The Application of Fuzzy Decision Tree for Voltage Collapse Analysis

by

Haji Izham Haji Zainal Abidin

A thesis submitted to the university of Strathclyde
In fulfilment for the degree of
Doctor of Philosophy

Department of Electronic and Electrical Engineering
University of Strathclyde
2002

Copyright Statement

Acknowledgement

I would like to express my utmost gratitude to Professor Kwok Lun Lo, head of the Power System Research Group, University of Strathclyde for his supervision and valuable guidance and patience throughout the duration of this research.

My appreciation also goes to my sponsor, Tenaga Nasional Berhad, Malaysia for their full financial support and help throughout this research.

To the Power System Research Group (PSRG), thank you for sharing your knowledge and experiences with me, hopefully I have done the same for all of you.

To my overseas colleagues, Carlos Lozano in Columbia and Dr Ashraf Abdelaal in Egypt, thank you for all your advice and 'interesting' discussion sessions which helped me a lot through out this research.

To Nor Anilah Mohamad, Hayati Abdullah, Ahmad Nizam Hassan, Amir Hisham Hashim, Faridah Hussin, Zuhaina, Yusri, Tuan Rashid and all my friends here in Glasgow, thank you for giving the much needed 'push' and encouragement. I pray for your future success. To all my friends that I have met throughout my stay in the United Kingdom thank you for being there for me and especially for being my friend.

Finally, my unlimited and sincerest gratitude to my parents Datin Hjh Esah and Datuk Haji Zainal Abidin, my sisters, Dr Nurhayati and Nurdiana, my brother Dr Imran for all their support, unconditional love and encouragement throughout my life.

Haji Izham Haji Zainal Abidin
23 April 2002

Abstract

In the time of rapid growth, there is an increase of demand for a reliable and stable power supply. Due to this, utility companies are forced to operate their power system nearer to its maximum capabilities since system expansion may be a costly option. As a result, the power system will be at risk to voltage collapse. Voltage collapse phenomenon is known to be complex and localised in nature but with a widespread effect. The ultimate effect of voltage collapse would be total system collapse which would incur high losses to utility companies.

This thesis discusses the voltage collapse phenomenon, its causes, effects and its analytical tools. Looking into its analytical tools, it is observed that it relies upon system equations and models. Published results from these techniques are accurate but may require long computation time for a big and complex system.

As a possible solution, this thesis looks into combining machine learning techniques with fuzzy logic in creating a fuzzy decision tree (FDT) tool for voltage collapse analysis. The algorithm utilises static power flow solution as data sets in partitioning the power system into strong and weak areas. From several test results and algorithm development, this research concludes with a possible voltage collapse analytical tool using a hybrid FDT approach based upon multiple attribute partitioning. This thesis concludes with discussions on test results highlighting the FDT performance and ends with a discussion on possible future development on the FDT in creating a more complete tool for voltage collapse analysis.

Glossary

LS_{node}	Learning set at the test node
μ	Fuzzy membership value
Attribute	A member of a test node. May be in various forms, e.g. bus voltage magnitude and but voltage angles.
Branch	Connector between nodes.
C4.5	A variant of Decision Tree improved by Quinlan (Ref Sec. 4.3.5)
EMV Tree	Extra Monetary Value Tree. (Ref. Sec. 4.3.1)
Entropy	Defined as the value of impurity in a sample collection
ID3	Induction Biased Decision Tree created by Quinlan (Ref. Sec. 4.3.4)
Information Gain	Also known as information quality assessment. A formulation to compare attribute quality from one set to the other. May be in various forms e.g. entropy before or after partition or direct attribute magnitude comparison e.g. Kolmogorov-Smirnov distance score or Chi Square tests.
Leaf node	The final product of a DT or FDT consisting of partitioned data.
Over partition	An instance when a test set is partitioned too much as such that the resultant nodes has a small population count or an empty resultant node.
Pruning	A DT or FDT stage which collapses over partitioned leaf nodes in creating a more generalised tree
Root	The start of a decision tree consisting of the original data set.
Recursive partitioning	A repetitive partitioning scheme
Test node	A node at which decision tree (DT) or fuzzy decision tree (FDT) formulations are applied.

Table of contents **Page**

Appendices

Chapter 1: Introduction

In the era of fast growth and rapid development there is a demand for a reliable power system network [1]. For a utility company, meeting the ever-increasing demand via network expansion would incur large costs; hence a logical solution is raising the existing system operational level to meet with the demand. In doing so, the system will be exposed to potential risks of system instability. These may range from localised instability event to a system wide collapse due to the reduction of system resource reserves [2]. A common term given for this type of system wide collapse is known as voltage collapse.

1.1 Voltage collapse and its effects

Voltage collapse is known to be complex in nature. The ultimate result of a voltage collapse event is total system shutdown which would lead to huge losses in revenue to the utility company. Due of this, efforts has been made over the years in order to understand and identify its cause and effects.

Research work has indicated that voltage collapse is a result from a combination of various power system phenomena, some may act to reinforce others but ultimately becomes a destructive element promoting system instability. In short, voltage collapse may start with a localised event and ultimately causes a system wide collapse. An example of this is the behaviour of load tap changing transformers (LTCs). Its primary function is to stabilise the corresponding bus voltage within a pre-determined standard level for all load conditions. But in some circumstances, the LTC actions may be detrimental as such that its resultant action creates a reactive sink in the system hence increasing the risk of system collapse. A solution to this is to have LTC blocking schemes to mitigate possible voltage collapse instances.

Apart from system component behaviours, the adequacy of system resource is important in maintaining system stability with the ever-increasing demand. Some of these resources have to be strategically located to optimise their effectiveness as supporting components to the system network.

In understanding the system behaviours prior to collapse, various techniques are created which can be in a form of either dynamic or static system analysis. Each has its own advantages and disadvantages. Dynamic analysis involves tracking the system operating point before and after the collapse. It is very useful in understanding system behaviours hence determining possible mitigation solutions [3-8]. Examples of these techniques are bifurcation analysis, eigenvalue analysis and the energy function analysis. All of them involve a lot of mathematical manipulations and rely on the accuracy of the system models. Bifurcation analysis and energy function techniques assume fast system dynamics as quasistatic in nature. Eigenvalue analysis on the other hand looks into the manipulation of the system jacobian. Results using these techniques are very good and have a lot of potential in further understanding system behaviours. But these techniques have the disadvantage of high computational time and hence not an ideal tool for online assessments.

Static analysis on the other hand is considered to be relatively simpler and has a lot of online potential. Static analysis may range from voltage collapse proximity indices to network partitioning techniques. All of them have the same objective of identifying voltage collapse and its origin before it occurs. Although static analysis is relatively simple compared to dynamic analysis, but some still rely on the manipulation of system equations that may be complex and time consuming for a big and meshed system. Some research may offer a faster way of analysis via the modification of the system jacobian and sparse matrix manipulation techniques but there are also other possible alternatives that may give an added advantage in terms of speed of calculation.

1.2 Artificial intelligence and fuzzy logic

The arrival of artificial intelligence (AI) techniques has led to various improvements in the study of voltage collapse. AI techniques offer an alternative approach to conventional methods by utilising statistical or probabilistic approaches. Examples of AI techniques are neural networks (NN), genetic algorithm and also machine learning techniques. Published results show a promising future for these techniques as a tool for voltage collapse analysis [9-13].

Combining fuzzy logic into AI techniques gives an added edge in terms of its speed and versatility. This is due to its capability in quantifying crisp data values into a range of normalised fuzzy membership. Examples of fuzzy-AI techniques range from neurofuzzy techniques to fuzzy decision tree methods. Research publications show that a lot of efforts have been given into utilising neurofuzzy approaches as a voltage collapse tools [10, 14].

Neural based methods have adaptive cognitive advantages via training simulations but they suffer from uncertainties for unforeseen events and also dimensionality problems. Some may argue that these problems are not critical as mentioned in [9] but a form of alternative has to be looked into. This leads to the development of decision tree based approaches. Early efforts of these techniques can be seen in [10, 15-20].

1.3 Decision tree techniques and its evolution

Originating from machine learning techniques, decision tree techniques (DT) have been a well-established method for data mining purposes. Its application varies from business decision tool, medical diagnostic systems, image processing algorithm to other engineering applications [21-31]. Generally similar to NN techniques, DT techniques involve plotting a decision path using constraints given in the problem scenario. Dependent upon its scope of application, there are various types of DT techniques available all of them have the same objective as a decision making tool. The difference between DT and NN based techniques is in terms of the decision processing. NN is based upon synapse connections between nodes in the NN model where the user has no control on how the synapse connections are made. In DT based techniques the user has more control towards the process of building the DT via formulation and rules. Therefore, DT techniques are known to be more predictable as compared to NN based techniques for unseen instances. Utilising fuzzy logic into DT techniques improves its performance in terms of its adaptability and compatibility in handling continuous data sets and fuzzy decisions. Examples of these techniques can be found in [24-26, 32-36].

1.4 FDT applications towards voltage collapse analysis

Using the basic understanding of voltage collapse and decision tree techniques, a new approach of analysing voltage collapse is proposed utilising the partitioning capabilities of fuzzy decision tree methods. Relatively different in terms of approach compared to conventional techniques, the proposed FDT method does not require the manipulation of the system jacobian or power flow equations. Instead it utilises standard power flow solutions in partitioning portions of the power system into strong and weak regions. Since voltage collapse are known to start from a localised event or a collection of localised events, hence the proposed FDT technique may be beneficial in identifying the location of these events in order to prevent voltage collapse from occurring. Adapting and improving these techniques show promising results as an alternative approach with possible future improvements for creating a more effective tool for voltage collapse analysis.

1.5 Original Contributions

The original contributions to this thesis are as follows:

a) The introduction of an FDT tool as a voltage collapse analysis tool using observable data obtained from standard load flow simulations

b) The developed FDT goes through an evolution and improvements in terms of algorithm structure and flow using logical and constructive approaches based upon preliminary and also main test results.

c) The introduction of multiple attributes scheme into the FDT, which would further improve the overall FDT performance.

d) Introducing a new form of dynamic decision tree technique that does not require much training time and improved adaptability for further improvements.

1.6 Thesis layout

This thesis has 6 chapters with a series of appendices at the end. Each chapter represents information required in understanding and building the proposed algorithm.

Chapter 1 outlines the general overview of the whole thesis that consists of a general overview of voltage collapse, its symptoms, its analytical tools and the proposed algorithm.

Chapter 2 goes through the theoretical background of voltage collapse highlighting the symptoms and its effects to voltage collapse.

Chapter 3 discuss several conventional and other researched methods in analysing voltage collapse In general, these analytical methods may be in the form of dynamic and static analysis. This chapter also highlights the merits and problems of each method.

Chapter 4 discuss the basic concepts of machine learning and decision tree. Various examples and simple test results are utilised to explain the formulation used in several types of decision trees. There is also a discussion on several types of decision trees and also fuzzy decision trees (FDT).

Chapter 5 goes through the proposed algorithm. It is based upon the FDT concepts presented in Chapter 4, voltage collapse concepts and its analytical tools presented in Chapter 2 and 3. The proposed algorithm is developed in terms of its program structure and FDT component stages with the objective of improving the overall FDT performance. The programming language used to create the algorithm is Borland C++ ver 4.5.

Chapter 6 concludes the test results, highlighting improvements and problems found throughout the FDT development. This chapter ends with a discussion of possible future work in developing further the proposed algorithm.

The appendices at the end of this thesis contain preliminary test results and program structures related to the development of the proposed algorithm.

1.7 References

1. Ali, Datuk Dr Ahmad Tajuddin. *Development of ASEAN Power Grid.* in *1st ASEAN Energy Business Forum.* 1997. Kuala Lumpur, Malaysia.
2. Taib, Masri, *Possibility of Voltage Collapse in the TNB Power System.* Network Planning Department, Tenaga Nasional Berhad, Malaysia, Internal Report. April 1997, 1997, 18 pages.
3. M Begaovic, J Bright, T Domin, S. Easterday-McPadden, A Girgis, W Hartmann, C Henville, M Ibrahim, K Kozminski, R Marttila, G Michel, K Mustaphi, D Novosel, B Pettigrew, M Sachdev, H Shuh, P Solanics, J Williams, *Voltage Collapse Mitigation.* IEEE Power System Relaying Committee. December, 1996, 33 pages.
4. Adriana L Tognete, Carlos A Castro, *An efficient sensitivity analysis based method for calculating load margins to voltage collapse.* Revista Controle & Automacao (Control and Automation), 1999. **10**(1), p. 73 - 83.
5. Carreras B A, Newman D E, Dobson I, Poole A B. *Initial evidence of self-organized criticality in electric power system blackouts.* in *Proceedings of Hawaii International Conference on System Sciences.* 2000. Maui, Hawaii: IEEE.
6. Dobson, I, Carreras B A, Lynch V E, Newman D E. *An initial model for complex dynamics in electrical power system blackouts.* in *Proceedings of Hawaii International Conference on System Science.* 2001. Maui, Hawaii: IEEE.
7. Carreras B A, Newman D E, Dobson I, Lynch V E, Sachtjen M L. *Modelling blackout dynamics in power transmission networks with simple structure.* in *Hawaii International Conference on System Sciences.* 2001. Maui, Hawaii: IEEE.
8. Scott Greene, Ian Dobson, Fernado L Alvarado, *Contingency ranking for voltage collapse via sensitivities from a single nose curve.* IEEE Transactions on Power Systems, 1999. **14**(1), p. 232-240.

9. Hasan, Usama, *Artificial Neural Networks for Voltage Collapse Monitoring.* University of London, Imperial College of Science, Technology and Medicine. 1998. PhD thesis, 181 pages.

10. Louis Wehenkel, *Automatic Learning Techniques in Power Systems.* The Kluwer International Series in Engineering and Computer Science. Vol. 429. 1997: Kluwer Academic Publishers, 279 pages. ISBN 0-7923-8068-1.

11. K Yabe, J Koda, K Yoshida, K H Chiang, P S Khedkar, D J Leonard, N W Miller, *Conceptual designs of AI based systems for local prediction of voltage collapse.* IEEE Transactions on Power Systems, 1996. **11**(1), p. 137 - 145.

12. K L Lo, L K Peng. *Design of Artificial Neural Networks for On-line static Assessment problems.* in *IEE International Conference on Advance in Power Systems Control, Operation and Management.* 1997. Hong Kong.

13. A.P Alves da Silva, C Ferreira, AC Zambroni de Souza, G Lambert-Torres, *A new constructive ANN and its application to electrical load representation.* IEEE Transactions on Power Systems, 1997. **12**(4), p. 1569 - 1575.

14. Christina Olaru, Louis Wehenkel, *On Neurofuzzy and fuzzy decision trees approaches*, in *Information, Uncertainty and Fusion*, Editor Ronald R. Yager Bernadette Bouchon-Meunier, A. Zadeh. 2000, Kluwer Academic Publishers: Boston, p. 131-145.

15. Xavier P Boyen, *Design of Fuzzy Logic Based Decision Trees Applied to Power System Transient Stability Assessment.* Department of Electrical and Computer Science. University of Liege, Belgium. 1995. Masters Thesis, 133 pages.

16. Xavier P Boyen, Louis Wehenkel. *Automatic Induction of Continuous Decision Trees.* in *Proceedings of Information Processing and Management of Uncertainty in Knowledge-Based Systems.* 1996. Granada, p. 419 - 424.

17. Xavier P Boyen, Louis Wehenkel. *Fuzzy Decision Tree Induction for Power System Security Assessment.* in *IFAC Symposium on Control of Power Plants and Power Systems.* 1995. Cancun, Mexico, p. 151-156.

18. Xavier P Boyen, Louis Wehenkel, *Automatic Induction of Fuzzy Decision Trees and its Applications to Power System Security Assessment.* International Journal of Soft Computing and Intelligence, 1999. **102**, p. 3 - 19.

19. Louis Wehenkel, C Lebrevelec, M Trotignon, J Batut. *A probabilistic approach to the design of Power System protection schemes against blackouts.* in *IFAC-Cigre Symposium on Control of Power Systems and Power Plants.* 1997. Beijing, p. 506 - 511.

20. Louis Wehenkel, I Houben, M Pavella, L Riverin, G Versailles. *Automatic Learning Approaches for On-line transient stability preventive control of the Hydro-Quebec System. Part 1: Decision Tree Approaches.* in *IFAC Symposium on Control of Power Plants and Power Systems.* 1995. Cancun, Mexico, p. 231-236.

21. Engineering_Statistics_Handbook, *Quantitative Techniques.* Internet Page. 2002. http://www.itl.nist.gov/div898/handbook/eda/section3.

22. Evans, David H, *Chapter 12: Decision Trees,* in *Probability and its Applications for Engineers.* 1992, ASWC Quality Press: New York, p. 555-585.

23. Irwin Miller, John E Freund, *Probability and Statistics for Engineers.* 3rd Edition ed. 1985: Prentice Hall International Inc. ISBN 0-13-711912-7.

24. Isao Hayashi, Toshiyuki Maeda, Motohide Umano, Lakhmi C Jain, *A Formulation of Learning Type Fuzzy ID3.* International Journal of IEEE Transactions on Fuzzy System, 2000.

25. Janikow, Cezary Z, *Examplar Learning in Fuzzy Decision Trees.* Proceedings of FUZZ-IEEE, 1996, p. 1500-1505.

26. Janikow, Cezary Z, *Fuzzy Decision Trees: Issues and Methods.* IEEE Transactions on Systems, Man and Cybernetics, 1998. **28**(1), p. 1 - 14.

27. Mitchell, Tom M, *Machine Learning*: McGraw-Hill Publishers Inc. ISBN 0070428077.

28. Quinlan, J R, *Improved used of Continuous Attributes in C4.5.* Journal of Artificial Intelligence Research, 1996. **4**, p. 77 - 90.

29. Slavco Velickov, Dimitri Solomatine. *Predictive Data Mining: Practical Examples.* in *Artificial Intelligence in Civil Engineering. Proceedings of 2nd Joint Workshop.* 2000. Cottbus, Germany, p. 3-19.

30. Vanguard, *Decision Tree Analysis.* Vanguard Software Corporation. Internet Article. 2002. http://www.vanguardsw.com/dphelp/dph00097.htm.

31. Friedman, Jerome H, *A Recursive Partitioning Decision Rule for Nonparametric Classification.* IEEE Transactions on Computers, 1977, p. 404 - 408.

32. K L Lo, Q Y Zheng, T X Zhu, S K Tso, *Fuzzy-set approach to dynamic voltage security assessment.* IEE Proceedings of Generation, Transmission and Distribution, 1995. **142**(2), p. 190 - 194.

33. K L Lo, H I Haji Zainal. *The Fuzzy Decision Tree Application to a Voltage Collapse Problem.* in *University Power Engineering Conference (UPEC).* 2001. University of Swansea, Wales.

34. K L Lo, Q Y Zheng, T X Zhu, S K Tso, *Fuzzy Reasoning for knowledge-based assessment of dynamic voltage security.* IEE Proceedings of Generation, Transmission and Distribution, 1996. **143**(No 2), p. 157 - 162.

35. K L Lo, Y J Lin, W H Siew, *Fuzzy-logic method for adjustment of variable parameters in load-flow calculation.* IEE Proceedings of Generation, Transmission and Distribution, 1999. **146**(3), p. 276-828.

36. Hussein Almuallim, Shigeo Kaneda, Yasuhiro Akiba, *Development and Applications of Decision Trees*, in *Expert Systems*, Editor Cornelius T Leondes. August 2001, Academic Press.

Chapter 2: The voltage collapse phenomenon

2.1 Introduction

The stability of a power system network is the main concern of every electrical utility company. The loss of stability in a power system network would mean power disruption on the customer side and hence a loss of income of the utility. Stability as aforementioned can be in various forms. But for the purpose of this research, more attention is given to the voltage collapse phenomenon. Voltage collapse is a critical problem in a power system network where its overall effect would lead to total system collapse. With the consequence of total system shut down means that the main generation source (e.g. coal fired power station) has to be taken off-line. This in effect will require long period of system restoration resulting in huge financial losses for the utility company. Several research has been made in analysing its origins, cause and effect [1-6]. This chapter will go through several explanations on voltage collapse and its mechanism from various points of view, e.g. transmission line, generator and load characteristic. Each has its own contributing role towards voltage collapse and they can either be the sole or one of the combinatory factors towards voltage collapse thus making this phenomenon very complex. As a result, a range of approach and methodologies has been created with the hope of understanding further the nature of voltage collapse. These approaches are mainly mathematical in nature. Before going through the mathematics of voltage collapse, a thorough understanding of the very nature of voltage collapse must be laid out.

2.2 Voltage Collapse definition

There are several definitions of voltage collapse in the literature [1, 7-11], in general all definition relates to the same basic understanding of voltage collapse which is loss of system stability. Voltage collapse corresponds to a situation when the system is unable to maintain a stable operating point due to constraints applied to the power system. These constraints may include several aspects:

2.2.1 Transmission System Aspect

In modern times, the power system grid is loaded or operated at maximum potential. This is done to maximise income with less expansion of the grid. The effect of this is restricted operational flexibility, i.e. the system operating point is driven nearer to the maximum system operation limit [4]. For example, a large power system network may consist of several clusters of generators spread across long distant transmission lines and support devices to supply power to customers. The nature of consumer demand looked in terms of type of load may consist of either pure active loads (e.g. home users) or it may be a combination of active and reactive power consumptions (e.g. industries). Simplifying the explanation, the power system may be reduced into 1 generator connected to a load bus with one load as shown in the figure below:

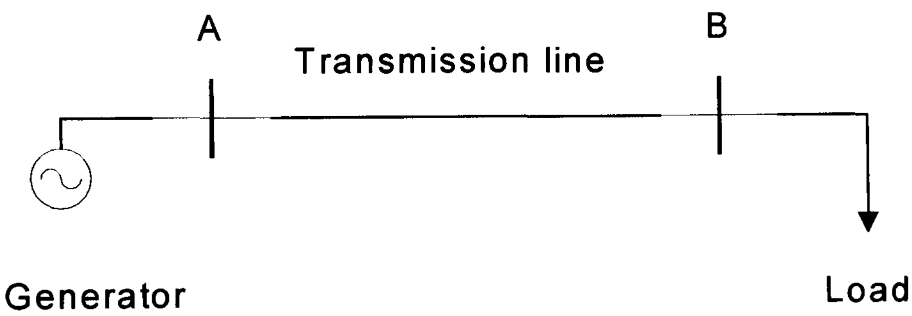


Figure 2.1: 1 Generator, 1 Load connected together via a load bus

In effect, all three elements shown in figure 2.1 (generator, transmission line and load) can be considered to represent the total equivalent of each element. From here, several system calculations can be made as shown. Referring to figure 2.1, lets assume the following:

$$\begin{aligned} &Generator = \overline{E_s} = E\angle 0 \\ &Transmission\ Line = Z_{Line}\ \angle\vartheta = R_{Line} + jX_{Line} \\ &Load = Z_{Load}\ \angle\phi = R_{Load} + jX_{Load} \\ &\mathrm{Re}ceiving\ end\ power = P + jQ \end{aligned} \qquad (2.0)$$

Using equations shown in Eq. 2.0, the maximum power can be derived given by the following:

Since transmission resistance, R is small, hence the load voltage V is denoted as:

$$\overline{V}_{Load} = \overline{E} - jX_{Line}I \tag{2.1}$$

Therefore the complex load power S is given by:

$$\begin{aligned} S_{Load} &= P + jQ \\ &= \overline{V}_{Load}\overline{I}^* \\ &= \overline{V}_{Load}\frac{\overline{E}^* - \overline{V}_{Load}^*}{-jX_{Line}} \\ &= \frac{j}{X_{Line}}\left(EV_{Load}\cos\phi + jEV_{Load}\sin\phi - V^2\right) \end{aligned} \tag{2.2}$$

Breaking Eq. 2.2 into real and imaginary parts, hence

$$P_{Load} = \frac{EV_{Load}}{X_{Line}}\sin\phi \tag{2.3}$$

$$Q_{Load} = -\frac{V^2{}_{Load}}{X_{Line}} + \frac{EV_{Load}}{X_{Line}}\cos\phi \tag{2.4}$$

Next, elimination of ϕ is yields the following steps:

$$\textit{Since } \sin^2\phi + \cos^2\phi = 1$$
$$\textit{Taking } (2.3) \textit{ and } (2.4) \textit{ yields}:$$
$$\sin^2\phi = \frac{X^2{}_{Line}P^2{}_{Load}}{E^2V^2{}_{Load}} \textit{ and } \cos^2\phi = \frac{\left(Q_{Load}X_{Line} - V^2{}_{Load}\right)^2}{E^2V^2{}_{Load}}$$
$$\therefore \sin^2\phi + \cos^2\phi = \frac{X^2{}_{Line}P^2{}_{Load}}{E^2V^2{}_{Load}} + \frac{\left(Q_{Load}X_{Line} - V^2{}_{Load}\right)^2}{E^2V^2{}_{Load}} = 1$$
$$\frac{X^2{}_{Line}P^2{}_{Load} + \left(Q_{Load}X_{Line} - V^2{}_{Load}\right)^2}{E^2V^2{}_{Load}} = 1$$
$$X^2{}_{Line}P^2{}_{Load} + \left(Q_{Load}X_{Line} - V^2{}_{Load}\right)^2 = E^2V^2{}_{Load}$$
$$\therefore \left(V^2{}_{Load}\right)^2 + \left(2Q_{load}X_{Line} - E^2\right)V^2{}_{Load} + X^2{}_{Line}\left(P^2{}_{Load} + Q^2{}_{Load}\right) = 0 \qquad (2.5)$$

Eq. 2.5 can be solved by deriving its roots with respect to V^2 that will yield the maximum power transferable by the transmission line, which as shown:

Conditions for Eq. 2.5 to have real roots are:

$$\left(2Q_{Load}X_{Line} - E^2\right)^2 - 4X^2{}_{Line}\left(P^2{}_{Load} + Q^2{}_{Load}\right) \geq 0$$
$$\textit{Simplifying gives}:$$
$$-P^2{}_{Load} - \frac{E^2}{X_{Line}}Q_{Load} + \left(\frac{E^2}{2X_{Line}}\right)^2 \geq 0 \qquad (2.6)$$

Taking Eq. 2.6 into account, setting P = 0 yields the maximum Q received by the load. On the other hand, setting Q = 0 yields the power transfer possible at unity power factor. The resulting equations for both scenarios are as shown below:

P = 0 yields:

$$Q_{Load} \leq \frac{E^2}{4X_{Line}} \qquad (2.7)$$

Q = 0 yields:

$$P_{Load} \leq \frac{E^2}{2X_{Line}} \qquad (2.8)$$

Eq. 2.7 and 2.8 shows that P and Q are dependent upon generated voltage E and line reactance X. The main point here is the X element in the formulation which gives an indication of the importance of reactance in maintaining P and Q in the power system. This will be explained further in this chapter.

Power Voltage Relationship

Referring to Eq. 2.5, we can derive the roots for V as shown in Eq. 2.9. This gives a power-voltage relationship where the voltage level V is dependent upon P and Q.

$$V = \sqrt{\frac{E^2}{2} - QX \pm \sqrt{\frac{E^4}{2} - X^2P^2 - XE^2Q}} \qquad (2.9)$$

Using Eq. 2.9 and using different values of V and P, the VP curve can be plotted as shown in figure 2.2(a).

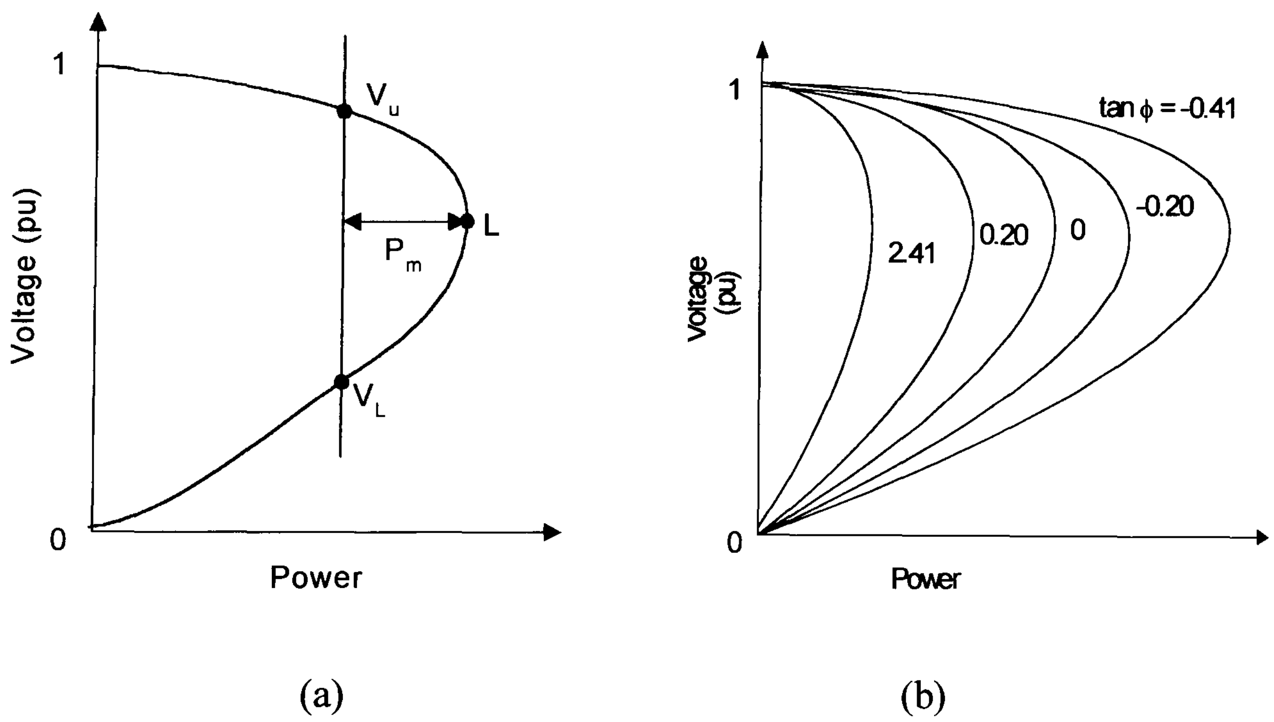


Figure 2.2: (a) VP Curve (b) VP curve with varying Power Factor

Figure 2.2(a), is an example of a VP curve. A VP curve is a range of system voltage level plotted with reference to a varying system load, P. The VP curve shows 2 possible operating points at any one instance (V_L and V_u) V_u is denoted as the practical operating point and V_L is the mathematical operating point, which does not exist in practice. P_m denotes the reserve real power available to support any increase of demand. As the operating system is pushed higher towards the limit (L), a reduction of reserve power P_m will lessen the system's ability to support more demand. The reduction of voltage level in the load bus scenario, an assumption is made that there are no other reactive resources available to maintain a constant voltage level.

The VP curve does not show the effect of reactive power scenario. However, the general characteristics of the VP curve changes as we vary the power factor of the system as shown in figure 2.2(b). Apart from that we can extend the VP curve to include the reactive component as shown in figure 2.3.

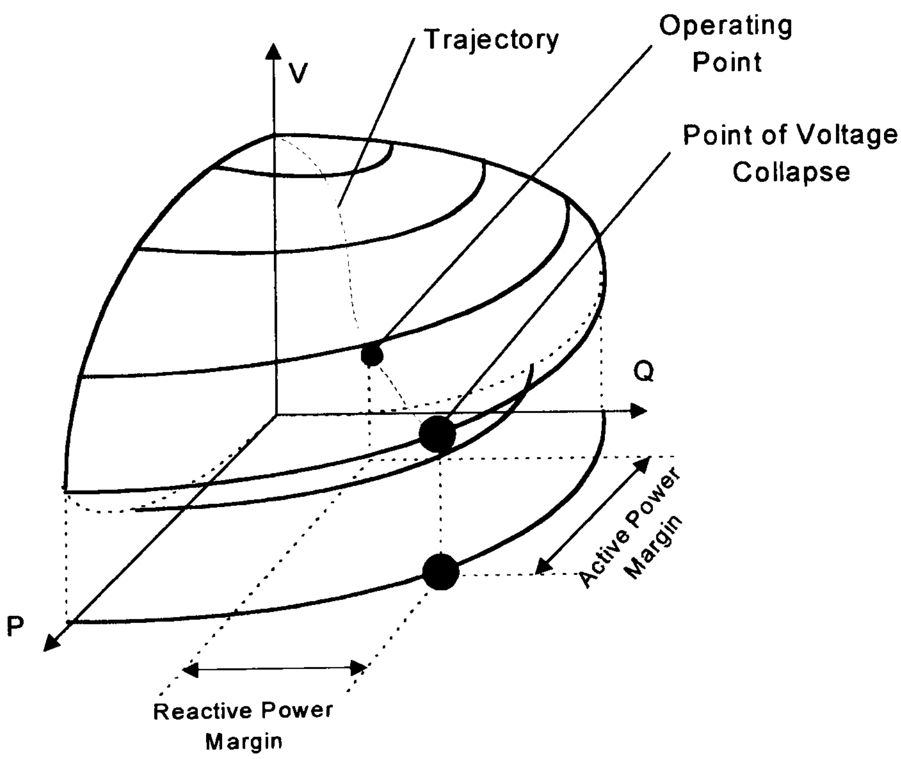


Figure 2.3: VPQ Curve

Figure 2.3 shows that the voltage stability boundary is denoted by the projection on the PQ axis. From here it is shown that as the system is loaded more towards the limit, the reactive and active power margin will shrink and hence reducing the ability of the system to sustain the required voltage and power levels.

Now, consider 2 scenarios related to voltage collapse, which would lead to voltage instability:

(a) A slow increase in system demand

A slow load increase corresponds to long-term constant load increase. It could either be seen as an actual slow load increase or a slow cascading loss of generation. From either point of view it shows an overall increase of load over time. The time constraint related to this may vary from a few minutes to several hours. Using a VP curve, the system operating point behaviour will shift as shown in figure 2.4:

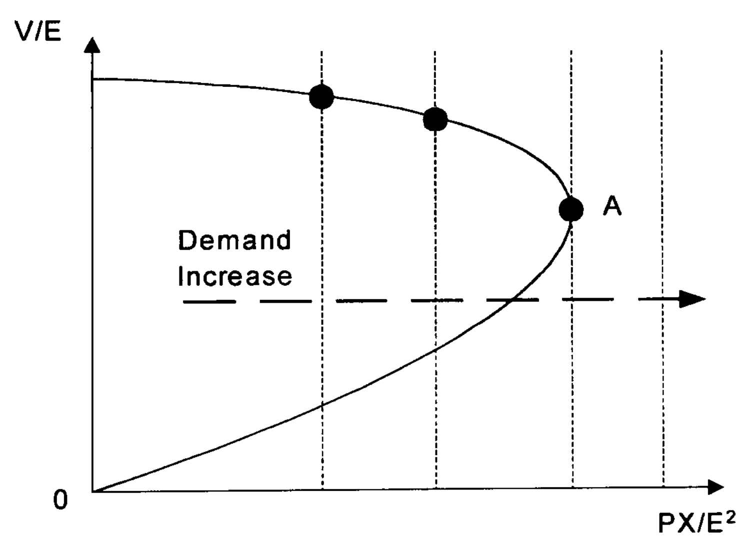


Figure 2.4: VP curve showing the equilibrium point as system demand is increased.

Referring to figure 2.4 and Eq. 2.9, as the system demand, P increases the equilibrium point is slowly shifted towards point A where point A denotes the maximum loadability point. Any further increment will render the equilibrium point to be outside the system characteristics hence the system will collapse.

(b) There is a large disturbance or sudden loss of generation.

This relates to transient fault scenarios leading to a bulk generation loss. The incidence time frame is very fast which is likely to be in milliseconds. During this short time frame, maintaining system stability will depend upon protection schemes employed to protect other system areas from going into instability. The loss of generation resources could be in the form of protection equipment action, major generation outage or tripping of transmission lines. Since the disturbance is fast and corrective actions did not have sufficient time to correct the problem (e.g. operator reaction and backup generation takes time to go online even from spinning reserve), the projected network equilibrium point is likely to be outside the stable. This in effect will create a reactive sink at the affected areas as the system tries to converge to a stable operating point by demanding more reactive power to maintain the increased power transfer. At this point there are several possibilities that can be considered.

- Reactive power reserves goes into action to compensate for the increase of reactive power demand.
- Tap changing transformer goes into action to maintain voltage level.
- AGC goes into action to compensate for the sudden increased demand of P and Q from the system.
- Protection scheme goes into action to protect generators or isolate the fault. In some power networks, islanding schemes are utilised.

All these actions may even restore the system to a new stable operating point or in some cases; it may have a detrimental effect to system stability. If the system is stabilised, the operating point may already be at the maximum loadability point of the system. If this is the case, any further changes to the system load profile demand and generation may put the system at risk to voltage collapse. If the system is not stabilised, i.e. the operating point goes outside the VP curve, the system is considered to have collapsed.

Both of the scenarios mentioned may contribute to voltage collapse. Although the first scenario is a slow process compared to the latter, but there are situations where there is a combination of the two scenarios where the system goes through a slow load increase followed by a fast transient disturbance that ultimately leads to voltage collapse. Since the scenarios were explained in a general term, a more detailed discussion into power system properties to understand other mechanisms or actions that may lead to voltage collapse will follow in subsequent sections.

Heavy reactive power flows

This corresponds to situations when there is a reactive sink in the system, which consumes large amount of reactive power. The emergence of sinks in the system is dependent upon the system operation state. If the system is already highly loaded, sudden increase of load requiring heavy reactive power injections to maintain the voltage levels may create sudden in rush of reactive power flow from nearby sources to compensate for the reactive demand. The amount of reactive support flowing to these sinks is dependent upon the availability of the local reactive support and line limits for reactive power transfer. If it is not adequate, the system operating point will slowly drift towards instability. Identifying these sinks can be made by recursive system simulations taking into account various system loading scenarios. Solving these problems requires strategic placement of reactive resources to high security load areas[1] and will affect grid design. Although strategic placement of reactive resources may be a sound investment, but future planning of cable and transmission line limits are also important, since heavy reactive flow are always limited by line and thermal limits.

[1] High security areas are areas at which constant uninterrupted power supply is important. E.g. Military installations, key load centres and high demand areas.

Inadequate Active/Reactive Power Supports

This corresponds to generator and other support reactive resources reaching their respective limits. From the generation point of view, voltage stability relates to the ability of the generator in maintaining a constant voltage level by varying the generated active and reactive power. If these limits are reached, the generator will not be able to cope with further increase of demand hence any load increase will aggravate further system stability.

From the support resources point of view, there are various types of resources available e.g. capacitor banks, LTC transformers dynamic and static compensators. These compensators may help in terms of satisfying the increased support requirements, but in certain conditions, it may be disastrous such as the operating characteristics of LTC transformers where they could be detrimental to the system stability during operational conditions.

Grid Design

This is more towards the unforeseen instances during the planning stage of the power system. Due to economical reasons, it is possible that the utility company builds a cluster of generators in one area of the power system, with limited support lines between generators and load points. Doing this, the system will be at risk of instability when there are possible line outages from the generator clusters during the crucial high load period. Although implementing passive support can be another viable planning and design solution, but it does have the disadvantage of high voltage support requirements, hence a possible implementation of active reactive support is recommended [11]. In short, the general solution to this is to do repeated simulations of the system and reinforcing the system design in terms of secondary lines and placement of reactive supports throughout the system [10, 11].

Line Limits

Transmission lines have limits due to the design, material strength and resistance. In practice, highly loaded lines are considered to be a risk to system stability. Referring to Eq. 2.8, it is shown that for a lossless system, i.e. active power, P is limited to the line reactance X and its total generation capacity. The same agreement goes for reactive power Q as shown in Eq. 2.7. Conditions when the system is assumed to be lossless is an ideal case. For a more realistic representation of line reactance, other reactance and resistive sources have to be taken into account.

Consider the case when there is a fault in the network which results in line outages. This will cause other local lines to be loaded more than their nominal operating levels. If these line limits have been reached in the respective lines and there are no other support lines available, any further increase of load onto the line will overload them and would have a detrimental effect to system stability.

In terms of grid design, it may be strategic for the utility company to install transmission lines in accordance to its projected demand, and hence the type of transmission lines and their configurations are fixed. A design solution to this is to have support lines to distribute the transmitted load. Due to economical reasons and cost effectiveness, only high security transmission lines have support lines.

The overall security of the system can then be determined by exhaustive system stability analysis using various possible scenarios. In theory this should be adequate in terms of system security tests, but the possibilities can be so large that not all of them can be tested. This has been the case in a 'near miss' voltage collapse event in Malaysia [10].

Tap Changing Transformer Effect

On load tap changing transformers (OLTC) are devices used to maintain a constant voltage level on the low voltage side. The tap of the transformer changes automatically with reference to the voltage level on the high voltage side. Considering that the system would have a significant reactive and active power reserve, the OLTC is useful in giving the customer a good, reliable and constant voltage level.

However, some actions of the OLTC can have detrimental effects and can be considered as one of the causes of voltage collapse. Research yielded possible scenarios where OLTC actions may play a significant role in pushing the system to voltage collapse [7, 12]. As the system is pushed towards its maximum operating point, the system will experience a decrease of voltage on the transmission side. This in effect will cause the OLTC to operate further in maintaining the voltage level on the low voltage side. This action however requires more reactive power to support the voltage level. If the reactive power is not adequate, it will push further the system operating point towards its maximum loadability. This in turn will effect the voltage level on the secondary side of the transformer which in turn will cause the OLTC to operate again. The actions of the OLTC will slowly push the system to collapse. Generally it can be concluded that in situations where there are a deficit of reactive and active power support, the effect of the OLTC action may push the system operating point towards voltage instability. Ideas in mitigating this effect has been suggested by blocking OLTC tap action during voltage instability occurrence [12].

Unscheduled line outage, generator outage and sudden load demand increase

Line and generator outage can be related to equipment malfunctions and generator faults. Line outage and generator outage can be treated as a sudden relative load increase to the system. The increase of demand may push the operating point to the maximum operating level. Usually, the AGC (Automatic Governor Control) will react by increasing the generator output. Unfortunately the AGC is only suitable for small changes of load profile. If the load change is great, the system operator will instruct generators on spinning reserve to become active. If the spinning reserves were adequate at that time, the system would maintain stability. But to a fault at key points in the system, there is not enough time to start supporting generators, the system will then be at risk to collapse. An alternative solution to this is to use load shedding schemes to free resources in the system.

2.2.2 Generation aspect

The amount of generated power is dependent upon the relative demand of the system. In practical situations, generator scheduling is determined by the system operator based on load forecasting. Each generator has its generated active power and also reactive power capabilities. This is denoted by the generator's capability curves as shown below:

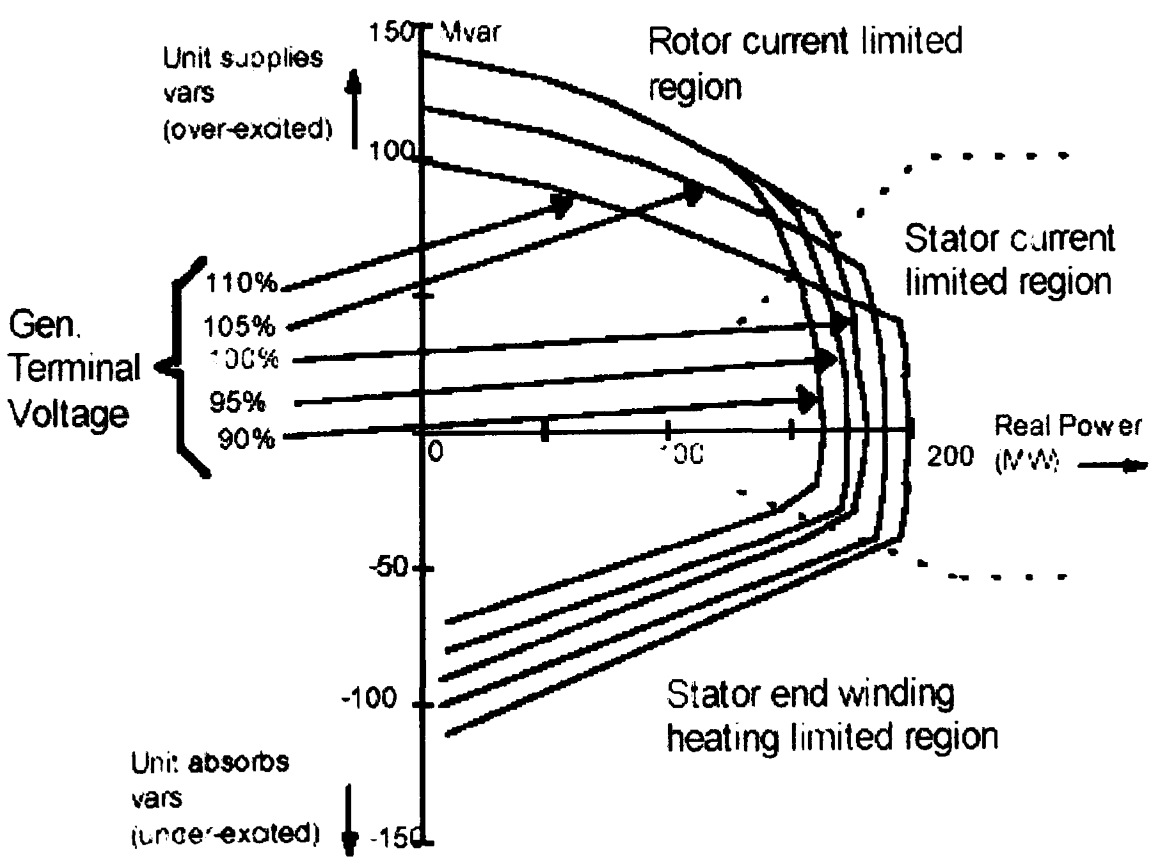


Figure 2.5: A 165 MW generator capability curve with varying terminal voltage

Figure 2.5 shows that by varying the terminal voltage it will vary the generator reactive support capabilities. Generator capability curves may vary in terms of its design and generator cooling system. An efficient cooling system will increase the generator's operating capabilities. This is due to the fact that the field windings are cooled sufficiently hence increasing the rotor field current capacity at higher operating level without compromising the safety of the field windings [12].

A highly loaded generator will have less free reactive resource capabilities as shown in figure 2.5, therefore it is normal that the system operator might not to load the generator too heavily as to increase its reactive support capabilities.

From the utility point of view, increasing the stability of the power system may involve strategic placement of generators, but this is limited by economic and governmental policies. Hence it is important that strategic generator scheduling are made to ensure adequate reactive resources are available if the need arises.

2.2.3 Load aspect

This deals with the types of loads connected to the power system and also their characteristics and effects to voltage stability. Although load effects have been discussed in a different manner beforehand, but there are still points about load that have yet to be discussed. In general, a fully loaded system will have higher possibilities to voltage instability. But this criterion is not the only one that would precipitate voltage collapse. Looking into it closely, the type of load applied to the system does play a role in making the system more vulnerable to voltage collapse.

One type of such loads is the low power factor and high reactive load. Low power factor loads are related to the requirement of having reactive power to maintain voltage at an acceptable level. In practice, the utility company would specify to industries of the required power factor levels in order to maintain stable operation level with enough excess of reactive power resources for emergency purposes.

Inadequate reactive reserve may jeopardize the security of the power system during the period of voltage instability.

In general, during voltage instability occurrences if all active and reactive support has reached its limit, the final mitigation solution will be load shedding. Load shedding scheme varies between utilities and power system networks all over the world. But all involve disconnecting the less important load in order to alleviate the voltage instability problem and protecting the more crucial ones.

2.3 Conclusion

An understanding of the voltage collapse phenomenon has been established with explanations made from various points of view. From these explanations, an observation can be made in terms of voltage collapse time domain. For operator intervention capabilities, short term or transient changes that would lead to voltage collapse may not be possible and hence a need to rely upon automatic mitigation devices. But for mid-term and long-term voltage collapse incidence, operator intervention is possible. Since load variations scenarios do play a major role in voltage collapse, load incremental scenarios are used as a case study when considering mid-term and long-term voltage collapse incidence. In understanding further the mechanism of voltage collapse, chapter 3 will explain further on past approaches in assessing voltage collapse with their advantages and disadvantages.

2.4 References

1. Kundur, Praba, *Power System Stability & Control*. EPRI Power System Engineering Series. 1994: McGraw Hill Inc.
2. Ian Dobson, Liming Lu, *Voltage Collapse Precipitated by the Immediate Change in Stability when Generator Reactive Power Limits are encountered.* IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications, 1992. **39**(9), p. 762 - 766.
3. Souza, Antonio Carlos Zambroni de, *New Techniques to Efficiently Determine Proximity to Static Voltage Collapse*. University of Waterloo. 1996. PhD Thesis, 133 pages.
4. Canizares, Claudio A, *Voltage Collapse and transient Energy Function Analysis of AC/DC Systems*. University of Wisconsin-Madison. 1991. PhD Thesis, 158 pages.
5. Hawkins, Nigel Trevor, *On-line reactive power management in electrical power systems*. Department of Electrical and Electronic Engineering. University of London, Imperial College of Science, Technology and Medicine. London. 1996. PhD Thesis, 195 pages.
6. Navarro-Perez, Rogelio, *Voltage collapse proximity assessment for the operational planning of power systems*. Department of Electrical Engineering. University of London, Imperial College of Science, Technology and Medicine. London. 1993. PhD Thesis, 207 pages.
7. Ian Dobson, Thierry Van Cutsem, Costas Vournas, Chris DeMarco, Mani Venkatasubramanian, Tom Overbye, Claudio Canizares, *Voltage Stability Assesment, Procedures and Guides: Chapter 2: Basic Theoretical Concept*. IEEE/PES Power System Stability Subcomittee, Special Publication. January, 2001, p. 2-100.
8. Thierry Van Cutsem, Costas Vournas, *Voltage Stability of Electrical Power Systems*. 1998 ed. 1998: Kulwer Academic Publishers, 378 pages.

9. Taylor, Carson W, *Power System Voltage Stability*. Power System Engineering Series. Vol. Electrical Power Research Institute (EPRI). 1994: McGraw-Hill Inc, 273 pages.

10. Hashim, Sabar Md. *Voltage Management in a tightly constrained grid system*. in *Tenaga Nasional Berhad Technical Conference*. 1998. Kuala Lumpur, Malaysia, p. 28 (total pages).

11. Taib, Masri, *Possibility of Voltage Collapse in the TNB Power System*. Network Planning Department, Tenaga Nasional Berhad, Malaysia, Internal Report. April 1997, 1997, 18 pages.

12. M Begaovic, J Bright, T Domin, S. Easterday-McPadden, A Girgis, W Hartmann, C Henville, M Ibrahim, K Kozminski, R Marttila, G Michel, K Mustaphi, D Novosel, B Pettigrew, M Sachdev, H Shuh, P Solanics, J Williams, *Voltage Collapse Mitigation*. IEEE Power System Relaying Committee. December, 1996, 33 pages.

Chapter 3: Voltage Collapse Analysis

3.1 Introduction

After defining voltage collapse and its symptoms, there is a need to look into existing methods of analysis. There are various researched methods available [1-15], and going through each available method is not the main objective of this research. Hence several main and important methods are chosen and explained. These include voltage collapse proximity indicators [7, 10, 15], bifurcation analysis [16], eigenvalue-eigenvector analysis [10], energy function method [16], network partitioning method, weak bus analysis and other approaches [1, 8, 9, 11, 17-20]. The chapter concludes with a general summary of the voltage collapse analysis tools.

3.2 Voltage Collapse Proximity Indicators (VCPI) [7, 10, 15, 21]

There are various methods of voltage collapse proximity indicators. Early proximity indicators are based upon the VP curve concept (refer to figure 2.2a) where it calculates the distance between the operating point and the point of collapse as explained in chapter 2. Research has made numerous developments in refining current and new indices are created all with the same objective of assessment towards voltage collapse e.g. reactive and complex power based VCPI [7], voltage existence index, line real and reactive power loadability index, line loss sensitivity index and eigenvalue indices [10]. These indices will have a range of value to represent the system state ranging from stable to voltage collapse prone state. To explain how voltage proximity works, lets look at one example of a voltage collapse proximity indicator presented by [7].

3.2.1 Reactive and Complex Power Based VCPI Analysis [7]

This is a novel and simplified approach in assessing the severity of the system to voltage collapse. Its basic formulation is as shown below:

$$VCPI_{Qi} = \frac{\Delta Q_i}{\sum_{j \in \Omega_g} \Delta Q_{gj}} \qquad , i \in \Omega_L \tag{3.1}$$

where:

Ω_g – System generator

Ω_L – System load

Basically Eq. 3.1 gives the ratio of changes in reactive power demand at load busses with respect to the change in total generated reactive power from generators. The VCPI index will vary from 0 (for weak busses) to 1 (for strong busses). The VCPI index in a stable system will be close to unity since reactive power transfer from generators are assumed to reach the reactive demand centres without much loss in transmission. Hence the VCPI index is close to 1. When the system is operating at maximum capacity, more reactive power are generated by the generators to compensate with this high operating level (refer to generator capability curves at figure 2.5). This can be represented by a high denominator value in Eq. 3.1. On the other hand, high increments of generated Q levels are represented with only small Q changes at load centres hence making the nominator of Eq. 3.1 to be small. Hence the VCPI index will be close to 0 at weak buses in the system. As explained in chapter 2, a weak bus in the system will generate reactive sinks in the system; hence the system will be prone to voltage collapse. The next VCPI index introduced is based upon the complex power S as shown in the formulation below:

$$VCPI_{Si} = \frac{\Delta S_i}{\sum_{j \in \Omega_g} \Delta S_{gj}} \qquad , i \in \Omega_L \tag{3.2}$$

where:

Ω_g – System generator

Ω_L – System load

Eq. 3.2 shows that the VCPI index for complex power is denoted by the ratio of changes in complex power at load centres and the total changes in generated complex power on the generator side. Again, this particular VCPI index will have a variation of values from 0 to 1. An index of 1 will denote a stable system because of small transmission losses in the system and the ability of the system generator to cope with the complex load demand. As the system is stressed further, any increase of complex load demand will require more generated power to compensate with transmission line and thermal limits, hence the index will reduce towards 0. Therefore, a zero index will denote a weak voltage collapse prone system. The calculation of both indices are made directly using the traditional liner power flow equations as shown:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = J \begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} \tag{3.3}$$

where:

θ – Bus angles

V – Bus voltage

The respective VCPI index is then tested to various test systems which consist of the AEP 14 bus, 30 bus and 57 bus system where each system are pushed towards collapse with transformer tap changing action blocked. The results are as shown below [7]:

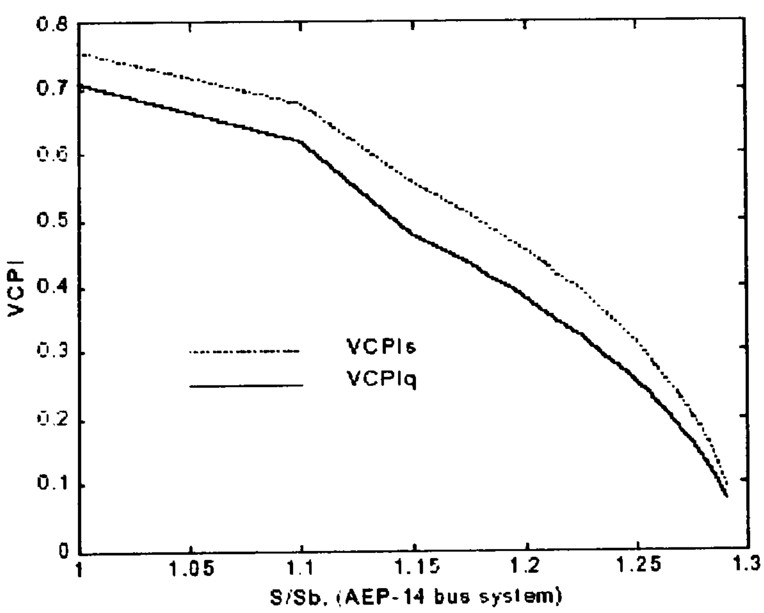


Figure 3.1: Results for 14 bus system [7]

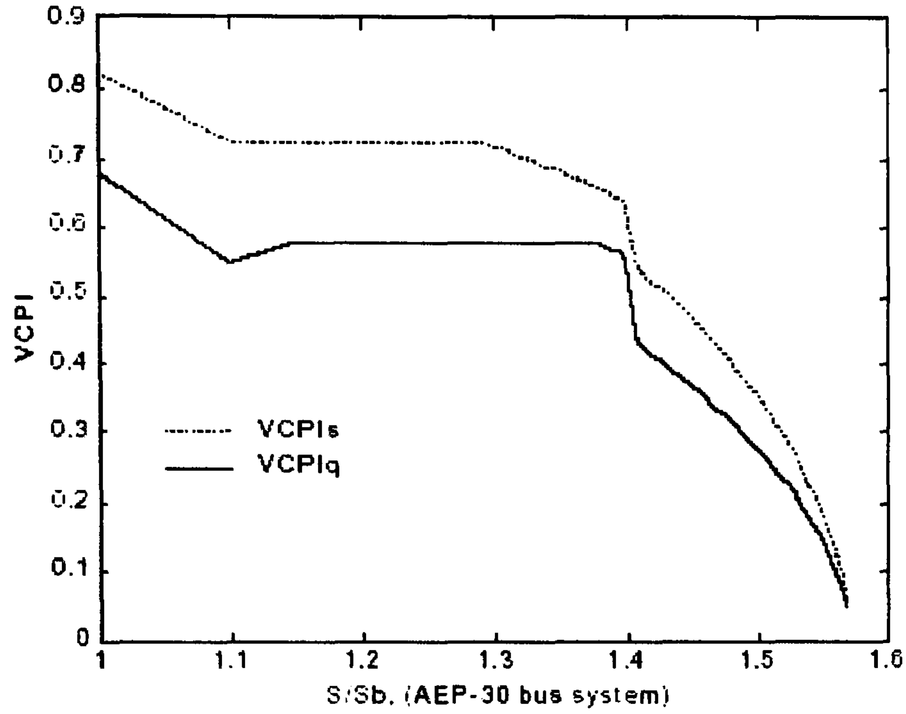


Figure 3.2: Results for 30 bus system [7]

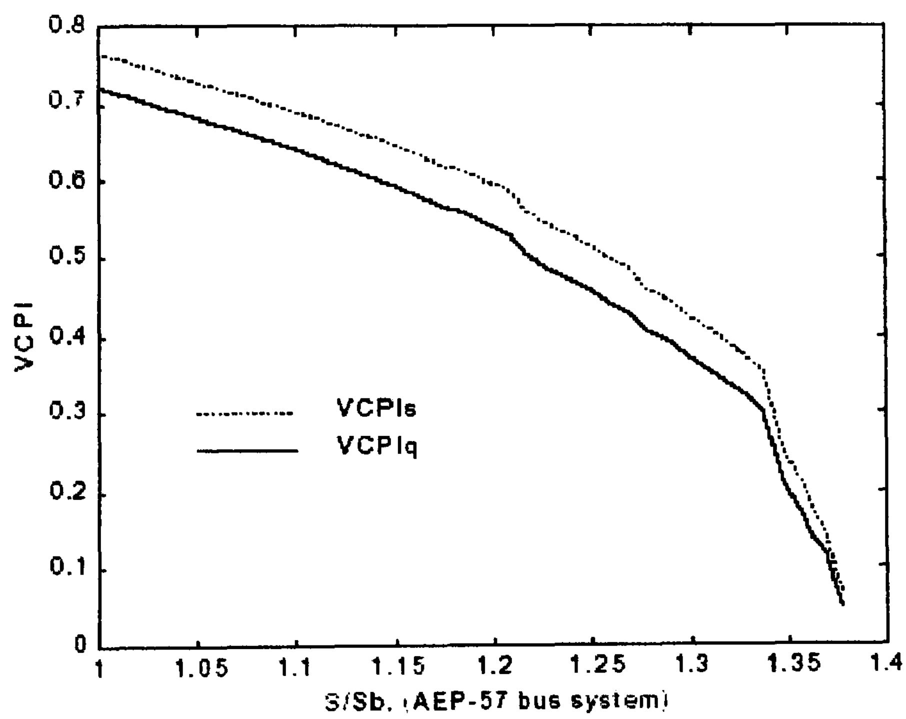


Figure 3.3: Results for 57 bus system [7]

Results shown in figure 3.1-3, shows that the VCPI indices reduces as the system is stressed further until the system goes to collapse.

Discussion

The result shown can be made for each bus in the system, but as the system gets bigger, VCPI calculations may take time as the number of buses and hence the size of the system Jacobean matrix will increase. Based on theoretical background, this method is a direct method as the VCPI indices can be obtained directly from power flow calculations. However, the method relies on the modifications of the power flow equations in calculating the indices. A suggestion can be made to improve the calculation time by using online Q and S values where some of them could be taken from periodic load flow calculations. In general, this method is also dependent upon observable values, which in this case are Q and S. This is similar to the proposed FDT approach that will be explained further in chapter 5.

In general, VCPI and other related indices can give an indication of the system's strength or weakness level with respect to voltage stability at a given operating level. Hence the indices are useful if a series of them are calculated and plotted with respect time to give a better representation of the system state. Apart from system indices, there are also methods of evaluation based upon the dynamics of the system known as bifurcation analysis.

3.3 Bifurcation Method [16, 22]

Bifurcation theory assumes that the power system parameters vary slowly and tries to chart the operating point movement as the system goes into instability. Bifurcation involves mathematical representation of each component in the system in terms of static, algebraic or differential-algebraic equations. The choice of equations used is dependent upon the type of action experienced by the power system elements whether it is fast or slow dynamics (or an equivalent static representation). Different sets of equations will give different results in locating the bifurcation point or also known as the threshold of instability as it represents a different set of assumptions.

System components are lumped into state and algebraic equations, which consists pf states and parameters. State element consists of reasonably fast varying elements, e.g. machine angles, bus voltage magnitudes and angles and current in generator windings. Parameters may consist of slow but varying parameters, e.g. real power or reactive demand, or any slowly varying parameters that will gradually change the system equations.

For power system analysis, these equations may be simplified by assuming a quasi-static-stability (QSS) solution of which only takes into account slow varying elements in the equations. Figure 3.4 shows a typical state space diagram plotting all possible operating points.

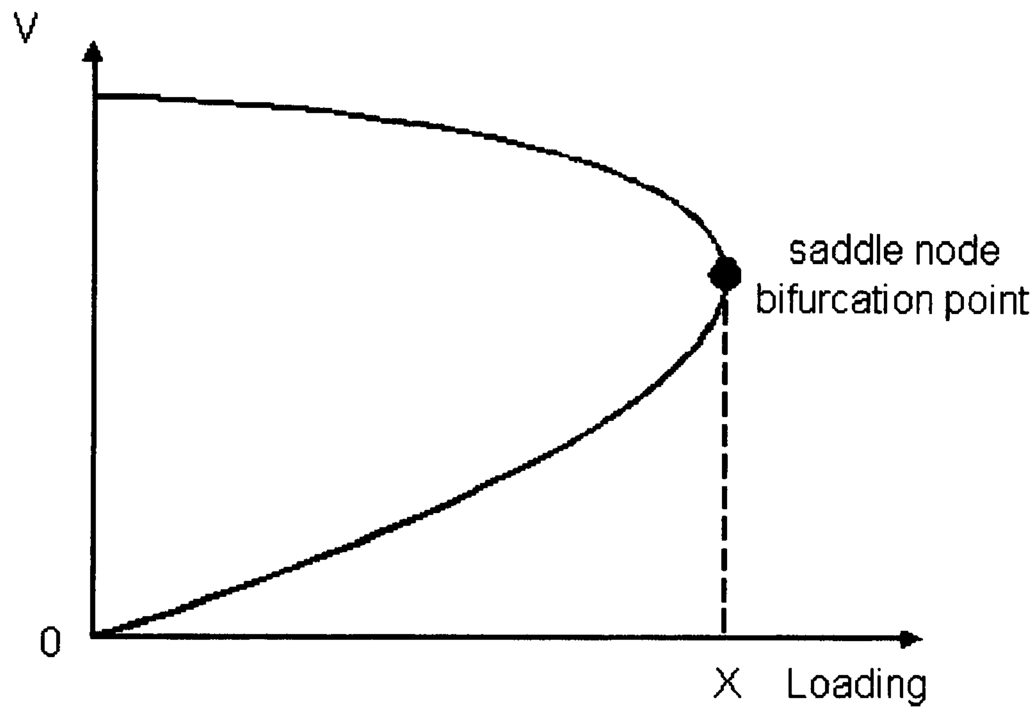


Figure 3.4: A bifurcation state space diagram plotting system stable operating points with varying loading parameters

Note that the bifurcation diagram shown neglects Hopf[1] and singularity induced bifurcation[2] where bifurcation may alter the high and low voltage equilibrium points. The profile of the state space behaviour above is very similar to the PV diagram showed in chapter 2. Figure 3.4 shows the that equilibrium points will slowly go towards the saddle node of bifurcation corresponding to the loading point X. Saddle node bifurcation or the threshold of stability denotes a singularity in the reduced

[1] Hopf bifurcation relate to parameter dynamics and non-linear elements in the power system
[2] Singularity induced bifurcation relates to fast network dynamics and differential-algebraic models of the power system elements.

Jacobean matrix. In other words it denotes the disappearance of system equilibrium as the system parameters are varied slowly. This can be seen in figure 3.5.

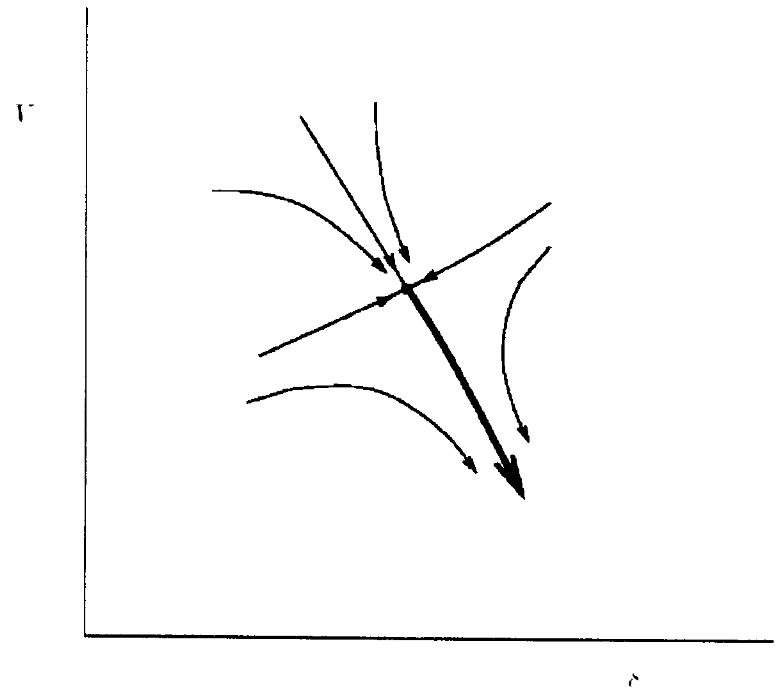


Figure 3.5: The operating point direction when there is a small perturbation to the system at the saddle node point in a state space diagram

Figure 3.5 shows that the operating point converges as the saddle node point before the application of perturbations into the system. The solid arrow line shows the direction of the operating point after perturbations. The direction of this arrow shows that the operating point goes away from the stable or equilibrium point. Hence system instability will occur.

Although the bifurcation point of a system can be calculated using static equations, but utilising dynamic parameters may enable further analysis of the bifurcation and its effects after bifurcation [16]. Looking into this, computation of the dynamic parameters may seem to be an interesting path of research but it has its own disadvantages as well as advantages. One of which is that the accuracy of the results depends on the accuracy of the dynamic model and any changes of the system (e.g. protection schemes) has to be noted as a change in the overall system model. Although accuracy of analysis is better but the computation burden is overwhelming.

Since bifurcation theory is dependent upon the QSS approximation, hence fast changes in the system e.g. large loss of load or generation can be difficult to be represented. On the other hand, utilising this approach, a better representation of the system behaviour is possible with the ability to determine the stable and unstable boundaries via solution of the state equations with certain constraints.

3.4 Eigenvalue analysis

This approach goes towards the same direction as bifurcation analysis. Eigenvalue approach looks into the eigenvalue of the jacobian matrix of the power system. In a stable system, the eigenvalues of the jacobian all have negative real parts[3]. As the system is slowly loaded to its maximum capability, the characteristics of the jacobian matrix changes up until the threshold of stability when one of the eigenvalue of the jacobian goes towards zero. Ultimately as the zero eigenvalue is reached, the system goes into oscillatory motion. Therefore minimum singularity in the jacobian matrix is considered to be one of the main factors to voltage collapse.

In dealing with large jacobian matrices, computing the eigenvalues may take time. Mathematical approaches have been utilised to alleviate this problem such as sparse techniques and also the Arnoldi approach as presented in [23]. Other approaches that looks into the eigenvalues are also explained in [16, 24].

3.5 Singularity decomposition method [23, 24]

This approach checks whether the jacobian matrix is singular. In short, singularity of the jacobian matrix shows that the power system operating point transgresses into the instability region and hence system collapse. The decomposition of the jacobian load flow matrix is by representing it in a form as shown in Eq. 3.4.

[3] This is possible for dynamic analysis purposes.

$$J = U \Sigma V^T \tag{3.4}$$

where:

J – system jacobian matrix

U, V – column matrix which has left and right singularity vectors

Σ - diagonal matrix of singular values

In [23], Eq. 3.4 is expanded as shown:

$$J = U \Sigma V^T = \sum_i \sigma_i u_i v_i^T \tag{3.5}$$

where:

σ - non-negative singular values

u – left singular vectors

v – right singular vectors

i – coordinates in the jacobian matrix

Taking Eq. 3.5 and the basic load flow equation (refer to Eq. 3.3) gives:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \sum_i \sigma_i u_i v_i^T \begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix}$$

manipulating the equation gives:

$$\begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} = \left(\sum_i \sigma_i u_i v_i^T \right)^{-1} \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix}$$

hence:

$$\begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} = \sum_i \sigma_i^{-1} u_i^T v_i \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \tag{3.6}$$

As the system goes towards collapse, the singular values goes towards zero or near to zero, hence Eq. 3.6 is simplified as shown:

$$\begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} \approx \sigma_{\min}^{-1} u_{\min}^T v_{\min} \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \tag{3.7}$$

Eq. 3.7 shows that any small variation in load will lead to extreme changes in terms of bus voltage and angles. In short, it is a form of voltage collapse indices as shown in [24] or its computation may be improved by utilising the Arnoldi algorithm as shown in [23]. The behaviour of the minimum eigenvalue is as shown in the graph below:

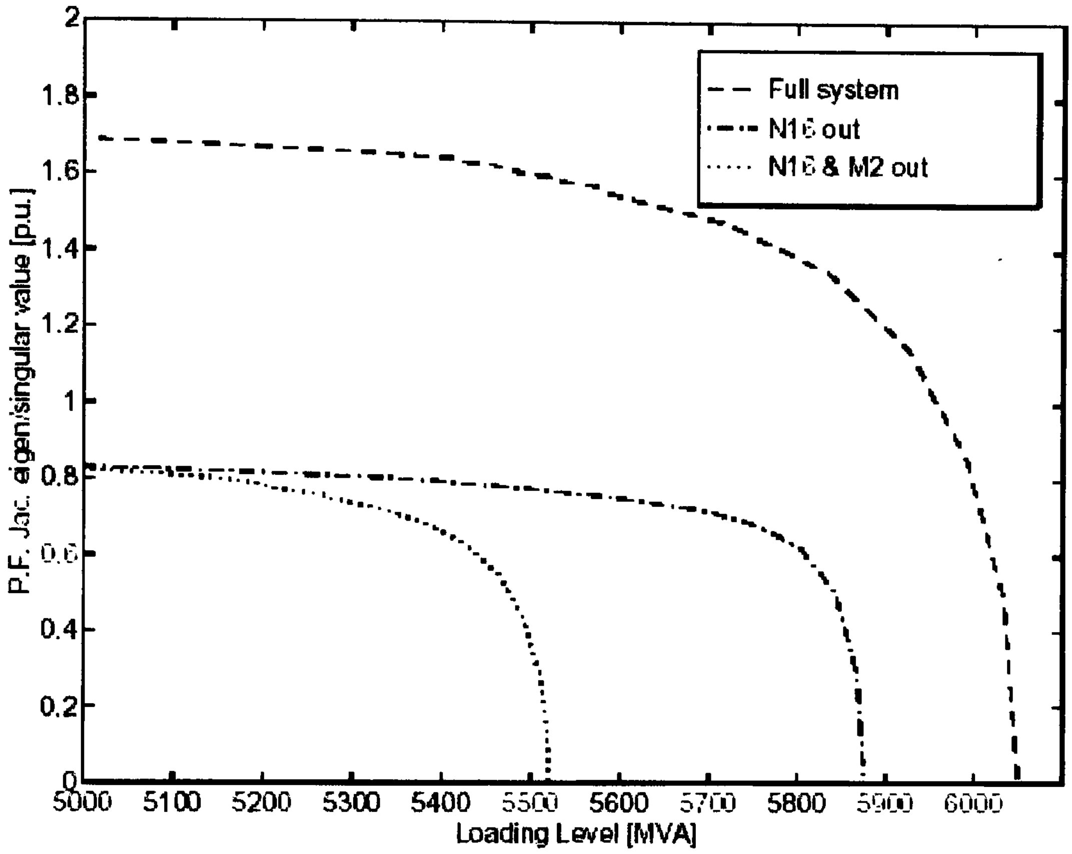


Figure 3.5: The behaviour of the power flow (PF) minimum eigenvalue or minimum jacobian singular value as the system loading level is increased. N16 representing a transmission line and M2 denotes a generator [25].

Figure 3.5 shows that as the system loading level is increased, the minimum eigenvalue of the system jacobian will slowly reduce until it is near the point of collapse where the reduction of the minimum jacobian eigenvalue is accelerated. Hence the system is rapidly moving towards collapse as indicated by the rapid fall in value.

Other approaches in detecting minimum singularity in the jacobian matrix involving matrix manipulations e.g. eigenvalue decomposition and the reduced jacobian determinant [26].

3.6 Network Partitioning method [4, 27]

The network partitioning method is a complement to all the voltage stability analysis method. It speeds up the computation by concentrating on the weak voltage area. This algorithm actually pinpoints the weak voltage area or in other words points out the area with the highest risk of voltage instability. Work presented in [27] shows that the network partitioning method is linked closely to the manipulation of the jacobian matrix, to be exact the usage of the reduced jacobian determinant method as presented in [24]. This method is based upon the equation as shown below:

$$\begin{bmatrix} \Delta P \\ \Delta Q \\ \Delta P_l \\ \Delta Q_l \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta V \\ \Delta \theta_l \\ \Delta V_l \end{bmatrix} \tag{3.8}$$

where:

P, Q, θ, V	- represents systems state and parameters
A, B, C, D	- represents the jacobian matrix elements
l	- represents the load bus analysed

Eq. 3.8 is very similar to the standard load flow equation as shown in Eq. 3.3 with the addition of analysed load bus states and parameters. Referring to Eq 3.8, and consider only the load variation at the analysed bus only, the equation is reduced to the following form:

$$\begin{bmatrix} 0 \\ 0 \\ \Delta P_l \\ \Delta Q_l \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta V \\ \Delta \theta_l \\ \Delta V_l \end{bmatrix} \tag{3.9}$$

Comparing the elements of the jacobian matrix, elements of sub-matrix D represents equations that involve bus voltage magnitude and angle of the analysed buses, hence the jacobian has to be manipulated to represent D as explained in [24], Eq. 3.9 is manipulated to show the following.

$$\begin{bmatrix} \Delta P_l \\ \Delta Q_l \end{bmatrix} = \left[D - CA^{-1}B\right]\begin{bmatrix} \Delta \theta_l \\ \Delta V_l \end{bmatrix} = \left[D_{ll}^{'}\right]\begin{bmatrix} \Delta \theta_l \\ \Delta V_l \end{bmatrix} \tag{3.10}$$

Further explanation on the manipulation of Eq. 3.9 can be found in [12, 24, 27]. Applying this concept yields:

$$\begin{bmatrix} \Delta \theta_l \\ \Delta V_l \end{bmatrix} = [L]\begin{bmatrix} \Delta P_l \\ \Delta L_l \end{bmatrix} \tag{3.11}$$

where L is the direct inverse of the reduced jacobian D'_{ll} . By right, L will have the same dimension as D'_{ll}. The composition of matrix L is as shown:

$$L = \begin{bmatrix} E & F \\ G & H \end{bmatrix} \tag{3.12}$$

Therefore using Eq. 3.11 and 3.12 a relation can be made for the analysed bus voltages and angles respectively. Expanding this method to include a second connected bus, using Eq. 3.8 to 3.12 and assuming analysis from one bus point of view (in this case in terms of the first bus) the resultant matrix L will consists of values with the connected second bus. From these, the determinant of the matrix L can be made where the connected bus with the lowest determinant will be considered to be the weakest in the system and is prone to instability. From the literature, voltage collapse is known to be localised in nature [16]. The use of Eq. 3.11 will enable the determination of neighbouring weak buses as proven in [24]. Repeating all these recursively for each load bus, a determination of the weak areas becomes possible.

This method has an advantage where the identification of weak areas are identified locally, i.e. calculations are made locally and do not involve the rest of the system, hence computation burden is significantly reduced together with the implementation of sparse techniques. On the other hand, the identified weak bus is only valid for a particular operating point. Hence for different operating points, re-identifications are needed.

3.7 Energy function method [16]

Energy function is a dynamic representation of the voltage collapse threshold. The method is in fact more suited to be called 'Quasistatic' analysis. This is similar to the QSS concept mentioned previously in this chapter where the fast dynamic is slowed down as to simplifying calculations. In a fault situation, the operating point will oscillate until it reaches the new steady stable state value.

Utilising Lyapunov functions and assuming quasistatic situations for dynamic models, the 'frozen equilibrium' points is tracked and stability limits calculated creating a series of maximum limit points. Utilising these maximum stability points, the nearest unstable point is calculated in terms of its potential energy required to cross into the unstable region. If the energy at this limit is higher than the operating point energy value, the operating point will revert back to the original operating point at the bottom of the 'potential well'. This effect can be best seen graphically in figure 3.6.

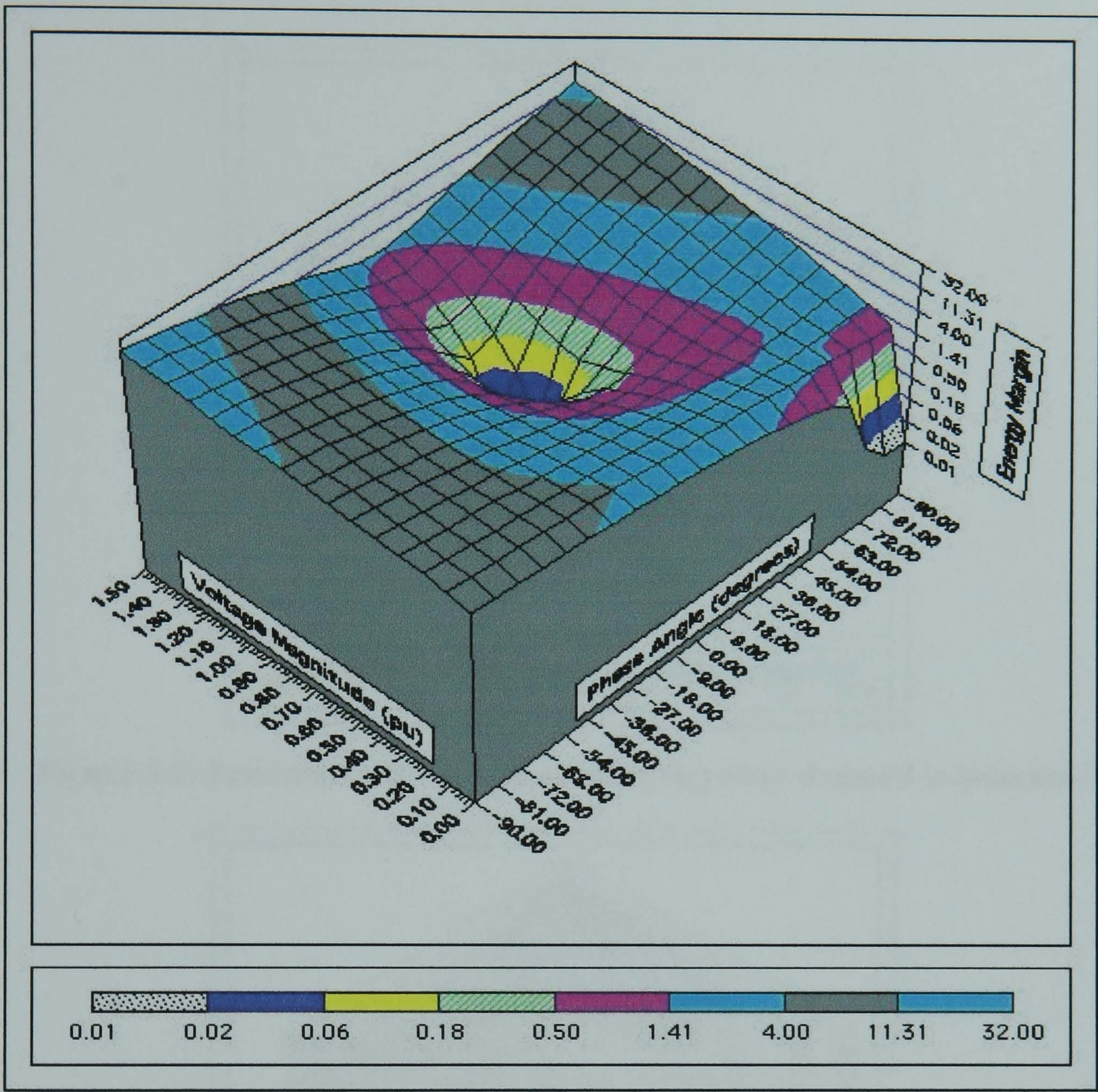


Figure 3.6: A graphical diagram showing a potential well with its maximum stability limits around it. This is for a 1 generator connected to 1 load via a transmission line [16].

