686;-
0.190881749861743,1.251248707620881,2.178706086059715;0,0,0;-
0.253506519048822,1.321386605438375,1.607455578714073;-0.583411899085838,3.170155625917729,2.825403277793874;-
0.743286925688952,3.572910486959825,1.876479489529138;0,0,0;0,0,0;39.85683967283665,227.67849845945264,-
117.94815578968978;-0.424586591704261,1.879592869983083,3.125938907498083;-
0.071028767522869,0.189283707082973,1.135860268791925;-
0.042980141088087,0.37194953774195,1.297603571945087;0,0,0;0.048592565141024,-
0.121519816072646,0.126857447755513;-0.330566038757092,1.580055817996209,1.58995789038348;-
0.269942089979936,1.834939140213502,1.483846993797511;0,0,0;0,0,0;0,0,0;0,0,0;];

LC4_5=[0.015849259745171,-0.062603788563808,0.288970643328178;-
0.000709731743802,0.008005644565748,0.014159534824319;-0.001711527342275,0.005216417979718,3.182610019272793;-
0.090559127646703,0.888815481044174,0.578123759759474;-0.074399454763017,0.738551734386064,0.887707181218645;-
0.408244397275166,2.039553181778232,1.462452556441028;0,0,0;-
0.516483385955103,2.081378581583507,1.057197601265744;-
0.664013292348045,3.433615866450326,2.613704184598348;6.217228668111415,-
12.720552508073204,11.386120998576299;0,0,0;0,0,0;67.32641463150904,175.01555183429102,-92.68673597972256;-
0.885051295364192,3.052719589632261,2.377150595871138;-0.152287763830349,0.315911722815883,1.08642321025769;-
0.088160394422304,0.595347263306249,1.020861815776939;0,0,0;0.108691212133384,-
0.210576360180844,0.15991947455558;-0.686780903858725,2.552619530770626,0.924700015298074;-
0.542950214796252,2.890616682388942,0.46114217127403;0,0,0;0,0,0;0,0,0;0,0,0;];

LC5_6=[0.048234626914753,-0.180610655978845,0.396611431098281;-
0.000275887929106,0.006265813825222,0.015905564468658;0.016304913052163,-0.033662277166081,3.203605105473267;-
0.142747487938251,1.169522277753156,0.200157119449488;-
0.131360052443412,1.056446267686101,0.443564305686329;0.566144097668365,-
2.078149101195905,5.374921699316942;0,0,0;-
0.911623254875692,2.997455882487379,0.525524961540908;3.100648663466863,-4.694912147166183,7.001472945836395;-
4.221074728061904,8.137781321671508,0.982821601463756;0,0,0;0,0,0;127.82395359078131,85.77712314656665,-
59.75098455353279;-1.576222099126451,4.464833236506172,1.654902575710849;-
0.277353831680995,0.472232277876009,1.037510156398498;-
0.154316097886677,0.85767288447103,0.760458503817909;0,0,0;0.206807658335373,-
0.327212776111871,0.194629720210587;-1.229626178506429,3.740961620568268,0.273462408092723;-
0.889556567339638,3.968934366595851,-0.378648561164965;0,0,0;0,0,0;0,0,0;0,0,0;];

LC6_7=[0.119515958479589,-
0.397667770511335,0.562002194712046;0.001156759917288,0.001421302140876,0.020004605134007;0.086759442149553,-
0.161775437888461,3.261893986628826;-0.180514670628478,1.341228020300851,0.004872558582508;-
0.206755538352016,1.408243347275296,0.032806163001139;0.262011641176063,-
1.248755449434069,4.816221601028696;0,0,0;-1.414148751596428,3.971611632059908,0.052977935727809;-
17.129250096331702,29.780004190845368,-7.529076503279897;-
3.274314881805202,7.392768470740621,0.924581558674517;0,0,0;0,0,0;233.26035883499196,-42.99823880605771,-
20.403608543152487;-2.520991636111799,6.07610787527809,0.967256679661992;-
0.45211038078916,0.654579371385206,0.989898228132705;-
0.241539404047492,1.14642513354587,0.521253985742941;0,0,0;0.352899172491381,-
0.472203388924199,0.230638291440043;-1.993342329032248,5.13637636600725,-0.364545899355009;-
0.960900528033638,4.17908748617573,-0.531134040069802;0,0,0;0,0,0;0,0,0;0,0,0;];