Referring to figure 3.6, as the reactive power demand is increased, the border of the potential well will react in respond to changes of system parameters. This in effect will change the shape of potential well as shown in figure 3.7, 3.8 and 3.9.

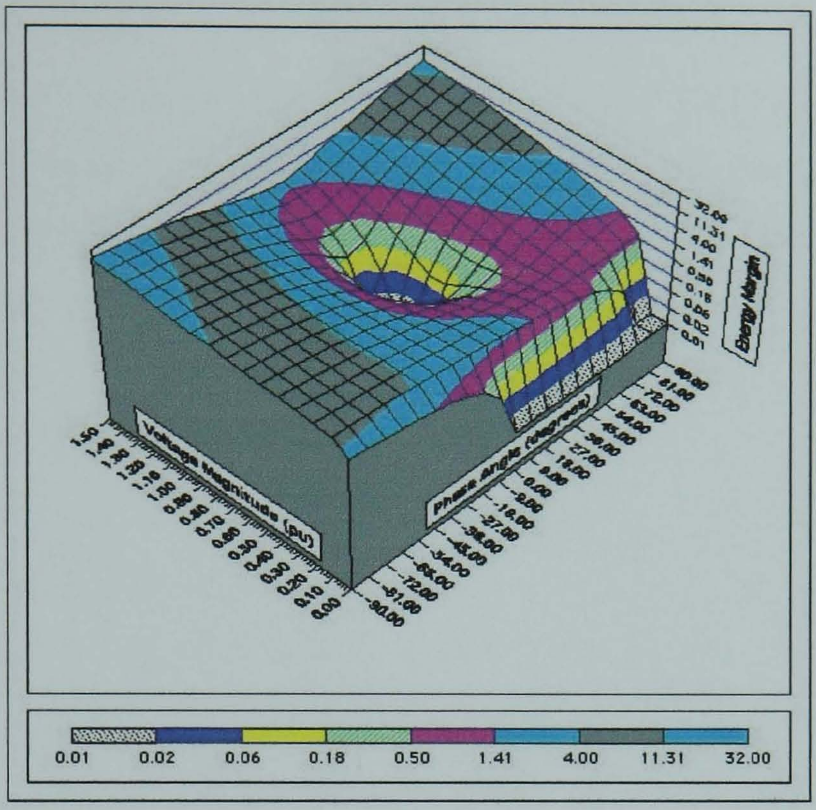


Figure 3.7: Potential well changes as reactive power demand is increased

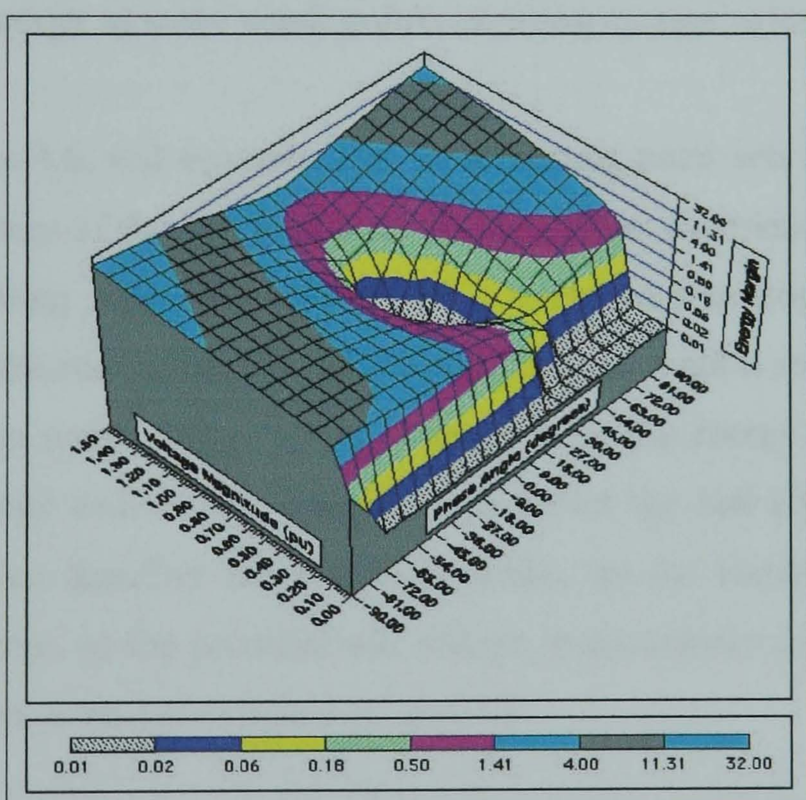


Figure 3.8: Further increased of reactive demand reduces the potential energy at some weak points, hence increasing the risk of instability.

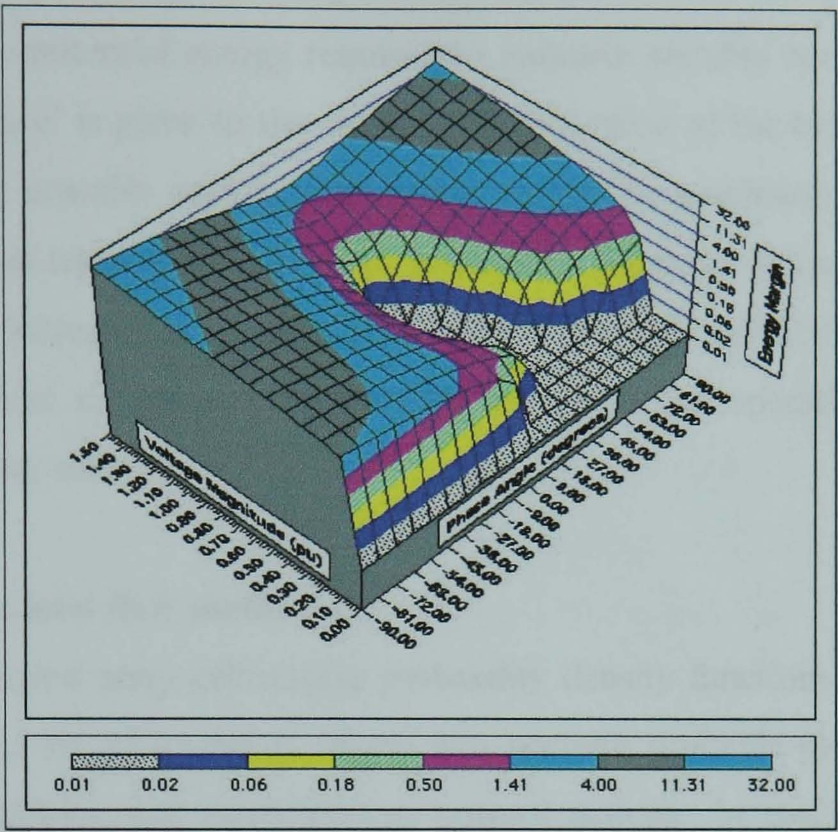


Figure 3.9: Further increased of reactive demand reduces further the potential energy at some weak points creating system collapse.

Looking at figure 3.6, and assuming that the operating point acts as the trajectory of a ball at the bottom of the potential well. As the system experiences a 'disturbance', the system operating point will oscillate until it resides to a new operating point. This is analogous to the ball rolling around the potential well until it settles down again at a new equilibrium point. In figure 3.6, since the potential energy required to escape the stable potential well is high enough to constrict the ball movement inside the potential well area therefore the system is stable. As the reactive load demand is increased, the shape of the potential will change in accordance to the change of the system parameters as shown in figure 3.7 and 3.8.

For figure 3.7, when a 'disturbance' is applied to the system, the ball will oscillate around the potential well. Provided that the disturbance magnitude is not severe, the ball will not have enough energy to escape from the potential well. Hence the system will ultimately reside to a stable operating point in the potential well.

In figure 3.8, one of the 'lips' of the potential well has reduced in height (in potential energy terms, the potential energy required to maintain stability has reduced). When a small 'disturbance' is given to the system, the oscillation of the ball will enable it to roll out into the unstable region. This is known as the maximum stable operating limit, or in another term, the saddle node bifurcation point. In figure 3.9, the reactive power is further increased until the system cannot maintain stability hence the system collapse since there are no stability borders to restrict the operation point to stay within the potential well.

3.8 Probabilistic load flow method [1]

This method is based upon calculating probability density functions (PDFs) for each bus based upon a set of scenarios taking into account probable generator dispatch, generator availabilities and other system control actions. In essence PDF can be represented as total of probable load flow values based upon scenario possibilities. This approach is very mathematical in nature and it is dependent upon the frequency of possible simulations or scenarios available for the system. In this approach, the author [1] based its example on the CIGRE study system that represented the 1982 blackout in the northern part of Belgium. The PDF of each bus is calculated and its variations relative to the simulation scenarios are recorded using standard deviation formulas. Weak busses will tend to have a high standard deviation value as opposed to strong busses. Using calculated bus PDF and its standard deviation as a yardstick, possible control actions are included and its effects to the PDF and its standard deviation is recorded. The best control action will be represented as the one which yields the greatest reduction of PDF standard deviation.

From the author point of view, this approach may represent an accurate probable behaviour of a system in a voltage instability scenario but when considering all possible events it may be difficult to calculate for a big and complex system. Although simplifying the system is a probable solution but it would reduce the accuracy of results.

3.9 Artificial neural network method [11, 28]

This approach employs the use of Artificial Intelligence (AI), to be exact Neural Networks (NN), in estimating a system state relative to voltage stability. This approach is inherently very similar to decision tree (DT) approach with several subtle differences. The theoretical background of DTs will be explained further in chapter 4.

In general, NN approaches are dependent upon the effectiveness of the training set and the NN ability to learn and adapt from these learning sets. Application of NN in power systems may vary from load scheduling applications, system stability evaluations and system security evaluations.

The author in [11] uses NN approaches to track the voltage collapse boundary from possible simulated operating region which ranges from stable operating points to unstable operating points. By tracking the voltage collapse boundary, the developed NN, are able to classify test system operating states into stable and collapse prone classes. This approach is very similar in concept with [29-35] with the exceptions of its tool of approach which is DT.

Another approach made in [28] utilises the operating P-V-Q values and system parameters to obtain the distance of the system operating point from the boundary of instability using a neuro-fuzzy approach known as ANFIS. Test results obtained are very promising in terms of its flexibility and adaptability of the algorithm.

From the author's point of view, NN approach is a branch of the AI tree with capabilities of full automatic learning as compared to the DT approach. However, NN also known as the 'black box' approach has a disadvantage of user controllability or information tracking between the input and the output of the NN module and also its dependence upon the extend of the training set. Due to this, for unforeseen situations, it is difficult to assess the NN decision or evaluation results. Apart from that there are dimensionality problems where the time taken to train NN modules is proportionally increased with the size of the system. The bigger the system, the more time it would take to train the NN module. A solution to this is the reduction of the system network into small and manageable areas.

3.10 Conclusions

The chapter describes some voltage collapse analytical methods, all of which have the objective of determining the proximity to collapse. From these methods, mitigation efforts are formulated and some of which help to alleviate the stress of the power system. This gives more room for new stable operating points should the system goes through a disturbance. Other mitigation efforts involve plans to stop power system components from reacting adversely, hence increasing the severity of voltage collapse.

One common point in these methods is that the majority of them manipulate the traditional load flow equations and extract information from the jacobian matrix in terms of its eigenvalues.

Other approaches mentioned do differ from traditional approach, artificial intelligence (AI) as one example, relies on non-linear mapping. The AI branch of analysis has the advantage of computational burden and its adaptability to different environments. Note that not all AI based methods are mentioned in this chapter as more examples will be given in chapter 4.

3.11 References

1. N D Hatziargyriou, T S Karakatsanis, *Probabilistic load flow for assesment of voltage instability*. IEE Proceedings of Generation, Transmission and Distribution, 1998. **145**(2), p. 196 - 202.
2. Thomas J Overbye, Ian Dobson, Christopher L DeMarco, *Q-V Curve Interpretations of Energy Measures for Voltage Securities*. IEEE Transactions on Power Systems, 1994. **9**(1), p. 331 - 340.

3. Scott Greene, Ian Dobson, Fernando L Alvarado, *Sensitivity of the loading margin to voltage collapse with respect to arbitary parameters.* IEEE Transactions on Power Systems, 1997. **12**(1), p. 262 - 272.
4. A Nuhavonic, M Glavic, N Ptljaca, *Validation of a Clustering Algorithm for Voltage Stability Analysis on the Bosnian Electrical Power System.* IEE Proceedings of Generation, Transmission and Distribution, 1998. **145**(1), p. 21-27.
5. Canizares, Claudio A, *Voltage Collapse and transient Energy Function Analysis of AC/DC Systems.* University of Wisconsin-Madison. 1991. PhD Thesis, 158 pages.
6. Ian Dobson, Liming Lu, *Voltage Collapse Precipitated by the Immediate Change in Stability when Generator Reactive Power Limits are encountered.* IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications, 1992. **39**(9), p. 762 - 766.
7. Yuan-Lin Chen, Wen-Bin Liao, Shun-CHung Wang, Kun-Yuan Shen, Shih-Chung Tuan. *Voltage Collapse Proximity Indicators of a Power System.* in *University Power Engineering Conference (UPEC).* 2001. Swansea, Wales, United Kingdom.
8. K L Lo, Q Y Zheng, T X Zhu, S K Tso, *Fuzzy-set approach to dynamic voltage security assessment.* IEE Proceedings of Generation, Transmission and Distribution, 1995. **142**(2), p. 190 - 194.
9. K L Lo, Q Y Zheng, T X Zhu, S K Tso, *Fuzzy Reasoning for knowledge-based assessment of dynamic voltage security.* IEE Proceedings of Generation, Transmission and Distribution, 1996. **143**(No 2), p. 157 - 162.
10. Mukhedkar R. A, Nouri H., Davies, T.S. *Comparative study and performance analysis of online static voltage collapse indices in interconnected systems.* in *University Power Engineering Conference (UPEC).* 2001. Swansea, Wales, United Kingdom.
11. Hasan, Usama, *Artificial Neural Networks for Voltage Collapse Monitoring.* University of London, Imperial College of Science, Technology and Medicine. 1998. PhD thesis, 181 pages.

12. Souza, Antonio Carlos Zambroni de, *New Techniques to Efficiently Determine Proximity to Static Voltage Collapse*. University of Waterloo. 1996. PhD Thesis, 133 pages.

13. Chiang Hsiao-Dong, Ian Dobson, Robert J Thomas, James S Thorp, Lazhar Fekih-Ahmed, *On Voltage Collapse in Electrical Power Systems.* IEEE Transactions on Power Systems, 1990. **5**(2), p. 601 - 611.

14. Navarro-Perez, Rogelio, *Voltage collapse proximity assessment for the operational planning of power systems.* Department of Electrical Engineering. University of London, Imperial College of Science, Technology and Medicine, London. 1993. PhD Thesis, 207 pages.

15. Hawkins, Nigel Trevor, *On-line reactive power management in electrical power systems.* Department of Electrical and Electronic Engineering. University of London, Imperial College of Science, Technology and Medicine, London. 1996. PhD Thesis, 195 pages.

16. Ian Dobson, Thierry Van Cutsem, Costas Vournas, Chris DeMarco, Mani Venkatasubramanian, Tom Overbye, Claudio Canizares, *Voltage Stability Assesment, Procedures and Guides: Chapter 2: Basic Theoretical Concept.* IEEE/PES Power System Stability Subcomittee, Special Publication. January, 2001, p. 2-100.

17. K L Lo, L K Peng. *Design of Artificial Neural Networks for On-line static Assessment problems.* in *IEE International Conference on Advance in Power Systems Control, Operation and Management.* 1997. Hong Kong.

18. K L Lo, T X Zhu, *Arnold Diffussion in a particular multi-machine power system.* IEE Proceedings of Generation, Transmission and Distribution, 1998. **146**(3), p. 257 - 264.

19. K L Lo, Y J Lin, W H Siew, *Fuzzy-logic method for adjustment of variable parameters in load-flow calculation.* IEE Proceedings of Generation, Transmission and Distribution, 1999. **146**(3), p. 276-828.

20. S K Tso, L Guan, Q Y Zeng, K L Lo, *Fuzzy assessment of power system transient stability level based on steady state data.* IEE Proceedings of Generation, Transmission and Distribution, 1997. **144**(6), p. 525 - 531.

21. M Begaovic, J Bright, T Domin, S. Easterday-McPadden, A Girgis, W Hartmann, C Henville, M Ibrahim, K Kozminski, R Marttila, G Michel, K Mustaphi, D Novosel, B Pettigrew, M Sachdev, H Shuh, P Solanics, J Williams, *Voltage Collapse Mitigation*. IEEE Power System Relaying Committee. December, 1996, p. 33 pages.

22. Thierry Van Cutsem, Costas Vournas, *Voltage Stability of Electrical Power Systems*. 1998: Kulwer Academic Publishers, 378 pages.

23. Garng Huan, Tong Zhu. *A new method to find the voltage collapse point*. in *Power Engineering Society Summer Meeting 1999*. 1999. Edmonton, Canada: IEEE, p. 1324 - 1329.

24. Souza, A C Zambroni de, *Identifying a vanishing eigenvalue in voltage collapse analysis with consideration of limits*. IEE Proceedings of Generation, Transmission and Distribution, 2001. **148**(2), p. 263 - 267.

25. Claudio Carnizares, Ian Dobson, Alberto Berizzi, Jozef Van Hecke, Paolo Marannino, Naoto Yorino, Kzuyuki Tanaka, Robert Schlueter, Sandro Corsi, Pozzi Massimo, *Voltage Stability Assessment, Procedures and Guides: Chapter 4: Voltage Stability Indices*. IEEE/PES Power System Subcommittee, Special Publication. January, 2001, p. 156-220.

26. Souza, A C Zambroni de, *Discussion on some voltage collapse indices*. Electric Power System Research, 2000(53), p. 53-58.

27. A C Zambroni de Souza, V H Quintana, *New technique of network partitionining for voltage collapse margin calculations*. IEE Proceedings of Generation, Transmission and Distribution, 1994. **141**(6), p. 630 - 636.

28. K Yabe, J Koda, K Yoshida, K H Chiang, P S Khedkar, D J Leonard, N W Miller, *Conceptual designs of AI based systems for local prediction of voltage collapse*. IEEE Transactions on Power Systems, 1996. **11**(1), p. 137 - 145.

29. Louis Wehenkel, *Automatic Learning Techniques in Power Systems*. The Kluwer International Series in Engineering and Computer Science. Vol. 429. 1997: Kluwer Academic Publishers, 279 pages. ISBN 0-7923-8068-1.

30. Louis Wehenkel, I Houben, M Pavella, L Riverin, G Versailles. *Automatic Learning Approaches for On-line transient stability preventive control of the Hydro-Quebec System. Part 1: Decision Tree Approaches.* in *IFAC Symposium on Control of Power Plants and Power Systems.* 1995. Cancun, Mexico, p. 231-236.

31. Louis Wehenkel, C Lebrevelec, M Trotignon, J Batut. *A probabilistic approach to the design of Power System protection schemes against blackouts.* in *IFAC-Cigre Symposium on Control of Power Systems and Power Plants.* 1997. Beijing, p. 506 - 511.

32. Xavier P Boyen, *Design of Fuzzy Logic Based Decision Trees Applied to Power System Transient Stability Assessment.* Department of Electrical and Computer Science. University of Liege, Belgium. 1995. Masters Thesis, 133 pages.

33. Xavier P Boyen, Louis Wehenkel. *Automatic Induction of Continuous Decision Trees.* in *Proceedings of Information Processing and Management of Uncertainty in Knowledge-Based Systems.* 1996. Granada, p. 419 - 424.

34. Xavier P Boyen, Louis Wehenkel. *Fuzzy Decision Tree Induction for Power System Security Assessment.* in *IFAC Symposium on Control of Power Plants and Power Systems.* 1995. Cancun, Mexico, p. 151-156.

35. Xavier P Boyen, Louis Wehenkel, *Automatic Induction of Fuzzy Decision Trees and its Applications to Power System Security Assessment.* International Journal of Soft Computing and Intelligence, 1999. **102**, p. 3 - 19.

Chapter 4: Machine Learning techniques

4.1 Introduction

This chapter starts with an introduction of machine learning techniques and their possible applications to power system problems. Taking into account the nature of the problem where the analyst are over burdened with the amount of data, a technique is needed to 'mine' the database to extract relevant information that may be of use to the power system analyst. Hence an introduction of decision tree (DT) and its evolution, e.g. EMV Tree, ID3, C4.5 and its evolution and the Top Down Induction Decision Tree (TDIDT). Adding a fuzzy element into the decision tree yielded a series of new DT approaches such as the fuzzy ID3, Fuzzy Induction of Decision Trees with Missing Values (FIDMV), Outbound Fuzzy Decision Tree (OFDT) and the Fussy Fuzzy Decision Tree (FFDT). Through these explanations of DTs and Fuzzy DTs (FDT), the author explains several important elements and issues used in the DT and FDT construction, which will be used further in the FDT proposal in Chapter 5. The Chapter concludes with a conclusion with regards to the DT and FDT and its important issues to be applied to the proposed FDT approach.

4.2 What are machine learning techniques

Machine learning techniques also known as automatic learning technique are related to a set of methods for extracting high level information (knowledge) from databases[1, 2]. The term database is referred to as a collection of information found from computer simulations or observations. For example, for power system analysis, a database may consist of a collection of load flow values of each bus and its connection or it could even consist of systems states in various scenarios. Hence in extracting useful information, we are actually discovering knowledge or recognising patterns in database and implementing that knowledge to cater for unseen instances.

From a macro level point of view, knowledge extraction is merely obtaining a conclusion from a set of behaviours or data sets. Alternatively, looking from a micro level point of view, data extraction is a complex combination of statistics, cognitive reasoning and also information manipulation [1].

Statistics as mentioned above are merely the tools to represent knowledge whereas cognitive reasoning is referred to the ability in highlighting important decisions from databases. Information Manipulation on the other hand enables better and more efficient algorithm building. Since its creation, Machine Learning Techniques have advanced in terms of approach and application in areas where knowledge acquisition from databases is important.

4.3 Machine learning techniques in power system analysis

A power system may be simplified to a 2 bus, 1 generator and one load with an infinite bus or it may be as complex as a 1000 bus network, which may consist of several hundred generators, and various network enhancements e.g. load tap transformers, shunt compensators and power electronic devices. As explained in Chapter 3, the best way to determine the state and effects of each component in the system is via mathematical formulation, i.e. energy function and bifurcation for system state calculations and static load flows for system components states calculations. Unfortunately, due to complexity of the system, a more statistical approach may be adopted with a small reduction of accuracy but with a reduced calculation time. This technique may be ideal and practical for industrial applications.

Machine learning methods can be seen as an alternative approach to power system analysis where it involves manipulation and understanding of databases. The database in question may consist of several attribute types for example system operating levels, operating limits and 'states' of each component in a power system. The ability to extract useful knowledge may be useful for automatic assessment of system without relying on the expert knowledge of the power engineers and network operators. Several applications have been made towards power system security assessments [1-6].

Machine learning technique can be applied in various forms of approach e.g. Artificial Neural Network (ANN) approach [2, 7, 8], DT approach [3, 6], Hybrid Neuro-Fuzzy approach [9] and also the FDT approach [2, 4, 10, 11]. Going through each method may be interesting but it would take a lot of time to understand each method as each has its own mathematical complexity and characteristics. On the other hand, looking at the results of these techniques, it opens up a new avenue of Intelligent Computing approach for power system applications. But for this research purposes, the decision tree and its evolutions are looked into.

4.4 Decision tree approach

It is also known as the 'Grey Box' approach or the 'Top Down Induction Tree' approach (TDIDT). It has some advantages comparable to the Neural Network Method (Black Box approach) where the Decision Tree approach give users an understanding on how information is classified. Therefore giving users tracking abilities during the knowledge acquisition process.

In general, the decision tree is shaped like an upside down tree as shown in figure 4.1. The tree starts with the top node known as the **'root'**. Emanating from it, are **'branches'** pointing towards other nodes. These nodes are initially known as **'test nodes'**. As the algorithm goes further, more branches will appear from the respective nodes, which is dependent upon the result of the partitioning test made at the test nodes. If the test yields a result of no more possible branch creation, the node is labelled as the **'leaf node'**. If the result is opposite, the algorithm will continue with the partitioning sequence. This will continue until all data are partitioned into useful groups or classes. The final sequence of knowledge extraction will correspond to the interpretation of each class. This can be in a form of stable and unstable classes or other form of value classes dependent upon the nature of the data analysed e.g. weak and strong buses in a power system or medical symptoms and conclusions to a disease in a medical diagnostic system. A typical DT will then go through a series of test data to determine its accuracy in selecting the best conclusion to a possible value hence creating a conclusion to the label of value tested.

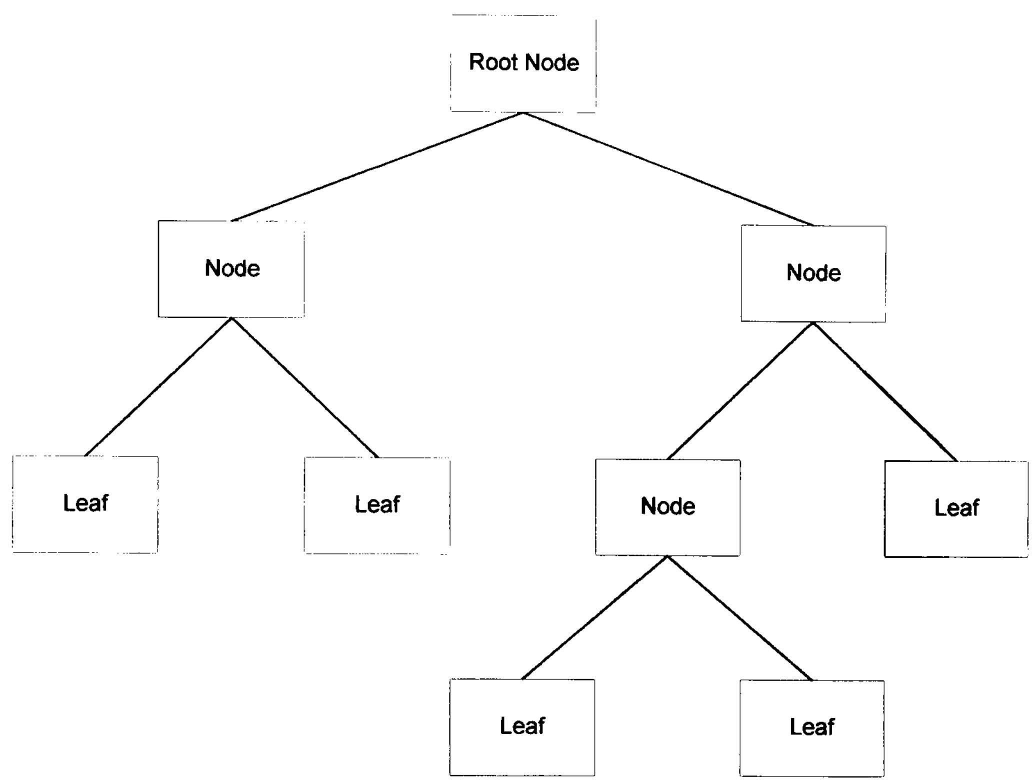


Figure 4.1: General Form of Decision Tree

In general, there are various configuration of Decision Tree available. Some relate to an uncertainty problem, general data mining sequence and decision-making problems. The following examples and discussions will give a basic understanding of a decision tree.

4.5 EMV decision tree [12]

Also known as the Extra Monetary Value decision tree. It analyses a decision problem and labels its consequences with a value. The labelled components are then used to calculate the ultimate effect if a particular decision is made. A simple example is the coin toss game.

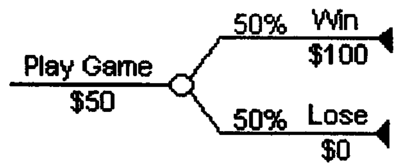


Figure 4.2: EMV decision tree

The possible result from this example is simply 'heads' or 'tails'. Since there are only 2 possible outcomes, probability shows that it has a 50-50 chance of either result to happen. In understanding the EMV Tree, it is broken down into several general components.

General Components:

Looking at the general structure of the EMV tree, the coin toss game does not represent the true picture of the EMV tree and its complexities. Therefore explanations of the general components will be based upon a general EMV tree model as shown in Figure 4.3 below:

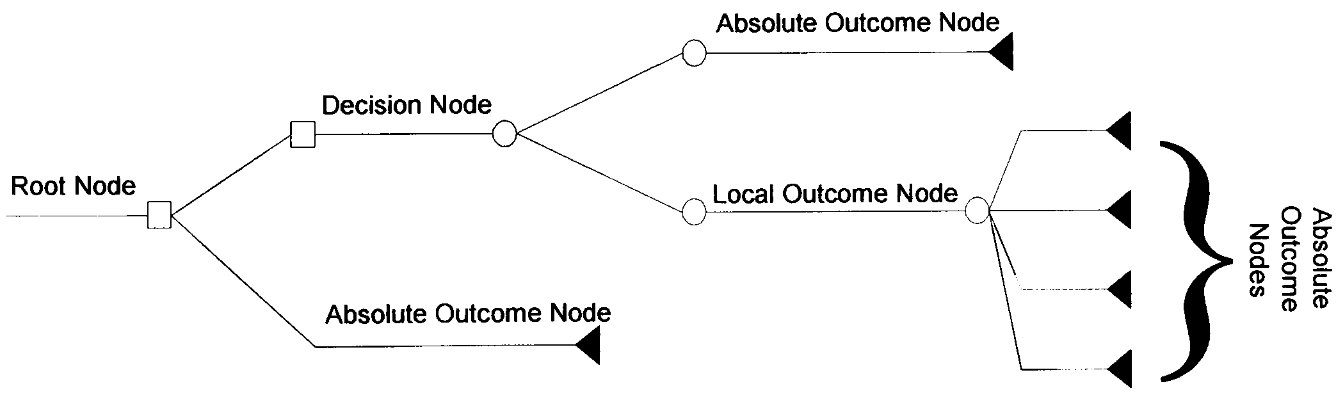


Figure 4.3: General Structure of the EMV tree

(i) Outcome or End Node

It denotes the possible outcome of a decision in terms probability. Referring to figure 4.3, the outcome could either be 'win' or 'lose'. Attached to each outcome is the value of each outcome. In this case of the coin toss game, winning the game will give $100 reward and losing the game will not give any value ($10). In a complex EMV tree, there are absolute and local outcome nodes (as shown in figure 4.3). An absolute outcome node represents the final effect of a decision and a local outcome node represents a local decision, which has further possible future outcomes attached to it.

(ii) Decision Node

This node represents an outcome without any probability value attached to it. The value it represents is obtained from calculated precedent EMV values or value obtained by the decision made at that stage. It can be seen more clearly in the illustrated example later.

(iii) Root Node

It denotes the node at which an event will take place. In this case is the coin-tossing event illustrated in figure 4.2 ('play game' node). The value at this node represents the final decision value based upon comparisons from immediate nodal EMV calculations. The node with the highest EMV score is selected for the root node value. Hence the final value of the event if the best decision steps is made.

EMV Calculations.

The EMV is calculated from the outcome nodes towards the root node; i.e. from the branch of the EMV tree and calculated back towards the root to obtain the decision value of an event. In the case of figure 4.3, the local EMV value can be calculated using the formulae as shown in Eq. 4.1 for each absolute outcome.

$$Local\ EMV = Value\ x\ P \qquad 4.1$$

where:

Value - The value of an outcome

P - The probability attached to the outcome

Each outcome will have its own local EMV value at which will be added up to create the total EMV value as shown by the Eq. 4.2:

$$\begin{aligned} Total\ EMV &= \sum_{n=1}^{N} Value_n\ x\ P_n \\ &= \sum_{n=1}^{N} Local\ EMV \end{aligned} \qquad 4.2$$

In the case of the EMV tree in figure 4.3, the total EMV may be represented as a local EMV value in local outcome branch. As for the value of the EMV at each outcome can be either positive or negative dependent upon the event itself.

Tree Building

The best way to explain the Tree Building process is via an example as shown.

4.5.1 Example: Bulk Ordering T-shirts for a local championship [12]

A shopkeeper who sells sporting goods in a small community whose local college basketball team has made it into the finals of the national championships. If the local team were to win, the expected sales will be between 2000 to 10,000 T-shirts at $20 each. The problem comes in when the ordering invoice has to be in a week before the championship games hence without knowing whether the local team would win or not. The probability of the local team to win is 60%. The cost of ordering will be $7 each. If there is any unsold t-shirt after the game, it can be sold as scrap for $2 each. Ordering the t-shirt has to be made at either 5000 or 10000 items only. What are the best decision steps to tackle this problem?

Assumptions:

Since the shopkeeper is confident in selling between 2000 to 10,000 T-shirts, hence the best estimate of demand will be as follows: 2000 – 4000, 4000 – 6000, 6000 – 8000 and 8000 – 10,000. Hence this will divide the probability label for the absolute outcome demand to be 25% for each demand band. Since the absolute outcome requires one value of demand rather than a range of demand possibilities, the middle demand value of each band is selected. Hence possible demand values are 3000, 5000, 7000 and 9000 with 25% probability attached. Taking into account the problem and also the assumptions made, the general tree structure is constructed as shown in figure 4.4.

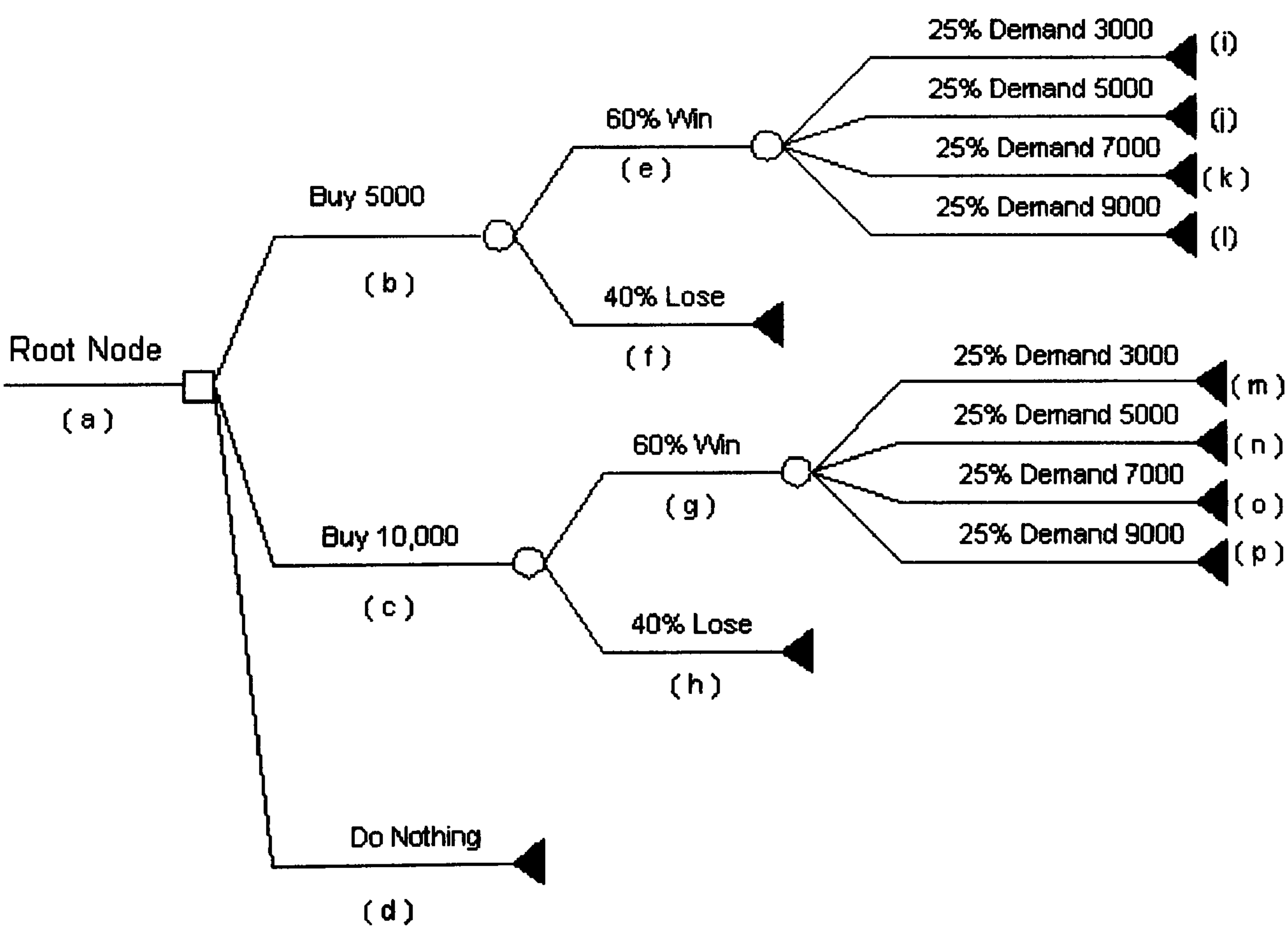


Figure 4.4: General tree structure for the t-shirt example

The next step is to calculate the respective outcome values and EMV score for branch (a – p). Calculations starts with outcome nodes (i – p) as shown below:

(i) Demand 3000

$$Outcome\ Value_{(i)} = (3000\ x\ \$20) - (5000\ x\ \$7) + (2000\ x\ \$2) = \$29{,}000$$

$$EMV_{(i)} = \$29{,}000\ x\ 25\% = \$7{,}250$$

(j) Demand 5000

$$Outcome\ Value_{(j)} = (5000\ x\ \$20) - (5000\ x\ \$7) = \$65{,}000$$

$$EMV_{(j)} = \$65{,}000\ x\ 25\% = \$16{,}250$$

(k) Demand 7000 and (l) Demand 9000

Since the amount of available T-Shirt is only 5000, hence the maximum outcome value and EMV score is similar to (j) which is $65,000 and $16,250 respectively.

(m) Demand 3000

$$Outcome\ Value_{(m)} = (3000\ x\ \$20) - (10{,}000\ x\ \$7) + (7000\ x\ \$2) = \$4000$$

$$EMV_{(m)} = \$4{,}000\ x\ 25\% = \$1{,}000$$

(n) Demand 5000

$$Outcome\ Value_{(n)} = (5000\ x\ \$20) - (10{,}000\ x\ \$7) + (5000\ x\ \$2) = \$40{,}000$$

$$EMV_{(n)} = \$40{,}000\ x\ 25\% = \$10{,}000$$

(o) Demand 7000

$$Outcome\ Value_{(0)} = (7000\ x\ \$20) - (10{,}000\ x\ \$7) + (3000\ x\ \$2) = \$76{,}000$$

$$EMV_{(o)} = \$76{,}000\ x\ 25\% = \$19{,}000$$

(p) Demand 9000

$$Outcome\ Value_{(p)} = (9000\ x\ \$20) - (10{,}000\ x\ \$7) + (1000\ x\ \$2) = \$112{,}000$$

$$EMV_{(p)} = \$112{,}000\ x\ 25\% = \$28{,}000$$

Next are the calculation of nodes (e), (f), (g) and (h).

(e) 5000 T-shirts with 60% probability the team will win

$$Outcome\ Value_{(e)} = EMV_{(i)} + EMV_{(j)} + EMV_{(k)} + EMV_{(l)} = \$56{,}000$$

$$EMV_{(e)} = \$56{,}000\ x\ 60\% = \$33{,}600$$

(f) 5000 T-shirts with 40% probability the team will lose

$$Outcome\ Value_{(f)} = -\ (5{,}000\ x\ \$7) + (5000\ x\ \$2) = -\ \$25{,}000$$

$$EMV_{(f)} = -\$25{,}000\ x\ 40\% = -\$10{,}000$$

(g) 10,000 T-shirts with 60% probability the team will win

$$Outcome\ Value_{(g)} = EMV_{(m)} + EMV_{(n)} + EMV_{(o)} + EMV_{(p)} = \$58{,}000$$

$$EMV_{(g)} = \$58{,}000\ x\ 60\% = \$34{,}800$$

(h) 10,000 T-shirts with 40% probability the team will lose

$$Outcome\ Value_{(h)} = -\ (10{,}000\ x\ \$7) + (10{,}000\ x\ \$2) = -\ \$50{,}000$$

$$EMV_{(h)} = -\$50{,}000\ x\ 40\% = -\$20{,}000$$

Similar procedures are applied for nodes (b), (c) and (d) where the EMV values for these nodes are:

$$Outcome\ Value_{(b)} = \$23{,}600$$

$$Outcome\ Value_{(c)} = \$14{,}800$$

$$Outcome\ Value_{(d)} = \$0$$

Note that for that for node (d); the outcome value is \$0 since if the shopkeeper did not do anything in the event, hence no bought and sold T-shirts.

Looking at the outcome values of node (b), (c) and (d) the highest outcome value is for (b) hence used for the root value situated at node (a). As a conclusion, looking at the t-shirt problem, it can be seen that buy buying 5000 T-shirts will yield the highest outcome value hence is the most probable solution to this problem. The completed tree is as shown in figure 4.5.

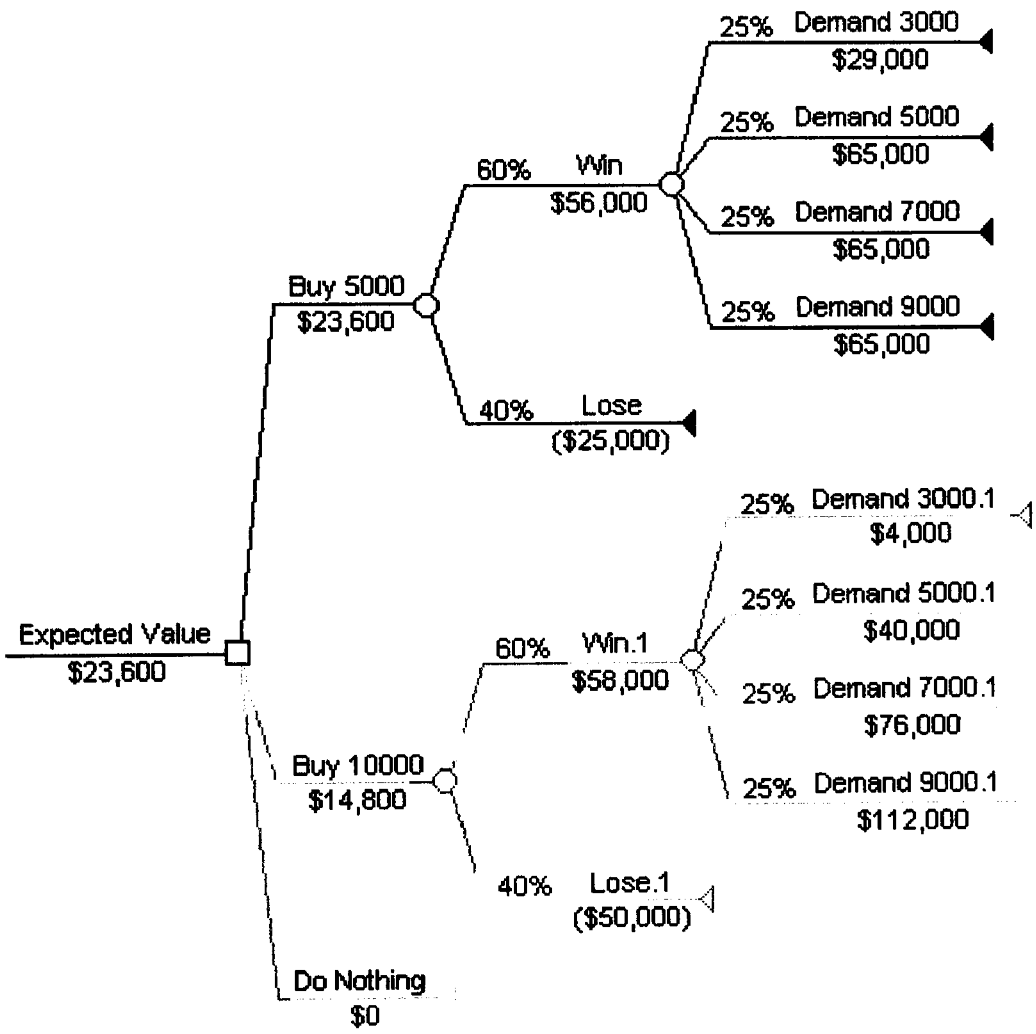


Figure 4.5: Completed tree for the t-shirt seller problem

Comments

In general the EMV decision tree relies on known or calculated results of an outcome. In a complex situation where there are multiple types of data taken into account, and millions of cause and effect using this method proves to be very time consuming in obtaining the root outcome value. Nevertheless, it shows that the resultant outcome is achieved via **recursive calculations** from the branch to the root of the tree. This is an important concept to note as it is used in all variants of DT.

The EMV decision tree has a limited range of operation due to its heavy reliance on probability and the nature of its analysed problem. In analysing large multi-type databases, the probability equations will be complicated. Hence other techniques are used utilising the same concept of recursive calculations in obtaining the final result. Its differences are the direction of the recursive calculations and in the equations used for partitioning and the information scoring techniques used.

To understand more on the concept of the next variants of the decision tree, an understanding of the information scoring and relevant statistical equations are required.

4.6 Attributes

Attributes are actually raw data to be analysed. Its representation is dependent upon the type of database and available data. As an example, in a medical database, attributes may consist of patient statistics and symptoms of diseases where else from a power systems point of view, attributes can be in a form of voltage levels and power flows.

Observing these attributes, some may be significant in an analysis especially those which that lie in the boundary of a stable and unstable state. In a decision tree analysis, determining these significant attributes is the underlying key in determining the direction and accuracy of the decision tree.

An attribute is known to be most significant when its value can be used to separate the overall data into 2 distinct groups of outcome. For example, looking into the boiling point of water (which is 100°C), the main concern is to group water states in terms of temperature for liquid and steam. Considering a list of water temperature values ranging from 0°C to 110°C, 100°C attribute value is the most significant attribute for separating water temperature between steam and liquid.

For more complex situations, the most significant attribute value can be found using a statistical approach. All attributes are tested for its significance in partitioning. This is known as **Information Gain.**

Taking the most significant attribute as a reference, descendent root nodes are created. Descendent nodes represent possible values or outcomes. This process is then repeated until no further partitioning is needed.

4.7 Information gain [8]

To define Information Gain, there must be a definition of **Entropy**. Statistics define Entropy as a value of impurity in a sample collection. It can be expressed in terms of equation (2) as shown:

$$\text{Entropy (S)} \equiv - p_+ \log_2 p_+ - p_- \log_2 p_- \qquad (4.3)$$

where:

P_+ = ratio of positive samples over the total samples.

P_- = ratio of negative samples over the total samples.

Note: Entropy = 0 when the samples are either all positive or negative.

Entropy = 1 when positive samples equals negative samples (i.e. $P_+ = P_-$)

Setting P- as a constant value and increasing P+ from zero to the same amount of P-, an entropy function plot can be made showing the characteristics of Entropy from P+ point of view as shown below:

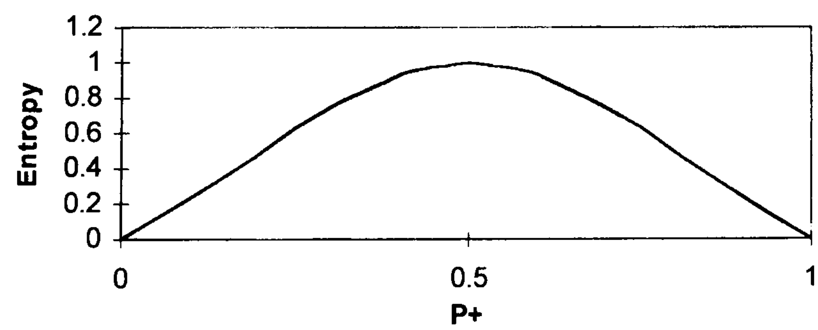


Figure 4.6: Entropy Vs P+ graph with constant P- and increasing P+

Figure 4.6 shows that entropy will highlight pure positive and negative areas by labelling it with an entropy value of 0 where purely P- areas and P+ areas are labelled as 0. A combination of P+ and P- values will result in an entropy value to increase and decrease dependent upon the concentration of P+ and P-.

Recalling again, Entropy is a measure of impurity in a collection of training examples. From here, we can calculate the Gain:

$$Gain\,(S, A) = Entropy(s) \; - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \qquad (4.4)$$

Where:

A = a set of possible values for attribute A.

S_v = subset for S for which attribute A has value v.

The first term in the equation describes the entropy formulae. The second term is defined as the expected entropy after S is partitioned using attribute A. Hence gain is defined as the reduction of entropy due to attribute A. High information gain indicates that the attribute element used is suited for a partitioning limit. This concept is applied into the next variant of DT, which is the ID3 as explained in the following section.

4.8 ID3 decision tree [8, 13]

Known as Inductive Biased Decision Tree or Interactive Dichotomizer 3 (ID3 in short), it was first realised by Quinlan [8]. Its partitioning approach is based upon the Information Gain and Entropy concepts as explained earlier. The ID3 algorithm goes by the following steps:

1) Get training set for a particular decision problem including possible attributes that may effect the decision.
2) Use Information Gain formulae (ref. Eq. 4.4) to determine the first root element and take the highest gain available.

3) From the root element, extend several branches from it. The number of branches possible is dependent upon the possible situations of the selected attribute. In this case it is the root attribute.
4) For each branch, use the Gain formulae again to determine the next root value. And the process is repeated again until all of the attributes are selected. Note that each attribute can only be selected once.

Looking into past analysis and reviews [1, 14], the ID3 approach has advantages and disadvantages. Some of them are as shown:

a. All available attributes in the Decision problem is automatically added into the tree, hence the tree will always be complete.
b. The original ID3 tree has no backtracking properties, which means that any mistakes in partitioning will incur in error on the overall results. Hence there is a possibility that ID3 will not converge to an optimal solution.
c. The tree result is based upon the dataset used during the construction of the ID3; hence any unknown test data may have a high chance of misclassification.

Further improvements are made to the basic ID3 structure by adding an Inductive Bias element as shown in the next section.

4.8.1 Inductive Bias element in ID3

Inductive Bias is defined as a set of assumptions that comes together with the training data. It helps the tree-building algorithm to deduce future possibilities hence optimising final decision tree. Possible advantages of implementing Inductive Bias elements are as below:

(a) **Shorter trees**

Since inductive bias would give a more controlled partition, hence over partitioning can be avoided giving a shorter tree with good generality properties.

(b) **Higher information gain**

Introducing an Inductive Bias element would give a better representation of the Information Gain result. Obtaining this, the right attribute will be chosen as the most significant attribute hence the partitioning limit.

The ID3 is then further improved by including Inductive Bias element such as the Stop Split Criterion leads to the creation of the C4.5 tree, as explained in Example 3 (Section 4.3.5).

4.9 C4.5 decision tree [13, 14]

C4.5 decision tree is an improvement on Quinlan's ID3 algorithm. Its basic principles are based upon the 'Divide and Conquer' approach. At each test node, the data set must be partitioned until the stopping criterion is met, the node will be labelled as the leaf node. In other words, data are partitioned into sets until there are no more data to be partitioned or when the stopping criterion is met. To partition the data set, C4.5 uses a **Splitting Criterion**.

4.9.1 Splitting Criterion

In general, the splitting criterion is defined as the highest gain between the data available, where as the stopping criterion is defined as zero or minimum gain. In this case, gain is defined as below [4, 14]:

$$Gain(D,T) = Info(D) - \sum_{i=1}^{k} \frac{|D_i|}{|D|} \times Info(D_i) \qquad (4.5)$$

Where information gain (**Info(D)**) is defined as below:

$$Info(D) = -\sum_{j=1}^{C} p(D,j) \times \log_2(p(D,j)) \qquad (4.6)$$

Where:

C - number of classes or attributes in the data set.

p(D,j) - the proportion of cases in D that belong to the jth class.

D - attribute set.

D_i - Value of D at the ith class.

Note that information gain in Eq. 4.6 denotes the residual uncertainty about the class at which a case in D belongs to. It can be seen that the above formulas (Eq. 4.5 and Eq. 4.6) has similarities with the formulation concept used in the ID3 algorithm (Eq. 4.3 and Eq. 4.4). The similarities can be seen where entropy is represented in terms of information gain. If there is enough Gain difference, between entropies, there will be further partitioning. If not, when Gain difference is zero, the partitioning stops.

The problem comes in when the C4.5 approach is applied to handle noisy data sets, i.e. the data may have error associated with it or the data is incomplete. Note that noisy data leads to a complex and inaccurate tree hence solutions have been introduced to reduce the complexity of the tree via a pruning algorithm as explained earlier.

The solution proposed by Quinlan is by adjusting the Minimum Description Length (MDL) of the data sets [14]. In simpler terms, an adjustment is made to the apparent information gain of attributes of the database. MDL theory states that the best theory minimises the sum of the theory[1] and exception[2] cost. In other words, MDL principle requires that the amount of information needed must be as minimal as possible but sufficient for generalisation. Hence it gives a trade off between accuracy and simplicity. The trick is utilising MDL effects to identify a noisy data situation. Comparisons and tests made by Quinlan yields a more structured tree, which is smaller and gives better pruning results with lower error levels.

Comments:

The C4.5 algorithm has the same general approach to the ID3 algorithm. It is in fact an improvement of the ID3 algorithm in dealing with error or 'noisy' attributes. Looking at the C4.5, ID3 and the EMV tree approach, there are similarities seen in terms of its general tree building concepts. They are as follows:

1. **Recursive partitioning**

   All 3 approaches use the recursive partitioning method.

2. **Gain and Split Criterion**

   Each of the tree building method uses a form of gain ratio in choosing the best attribute or most significant for partitioning. A split criterion can be seen evident in the C4.5 and ID3 approaches where it utilises the gain results as a yardstick.

These similarities are also evident in other variants of DTs and FDTs that will be explained later.

[1] Theoretical cost can be understood as the amount of complexity and computing time required to encode or represent the data set in a form to be used in the decision tree.

[2] Exception cost can be understood by the amount of computing time required to rectify and misrepresentation of data set after data classification.

The next variant of decision tree is known as the Top Down Induction Decision Tree (TDIDT). TDIDT forms the basis of the Fuzzy Decision Tree application to power systems as introduced by Louis Wehenkel and Xavier P Boyen [2-4, 6].

4.10 Top Down Decision Tree (TDIDT) [2]

The TDIDT shares the same objective of any decision tree, which is classification but may differ to other methods in terms of its approach. The basic TDIDT algorithm works on a 'successive refinement approach'. This means that at each stage, the nodes are tested repetitively for its quality each time with a new set of quality values. The general steps are as shown below:

1. Create a node (test node) and read the learning set to this node
2. Compute its quality measure or purity measure tests and apply the stop split rule. If the stop rule applies, leave the node as a leaf node. If the split rule applies, further splitting will occur (step 3).
3. Splitting occurs by using the approach below:
    a. Using an optimal splitting rule, the best attribute is selected for splitting the current node.
    b. Using the best attribute to split the set into 2 new subsets.
    c. Apply step 1 into the new subsets created.

From the general procedure listed above, there are several terms that needed some explanation. The first term is the **stop split rule**. This checks if there should be any further partitioning required. This is critical in creating a tree with less over-fitting problem, i.e. decision tree which has lost its generality but accurate in terms of its classification.

The second term is the **optimal splitting rule** which defines the criterion and search procedure in choosing the best attribute as a split threshold for the test node. This can also be important, as the selected attribute will effect the direction of the decision tree. The best way to see how TDIDT works is via a working example as illustrated.

4.10.1 Example of TDIDT Application [3, 5, 15]

In **Voltage stability** analysis, one of the main attributes that can be used to differentiate between stability and instability is the **Critical Clearing Time** (CCT). CCT denotes the maximum time allowed before the line must be put out before instability and machine or component damage occurs.

A database of possible P and Q are presented based upon the scenarios applied to the power system. From there, the corresponding CCT values are calculated. The CCT values are used to basically classify the general data into 2 states (stable and unstable regions).

From the CCT values, an information scoring technique is applied to determine the optimal value of P and Q which will classify the boundary between a stable and an unstable region. The corresponding optimal values are then used to classify the data for the next learning set test where the same previous method is repeated. This is known as **recursive partitioning**. An example of the finished DT is as shown in figure 4.7 [3].

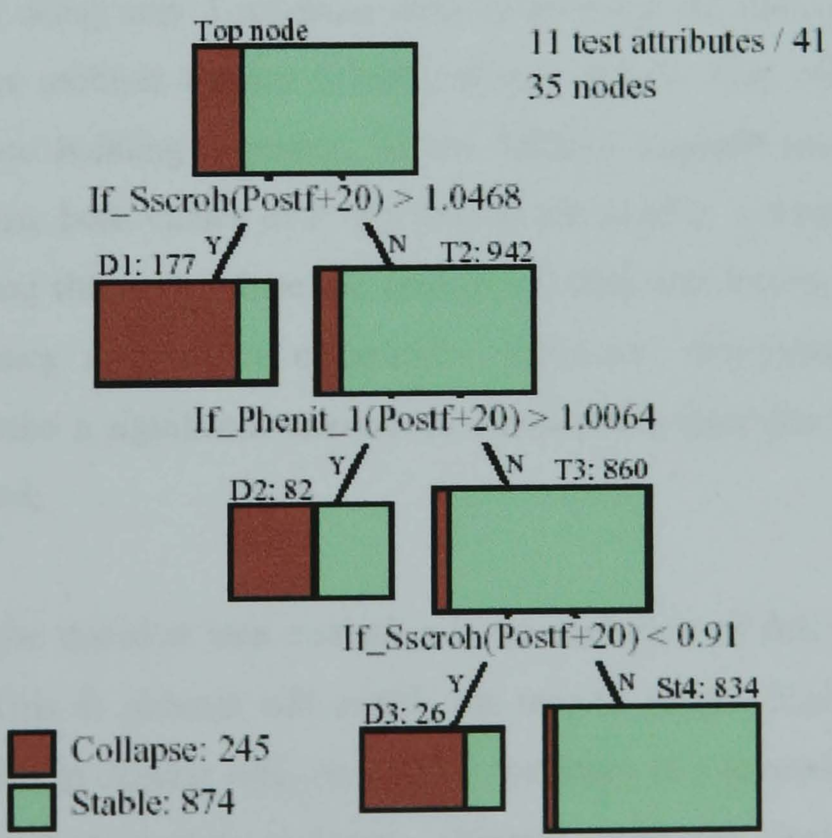


Figure 4.7 An example of a partial tree from a TDIDT results [3]

Figure 4.7 shows an example of a TDIDT for voltage collapse analysis using critical clearing time settings used for the French grid system (EDF)[3]. The overall result produced several stability and instability classifications. The finished tree was tested with an actual scenario test for its classification accuracy. From the results obtained [3], it can be seen that the TDIDT is fairly accurate with a non-detection rate of 6% and a false alarm rate of 19%.

Comments

The TDIDT has a very interesting approach in terms of its successive refinement methods. This means that each attribute is re-evaluated, giving a much better purity analysis for the splitting exercise. Its application to a power system problem has given an incentive of using this general approach to the Voltage Collapse problem discussed.

4.11 Oblique decision tree

It is a variant of the TDIDT, with added improvement in terms of its optimal split values. Rather than using only 1 attribute class to partition the dataset, the oblique decision tree utilises multiple classes of optimal split values. This will increase the efficiency of the tree building sequence. In the TDIDT example mentioned, apart from the CCT values, both values of P and also Q are used as a threshold limit for partitioning. Applying this will reduce the amount of dead end leaves, which in turn will improve accuracy and reduce complexity. However, determining a multiple attribute limit will take a significant amount of computation time due to the amount of data to be analysed.

The next stage of the decision tree evolution is the inclusion of fuzzy element into the decision tree. This in general will enable the tree to handle fuzzy data values, which may be suitable in dealing with probability instances in a scenario. Rather than obtaining a fixed representation, a leaves obtained from the decision tree will represent a fuzzy gradient on how severe a decision may give to the overall picture.

4.12 Fuzzy decision tree

The crisp decision trees are suitable in partitioning discrete data. In situations where the data sets involved does not show much extreme in terms of its value the data set is known to be continuous. Hence the usage of discrete decision trees would result to an approximate and coarse tree. Unlike crisp decision trees, fuzzy decision trees are able to handle continuous and numerical data with less possibility of dead end nodes, which could occur in a crisp decision trees.

In general, a fuzzy decision tree can be built using the following steps [4]:

- **Partitioning**

  An algorithm is used to classify the set of data into groups/sets of data. To obtain this a 'Discrimination Test' is made.

- **Scoring**

  The partitioned data is now tested in terms of the quality of the data at the particular node. This is called the 'Quality Measure' test.

- **Splitting**

  When the best data is found, the set of the node is then split accordingly into several other subsets.

- **Termination**

  Before a new test node is created, the current data is tested with a series of Stop Splitting tests. This will determine whether the data need to be split further or not.

- **Labelling**

  Once stop split criterion are met, the node is now known as a leaf node. Hence a label is computed for the particular leaf.

- **Pruning**

  This is done to reduce the complexity of the tree by collapsing irrelevant branches without affecting the accuracy of the tree structure.

The following shows several examples of Fuzzy Decision Trees (FDT).

4.13 Fuzzy ID3 [4, 16]

Proposed by Motohide Umano [16], this is one of the first effort in utilising Fuzzy properties into a Decision Tree by applying fuzzy principles to Quinlan's ID3. Attribute selection are made via ranking and score measurement method. Like an ID3 algorithm, it uses a fuzzy version of Information gain in selecting attribute for the test node.

$$IG[v] \equiv H_N - \sum_{s=1}^{\#subnodes} \frac{\sum_x \mu_s[x]}{\sum_x \mu_N[x]} H_S \qquad (4.7)$$

where:

$$\mu_S = \mu_N[x]v[x;s] \quad and \quad \sum_{s=1}^{\#subnodes} \mu_s[x] = \mu_N[x]$$

Referring to Eq. 4.7, the information gain formulae are very similar to the Eq. 4.5 where H_N and H_S represent the entropy value for the test set (H_N) and the partitioned set (H_S). Instead of using the standard entropy formulation, this approach uses a fuzzified version of entropy ratio v. v represents the difference of entropy content at the node and the weighted sum of the successor node should splitting be made.

Stop splitting criteria

3 criteria are tested and if anyone of it is met, splitting of the particular node is stopped. The first two criteria deal with the **purity or cardinality of learning set**. If it is below the threshold level, splitting is stopped. These criteria are represented as equations 4.8 and 4.9.

$$\frac{\sum_{x} \mu LS_{node}\,(partitioned)}{\sum_{x} \mu LS_{node}} \geq TH_1 \qquad (4.8)$$

$$\sum_{x} \mu LS_{node} \leq TH_2 \qquad (4.9)$$

For Eq. 4.8, the nominator represents the weighted sum of the partitioned data set or learning set whereas the denominator represents the weighted sum of the total learning set. The ratio of both of these must not exceed a threshold value TH1, predetermined by the user. For Eq. 4.9, the total weighted sum of the partitioned must not exceed TH2, a predetermined threshold value. The algorithm also has a third criterion which checks for **attribute availability** hence preventing further partitioning.

Label assignment

There are 3 basic approaches:

1. 'Winner takes all', where the best classification attribute are used as the label.
2. 'Restricted winner takes all' is selected when an over mixed leaf is encountered.
3. 'Weighted mean approach', i.e. each leaf is labelled with its own average membership grade.

Discussion

This method still relies on the user expert knowledge to determine the partitioning which is the main core of the tree building algorithm. The data values used also must be perfect and complete. This is because it does not have any backup algorithm to compute imperfect data sets. Further improvements have been made to this FDT by adding in a new approach in assessing entropy and information gain during the tree building process [16]. But generally, its approach does follow the basic DT approach of recursive partitioning when compared with its non-fuzzy counterparts explained earlier.

4.14 Fuzzy Induction of Decision Trees with Missing Values (FIDMV) [1, 4]

Very similar to Fuzzy ID3, this algorithm is proposed by Cezary Z Janikow [1] where the algorithm takes into account missing values situations. Apart from that the defuzzification formulae used is different. The FIDMV uses the centre of gravity method for defuzzification:

$$\text{output}[x] = \frac{\sum_{l=1}^{\#\text{leaves}} \mu_l[x]\text{mass}_l[x]\text{label}_l[x]}{\sum_{l=1}^{\#\text{leaves}} \mu_l[x]\text{mass}_l[x]} \qquad (4.10)$$

where:

μ - fuzzy membership value

label - crisp representation value attached to the fuzzy membership value

mass - the importance weighting factor of the attribute in creating partition

#leaves - total attribute count

Referring to Eq. 4.10, the defuzzification formula can be seen to be almost similar to the standard fuzzy centre of gravity formulation with an addition of the variable mass[x] added into the equation. The effect of adding this variable is to simplify its defuzzification calculation as explained by Janikow [1].

Dealing with missing values

During the learning period, the tree disregards any instances, which possess any missing values. If by chance the missing value instances are met, the attributes are split evenly between the two successive nodes. During pruning, if the missing value attribute prevents proper evaluation of the tree, that particular attribute is labelled as void.

Discussion

Indeed from the literature, FIDMV has the advantage of dealing with missing values, but it still relies on the user expert knowledge to do the partitioning step. But it seems that the method at which it deals with missing values is that it totally disregards the particular attribute and not used it at all. All it does is it carries the attribute further down the tree, splitting it evenly across the branches. The question which may be asked is what if the missing attribute is the critical attribute for accurate classification. One suggestion is to recalculate the missing attribute values. This is possible in power system analysis with the help of already available software packages to analyse and test the power system integrity.

The following FDT algorithms are the actual practical ones to be applied for reason of power system analysis. Note that both tree types uses the same stop-split criterion and Unseen instances algorithm.

4.15 Outbound Fuzzy Decision Tree (OFDT) [4, 15]

The basic algorithm steps are similar to the General Form of the Decision tree noted earlier. The difference will be in the formulae used in the OFDT algorithm. Compared to previous fuzzy decision tree, the OFDT has a contrast discriminator included in the algorithm.

Outbound Fuzzy memberships

The fuzzy membership used in the OFDT is very unique due to its application towards critical clearing time assessment towards the stability of the power system. Critical clearing time values has 2 overall effect to the system which is either the system stays stable or goes towards instability. Hence looking at possible conclusions available is either stable or unstable. In terms of a crisp situation the decision of stable and unstable can be shown in figure 4.8.

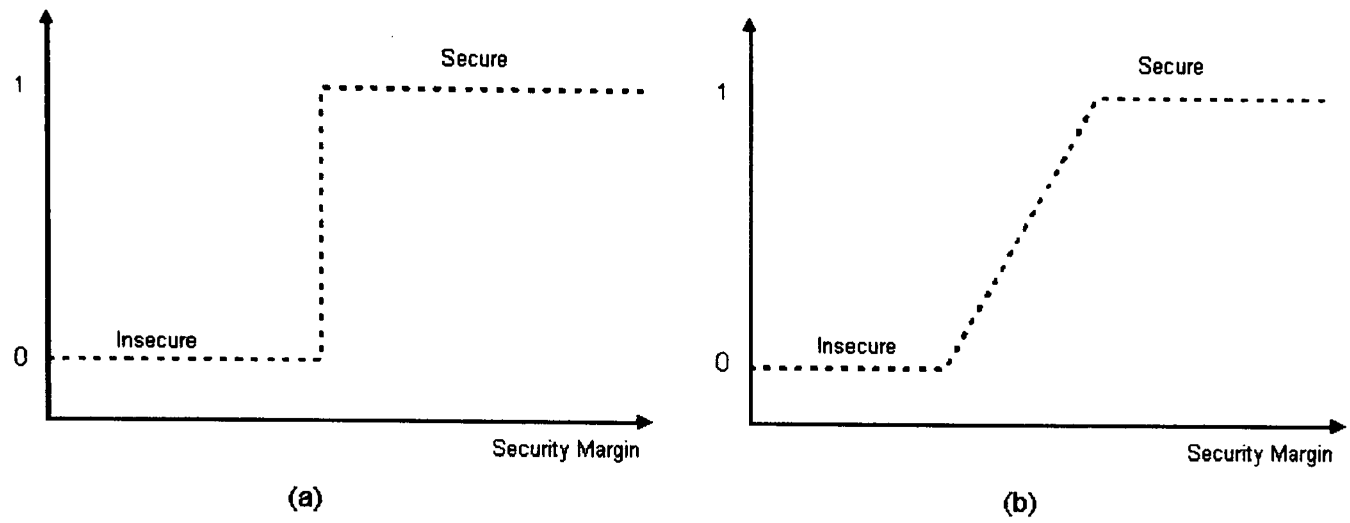


Figure 4.8: Security states (a) in a crisp situation (b) in a fuzzy situation

Figure 4.8(a) shows the interpretation of a crisp decision ('yes' or 'no' situation) where the transition between insecure and secure situations is very sharp. Whereas figure 4.8(b) shows a more gradual transition from insecure to insecure for fuzzy decision interpretations. Since the OFDT has as an objective of a smooth partition or decision the fuzzy profile shown in figure 4.8(b) is made as the general shape of the discrimination values. Note that in this matter, by using standard fuzzy concepts, the fuzzy membership function and the discrimination function are 2 separate entities. The discrimination function acts as a biasing effect in the partitioning sequence and the membership functions generates fuzzy membership values of an instance. Hence in a typical fuzzy decision tree, these membership values are then partitioned in parallel from the root node down to the last leaf nodes of the DT. This in effect can be computationally complicated and may take a lot of computing time to calculate the final DT results.

The OFDT opt for a simpler assumption by assuming the discrimination function and the fuzzy membership function as one entity during the CCT evaluation example [15].

Discriminator function

$$\nu[x;\alpha;\beta;\gamma;\eta] = 0.5\left(1+\gamma\frac{1-e^{-4\alpha(x-\beta)}}{1+e^{-4\alpha(x-\beta)}}\right) \tag{4.11}$$

where:

α - normalised slope or crispness

β - midpoint or threshold

γ - scale factor with respect to the median line (0.5)

η - attribute index

x – attribute value

The discrimination function acts as a weighting factor during attribute splitting as explained earlier. The OFDT discrimination has overshoot properties. This is controlled by the term γ. Maximum value recommended in the literature is 2 [4]. An example profile of the discriminator function is as shown:

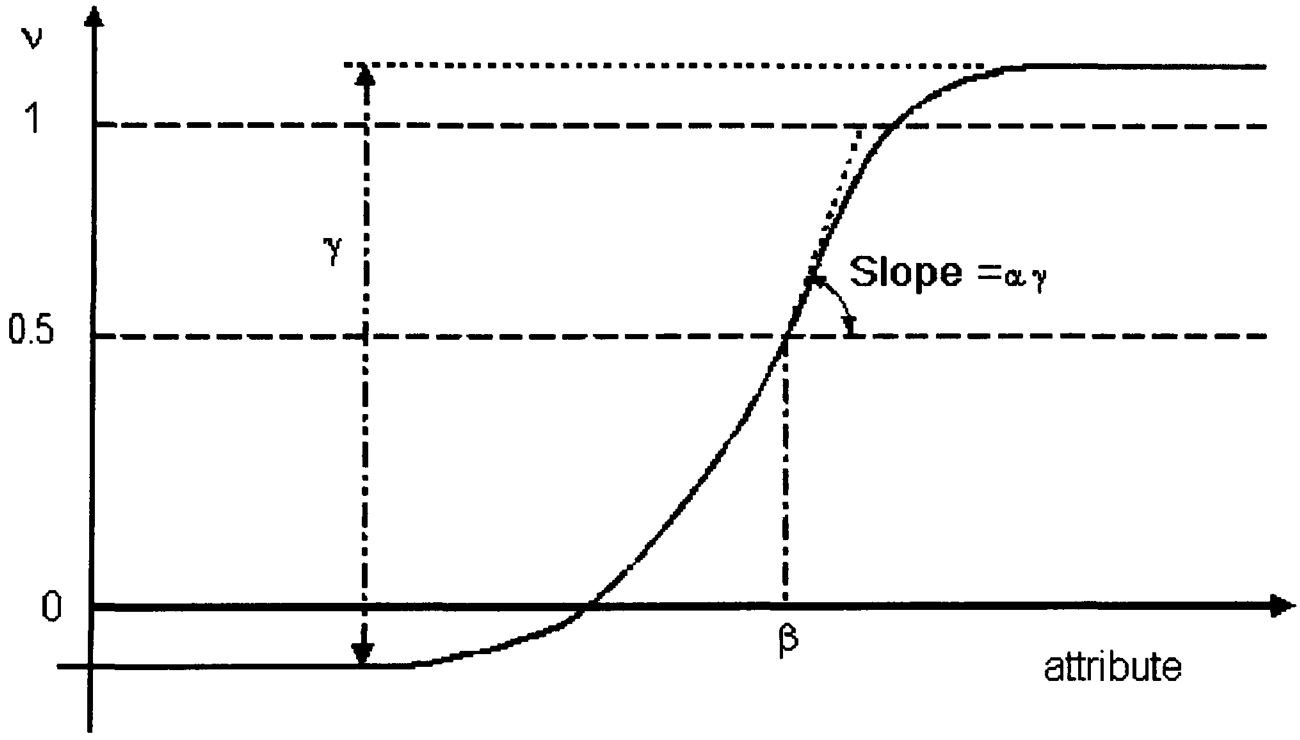


Figure 4.9: OFDT discriminator function profile [4]

The reason of the overshoot properties is to give the OFDT the ability to deal with undefined maximum differences. Say for example at the difference between hot and cold, there is a distinct difference between them. But if a comparison is made between cold, hot and extremely hot, there is a problem in differentiating between hot and extremely hot if the discrimination function are bound between hot and cold. By giving overshoots to the discrimination function, there is an extension on both extreme conditions to compensate with possible undefined extremes. To give an additional control or biasing to the DT in half black half white situation, a contrast element is added.

Contrast

$$Contrast[v, x] = |v[x] - (1 - v[x])| \tag{4.12}$$

where:

ν – Fuzzy discriminator value

x – Fuzzy attribute value

This formula maximises in situations of half black half white situations where the discriminator value is 0.5. This can be illustrated by the graph shown in figure 4.10.

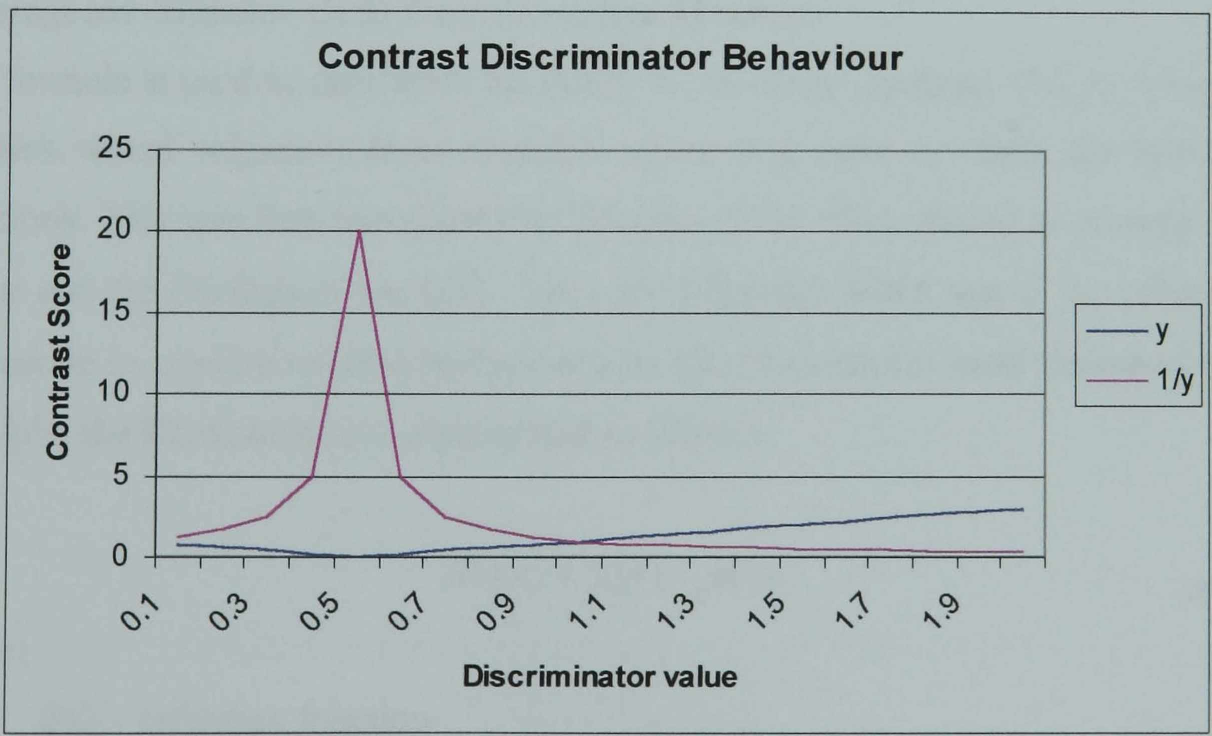


Figure 4.10: A graph showing the contrast discriminator behaviour with varying discriminator values

From figure 4.10, it can be seen that the contrast score goes to a maximum when the formula is inverted. Mathematically, the score when $\nu = 0.5$ should be infinite, but for illustration and scaling purposes, a value of 20 is selected assuming that the value 20 is considered to be significantly high compared to the other scores. Multiplying the inverted values of the contrast score is useful in drawing a finite line between attributes that has a discriminator value near the 0.5 border.

Normalisation Constraint

$$\sqrt{\left\langle |2\nu[x] - 1|^2 \right\rangle_{LS_{Node}}} = 1 \qquad (4.13)$$

The equation above means that the RMS (Root Mean Square) contrast of ν over each node learning set is unity. The reason for the equation is to ensure that the left and right branching will not overlap each other with respect to the average on the overall learning set [4].

Kolmogorov-Smirnov (KS) Discrimination Measure

This formula is used to determine the purity of the tested attribute. The Kolmogorov-Smirnov tested originates from statistics where it is used to check for best of fit situations. This was first introduced by Friedman [17]. Very similar in concept to the χ^2 test and the Darlington test [12]. The only difference of KS test to the others is its application to continuous distributions and its effectiveness for small datasets [18]. In principle, the KS distance are represented as follows:

$$KS(x) \equiv f(x) - g(x) \qquad (4.14)$$

where:

f(x) – reference function

g(x) – test function

x – attribute instance

In essence, the KS score represents the maximum distance of corresponding instances between a reference function and the test function. This can be easily seen in the illustration shown:

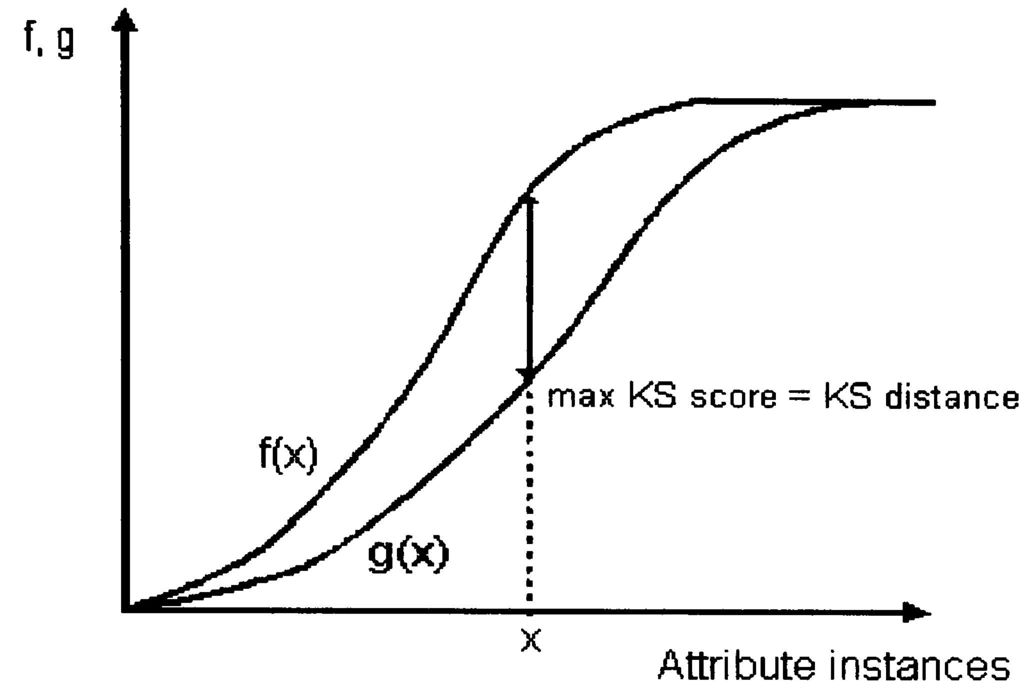


Figure 4.11: KS distance between a reference function and a test function

For the OFDT algorithm, the KS measure is defined as in Eq. 4.15 shown:

$$KS\left[v_N \mid \vartheta=\mu_c\right] \equiv \left|\langle v_N\rangle\, S_{N\cap C} - \langle v_N\rangle\, S_{N\cap\bar{C}}\right| \tag{4.15}$$

where:

S_N	- Structuring Set at node N
v_N	- discrimination function at node N
c	- target class
μ_c	- membership function of C
ϑ	- target function

also < f > is known as the weighted average over the elements of a fuzzy sets:

$$\langle f\rangle_S \equiv \frac{\sum_x \mu_S[x]\, f[x]}{\sum_x \mu_S[x]} \tag{4.16}$$

Leaf labelling

The OFDT uses a conventional weighted mean (or also known as weighted average) approach as shown in Eq. 4.16.

4.16 Fussy Fuzzy Decision Tree (FFDT) [2, 4, 10]

The FFDT is very similar in terms of concept compared to the OFDT. The differences are seen in terms certain formulation in the DT building sequence. The first differences is in terms of its discriminator function.

Discriminator function.

$$v[x, \alpha, \beta, \eta] = \begin{cases} 0 \text{ if } \alpha(x_\eta - \beta) \leq -0.5 \\ 0 \text{ if } \alpha(x_\eta - \beta) \geq +0.5 \\ 0.5 + \alpha(x_\eta - \beta) \text{ if otherwise} \end{cases} \tag{4.17}$$

where:

α - sharpness of transition region

β - midpoint of transition region

η - attribute index

The function of the discriminator function has the same objective as in the OFDT apart from the profile shape of the discriminator function shown in figure 4.12 . In the FFDT, the discriminator function is as shown in Eq. 4.17. Compared to Eq. 4.11, Eq. 4.17 has 3 distinct discriminate values with no local overshoot properties whereas the latter has a smoother discriminator values denoted by figure 4.9.

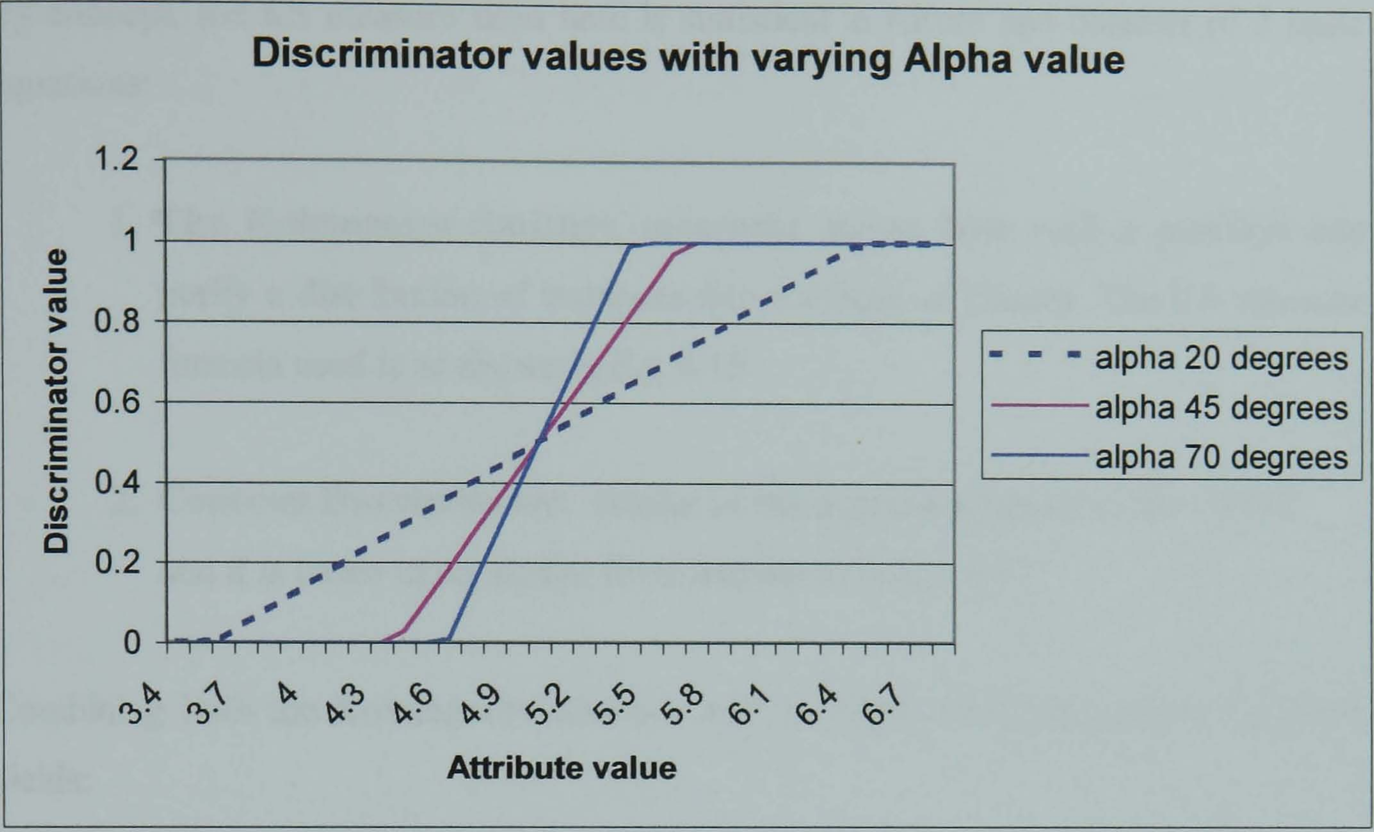


Figure 4.12: FFDT discriminator function profile using varying attribute values and midpoint transition setting of 5.0

Referring to figure 4.12, by setting the midpoint transition to be at 5.0, the corresponding discrimination value at all three profiles is 0.5. As for the alpha values, it is shown that it reflects the angle of the discrimination template. The combination of the midpoint transition value and alpha denotes the coverage of the discrimination template. These criteria will be expanded in the researched algorithm in chapter 5. Note that all the 3 attributes are determined by the FFDT user to maximise the quality of measure

Information quality discriminator

This measure will test the purity of the attribute tested for classification/splitting. The term used in FFDT is the **Kolmogorov-Smirnov quality measure (KS measure)**. Since the KS measure selects the most significant attribute for partitioning, it acts like a post pruning sequence like in the OFDT. By optimising the KS measure, the overall pruning time required for the tree is minimised. The KS measure is very similar in nature compared to the KS measure used in the OFDT.

By concept, the KS measure used here is statistical in nature and consists of 2 basic equations:

1. **The Kolmogorov-Smirnov measure:** shows how well a partition can purify a distribution of instances into a couple of classes. The KS measure formula used is as shown in Eq. 4.15.

2. **Contrast Discriminator:** similar to the contrast measure in the OFDT and it is based upon similar formulations as in Eq. 4.12.

Combining both the Kolmogorov-Smirnov and the Contrast Discriminator Equation yields:

$$B_{\xi}\left[v_N \mid \vartheta = \mu_C\right] \equiv \frac{\sqrt[\xi]{2} \left| \langle v_N \rangle_{S_{N \cap C}} - \langle v_N \rangle_{S_{N \cap \bar{C}}} \right|}{\sqrt[\xi]{\left\langle \left|2v_N - 1\right|^{\xi} \right\rangle_{S_{N \cap C}} + \left\langle \left|2v_N - 1\right|^{\xi} \right\rangle_{S_{N \cap \bar{C}}}}} \qquad (4.18)$$

ξ is a parameter which determines the fuzziness required. The larger the value the more crisp the results. According to tests made, the best value is 2 [2-6, 10, 15]. By dividing the Kolmogorov-Smirnov measure by the contrast Discriminator the following advantages can be obtained [10]:

- Stops sharp transitions and gives a natural way to select the amount of fuzziness. For crisp attributes, B_x will be maximum.
- Auto centering when $v = 0.5$ by favouring discriminators that minimise $|v[x] - 0.5|$
- The measure of B_x is independent compared to the scale factors of the discriminators.

Other features that can be obtained by B_x are:

- The contrasts are averaged separately to maintain symmetry and compensate the differences between C and (1 - C).
- It has a regularisation parameter (ξ) to determine how fuzzy we want the system to be.
- Able to distinguish poorly partitioned classes.

Stop Splitting Criteria [4]

The FFDT uses a specific method as its Stop Splitting Criteria. In general, the criterions used are as shown:

a) Test node already at the pre-specified depth.
b) Under-population in either of the sub node sets, i.e. not enough data sets to continue splitting.
c) The node would be able to exhibit good enough classification characteristics compared to the ones when the node is split further (a look ahead feature). This can be evaluated by the following equations:

$$\sqrt[ord]{\left\langle \left| \tilde{\vartheta}_{node}[x] - \vartheta[x] \right|^{ord} \right\rangle_{LS_{node}}} \leq tol \qquad (4.19)$$

where:

ord - stop split parameter

tol - stop split parameter

$\tilde{\vartheta}_{node}$ - is the equivalent classification function with its 2 successor nodes.

$\vartheta[x]$ - the actual node to be tested.

Equation 4.19 is very similar to the RMS value formulae (refer to Eq. 4.13). By calculating and comparing the RMS values of the test node and the 2 successor nodes, an evaluation of significant partitioning can be made. If the both RMS values do not vary as much, hence over partitioning has occurred.

Pruning characteristics

As stated earlier, pruning is made to optimise the tree structure hence avoiding complex tree and over fitting. What pruning does is that it will scan through the whole tree structure and locate least significant branches, hence reducing the complexity but retain a reasonable amount of overall classification.

Other methods include building several versions of simplified trees and the tree with the lowest error rate will be chosen. Hence what is observed is the standard deviation between the parent tree and the modified tree.

$$\sigma_E \approx \sqrt{\frac{\|LS_{total} \cap C\| . \|LS_{total} \cap \overline{C}\|}{\|LS_{total}\|}} \quad (4.20)$$

The FFDT utilises the KS score combined with the standard deviation formula shown in Eq. 4.18. Hence the FFDT pruning formula gives:

$$\sqrt{\frac{\|LS_{total} \cap C\| . \|LS_{total} \cap \overline{C}\|}{\|LS_{total} \cap C\| + \|LS_{total} \cap \overline{C}\|} . D} \quad (4.21)$$

where:

D - Kolmogorov-Smirnov measure

Note that in Eq. 4.21, the denominator represents the total learning set count as shown in Eq. 4.20.

Classification of unseen instances. [10]

When the algorithm is tested with a test set, there are possibilities that there is an unpredicted instance. Note that in the FFDT, the test set may consist of different attribute sets compared to the learning set. The FFDT will execute recursive fuzzification and defuzzification of the instance. Defuzzification is made using the weighted mean formulae:

$$\tilde{\vartheta}[x] = \frac{\sum_{leaves} \mu_{leaf}[x]\tilde{\vartheta}_{leaf}}{\sum_{leaves} \mu_{leaf}[x]} \tag{4.22}$$

Eq. 4.22 can be simplified (if $\mu = 1$) to the form below:

$$\tilde{\vartheta}[x] = \sum_{leaves} \tilde{\vartheta}_{leaf} \prod_{root}^{leaf} \begin{cases} \nu_{node} & \text{if left branch} \\ 1 - \nu_{node}[x] & \text{if right branch} \end{cases} \tag{4.23}$$

Comments

It can be seen that the OFDT and FFDT follows the basis of TDIDT where both methods follow the 'successive refinement' approach. The inclusion of a fuzzy element makes handling continuous data sets much easier by replacing the attribute with a fuzzified value. The use of KS Measure is a very good choice as comparable to the χ method where the KS approach gives a better representation when dealing with continuous and small datasets [18, 19]. The fact that both methods are tested to analyse CCTs in a power system, makes this approach a favourable direction for the proposed FDT in chapter 5. The stop split criteria used in the OFDT and FFDT are considered to be extensive and thorough to ensure that the tree is general enough without loss of accuracy.

4.17 Conclusion

The explanations of DTs and FDTs give a brief structure with the inclusion of a recursive partitioning action or successive refinement approach to the proposed FDT. Although it is not easy to get a 100% pure leaf set, a stop split criterion is applied to ensure that over partitioning does not occur. The addition of Fuzzy Elements gives the DT an added edge in dealing with continuous data, which is similar to the nature of the proposed FDT application. Looking into the tool used in finding the most significant attribute, the KS score approach is most favoured due to its capability in dealing with continuous data sets. This sets the right mind set to understand the researched algorithm that will be explained further in chapter 5.

4.18 References

1. Janikow, Cezary Z, *Fuzzy Decision Trees: Issues and Methods.* IEEE Transactions on Systems, Man and Cybernetics, 1998. **28**(1), p. 1 - 14.
2. Louis Wehenkel, *Automatic Learning Techniques in Power Systems.* The Kluwer International Series in Engineering and Computer Science. Vol. 429. 1997: Kluwer Academic Publishers, 279 pages. ISBN 0-7923-8068-1.
3. Louis Wehenkel, C Lebrevelec, M Trotignon, J Batut. *A probabilistic approach to the design of Power System protection schemes against blackouts.* in *IFAC-Cigre Symposium on Control of Power Systems and Power Plants.* 1997. Beijing, p. 506 - 511.
4. Xavier P Boyen, *Design of Fuzzy Logic Based Decision Trees Applied to Power System Transient Stability Assessment.* Department of Electrical and Computer Science. University of Liege, Belgium. 1995. Masters Thesis, 133 pages.
5. Xavier P Boyen, Louis Wehenkel, *Automatic Induction of Fuzzy Decision Trees and its Applications to Power System Security Assessment.* International Journal of Soft Computing and Intelligence, 1999. **102**, p. 3 - 19.

6. Louis Wehenkel, I Houben, M Pavella, L Riverin, G Versailles. *Automatic Learning Approaches for On-line transient stability preventive control of the Hydro-Quebec System. Part 1: Decision Tree Approaches.* in *IFAC Symposium on Control of Power Plants and Power Systems.* 1995. Cancun, Mexico, p. 231-236.
7. Hasan, Usama, *Artificial Neural Networks for Voltage Collapse Monitoring.* University of London, Imperial College of Science, Technology and Medicine. 1998. PhD thesis, 181 pages.
8. Mitchell, Tom M, *Machine Learning*: McGraw-Hill Publishers Inc. ISBN 0070428077.
9. Christina Olaru, Louis Wehenkel, *On Neurofuzzy and fuzzy decision trees approaches*, in *Information, Uncertainty and Fusion*, Editor Ronald R. Yager Bernadette Bouchon-Meunier, A. Zadeh. 2000, Kluwer Academic Publishers: Boston, p. 131-145.
10. Xavier P Boyen, Louis Wehenkel. *Automatic Induction of Continuous Decision Trees.* in *Proceedings of Information Processing and Management of Uncertainty in Knowledge-Based Systems.* 1996. Granada, p. 419 - 424.
11. K L Lo, H I Haji Zainal. *The Fuzzy Decision Tree Application to a Voltage Collapse Problem.* in *University Power Engineering Conference (UPEC).* 2001. University of Swansea, Wales.
12. Vanguard, *Decision Tree Analysis.* Vanguard Software Corporation. Internet Article. 2002. http://www.vanguardsw.com/dphelp/dph00097.htm.
13. Slavco Velickov, Dimitri Solomatine. *Predictive Data Mining: Practical Examples.* in *Artificial Intelligence in Civil Engineering. Proceedings of 2nd Joint Workshop.* 2000. Cottbus, Germany, p. 3-19.
14. Quinlan, J R, *Improved used of Continuous Attributes in C4.5.* Journal of Artificial Intelligence Research, 1996. **4**, p. 77 - 90.
15. Xavier P Boyen, Louis Wehenkel. *Fuzzy Decision Tree Induction for Power System Security Assessment.* in *IFAC Symposium on Control of Power Plants and Power Systems.* 1995. Cancun, Mexico, p. 151-156.

16. Isao Hayashi, Toshiyuki Maeda, Motohide Umano, Lakhmi C Jain, *A Formulation of Learning Type Fuzzy ID3.* International Journal of IEEE Transactions on Fuzzy System, 2000.

17. Friedman, Jerome H, *A Recursive Partitioning Decision Rule for Nonparametric Classification.* IEEE Transactions on Computers, 1977, p. 404 - 408.

18. Irwin Miller, John E Freund, *Probability and Statistics for Engineers.* 3rd Edition. 1985: Prentice Hall International Inc. ISBN 0-13-711912-7.

19. Engineering_Statistics_Handbook, *Quantitative Techniques.* Internet Page. 2002. http://www.itl.nist.gov/div898/handbook/eda/section3.

Chapter 5: Fuzzy decision tree development and evolution

5.1 Introduction

It is noted that a deciding factor of selecting FDT as a tool to study voltage collapse has already been discussed in the previous chapter. Summarizing, FDT techniques is suitable to its ability to deal with continuous data related to voltage collapse analysis.

Based on the FDT basics explained in chapter 4, voltage collapse behaviour explained in chapter 2 and 3, this chapter presents the development of the researched algorithm. However, there is a need to include several explanations on the formulations used for the construction of the FDT. Details of the FDT components are expanded and worked out in terms of its behaviour via preliminary tests. These include:

a) Fuzzy template
b) Kolmogorov-Smirnov (KS) approaches as a measure for information quality
c) Partitioning
d) Defuzzification template
e) Multiple partitioning and quality assessment improvement to the FDT

Explanations are presented via working examples. The constructed FDT is tested in terms of its capabilities and effectiveness using the IEEE 300-bus as a test system [1]. Using initial test results as a reference, the developed FDT goes through some changes and modifications. The improved FDT is then tested again, results preceding these tests led to the creation of a hybrid FDT algorithm. The hybrid FDT concept is then tested on the same IEEE test system. The chapter concludes with a discussion on possible FDT improvements which will be further expanded in chapter 6.

5.2 General algorithm structure

It is best to understand the developed FDT technique based upon the general algorithm structure presented below:

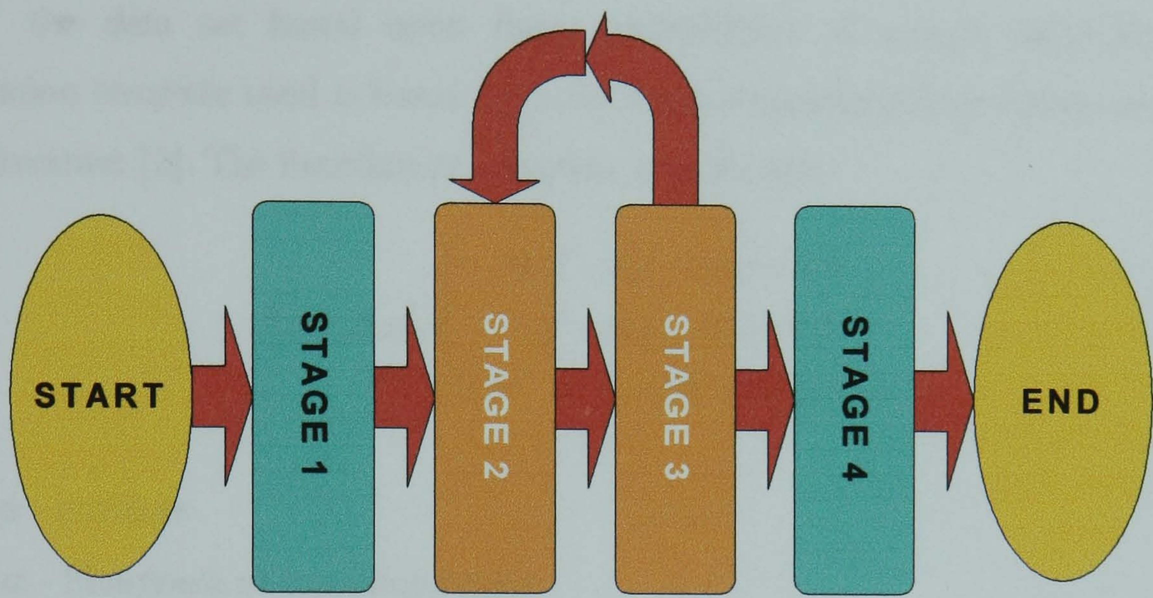


Figure 5.1: FDT program structure

Referring to figure 5.1, the algorithm consist of several stages or modules. Stage 1 represents the data reading and fuzzification. Stage 2 represents the information quality stage, utilising the KS score approach due to reasons that will be explained later in this chapter. Stage 3 is the stop split stage where the algorithm goes through purity check on the validity of future partitioning. Stage 4 completes the FDT process by converting the test nodes into leaf nodes. It is worth noting that the research algorithm is similar in concept compared to the FFDT approaches explained in chapter 4 where recursive partitioning is applied to each test node.

5.3 Stage 1: Data reading and fuzzification

Stage 1 involves converting the crisp data values into fuzzy values. Looking into literature and also from preliminary results, conversion into fuzzy values would 'adjust' the data set based upon fuzzy probabilities of system instability. The fuzzification template used is based upon the fuzzy discrimination template proposed in the literature [2]. The fuzzification template is as shown:

$$v[x,\alpha,\beta,\eta] = \begin{cases} 0 \; if \;\; \alpha(x_\eta - \beta) \leq -0.5 \\ 1 \; if \;\; \alpha(x_\eta - \beta) \geq +0.5 \\ 0.5 + \;\; \alpha(x_\eta - \beta) \; if otherwise \end{cases} \qquad (5.1)$$

Where:

x - attribute

α - Sharpness of transition region

β - Midpoint of transition region

η - Attribute index

The discrimination template forces attributes at the higher and lower end of the data range to have a fuzzy value of 1 and 0 respectively. As for the in-between attributes, they will each have a fuzzy value (between 1 and 0) based upon Eq. 5.1. The variables α and β are determined by the FDT user to maximise the quality of measure. Characteristics of the template are as shown in figure 4.12. The following gives the derivation of α and β:

Referring to figure 4.12, α represents the gradient of the template profile, thus it determines the coverage of the data set to be fuzzified β represents the midpoint of the covered region, hence:

$$\beta = \frac{\max - \min}{2} + \min = \frac{\max + \min}{2} \qquad (5.2)$$

where :

max – the maximum crisp data value

min – the minimum crisp data value

Eq. 5.2 gives the algorithm an automatic setting for β for any range of data. As for the value for α, it can be adapted from Eq. 5.1, taking η as the attribute sequence:

$$v = 0.5[\alpha(x - \beta)]$$
$$v - 0.5 = \alpha(x - \beta)$$
$$\alpha = \frac{v - 0.5}{x - \beta} \qquad (5.3)$$

In a full fuzzy distribution, the maximum fuzzy value is 1 and the minimum fuzzy value is 0. Taking either the maximum or minimum fuzzy value obtained from the fuzzified input data, Eq. 5.3 can be simplified to give Eq. 5.4 or Eq 5.5.

$$\alpha = \frac{0.5}{x_{\max} - \beta} \qquad (5.4)$$

$$\alpha = \frac{-0.5}{x_{\min} - \beta} \qquad (5.5)$$

By incorporating Eq. 5.2 and Eq. 5.5 into the algorithm, automatic initialisation of the fuzzification process is possible. Using a series of numbers with an arbitrary starting point, the following shows the profile of the FDT template.

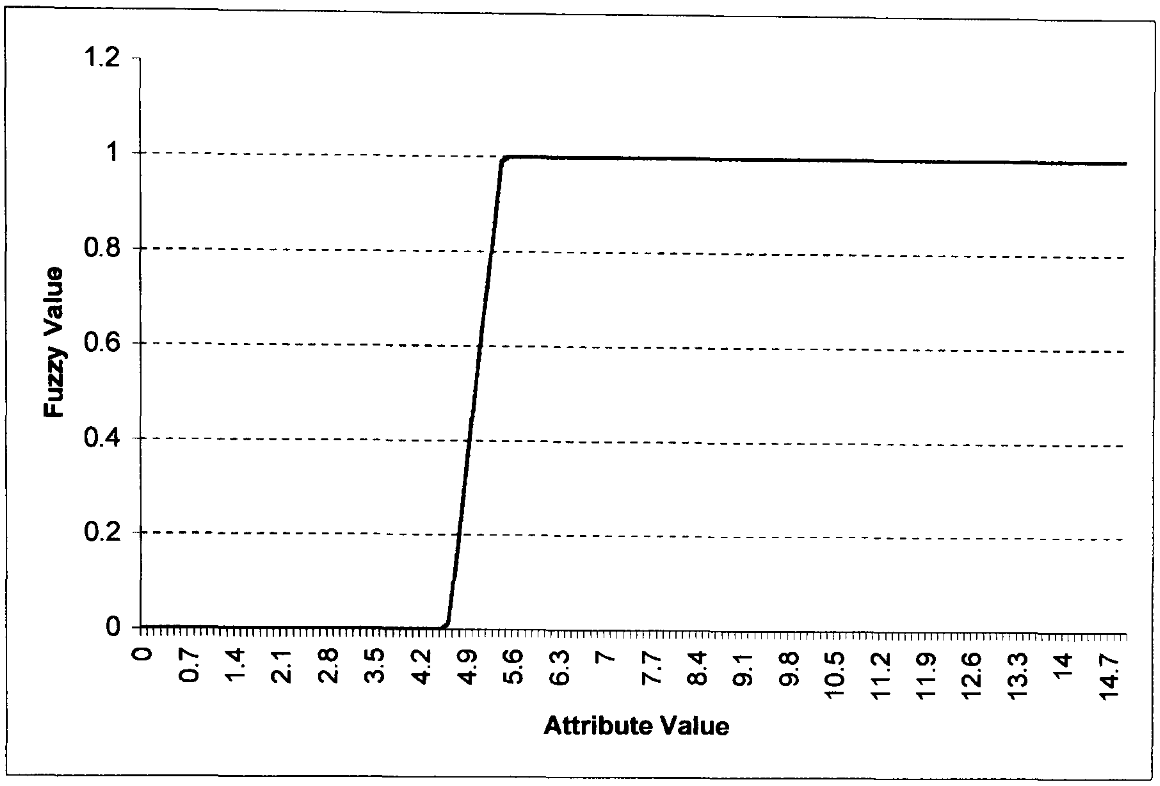


Figure 5.2: An example profile of the fuzzy template

Referring to figure 5.2, the template gives a low fuzzy value for low bus voltage magnitudes and vice versa for high bus voltage magnitudes. Applying this to bus voltage values obtained from a load flow solution, the bus voltage values should spread out between the fuzzy value of 0 and 1 as shown in figure 5.3. In the case of voltage collapse due to slow load increments, certain buses in the network will exhibit a reduction of voltage magnitude as the system load is slowly increased. Therefore, utilising the fuzzy template, changes in the system profile can be presented without losing the overall bus voltage profile shape as all the bus voltage values will be represented as fuzzy values 'stretched' between a fuzzy value of 0 and 1.

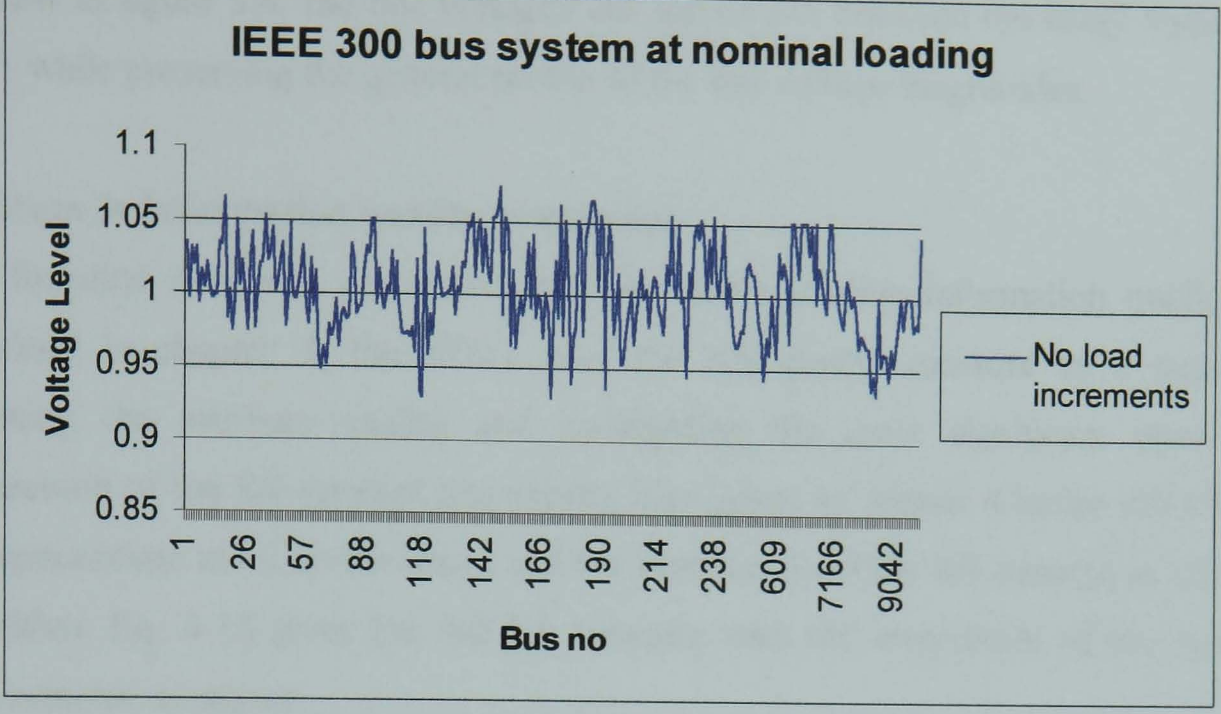


Figure 5.3: A load flow solution of bus voltage values for an IEEE 300 bus system

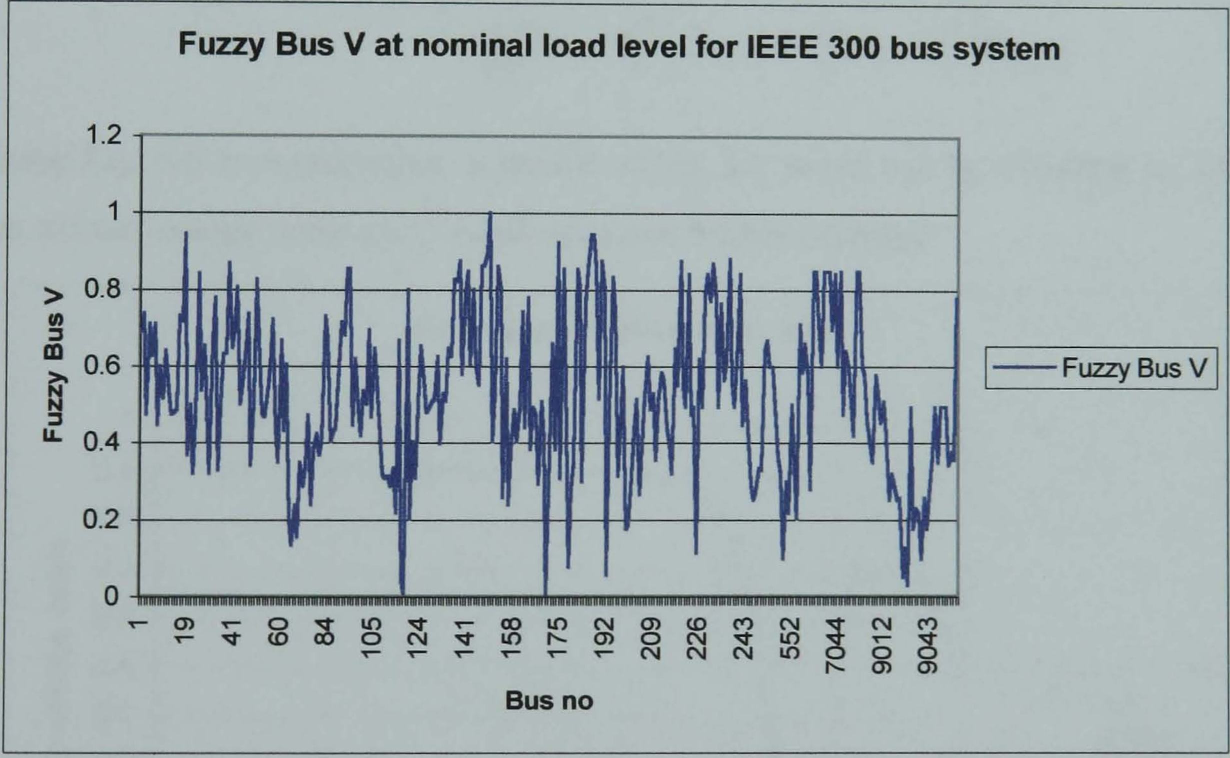


Figure 5.4: A fuzzified version of bus V magnitudes for IEEE 300 bus system at nominal loading factor (load factor 1.0x)

Looking at the bus voltage profile in figure 5.3 and comparing it with figure 5.4, the general shape is similar with a difference of 'spread' in terms of fuzzy bus voltage values. In figure 5.3, the bus voltage varies between 0.929 and 1.073 per unit value whereas in figure 5.4, the bus voltages are spread out between the fuzzy value of 0 and 1 while preserving the general profile of the bus voltage magnitudes.

5.4 Stage 2: Information quality assessment

The fuzzified data sets are tested in terms of its relative information quality. As explained in chapter 4, the FFDT uses the KS quality measure as a means of analysing the attribute quality and highlighting the most significant ones. The explanation of the KS concept has already been given in chapter 4 hence this chapter will concentrate more on the usage and the explanation of the KS concept in the FDT algorithm. Eq. 4.18 gives the full KS measure with the integration of the contrast discriminator as shown:

$$B_{\xi}[v_N \mid \vartheta = \mu_C] = \frac{\sqrt[\xi]{2} \mid \langle v_N \rangle S_{N \cap C} - \langle v_N \rangle S_{N \cap \bar{C}} \mid}{\sqrt[\xi]{\langle |2v_N - 1|^{\xi} \rangle S_{N \cap C} + \langle |2v_N - 1|^{\xi} \rangle S_{N \cap \bar{C}}}} \quad (5.6)$$

Utilising Eq. 5.6 as a reference, a profile of the KS score can be obtained by using a series of bus voltage values in a weak area just before collapse:

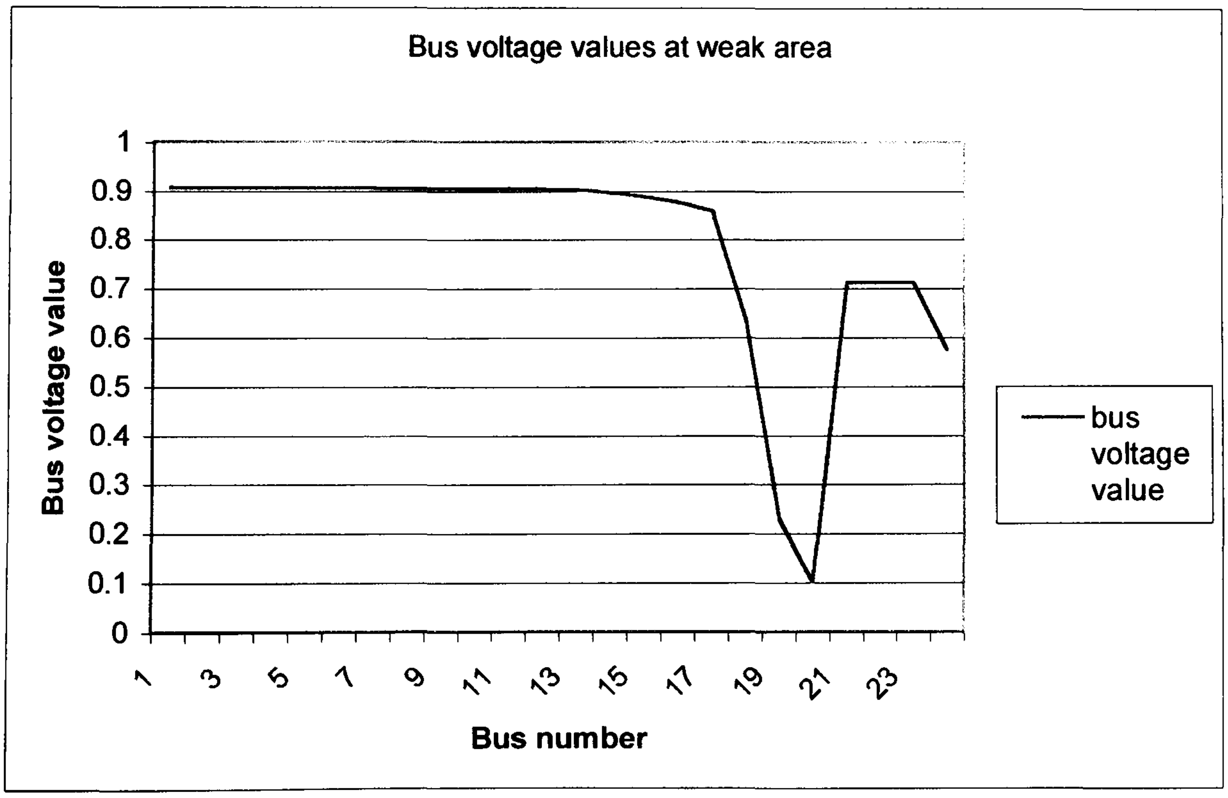


Figure 5.5: Bus voltage values at a weak area just before collapse.

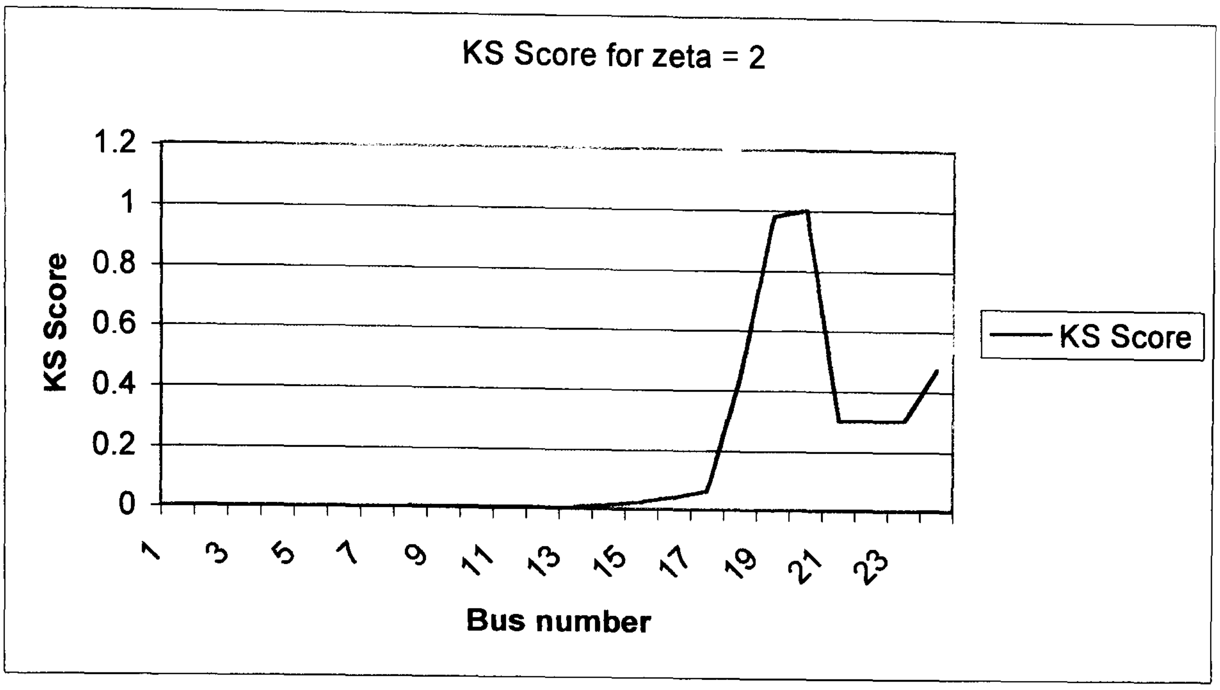


Figure 5.6: Corresponding KS score with respect to figure 5.5.

Figure 5.6 shows the corresponding KS score with respect to the bus voltage magnitudes shown in figure 5.5. The KS score shown is made with reference to a constant value. In this case the reference point is the voltage magnitude at bus 1. Comparing between figure 5.5 and 5.6, it is shown that as the bus voltage magnitude varies from the reference point (voltage level of bus 1) the KS formulation will give a corresponding score. This score will increase or decrease dependent upon the direction of the variation with respect to the reference point. The KS gives a negative score if the value has increased in value or positive score if the opposite. The highest variation will give the highest KS score as shown in figure 5.6. One may argue the rational of choosing the first bus value as a reference and what effect it would have if other form of referencing values are chosen. This issue is further discussed and presented in Appendix A. A conclusion can be made that regardless what type of reference value chosen, the KS score profile will identify the most significant attribute and it is dependent upon the assumptions made during the process.

Notice that the value ξ is mentioned and set as 2. As explained in chapter 4, ξ denotes the amount of fuzziness required for the KS score. The characteristics of KS score with variations of ξ can be seen in Appendix B. Conclusion found in Appendix

B show that the most suitable value of ξ is 2. This value will be used on all tests using the FFDT formulation as an information scoring technique.

In the FFDT proposed in [3-6], utilising Eq. 5.6 as the information quality measure is due to the nature of the database in question. Referring to literature, the dataset consists of CCT values representing stable and unstable points in the CCT space. Therefore it is imperative for the FFDT to plot the stability margin in terms of CCT values hence partitioning sets of CCT values into stable and unstable sets. By utilising the contrast discriminatory measure, local overshoots over the CCT margins will be eliminated hence a smoother margin is possible. For this application, there are no unstable points available due to the limitation of static load flow analysis, hence there is an option to exclude the contrast discriminator from the KS score, however, the addition of contrast into the algorithm will be looked into later in this chapter.

5.4.1 Why KS rather than other discrimination methods?

Digressing a bit from the FDT algorithm, clarification is needed with regard to the KS as an information quality assessment tool. KS measure is one of the most important module in the FDT as it selects the best or most significant attribute from the test set for partitioning purposes [7]. Looking into the evolution of the FDT and DT, there are various methods in selecting the most significant attribute. One of them is the information gain concept applied to the early DTs e.g. ID3, C4.5 and FID3 [8-10] some of them are explained in chapter 4.

In general, the nature of the database determines the type of measure required. From a statistical point of view, there are various classes of distribution. For data mining purposes, DT deals with either continuous or discrete distribution. The term continuous and discrete refers to the nature of how the data in question is represented. In some applications, say for example, colour of black and white, it is safe to assume that colour is discrete in nature as black and white are an absolute value of colour. Another example to illustrate this is height measurements; a person

(A) who is 6 feet in height is definitely taller than a person (B) who is 4 feet in height with a difference of 2 feet.

On the other hand, continuous data are sets of values that do not have a quantifiable value. For example, looking back into colour class of black and white, between them are shades of grey, which moves to dark grey to light grey. These intermediate colours are difficult to quantify in terms of how one shade of grey differs from the other. Another example of this can be seen in the fuzzy membership values. Fuzzy membership values are a range of numbers between 2 absolute value of 1 and 0, hence values in between are known to be 'fuzzy' or grey values. These values are known to be continuous in nature.

Another aspect that should be taken into account for continuous data is that its measured value is changing with time. So there is the issue of what is the absolute value for that particular data with respect to time. For the case of a discrete data, it is fixed that black is black and white is white regardless how we look at it. But if we look at the height comparison example illustrated above, giving the assumption that both them are growing at an independent rate with each other hence there is a possibility that B can be taller than A given time and if the rate of growth for B is much faster than A. If we were to chart the height of both persons with time, we will get a range of heights, which in this case changes over time. This can be termed to be a continuous set of distributions as it changes over time.

Looking from the power system aspect, measured and calculated values that represent the condition of the system are always changing or dynamic in nature. This means that data retrieved from the power system are considered to be continuous in nature. Even if we look at the values derived from static power flow simulations, the values are fixed for a given operating point, but this is correct for that particular point of time. Given a new set of environment constraint to the power system on a different time slot, it will give a different set of values. Since the approach for this research does correlate with events that is changing all the time, hence the data values obtained are treated to be continuous in nature hence the use of KS measure as an information quality evaluation tool.

5.4.2 Effects of contrast discriminator

As explained in chapter 4, the FFDT has a built in addition to the KS formulation known as the contrast discriminator. Recalling, the contrast discriminator in Eq. 4.12 and its behaviour as shown in figure 4.10, we can observe its effects in the KS score denoted by Eq. 5.16. Applying Eq. 5.16 to an IEEE 300 bus test system using the first bus value as a reference and comparing it with the conceptual KS formulae (Eq. 4.14) gives the following differences as shown in figure 5.7. The values used for the KS formulation is the bus voltage magnitudes obtained from load flow simulations.

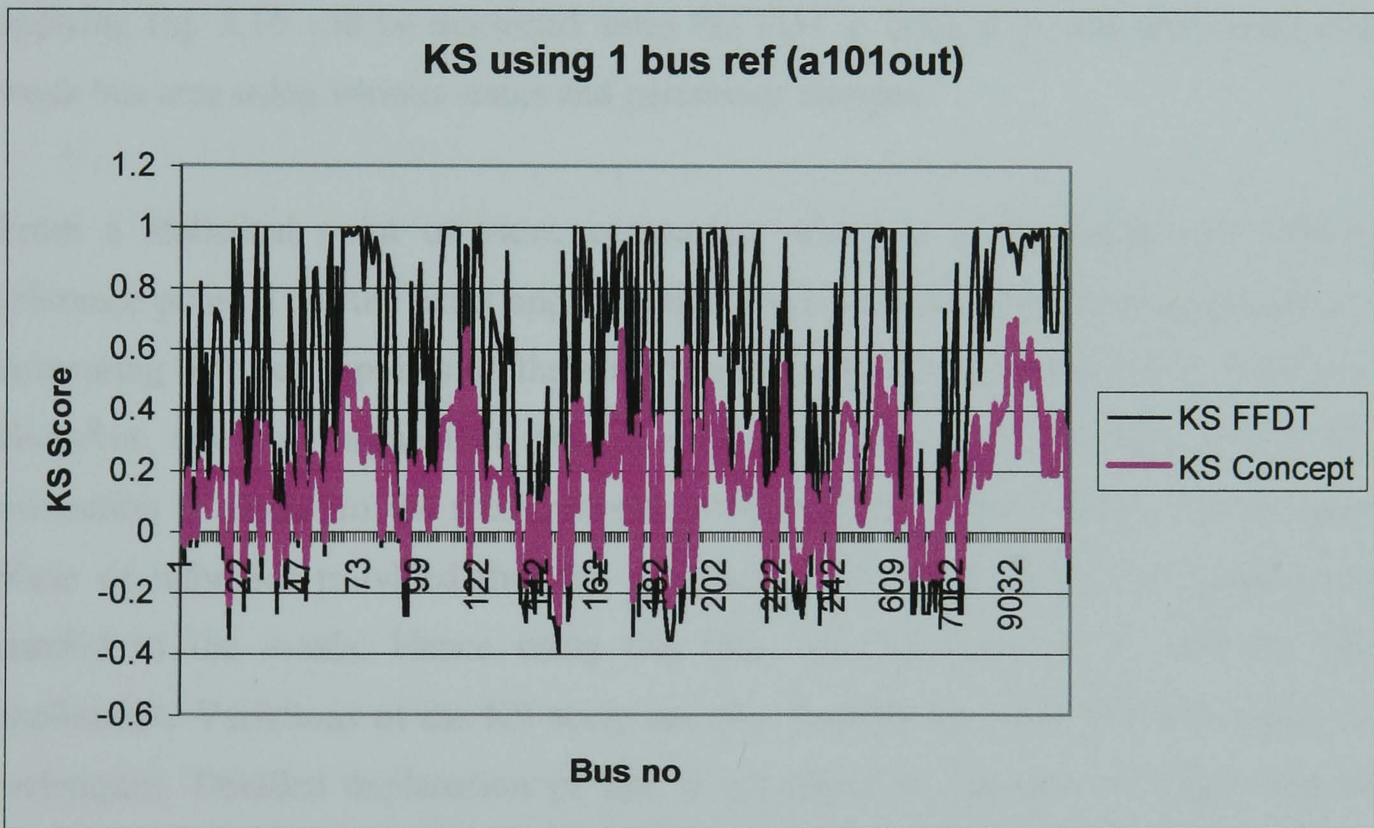


Figure 5.7: KS score for an IEEE 300 bus system using KS score of Eq. 5.16 (KS FFDT) and of Eq. 4.14 (KS Concept) at a loading factor of 1.01.

Figure 5.7 shows that the KS score for both equations indicate an increase of value at areas with high deviations from the reference value. The overall profile is generally similar with the exception that Eq. 5.16 shows a maximum KS score of 1 at high value differences compared to Eq. 4.14, the KS score does not go as high as 1. This can be explained by looking into the concept of Eq. 4.14 where it measures the difference between 2 points, hence Eq. 4.14 will only yield the magnitude of the difference, whereas in Eq. 5.16, has the addition of ξ and also the integration of contrast discriminator. All these additions would give a maximum KS score when a significant difference is encountered.

5.4.3 Validity of the KS approach

The KS approach can be applied in various ways, from the literature [2, 3, 6, 11] the KS is applied to differentiate between clusters of stable and unstable points in the CCT space using a formula similar to Eq. 5.16. For this particular application, the effects of contrast discriminator can be seen more clearly in preventing overshoots in charting the stability manifold in the CCT space. For this research, the effects of applying Eq. 5.16 will be discussed since the FDT is utilised to plot the strong and weak bus area using various states and parameter changes.

From a statistical point of view, comparing values in a distribution set with a reference point is not the exact application of the KS formulation. Its main purpose is comparing between 2 points on the same plane of reference as explained in chapter 4. However, one may argue that using a single reference point through out the KS evaluation is similar to the measurement of distance between 2 points on the same plane of reference provided that the reference value is assumed as a straight line parallel to the x-axis. Hence using this idea, the KS function is valid for this application. Variations of the KS score are also possible by using different reference techniques. Detailed explanation of this is presented in Appendix C. Referring to Appendix C's discussion and results in Appendices D – E, it can be concluded that using every bus voltage magnitude obtained from load flow solution at a loading factor of 1.0 as a reference would give better sensitivity to the KS score hence partitioning will be more accurate. For other referencing methods, sensitivity will increase as the loading factor is increased towards collapse. The overall effect of using 2 different referencing techniques will be compared in the FDT test results presented later in this chapter.

5.5 Stage 3: Using KS score as a partitioning rule

Based upon the KS measure results, it is now possible to select the most significant node for a partitioning rule. In general, the partitioning rule for this FDT is considered to be very straight forward using the definition in Eq 5.7. The attribute labelling system used in this algorithm is based upon the bus number. Each bus number will have attributes connected to it, depending upon the attribute used for the FDT algorithm. The reason behind this is to enable multiple attribute analysis that will be presented later in this chapter.

$$Partition\ node \equiv \begin{cases} left\ branch\ for\ bus\ numbers\ from\ 1\ to\ the\ significant\ attribute \\ right\ branch\ for\ the\ rest\ of\ the\ bus\ numbers \end{cases} \qquad (5.7)$$

Where: left and right means attribute path direction towards the new test nodes.

5.6 Stage 4: Stop split check

Stage 4 checks whether further partitioning is possible. This stage is considered to be important ensuring that the data set is not over partitioned (which leads to complex FDT structure) or under partitioned (which leads to inaccurate classification). The stop split check can be applied in various forms. One of which is in terms of test node purity check. Purity is defined as the variation of the attribute value within the test node. A pure test node is known to have attribute values that has minimal or zero deviations from each other. For this research, zero deviation check is impractical since the probability of the attribute within the test node to have similar values with the neighbouring attributes is very low. As an alternative, another form of purity check may be performed which will be explained later in this section and explored further later in this chapter.

Apart from that, there can be other forms of stop split checks such as the look-ahead scheme. This scheme checks the purity of future nodes if the tested node is further partitioned. The FFDT uses a stop split rule defined by Eq. 4.19, similar to the look-ahead scheme concept. For this research, several stop split checks are tried and tested in terms of their effectiveness. They are:

(a) The amount of attribute of the test node (minimum population check)

This is a very crude method of stop split checks, if the test node does not have enough attribute population, the FDT will stop partitioning that particular node and converts the test node into a leaf node. This is demonstrated in Test 2 later in the chapter.

(b) Purity check

This is an improvement of (a) where it checks the purity of the test node. Purity in this research is in terms of the population of weak and strong buses. If the test node is relatively pure, hence no partitioning is required. If not, the test node is further checked in terms of the location of the weak and strong buses in the test node. If the location is as such that further partitioning would lead to single attribute node (over-partitioning) the test node will be labelled as a leaf node. This concept is further expanded later in this chapter.

5.7 Stage 5: Leaf labelling

This is a process of reverse fuzzification. Manipulating Eq. 5.3 and equating it in terms of x will give the following form:

$$x = \frac{\nu - 0.5}{\alpha} + \beta \tag{5.8}$$

Since we have α and β values earlier, we can determine the crisp value of x. One may argue the reasoning behind reverse fuzzification rather than using more conventional defuzzification approach in the literature. Conventional defuzzification approach gives a generalised 'conclusion' or deduction on the fuzzy sets. Since the idea of fuzzification in this research is to 'stretch' the attributes to span between a 1 and zero, direct reverse fuzzification is adequate. However, future possibilities of using conventional defuzzification approach will be discussed further in Chapter 6.

5.8 FDT data preparations

5.8.1 Data for basic FDT engine test

For preliminary FDT engine test, fictional bus voltage values are created using a random number generator with predefined limits shown in table 5.0. The reason of this data type is for testing FDT partitioning ability in discriminating between normal and dangerous voltage levels. The term normal voltage values denote the voltage range within the standard ± 5% deviation from the nominal 1.0 pu value. Any values above and below this range are considered to be dangerous to the system. The program used to create these data value is shown in Appendix H. A summary of the created data set is as follows:

Number of bus: 351

Table 5.0: Voltage and Data Range for basic FDT engine test

Voltage Range	Data Range
Lower Unstable Region	0.85 – 095
Stable Operating Range	0.95 – 1.05
Upper Unstable Region	1.05 – 1.15

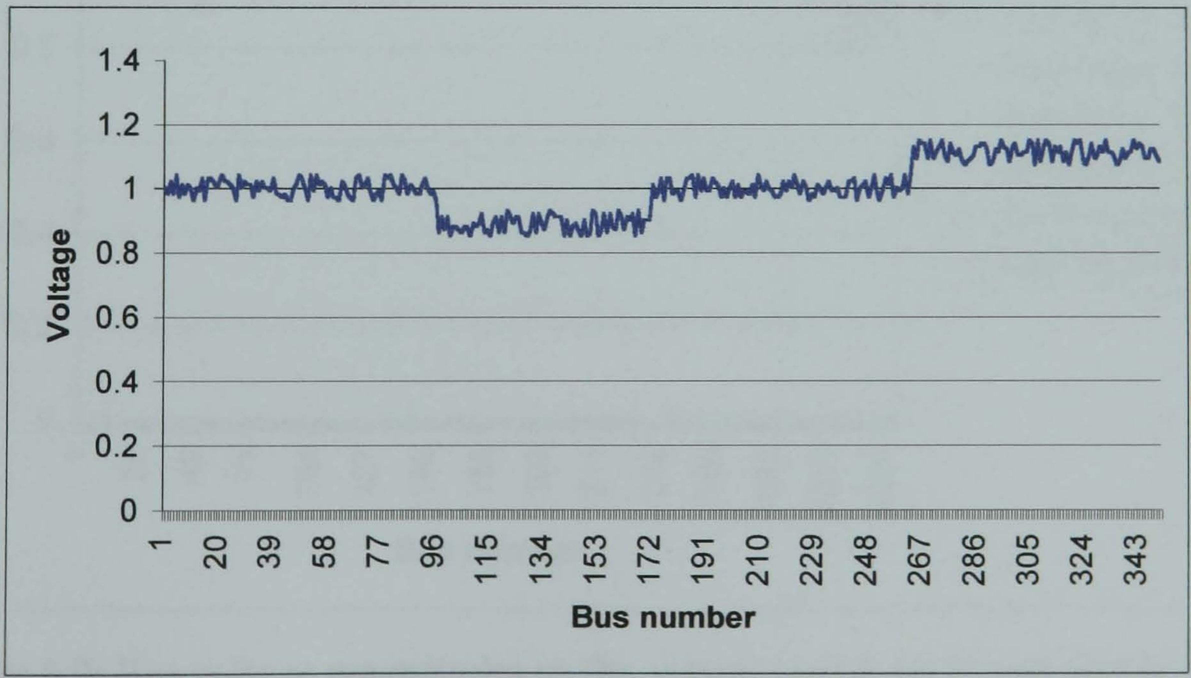


Figure 5.8. Profile of the bus values for the basic FDT test

5.8.2 Data for FDT test using IEEE 300 bus system

Preparation of these attribute data is based upon data obtained from static load flow simulations. The method of load flow simulation chosen is the Newton Raphson iteration process. The program used for this research is the MATPower module a power flow package programmed under Matlab™. For more information on the MATPower module, please refer to [12]. Some may argue on the accuracy of the load flow program and the type of iterative process used. Since this research is focused on developing a partitioning algorithm, load flow solution accuracy is not of the main concern.

Looking back to chapter 2, there are various forms and symptoms that may lead to voltage collapse. Each symptom corresponds to a different system behaviour as the system operating point drifts towards instability. An example of this can be seen in figure 5.9 where it shows a profile of bus voltage magnitudes as the system load is increased until the system collapse.

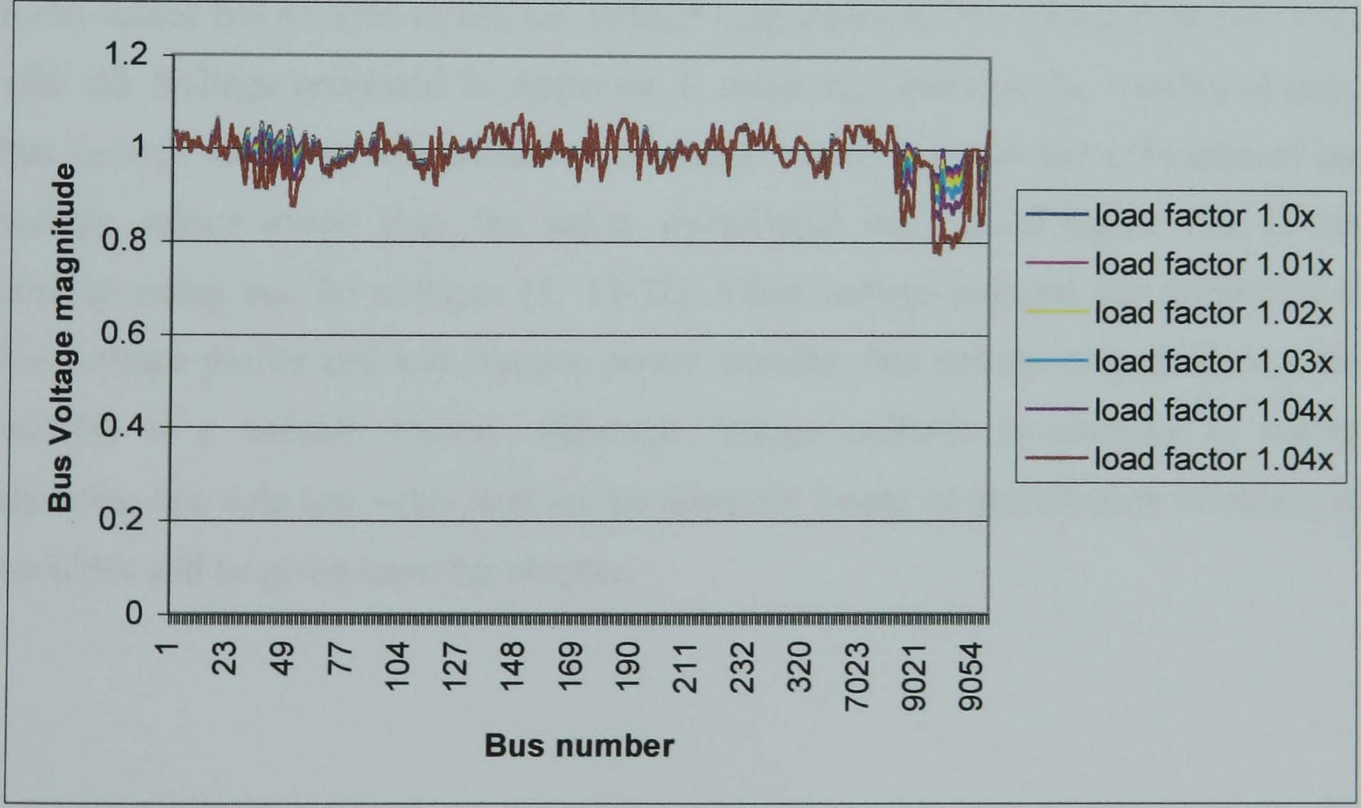


Figure 5.9: Bus voltage magnitudes as the system load is increased slowly at all load centres.

Figure 5.9 show that some portions in the bus voltage profile will experience a reduction of value as the system loading is increased. These portions are known as weak bus areas. For figure 5.9, there are 2 significant weak areas that can be seen which are on the left and right hand side of the bus voltage profile. One may argue that this characteristic is dependent upon how the system load is increased. On the other hand, similar behaviours can be seen on a different loading strategy as shown in Appendix I. To achieve automatic variable load increments, the author has written a program that reads the load flow input data and changes the data automatically. This program is presented in Appendix J. The current program shown in Appendix J is designed to change the load values for all load busses in the load flow database. This program can be modified to change loading values at selected buses or at all buses depending on the user's requirements.

For the application of the FDT as an analytical tool, selection of attributes is important. By choosing a suitable attribute, significant behaviours can be observed as shown in figure 5.9. For this research and based upon findings presented in Appendix I, the author has decided to use bus voltage magnitudes as one of the attributes. Even with the findings presented in Appendix I, some may question the validity of using bus voltage values as test attribute. Literature has shown that the reduction of bus voltage values lower than the safety operational level gives higher risk to the corresponding bus for collapse [1, 13-22]. Since voltage collapse has symptoms of low voltage profile and low reactive power transfer, bus voltage magnitude as a test variable is a suitable choice. Although voltage collapse is complex in nature, choosing one sole test value may not be adequate hence an introduction of other test variables will be given later this chapter.

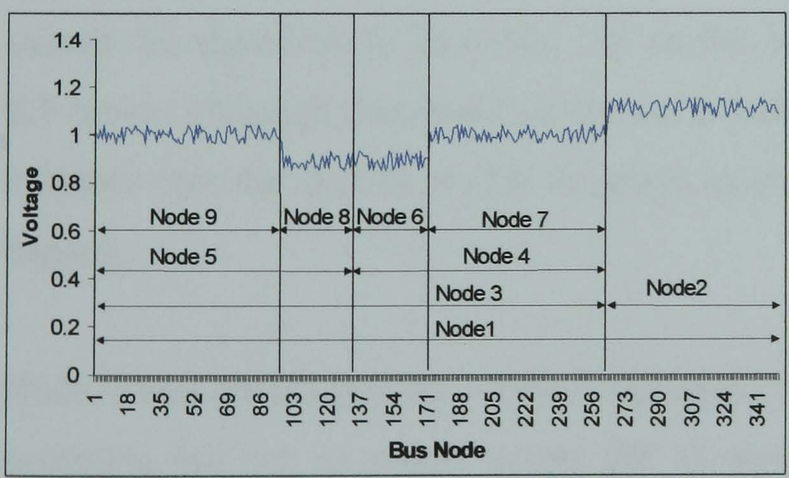
5.9 Test 1: FDT Basic Engine Test using FDT ver 1.0

The summary of the test 1 and its FDT parameters are as follows:

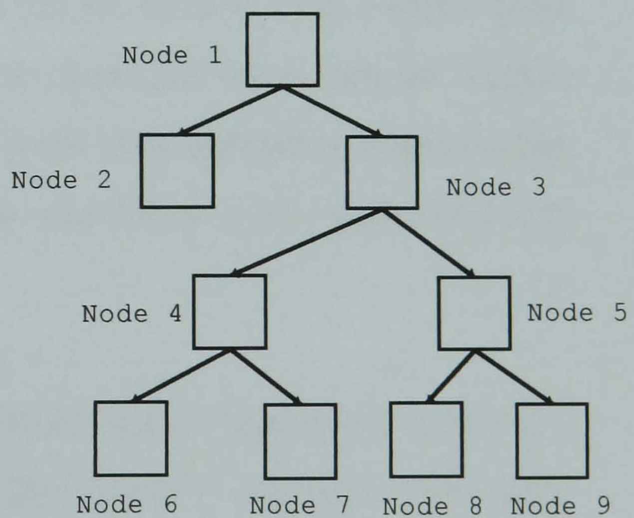
Table 5.1: Test 1 Program summary:

Version of FDT	FDT ver 1
KS technique	KS formulation based upon the FFDT concept using the first bus voltage value as reference.
Partitioning technique	Using maximum KS score location as partitioning limit.
Stop split rule	The rules for the stop split stage are based on user intervention. The user will observe the population count and also maximum and minimum attribute value in the test node.
Pruning technique	Manual pruning technique
Test system used	Using a fictional 351-bus system as explained in section 5.8.1. The bus voltage values are generated using a random number generator within limits of a normal operating level, low bus voltage level limit and high bus voltage level limit.
Computer used	Intel Pentium 120 MHz

Results:



(a)



(b)

Figure 5.10: FDT results using 351 simulated bus voltage values (a) In terms of the partitioning location (b) In terms of the nodal map of the FDT

Discussion of results

Referring to figure 5.10, it is shown that the algorithm is capable of partitioning the overall system voltage values into groups of stable and unstable bus voltage sets. Analysis started from the root node (node 1), which showed the original set of bus voltage levels. Node 1 was partitioned into 2 sections, shown as node 2 and node 3. Node 2 represents the upper unstable region of the system where it starts from bus 262 to bus 351. Applying user intervention on the stop split and pruning stage, node 2 is labelled as a leaf node. Node 3 represents the remaining of the system excluding node 2, which is from bus 1 to bus 261. Since Node 2 is a leaf node and no significant partitioning is possible, the algorithm stops splitting node 2 and transfers its partitioning operation to node 3.

Node 3 was then partitioned into nodes 4 and 5. Node 5 represents the first half of node 3, which is bus 1 to bus 144. Node 4 represents the other half of node 3, which is bus 145 to bus 261. Based upon the algorithm logic, partitioning should incur at just the stable regions and unstable regions. Instead, it cuts right through the middle of node 3. The reason is that during the generation of the data sets, there was noise in the data (unwanted data) generated due to inaccuracy of the random number generator. Hence the algorithm detected a value, which spiked above the lower unstable region in node 3, which was bus 145. In the KS measure, this sudden spike caused the algorithm to label bus 145 as the most significant node with the highest KS score. Although this could be seen as a fault in the data generation sequence but it shows that the system is able to adapt to noisy conditions where data error may happen.

Node 4 was partitioned into node 6 and node 7 where node 6 represents the normal operating bus set of which is bus 166 to bus 261. Node 7 represents the lower unstable bus voltage values, which are from bus 145 to bus 165. Using user intervention on the stop split and pruning schemes sets node 6 and node 7 to be leaf nodes. Partitioning continued for node 5.

Node 5 was partitioned into nodes 8 and 9 where node 8 represents the lower unstable bus set, which starts from bus 95 to bus 144. Node 9 represents the normal operating bus set, which consists of bus 1 to bus 94. Again at this stage, using the same technique for the stop-split and pruning algorithm sets nodes 8 and 9 to be leaf nodes.

After reviewing the leaf nodes (nodes 2, 6, 7, 8 and 9) it shows that the algorithm is capable of partitioning the overall system voltage values into groups of stable and unstable voltage sets. In summary, Node 2 represents bus sets with high unstable voltage levels. Nodes 9 and 6 represent the stable operating voltage level. Nodes 8 and 7 represent the lower unstable voltage levels.

As a conclusion, the FDT engine has the ability to exhibit good partitioning although the KS technique is relatively crude with user intervention at the stop split and pruning stage. In the next test the FDT algorithm is further improved and optimised in terms of its programming flow and stop split capabilities.

5.10 Improvement on the stop split capabilities

The FDT used for test 1, needed an automatic stop split stage. An automatic stop split addition will improve the execution flow of the FDT algorithm. As discussed earlier, there are several stop split approaches. One of which is a crude technique based upon population size. A population check finds the population size of the test node. If the test node population is too small, further partitioning will give leaf nodes with a very small attribute population or even test nodes with only 1 attribute. In other words, further partitioning will cause over partitioning.

Using the IEEE 300 bus test system, an investigation is made in terms of the FDT performance with variations of population limits as a stop split requirement. Details of these tests are presented in Appendix K. Results show that a low attribute limit will give detailed partitioning capability but with longer execution time whereas high attribute limit will give too general partitioning capability but with better execution time. For the next FDT, an attribute limit of 150 is set, which from the tests in Appendix K gives a good partitioning capability with an acceptable execution time.

5.11 Test 2: Application to an IEEE 300 bus system using FDT ver 2.0

Table 5.2: Test 2 Program summary:

Version of FDT	FDT ver 2
KS technique	Using KS formulation based upon the FFDT concept using the first bus voltage value as reference.
Partitioning technique	Using maximum KS score location as partitioning limit.
Stop split rule	The rules for the stop split stage are based upon minimum (minimum population value used is 150 attributes) of the test node. If the population of the test node is within the minimum population range, the node will stop partitioning and label the test node as a leaf node.
Pruning technique	No pruning technique applied
Test system used	IEEE 300 bus system using all load bus increase at a loading strategy of 1.05x. The bus voltage profile is based upon the system loading level just before collapse. The rational behind this is presented in appendix I where at higher system loading, there is a significant change in bus voltage values hence aiding the FDT in its classification objective.
Computer used	Intel PIII 1 GHz

Results:

Table 5.3: Summary of results for loading factor of 1.05x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.05x at all load buses	Attribute range in leaf node (Bus No.)
10	1 – 49
14	51
18	52 – 117
20	Empty
22	118 – 169
23	170 – 7166
13	9003
9	9004 – 9035
7	9036 – 9038
5	9041 – 9042
3	9043 – 9533

Based upon table 5.3, the location of the leaf nodes are as shown:

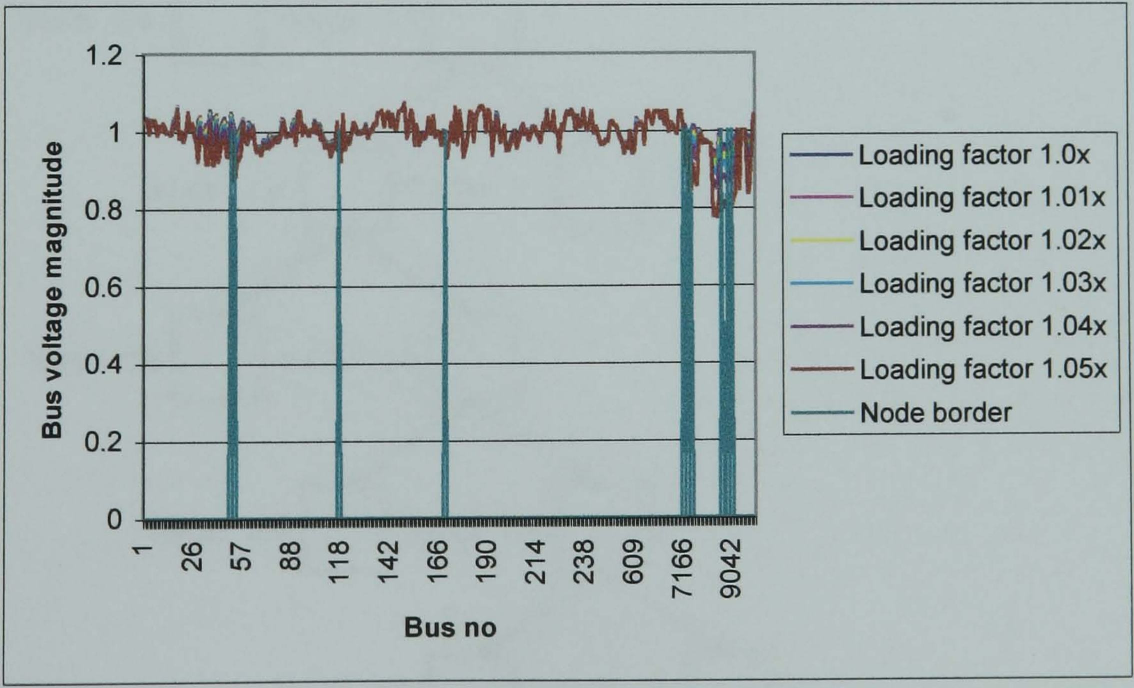


Figure 5.11a: Unprunned FDT results using IEEE 300 bus system with loading factor of 1.05x at all load bus.

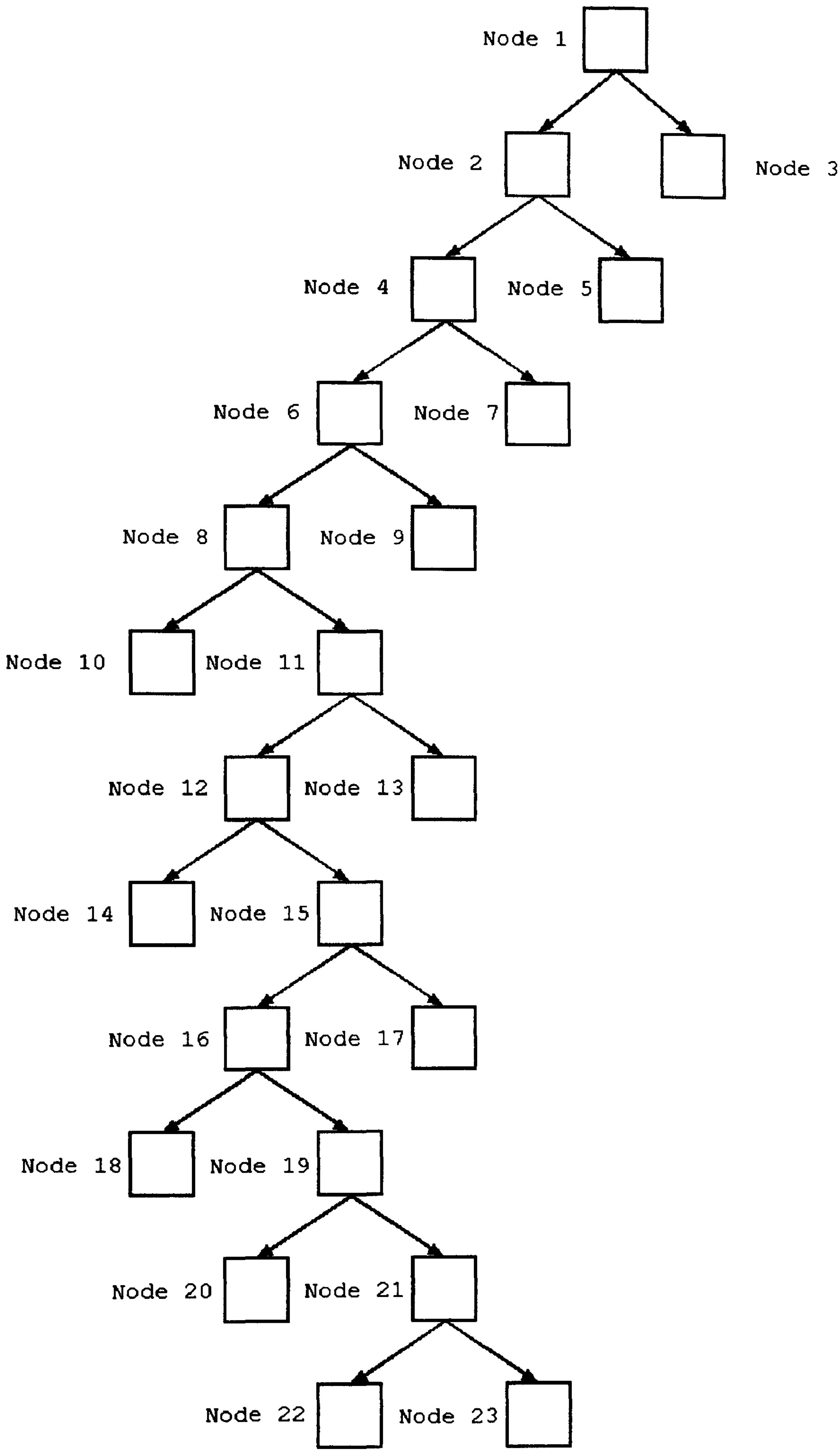


Figure 5.11: Full nodal map for test 2

Discussion of Results

Figure 5.11(a & b) show that the algorithm can partition the data set into several groups. On the other hand, looking at Table 5.3 and its corresponding leaf map in figure 5.11 it can be seen that the performance of the FDT is not as good as in test 1. There are too much over partitioning instances that caused several nodes to have a small population of attributes or even empty nodes as shown at leaf node No. 20.

Looking at the significance of the leaf node results, although it does partition into weak and strong bus areas, but the amount of leaf nodes especially at the weak bus area is high as such that the majority of the leaf nodes contained only 5 or less attributes. It would be useful if all these small leaf nodes were combined to make one leaf node that consists of a wider range of weak bus attributes.

One solution to over partitioning is to prune the algorithm manually. Pruning will collapse empty and low population nodes to its nearest test node hence simplifying the overall tree structure.

Another solution is to improve the partitioning stage where currently it is only based upon the location of the maximum KS score only. This unfortunately would give over partitioning problems as illustrated in this test.

5.12 Improving the partitioning rule

As mentioned earlier, the partitioning rule uses the maximum KS score as a guideline to create a partitioning limit. This may be suitable if the maximum KS score is located in the middle of the test set. If the maximum KS score is located at the extreme right and extreme left of the test node, there is a possibility that the FDT will over partition or will partition to such an extend that new nodes will consist of the original data set and an empty node. This can be explained as shown in figure 5.12.

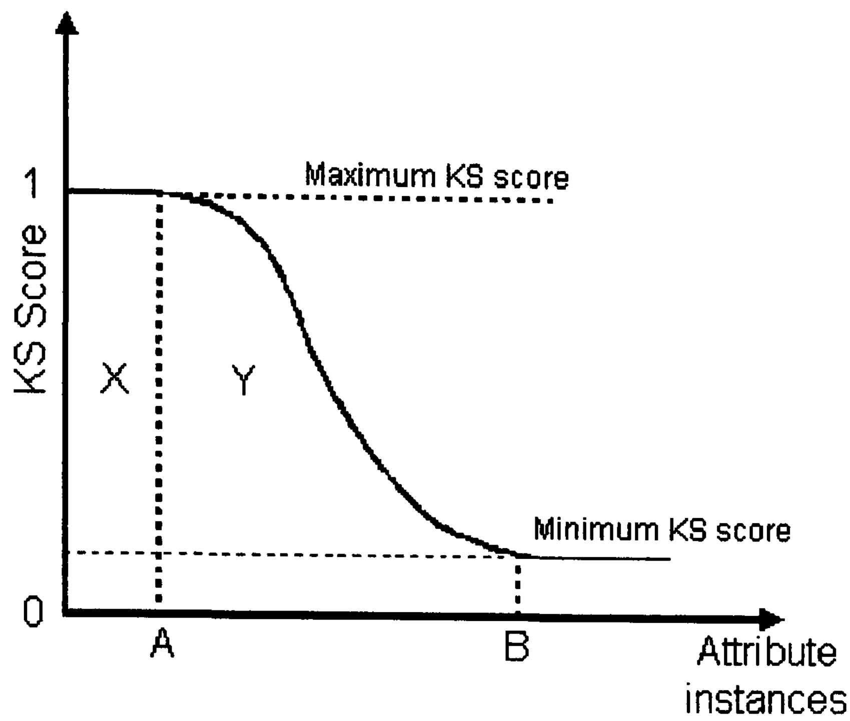


Figure 5.12: An example of KS score profile

Referring to figure 5.12, if the FDT uses the maximum KS score as a partition limit, the FDT will use the location of point A with respect to location of the first attribute in the test node as a limit and partition in test node into 2 new nodes consisting of a node that represents area X and Y respectively. Unfortunately, this will give a node with small population (in this example node that represents area X) or for extreme cases, an empty node. However, this can be improved by instructing the algorithm to look for the minimum KS score (which is the KS score at point B) and use it as a partitioning limit instead. Hence the new partitioned node will be area X and Y combined in one node and the rest of the test node in the other. However, in applying this concept, care has to be taken that the same over partitioning incident does not happen when using the location of the minimum KS score as a partitioning limit. Using these concepts, the program flow is as shown in figure 5.13.