LC7_8=[0.268300856883964,-0.786705112629772,0.816484322489211;0.004547842685565,-
0.008443685796175,0.027183942995522;0.279690270344851,-0.463844795606099,3.380207692013693;6.452798470428074,-
25.18065095625354,26.4237086148183;-0.27472078889442,1.682231906917395,-0.243493300027261;5.41104797367807,-
11.73214110017113 4,10.153635888048921;0,0,0;26.72450135174944,-41.33491848651446,18.283542258380855;-
31.858196502151884,54.66669805884,-18.028922816741453;15.570239113084062,-
19.412096458708152,10.460666926469298;0,0,0;0,0,0;460.8062601841272,-282.5156321032686,42.65659995784124;-
3.715056815339707,7.824016628140488,0.327150758921626;-0.678515002959417,0.85733989270503,0.944470183908314;-
0.345033411606096,1.440602444971265,0.312062353610827;0,0,0;0.556280742250661,-
0.645447741246832,0.267557211526088;-3.001172081308224,6.716767076855673,-0.98453530675544;-
99.91038799269744,217.28149061081768,-115.2279541737706;0,0,0;0,0,0;0,0,0;0,0,0;];

LC8_9=[0.570262227958003,-1.478475794705047,1.212888033173899;0.011724195415904,-
0.026744382951954,0.038857289587369;0.726231681137065,-1.0771423946063,3.590896790281554;0.961099755608539,-
5.564458620352473,8.942516374873964;-0.233837184496307,1.545656349259754,-0.12962125579555;-
1.623885867522058,1.650464260720733,3.808798194294187;0,0,0;35.5657695969276 6,-
54.42892482407018,23.090499268223585;6.29595008908402,-2.021652145726305,3.008910779981228;-
19.056927167388896,26.530559598085595,-4.738558135007964;0,0,0;0,0,0;354.14283291732283,-
175.06856602360835,15.700563305951196;-5.103125009370497,9.604248952186527,-0.243951091703541;-
0.952191303572856,1.072038447777883,0.9023400697119;-
```

```
0.450168934389292,1.702693356737889,0.148636342306944;0,0,0;0.824647024224717,-
0.845677163668485,0.304925190141454;21.308980093909568,-
25.118082825947372,9.42801139560297;6.638045392176876,0.807165453833,-
5.422449622985625;0,0,0;0,0,0;0,0,0;0,0,0;];

LC9_10=[1.173533288910569,-2.708197455897803,1.839817523227191;0.026422174698583,-
0.060094271304972,0.05778282586703;1.666485761740597,-2.226874043955804,3.942509269422466;-
9.900113040018752,26.675174316683485,-14.97113833929537;-2.93179495081319,6.57632094250737,-
1.475088146902594;-52.61521789523362,84.6777646231361,-29.945404828905726;0,0,0;-
27.662152695247464,30.42105777680397,-5.370730444231922;-121.49700625440757,160.04122301146268,-
48.340466047863664;-167.90406622210963,196.21145042268506,-53.017234006721225;0,0,0;0,0,0;1173.7782499644422,-
859.8671446546682,158.6823192639221;-6.528021970435493,11.231425775759982,-0.708687293809202;-
1.258120015022763,1.285606728694554,0.865051491105619;-
0.521406692462943,1.8615933290342,0.059997920137254;0,0,0;1.162434180520986,-
1.069886732617105,0.342146221657024;37.95506550672553,-
32.838280999142505,7.921116141820742;3.824046091817671,-
6.93206295132747,3.38119118651497;0,0,0;0,0,0;0,0,0;0,0,0;];