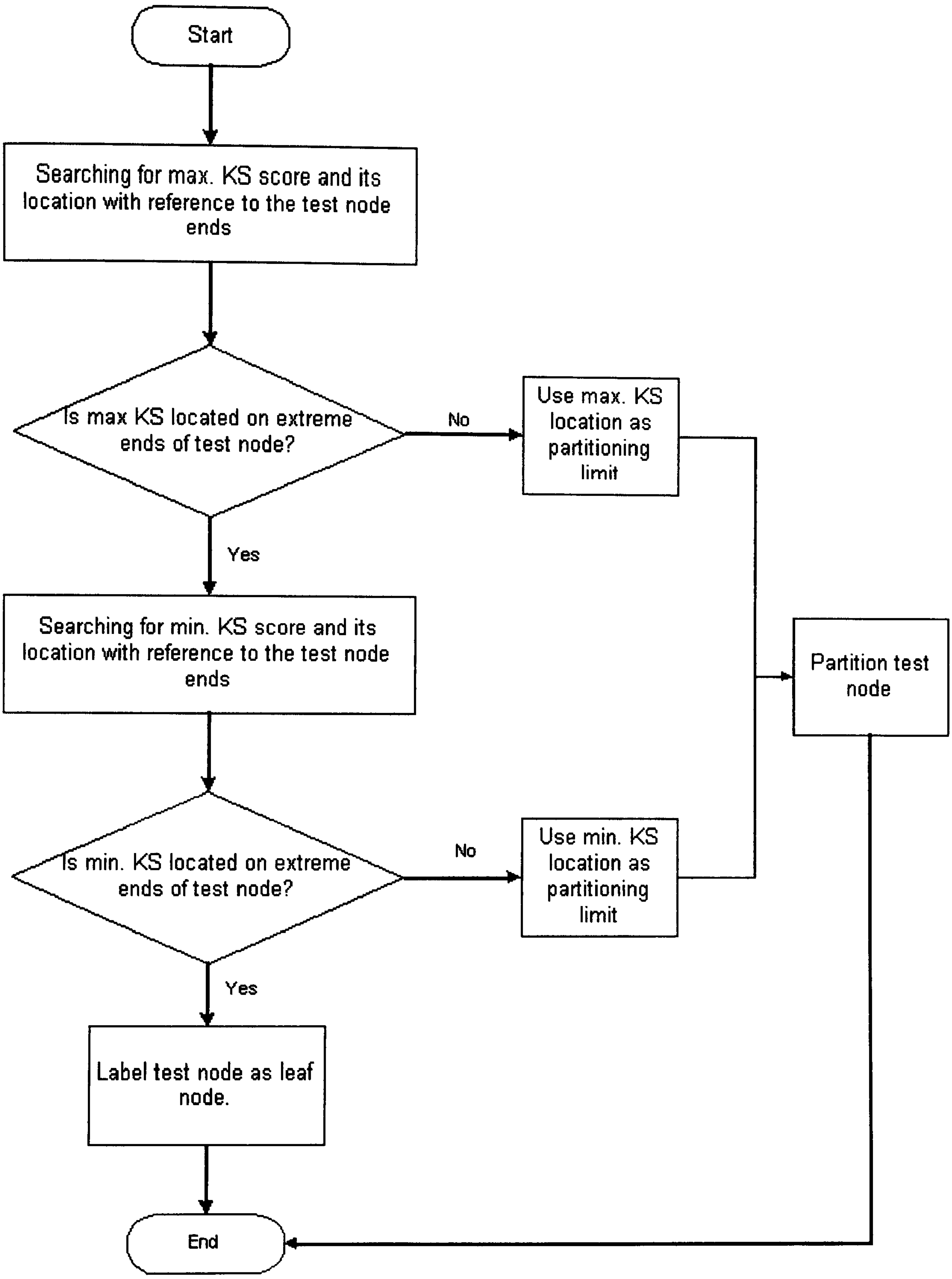


Figure 5.13: Improvements on the partitioning stage in terms of KS algorithm flow

Applying this concept will increase the chances of the FDT to partition significant portion of the test set hence giving better accuracy with fewer number of over partitioned leaf nodes.

5.13 Improving the stop split stage

The last 2 tests exhibit the use of crude techniques for a stop split rule, which are user intervention based and also population size based. Although population size in a node is important to ensure good generality properties, but there are also forms of other limits that be added and combined into the stop split range. For this stage, the stop split criteria will have the following elements or steps:

(a) Purity test.
(b) Population limit test.

The term purity test was explained briefly before in this chapter. However, summarising, purity test checks the test node whether it consist of a majority of strong buses or majority weak buses. If the aforementioned case is true, hence the test node is labelled as pure and changed into a leaf node. If not the test node has to be partitioned further. The purity test if applied on its own, could give an over partitioning possibility. A more meaningful solution to this problem is to use the purity and population test combined. Combining both the purity and population limit test would maintain good stop split performance while reducing possibilities of over partitioning. Applying these concepts into the FDT would yield the following program flow at the stop split stage as shown:

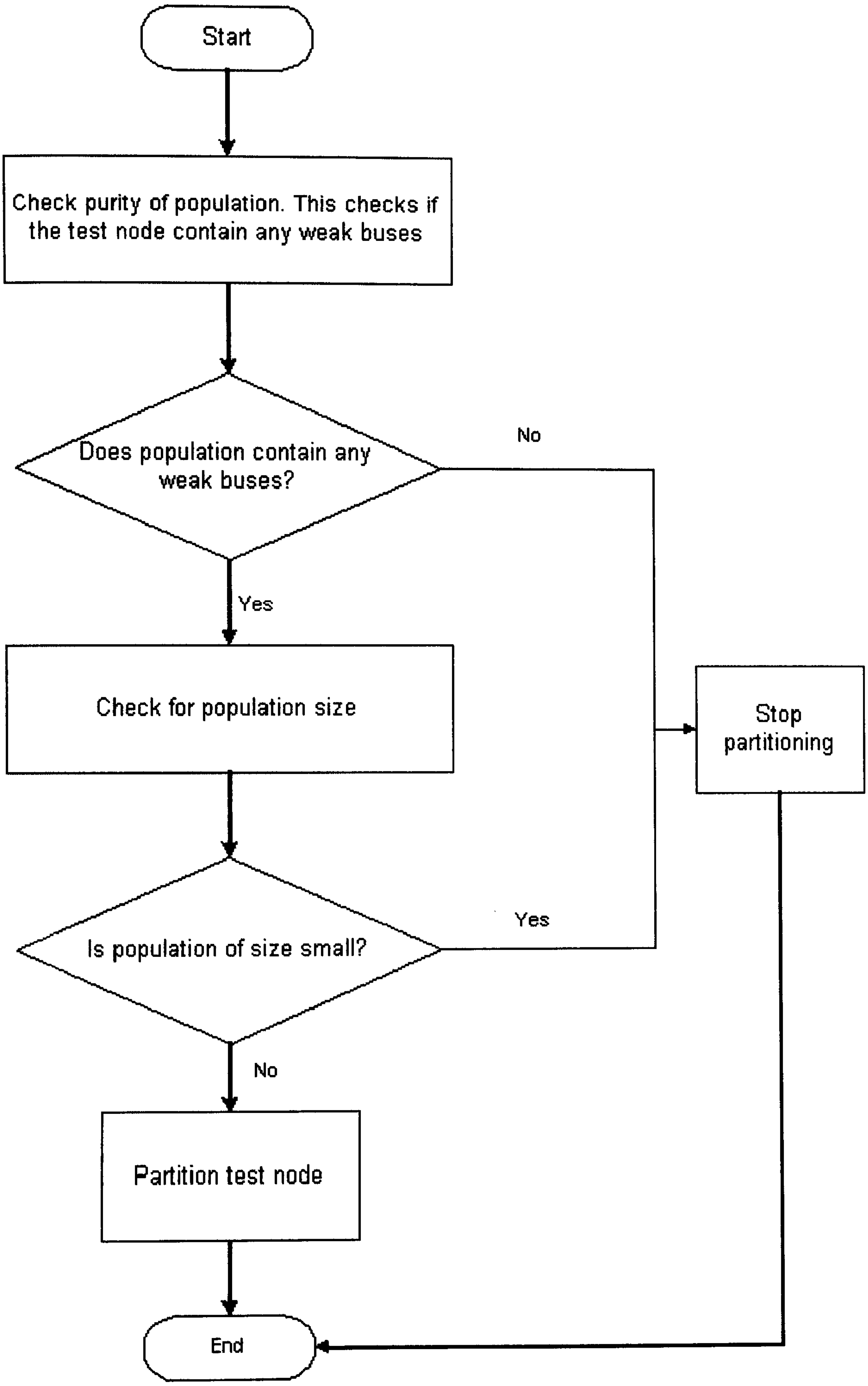


Figure 5.14: Program structure for stop split improvements

Using the program structure in figure 5.14, further improvement in terms of its stop split capabilities can be made. It is noted that there is a need to explain in detail with regards to the purity test mentioned earlier. For this research, the purity test will search the test node for the availability of attributes that represents weak buses. The term weak bus for this stage represents a bus voltage magnitude below a pre-defined limit known as the minimum attribute limit. An investigation was made towards determining a suitable minimum attribute limit as presented in Appendix L. Based upon findings obtained, the minimum limit is set to 0.90. Notice that the program flow shown in figure 5.13 and 5.14 has overlapping functions where figure 5.13 has a form of stop split rule embedded in the partitioning rule. This should not pose as a problem but should improve the FDT performance.

5.14 Other improvements

Other improvements made to the FDT are in terms of program structure. As mentioned earlier, there are considerations to include other attribute elements into the FDT to improve its partitioning capability. Although this will be looked into later in the chapter, it is important that the FDT program structure is ready to receive new attribute element without major modifications to the program structure.

Applying these modifications, the improved version of FDT will be tested to the IEEE 300 bus system with different loading strategies.

5.15 Test 3: Application to IEEE 300 bus system using FDT ver 3.0

Table 5.4: Test 3 Program summary:

Version of FDT	FDT ver 3.0
KS technique	Using KS formulation based upon the FFDT concept using bus voltage values (of the IEEE 300 bus test system at a loading factor of 1.0x at all load buses) as reference
Partitioning technique	Comparing the location of the maximum and minimum KS score. • If theses scores are located at both ends of the test nodes, partitioning will stop. • If the maximum KS score is on the extreme ends of the test node, and the location of the minimum KS score is not located at the extreme ends of the test node, the partitioning limit will be based upon the location of the minimum KS score. • If the maximum KS score is not at extreme ends, partitioning will be based upon the maximum KS location as a partitioning limit. The location of the KS scores are considered to be on the extreme ends if is located within the range of 10 attributes from each end.
Stop split rule	The rules for the stop split stage are based upon the purity of the node and the minimum population (minimum population value is 50 attributes) of the test node. The summary of the rules are: • If the test node consist of all strong buses (minimum attribute value is more than 0.9pu) stop splitting the node and label is as a leaf node.

	• If the test node consist of a combination of strong buses and weak buses (minimum attribute value is less than 0.9pu). Check the total population of the test node. If it exceeds the minimum population limit continue partitioning. If not, stop partitioning and label the test node as a leaf node.
Pruning technique	No pruning technique is used
Test system used	IEEE 300 bus system using 3 different loading strategies: • Loads in all load bus are incremented • Loads in only heavily loaded buses are incremented • Loads in only lightly loaded buses are incremented
Computer used	Intel PIII 1GHz

Results

The results are presented is in the form of leaf node numbers and its location. The location of each leaf node is presented on a general bus voltage magnitude profile with all the loading factors superimposed on it. The reason for this is to highlight the weak buses area (denoted by varying bus voltage magnitude when the loading factor is increased) and how good the FDT partition the data set into sections of strong and weak buses.

5.15.1 All load bus loading strategy

Loading factor 1.01x

Table 5.5: Summary of results for loading factor of 1.01x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.01x at all load buses	Attribute range in leaf node (Bus No.)
2	1 – 159
3	160 – 9533

Based upon table 5.5, the location of the leaf nodes are as shown:

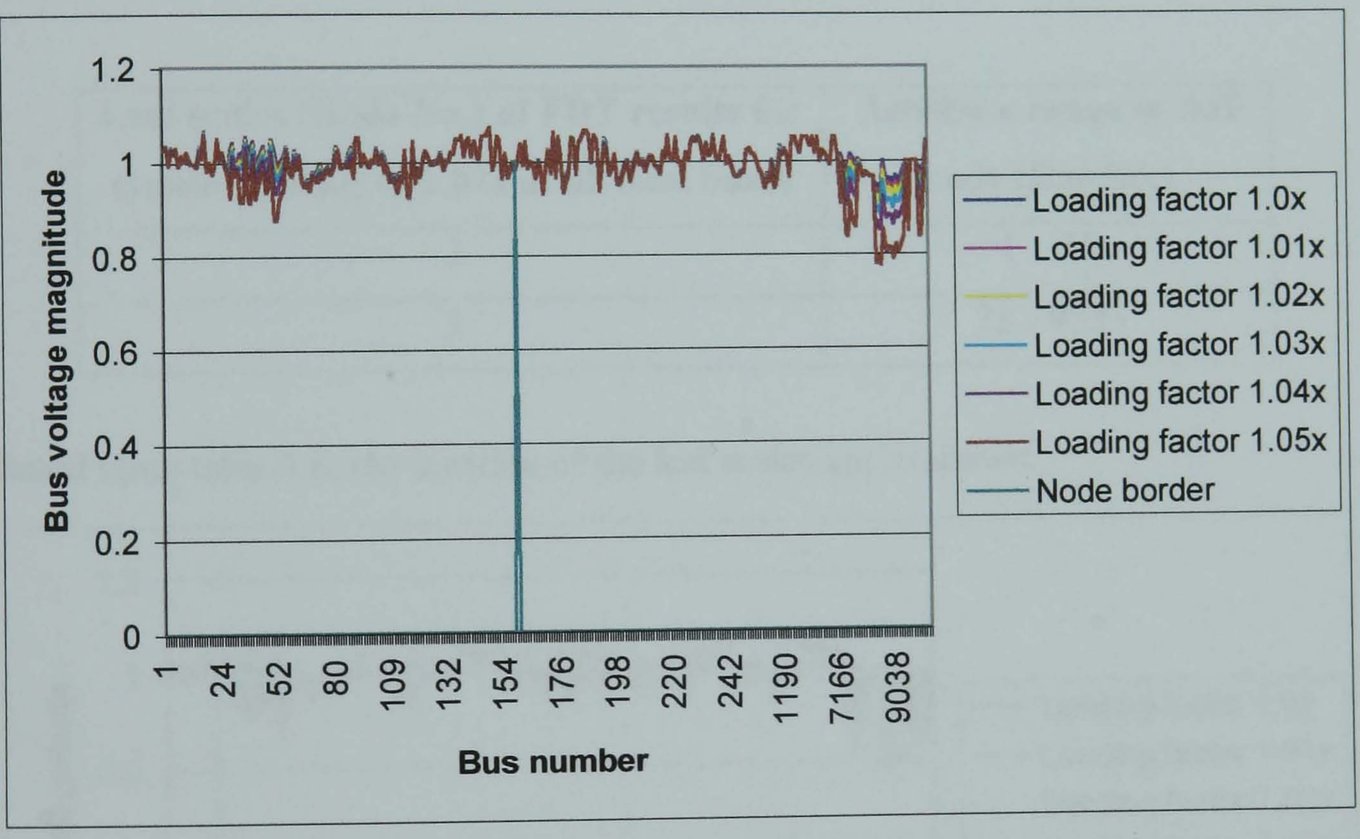


Figure 5.15a: Leaf node location for all load bus loading strategy

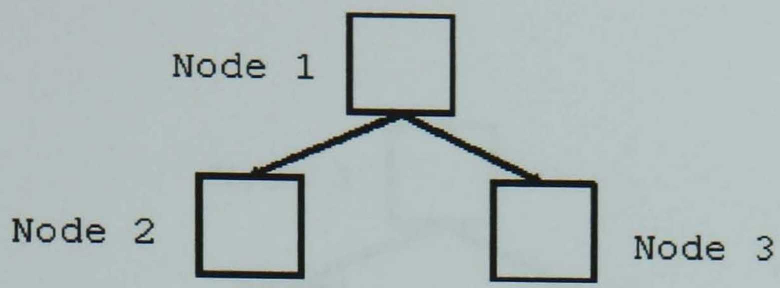


Figure 5.15b: Nodal map of FDT

Loading factor 1.02x

Table 5.6: Summary of results for loading factor of 1.02x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.02x at all load buses	Attribute range in leaf node (Bus No.)
2	1 – 21
3	22 - 9533

Based upon table 5.6, the location of the leaf nodes are as shown:

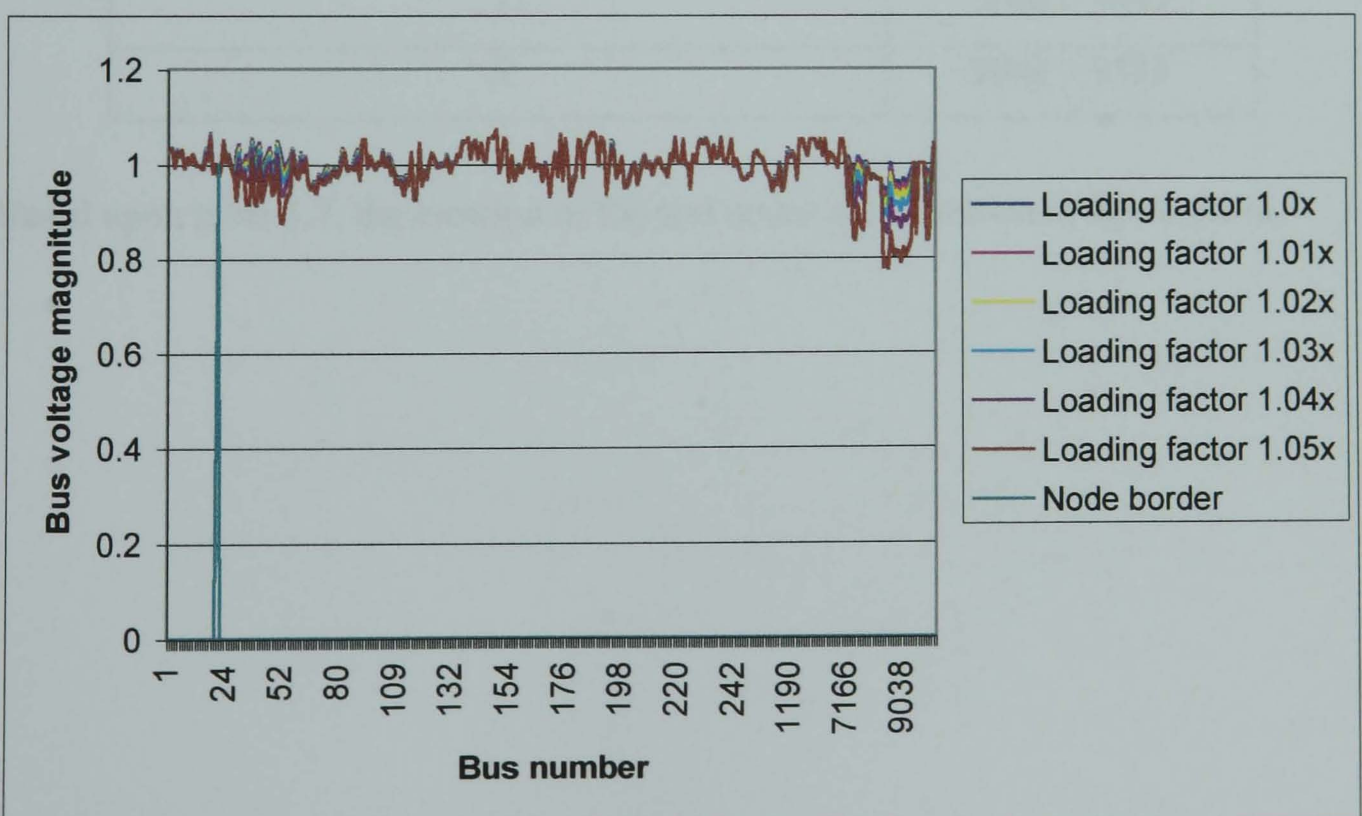


Figure 5.16a: Leaf node location for all load bus loading

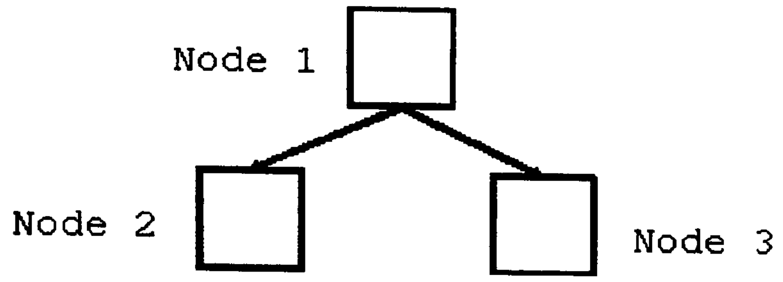


Figure 5.16b: Nodal map of FDT

Loading factor 1.03x

Table 5.7: Summary of results for loading factor of 1.03x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.03x at all load buses	Attribute range in leaf node (Bus No.)
2	1 – 213
6	214 – 9005
7	9006 – 9033
5	9034 – 9533

Based upon table 5.7, the location of the leaf nodes are as shown in figure 5.17a.

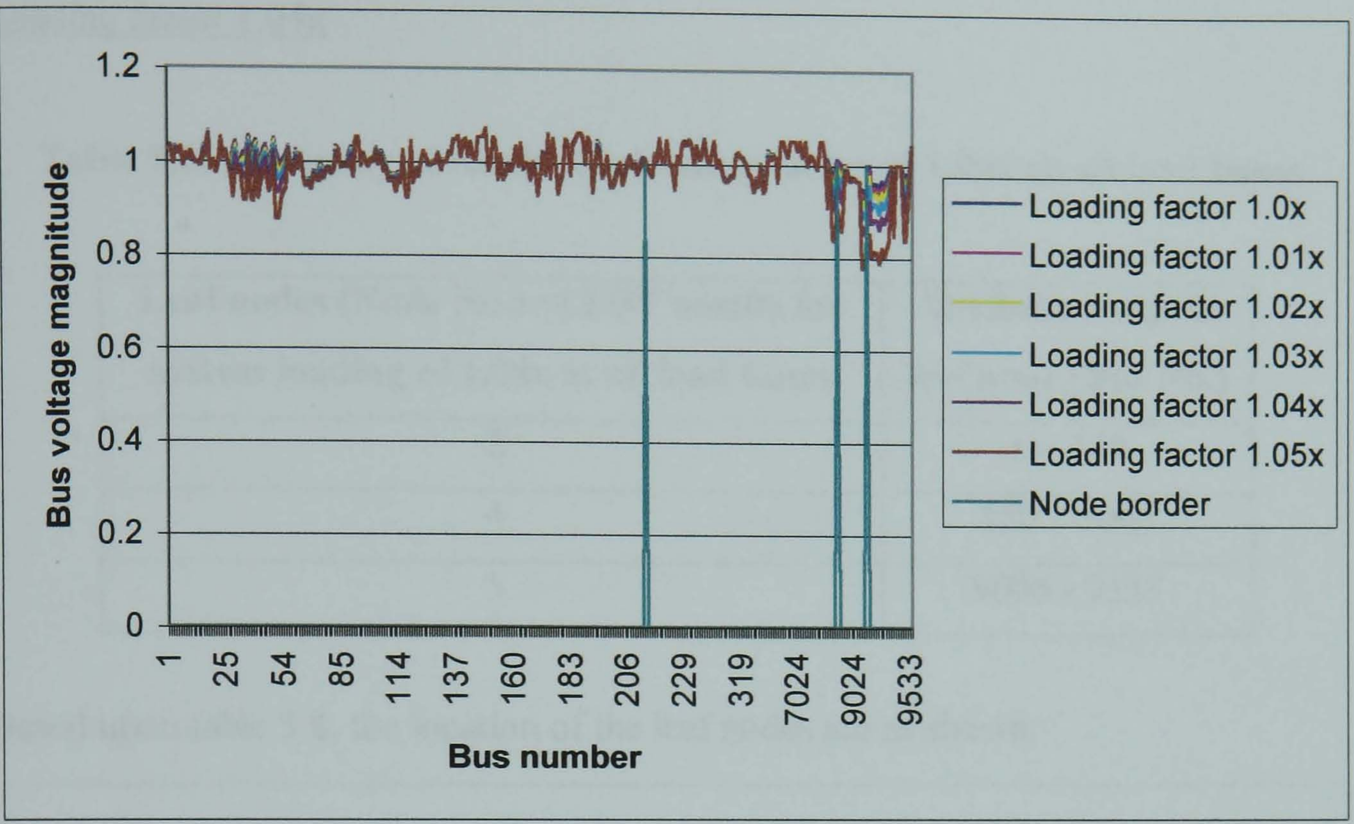


Figure 5.17a: Leaf node location for all load bus loading

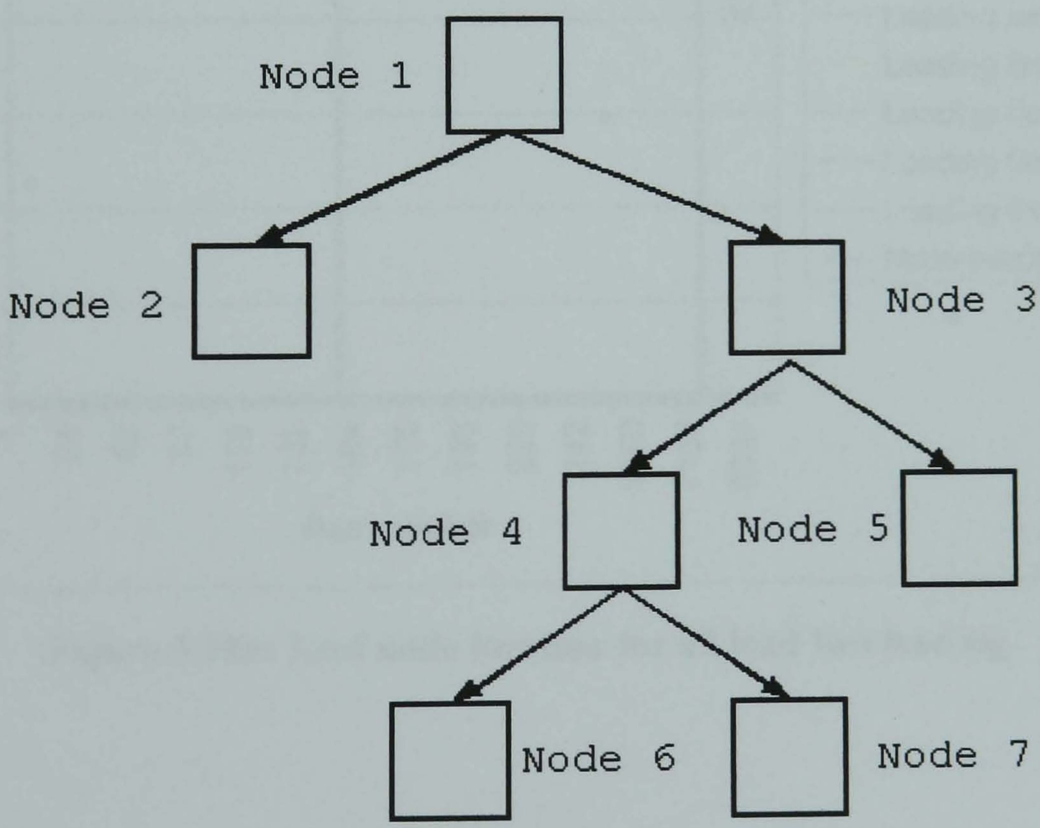


Figure 5.17b: Nodal map of FDT

Loading factor 1.04x

Table 5.8: Summary of results for loading factor of 1.04x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.04x at all load buses	Attribute range in leaf node (Bus No.)
2	1 – 149
4	150 – 9005
5	9006 - 9533

Based upon table 5.8, the location of the leaf nodes are as shown:

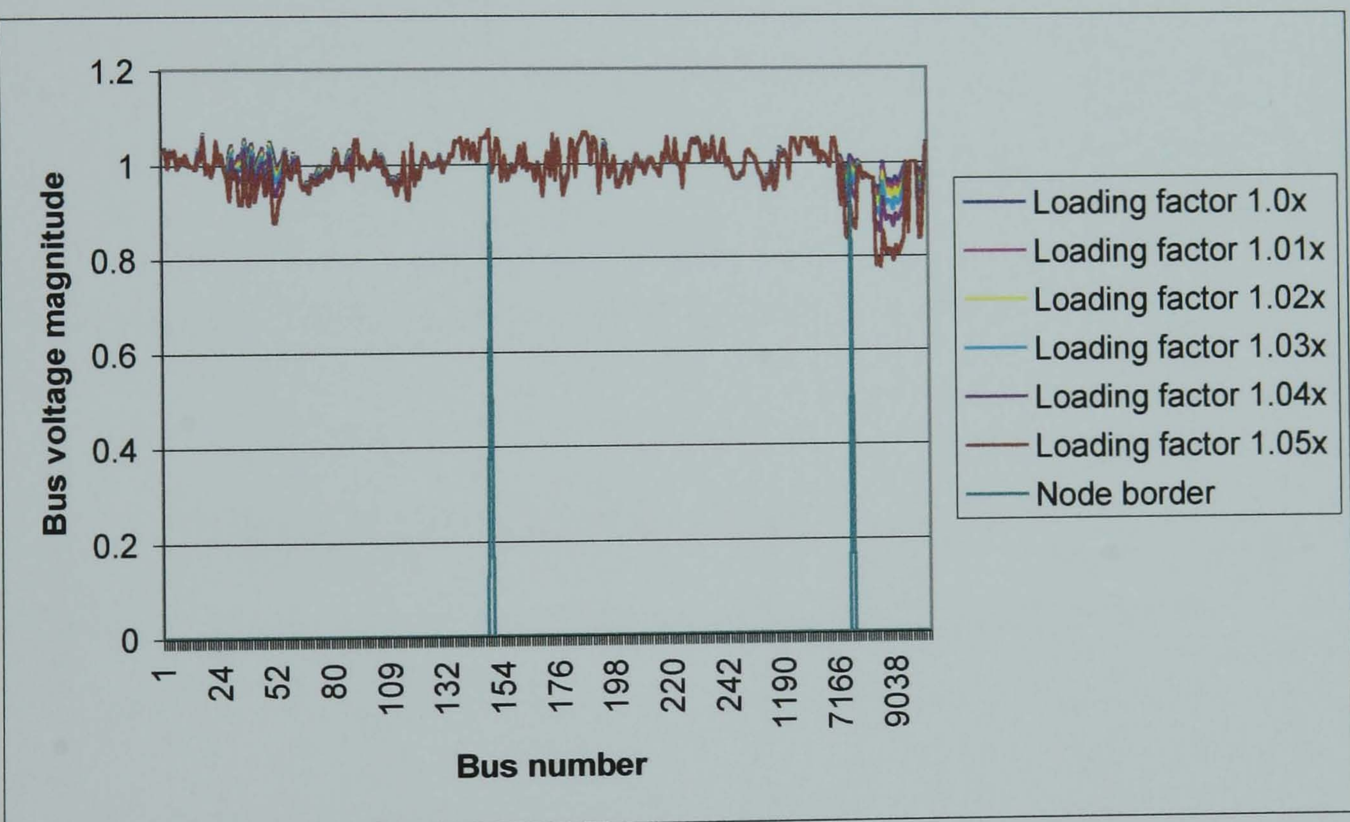


Figure 5.18a: Leaf node location for all load bus loading

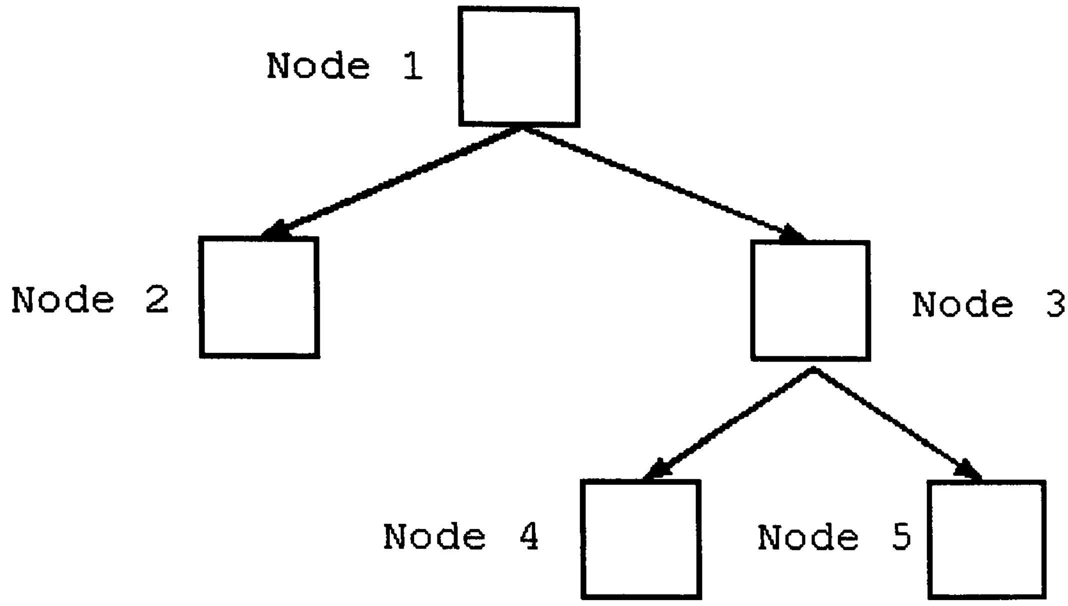


Figure 5.18b: Nodal map of FDT

Loading factor 1.05x

Table 5.9: Summary of results for loading factor of 1.05x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.05x at all load buses	Attribute range in leaf node (Bus No.)
16	1 – 19
17	20 – 60
15	61 – 85
13	86 – 98
9	99 – 107
5	108 – 247
6	248 – 526
10	528 – 9005
11	9006 – 9533

Based upon table 5.9, the location of the leaf nodes are as shown:

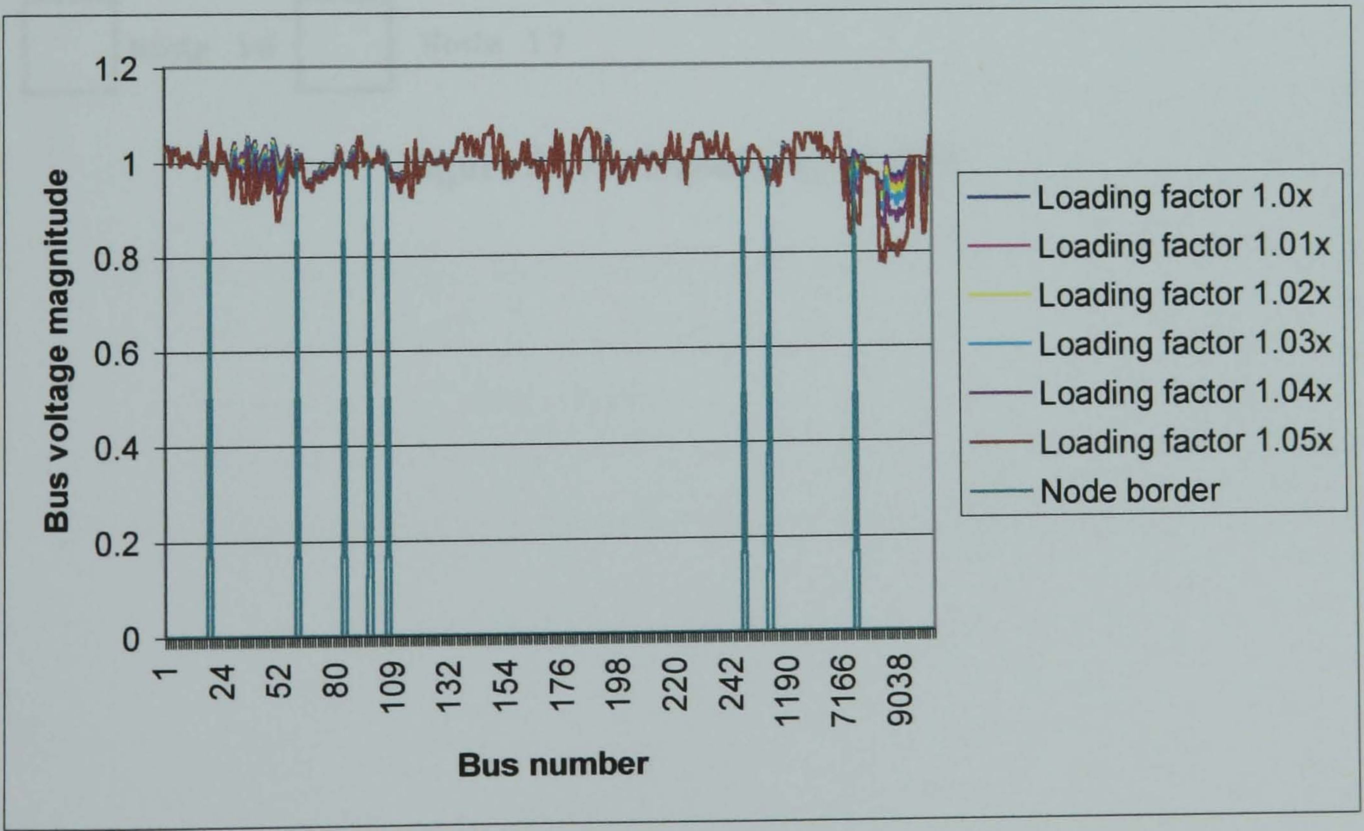


Figure 5.19a: Leaf node location for all load bus loading

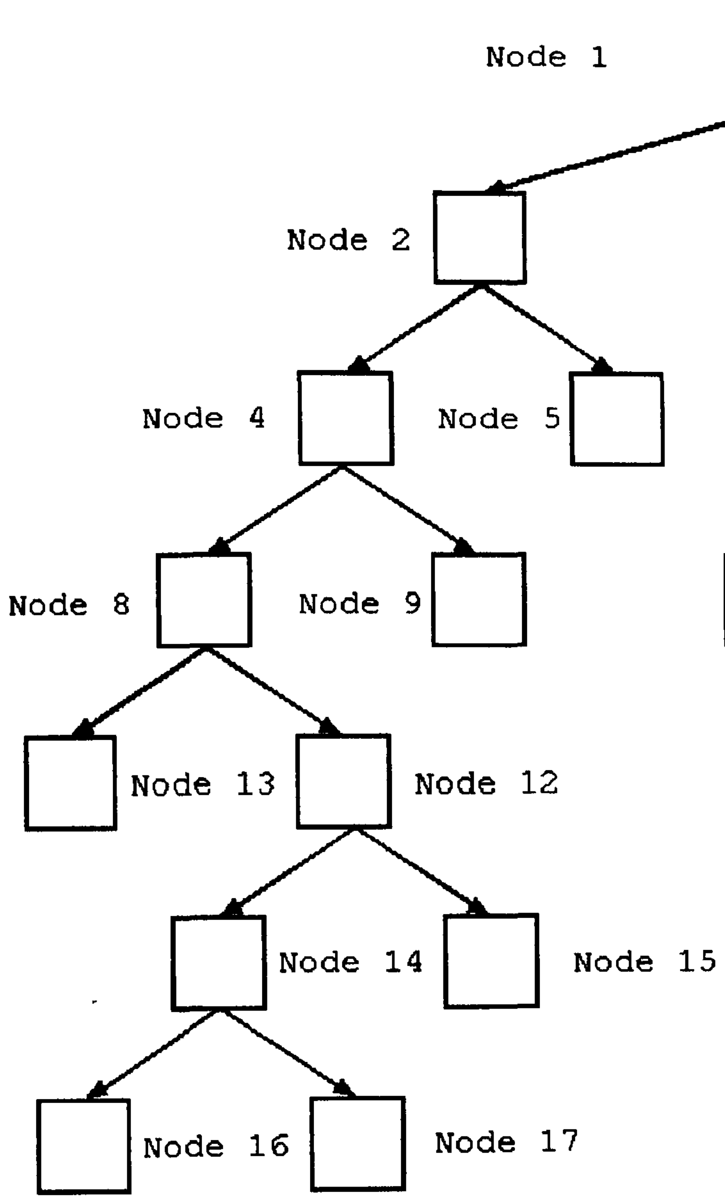


Figure 5.19b: Nodal map of FDT

5.15.2 Heavily loaded bus loading strategy

Loading factor 1.1x

Table 5.10: Summary of results for loading factor of 1.1x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.1x at heavily loaded buses	Attribute range in leaf node (Bus No.)
2	1 – 320
3	322 – 9533

Based upon table 5.10, the location of the leaf nodes are as shown:

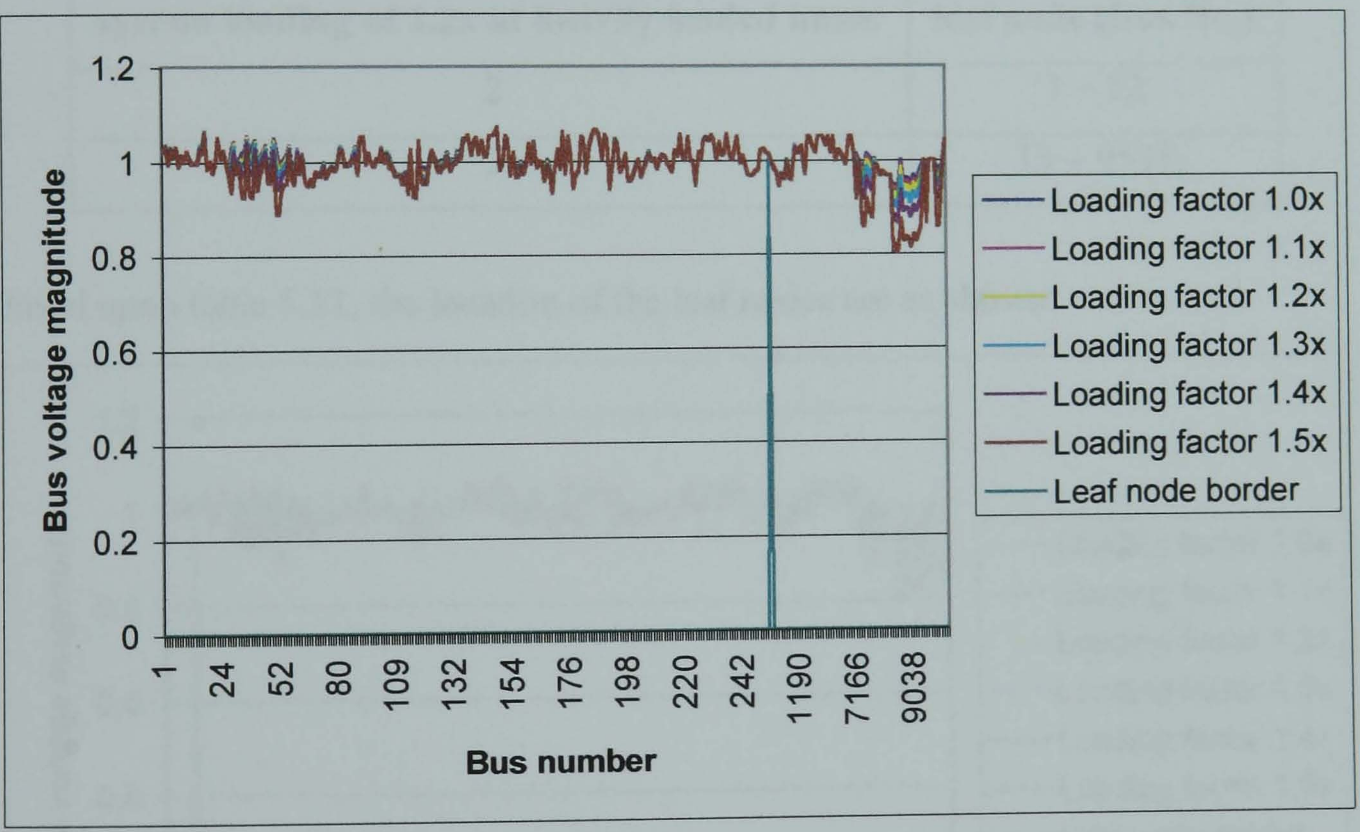


Figure 5.20a: Leaf node location for heavily loaded bus loading strategy

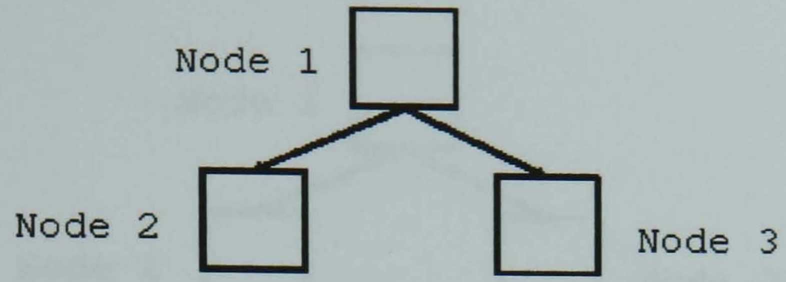


Figure 5.20b: Nodal map of FDT

Loading factor 1.2x

Table 5.11: Summary of results for loading factor of 1.2x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.2x at heavily loaded buses	Attribute range in leaf node (Bus No.)
2	1 – 12
3	13 – 9533

Based upon table 5.11, the location of the leaf nodes are as shown:

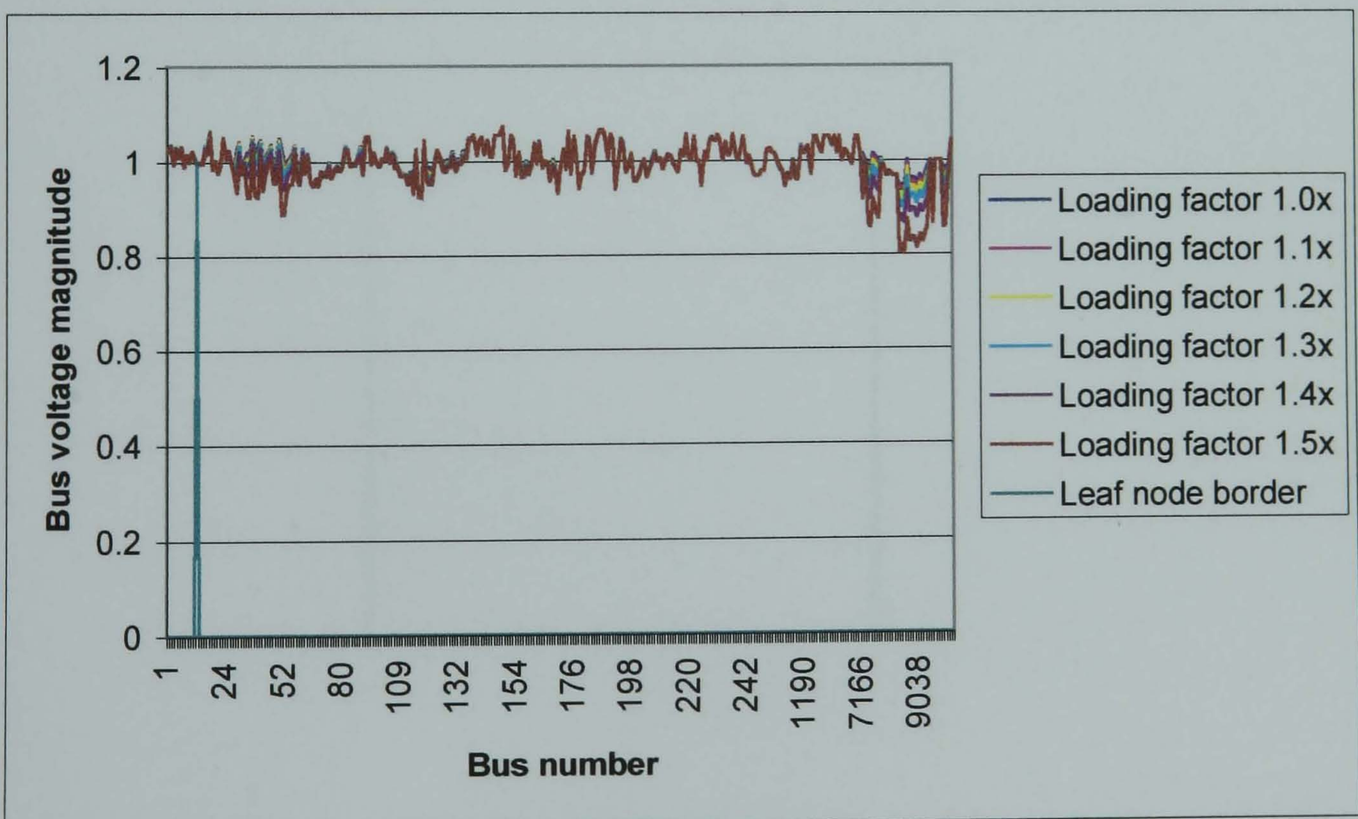


Figure 5.21a: Leaf node location for heavily loaded bus loading strategy

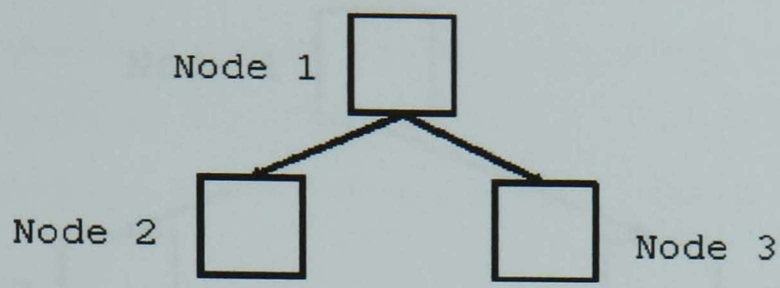


Figure 5.21b: Nodal map of FDT

Loading factor 1.3x

Table 5.12: Summary of results for loading factor of 1.3x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.3x at heavily loaded buses	Attribute range in leaf node (Bus No.)
2	1 – 92
4	94 – 9005
5	9006 – 9533

Based upon table 5.12, the location of the leaf nodes are as shown:

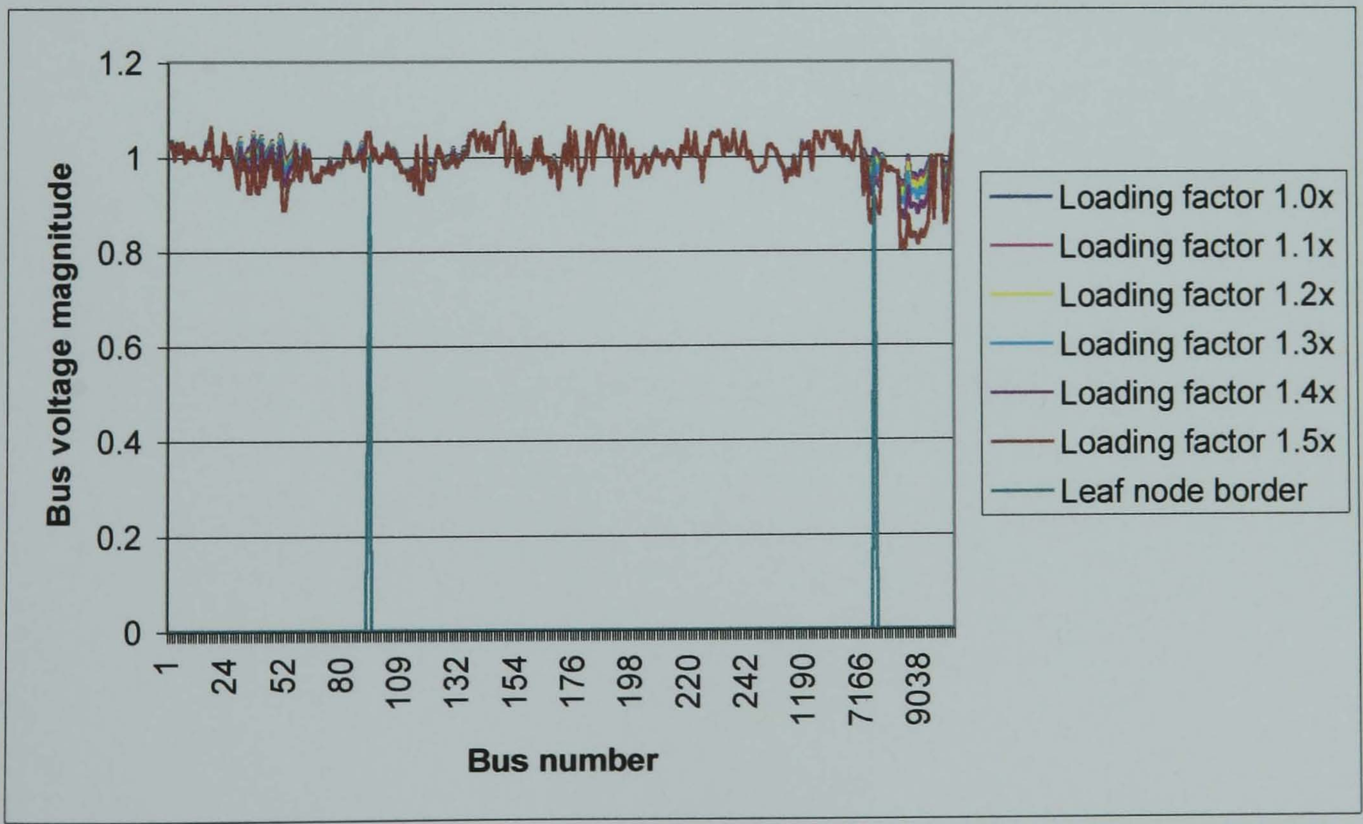


Figure 5.22a: Leaf node location for heavily loaded bus loading strategy

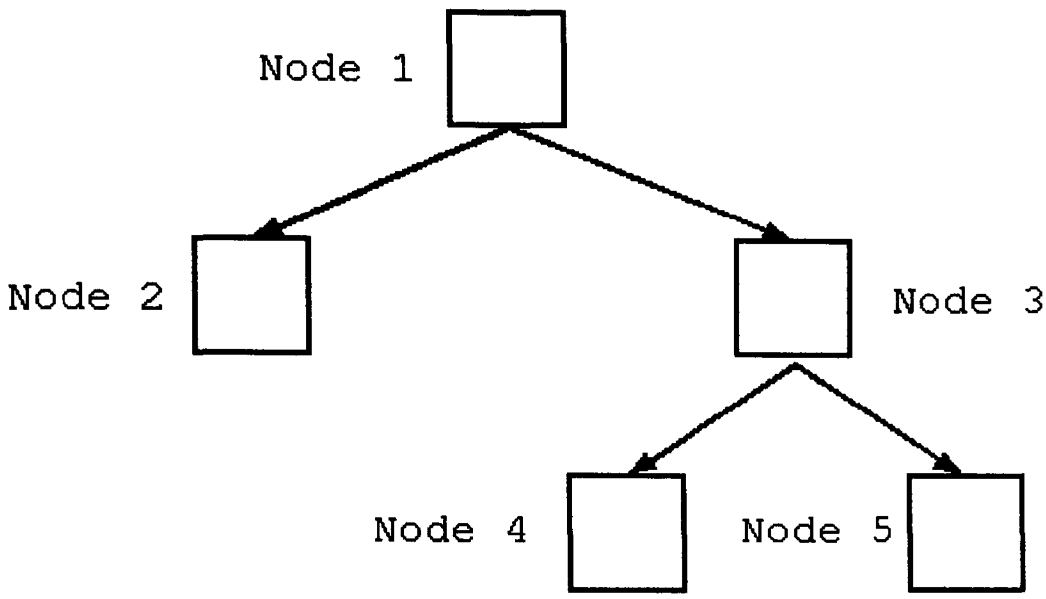


Figure 5.22b: Nodal map of FDT

Loading factor 1.4x

Table 5.13: Summary of results for loading factor of 1.4x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.4x at heavily loaded buses	Attribute range in leaf node (Bus No.)
2	1 – 84
4	85 – 7166
5	9001 – 9533

Based upon table 5.13, the location of the leaf nodes are as shown:

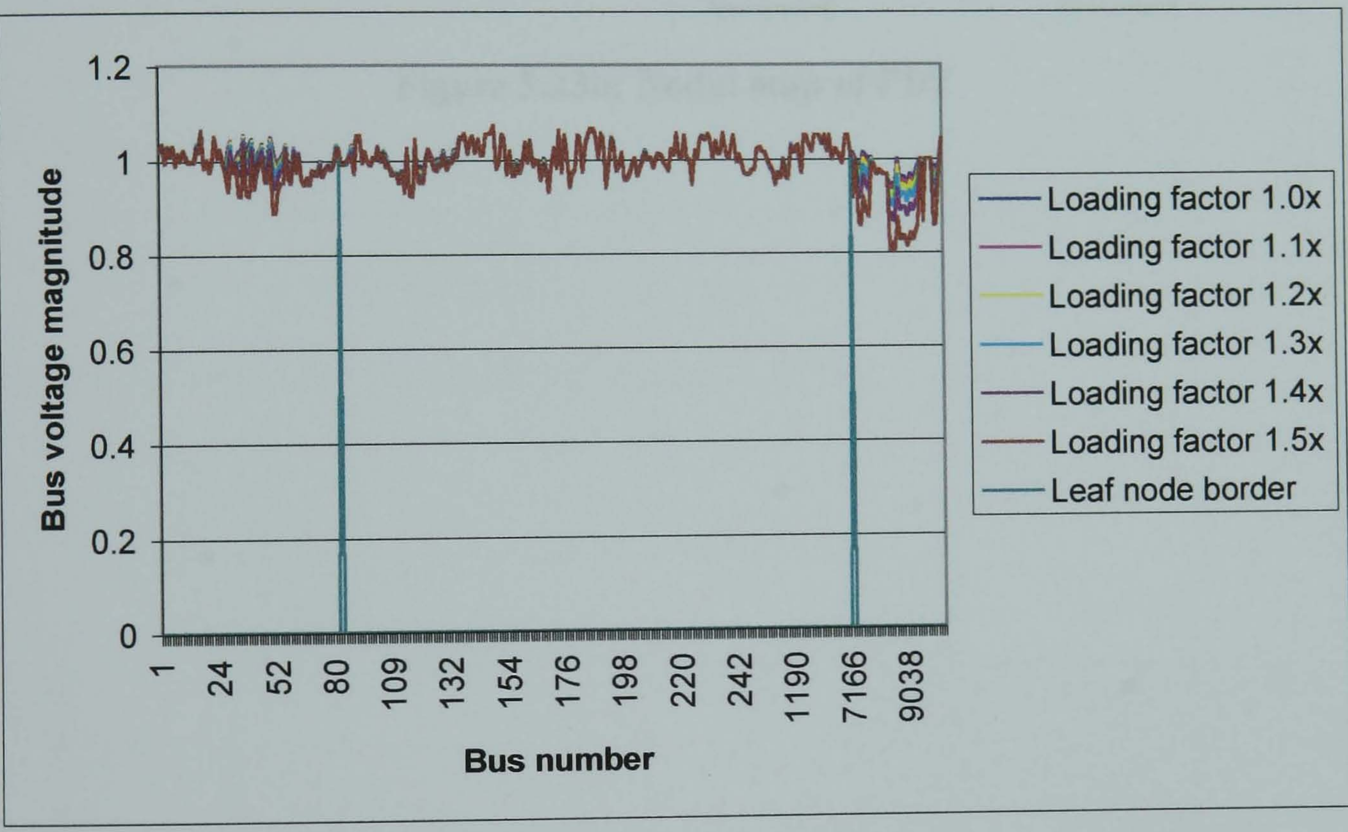


Figure 5.23a: Leaf node location for heavily loaded bus loading strategy

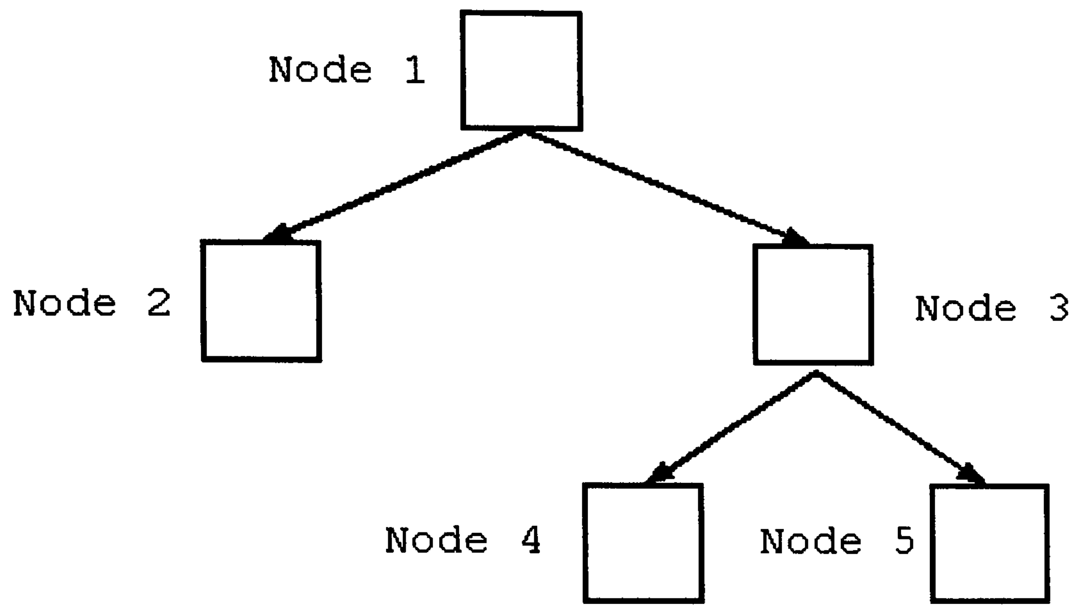


Figure 5.23b: Nodal map of FDT

Loading factor 1.5x

Table 5.14: Summary of results for loading factor of 1.5x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.5x at heavily loaded buses	Attribute range in leaf node (Bus No.)
12	1 – 27
13	33 – 85
9	86 – 161
5	162 – 179
6	180 – 247
10	248 – 9023
11	9024 – 9533

Based upon table 5.14, the location of the leaf nodes are as shown:

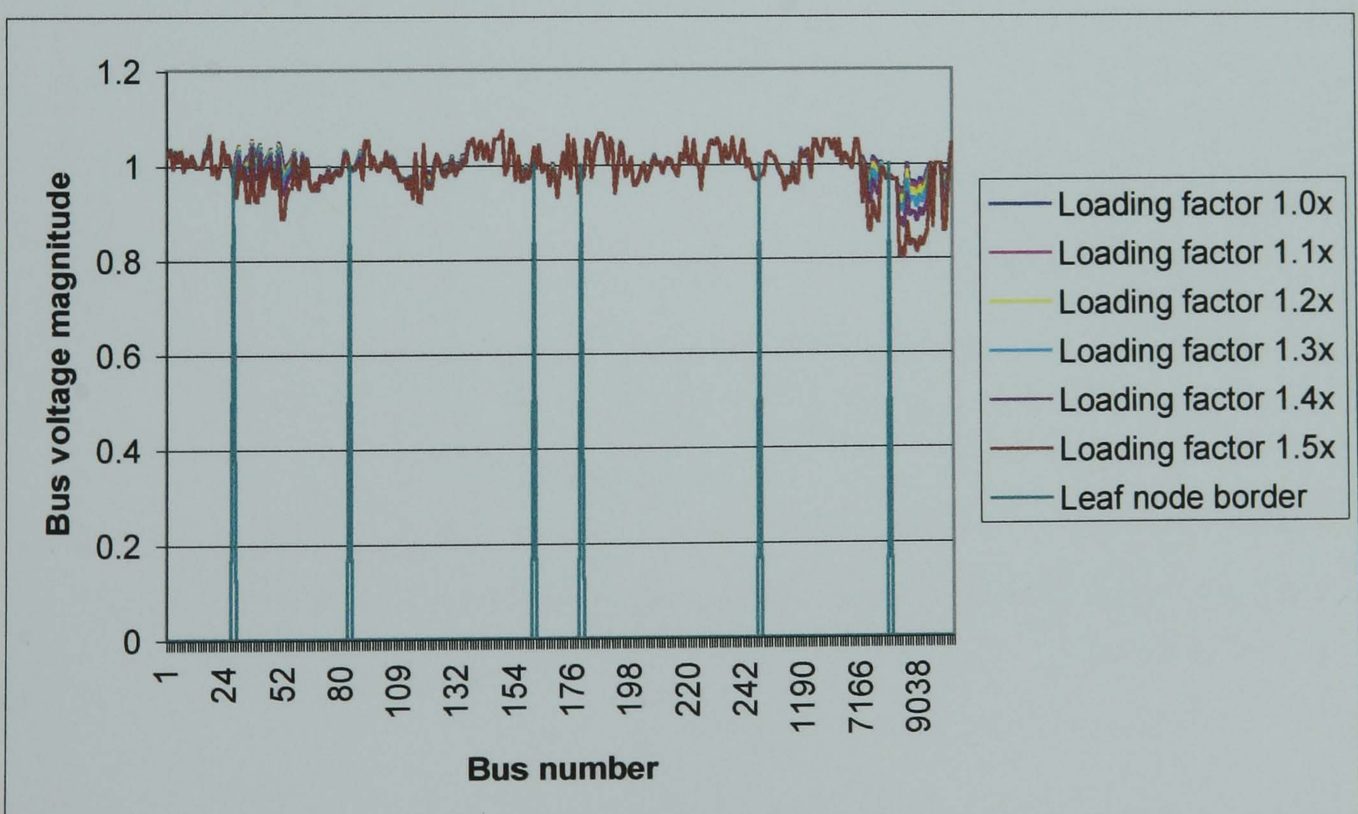


Figure 5.24a: Leaf node location for heavily loaded bus loading strategy

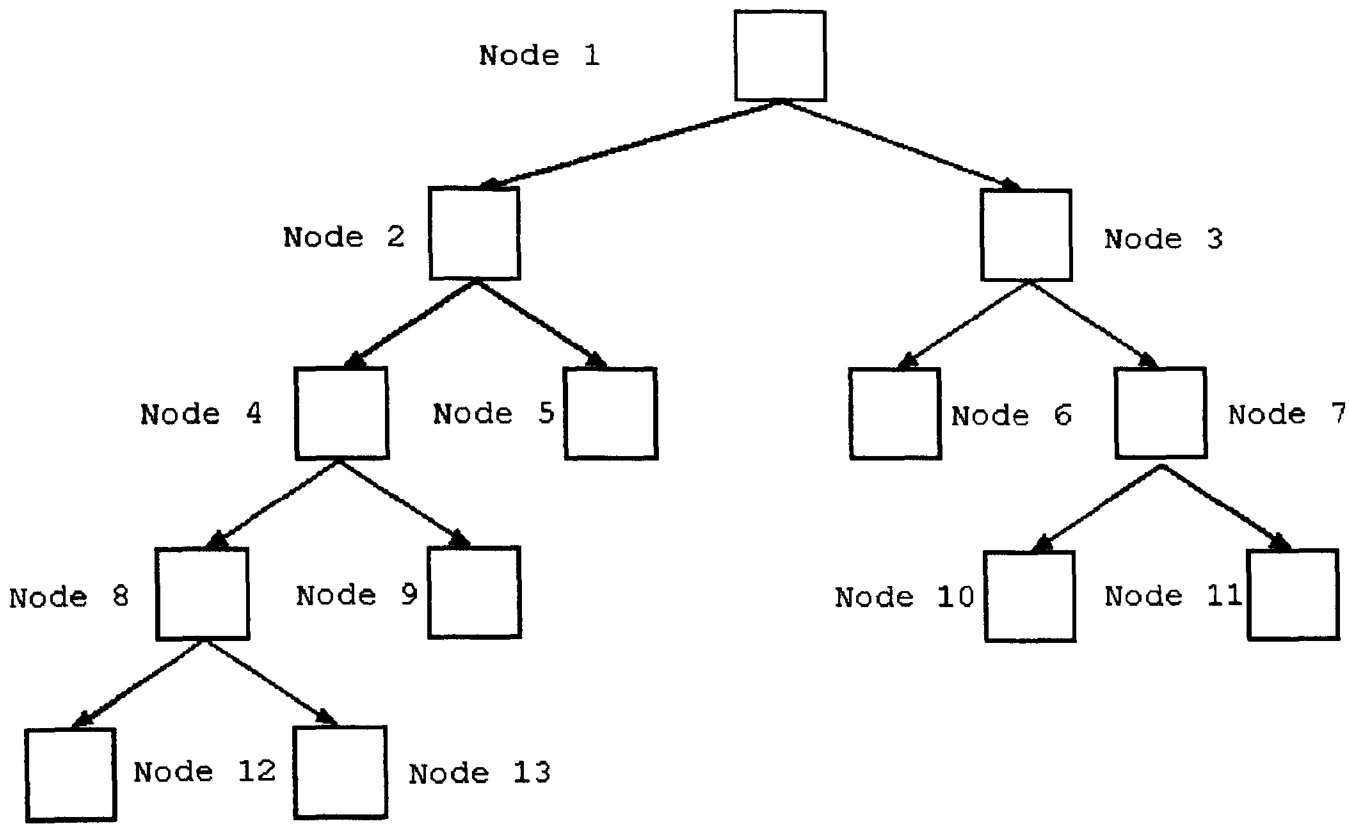


Figure 5.24b: Nodal map of FDT

5.15.3 Lightly loaded bus loading strategy

Loading factor 1.4x

Table 5.15: Summary of results for loading factor of 1.4x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.4x at lightly loaded buses	Attribute range in leaf node (Bus No.)
2	1 – 126
3	127 – 9533

Based upon table 5.15, the location of the leaf nodes are as shown:

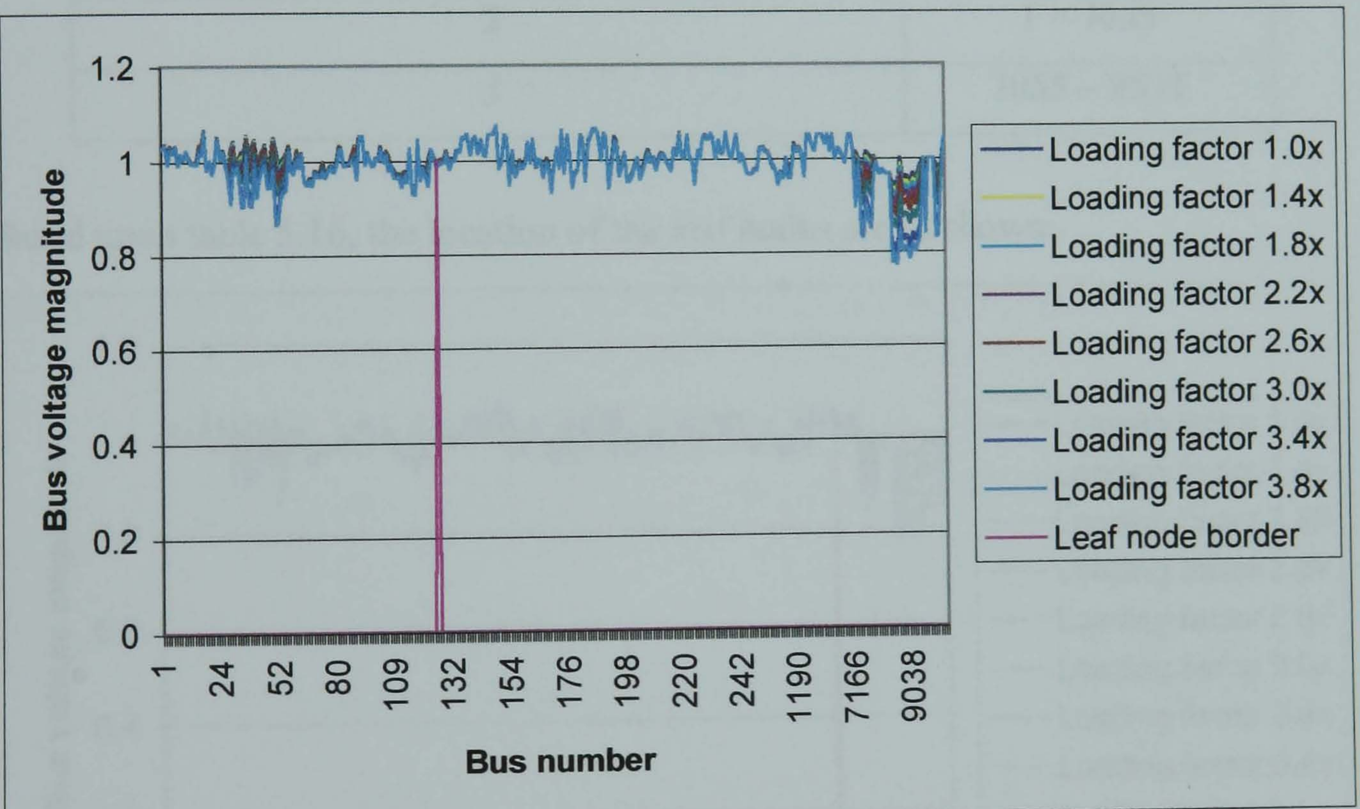


Figure 5.25a: Leaf node location for lightly loaded bus loading strategy

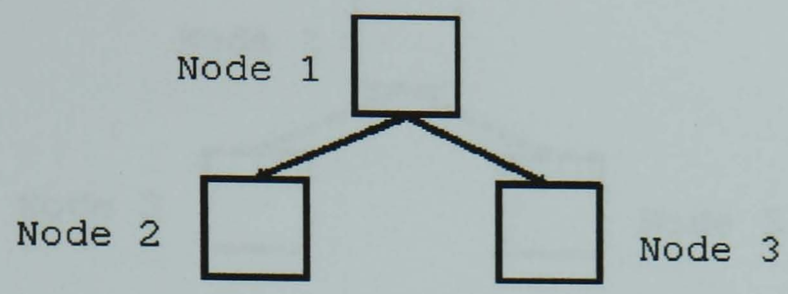


Figure 5.25b: Nodal map of FDT

Loading factor 1.8x

Table 5.16: Summary of results for loading factor of 1.8x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.8x at lightly loaded buses	Attribute range in leaf node (Bus No.)
2	1 – 7049
3	7055 – 9533

Based upon table 5.16, the location of the leaf nodes are as shown:

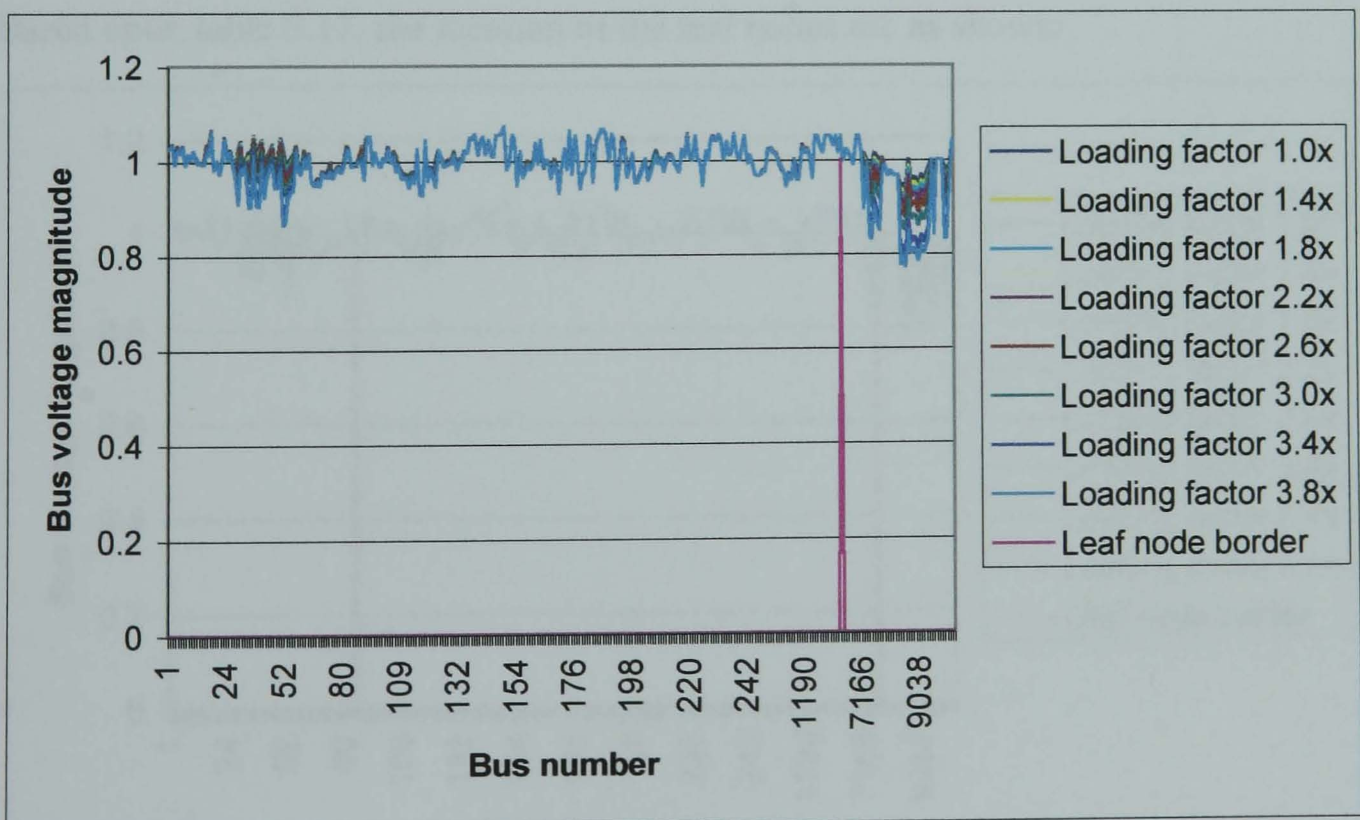


Figure 5.26a: Leaf node location for lightly loaded bus loading strategy

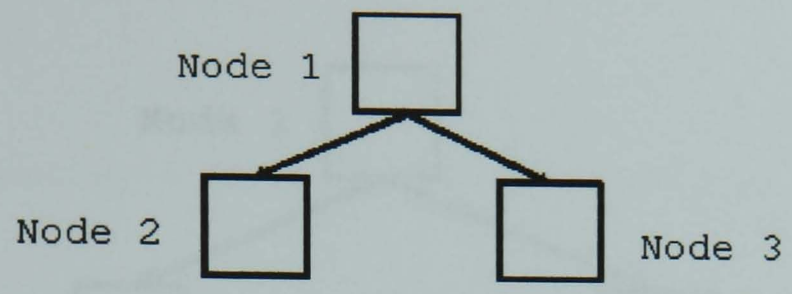


Figure 5.26b: Nodal map of FDT

Loading factor 2.2x

Table 5.17: Summary of results for loading factor of 2.2x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 2.2x at lightly loaded buses	Attribute range in leaf node (Bus No.)
2	1 – 86
4	87 – 9005
5	9006 – 9533

Based upon table 5.17, the location of the leaf nodes are as shown:

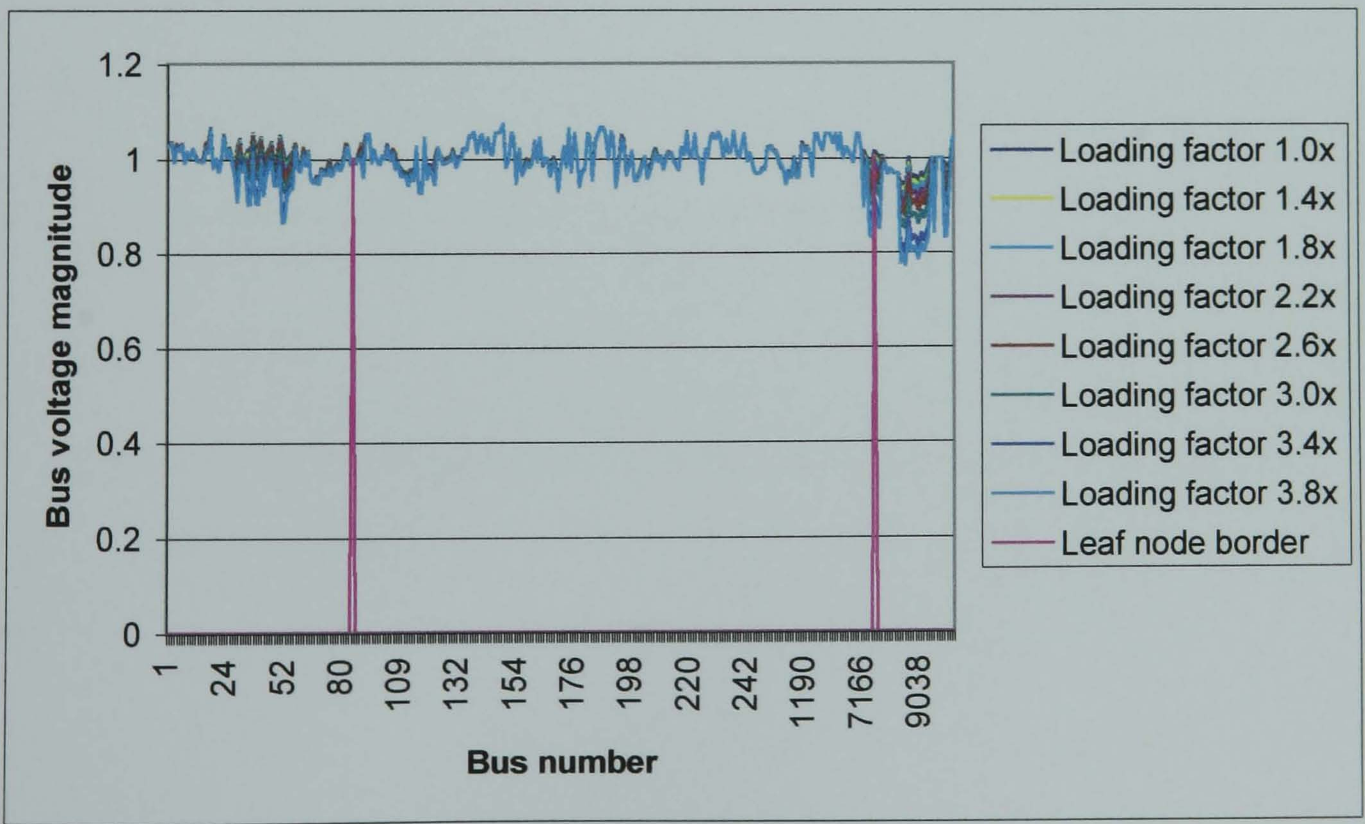


Figure 5.27a: Leaf node location for lightly loaded bus loading strategy

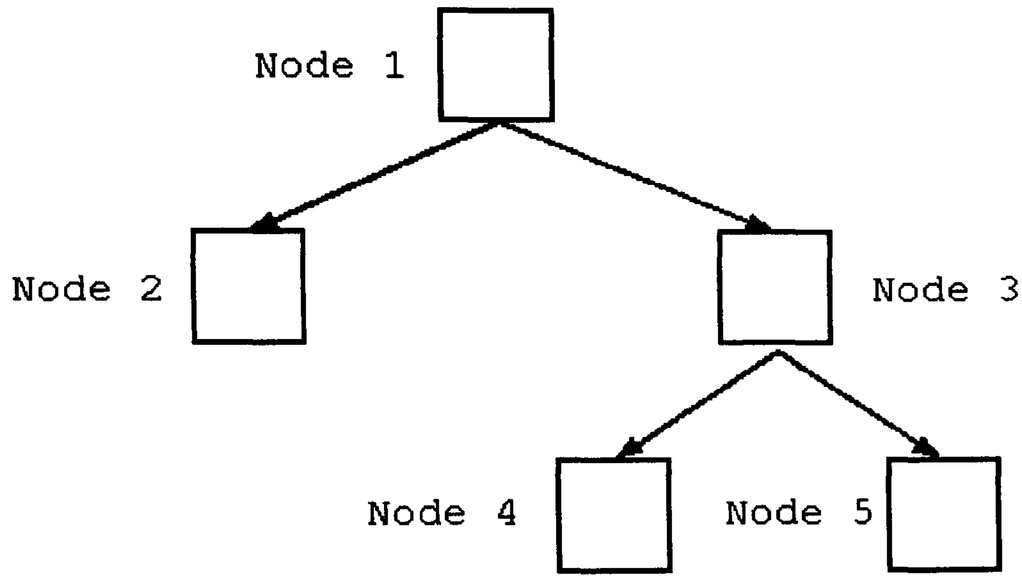


Figure 5.27b: Nodal map of FDT

Loading factor 2.6x

Table 5.18: Summary of results for loading factor of 2.6x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 2.6x at lightly loaded buses	Attribute range in leaf node (Bus No.)
2	1 – 107
4	108 – 7166
5	9001 – 9533

Based upon table 5.18, the location of the leaf nodes are as shown:

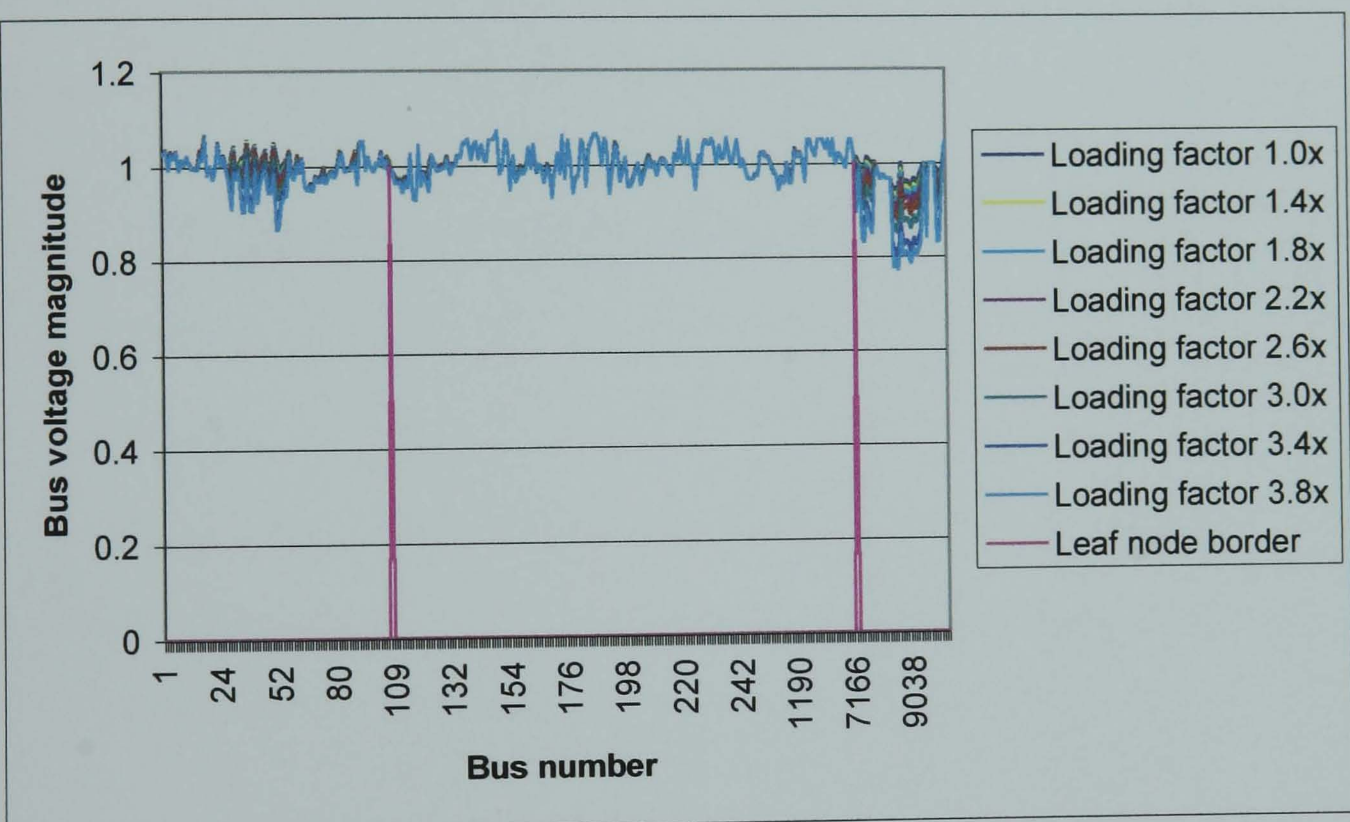


Figure 5.28a: Leaf node location for lightly loaded bus loading strategy

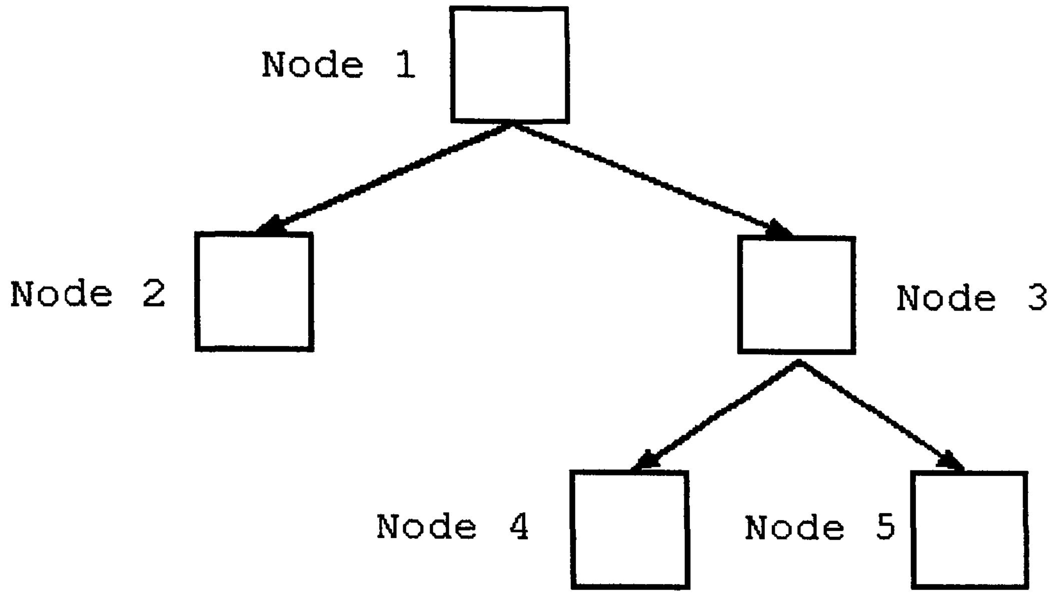


Figure 5.28b: Nodal map of FDT

Loading factor 3.0x

Table 5.19: Summary of results for loading factor of 3.0x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 3.0x at lightly loaded buses	Attribute range in leaf node (Bus No.)
2	1 – 149
4	150 – 9005
5	9006 – 9533

Based upon table 5.19, the location of the leaf nodes are as shown:

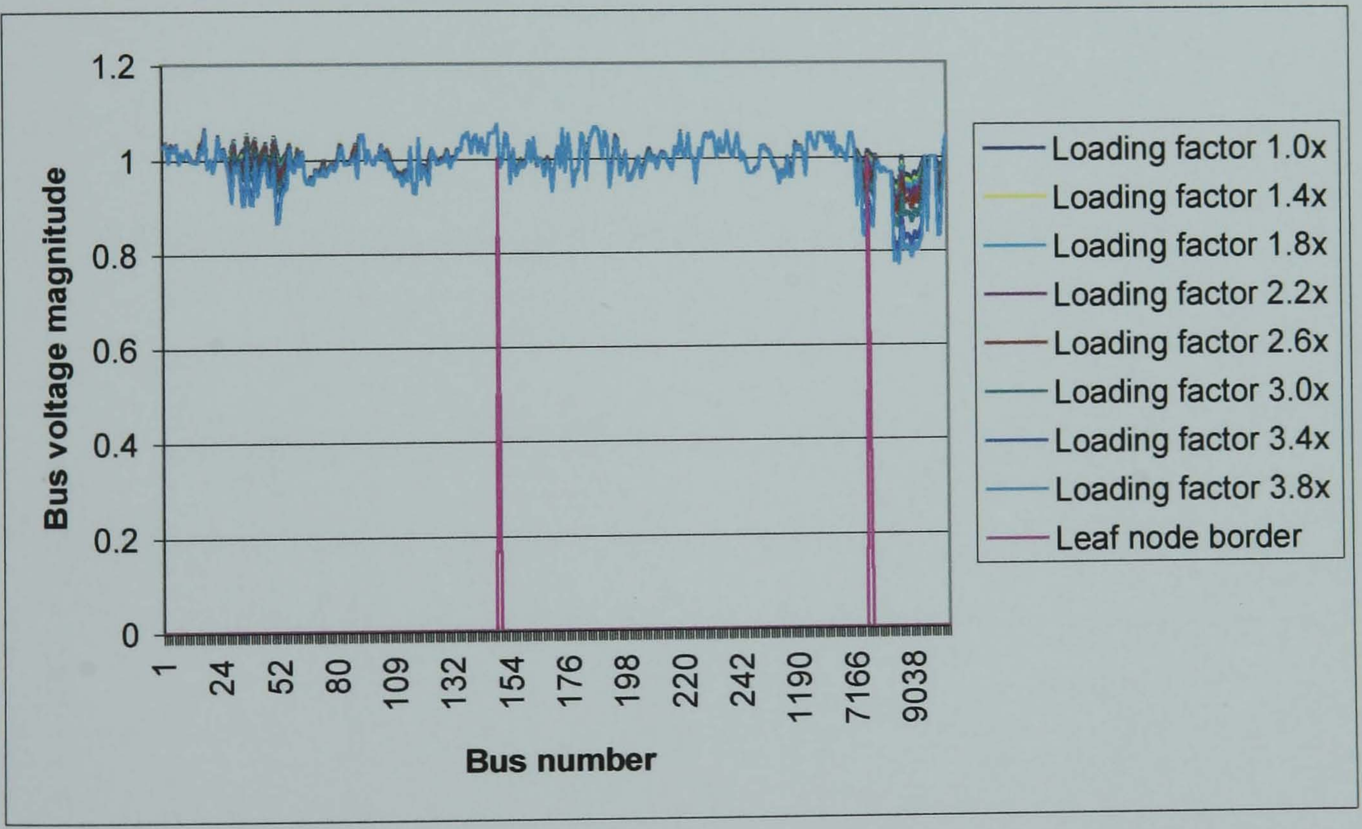


Figure 5.29a: Leaf node location for lightly loaded bus loading strategy

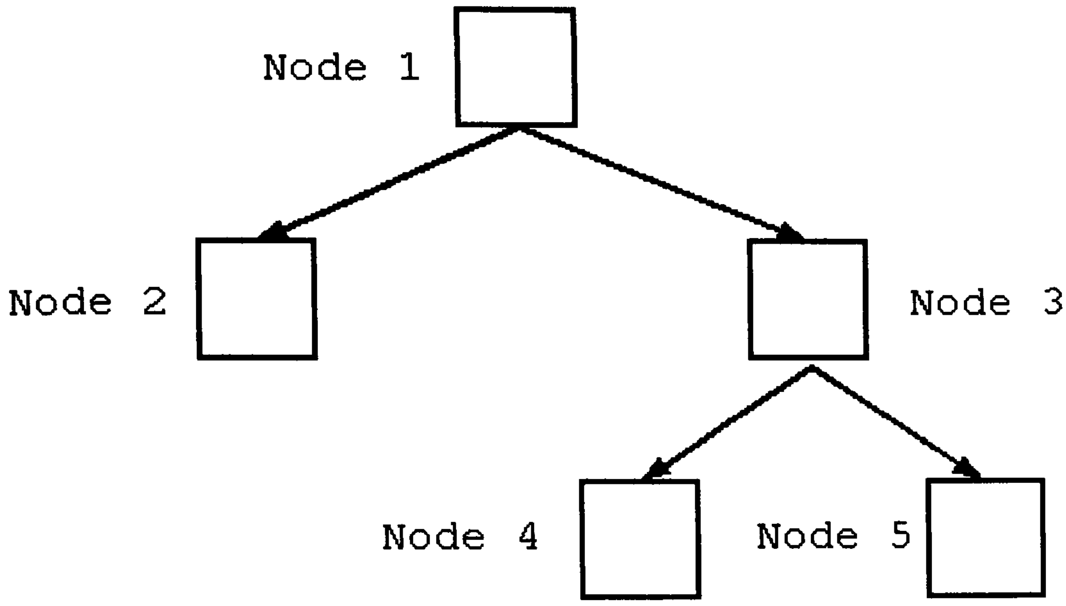


Figure 5.29b: Nodal map of FDT

Loading factor 3.4x

Table 5.20: Summary of results for loading factor of 3.4x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 3.4x at lightly loaded buses	Attribute range in leaf node (Bus No.)
14	1 – 43
15	44 – 85
13	86
9	87 – 92
5	94 – 209
6	210 – 552
10	562 – 9023
11	9024 - 9533

Based upon table 5.20, the location of the leaf nodes are as shown:

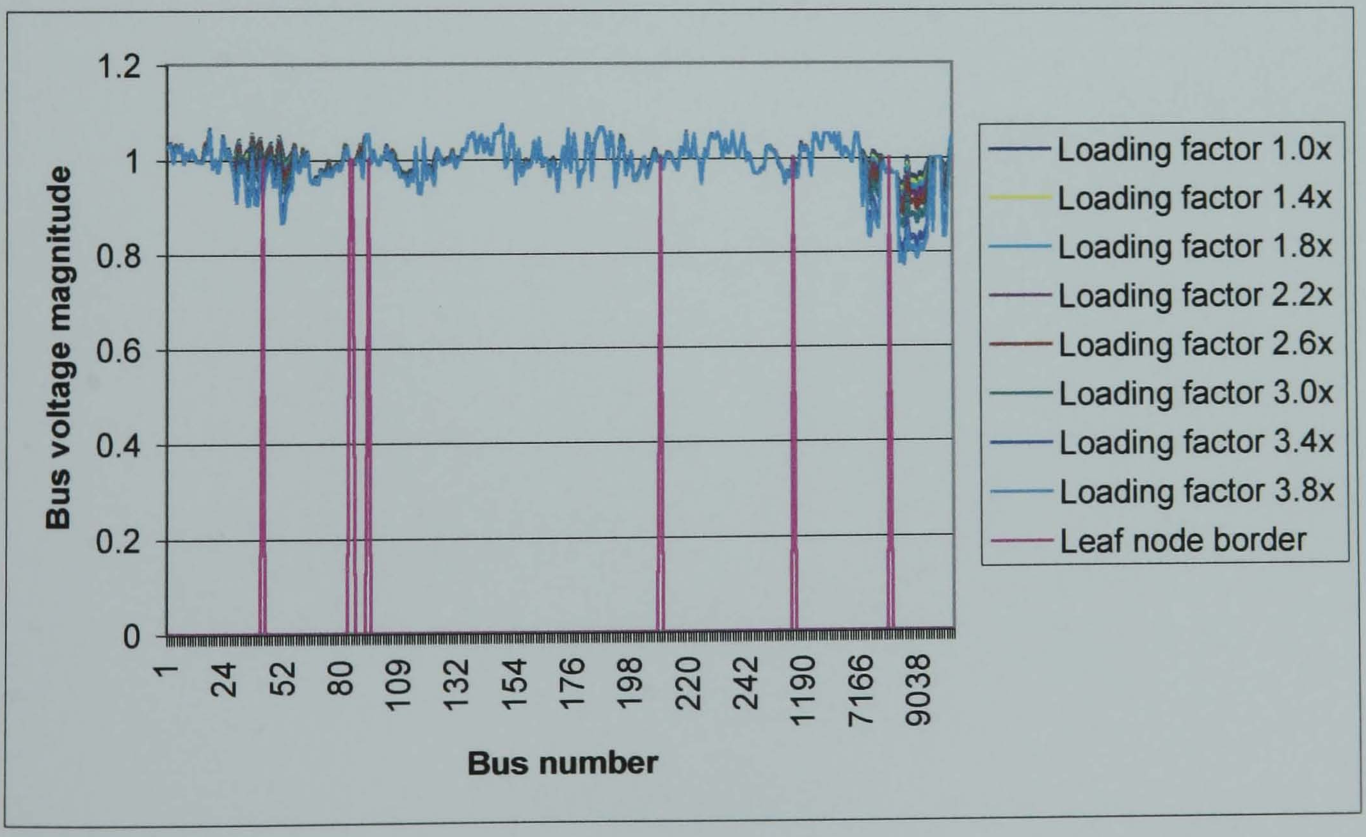


Figure 5.30a: Leaf node location for lightly loaded bus loading strategy

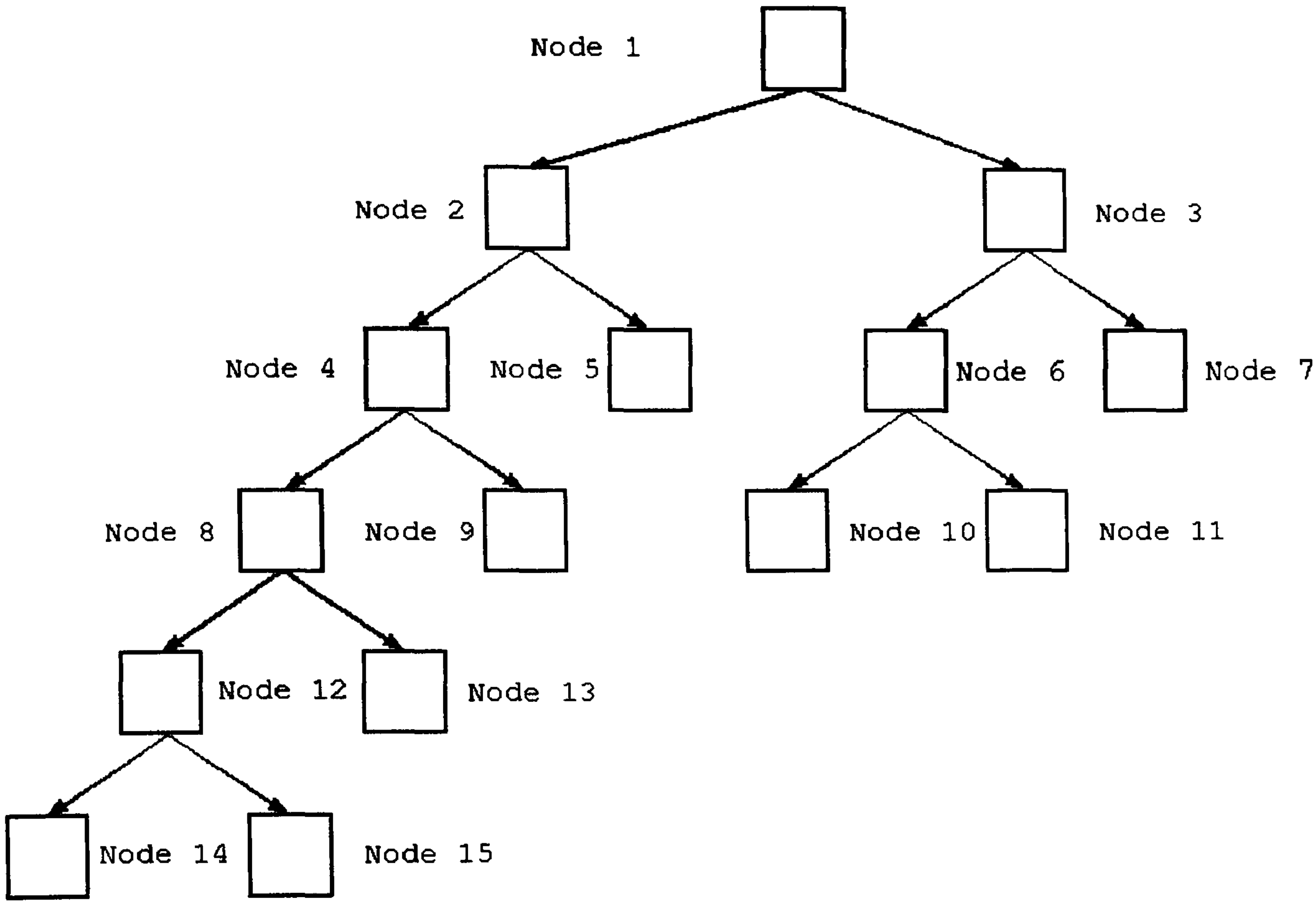


Figure 5.30b: Nodal map of FDT

Loading factor 3.8x

Table 5.21: Summary of results for loading factor of 3.8x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 3.8x at lightly loaded buses	Attribute range in leaf node (Bus No.)
14	1 – 19
15	20 – 60
13	61 – 73
9	74 – 84
5	85 – 248
6	249 – 1201
10	2040 – 9005
11	9006 - 9533

Based upon table 5.21, the location of the leaf nodes are as shown:

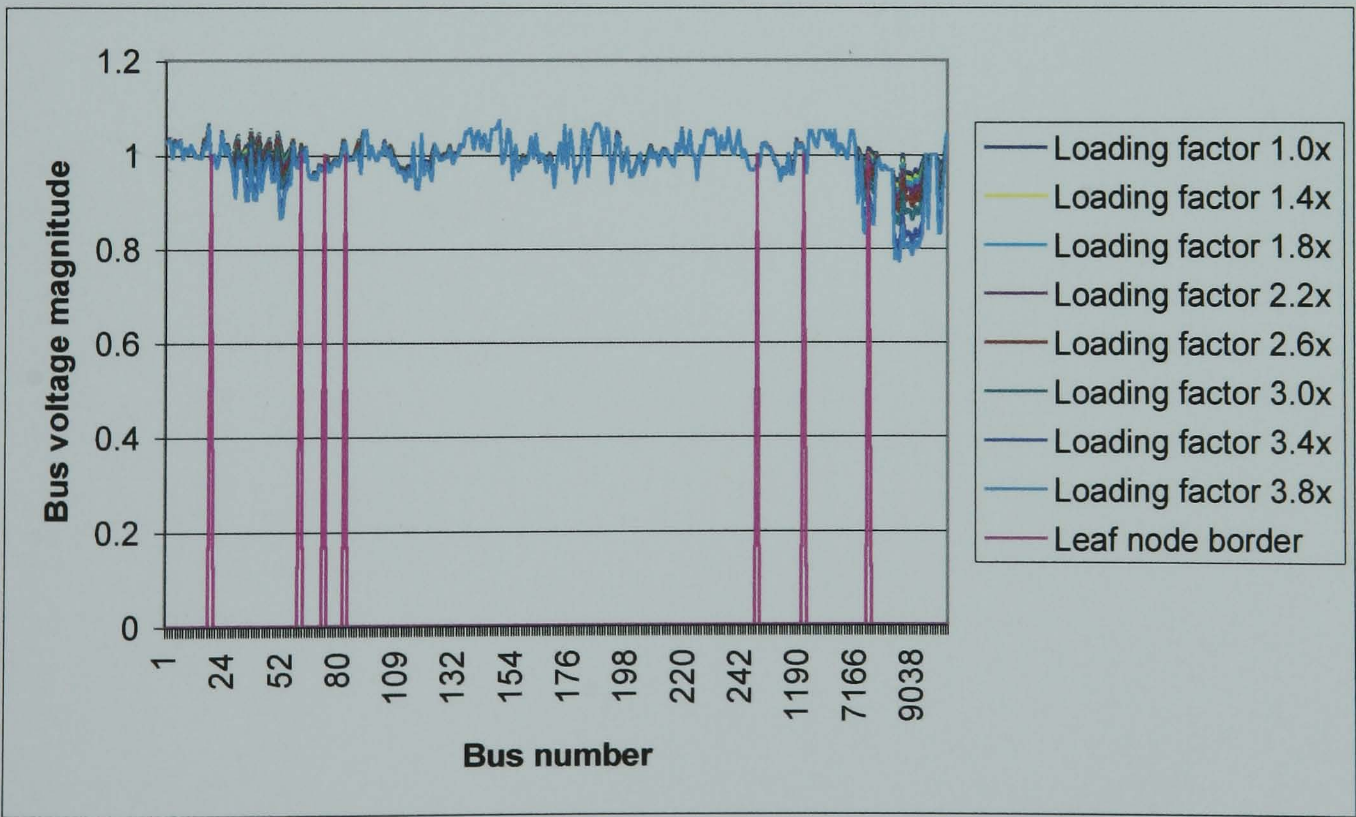


Figure 5.31a: Leaf node location for lightly loaded bus loading strategy

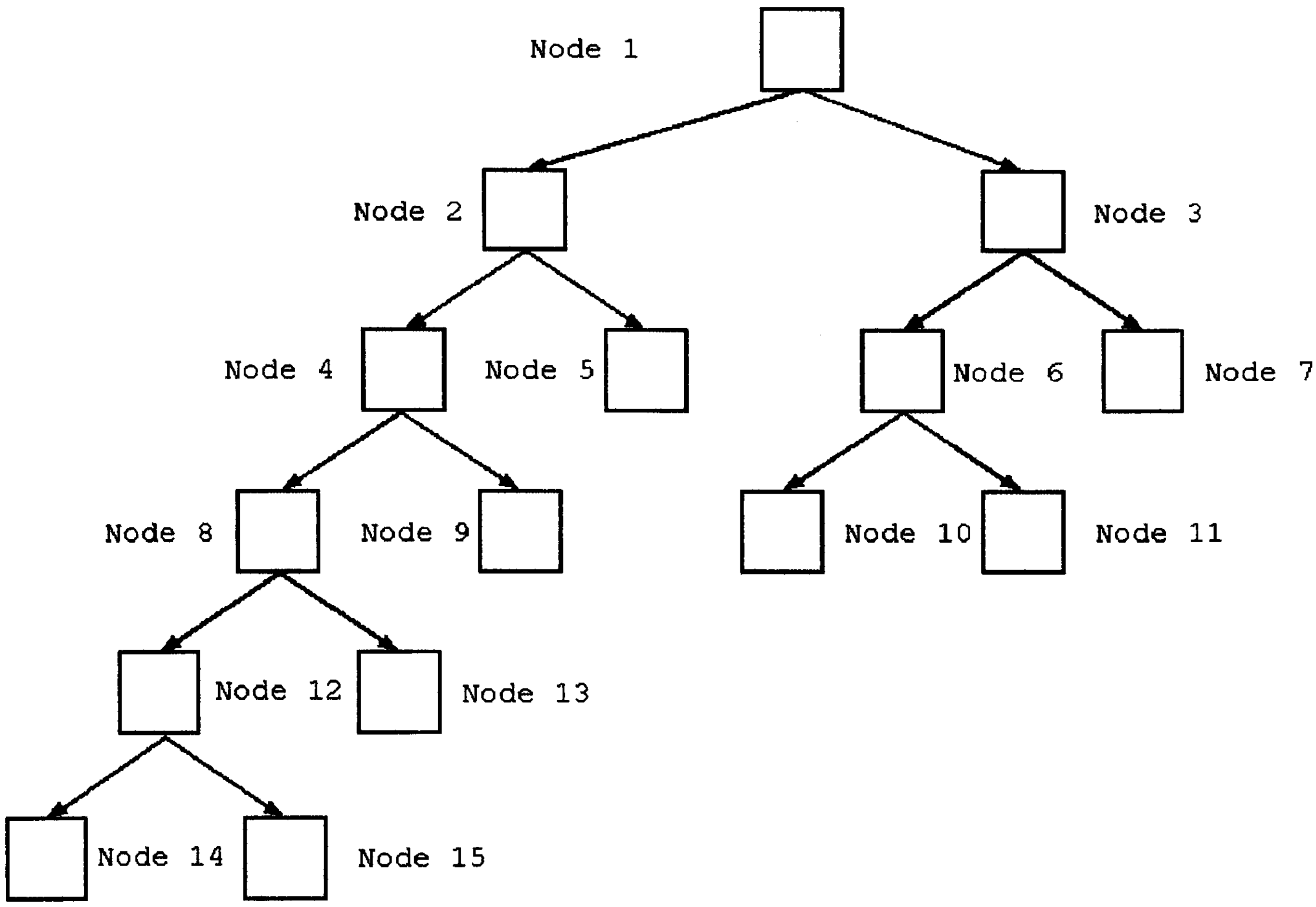


Figure 5.31b: Nodal map of FDT

Discussion

In general, the FDT shows significant partitioning results at medium to high loading factors. This can be seen in all 3 loading strategies. Considering the fact that the FDT does not utilise any pruning algorithm, the performance of the FDT is very good since the amount of leaf nodes is sufficient and does cover strong and weak bus areas. Although there is an instance of over-partitioning which can be seen in the results for loading factor of 3.4x in the 3rd system loading strategy. But looking at the FDT performance in general shows that the over-partitioning issue is not critical.

Another point that can be observed is in terms of the leaf nodes location. As loading factors increased, the location of leaf nodes changes. This gives the indication that for a big and highly meshed system, the loading level of each bus will experience non-linear changes. This fits nicely with the voltage collapse phenomenon where the mechanism of it is highly complex. However, taking into account of the system behaviour volatility, the FDT is still capable of highlighting weak buses as shown in the test results.

The next test will compare the FDT performance using the conceptual KS formula explained earlier in this chapter.

5.16 Test 4: Application to IEEE 300 bus system using FDT ver 4.0

Table 5.22: Test 4 Program summary:

Version of FDT	FDT ver 4
KS technique	Using conceptual KS formulation using bus voltage values (of the IEEE 300 bus test system at a loading factor of 1.0x at all load buses) as reference
Partitioning technique	Comparing the location of the maximum and minimum KS score. • If theses scores are located at both ends of the test nodes, partitioning will stop. • If the maximum KS score is on the extreme ends of the test node, and the location of the minimum KS score is not located at the extreme ends of the test node, the partitioning limit will be based upon the location of the minimum KS score. • If the maximum KS score is not at extreme ends, partitioning will be based upon the maximum KS location as a partitioning limit. The location of the KS scores are considered to be on the extreme ends if is located within the range of 10 attributes from each end.

Stop split rule	The rules for the stop split stage are based upon the purity of the node and the minimum population (minimum population value is 50 attributes) of the test node. The summary of the rules are: • If the test node consist of all strong buses (minimum attribute value is more than 0.9pu) stop splitting the node and label is as a leaf node. • If the test node consist of a combination of strong buses and weak buses (minimum attribute value is less than 0.9pu). Check the total population of the test node. If it exceeds the minimum population limit continue partitioning. If not, stop partitioning and label the test node as a leaf node.
Pruning technique	No pruning technique is used
Test system used	IEEE 300 bus system using 3 different loading strategies: • Loads in all load bus are incremented • Loads in only heavily loaded buses are incremented Loads in only lightly loaded buses are incremented
Computer used	Intel PIII 1GHz

Results

The results are presented is in the form of leaf node numbers and its location. The location of each leaf node is presented on a general bus voltage magnitude profile with all the loading factors superimposed on it. The reason for this is to highlight the weak buses area (denoted by varying bus voltage magnitude when the loading factor is increased) and how good the FDT partition the data set into sections of strong and weak buses.

5.16.1 All load bus loading strategy

Loading factor 1.01x

Table 5.23: Summary of results for loading factor of 1.01x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.01x at all load buses	Attribute range in leaf node (Bus No.)
2	1 – 159
3	160 – 9533

Based upon table 5.23, the location of the leaf nodes are as shown:

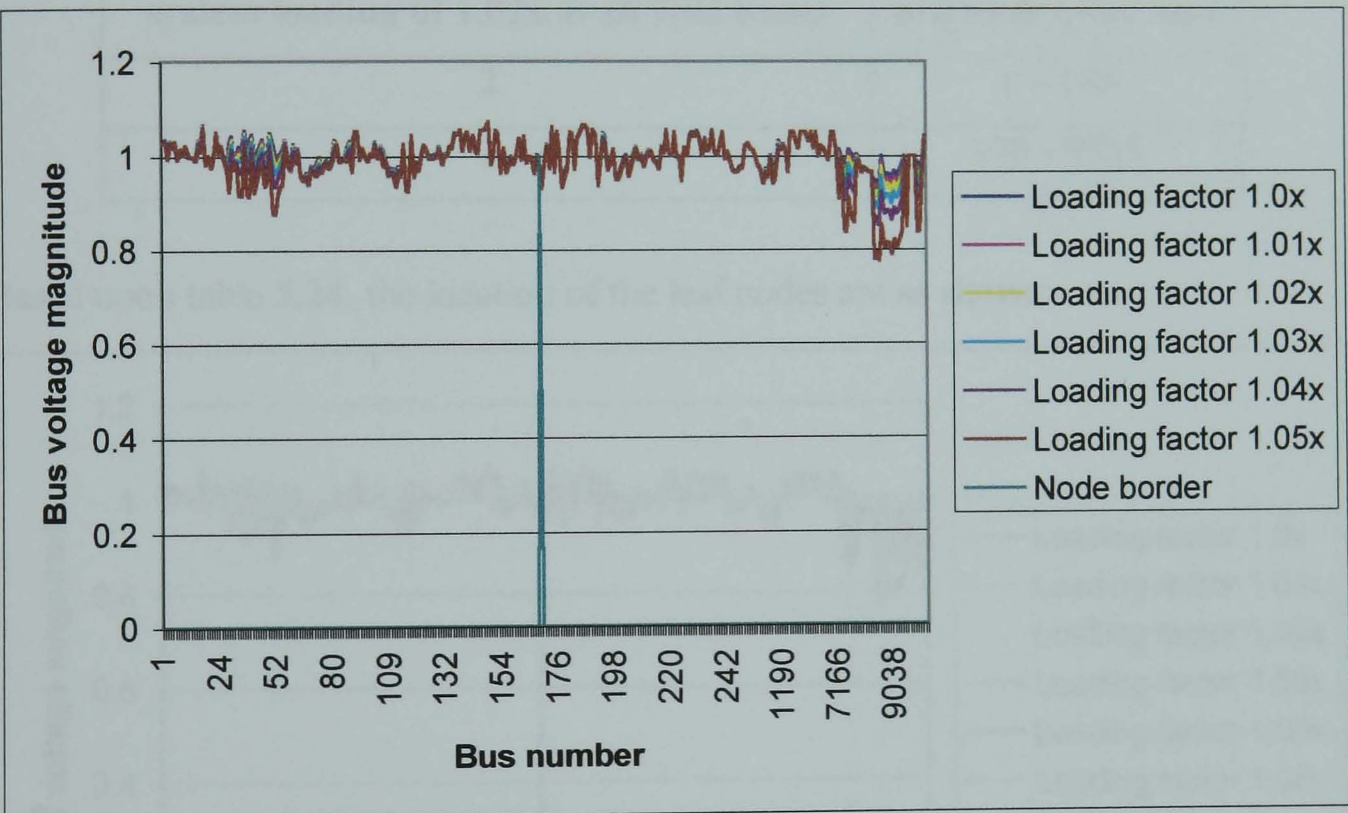


Figure 5.32a: Leaf node location for all load bus loading strategy

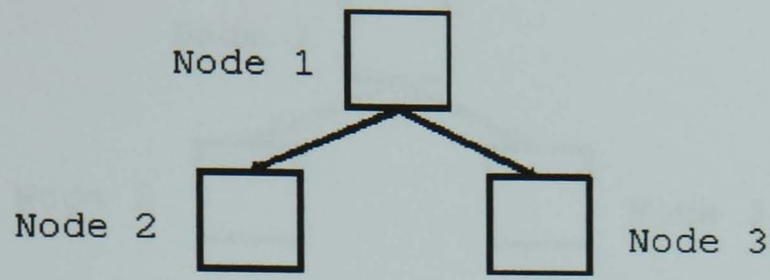


Figure 5.32b: Nodal map of FDT

Loading factor 1.02x

Table 5.24: Summary of results for loading factor of 1.02x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.02x at all load buses	Attribute range in leaf node (Bus No.)
2	1 – 169
3	170 - 9533

Based upon table 5.24, the location of the leaf nodes are as shown:

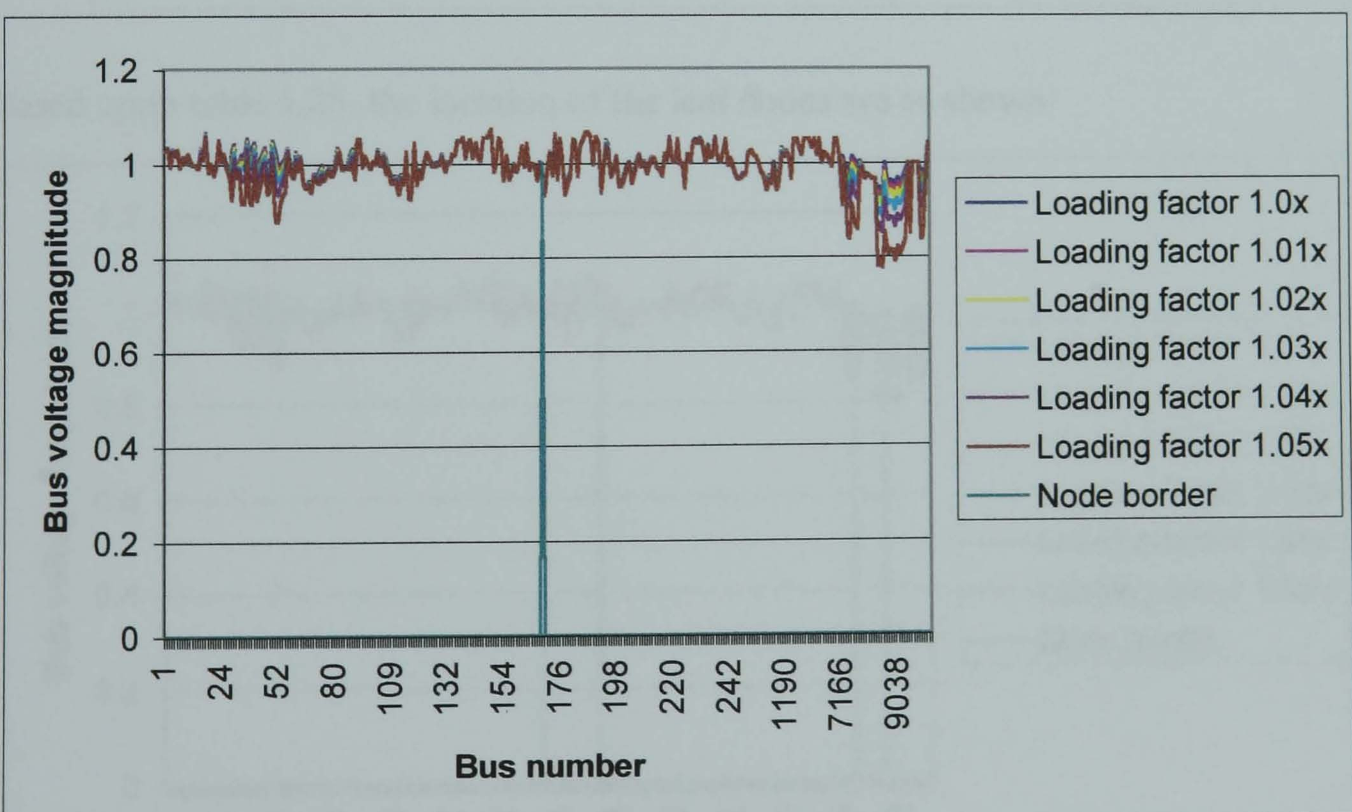


Figure 5.33a: Leaf node location for all load bus loading

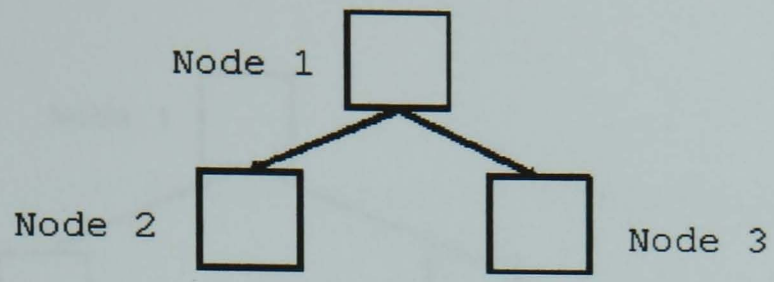


Figure 5.33b: Nodal map of FDT

Loading factor 1.03x

Table 5.25: Summary of results for loading factor of 1.03x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.03x at all load buses	Attribute range in leaf node (Bus No.)
2	1 – 169
4	170 – 191
8	192 – 9005
9	9006 – 9033
7	9034 – 9533

Based upon table 5.25, the location of the leaf nodes are as shown:

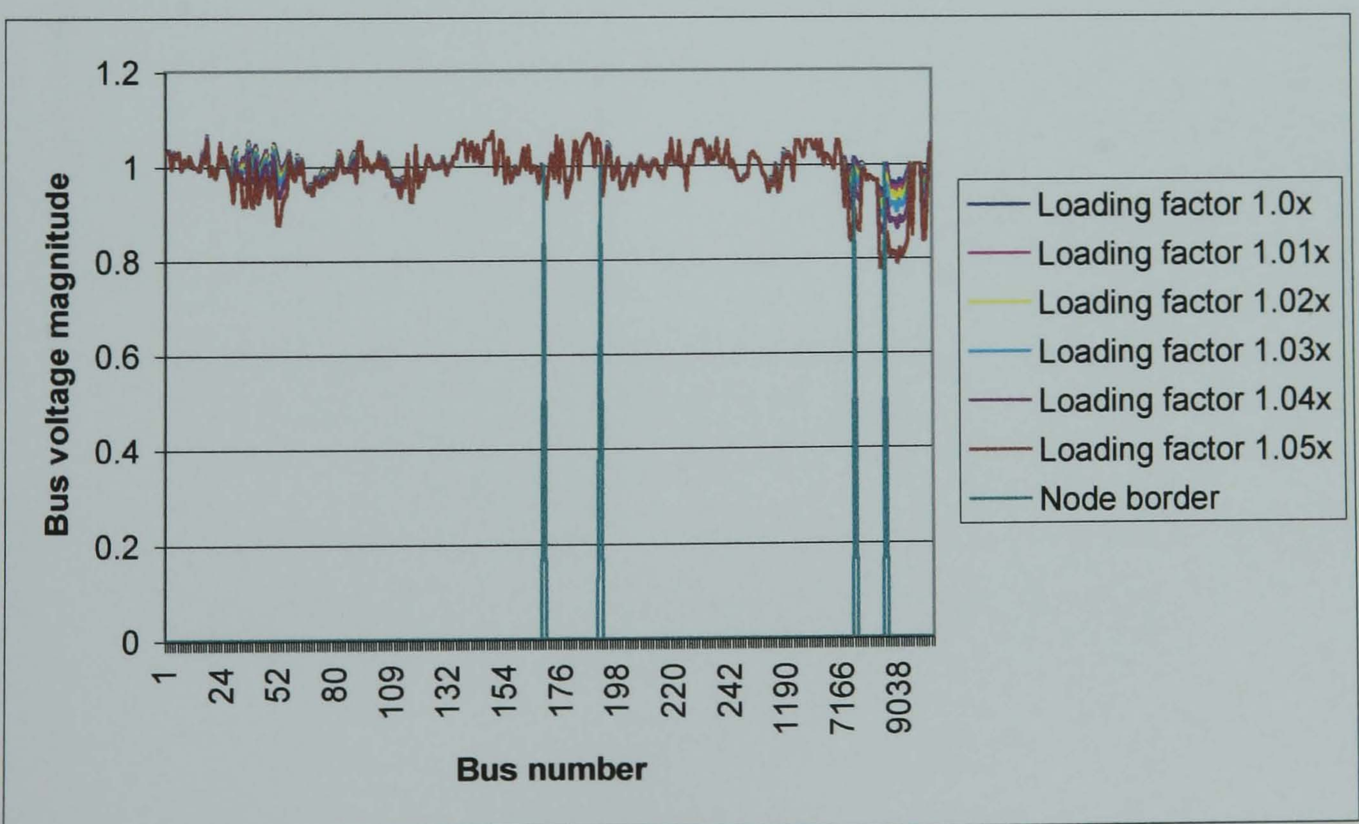


Figure 5.34a: Leaf node location for all load bus loading

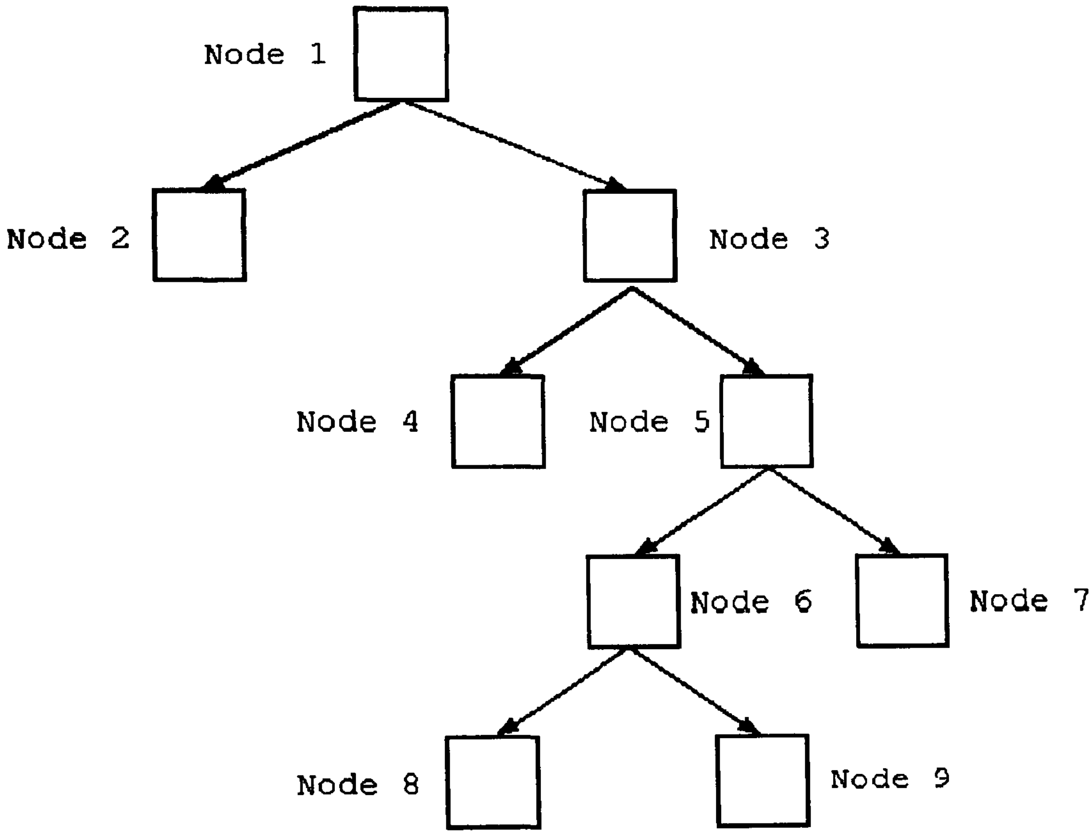


Figure 5.34b: Nodal map of FDT

Loading factor 1.04x

Table 5.26: Summary of results for loading factor of 1.04x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.04x at all load buses	Attribute range in leaf node (Bus No.)
2	1 – 169
4	170 – 191
8	192 – 9005
9	9006 – 9033
7	9034 – 9533

Based upon table 5.26, the location of the leaf nodes are as shown:

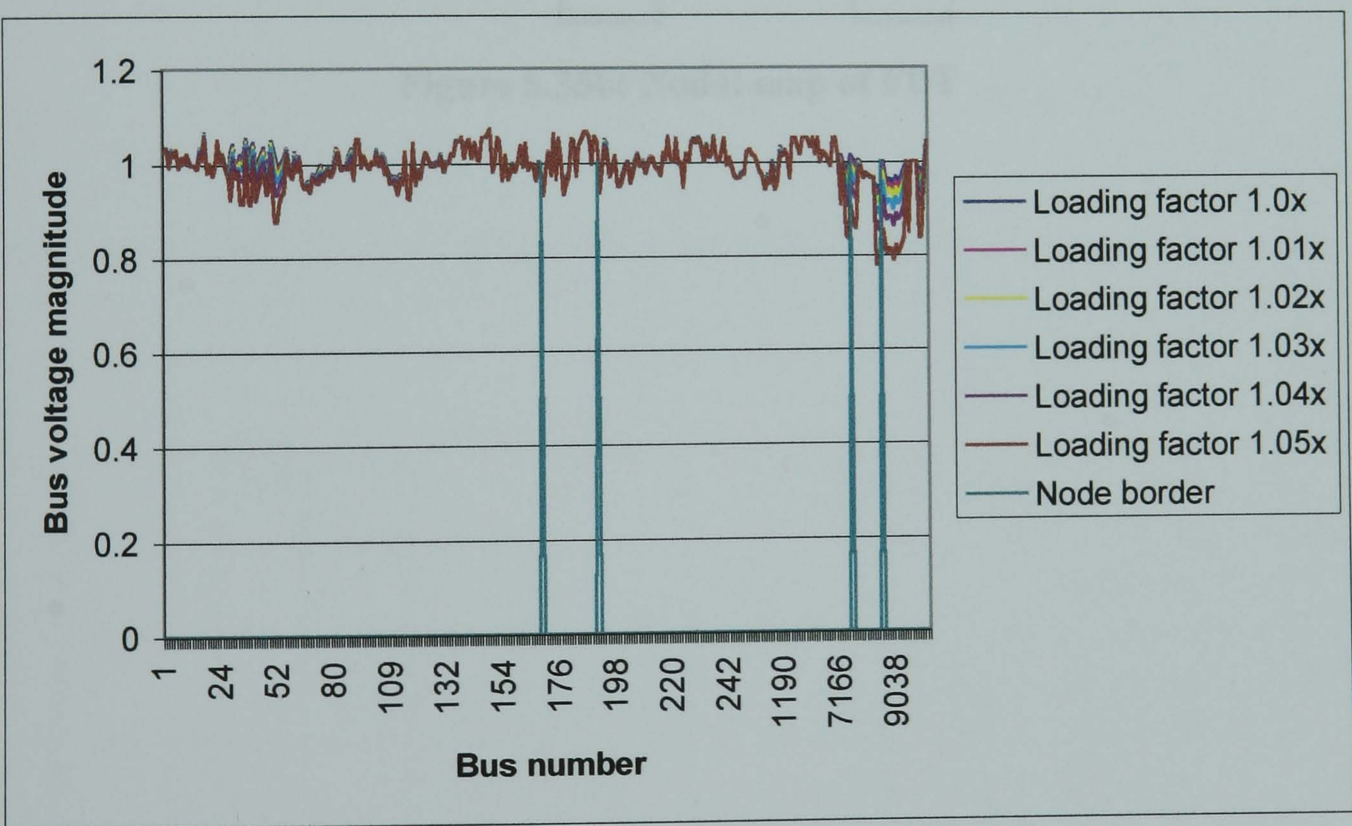


Figure 5.35a: Leaf node location for all load bus loading

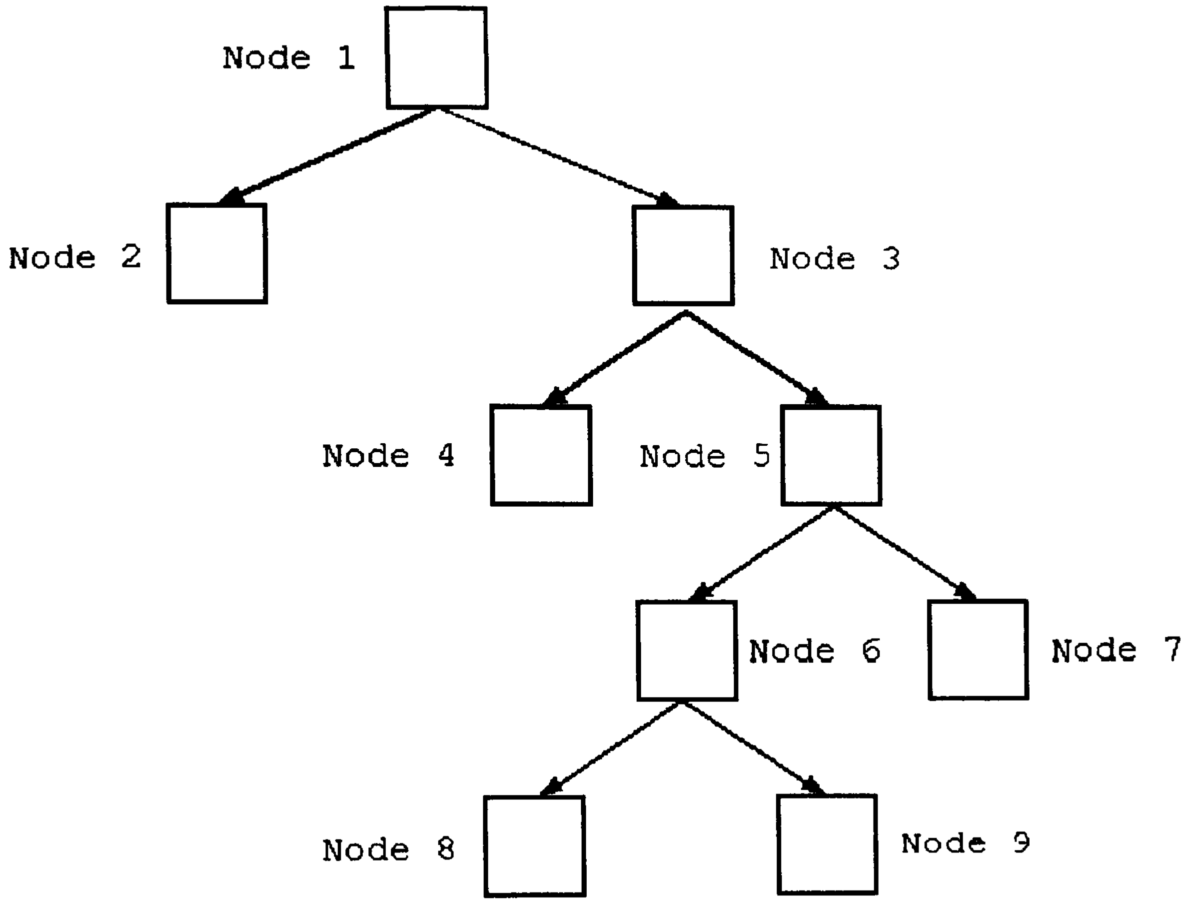


Figure 5.35b: Nodal map of FDT

Loading factor 1.05x

Table 5.27: Summary of results for loading factor of 1.05x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.05x at all load buses	Attribute range in leaf node (Bus No.)
16	1 – 20
17	21 – 62
13	63
9	64 – 117
5	118 – 169
14	170 – 191
18	192 – 224
19	225 – 9005
11	9006 – 9033
7	9034 – 9533

Based upon table 5.27, the location of the leaf nodes are as shown:

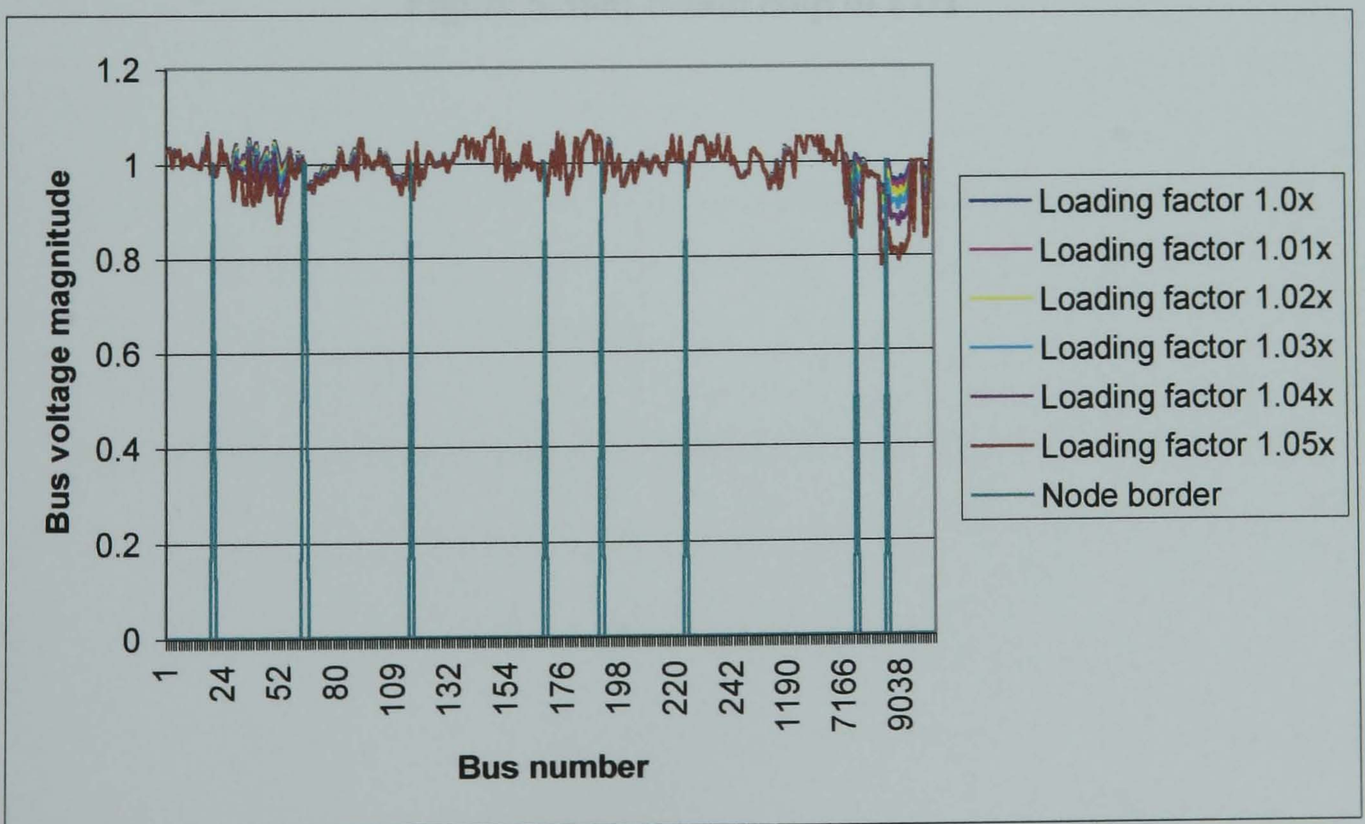


Figure 5.36a: Leaf node location for all load bus loading

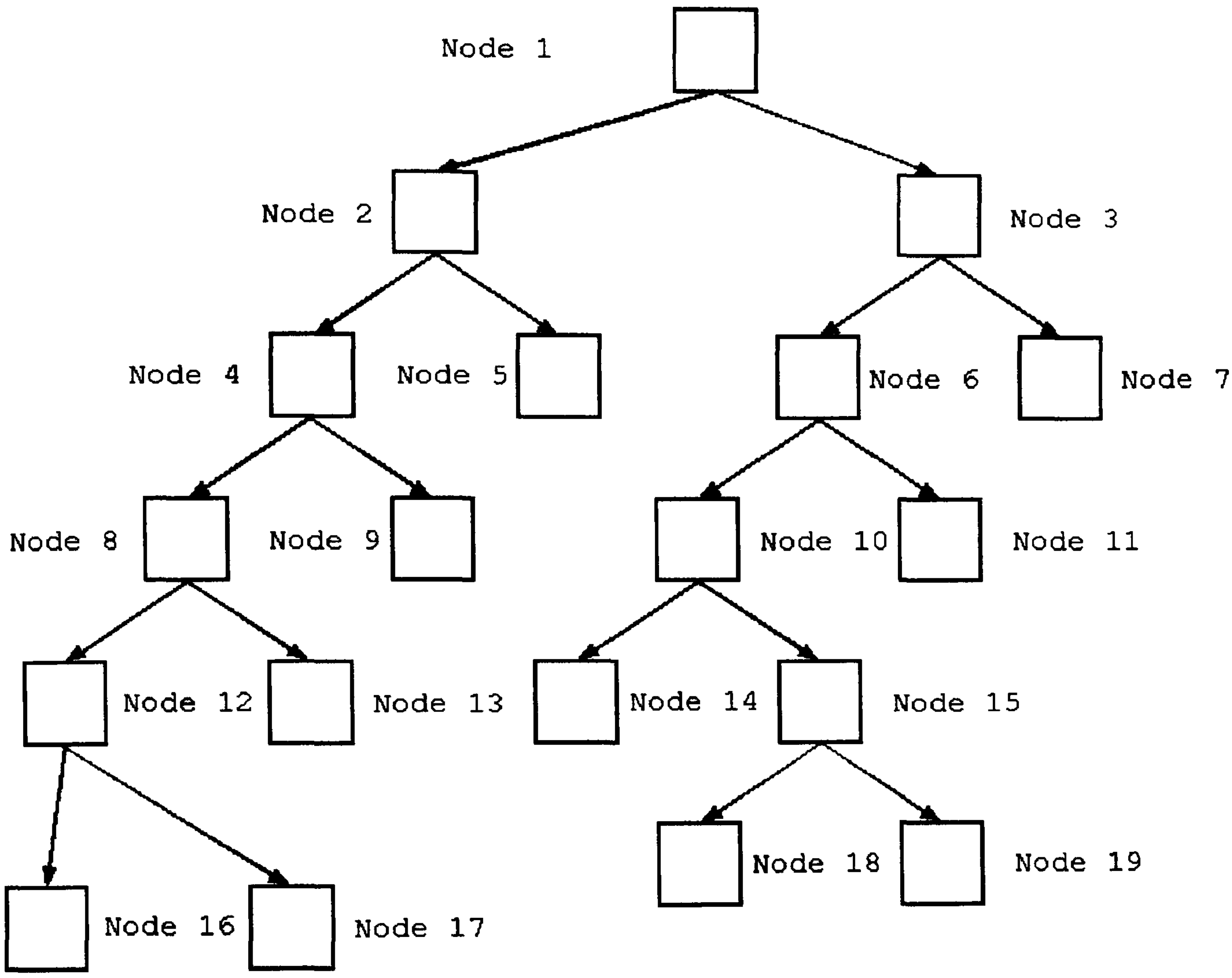


Figure 5.36b: Nodal map of FDT

5.16.2 Heavily loaded bus loading strategy

Loading factor 1.1x

Table 5.28: Summary of results for loading factor of 1.1x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.1x at heavily loaded buses	Attribute range in leaf node (Bus No.)
2	1 – 1200
3	1201 – 9533

Based upon table 5.28, the location of the leaf nodes are as shown:

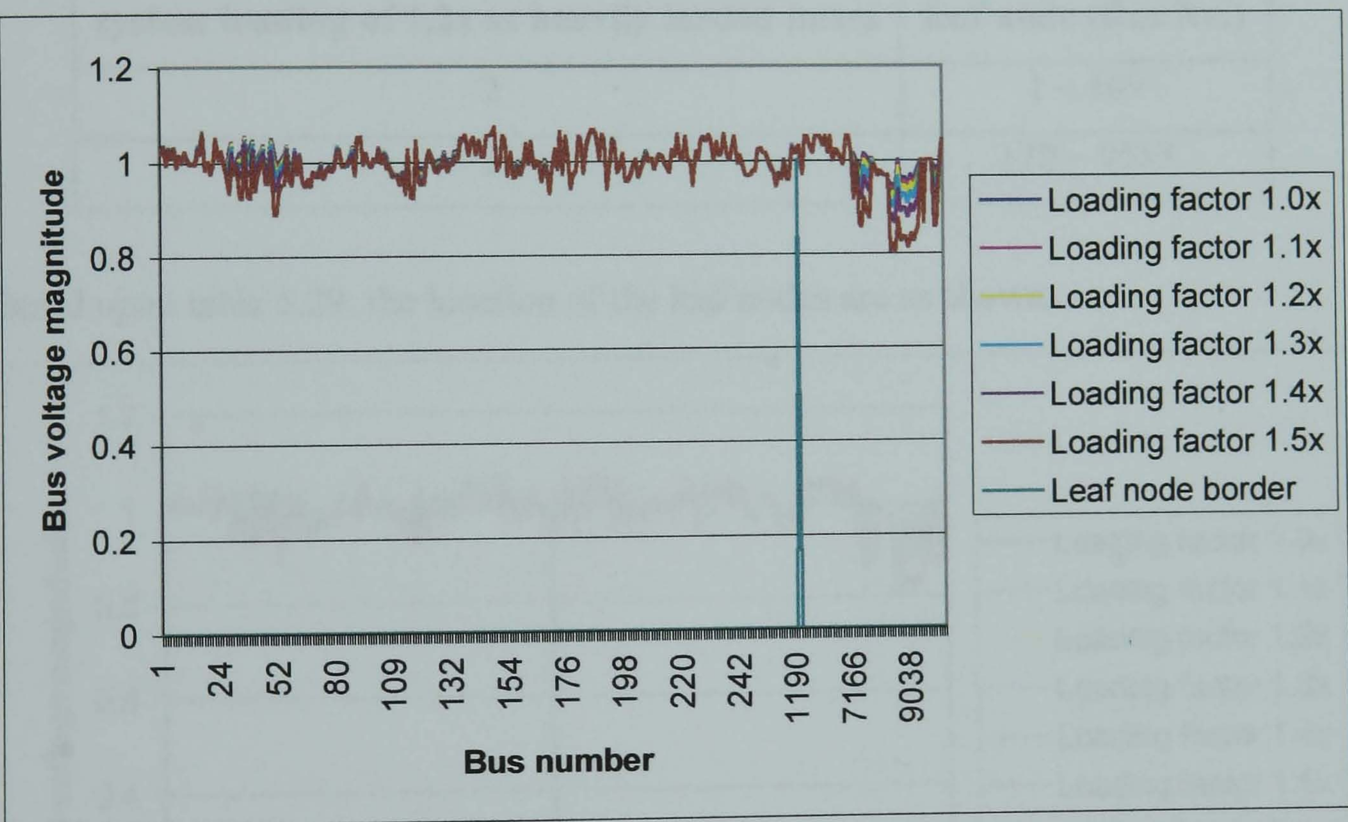


Figure 5.37a: Leaf node location for heavily loaded bus loading strategy

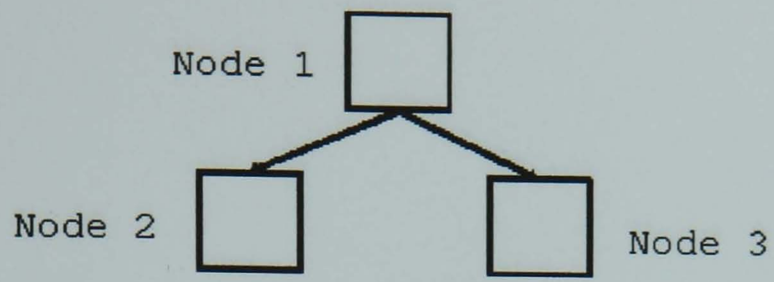


Figure 5.37b: Nodal map of FDT

Loading factor 1.2x

Table 5.29: Summary of results for loading factor of 1.2x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.2x at heavily loaded buses	Attribute range in leaf node (Bus No.)
2	1 – 169
3	170 – 9533

Based upon table 5.29, the location of the leaf nodes are as shown:

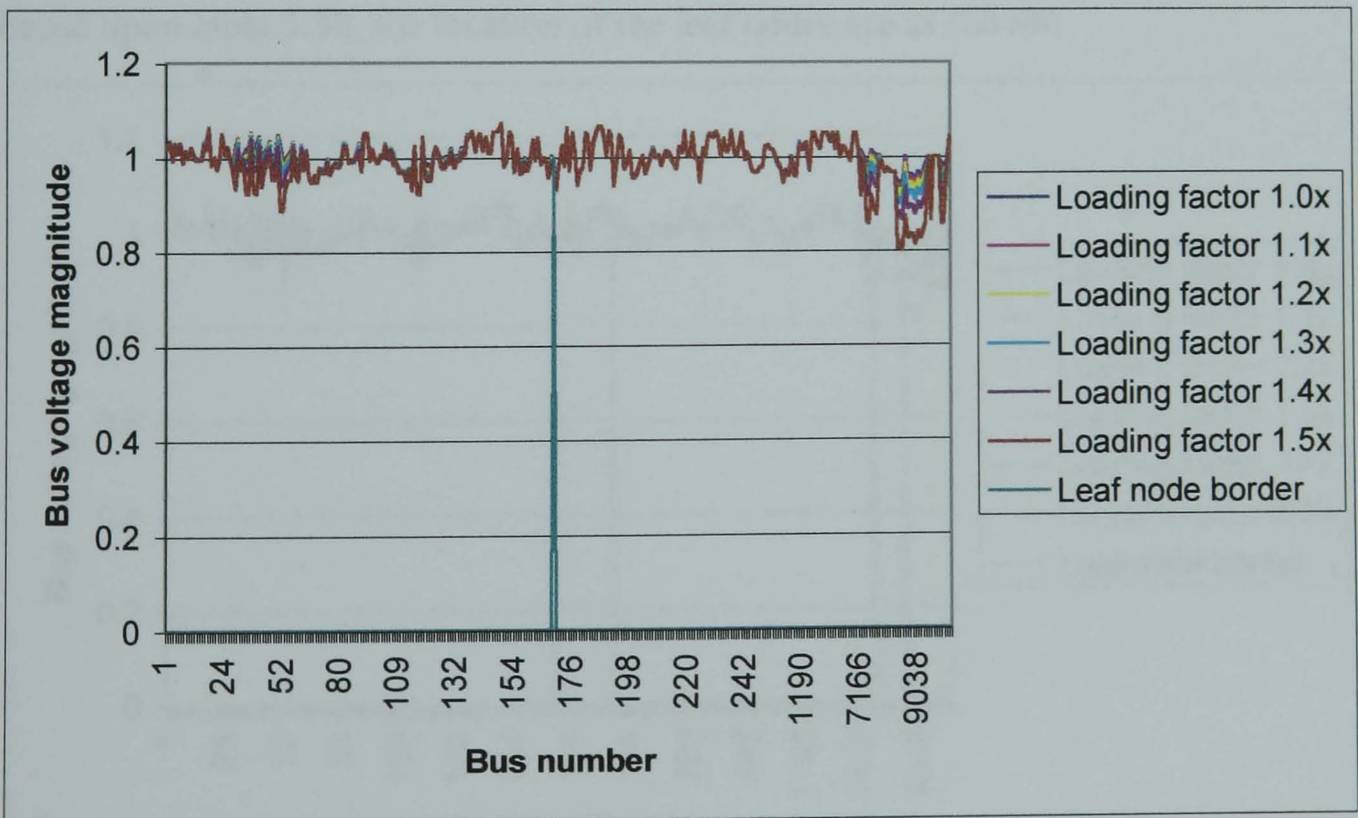


Figure 5.38a: Leaf node location for heavily loaded bus loading strategy

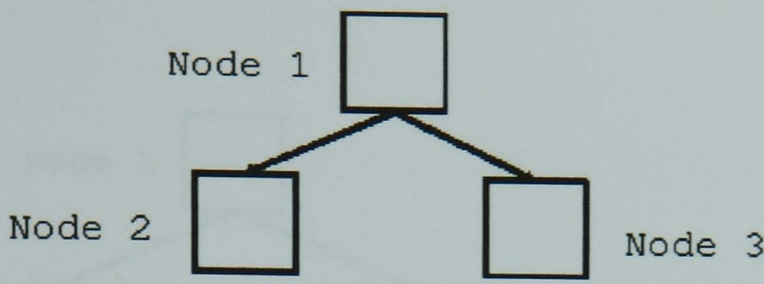


Figure 5.38b: Nodal map of FDT

Loading factor 1.3x

Table 5.30: Summary of results for loading factor of 1.3x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.3x at heavily loaded buses	Attribute range in leaf node (Bus No.)
2	1 – 169
4	170 – 191
8	192 – 9005
9	9006 – 9033
7	9034 – 9533

Based upon table 5.30, the location of the leaf nodes are as shown:

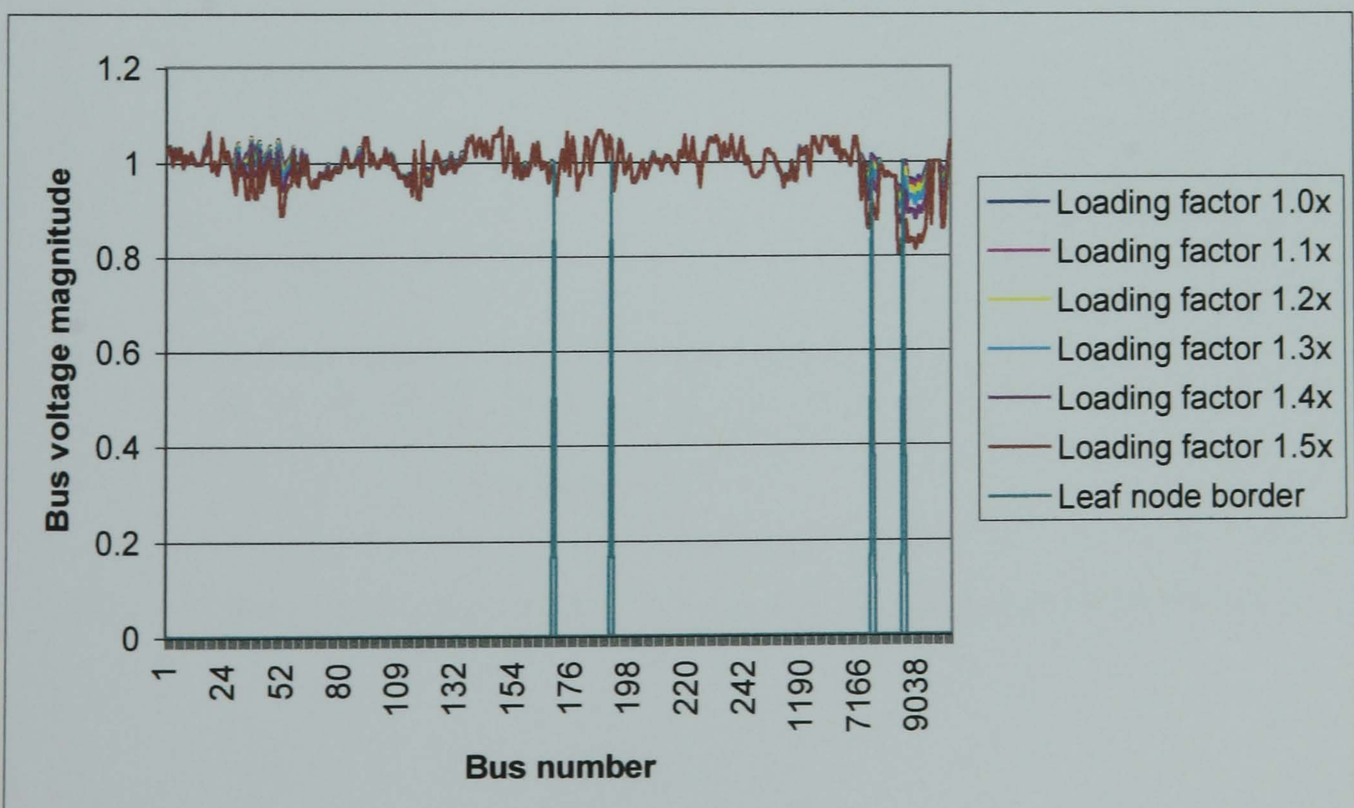


Figure 5.39a: Leaf node location for heavily loaded bus loading strategy

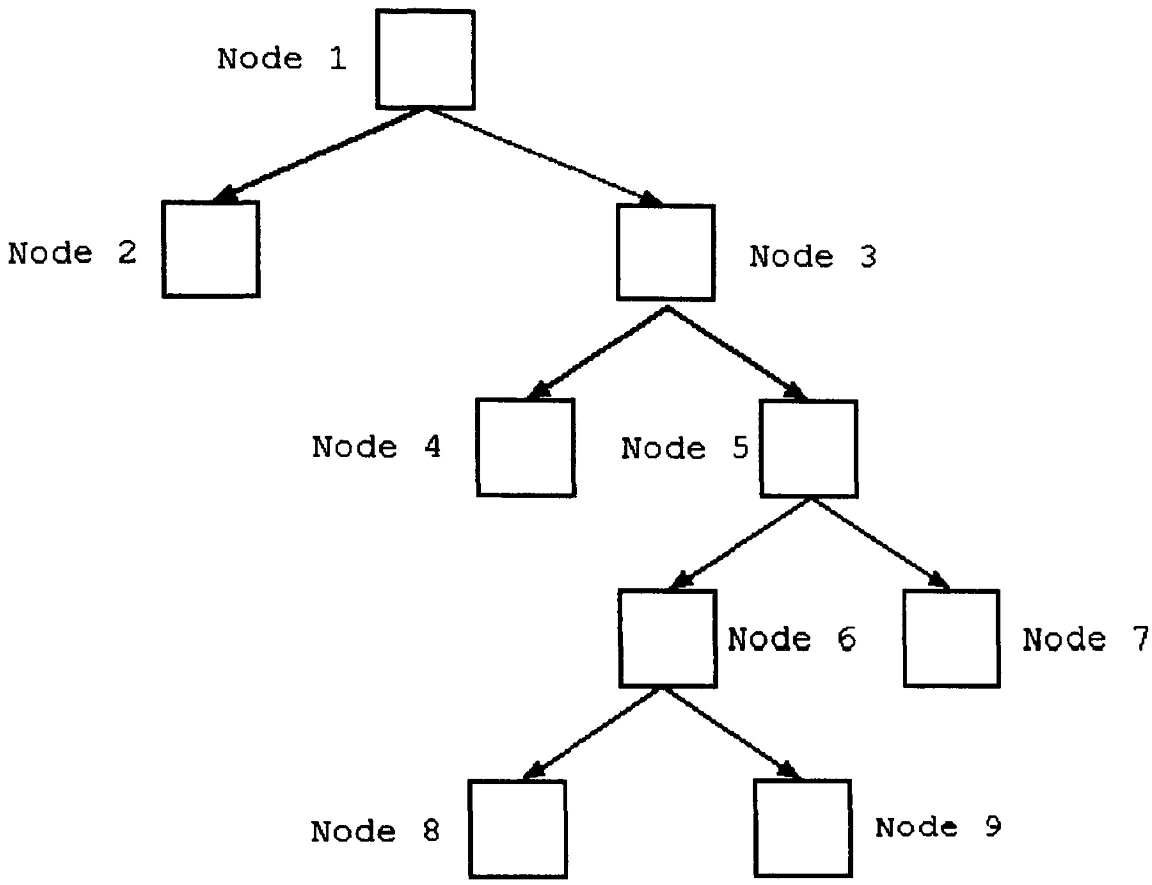


Figure 5.39b: Nodal map of FDT

Loading factor 1.4x

Table 5.31: Summary of results for loading factor of 1.4x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.4x at heavily loaded buses	Attribute range in leaf node (Bus No.)
2	1 – 169
4	170 – 191
8	192 – 9005
9	9006 – 9033
7	9034 - 9533