LC10_11=[2.37244184638959,-4.909373023500266,2.850481872821089;0.057270521048713,-
0.123117177450717,0.089982199225471;3.543725421098166,-
4.295004810997486,4.512305970893847;11.157800659128998,-
30.335681519407316,23.629462940696573;4.684147545470973,-
14.235494056081814,12.733074742771716;43.478321031376296,-
58.13461173274406,23.12824096313761;0,0,0;62.07221066295681,-
74.3732544314458,25.235780421822486;87.37493271558466,-
81.42943040920635,21.462855174256813;117.28195323434686,-
100.6611520309188,24.25157434033935;0,0,0;0,0,0;293.65575102277035,-186.50149262169663,29.868704731632118;-
7.613613158278486,12.351764444035407,-0.997824505237348;-
1.56230314074919,1.476994454657819,0.834936735377764;-
0.475653074166456,1.773186546447274,0.102680805156756;0,0,0;1.567766009352018,-
1.312206684489523,0.378375200399699;55.7057698042819,-52.37044816732592,13.296089380652509;3.091477188869259,-
5.767281123870546,2.918214798586635;0,0,0;0,0,0;0,0,0;0,0,0;];
LC=[LC3_4 LC4_5 LC5_6 LC6_7 LC7_8 LC8_9 LC9_10 LC10_11];

%=====================================================================
%|                          Run Prediction                           |
%=====================================================================

floorabba=floor(abba*10)/10;
chooseLC=(floorabba*10-3)*3+1;
selectedLC=[LC(:,chooseLC) LC(:,chooseLC+1) LC(:,chooseLC+2)];
xo=selectedLC;

    runpf
    xload2=xload2;
    uload3=xload2/100;
    uload1=[];
    for i=1:nb
        if uload3(i,1)>0
        uload1(i,1)=1/uload3(i,1);
        else
            uload1(i,1)=0;
        end
    end
    uload2=ones(nb,1);
    uload=[uload1 uload2 uload3];
    preloss=uload.*xo;
    tpreloss=preloss(:,1)+preloss(:,2)+preloss(:,3);
    ttpreloss=sum(tpreloss);
    tactloss=imp2;
    ttactloss=sum(tactloss);
    MAPE=[];
    for i=1:nb
        if tactloss(i,1)>0
    MAPE(i,1)=abs(tactloss(i,1)-tpreloss(i,1))/tactloss(i,1)*100;
        else
            MAPE(i,1)=0;
        end
    end
    TMAPE=abs(ttactloss-ttpreloss)/ttactloss*100;


%=====================================================================
%|                          Print Results                            |
%=====================================================================

    fprintf(1, '\n');
    fprintf(1, '\n===================================================');
    fprintf(1, '\n|                 Forecasting Result              |');
    fprintf(1, '\n===================================================');
    fprintf(1, '\n     Bus     Actual    Predicted      MAPE ');
    fprintf(1, '\n      #      Losses     Losses       errors ');
    fprintf(1, '\n     -----  ----------  ----------  ----------');
    for i=1:nb
    fprintf(1, '\n     %4d  %10.4f  %10.4f  %10.4f'   ,[i, tactloss(i,1), tpreloss(i,1), MAPE(i,1)]'  );
    end
    fprintf(1, '\n     -----  ----------  ----------  ----------');
    fprintf(1, '\n     Total%10.4f  %10.4f  %10.4f'   ,[ttactloss, ttpreloss, TMAPE]'  );
    fprintf(1, '\n');
    fprintf(1, '\n');
```

6. Integration of all developed subroutines into a MATLAB GUI

```
%% Matlab GUI for Loss Prediction

function varargout = Loss_Analyzer_Predictor_v2011(varargin)

global total extrao

% LOSS_ANALYZER_PREDICTOR_V2011 M-file for Loss_Analyzer_Predictor_v2011.fig
%      LOSS_ANALYZER_PREDICTOR_V2011, by itself, creates a new LOSS_ANALYZER_PREDICTOR_V2011 or raises the 
existing
%      singleton*.
%
%      H = LOSS_ANALYZER_PREDICTOR_V2011 returns the handle to a new LOSS_ANALYZER_PREDICTOR_V2011 or the 
handle to
%      the existing singleton*.
%
%      LOSS_ANALYZER_PREDICTOR_V2011('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in LOSS_ANALYZER_PREDICTOR_V2011.M with the given input arguments.
%
%      LOSS_ANALYZER_PREDICTOR_V2011('Property','Value',...) creates a new LOSS_ANALYZER_PREDICTOR_V2011 or 
raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before TeeMarket_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to Loss_Analyzer_Predictor_v2011_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Loss_Analyzer_Predictor_v2011

% Last Modified by GUIDE v2.5 13-Apr-2011 14:21:48

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Loss_Analyzer_Predictor_v2011_OpeningFcn, ...
                   'gui_OutputFcn',  @Loss_Analyzer_Predictor_v2011_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 Loss_Analyzer_Predictor_v2011 is made visible.

%% Opening Function
function Loss_Analyzer_Predictor_v2011_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Loss_Analyzer_Predictor_v2011 (see VARARGIN)
%set(handles.text89,'String',date);

[a,map]=imread('Table.jpg');
[r,c,d]=size(a);
x=ceil(r/100);
y=ceil(c/100);
g=a(1:x:end,1:y:end,:);
g(g==255)=5.5*255;
set(handles.populatetable_pushbutton,'CData',g);

[a,map]=imread('dinos.jpg');
[r,c,d]=size(a);
x=ceil(r/100);
y=ceil(c/100);
g=a(1:x:end,1:y:end,:);
g(g==255)=5.5*255;
set(handles.textmessage_pushbutton,'CData',g);

[a,map]=imread('robot2.jpg');
[r,c,d]=size(a);
x=ceil(r/100);
y=ceil(c/100);
g=a(1:x:end,1:y:end,:);
g(g==255)=5.5*255;
set(handles.sendemail_pushbutton,'CData',g);
```

```
[a,map]=imread('1step.jpg');
[r,c,d]=size(a);
x=ceil(r/100);
y=ceil(c/100);
g=a(1:x:end,1:y:end,:);
g(g==255)=5.5*255;
set(handles.htmlpublisher_pushbutton,'CData',g);

[a,map]=imread('upp.jpg');
[r,c,d]=size(a);
x=ceil(r/100);
y=ceil(c/100);
g=a(1:x:end,1:y:end,:);
g(g==255)=5.5*255;
set(handles.pushbutton9,'CData',g);

[a,map]=imread('downn.jpg');
[r,c,d]=size(a);
x=ceil(r/100);
y=ceil(c/100);
g=a(1:x:end,1:y:end,:);
g(g==255)=5.5*255;
set(handles.pushbutton11,'CData',g);

[a,map]=imread('upp.jpg');
[r,c,d]=size(a);
x=ceil(r/100);
y=ceil(c/100);
g=a(1:x:end,1:y:end,:);
g(g==255)=5.5*255;
set(handles.pushbutton12,'CData',g);

[a,map]=imread('downn.jpg');
[r,c,d]=size(a);
x=ceil(r/100);
y=ceil(c/100);
g=a(1:x:end,1:y:end,:);
g(g==255)=5.5*255;
set(handles.pushbutton13,'CData',g);

[a,map]=imread('upp.jpg');
[r,c,d]=size(a);
x=ceil(r/100);
y=ceil(c/100);
g=a(1:x:end,1:y:end,:);
g(g==255)=5.5*255;
set(handles.pushbutton14,'CData',g);

[a,map]=imread('downn.jpg');
[r,c,d]=size(a);
x=ceil(r/100);
y=ceil(c/100);
g=a(1:x:end,1:y:end,:);
g(g==255)=5.5*255;
set(handles.pushbutton15,'CData',g);

%load the background image into Matlab
%if image is not in the same directory as the GUI files, you must use the
%full path name of the iamge file
backgroundImage = importdata('ha7.png');
%select the axes
axes(handles.axes14);
%place image onto the axes
image(backgroundImage);
%remove the axis tick marks
axis off

% Choose default command line output for Loss_Analyzer_Predictor_v2011
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

%PUT Splash

% UIWAIT makes Loss_Analyzer_Predictor_v2011 wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = Loss_Analyzer_Predictor_v2011_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


%% Week, day & hour
```

```
function week_editText_Callback(hObject, eventdata, handles)
% hObject    handle to week_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of week_editText as text
%        str2double(get(hObject,'String')) returns contents of week_editText as a double
%===============================================
%store the contents of input1_editText as a string. if the string
%is not a number then input will be empty
input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default input1_editText to zero
if (isempty(input))|| (input>52) || (input<1)
     set(hObject,'String','1')
end
guidata(hObject, handles);
%===============================================


% --- Executes during object creation, after setting all properties.
function week_editText_CreateFcn(hObject, eventdata, handles)
% hObject    handle to week_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function day_editText_Callback(hObject, eventdata, handles)
% hObject    handle to day_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of day_editText as text
%        str2double(get(hObject,'String')) returns contents of day_editText as a double
%===============================================
%store the contents of input1_editText as a string. if the string
%is not a number then input will be empty
input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default input1_editText to zero
if (isempty(input)) || (input>7) || (input<1)
     set(hObject,'String','1')
end
guidata(hObject, handles);
%===============================================