Based upon table 5.31, the location of the leaf nodes are as shown:

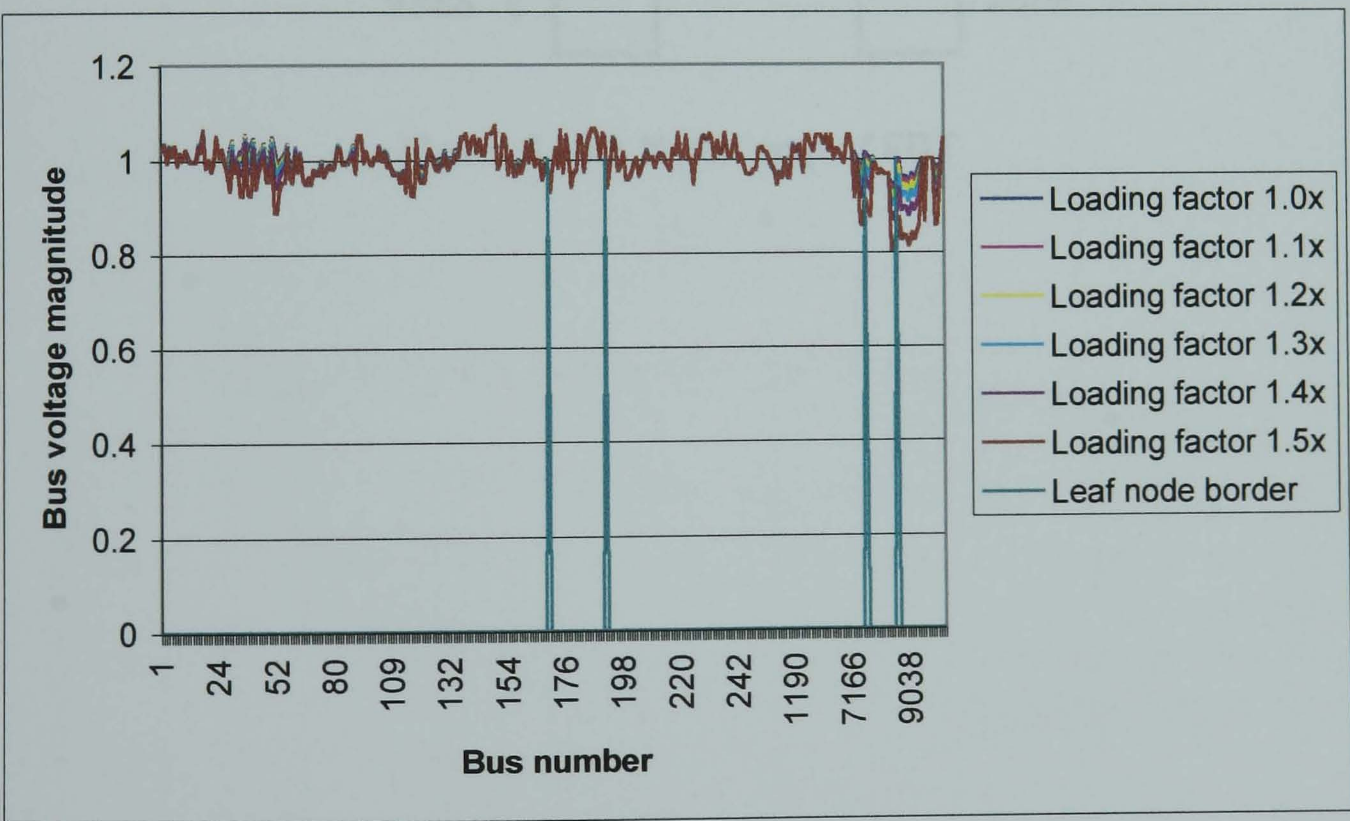


Figure 5.40a: Leaf node location for heavily loaded bus loading strategy

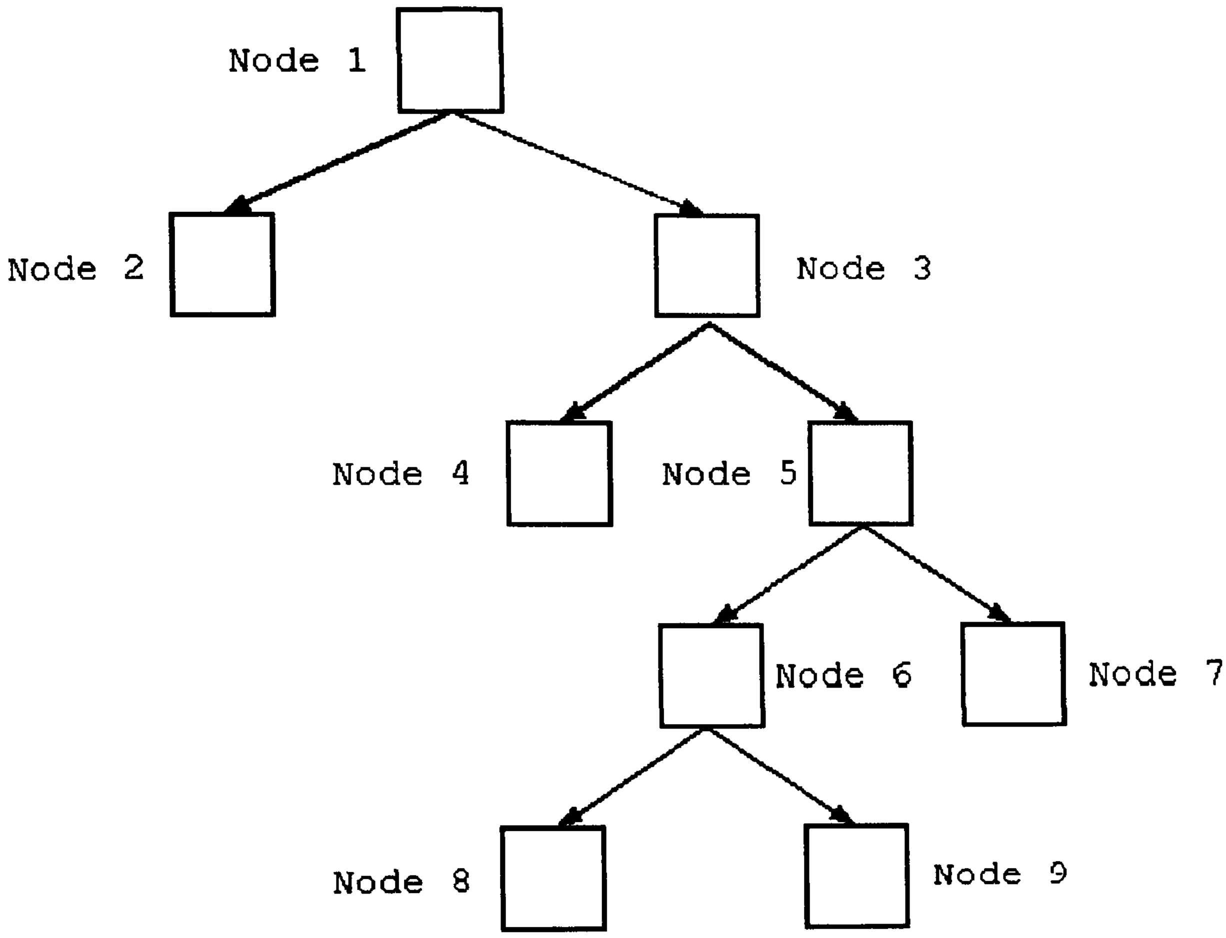


Figure 5.40b: Nodal map of FDT

Loading factor 1.5x

Table 5.32: Summary of results for loading factor of 1.5x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.5x at heavily loaded buses	Attribute range in leaf node (Bus No.)
12	1 – 26
13	27 – 62
9	63
5	64 – 169
14	170 – 191
16	192 – 324
17	526 – 9005
11	9006 – 9033
7	9034 – 9533

Based upon table 5.32, the location of the leaf nodes are as shown:

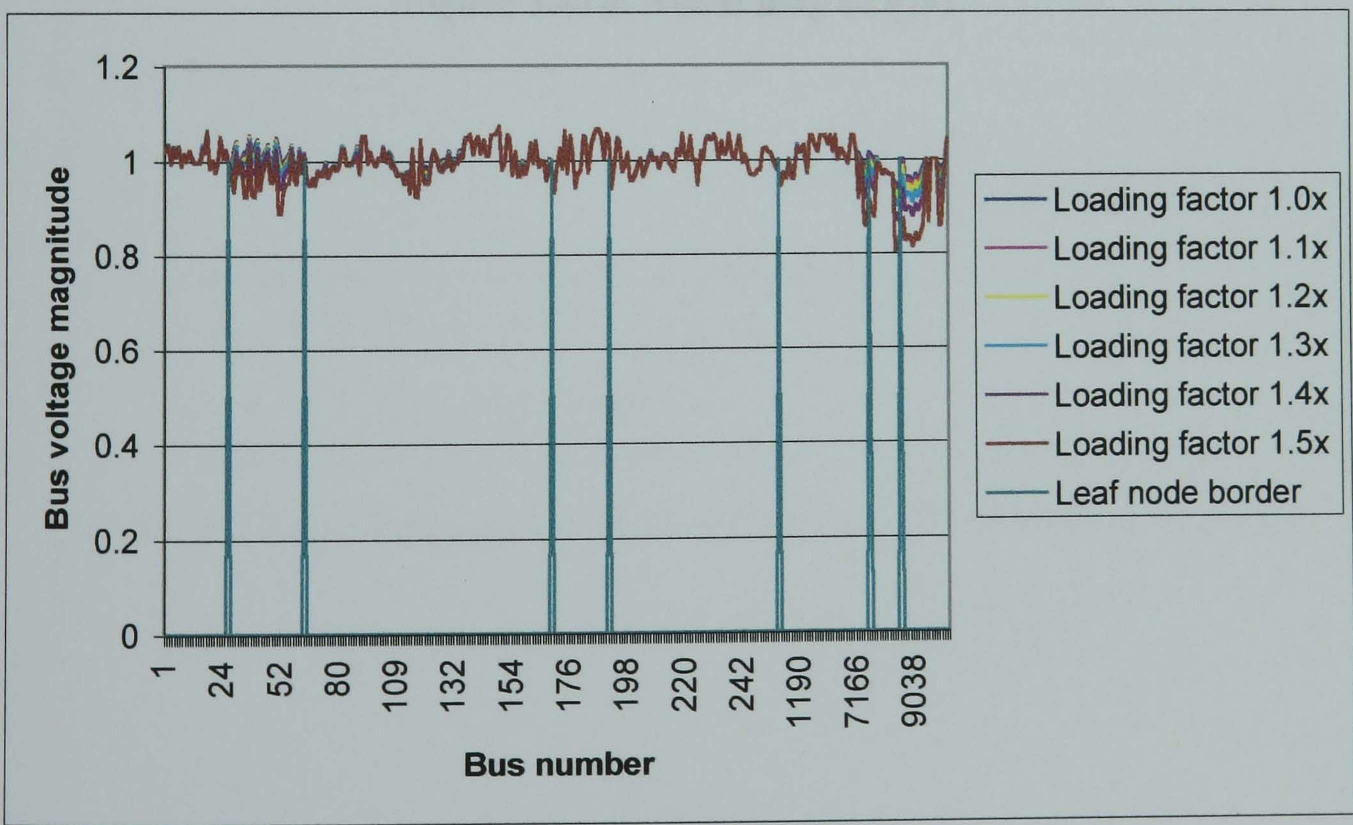


Figure 5.41a: Leaf node location for heavily loaded bus loading strategy

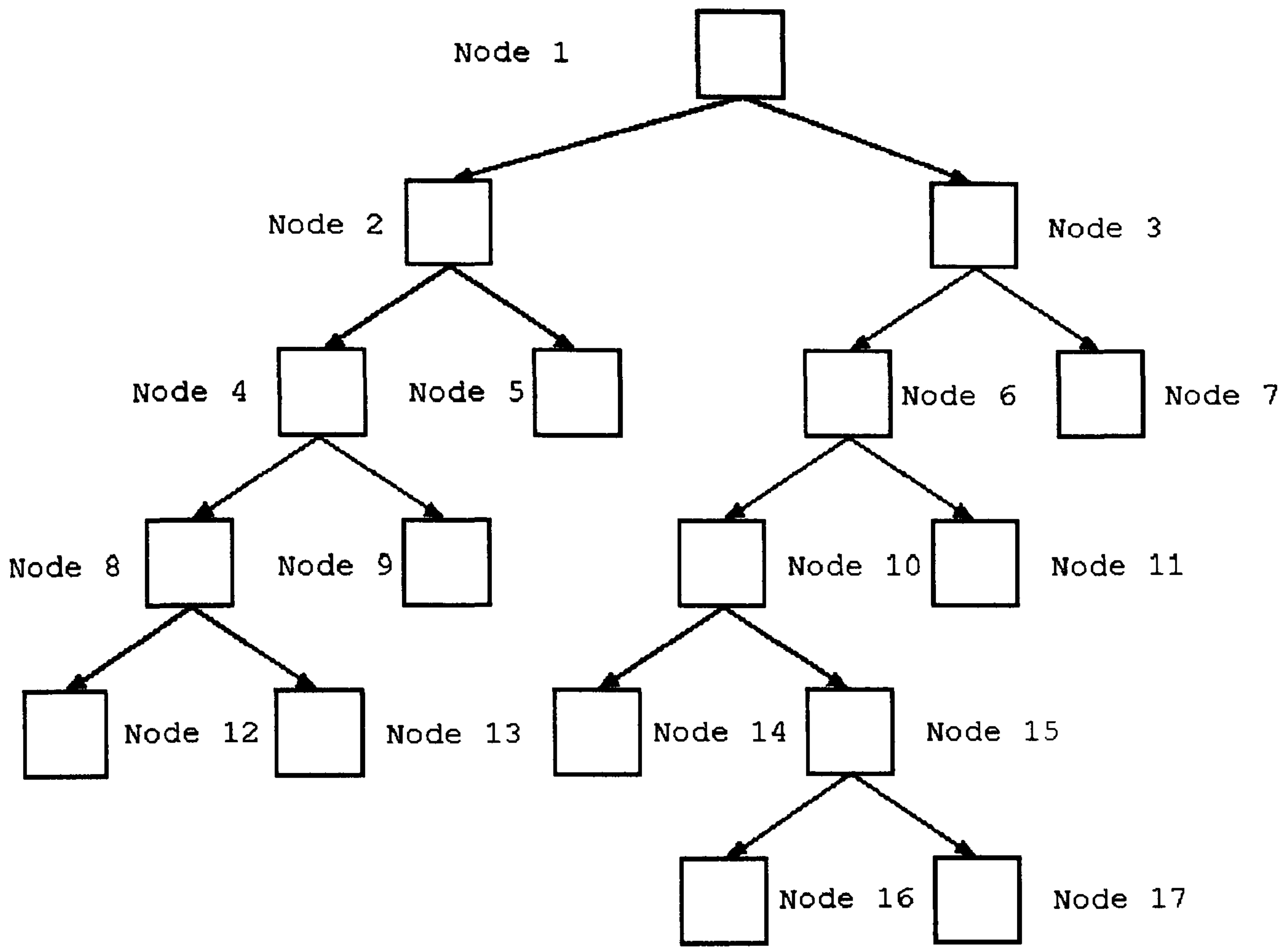


Figure 5.41b: Nodal map of FDT

5.16.3 Lightly loaded bus loading strategy

Loading factor 1.4x

Table 5.33: Summary of results for loading factor of 1.4x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.4x at lightly loaded buses	Attribute range in leaf node (Bus No.)
2	1 – 169
3	170 – 9533

Based upon table 5.33, the location of the leaf nodes are as shown:

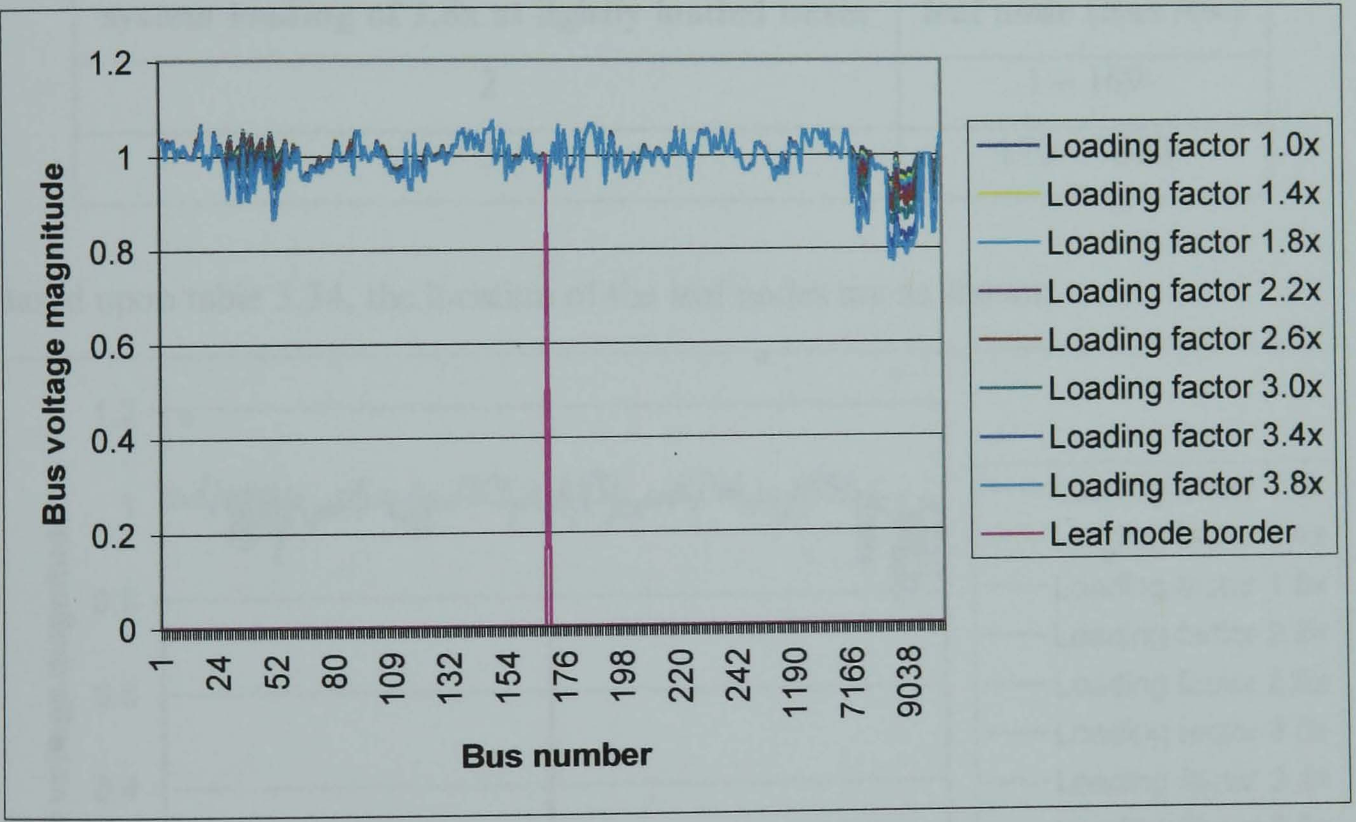


Figure 5.42a: Leaf node location for lightly loaded bus loading strategy

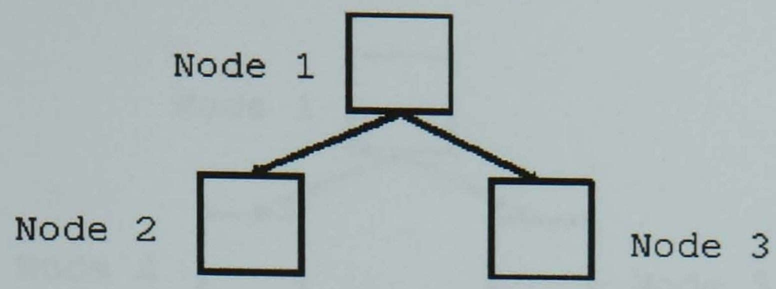


Figure 5.42b: Nodal map of FDT

Loading factor 1.8x

Table 5.34: Summary of results for loading factor of 1.8x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.8x at lightly loaded buses	Attribute range in leaf node (Bus No.)
2	1 – 169
3	170 – 9533

Based upon table 5.34, the location of the leaf nodes are as shown:

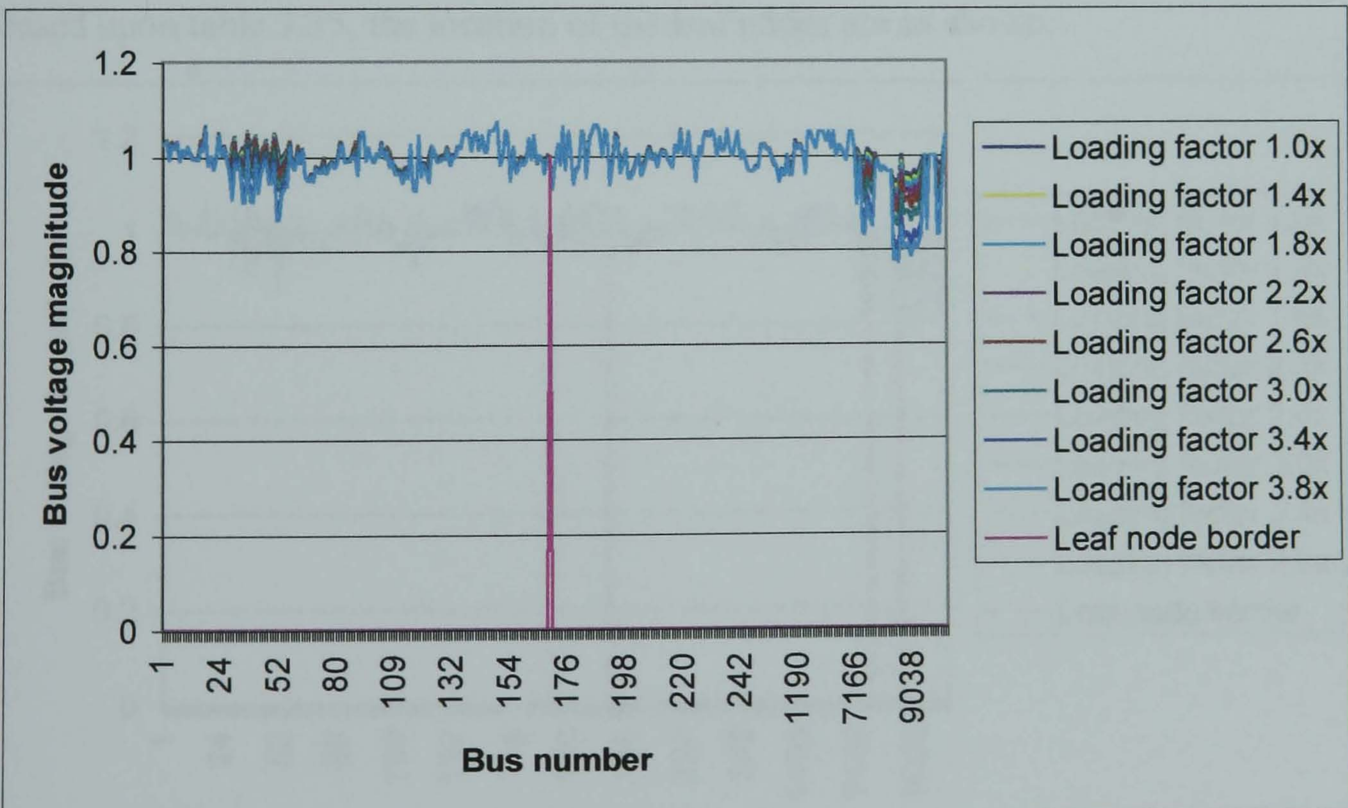


Figure 5.43a: Leaf node location for lightly loaded bus loading strategy

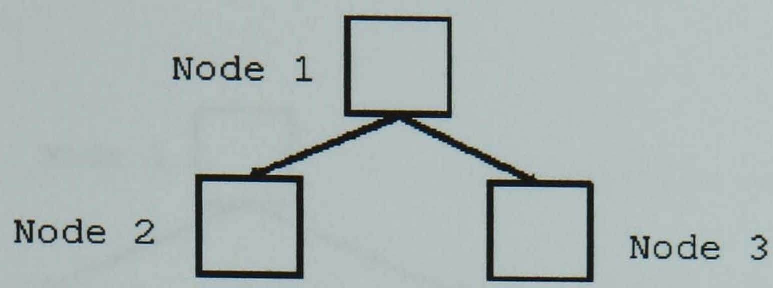


Figure 5.43b: Nodal map of FDT

<u>Loading factor 2.2x</u>

Table 5.35: Summary of results for loading factor of 2.2x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 2.2x at lightly loaded buses	Attribute range in leaf node (Bus No.)
2	1 – 169
4	170 – 191
8	192 – 9005
9	9006 - 9033
7	9034 - 9533

Based upon table 5.35, the location of the leaf nodes are as shown:

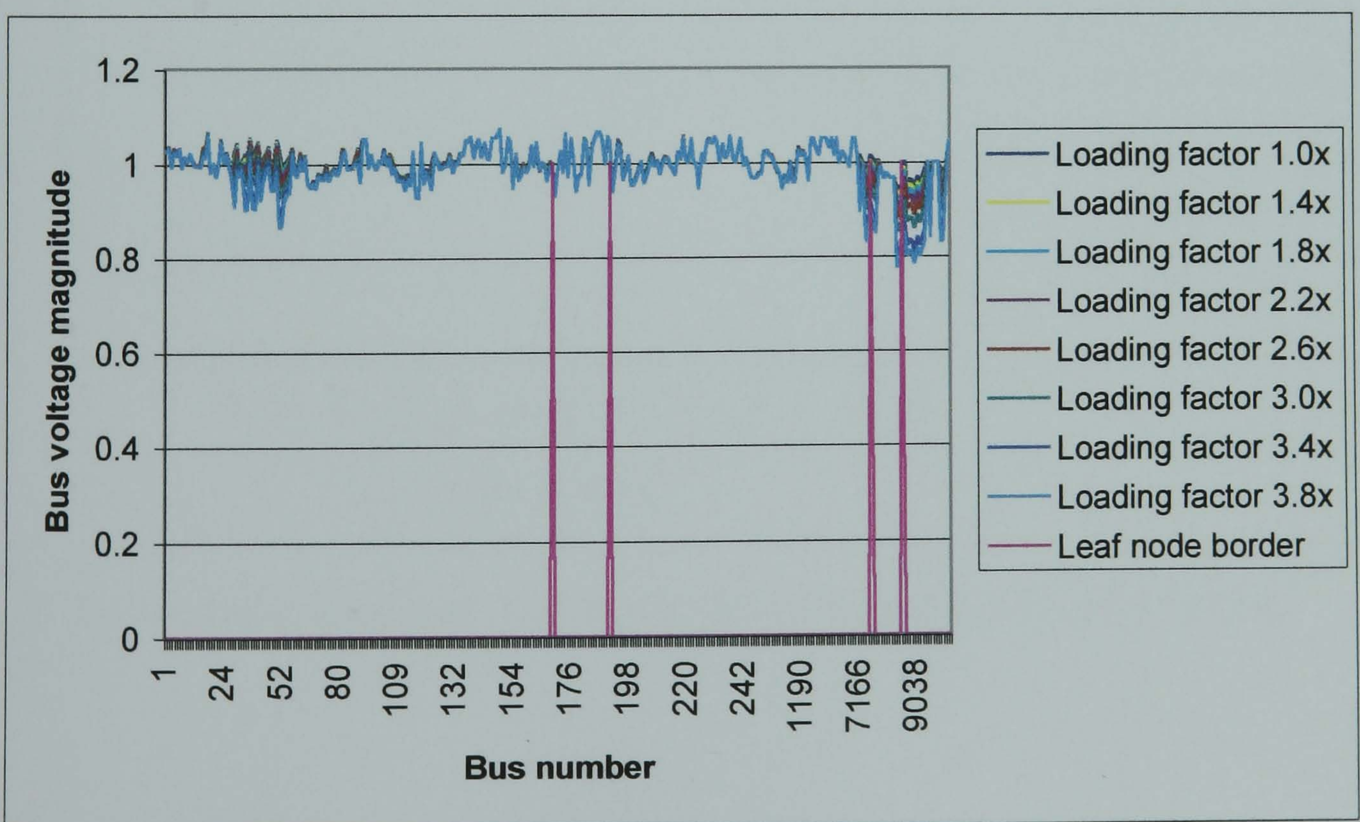


Figure 5.44a: Leaf node location for lightly loaded bus loading strategy

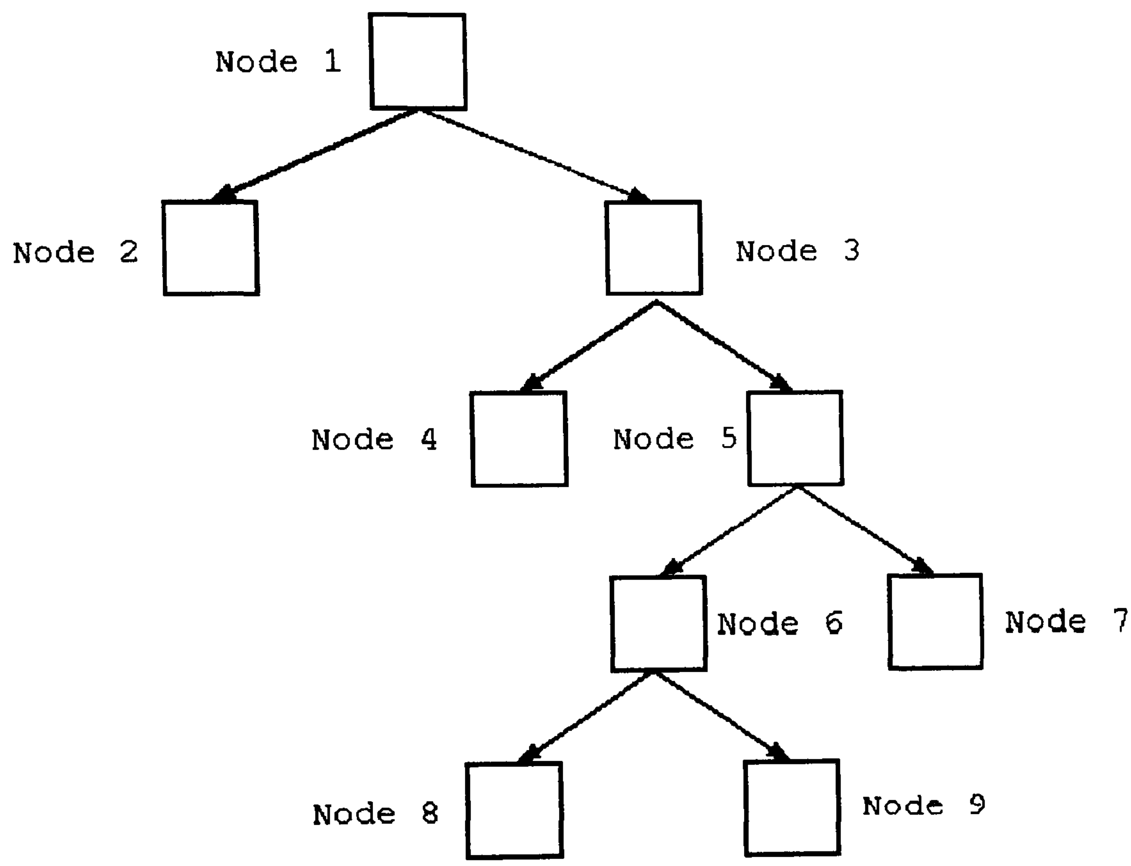


Figure 5.44b: Nodal map of FDT

Loading factor 2.6x

Table 5.36: Summary of results for loading factor of 2.6x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 2.6x at lightly loaded buses	Attribute range in leaf node (Bus No.)
2	1 – 169
4	170 – 191
8	192 – 9005
9	9006 - 9033
7	9034 - 9533

Based upon table 5.36, the location of the leaf nodes are as shown:

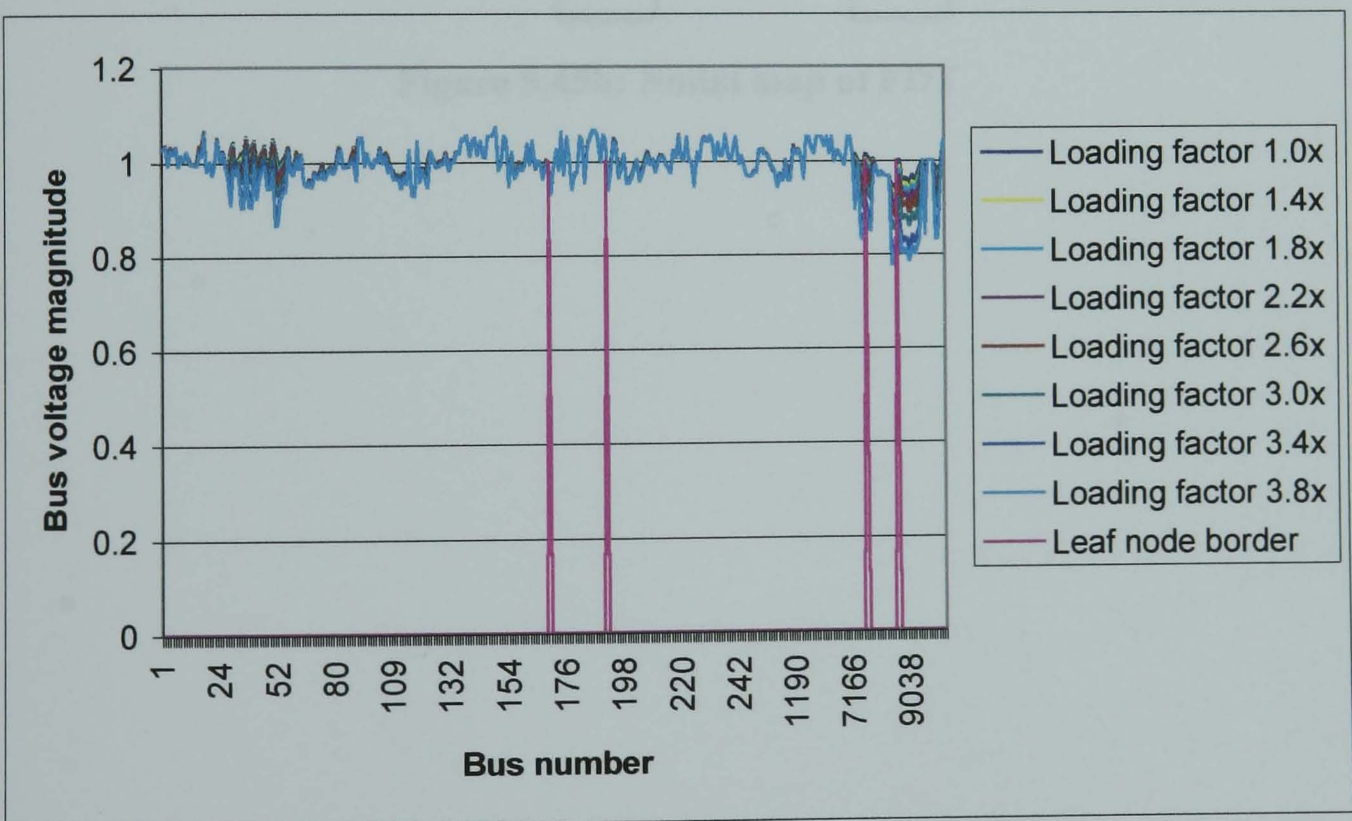


Figure 5.45a: Leaf node location for lightly loaded bus loading strategy

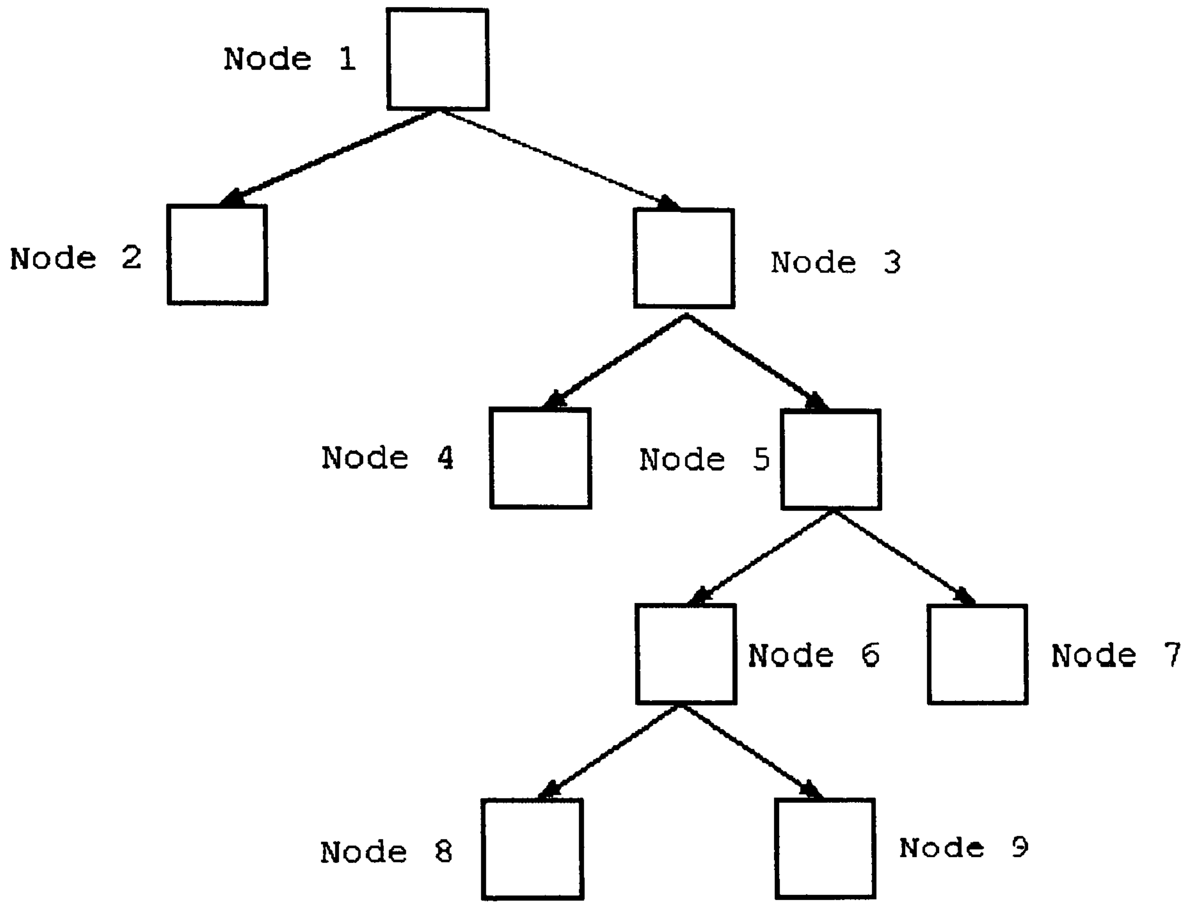


Figure 5.45b: Nodal map of FDT

Loading factor 3.0x

Table 5.37: Summary of results for loading factor of 3.0x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 3.0x at lightly loaded buses	Attribute range in leaf node (Bus No.)
2	1 – 169
4	170 – 191
8	192 – 9005
9	9006 - 9033
7	9034 - 9533

Based upon table 5.37, the location of the leaf nodes are as shown:

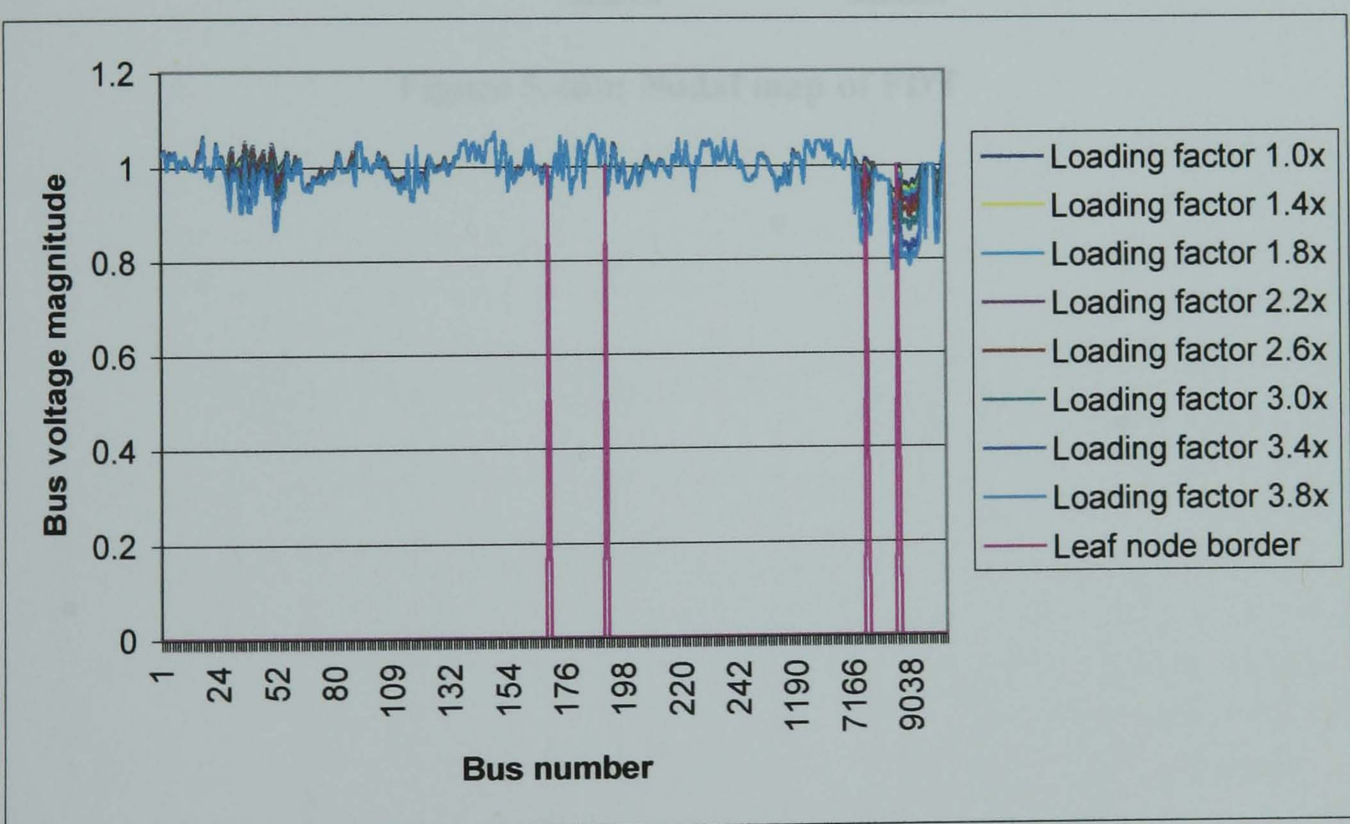


Figure 5.46a: Leaf node location for lightly loaded bus loading strategy

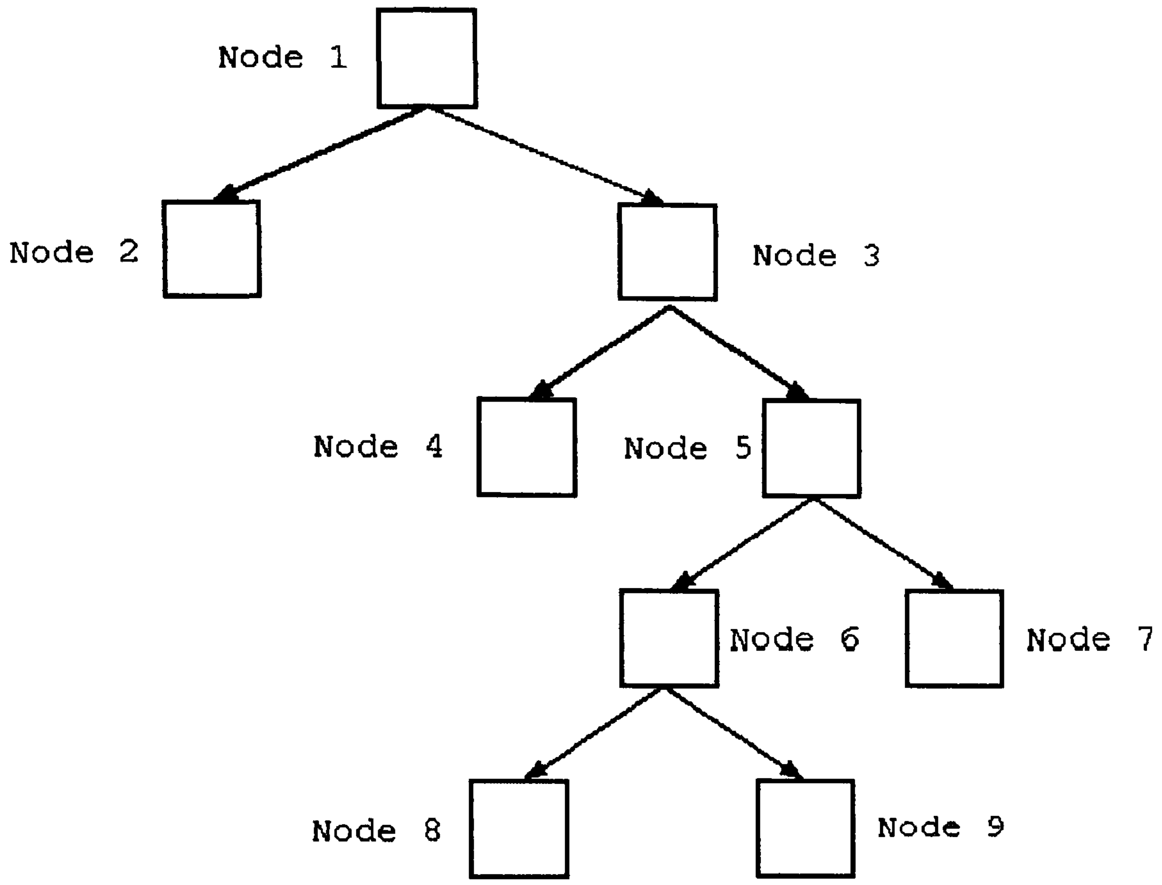


Figure 5.46b: Nodal map of FDT

Loading factor 3.4x

Table 5.38: Summary of results for loading factor of 3.4x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 3.4x at lightly loaded buses	Attribute range in leaf node (Bus No.)
20	1 – 26
21	27 – 62
17	63
13	64 – 116
9	117
5	118 – 169
14	170 – 191
18	192 – 324
19	526 – 9005
11	9006 – 9033
7	9034 – 9533

Based upon table 5.38, the location of the leaf nodes are as shown:

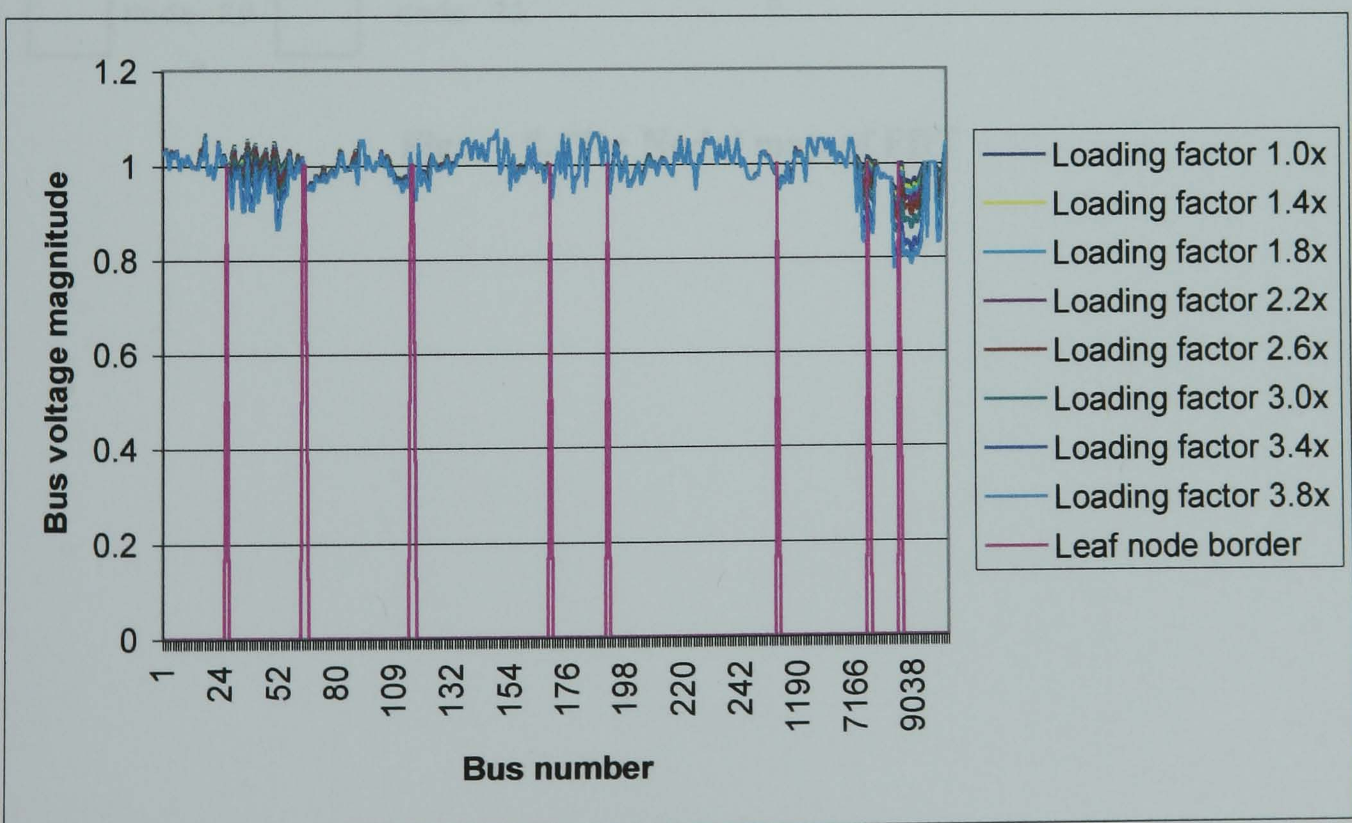


Figure 5.47a: Leaf node location for lightly loaded bus loading strategy

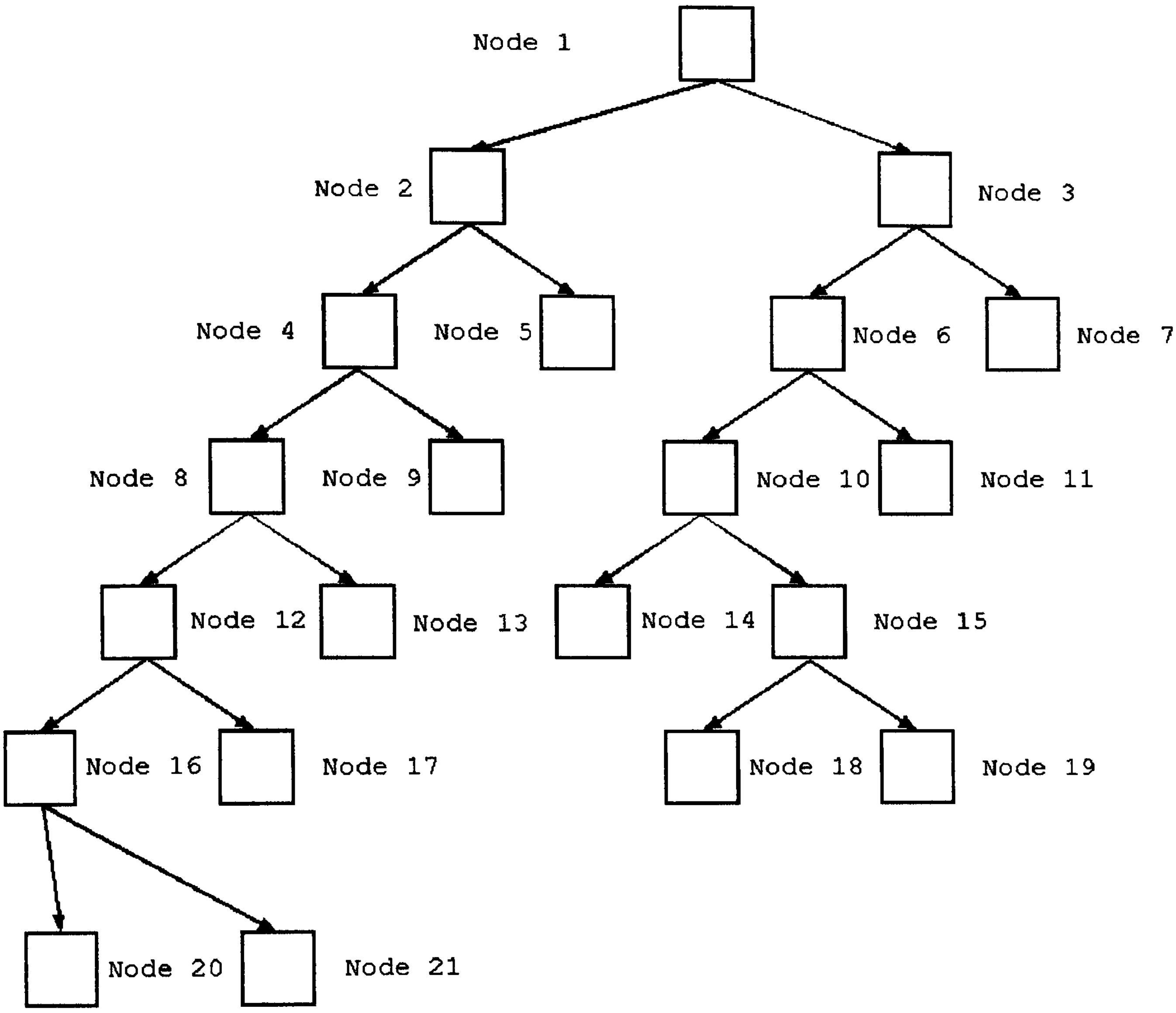


Figure 5.47b: Nodal map of FDT

Loading factor 3.8x

Table 5.39: Summary of results for loading factor of 3.8x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 3.8x at lightly loaded buses	Attribute range in leaf node (Bus No.)
20	1 – 26
21	27 – 62
17	63
13	64 – 116
9	117
5	118 – 169
14	170 – 191
18	192 – 324
19	526 – 9005
11	9006 – 9033
7	9034 – 9533

Based upon table 5.39, the location of the leaf nodes are as shown:

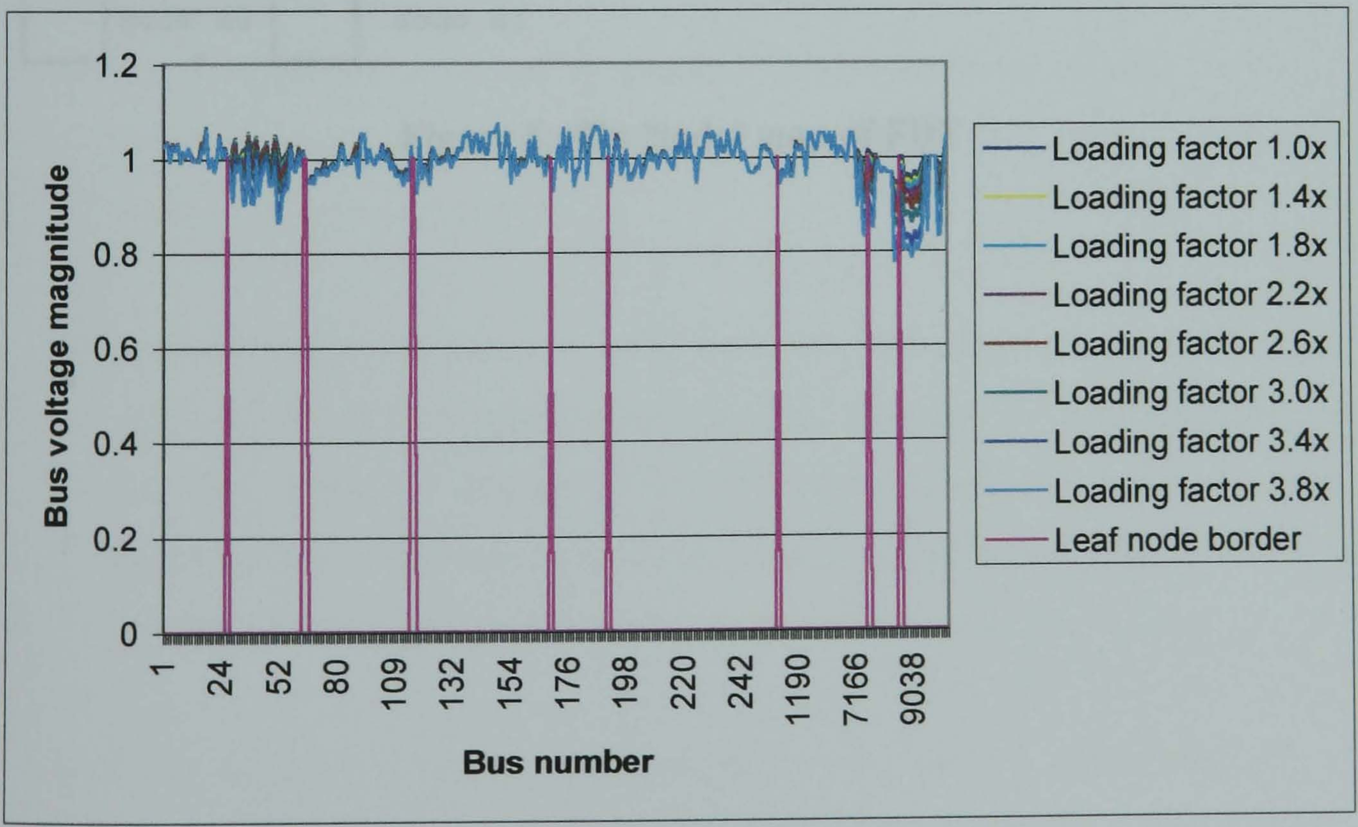


Figure 5.48a: Leaf node location for lightly loaded bus loading strategy

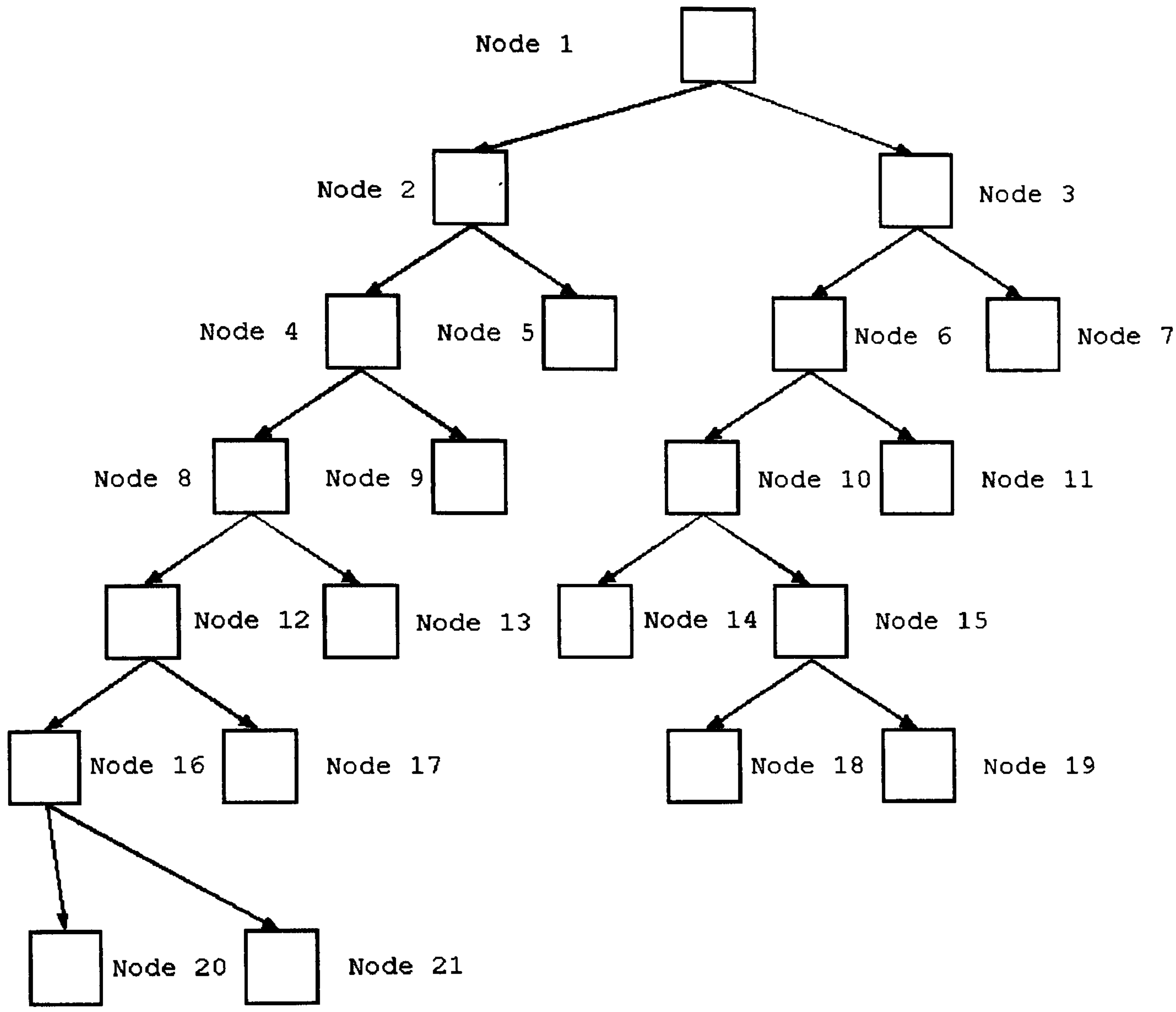


Figure 5.48b: Nodal map of FDT

Discussion

The FDT performance for this test is good with the ability to partition the data into nodes that contain weak and strong buses. Similar to test 3, significant partitioning results is possible for medium to high loading factors. Unfortunately, the FDT does suffer several problems.

(a) FDT sensitivity

Comparing with test 3, the FDT in test 4 is more sensitive at the medium to high loading factor. Sensitivity in this sense means that the leaf node locations do cover the weak and strong bus area with good partitioning capabilities at weak bus areas. Unfortunately, the FDT produces too much redundant leaf nodes i.e. there are possibilities that several leaf nodes can be collapsed and the leaf node label transferred to the nearest test node. A possible solution is to introduce a pruning algorithm. This issue is looked into in chapter 6.

(b) Over partitioning

As a result of increased sensitivity, there are more instances of single occupant nodes (over-partitioning) as compared to test 3 where it only has 1 instance of over-partitioning.

(c) Reduction of leaf nodes variance at higher loading factors

Although the FDT is sensitive as such that it produces more leaf nodes, but the FDT failed to distinguish the difference between 2 different loading factors. This is shown as identical leaf node combination from one loading level to the next. This problem rarely happens in test 3.

Looking into all these points mentioned above, a conclusion is made where the KS formulation based upon the KS method in the FFDT has a better advantage compared to the conceptual KS version. The reason is because of the embedded contrast discriminator in the KS formulation that increases the KS score and hence improving its reliability.

Considering the complexity of the voltage collapse phenomenon, adding more attribute types into the FDT will logically increase its efficiency and accuracy. From chapter 2, it is shown that there are other contributors where each gives a different effect to the system stability in its own way. Considering all these contributors and also other possible indicators, there is a lot of possible combination that may be beneficial to the performance of the FDT. This led to the introduction of a Hybrid FDT that utilises multiple attribute types in its algorithm structure.

5.17 Multiple attribute partitioning – Hybrid FDT

The idea of introducing multiple attribute partitioning has been proposed before with the emergence of the option trees as outlined in [2, 7]. Using these ideas as a concept, the author proposed the next improvement to the FDT in line with the addition of other attributes that may enhance the performance of the FDT. The amount of possible combination may be limitless and will be discussed further in chapter 6 but for this research, the author has decided to include bus voltage angle as the next choice of attribute.

5.17.1 Introducing bus voltage angle as a second attribute

Similar to bus voltage magnitude, bus voltage angle has a distinct behaviour as the system is slowly loaded to its maximum capabilities. Looking back at the bus voltage behaviour outlined earlier, similar experiments are made to observe bus voltage angles as it is applied at different loading strategies. A summary of the test results is presented in Appendix M where the bus voltage angle (in radians) increases exponentially as the system load is increased. Based upon findings and conclusion presented at Appendix M, the author has decided to include bus voltage angle as a supporting attribute to the FDT algorithm.

The bus voltage angle difference will be added into the FDT at the KS scoring stage as an enhancement factor. The reason for this is presented in Appendix N where the introduction of the bus voltage angle values gives a better representation of the KS score profile highlighting weak and strong bus areas. Based upon the deduction made in Appendix N, the FDT is modified at the KS stage to include the bus voltage angle difference as the new attribute. As of the choice of FDT used, based upon the results presented in test 3 and test 4, the best choice of FDT will be the one based upon the FFDT KS concept, hence this configuration is maintained for the next test.

5.18 Test 5: Application to IEEE 300 bus system using FDT ver 5.0

Table 5.40: Test 5 Program summary:

Version of FDT	FDT ver 5.0
KS technique	Hybrid KS multiplying in the bus voltage angle difference into the KS formulation.
Partitioning technique	Comparing the location of the maximum and minimum KS score. • If theses scores are located at both ends of the test nodes, partitioning will stop. • If the maximum KS score is on the extreme ends of the test node, and the location of the minimum KS score is not located at the extreme ends of the test node, the partitioning limit will be based upon the location of the minimum KS score. • If the maximum KS score is not at extreme ends, partitioning will be based upon the maximum KS location as a partitioning limit. The location of the KS scores are considered to be on the extreme ends if is located within the range of 10 attributes from each end.

Stop split rule	The rules for the stop split stage are based upon the purity of the node and the minimum population (minimum population value is 50 attributes) of the test node. The summary of the rules are: • If the test node consist of all strong buses (minimum attribute value is more than 0.9pu) stop splitting the node and label is as a leaf node. • If the test node consist of a combination of strong buses and weak buses (minimum attribute value is less than 0.9pu). Check the total population of the test node. If it exceeds the minimum population limit continue partitioning. If not, stop partitioning and label the test node as a leaf node.
Pruning technique	No pruning technique is used
Test system used	IEEE 300 bus system using all load bus increment strategy as explained in the data preparation section
Computer used	Intel PIII 1GHz

Results

The results are presented is in the form of leaf node numbers and its location. The location of each leaf node is presented on a general bus voltage magnitude profile with all the loading factors superimposed on it. The reason for this is to highlight the weak buses area (denoted by varying bus voltage magnitude when the loading factor is increased) and how good the FDT partition the data set into sections of strong and weak buses.

5.18.1 All load bus loading strategy

Loading factor 1.01x

Table 5.41: Summary of results for loading factor of 1.01x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.01x at all load buses	Attribute range in leaf node (Bus No.)
2	1 – 159
3	160 - 9533

Based upon table 5.41, the location of the leaf nodes are as shown:

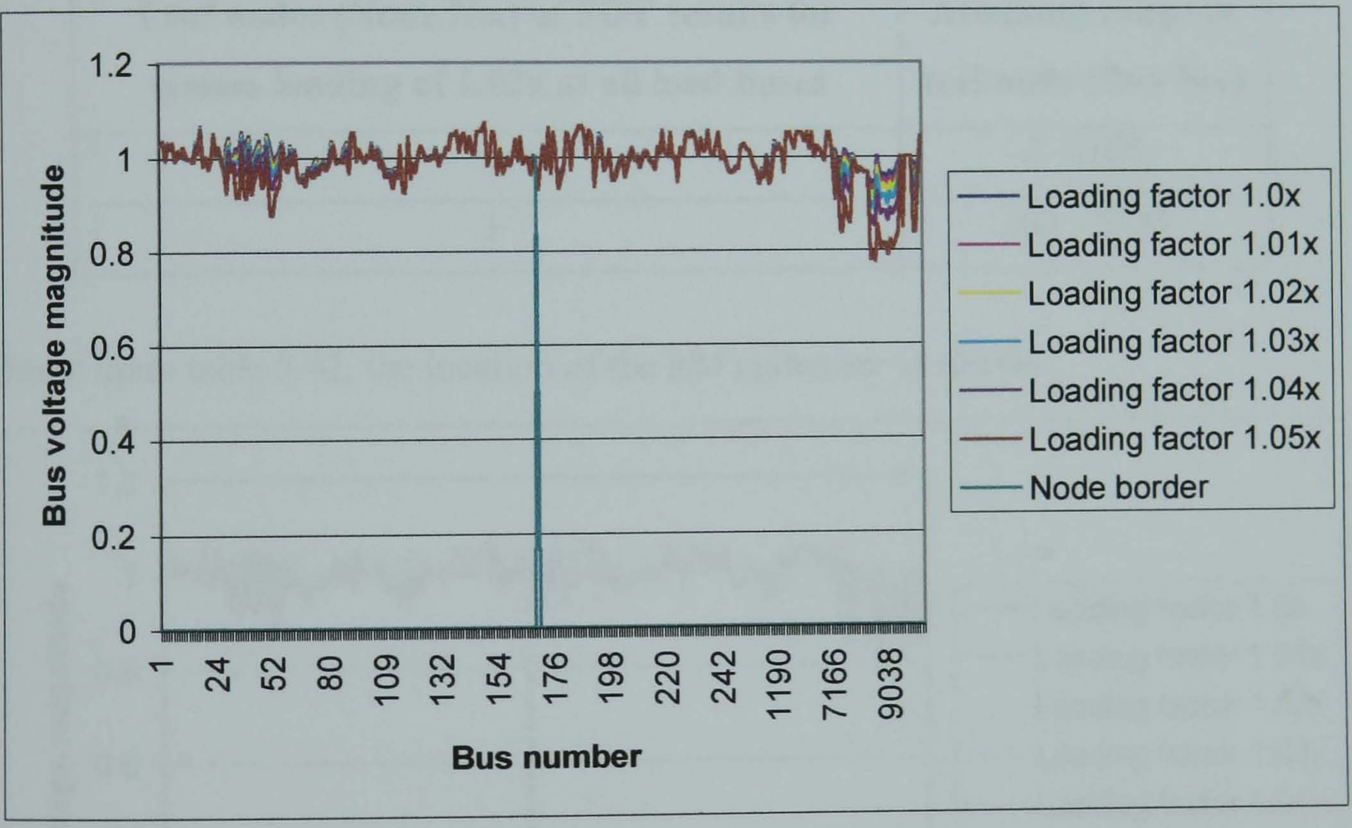


Figure 5.49a: Leaf node location for all load bus loading strategy

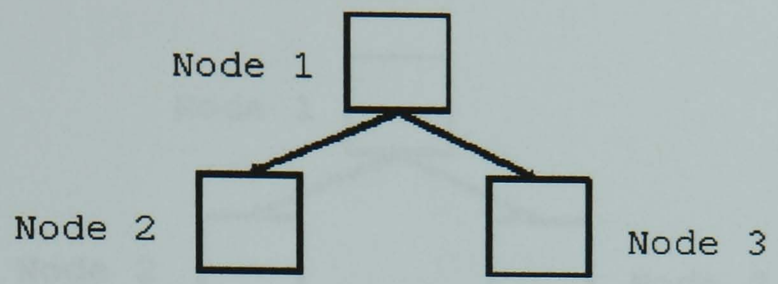


Figure 5.49b: Nodal map of FDT

Loading factor 1.02x

Table 5.42: Summary of results for loading factor of 1.02x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.02x at all load buses	Attribute range in leaf node (Bus No.)
2	1 – 165
3	166 - 9533

Based upon table 5.42, the location of the leaf nodes are as shown:

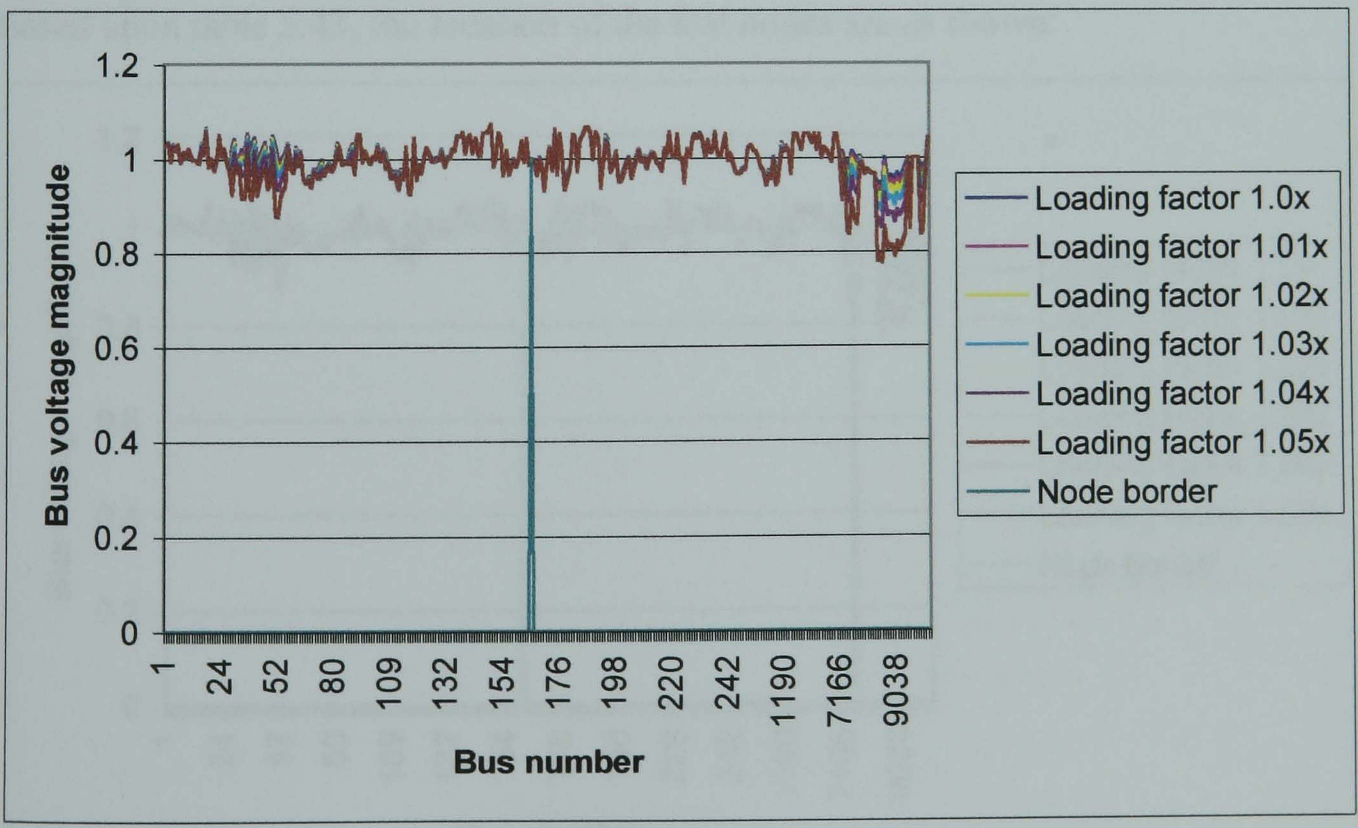


Figure 5.50a: Leaf node location for all load bus loading

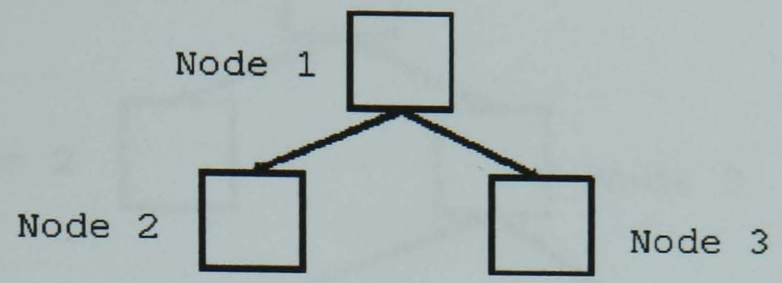


Figure 5.50b: Nodal map of FDT

Loading factor 1.03x

Table 5.43: Summary of results for loading factor of 1.03x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.03x at all load buses	Attribute range in leaf node (Bus No.)
2	1 – 161
4	162 – 9005
5	9006 - 9533

Based upon table 5.43, the location of the leaf nodes are as shown:

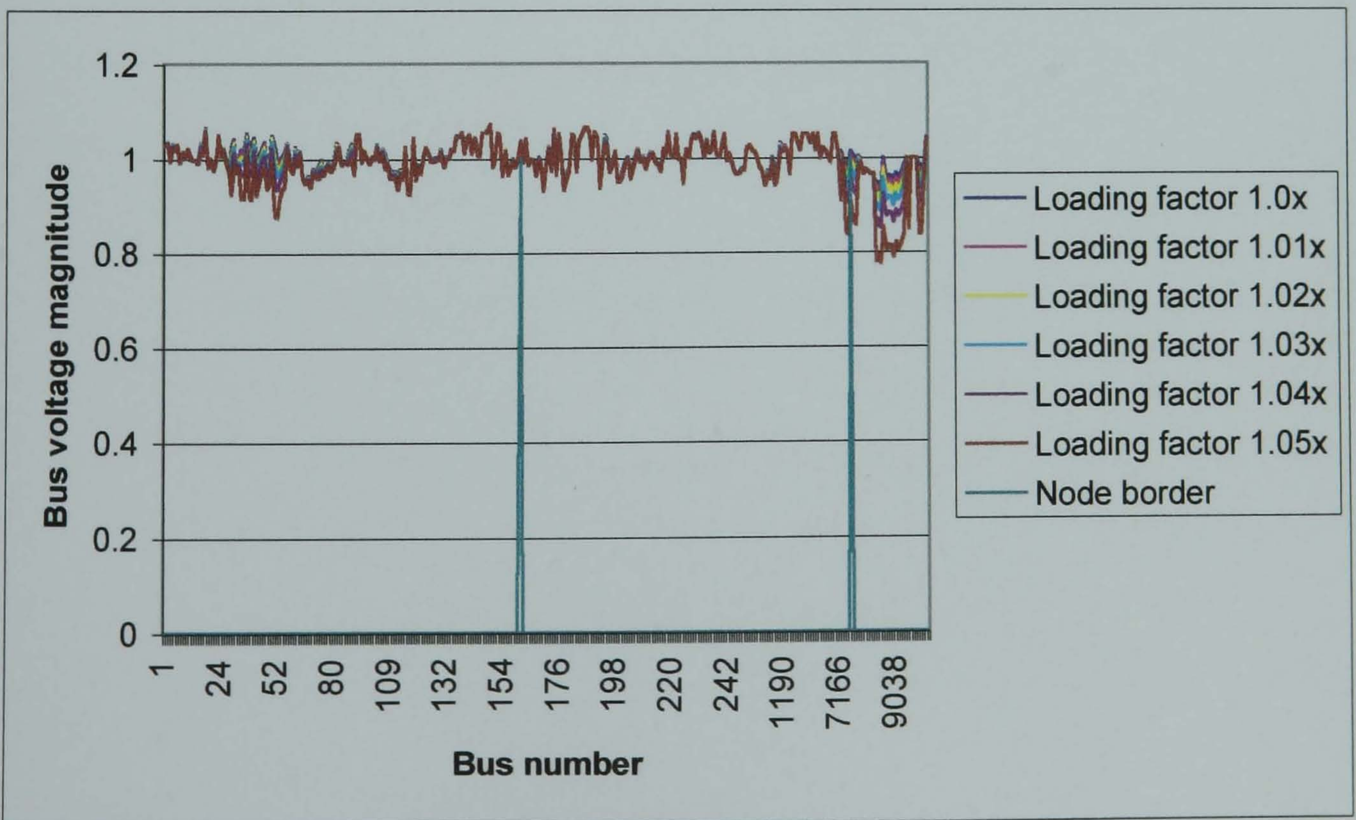


Figure 5.51a: Leaf node location for all load bus loading

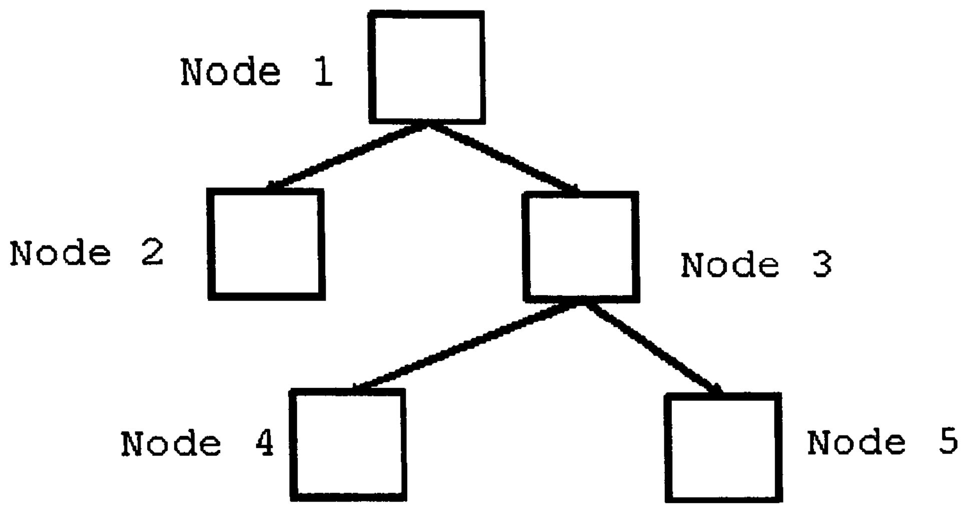


Figure 5.51b: Nodal map of FDT

Loading factor 1.04x

Table 5.44: Summary of results for loading factor of 1.04x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.04x at all load buses	Attribute range in leaf node (Bus No.)
2	1 – 120
4	121 – 9005
5	9006 - 9533

Based upon table 5.44, the location of the leaf nodes are as shown:

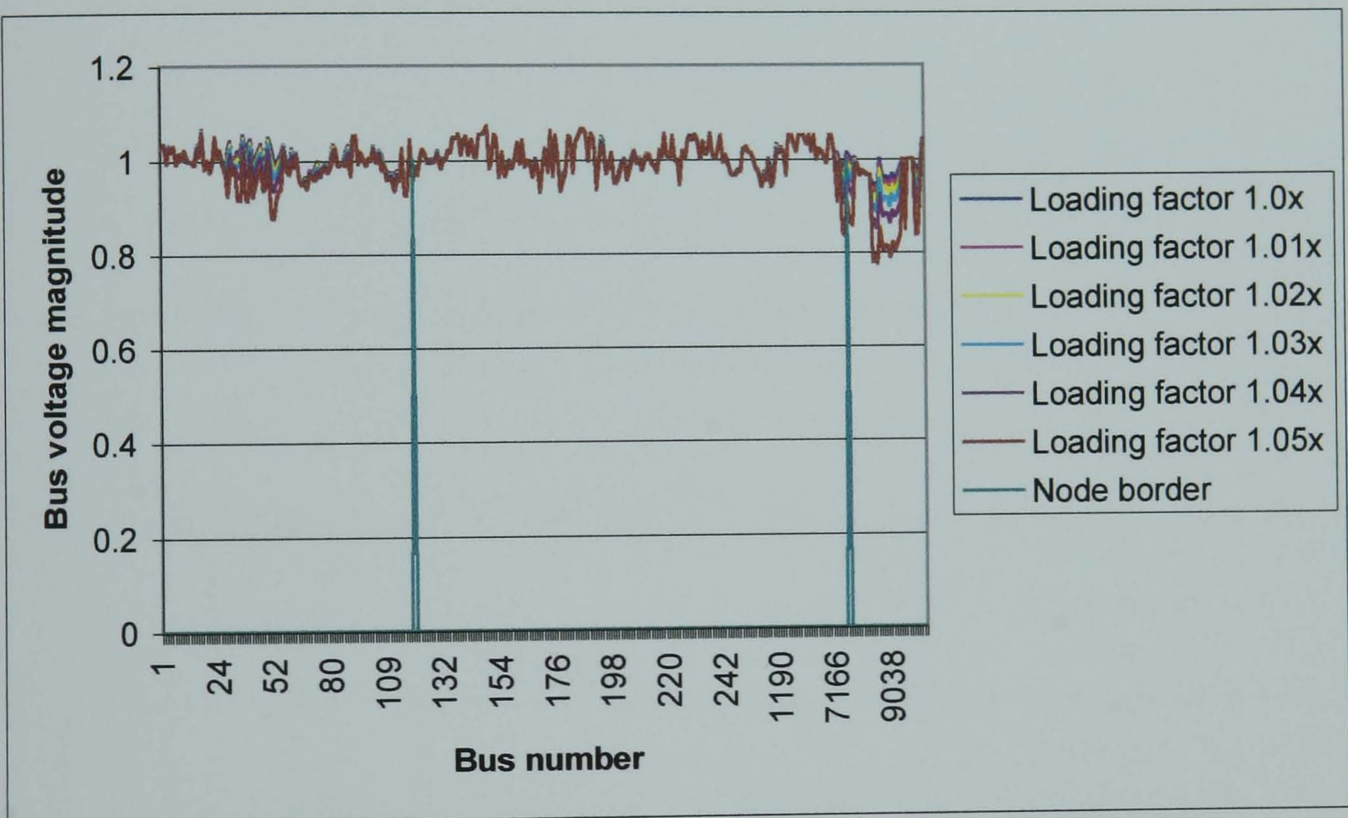


Figure 5.52a: Leaf node location for all load bus loading

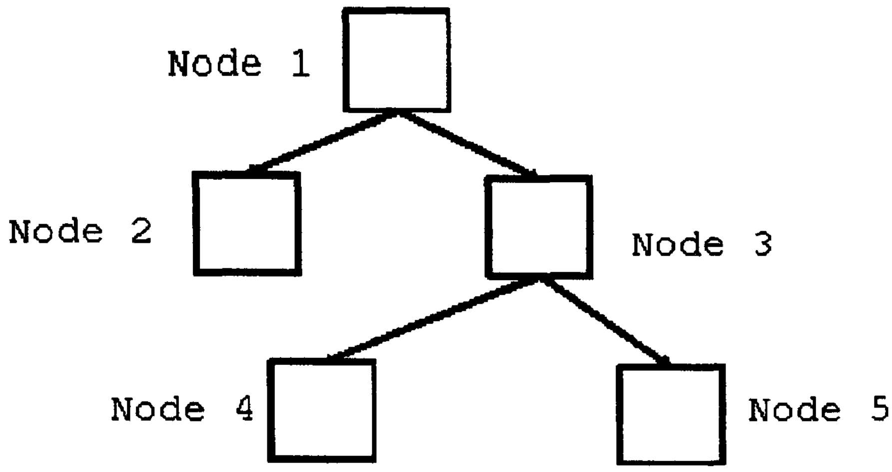


Figure 5.52b: Nodal map of FDT

Loading factor 1.05x

Table 5.45: Summary of results for loading factor of 1.05x on all load buses.