% --- Executes during object creation, after setting all properties.
function day_editText_CreateFcn(hObject, eventdata, handles)
% hObject    handle to day_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function hour_editText_Callback(hObject, eventdata, handles)
% hObject    handle to hour_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of hour_editText as text
%        str2double(get(hObject,'String')) returns contents of hour_editText as a double
%===============================================
%store the contents of input1_editText as a string. if the string
%is not a number then input will be empty
input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default input1_editText to zero
if (isempty(input)) || (input>23) || (input<0)
     set(hObject,'String','0')
end
guidata(hObject, handles);
%===============================================

% --- Executes during object creation, after setting all properties.
function hour_editText_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hour_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
    % Hint: edit controls usually have a white background on Windows.
    %       See ISPC and COMPUTER.
    if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

    %% Pushbuttons
    % --- Executes on button press in getabba_pushbutton.
    function getabba_pushbutton_Callback(hObject, eventdata, handles)
    global total


    % hObject    handle to getabba_pushbutton (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    a = get(handles.week_editText,'String');
    b = get(handles.day_editText,'String');
    c = get(handles.hour_editText,'String');
    % a and b are variables of Strings type, and need to be converted
    % to variables of Number type before they can be added together

    week=str2num(a);
    day=str2num(b);
    hour=str2num(c);

    tweek=[0.862;0.9;0.878;0.834;0.88;0.841;0.832;0.806;0.74;0.737;0.715;0.727;0.704;0.75;0.721;0.8;0.754;0.837
;0.87;0.88;0.856;0.811;0.9;0.887;0.896;0.861;0.755;0.816;0.801;0.88;0.722;0.776;0.8;0.729;0.726;0.705;0.78;0.6
95;0.724;0.724;0.743;0.744;0.8;0.881;0.885;0.909;0.94;0.89;0.942;0.97;1;0.952;];
    tday=[0.93;1;0.98;0.96;0.94;0.77;0.75;];
    thour=[0.67,0.78,0.64,0.74,0.63,0.75;0.63,0.72,0.6,0.7,0.62,0.73;0.6,0.68,0.58,0.66,0.6,0.69;0.59,0.66,0.56
,0.65,0.58,0.66;0.59,0.64,0.56,0.64,0.59,0.65;0.6,0.65,0.58,0.62,0.65,0.65;0.74,0.66,0.64,0.62,0.72,0.68;0.86,
0.7,0.76,0.66,0.85,0.74;0.95,0.8,0.87,0.81,0.95,0.83;0.96,0.88,0.95,0.86,0.99,0.89;0.96,0.9,0.99,0.91,1,0.92;0
.95,0.91,1,0.93,0.99,0.94;0.95,0.9,0.99,0.93,0.93,0.91;0.95,0.88,1,0.92,0.92,0.9;0.93,0.87,1,0.91,0.9,0.9;0.94
,0.87,0.97,0.91,0.88,0.86;0.99,0.91,0.96,0.92,0.9,0.85;1,1,0.96,0.94,0.92,0.88;1,0.99,0.93,0.95,0.96,0.92;0.96
,0.97,0.92,0.95,0.98,1;0.91,0.94,0.92,1,0.96,0.97;0.83,0.92,0.93,0.93,0.9,0.95;0.73,0.87,0.87,0.88,0.8,0.9;0.6
3,0.81,0.72,0.8,0.7,0.85;];

    m=[];
    if day<6
        m=1;
    else
        m=2;
    end
    if (week>17)&&(week<31)
    m=m+2;
    end
    if (week>8)&&(week<18)||(week>30)&&(week<44)
    m=m+4;
    end
    hour=hour+1;
    ahour=thour(hour,m);
    aweek=tweek(week,1);
    aday=tday(day,1);

    total = 1.1*aweek*aday*ahour;
    %abba=total;
    d = num2str(total);
    % need to convert the answer back into String type to display it
    set(handles.abba_staticText,'String',['Percentage = ',d]);

    guidata(hObject, handles);

    % --- Executes on button press in populatetable_pushbutton.
    function populatetable_pushbutton_Callback(hObject, eventdata, handles)
    % hObject    handle to populatetable_pushbutton (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    Iwanwan
    extraol1=extrao(1,1);
    extraol1d = num2str(extraol1);
    set(handles.text10,'String',extraol1d);
    guidata(hObject, handles);

    extraol1=extrao(2,1);
    extraol1d = num2str(extraol1);
    set(handles.text11,'String',extraol1d);
    guidata(hObject, handles);

    extraol1=extrao(3,1);
    extraol1d = num2str(extraol1);
    set(handles.text12,'String',extraol1d);
    guidata(hObject, handles);

    extraol1=extrao(4,1);
    extraol1d = num2str(extraol1);
    set(handles.text13,'String',extraol1d);
    guidata(hObject, handles);

    extraol1=extrao(5,1);
    extraol1d = num2str(extraol1);
    set(handles.text14,'String',extraol1d);
```

```
guidata(hObject, handles);

extraoll=extrao(6,1);
extraolld = num2str(extraoll);
set(handles.text15,'String',extraolld);
guidata(hObject, handles);

extraoll=extrao(7,1);
extraolld = num2str(extraoll);
set(handles.text16,'String',extraolld);
guidata(hObject, handles);

extraoll=extrao(8,1);
extraolld = num2str(extraoll);
set(handles.text17,'String',extraolld);
guidata(hObject, handles);

extraoll=extrao(9,1);
extraolld = num2str(extraoll);
set(handles.text18,'String',extraolld);
guidata(hObject, handles);

extraoll=extrao(10,1);
extraolld = num2str(extraoll);
set(handles.text19,'String',extraolld);
guidata(hObject, handles);

extraoll=extrao(11,1);
extraolld = num2str(extraoll);
set(handles.text20,'String',extraolld);
guidata(hObject, handles);

extraoll=extrao(12,1);
extraolld = num2str(extraoll);
set(handles.text21,'String',extraolld);
guidata(hObject, handles);

extraoll=extrao(13,1);
extraolld = num2str(extraoll);
set(handles.text22,'String',extraolld);
guidata(hObject, handles);

extraoll=extrao(14,1);
extraolld = num2str(extraoll);
set(handles.text23,'String',extraolld);
guidata(hObject, handles);

extraoll=extrao(15,1);
extraolld = num2str(extraoll);
set(handles.text24,'String',extraolld);
guidata(hObject, handles);

extraoll=extrao(16,1);
extraolld = num2str(extraoll);
set(handles.text25,'String',extraolld);
guidata(hObject, handles);

extraoll=extrao(17,1);
extraolld = num2str(extraoll);
set(handles.text26,'String',extraolld);
guidata(hObject, handles);

extraoll=extrao(18,1);
extraolld = num2str(extraoll);
set(handles.text35,'String',extraolld);
guidata(hObject, handles);

extraoll=extrao(19,1);
extraolld = num2str(extraoll);
set(handles.text36,'String',extraolld);
guidata(hObject, handles);

extraoll=extrao(20,1);
extraolld = num2str(extraoll);
set(handles.text37,'String',extraolld);
guidata(hObject, handles);

extraoll=extrao(21,1);
extraolld = num2str(extraoll);
set(handles.text38,'String',extraolld);
guidata(hObject, handles);

extraoll=extrao(1,2);
extraolld = num2str(extraoll);
set(handles.text39,'String',extraolld);
guidata(hObject, handles);

extraoll=extrao(2,2);
extraolld = num2str(extraoll);
set(handles.text40,'String',extraolld);
guidata(hObject, handles);
```

```
extrao11=extrao(3,2);
extrao11d = num2str(extrao11);
set(handles.text41,'String',extrao11d);
guidata(hObject, handles);

extrao11=extrao(4,2);
extrao11d = num2str(extrao11);
set(handles.text42,'String',extrao11d);
guidata(hObject, handles);

extrao11=extrao(5,2);
extrao11d = num2str(extrao11);
set(handles.text43,'String',extrao11d);
guidata(hObject, handles);

extrao11=extrao(6,2);
extrao11d = num2str(extrao11);
set(handles.text44,'String',extrao11d);
guidata(hObject, handles);

extrao11=extrao(7,2);
extrao11d = num2str(extrao11);
set(handles.text45,'String',extrao11d);
guidata(hObject, handles);