Leaf nodes (Node No.) of FDT results for system loading of 1.05x at all load buses	Attribute range in leaf node (Bus No.)
14	1 – 19
15	20 – 60
13	61 – 77
11	78 – 79
9	80 – 120
5	121
6	122 – 7166
7	9001 - 9533

Based upon table 5.45, the location of the leaf nodes are as shown:

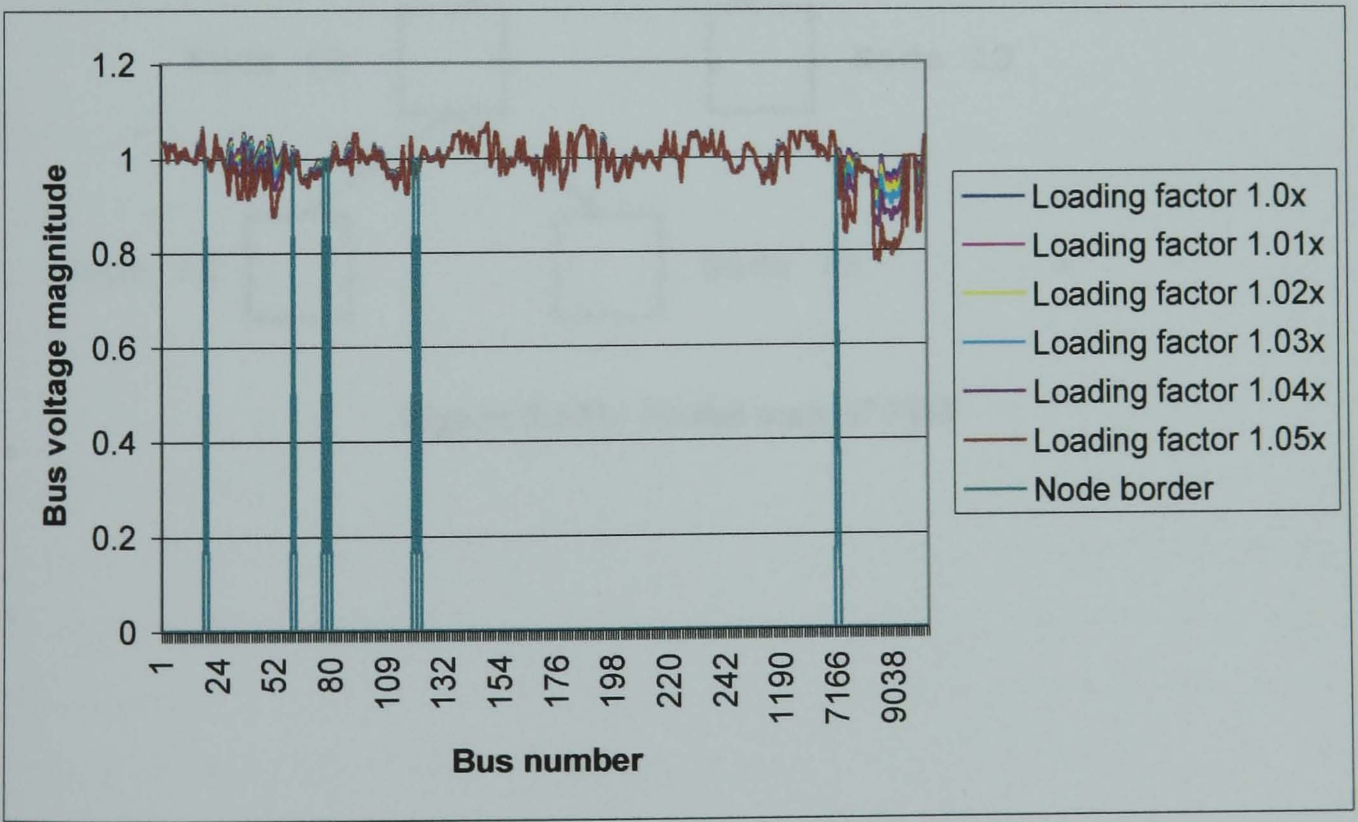


Figure 5.53a: Leaf node location for all load bus loading

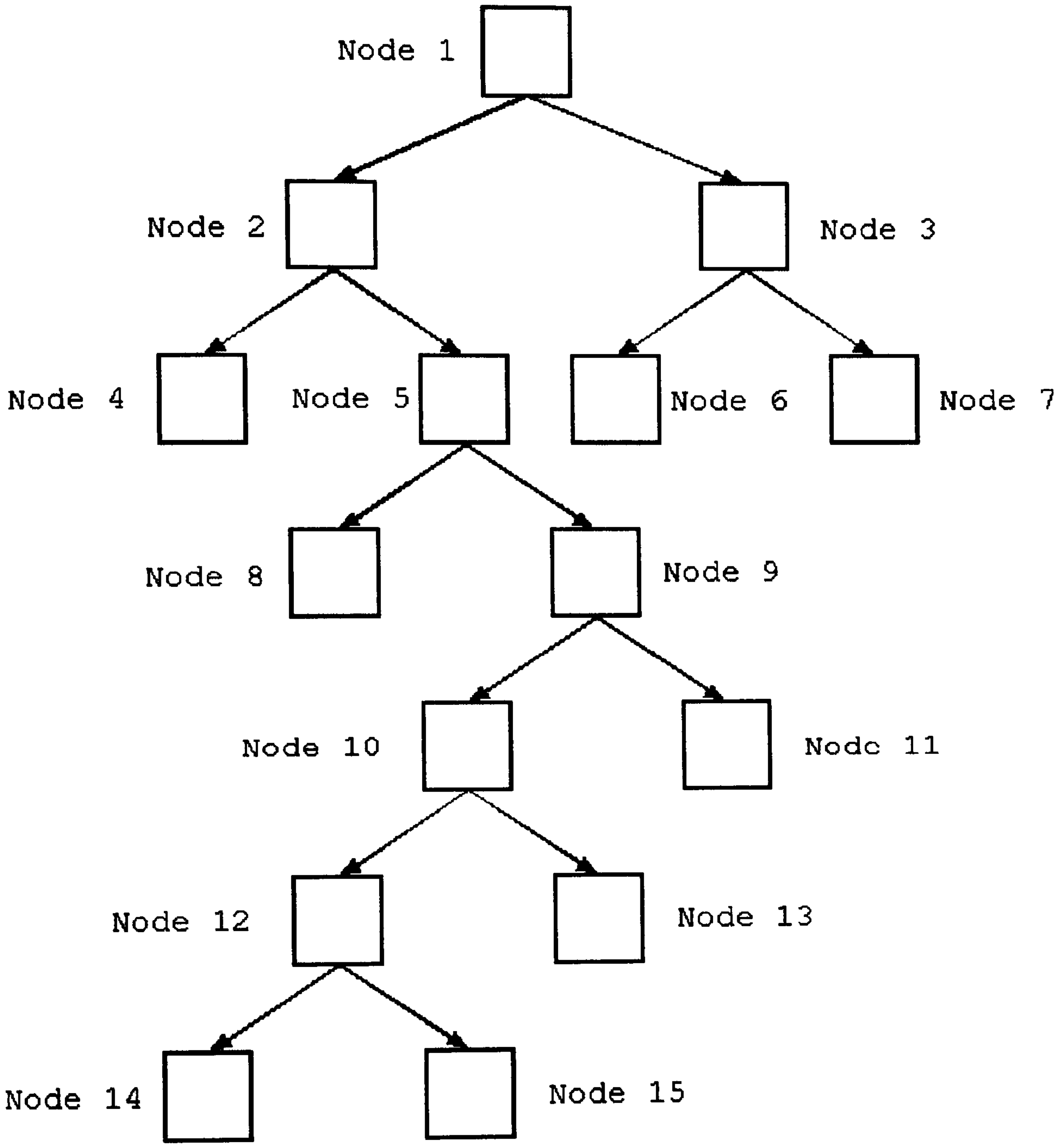


Figure 5.53b: Nodal map of FDT

Discussions

Comparing between test 5 (using hybrid FDT) and test 3 (using standard FDT) there are several improvements that can be seen. They are presented below:

a) Partitioning position/location

   Comparing between the two tests, test 5 gives a more consistent and predictable leaf node border.

b) Leaf node partition accuracy

   In terms of leaf node partition accuracy, test 5 gives better and more accurate representation. This is most significant at loading factor of 1.05x. In test 3, there are 9 leaf nodes and in test 5 there are only 8 leaf nodes. Although there are more instances of over partitioning at the highest loading factor in test 5 indicated at leaf node No. 5 and 11. But if the FDT uses a pruning sequence to the finalised results, the FDT would give better overall performance. This will be further discussed in the future work presented in chapter 6.

The overall conclusion for this test gives the indication that using other attribute values and combining it into the FDT will improve the FDT in terms of its performance and accuracy. The program code for this FDT is as shown in Appendix O.

5.19 Conclusion

In this chapter, the development of an FDT is explained and proved via preliminary test illustrated in the appendices and also in the main tests presented in this chapter. Although the developed FDT does not follow the traditional guideline of machine learning practices, this form of FDT opens up to a new avenue of dynamic FDT construction, which is fast in terms of computational speed and does not require initial learning stage. This in general gives the FDT good flexibility and adaptability especially towards a larger framework of voltage collapse analysis and mitigation solutions. This issue will be discussed further in chapter 6.

5.20 References

1. K L Lo, H.I Haji Zainal Abidin, *The Fuzzy Decision Tree Application to a Voltage Collapse Problem.* in *University Power Engineering Conference (UPEC).* 2001. University of Swansea, Wales.
2. Xavier P Boyen, *Design of Fuzzy Logic Based Decision Trees Applied to Power System Transient Stability Assessment*, MSc thesis in *Department of Electrical and Computer Science.* University of Liege, Belgium.1995, 133 pages
3. Xavier P Boyen, L.Wehenkel, *Automatic Induction of Continuous Decision Trees.* Procedings of Information Processing and Management of Uncertainty in Knowledge-Based Systems, p. 419 - 424.
4. Louis Wehenkel, C.Lebrevelec, M Trotignon, J Batut. *A probabilistic approach to the design of Power System protection schemes against blackouts.* in *IFAC-Cigre Symposium on Control of Power Systems and Power Plants*, 1997. Beijing, p. 506-511.
5. Louis Wehenkel, I.Houben, M Pavella, L Riverin, G Versailles. *Automatic Learning Approaches for On-line transient stability preventive control of the Hydro-Quebec System. Part 1: Decision Tree Approaches.* in *IFAC Symposium on Control of Power Plants and Power Systems.* 1995. Cancun, Mexico, p.231-236.
6. Xavier P Boyen, L.Wehenkel, *Fuzzy Decision Tree Induction for Power System Security Assessment.* in *IFAC Symposium on Control of Power Plants and Power Systems.* 1995. Cancun, Mexico, p. 151-156.
7. Louis Wehenkel, *Automatic Learning Techniques in Power Systems.* The Kluwer International Series in Engineering and Computer Science. Vol. 429. 1997: Kluwer Academic Publishers, 279 pages. ISBN 0-7923-8068-1.
8. Friedman, J.H., *A Recursive Partitioning Decision Rule for Nonparametric Classification.* IEEE Transactions on Computers, 1977, p. 404 - 408.

9. Janikow, C., *Fuzzy Decision Trees: Issues and Methods.* IEEE Transactions on Systems, Man and Cybernetics, 1998. **28**(1): p. 1 - 14.

10. Quinlan, J.R., *Improved used of Continuous Attributes in C4.5.* Journal of Artificial Intelligence Research, 1996. **4**: p. 77 - 90.

11. Engineering_Statistics_Handbook, *Quantitative Techniques.* Online reference, http://www.itl.nist.gov/div898/handbook/eda/section3.

12. Ray D Zimmerman, D.Gan, *MATPOWER - A MATLAB(TM) Power System Simulation Package.* 1997, Power System Engineering Research Centre (PSERC), School of Engineering, Cornell University: Ithaca, New York.

13. Ian Dobson, Thierry Van Cutsem, Costas Vournas, Chris DeMarco, Mani Venkatasubramanian, Tom Overbye, Claudio Canizares, *Voltage Stability Assesment, Procedures and Guides: Chapter 2: Basic Theoretical Concept.* IEEE/PES Power System Stability Subcommittee, Special Publication. January, 2001, p. 2-100

14. A Nuhavonic, M.Glavic, N Ptljaca, *Validation of a Clustering Algorithm for Voltage Stability Analysis on the Bosnian Electrical Power System.* IEE Proceedings of Generation, Transmission and Distribution, 1998. **145**(1), p. 21-27.

15. K L Lo, L.K.Peng. *Design of Artificial Neural Networks for On-line static Assessment problems.* in *IEE International Conference on Advance in Power Systems Control, Operation and Management.* 1997. Hong Kong.

16. M Begaovic, J.Bright, T Domin, S. Easterday-McPadden, A Girgis, W Hartmann, C Henville, M Ibrahim, K Kozminski, R Marttila, G Michel, K Mustaphi, D Novosel, B Pettigrew, M Sachdev, H Shuh, P Solanics, J Williams, *Voltage Collapse Mitigation.* 1996, IEEE Power System Relaying Committee, 33 pages.

17. Mukhedkar R. A, Nouri H., Davies, T.S. *Comparitive study and performance analysis of online static voltage collapse indices in interconnected systems.* in *University Power Engineering Conference (UPEC).* 2001. Swansea, Wales, United Kingdom.

18. Navarro-Perez, R., *Voltage collapse proximity assessment for the operational planning of power systems*, in *Department of Electrical Engineering*. 1993, PhD thesis, Imperial College of Science, Technology and Medicine: University of London. 207 pages.

19. Souza, A.C.Z.d., *New Techniques to Efficiently Determine Proximity to Static Voltage Collapse*. PhD thesis 1996, University of Waterloo. 133 pages.

20. Taib, M., *Possibility of Voltage Collapse in the TNB Power System*. 1997, Network Planning Department, Tenaga Nasional Berhad, Malaysia. 18 pages.

21. Thierry Van Cutsem, C.Vournas, *Voltage Stability of Electrical Power Systems*. 1998 ed. 1998: Kulwer Academic Publishers. 378 pages.

22. Yuan-Lin Chen, Wen-Bin Liao, Shun-CHung Wang, Kun-Yuan Shen, Shih-Chung Tuan. *Voltage Collapse Proximity Indicators of a Power System*. in *University Power Engineering Conference (UPEC)*. 2001. Swansea, Wales, United Kingdom.

Chapter 6: Conclusions and future work

6.1 Conclusions

This research has demonstrated an alternative approach of using FDT techniques as a tool for voltage collapse analysis. The designed algorithm is based upon conventional FDT and machine learning concepts with modifications for voltage collapse analysis.

The algorithm was designed and developed from the basic FDT engine. By observing its performance from test results, its structure was improved by adding or optimising the information quality assessment and partitioning stage. Test results using an IEEE test system with varying loading strategies showed that the FDT is capable of partitioning the power system network into areas of weak and strong buses. The developed FDT conclude with a novel FDT approach utilising 2 attribute partitioning system with dynamic tree building capabilities. Test results of this new algorithm gave promising results with indications of its future development into a voltage collapse analytical tool.

The term dynamic FDT can be explained in terms of its algorithm procedures. Conventional FDT techniques will construct its DT structure using a series of training sets. The constructed tree is then pruned and rechecked in terms of its generality and partitioning accuracy via a series of test sets. Finally, the constructed tree will be applied for daily use.

In comparison, the FDT presented in this thesis is designed with reduced training set requirement. The only data set needed apart from the current power flow results is the reference data set based upon the power flow simulation result during nominal load condition (load factor 1.0x). As new system variations are introduced to the FDT, it builds a new set of tree based upon its relation to the reference data set initialised earlier. Therefore, this approach gives an equivalent of a generalised system 'snap-shot' showing the system stability status via the size and location of the weak bus and strong bus areas. Although this representation may not be accurate with reference to voltage collapse concepts, but this algorithm can be further improved by adding more system parameters hence giving better representation of system stability.

Apart from that, this research gives a new and alternative perspective to voltage collapse analysis without direct manipulation of system jacobian and its corresponding equations. The developed algorithm analyses the system based upon observed values obtained from power flow simulation results. Due to the simplicity of its approach, the designed algorithm has a lot of potential for online capabilities.

Looking into the possibility of dimensionality problem, test results indicate that the FDT is capable of processing data from a 300-bus test system. Handling bigger systems in theory would require more computation time since the bigger the system, the more data the algorithm will have to handle. Possible solutions may include optimising the program structure via object oriented programming and parallel processing techniques.

6.2 Future FDT improvement issues

6.2.1 Information quality assessment and partitioning scheme

This is an important stage that determines FDT accuracy in terms of its leaf locations. Improving this stage would ultimately improve the general performance of the FDT. In the previous chapter, an introduction of a second attribute element into the FDT structure led to the modification of the KS scoring technique. Test results showed improvements in terms of the FDT leaf location and partitioning ability although it over-partitioned. However, the KS stage can be improved by adding other attributes taken directly from the power flow simulation. In theory, this will enhance the value and usefulness of the KS score. These attributes may be in a form of equipment states and limit level that can be converted into fuzzy weights to enhance the significance of the KS profile.

Examples of these attributes are as follows:

a) Generator operation status

In practice, a system control centre incorporates the use of the AGC (Automatic Generator Control) to handle small variations of load in the system. The AGC will instruct relevant generators in the system to increase or decrease its generated output to meet the varying load demand. Apart from that the system control centre will monitor system generation level via installed remote terminal units (RTU). Since voltage collapse corresponds directly to system reserves inadequacy, a list of generator operation status may be beneficial in giving a better representation of the system operating level and its support capabilities to the surrounding areas. Utilising these information such as maximum generation capacities, generator locations, speed of generation responses and also their reactive capabilities, a more detailed and meaningful representation in terms of generator effects to the system and hence improving the performance of the FDT.

b) Support devices status

Apart from despatched system generators, there is a need to monitor supplementary reactive power support located in the system. As mentioned in chapter 2, the purpose of reactive power supports is to maintain a stable operating level during variations of system demand. They may be in a form of capacitor banks, dynamic and static compensators. Taking into account their current operating level and maximum limit, incorporating these information as fuzzy weights may help enhance the KS score profile and hence improving the overall FDT result.

c) Line status

As part of the daily network maintenance schedule, certain parts of the transmission line may be offline for routine repairs or for a check on system components. To ensure system security throughout the maintenance programme, the system operator will plan alternative power supply routes without much disruption to the customers. In addition to that, some lines may be offline due to the operation of protecting schemes. Utilising information on line outage, location and line limits fuzzy risk weights can be added into the FDT algorithm ensuring better results.

In terms of partitioning scheme, Tests 3-5 demonstrate the use a form of a rule based partitioning scheme using maximum KS score and its location. This partitioning scheme will first select the maximum KS score and its location in the test node. If the location of the maximum KS value is within the pre-defined limit from each end of the test node, the algorithm will choose this location as a partitioning limit. If not the algorithm will automatically search for the location of the minimum KS score. If its location is within the predefined limit from each end of the test node, the algorithm will use this location as a partition limit. If not, the algorithm will label this test node as a leaf node.

Looking at the bus voltage profile at varying loads presented in chapter 5, weak bus areas may include several buses located near to each other. Therefore partitioning can be improved by using the program steps in figure 6.1.

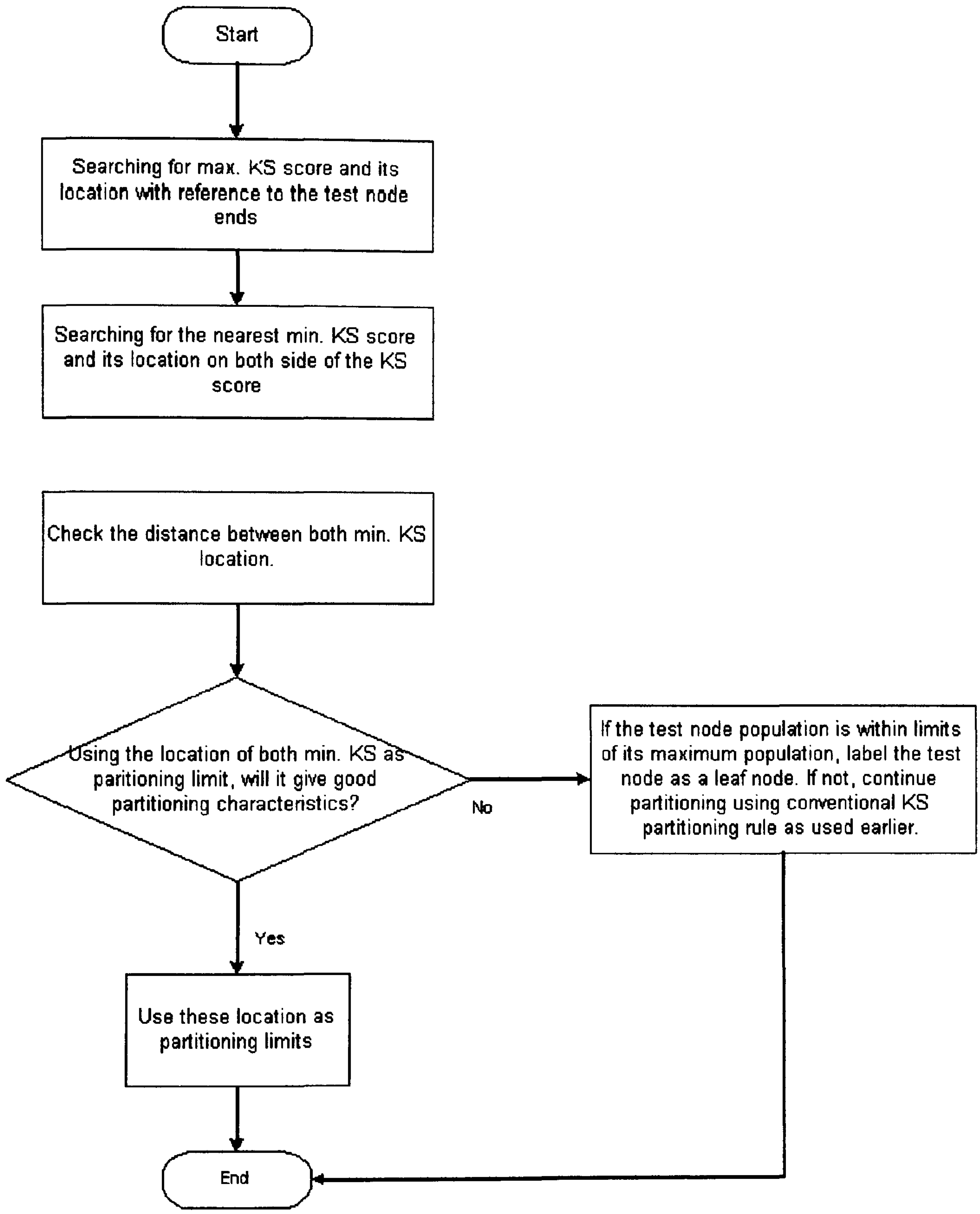


Figure 6.1: Improved partitioning scheme

Referring to figure 6.1, there is a need to address the definition of good partitioning characteristics. From the FDT concepts, good partitioning characteristic is defined as the algorithm's ability in partitioning the required sections of the data set. In terms of bus voltage levels, this definition is presented in terms of the FDT ability to partition a test node as such that the resultant nodes will consist of a majority weak or a majority strong buses. If this is possible without over-partitioning the FDT, then the FDT is known to be able to exhibit good partitioning capabilities.

Basically, rather than looking only for the location of maximum KS score, the algorithm can be improved by locating its neighbouring minimum KS locations on both side of the maximum KS score location. If using both minimum KS locations give a good partitioning characteristic, these locations will be used as partitioning limits. The corresponding area covered by these range will be labelled directly as a leaf node and the rest of the test node will go through further partitioning if required. Therefore, if this improvement is applied to the FDT, certain portions of the tree will experience partitioning into 3 sections rather than 2 sections with the 3 rd section automatically labelled as a leaf node. This would simplify the generated tree structure and may not require the use of a pruning algorithm and it will make the FDT more efficient.

6.2.2 Stop split stage improvements

The current stop split rule checks for population size in the test node and its purity content. This stage can be improved by projecting the location of possible maximum and minimum KS, vital for future partitioning. This is a form of look-ahead schemes checking for future partitioning possibilities. If these locations show possible partitioning, partitioning will continue. If not, the test node will be labelled as a leaf node.

6.2.3 Addition of pruning stage

Although the improved FDT algorithm may not require a pruning stage, but having a pruning stage could be beneficial for unexpected cases for possible over partitioning occurrences. The created pruning algorithm will execute after the completion of the FDT by searching through all leaf nodes in terms of its population and partitioning characteristics. If there is over partitioning i.e. the population count of the leaf node is less than the minimum population limit, the corresponding branch and leaf node will be collapsed to its nearest test node, converting it into a new leaf node.

6.2.4 Utilising different defuzzification approach

This research applies a direct reversal of the fuzzification process for its defuzzification stage. However, the leaf node may be defuzzified as such that it will to create a form of FDT indicator relating the leaf nodes occupants to system stability. This would require further research in defuzzification techniques.

6.2.5 Voltage collapse analysis and contingency package

Finally, this section looks into the future possibility of using the FDT for a complete voltage collapse analysis with contingency analysis capabilities. Earlier in this chapter, there were discussions with regards to dynamic FDT concept and its possible future developments. Employing the dynamic FDT concept online, system 'snap-shots' can be made periodically. These 'snap-shots' are stored as historical data for future referencing. In doing so, it is possible to track changes in terms of the system operating level and stability status.

The proposed FDT implementation layout is as shown in figure 6.2. With reference to figure 6.2, efficiency of the complete package will rely on the FDT computation speed, accuracy of its results and also calculation time for the contingency analysis stage.

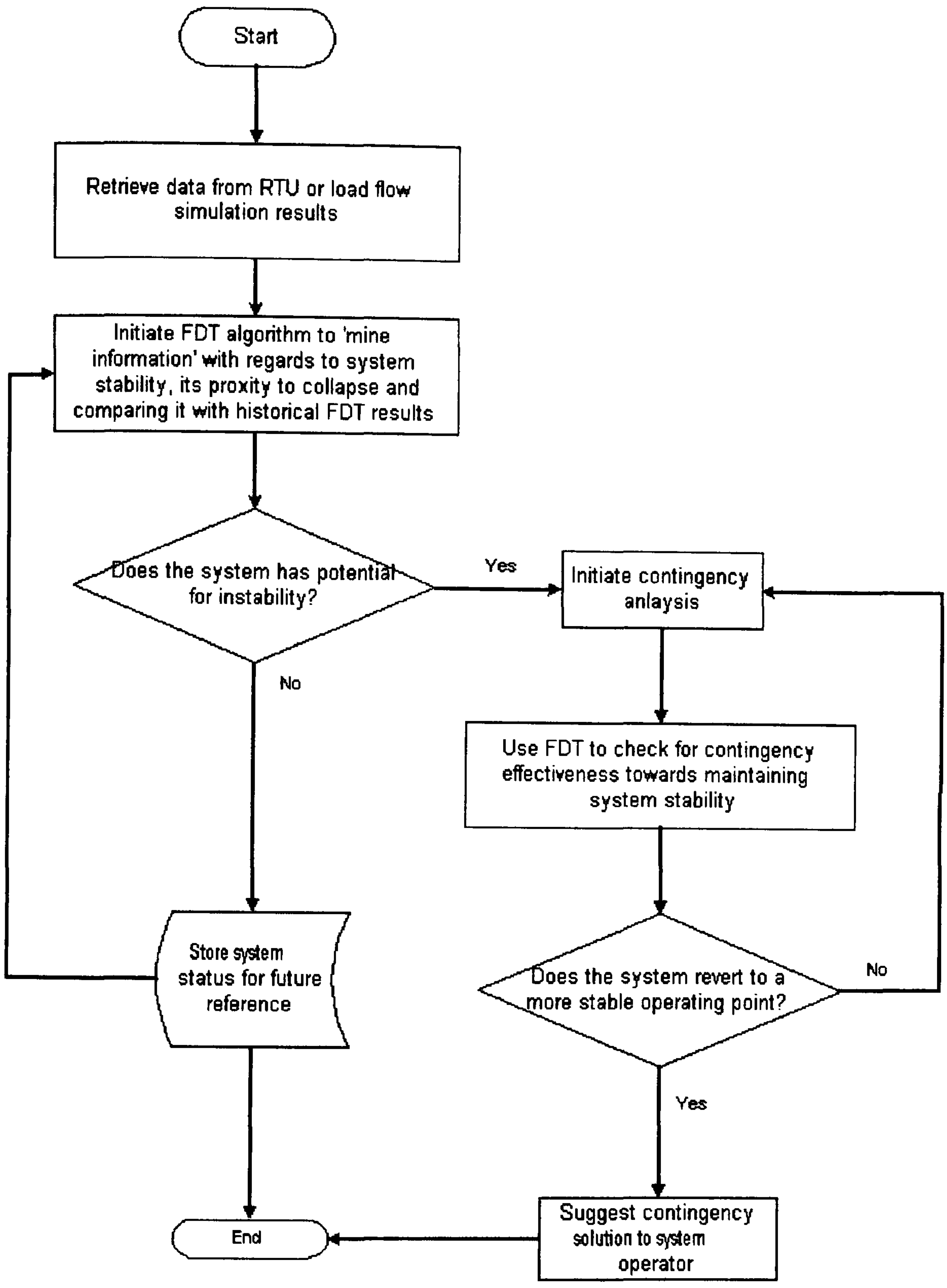


Figure 6.2: FDT implementation as a complete voltage stability and contingency package.

In relation to the contingency analysis stage, more research is required in terms of its techniques and algorithm optimisation. Assuming that research in this matter would yield promising results, implementing the overall package would mean online possibilities to aid system operators in maintaining system security and possible voltage collapse scenarios.

Appendix A

Varying the bus voltage reference value and its effect to the KS score

The data for the bus values are as shown:

Bus No.	Bus voltage magnitude
1	0.90668
2	0.906445
3	0.906209
4	0.905973
5	0.905738
6	0.905502
7	0.905188
8	0.904952
9	0.904717
10	0.904402
11	0.904167
12	0.903853
13	0.902596
14	0.89592
15	0.887202
16	0.875578
17	0.858849
18	0.636738
19	0.228958
20	0.101174
21	0.711116
22	0.711116
23	0.711116

Note that these bus voltage magnitudes are fictitious, created to simulate the functionality of the KS Score.

Using 0.90668 as a reference value and ξ set to 2

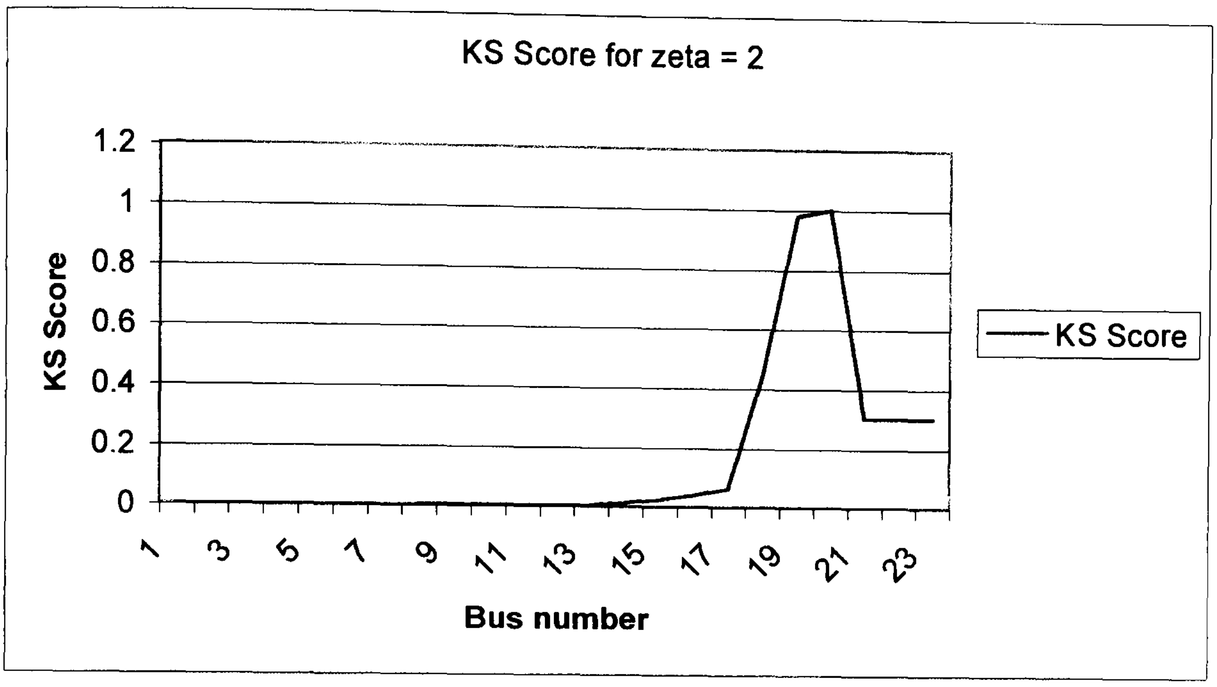


Figure A(1) : KS score with 0.90668 as a reference value

Using 0.960445 as a reference value and ξ set to 2

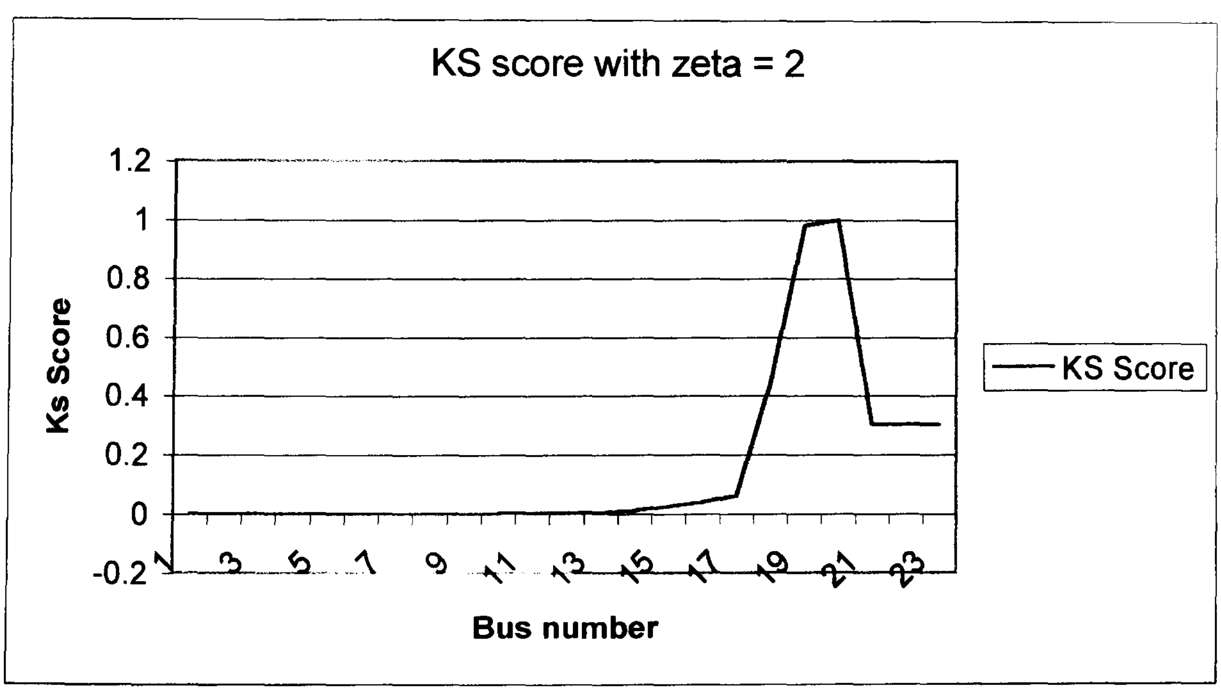


Figure A(2): KS score with 0.960445 as a reference value

Using 0.906209 as a reference value and ξ set to 2

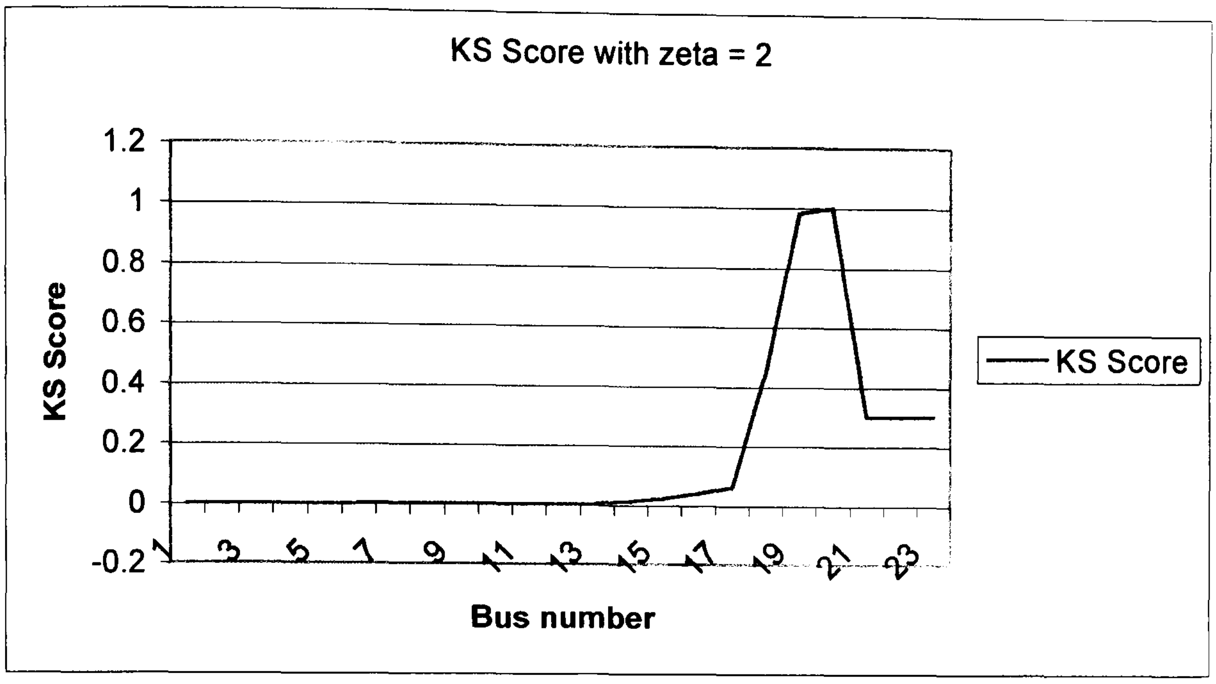


Figure A(3): KS score with 0.906209 as a reference value

Using 0.905973 as a reference value and ξ set to 2

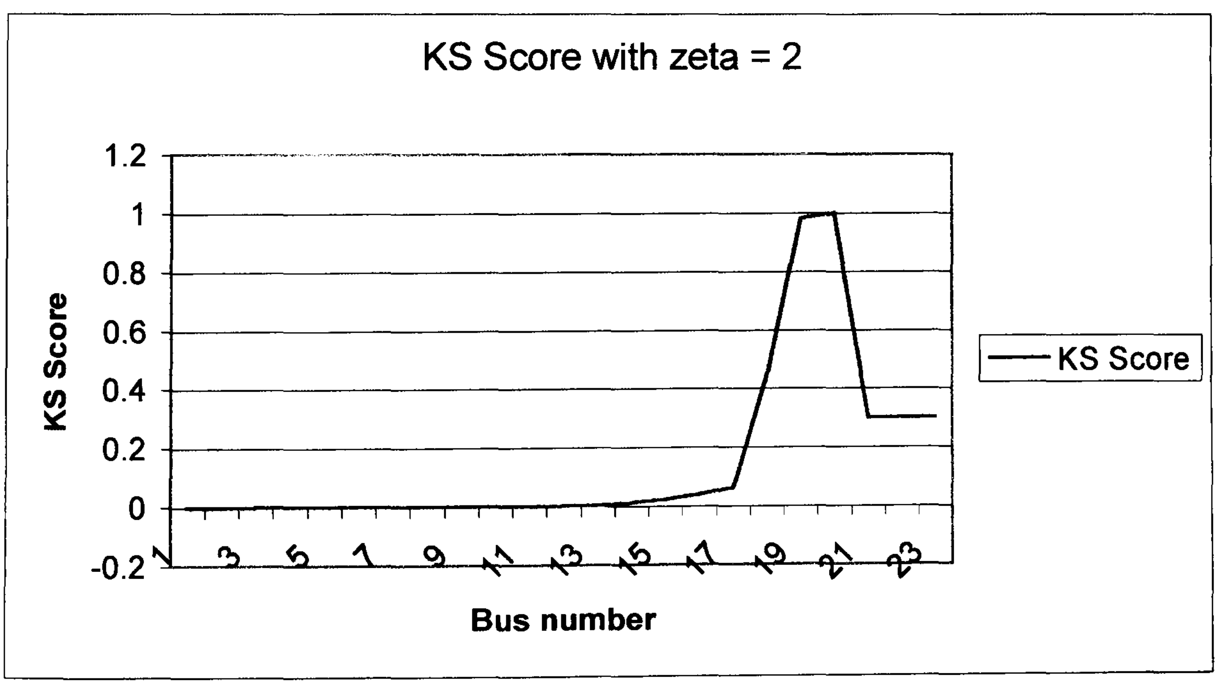


Figure A(4): KS score with 0.905973 as a reference value

Using 0.904167 as a reference value and ξ set to 2

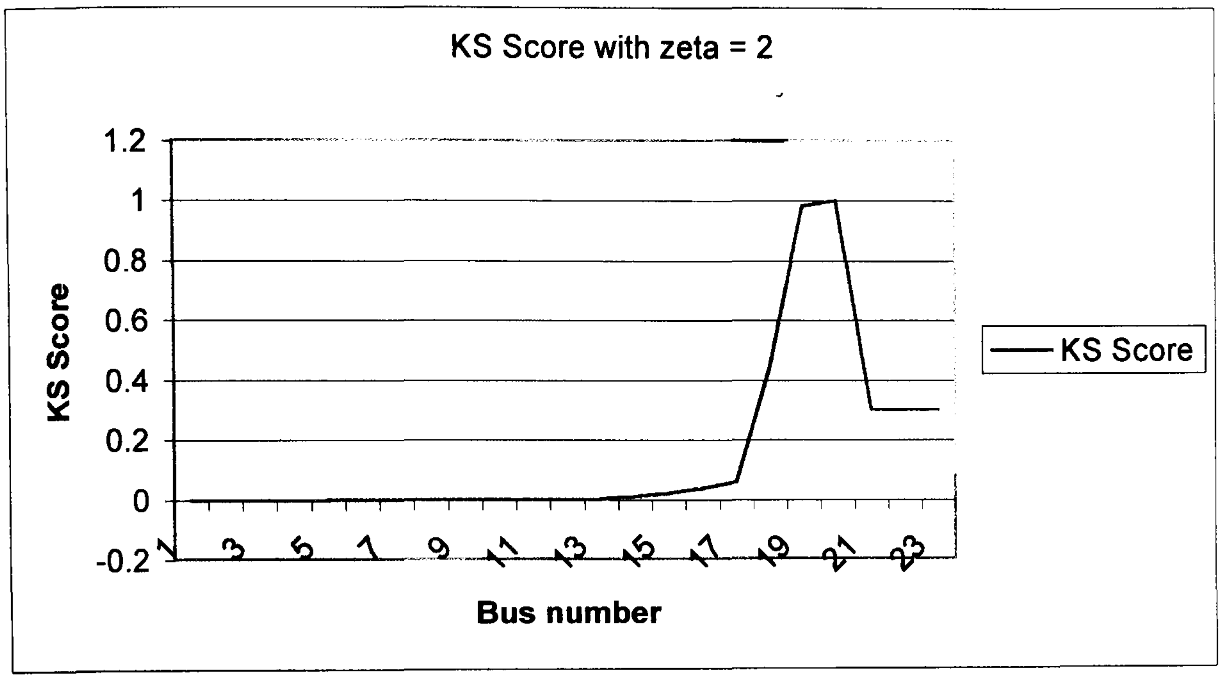


Figure A(5): KS score with 0.904167 as a reference value

Using 0.636738 as a reference value and ξ set to 2

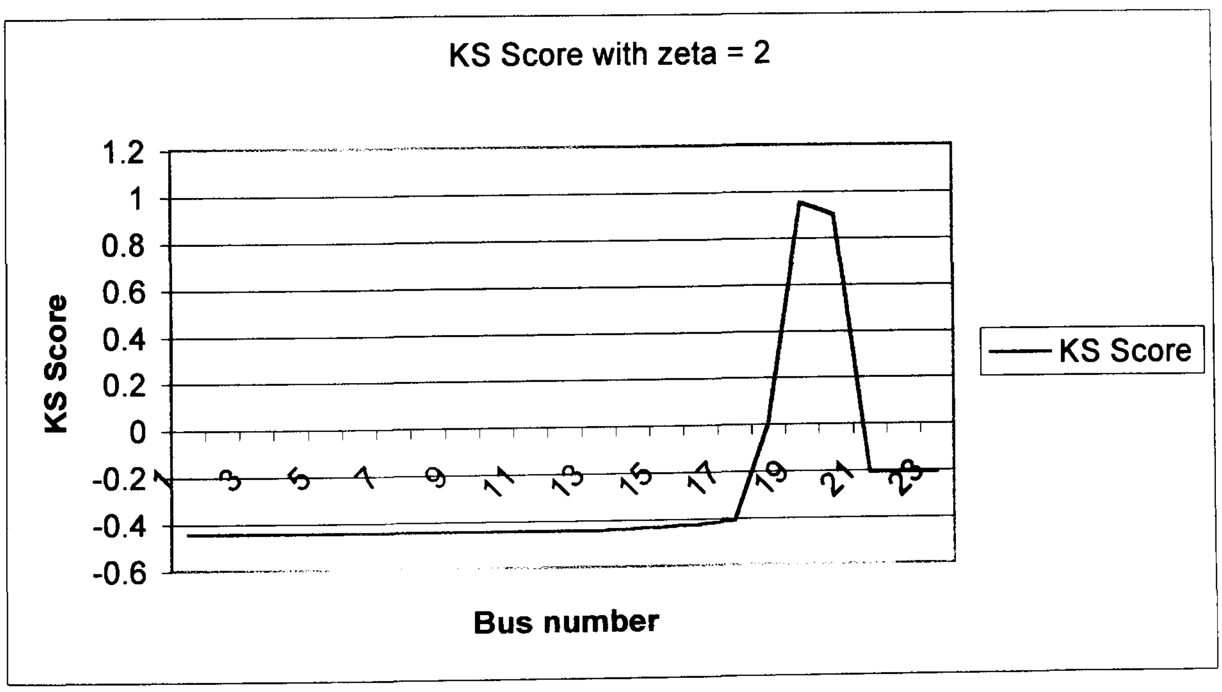


Figure A(6): KS score with 0.636738 as a reference value

Using 0.228958 as a reference value and ξ set to 2

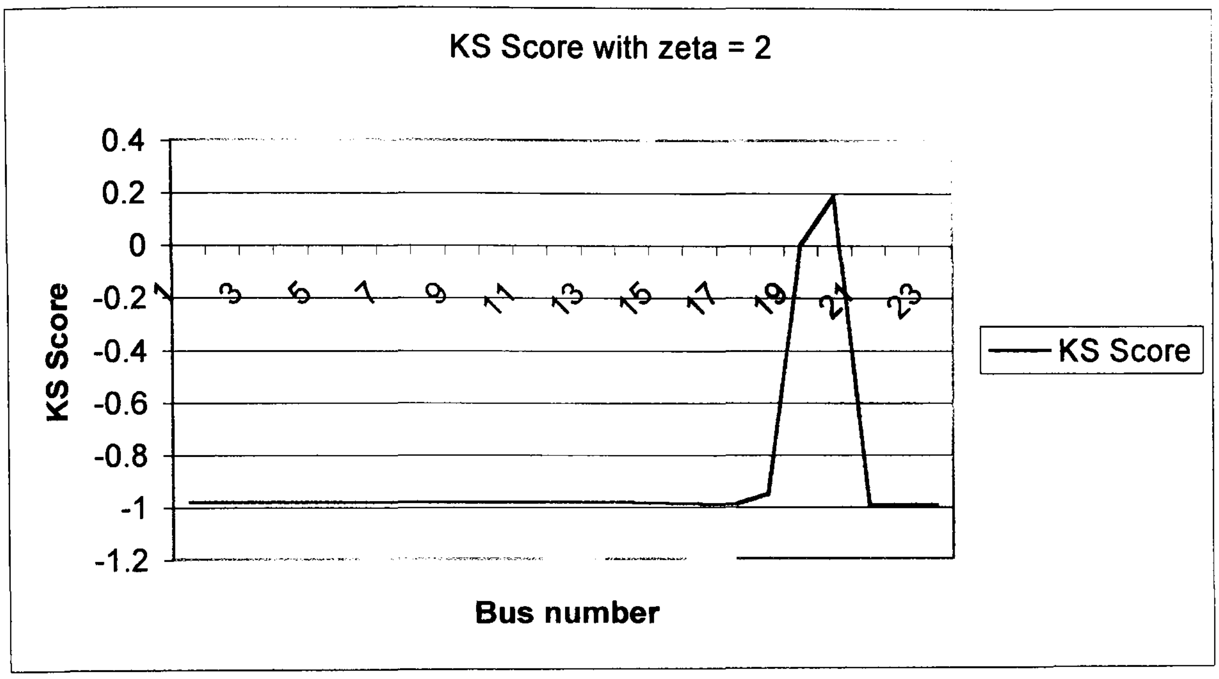


Figure A(7): KS score with 0.228958 as a reference value

Figure A(1-7), shows that the KS profile goes through a shift on the X-axis as the reference value is changed. But the most important point observed here is the maximum KS score is still at the lowest point of the bus voltage value, which in this case is for bus 20. This conclusion is valid provided that the interpretation of the KS score is not based upon absolute values.

Appendix B

Investigating the variation of ξ in the KS score profile.

Using the same bus data values denoted in Appendix 5a, an investigation is made to determine the variation of the KS score profile. For this investigation, the same reference value is used to calculate the KS score. The reference value is 0.90668

$\xi = 1$

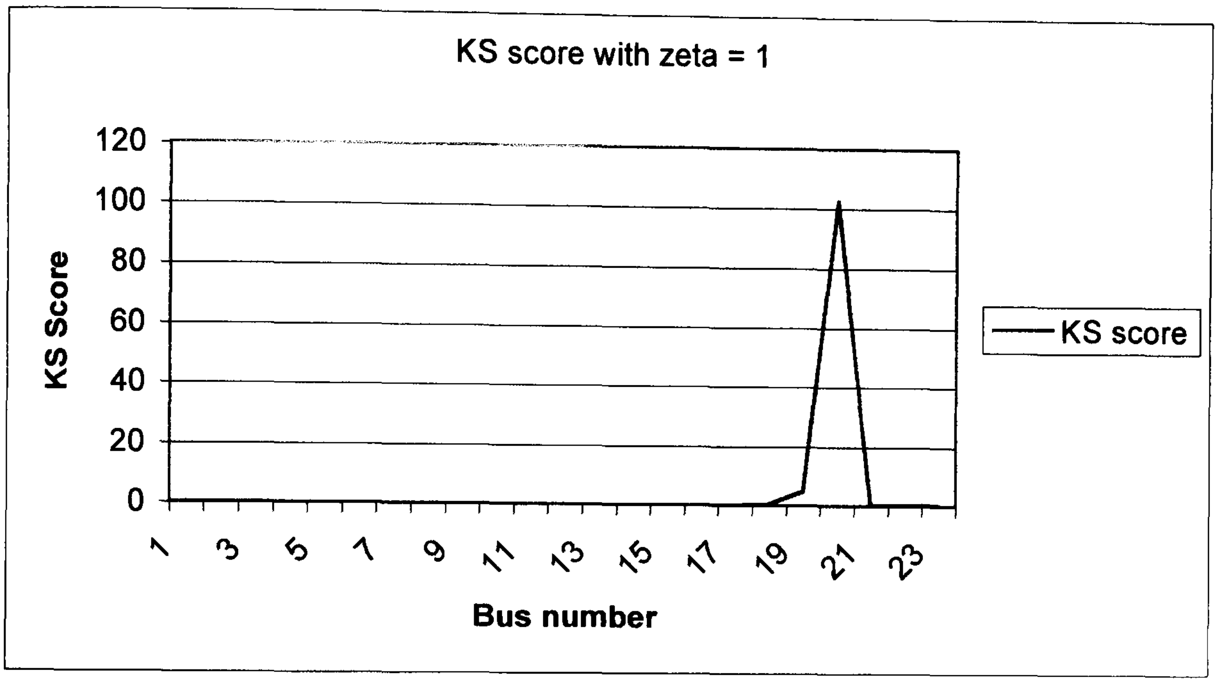


Figure B(1): KS score with ξ = 1

$\xi = 2$

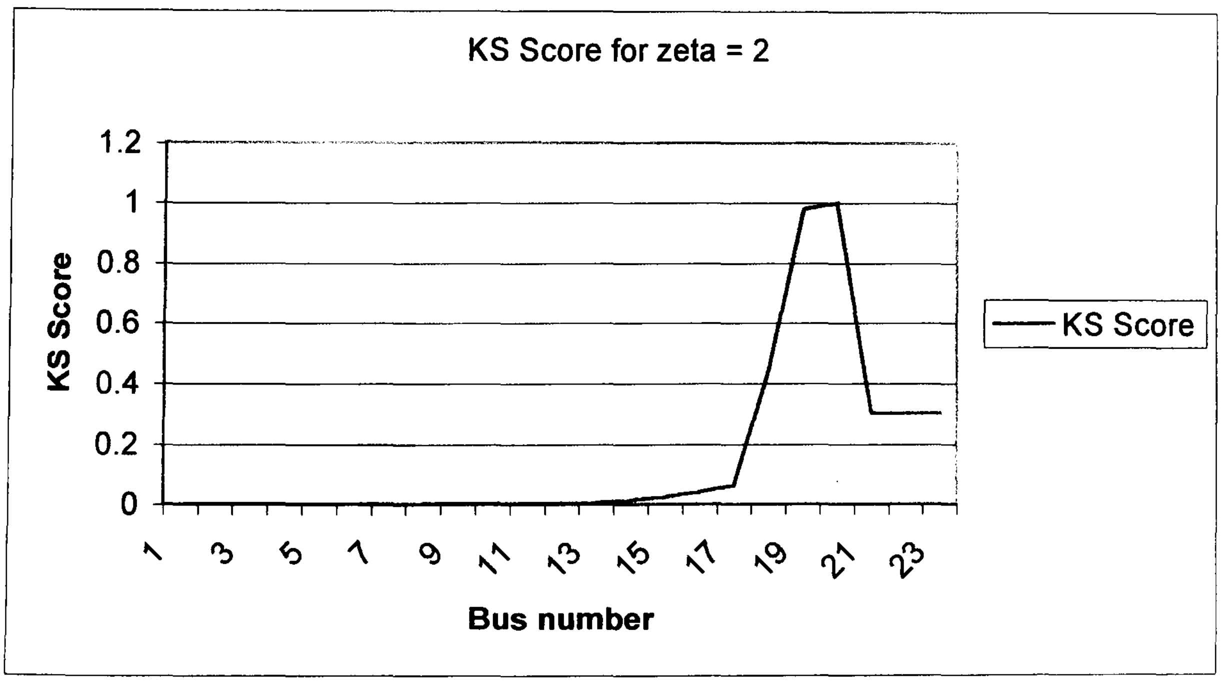


Figure B(2): KS score with ξ = 2

$\xi = 1000$

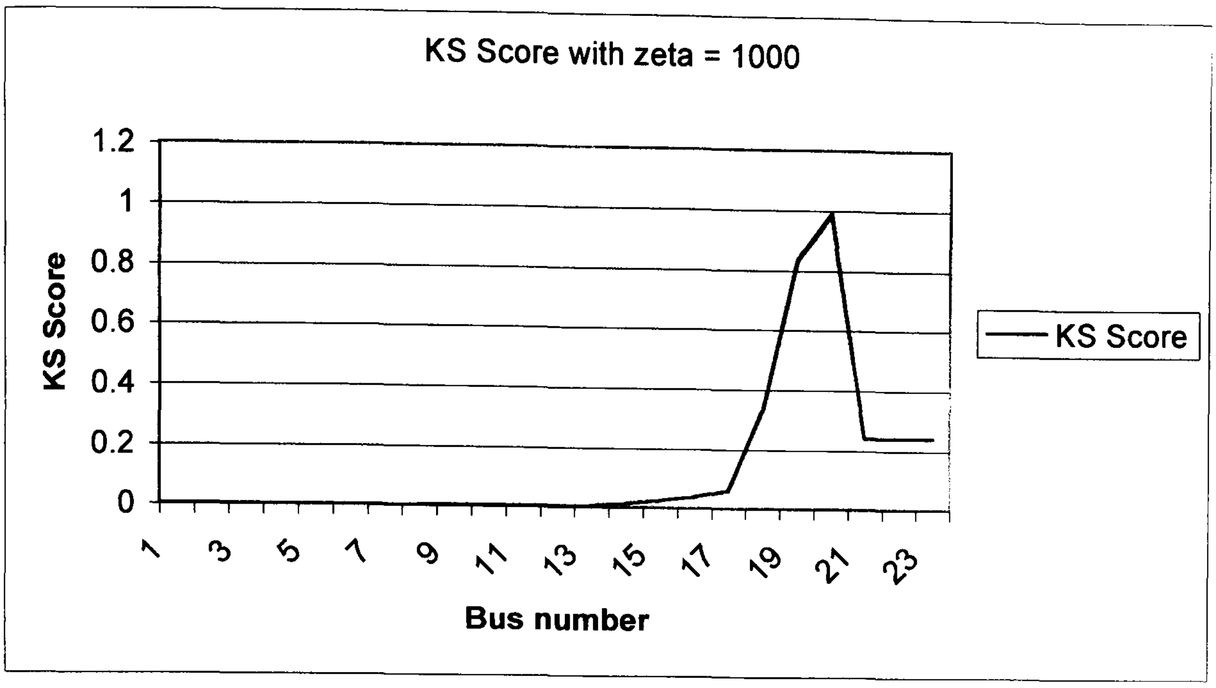


Figure B(3): KS score with $\xi = 1000$

$\underline{\xi = 1999}$

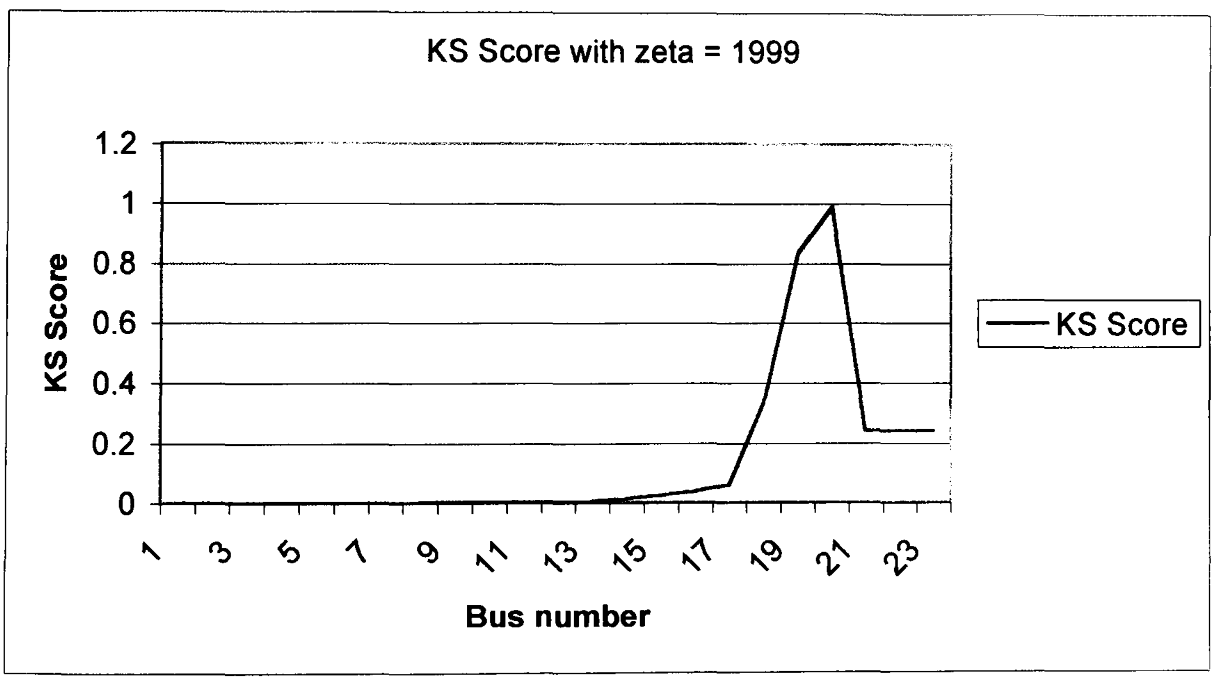


Figure B(4): KS score with $\xi = 1999$

Figure B(1-4) show that the characteristics of the KS score profile with changing ξ values. The most significant difference is when ξ is 1 and when ξ is 2. Looking at the KS score range, it is shown that the KS value changes in terms of magnitude and profile shape. As ξ is increased further, the KS profile is retains the same shape. Hence a conclusion can be made that the most suitable ξ value used is 2 since at this value, the KS profile are more proportional without loss of the overall shape.

Appendix C

Comparing the effects of different type of referencing technique to the KS score profile

There are other various referencing techniques that can be included into the KS score formula and its effects investigated. The data set used for this application will be similar to the ones used to generate the KS score profile in figure 5.7. The data set in question consists of an IEEE 300 bus system with a load increase by a factor of 1.01 at all load centres in the system. Values used for the KS formulation is the bus voltage magnitudes obtained from load flow simulations. Results shown in figure 5.7 will be repeated here again for comparison purposes.

<u>Using the first bus voltage magnitude as reference</u>

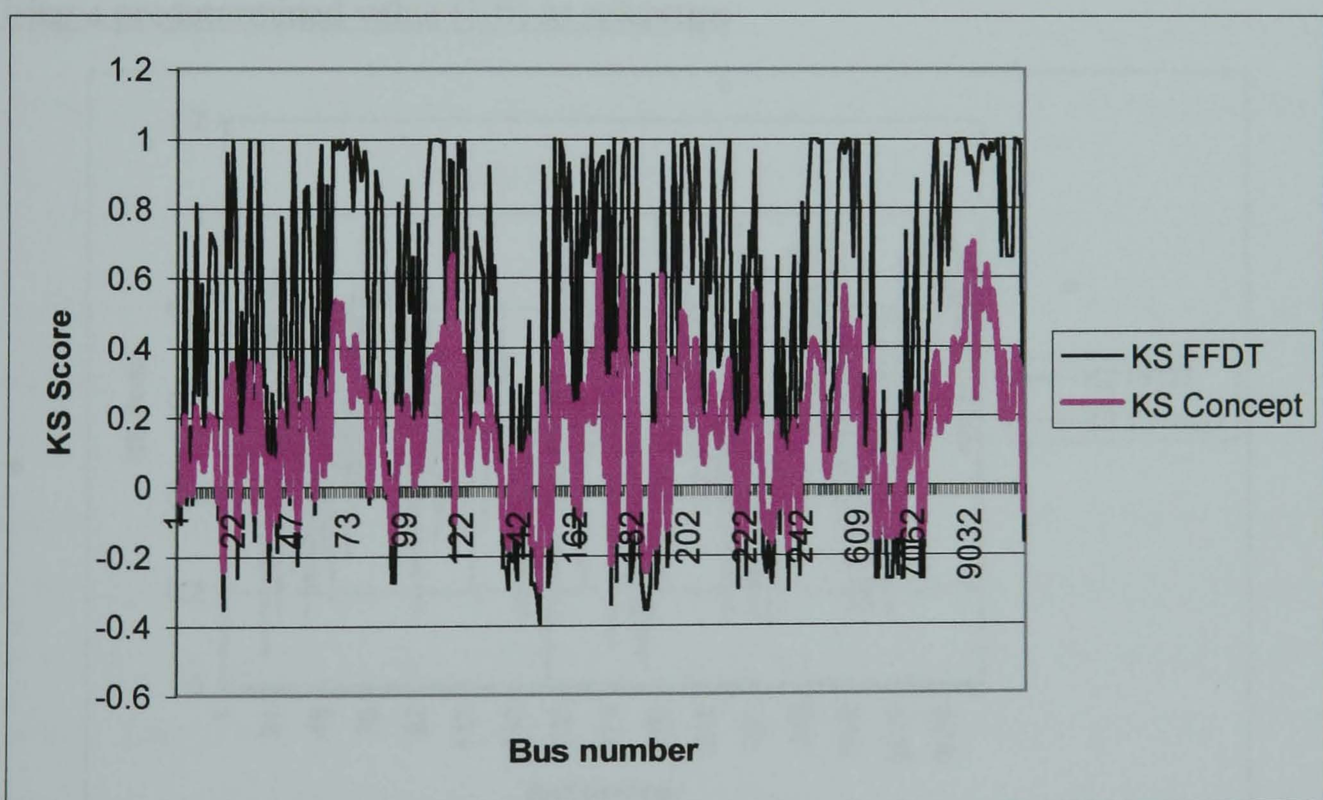


Figure C(1): KS score profile using the first bus voltage magnitude as a reference with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Using a zero (null) value reference

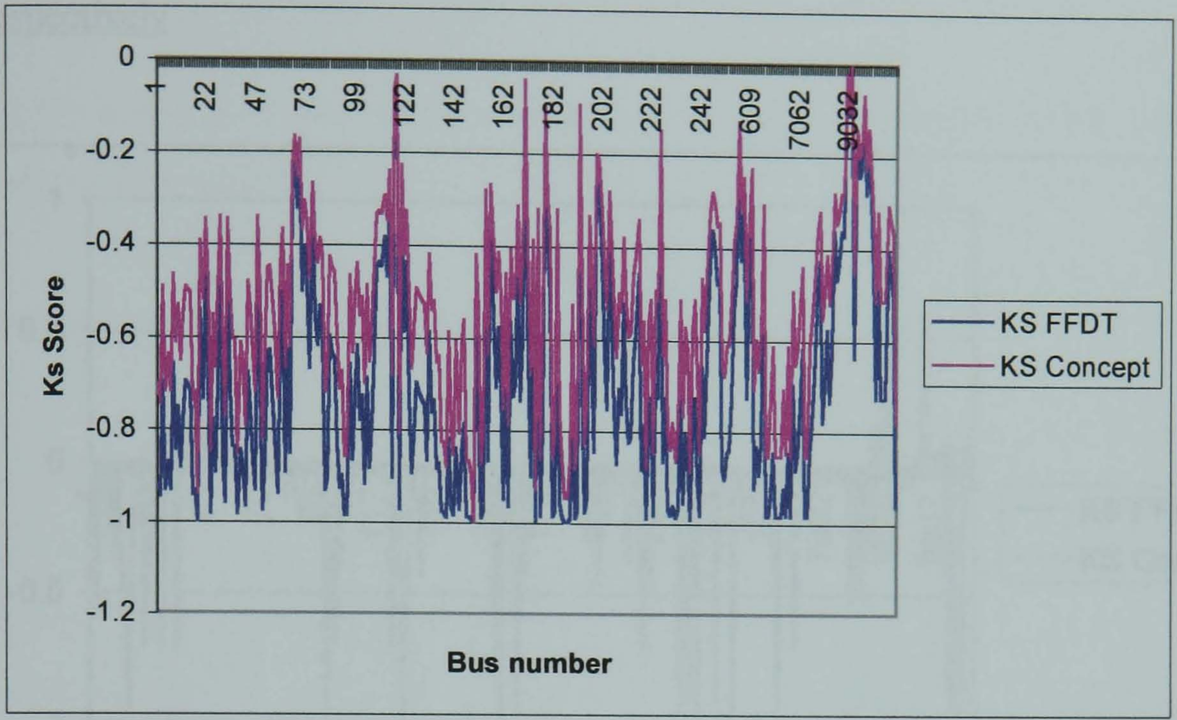


Figure C(2): KS score profile using zero (null) value as a reference point with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Using a predetermined value (1.0) as reference

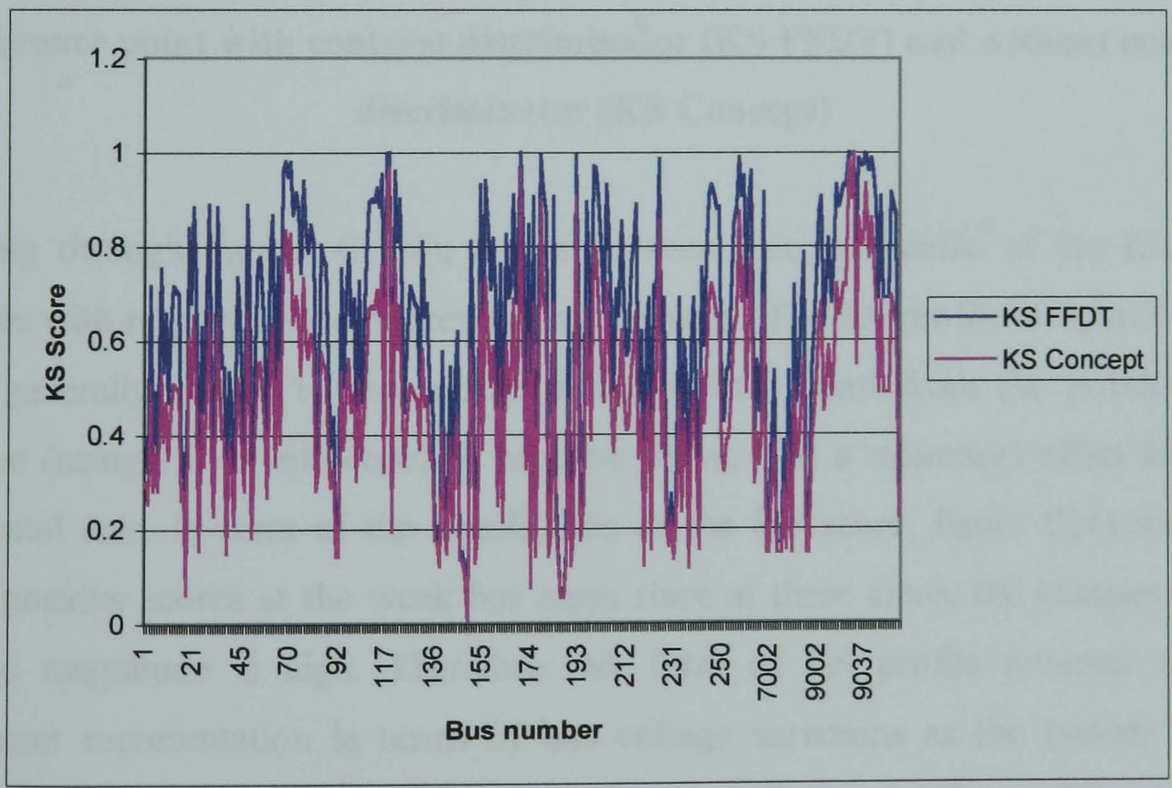


Figure C(3): KS score profile using a predetermined value (1.0) as a reference point with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Using reference each bus voltage magnitudes (load flow results with load factor of 1.0) respectively

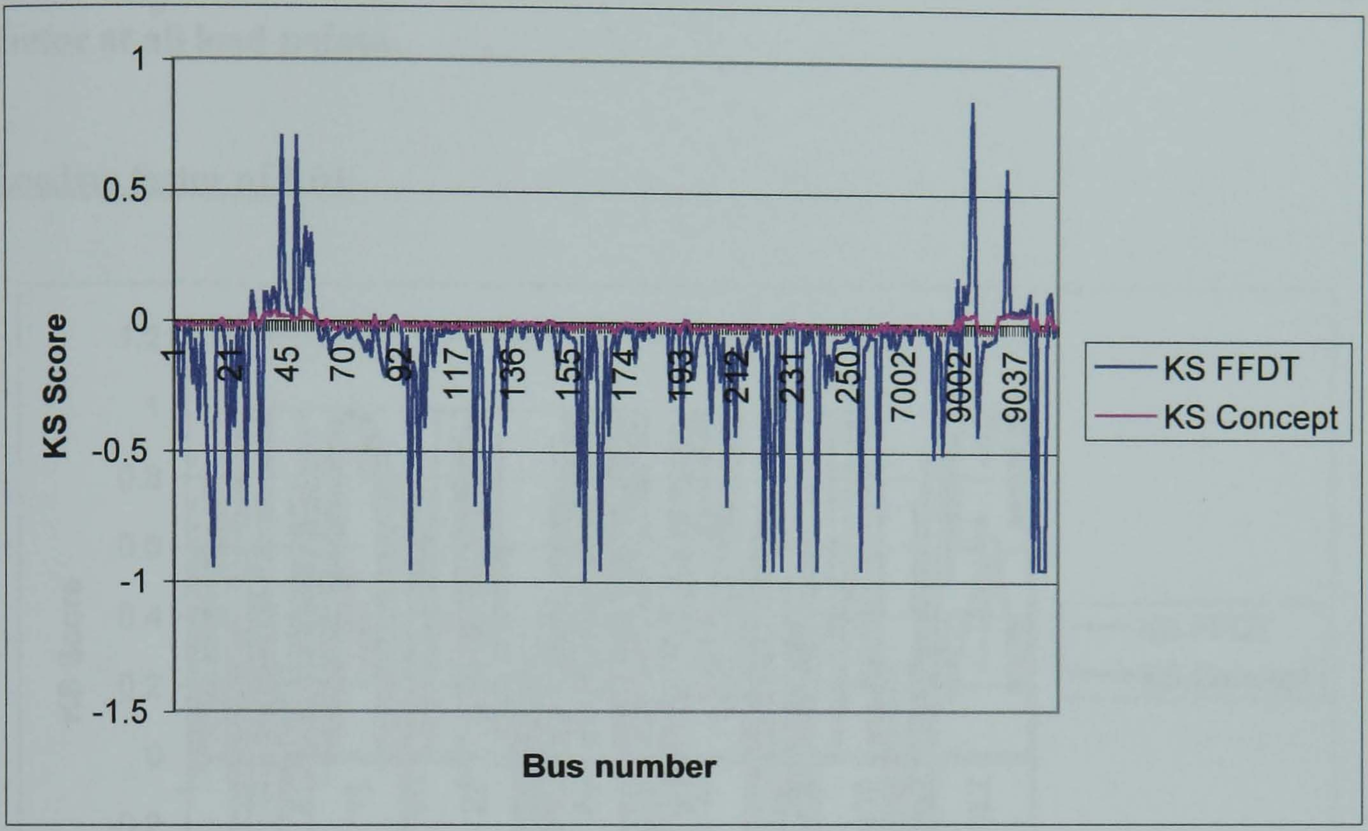


Figure C(4): KS score profile using values for each bus for load factor of 1.0 as a reference point with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Looking through figures C(1-4), it can be seen that the profile of the KS score changes with respect to the referencing values used. The KS profile in figures C(2 & 3) is generally similar to the profile in figure C(1) apart from the profile has a positive (using 1.0 as reference) or negative (using 0 as a reference) offset from the horizontal axis. In term of the significance of the KS score, figure C(4) seems to show positive scores at the weak bus areas since at these areas, the changes of bus voltage magnitude is high. Therefore this form of KS profile presents a more significant representation in terms of bus voltage variations as the system load is increased. However, there is a need to review the effects of different KS references on different loading factors. These are presented in Appendices D, E, F and G.