extrao11=extrao(8,2);
extrao11d = num2str(extrao11);
set(handles.text46,'String',extrao11d);
guidata(hObject, handles);

extrao11=extrao(9,2);
extrao11d = num2str(extrao11);
set(handles.text47,'String',extrao11d);
guidata(hObject, handles);

extrao11=extrao(10,2);
extrao11d = num2str(extrao11);
set(handles.text48,'String',extrao11d);
guidata(hObject, handles);

extrao11=extrao(11,2);
extrao11d = num2str(extrao11);
set(handles.text49,'String',extrao11d);
guidata(hObject, handles);

extrao11=extrao(12,2);
extrao11d = num2str(extrao11);
set(handles.text50,'String',extrao11d);
guidata(hObject, handles);

extrao11=extrao(13,2);
extrao11d = num2str(extrao11);
set(handles.text51,'String',extrao11d);
guidata(hObject, handles);

extrao11=extrao(14,2);
extrao11d = num2str(extrao11);
set(handles.text52,'String',extrao11d);
guidata(hObject, handles);

extrao11=extrao(15,2);
extrao11d = num2str(extrao11);
set(handles.text53,'String',extrao11d);
guidata(hObject, handles);

extrao11=extrao(16,2);
extrao11d = num2str(extrao11);
set(handles.text54,'String',extrao11d);
guidata(hObject, handles);

extrao11=extrao(17,2);
extrao11d = num2str(extrao11);
set(handles.text55,'String',extrao11d);
guidata(hObject, handles);

extrao11=extrao(18,2);
extrao11d = num2str(extrao11);
set(handles.text56,'String',extrao11d);
guidata(hObject, handles);

extrao11=extrao(19,2);
extrao11d = num2str(extrao11);
set(handles.text57,'String',extrao11d);
guidata(hObject, handles);

extrao11=extrao(20,2);
extrao11d = num2str(extrao11);
set(handles.text58,'String',extrao11d);
guidata(hObject, handles);

extrao11=extrao(21,2);
extrao11d = num2str(extrao11);
```

```
set(handles.text59,'String',extraolld);
guidata(hObject, handles);

extraol1=extrao(1,3);
extraolld = num2str(extraol1);
set(handles.text60,'String',extraolld);
guidata(hObject, handles);

extraol1=extrao(2,3);
extraolld = num2str(extraol1);
set(handles.text61,'String',extraolld);
guidata(hObject, handles);

extraol1=extrao(3,3);
extraolld = num2str(extraol1);
set(handles.text62,'String',extraolld);
guidata(hObject, handles);

extraol1=extrao(4,3);
extraolld = num2str(extraol1);
set(handles.text63,'String',extraolld);
guidata(hObject, handles);

extraol1=extrao(5,3);
extraolld = num2str(extraol1);
set(handles.text64,'String',extraolld);
guidata(hObject, handles);

extraol1=extrao(6,3);
extraolld = num2str(extraol1);
set(handles.text65,'String',extraolld);
guidata(hObject, handles);

extraol1=extrao(7,3);
extraolld = num2str(extraol1);
set(handles.text66,'String',extraolld);
guidata(hObject, handles);

extraol1=extrao(8,3);
extraolld = num2str(extraol1);
set(handles.text67,'String',extraolld);
guidata(hObject, handles);

extraol1=extrao(9,3);
extraolld = num2str(extraol1);
set(handles.text68,'String',extraolld);
guidata(hObject, handles);

extraol1=extrao(10,3);
extraolld = num2str(extraol1);
set(handles.text69,'String',extraolld);
guidata(hObject, handles);

extraol1=extrao(11,3);
extraolld = num2str(extraol1);
set(handles.text70,'String',extraolld);
guidata(hObject, handles);

extraol1=extrao(12,3);
extraolld = num2str(extraol1);
set(handles.text71,'String',extraolld);
guidata(hObject, handles);

extraol1=extrao(13,3);
extraolld = num2str(extraol1);
set(handles.text72,'String',extraolld);
guidata(hObject, handles);

extraol1=extrao(14,3);
extraolld = num2str(extraol1);
set(handles.text73,'String',extraolld);
guidata(hObject, handles);

extraol1=extrao(15,3);
extraolld = num2str(extraol1);
set(handles.text74,'String',extraolld);
guidata(hObject, handles);

extraol1=extrao(16,3);
extraolld = num2str(extraol1);
set(handles.text75,'String',extraolld);
guidata(hObject, handles);

extraol1=extrao(17,3);
extraolld = num2str(extraol1);
set(handles.text76,'String',extraolld);
guidata(hObject, handles);

extraol1=extrao(18,3);
extraolld = num2str(extraol1);
set(handles.text77,'String',extraolld);
guidata(hObject, handles);
```

```
extraol1=extrao(19,3);
extraol1d = num2str(extraol1);
set(handles.text78,'String',extraol1d);
guidata(hObject, handles);

extraol1=extrao(20,3);
extraol1d = num2str(extraol1);
set(handles.text79,'String',extraol1d);
guidata(hObject, handles);

extraol1=extrao(21,3);
extraol1d = num2str(extraol1);
set(handles.text80,'String',extraol1d);
guidata(hObject, handles);

try
    v=get(handles.populatetable_pushbutton,'Value');

    while v


        v=get(handles.populatetable_pushbutton,'Value');
        t=clock;
        set(handles.text89,'String',date);
        set(handles.text87,'String',num2str(fix(t(4))))
        set(handles.text90,'String',[num2str(fix(t(5))),':'])
        set(handles.text88,'String',num2str(fix(t(6))))
        drawnow
    end
catch
end

% --- Executes on button press in textmessage_pushbutton.
function textmessage_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to textmessage_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%sendmailgui
Loss_Graph

% --- Executes on button press in sendemail_pushbutton.
function sendemail_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to sendemail_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%sendmail22
Sendgmail
%speak('Tee Ping Hong! Well done!');
%BlockClock

% --- Executes on button press in htmlpublisher_pushbutton.
function htmlpublisher_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to htmlpublisher_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Publish

% --------------------------------------------------------------------
function Info_ClickedCallback(hObject, eventdata, handles)
% hObject    handle to Info (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
IEEE_24bus_RTS

% --------------------------------------------------------------------
function uitoggletool2_ClickedCallback(hObject, eventdata, handles)
% hObject    handle to uitoggletool2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
sungnying

% --- Executes during object creation, after setting all properties.
function axes2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes2

% --- Executes during object creation, after setting all properties.
function text10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% --- Executes during object creation, after setting all properties.
function abba_staticText_CreateFcn(hObject, eventdata, handles)
% hObject    handle to abba_staticText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% --------------------------------------------------------------------
function Untitled_1_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
antm = get(handles.week_editText,'String');
antm1=str2num(antm);
antm2=antm1+1;
if antm2>52
    antm2=antm2-52;
end
set(handles.week_editText,'String',num2str(antm2))


% --- Executes on button press in pushbutton11.
function pushbutton11_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
antm = get(handles.week_editText,'String');
antm1=str2num(antm);
antm2=antm1-1;
if antm2<1
    antm2=antm2+52;;
end
set(handles.week_editText,'String',num2str(antm2))

% --- Executes on button press in pushbutton12.
function pushbutton12_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
bntm = get(handles.day_editText,'String');
bntm1=str2num(bntm);
bntm2=bntm1+1;
if bntm2>7
    bntm2=bntm2-7;;
end
set(handles.day_editText,'String',num2str(bntm2))

% --- Executes on button press in pushbutton13.
function pushbutton13_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
bntm = get(handles.day_editText,'String');
bntm1=str2num(bntm);
bntm2=bntm1-1;
if bntm2<1
    bntm2=bntm2+7;
end
set(handles.day_editText,'String',num2str(bntm2))

% --- Executes on button press in pushbutton14.
function pushbutton14_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
cntm = get(handles.hour_editText,'String');
cntm1=str2num(cntm);
cntm2=cntm1+1;
if cntm2>23
    cntm2=cntm2-24;
end
set(handles.hour_editText,'String',num2str(cntm2))

% --- Executes on button press in pushbutton15.
function pushbutton15_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
cntm = get(handles.hour_editText,'String');
cntm1=str2num(cntm);
cntm2=cntm1-1;
if cntm2<0
    cntm2=cntm2+24;
end
set(handles.hour_editText,'String',num2str(cntm2))

% --- Executes during object creation, after setting all properties.
function uipanel3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to uipanel3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```