Appendix D

KS profile using first bus voltage magnitude as reference with varying loading factor at all load points.

Loading factor of 1.01

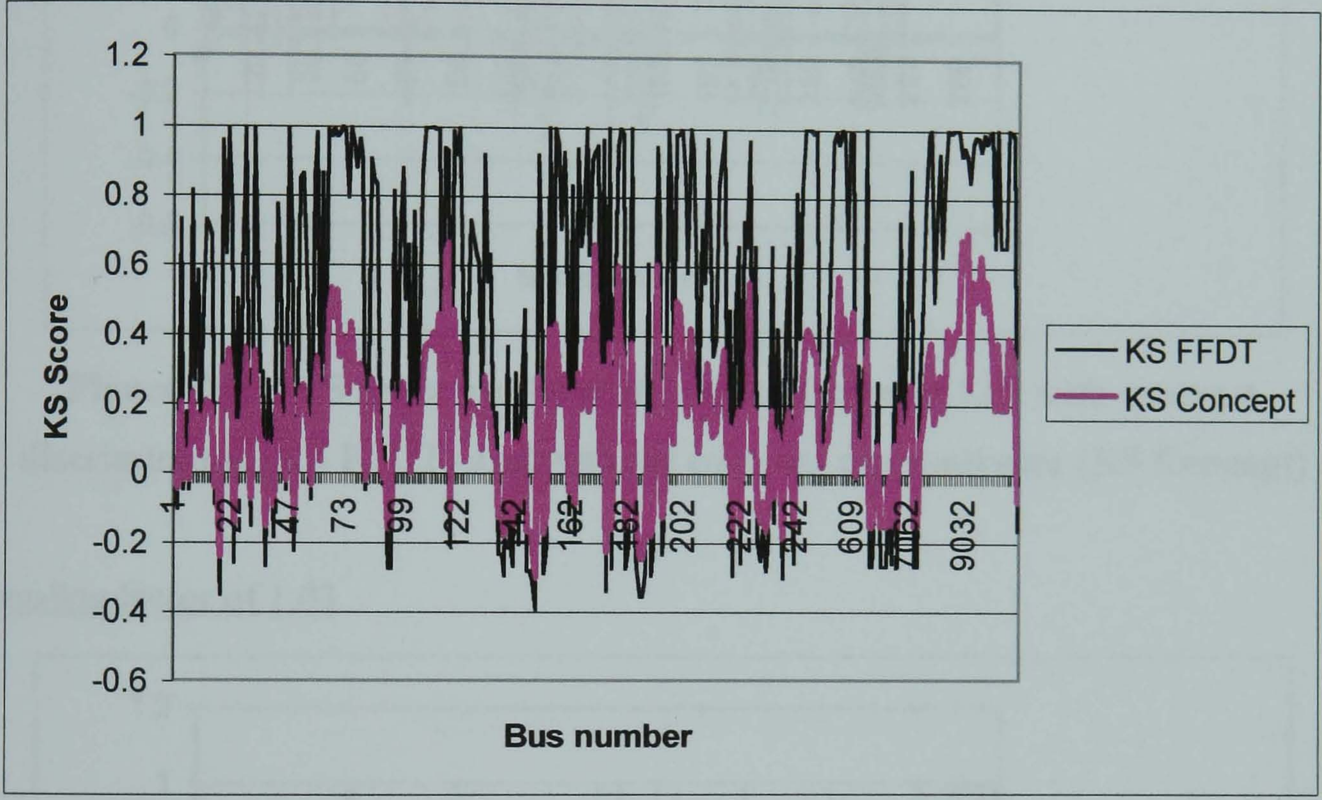


Figure D(1): KS score profile with loading factor of 1.01 with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Loading factor of 1.02

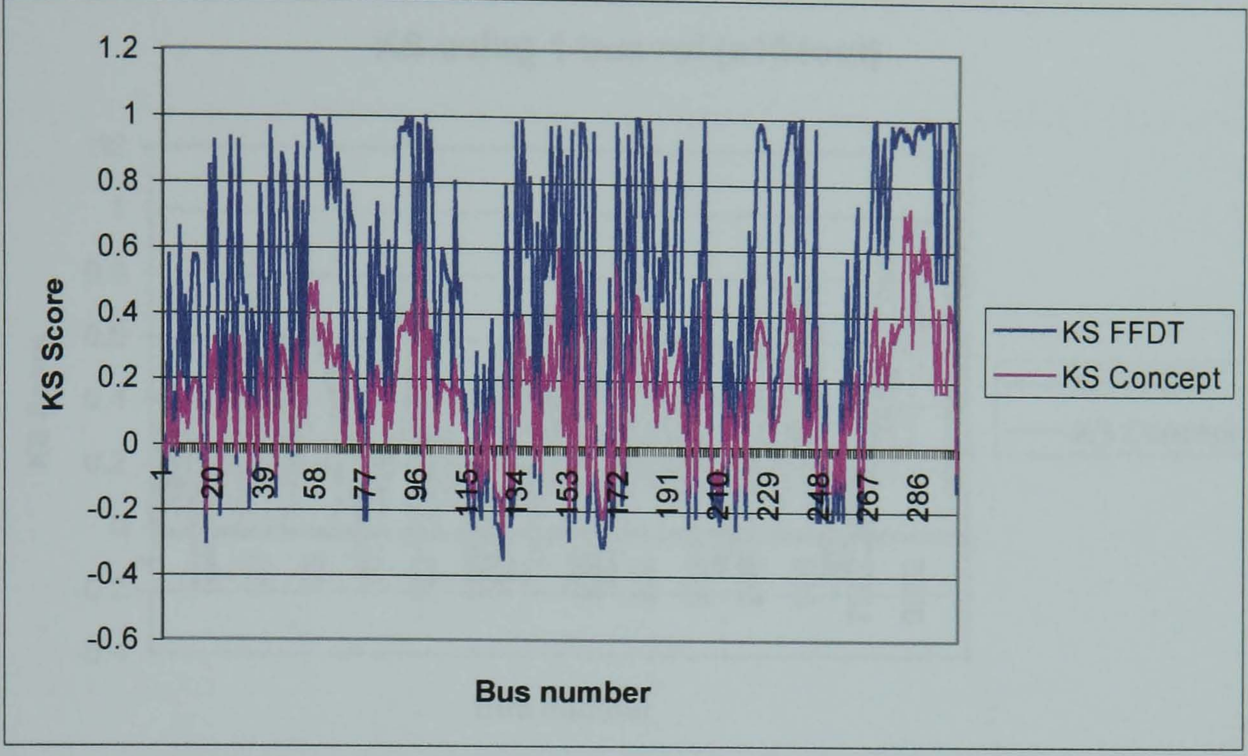


Figure D(2): KS score profile with loading factor of 1.02 with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Loading factor of 1.03

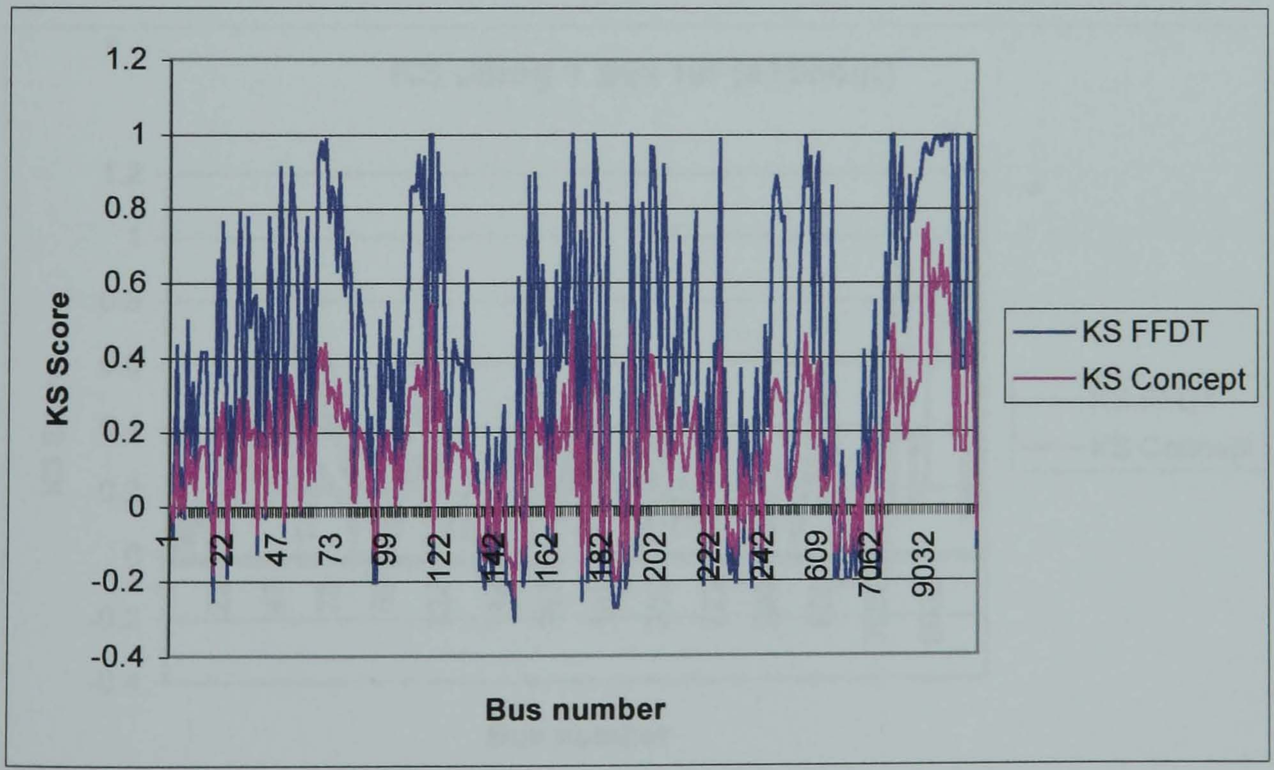


Figure D(3): KS score profile with loading factor of 1.03 with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Loading factor of 1.04

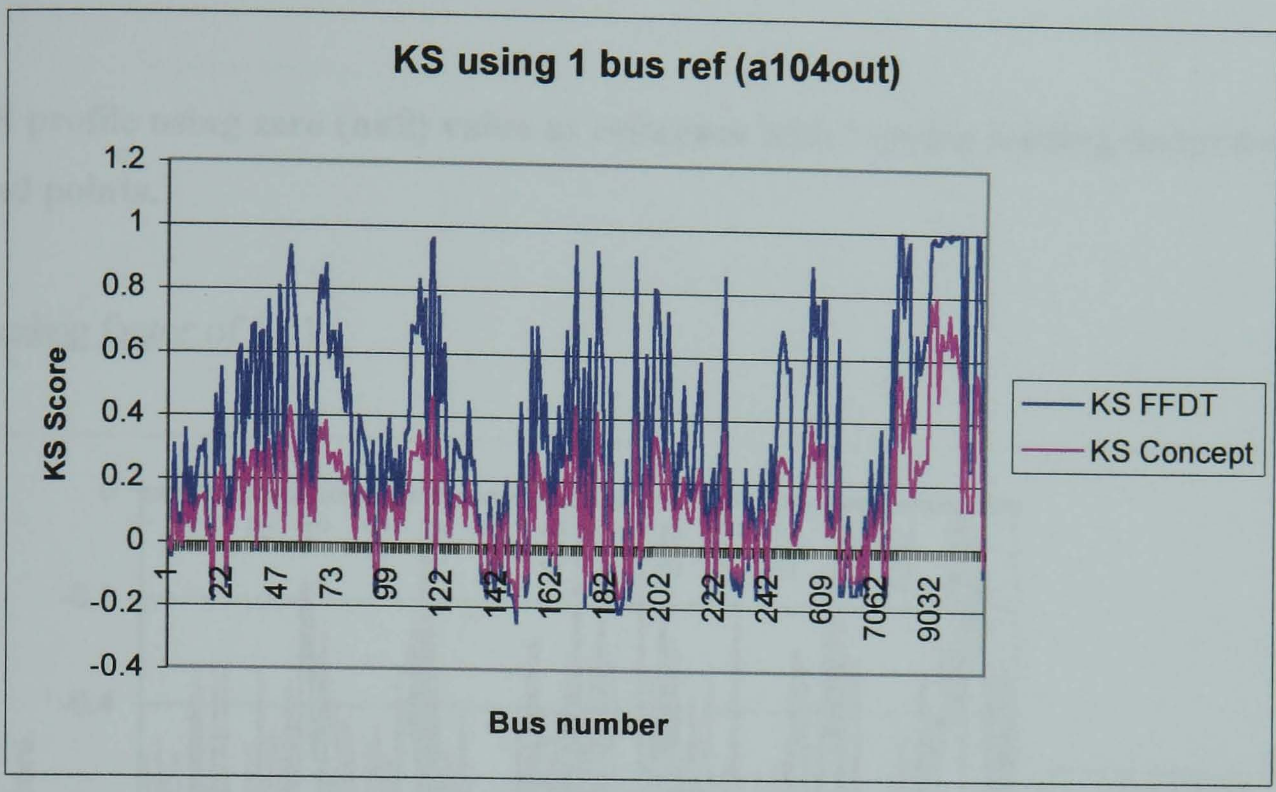


Figure D(4): KS score profile with loading factor of 1.04 with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Loading factor of 1.05

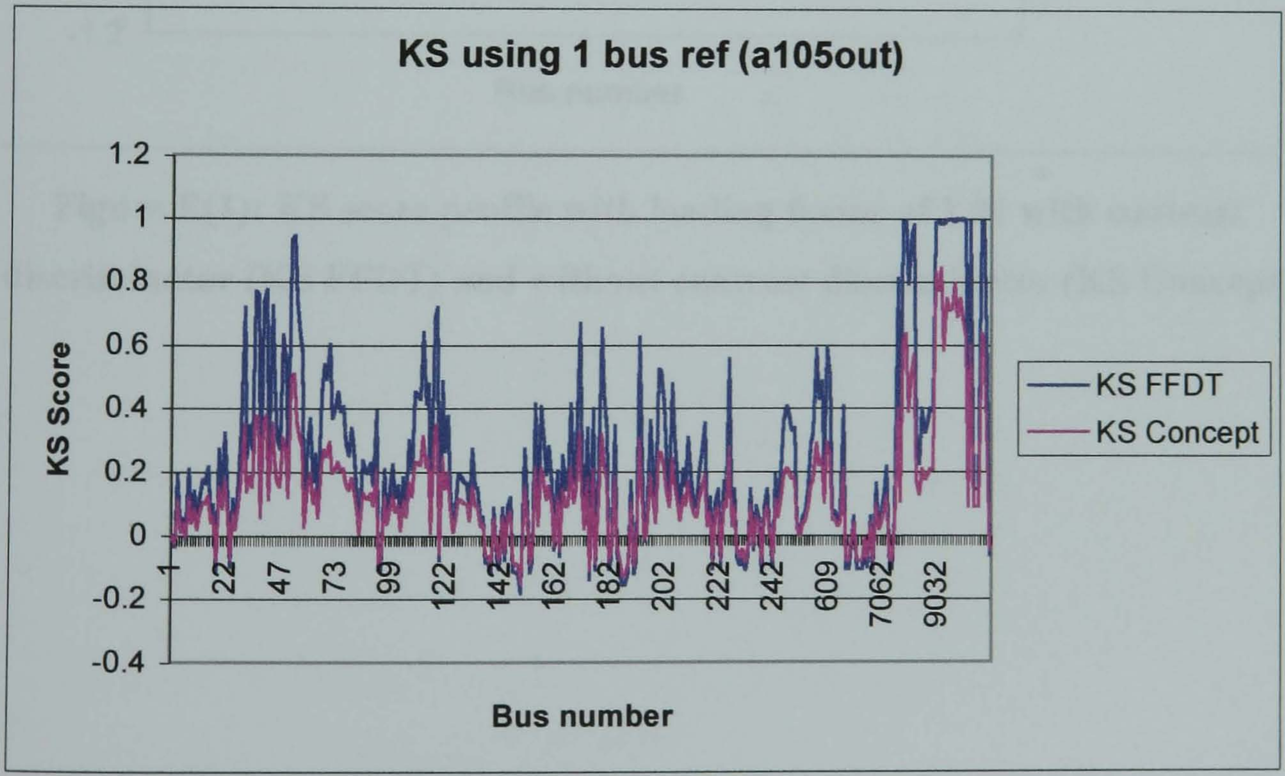


Figure D(5): KS score profile with loading factor of 1.05 with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Appendix E

KS profile using zero (null) value as reference with varying loading factor for all load points.

Loading factor of 1.01

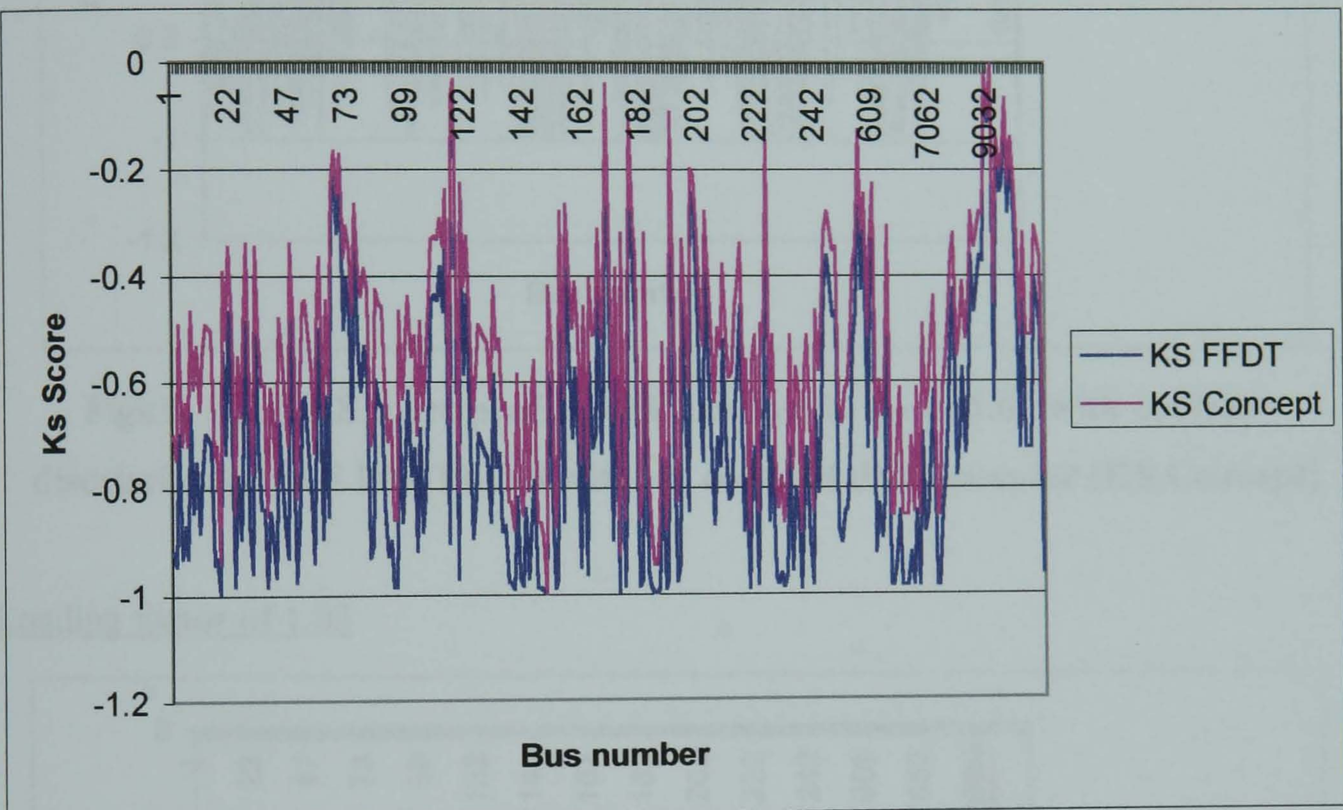


Figure E(1): KS score profile with loading factor of 1.01 with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Loading factor of 1.02

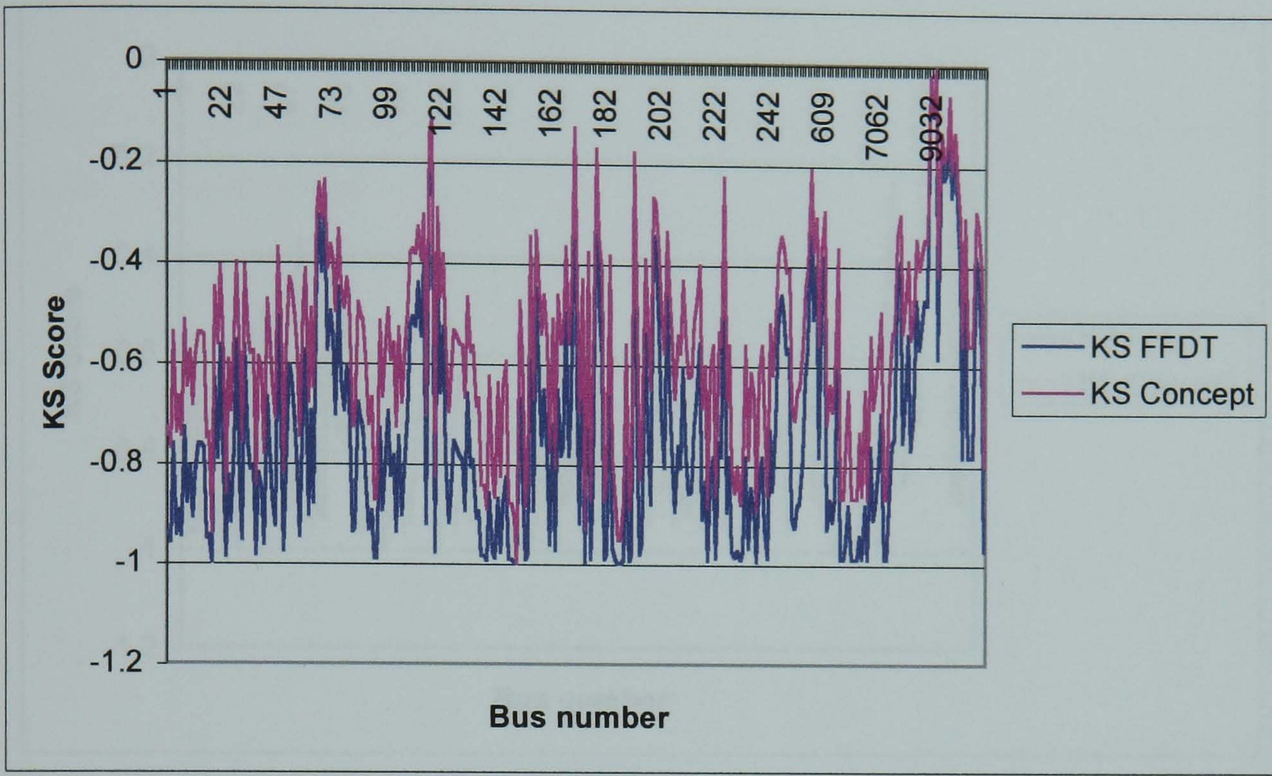


Figure E(2): KS score profile with loading factor of 1.02 with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Loading factor of 1.03

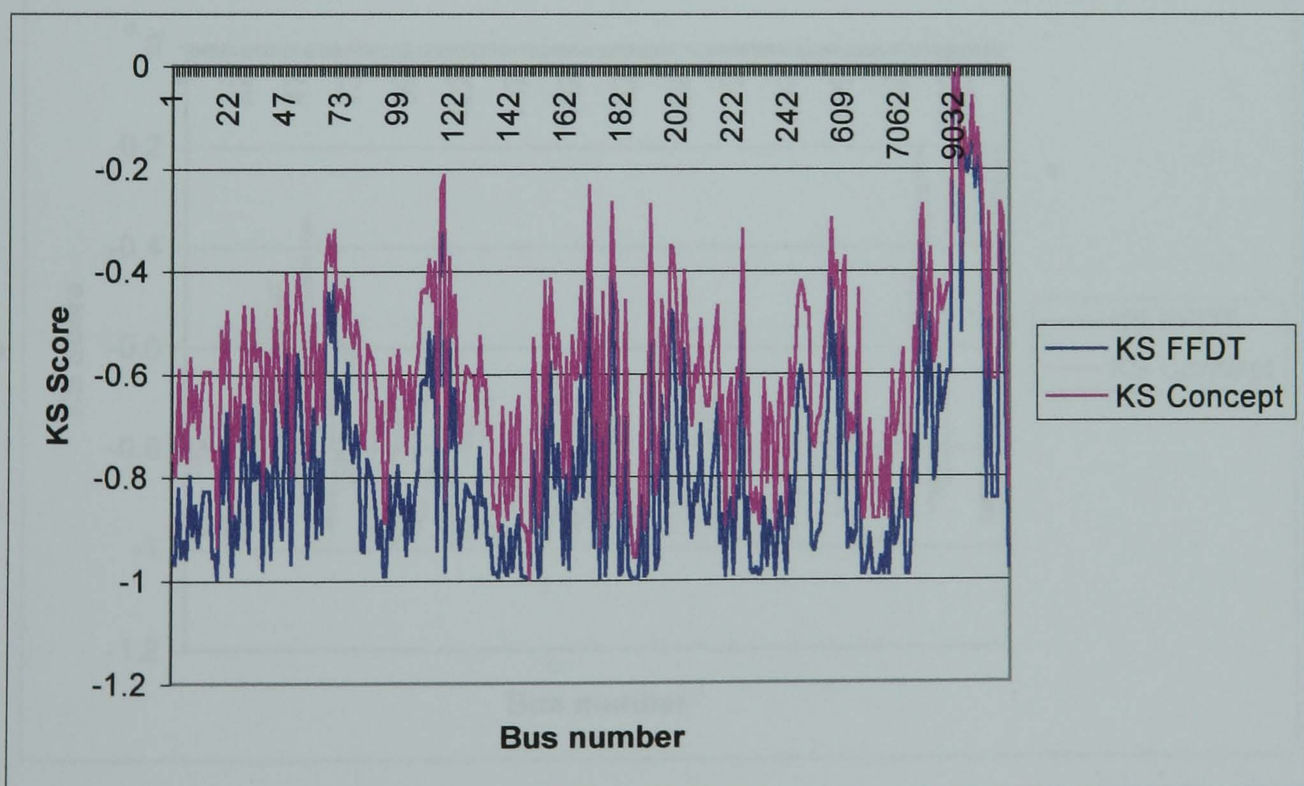


Figure E(3): KS score profile with loading factor of 1.03 with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Loading factor of 1.04

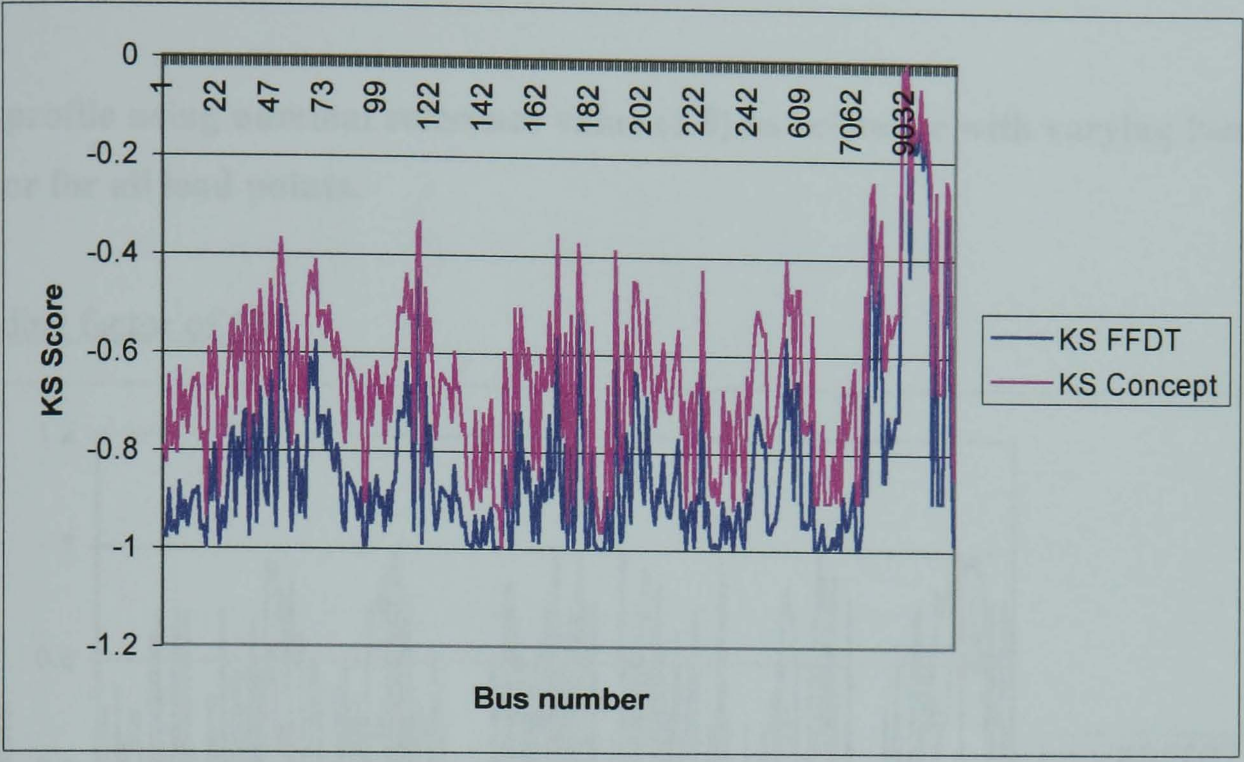


Figure E(4): KS score profile with loading factor of 1.04 with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Loading factor of 1.05

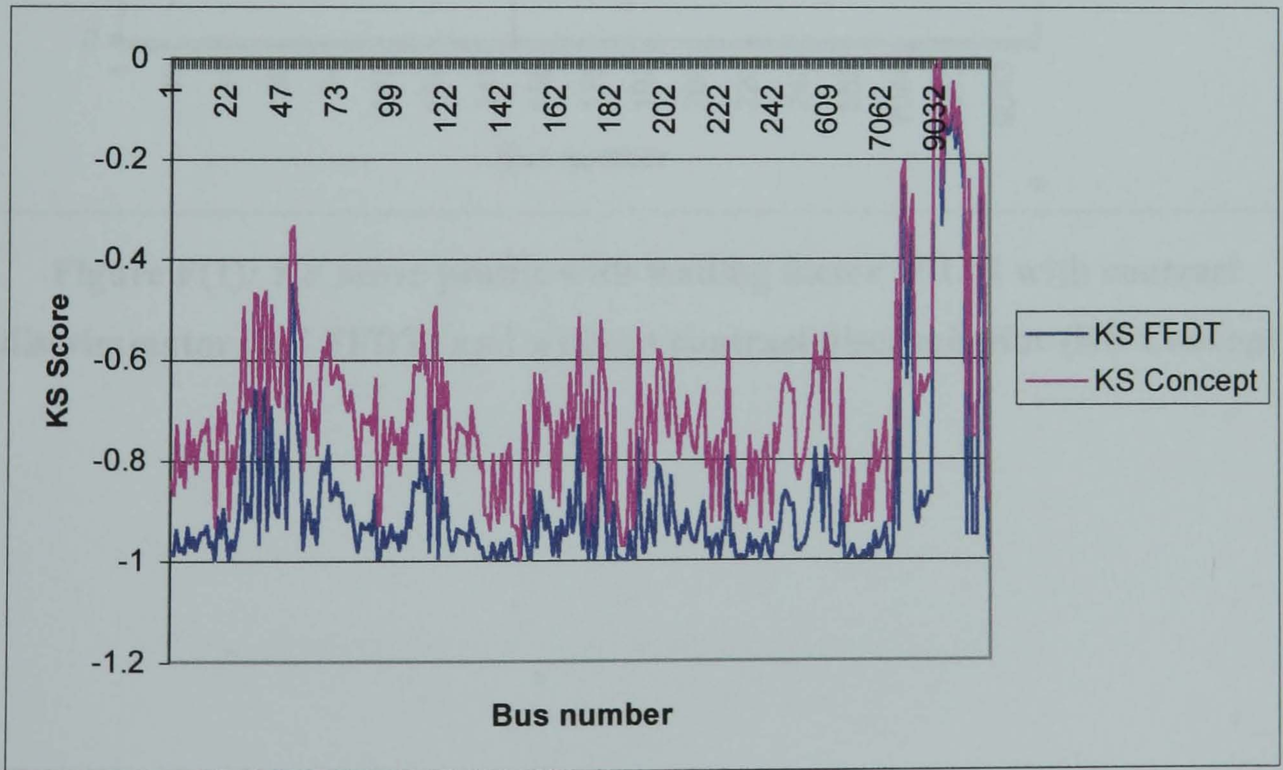


Figure E(5): KS score profile with loading factor of 1.05 with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Appendix F

KS profile using nominal reference value (1.0) as reference with varying loading factor for all load points.

Loading factor of 1.01

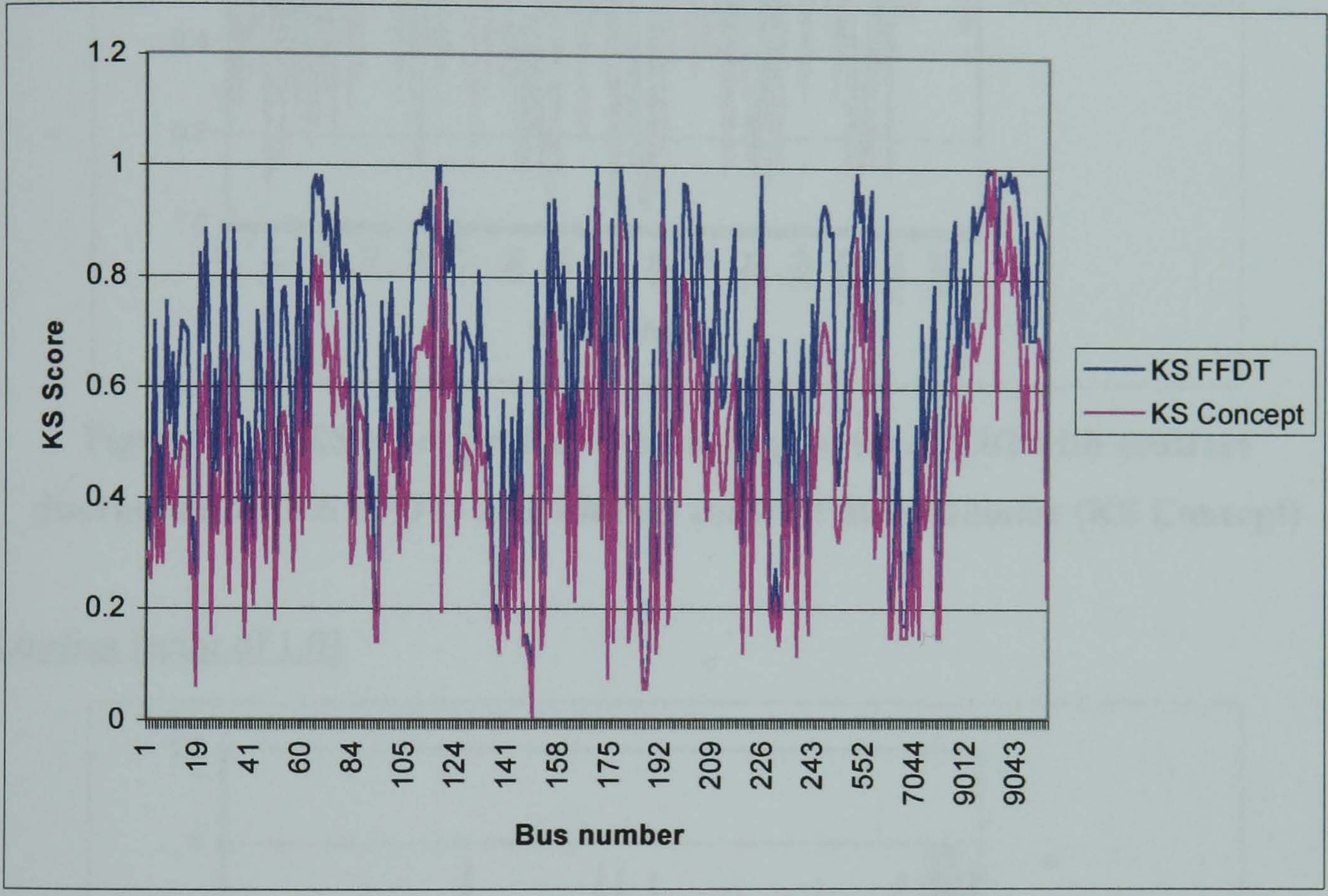


Figure F(1): KS score profile with loading factor of 1.01 with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Loading factor of 1.02

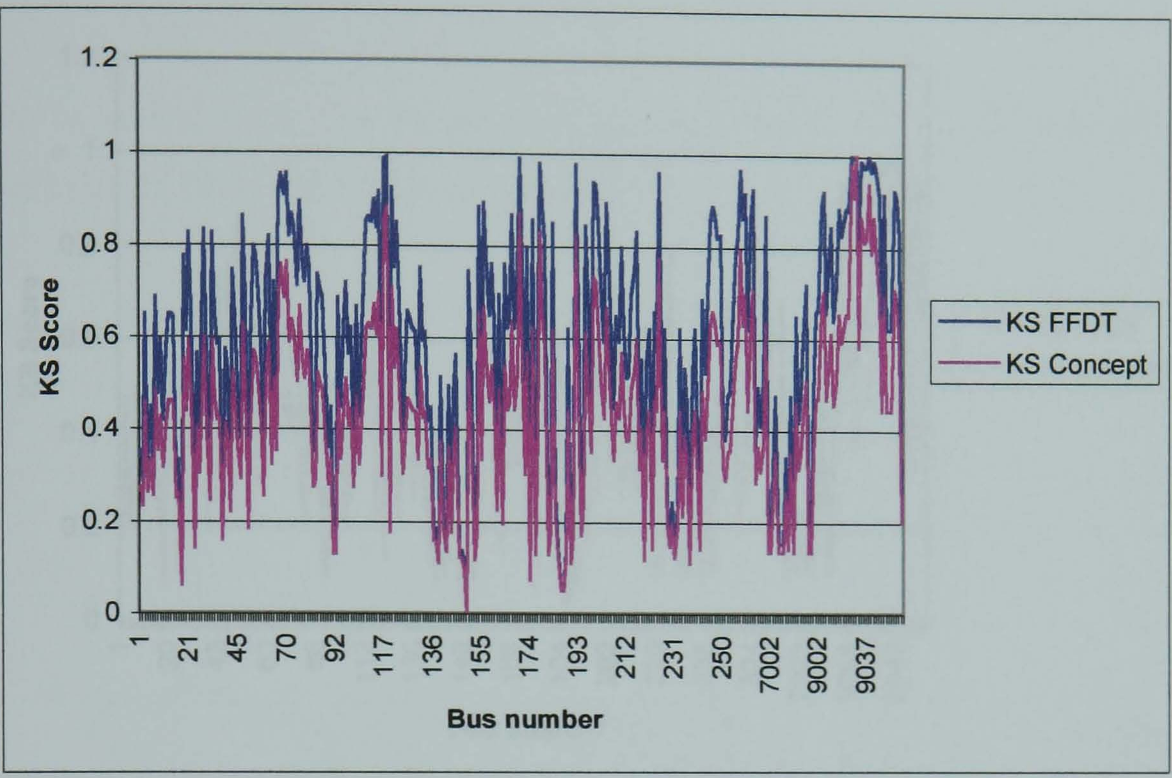


Figure F(2): KS score profile with loading factor of 1.02 with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Loading factor of 1.03

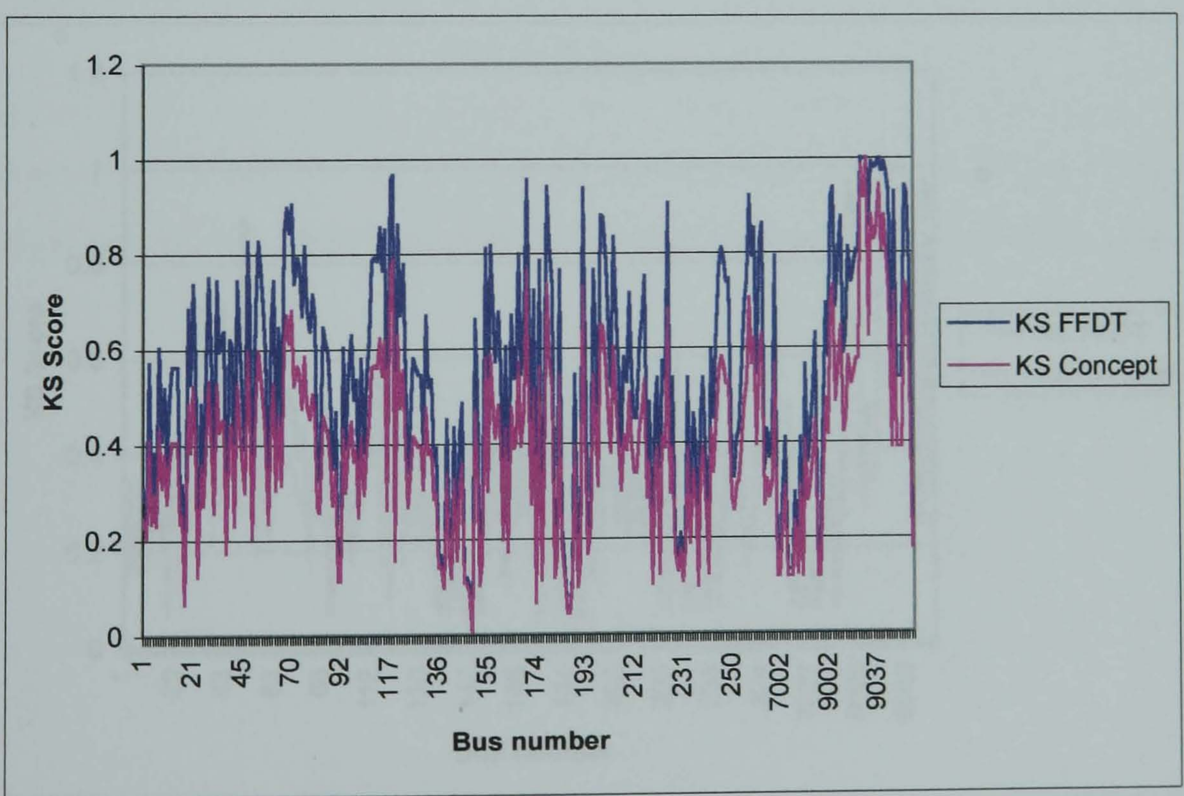


Figure F(3): KS score profile with loading factor of 1.03 with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Loading factor of 1.04

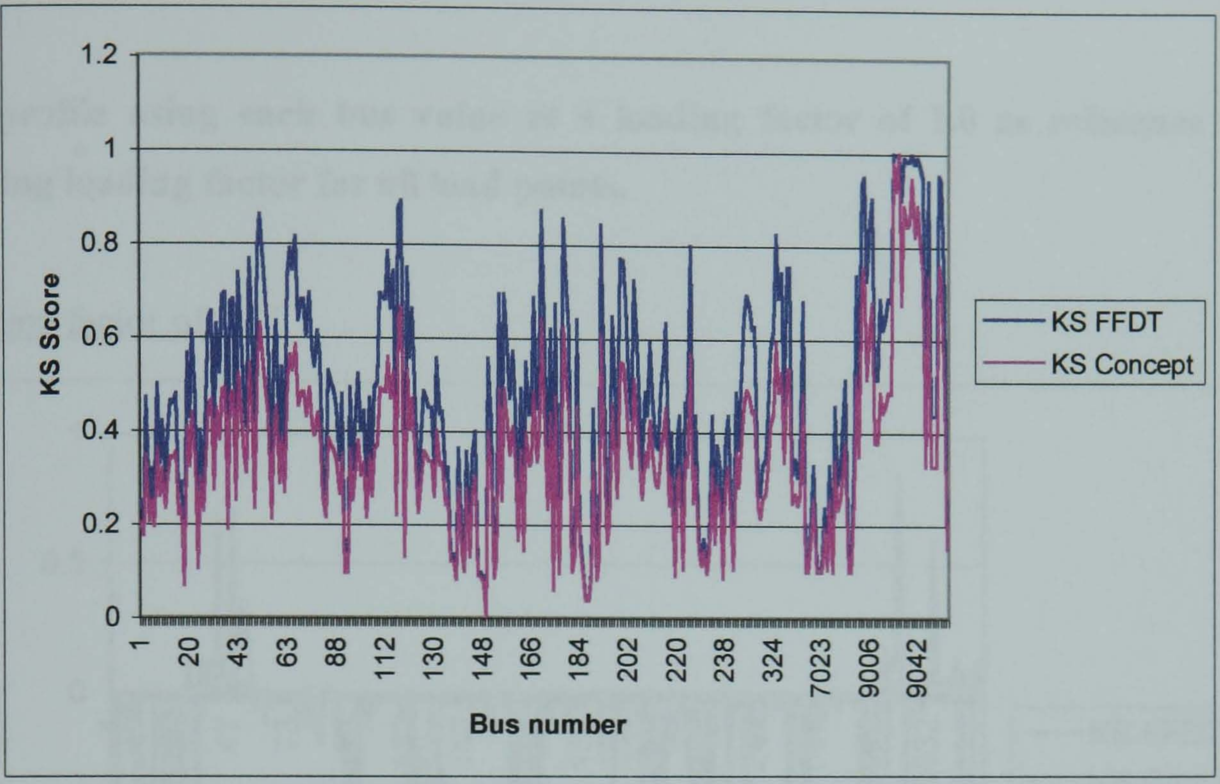


Figure F(4): KS score profile with loading factor of 1.04 with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Loading factor of 1.05

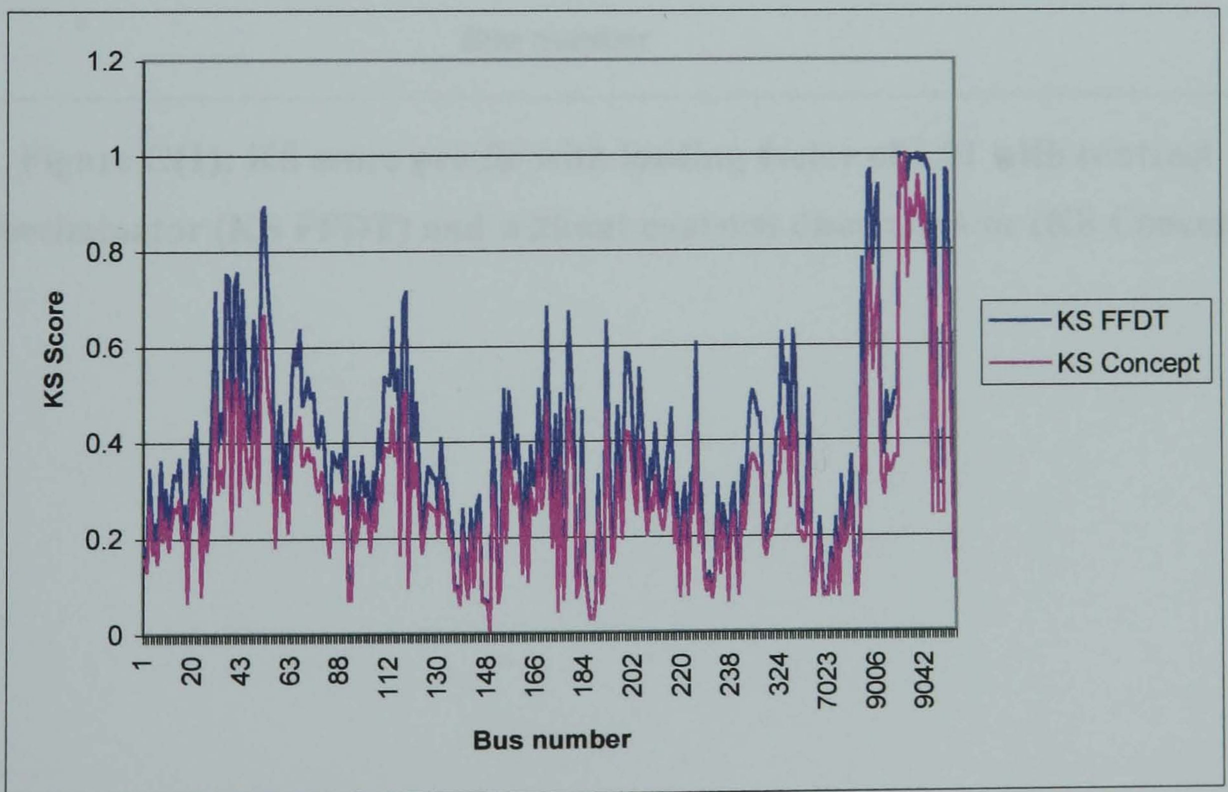


Figure F(5): KS score profile with loading factor of 1.05 with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Appendix G

KS profile using each bus value at a loading factor of 1.0 as reference with varying loading factor for all load points.

Loading factor of 1.01

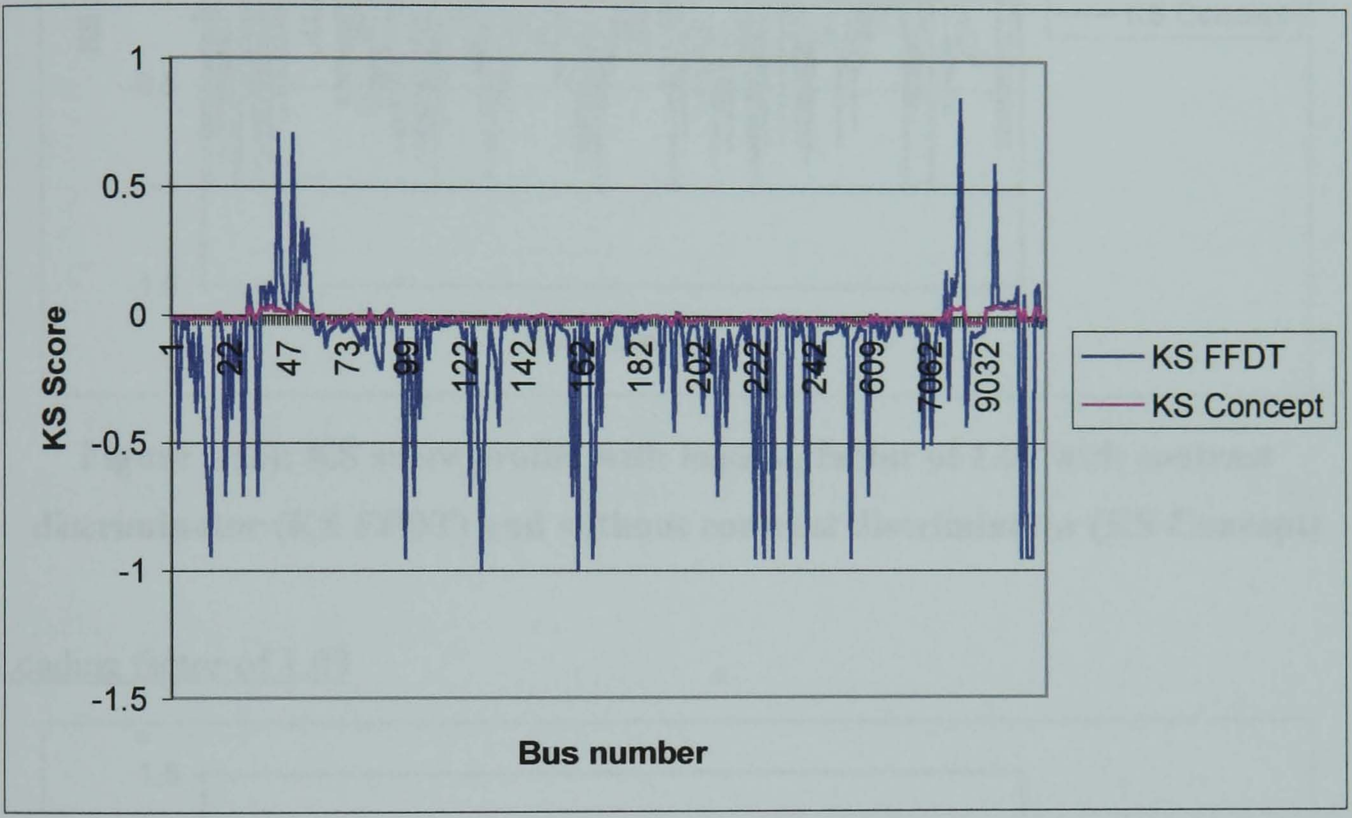


Figure G(1): KS score profile with loading factor of 1.01 with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Loading factor of 1.02

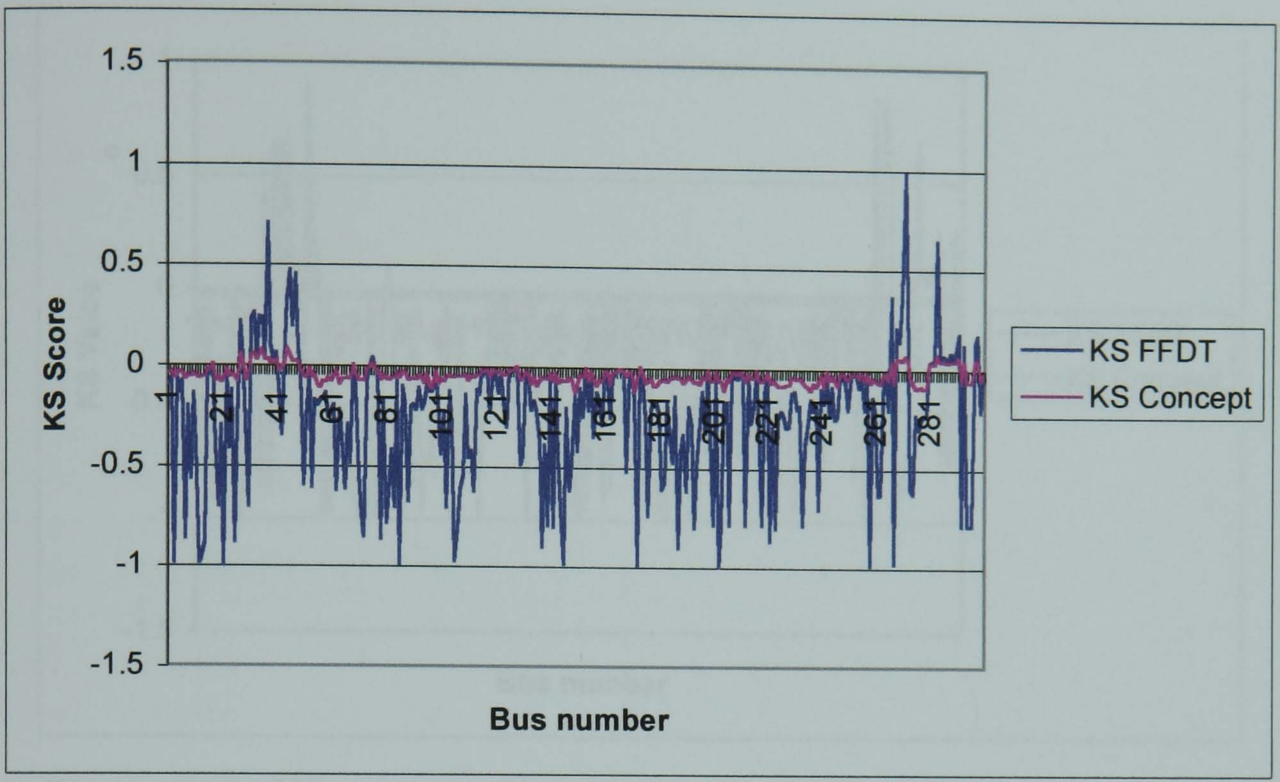


Figure G(2): KS score profile with loading factor of 1.02 with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Loading factor of 1.03

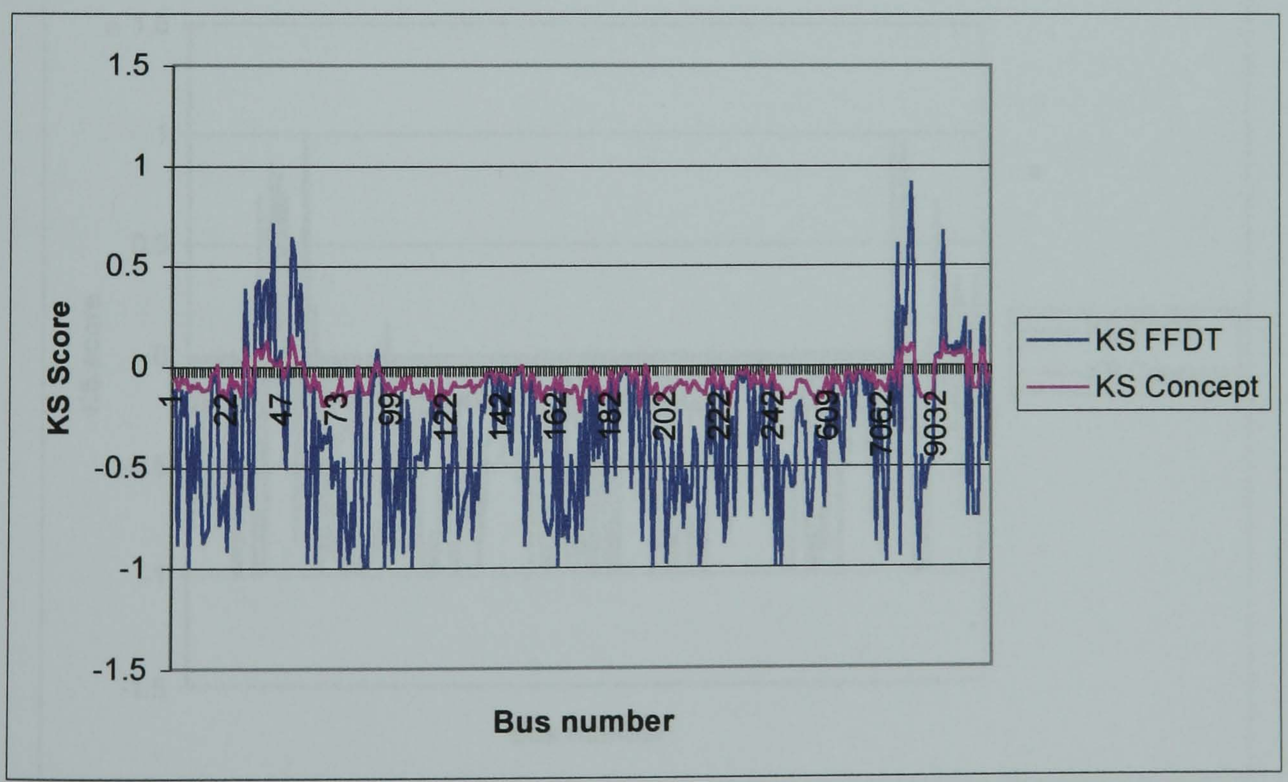


Figure G(3): KS score profile with loading factor of 1.03 with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Loading factor of 1.04

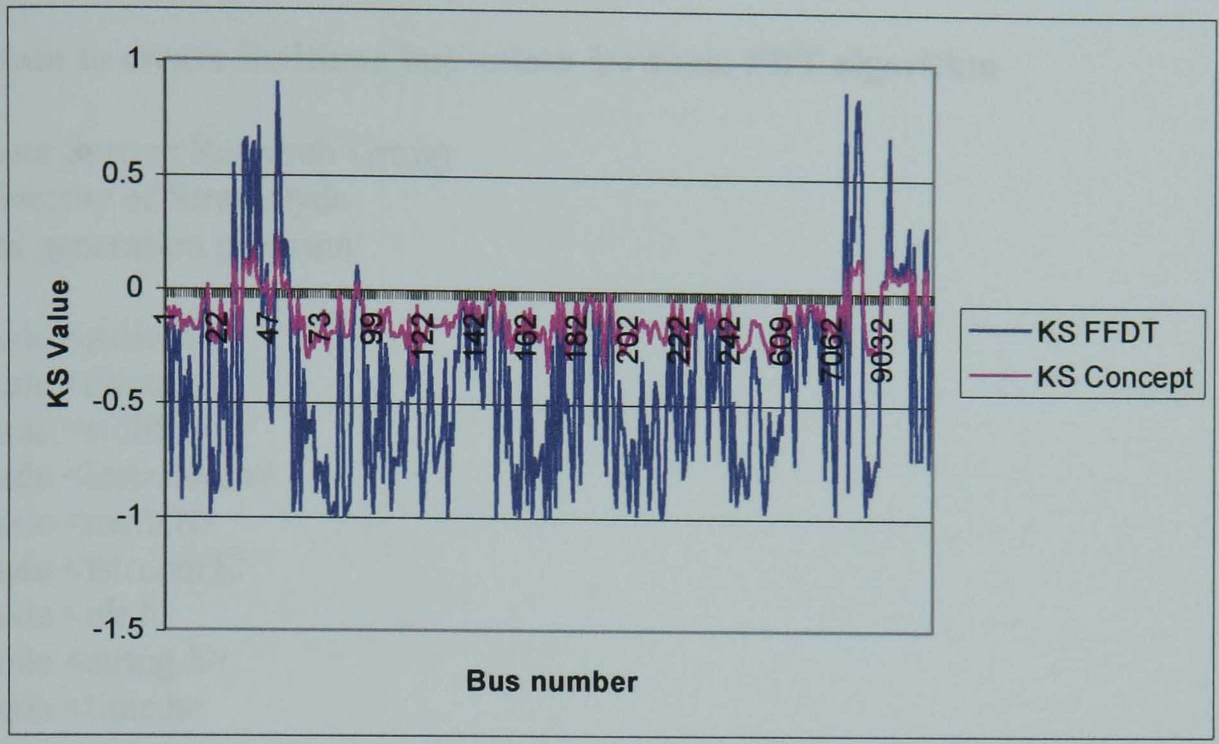


Figure G(4): KS score profile with loading factor of 1.04 with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Loading factor of 1.05

Figure G(5): KS score profile with loading factor of 1.05 with contrast discriminator (KS FFDT) and without contrast discriminator (KS Concept)

Appendix H

Program to create fictitious bus values for basic FDT algorithm

```
// Power System Research Group
// University of Strathclyde
// Data generation program

#include <stdio.h>
#include <direct.h>
#include <stdlib.h>
#include <iostream.h>
#include <math.h>
#include <fstream.h>
#include <dir.h>
#include <string.h>
#include <time.h>

void main ()
{

int count, limit;
float ans, result;

ofstream output;
output.open ("c:\\windows\\profiles\\agie\\desktop\\testin.dat");

limit = 1000;
cout << "Data building....." << endl;
randomize();

// First stable range..
for (count = 0; count <= limit; count=count + 1)
        {
                ans = rand () % 105;
                result = ans/100;
                if (result >= 0.95)
                output << result << endl;
        }
```

```
// Unstable Range - (lower end)
for (count = 0; count <= limit; count=count + 1)
        {
                ans = rand () % 94;
                result = ans/100;
                if (result >= 0.85)
                output << result << endl;
        }

// Second Stable Range...
for (count = 0; count <= limit; count=count + 1)
        {
                ans = rand () % 105;
                result = ans/100;
                if (result >= 0.95)
                output << result << endl;
        }

// Unstable Range - (upper end)
for (count = 0; count <= limit; count=count + 1)
        {
                ans = rand () % 116;
                result = ans/100;
                if (result >= 1.06)
                output << result << endl;
        }
output.close ();

}
```

Appendix I

Bus voltage variation analysis with different load strategies

Overview

System : Standard IEEE 300 bus system

Power flow software used : MatPower operating under MatLab.

Type of test made : (a) Load increment on all load buses

(b) Load increment on selected heavily loaded buses

(c) Load increment on selected lightly loaded buses.

(d) Load increment on 1 bus only

Approach to analysis:

(a) Run power flow program using different load increase on all or selected load busses.

(b) Use output results for analysis

Results:

Looking at the bus voltage levels in general, there is a significant characteristic where in an increase of system load, certain groups of busses are shown to exhibit significant changes. This can be seen in all 3 sets of test results shown in figures I(1-4). In figure I(1-3), the loading factor increase is small when compared with other loading strategies. This shows that the system loading capability is dependent upon the loading strategies applied to its load centres. For figure I(4) there is no loading factor key in the graph. This is due to its very high loading factors, but the main concern is not the amount of loading factor required but the similarity of system behaviour as the system load is increased. Looking into all the figures shown, it can be seen that the system exhibits a similar behaviour in terms of bus voltage magnitudes on different loading strategies. Therefore it is safe to conclude that bus voltage magnitude is suitable as a test attribute for the FDT as mentioned in chapter 5.

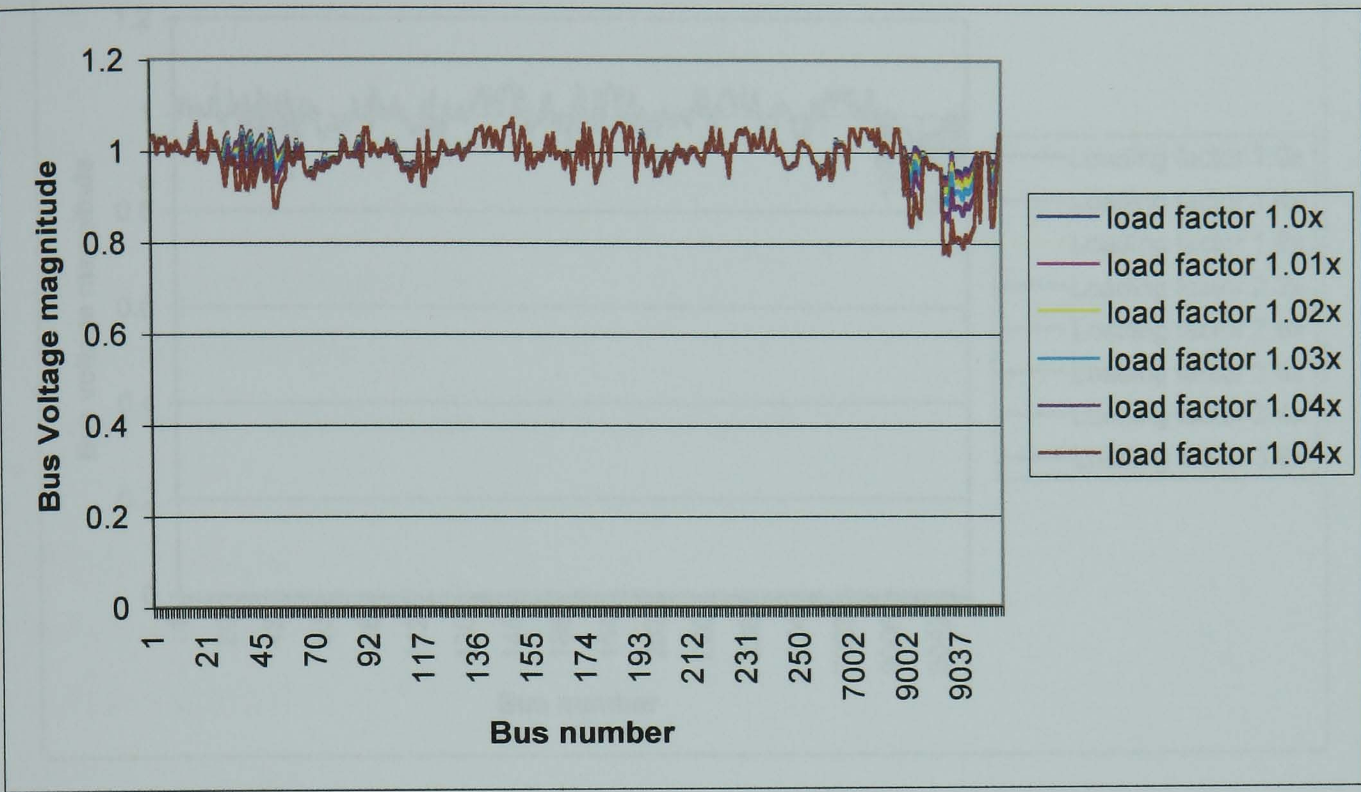


Figure I(1): Load increments at all load centres

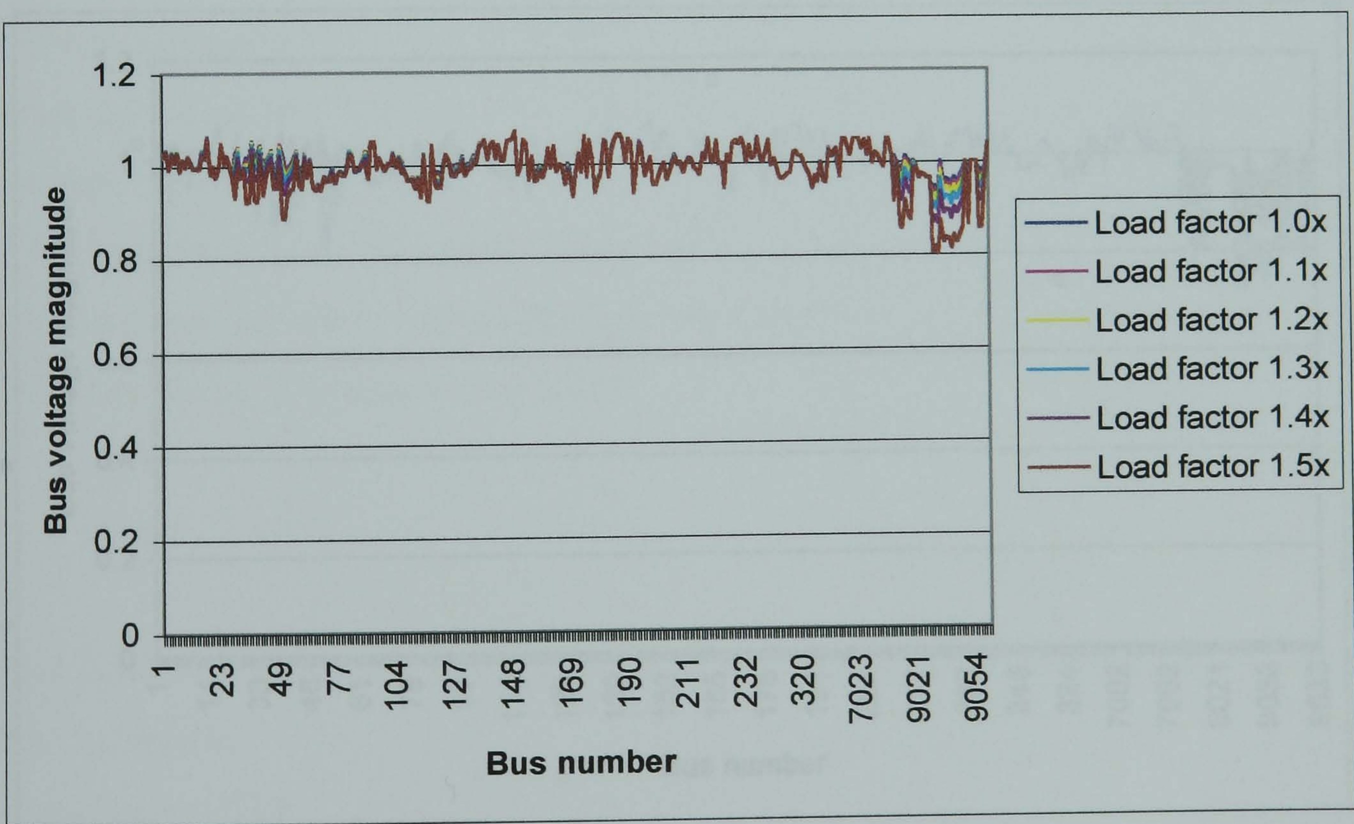


Figure I(2): Load increments on selected heavily loaded buses

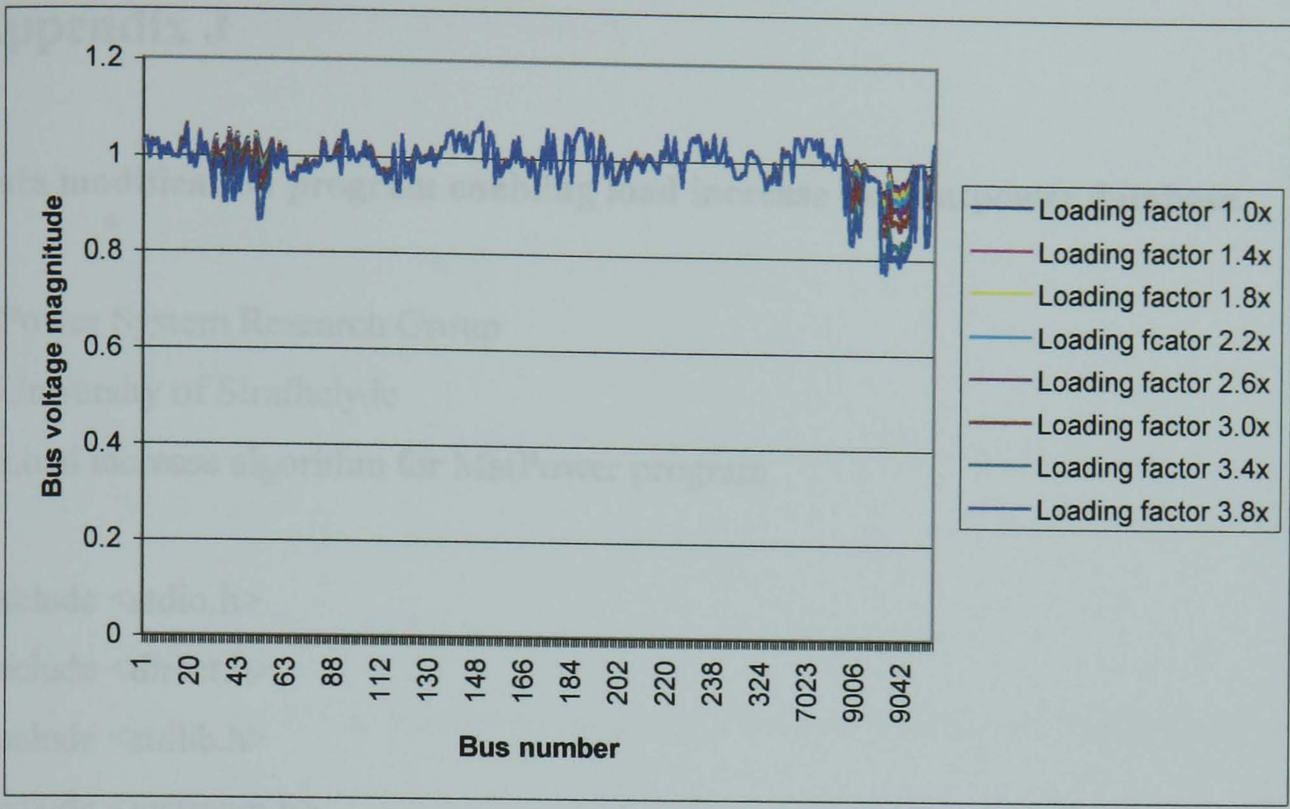


Figure I(3): Load increments on selected lightly loaded buses

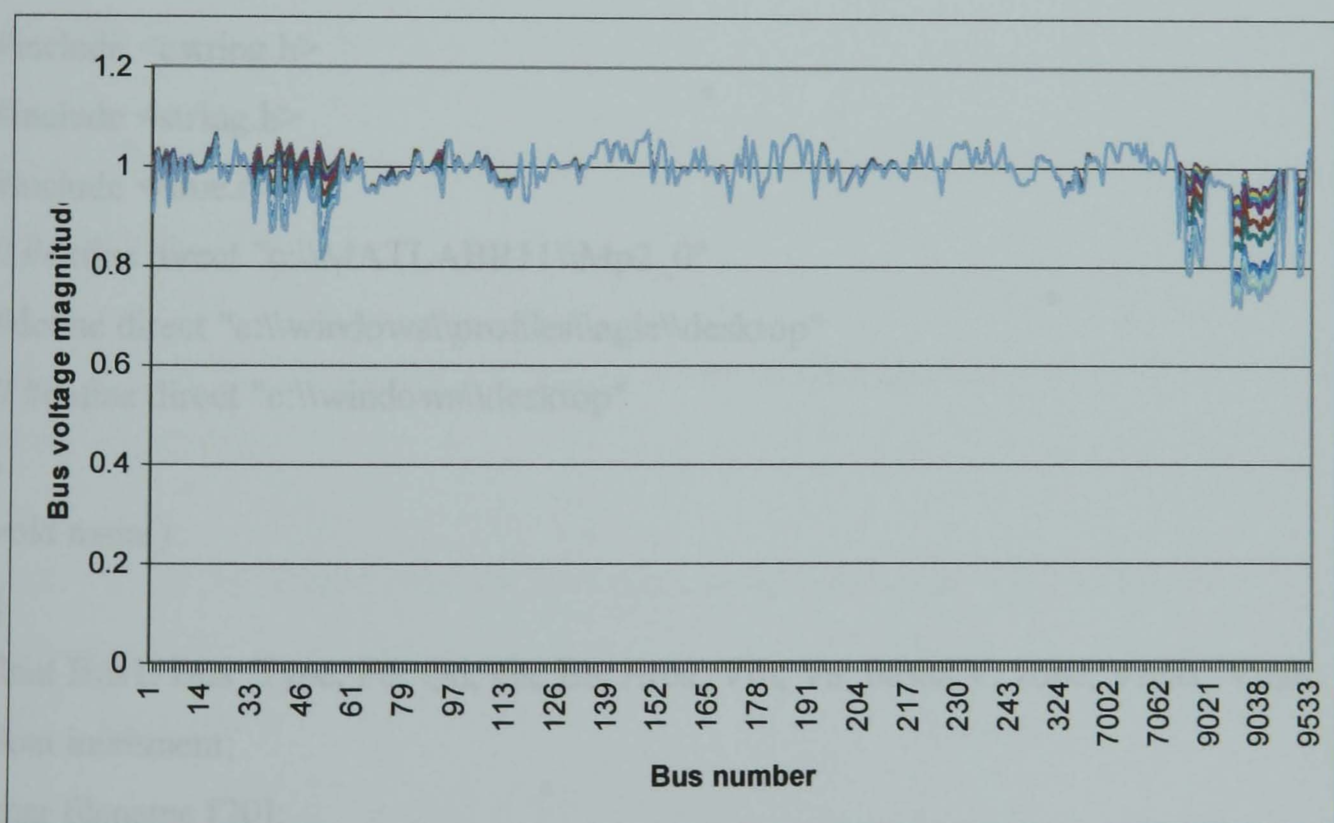


Figure I(4) Load increments on 1 load bus (bus1)

Appendix J

Data modification program enabling load increase for Matpower database.

```
// Power System Research Group
// University of Strathclyde
// Load increase algorithm for MatPower program

#include <stdio.h>
#include <direct.h>
#include <stdlib.h>
#include <iostream.h>
#include <math.h>
#include <fstream.h>
#include <dir.h>
#include <cstring.h>
#include <string.h>
#include <time.h>
// #define direct "c:\\MATLABR11\\Mp2_0"
#define direct "c:\\windows\\profiles\\agie\\desktop"
// #define direct "c:\\windows\\desktop"

void main()
{
float Bus1, Bus_Type, Pd, Qd, Gs, Bs, Area, Vm, Va, baseKV, zone, Vmax, Vmin;
float increment;
char filename [20];
char outputfile [20];
string header;
int loop, busnumber;
```

```
// Dos Branch initialisation determination

        _chdrive (3);               // change active drive to drive c:
        chdir (direct);

// Variable initialisation
Bus1 = Bus_Type = Pd = Qd = Gs = Bs = Area = Vm = Va = baseKV = zone =
Vmax = Vmin =0;

// Start conversion
cout << "Welcome to the MatPower Load Increamental tool for the MatPower
Program " << endl;
cout << "by Haji Izham Haji Zainal Abidin from the Power System Research Group"
<< endl;
cout << "Please state the required MatPower file (filename.m) : ";
gets(filename);

cout << endl << "Please state the number of bus in this system : ";
cin >> busnumber;

cout << endl << "Please give the output file name (filename.m) : ";
gets (outputfile);

cout << endl << "Please state the increment value (new value = old value x
increment) : ";
cin >> increment;

cout << endl << "Reading data file........." << endl;
ifstream datin;
datin.open(filename);
ofstream out;
```

```
out.open(outputfile);
cout << "Transfering header files....... " << endl;
loop = 1;
while (loop <= 7)
{
// Reading the header files and transfering to temp files
getline (datin, header);
out <<  header << endl;

if (loop == 4)
        out << endl;
else if (loop == 5)
        out << endl;

loop = loop + 1;

}

// Reading Bus Data file
cout << "Reading bus data .......... " << endl;
loop = 0;
while (loop < busnumber)
{
        datin >> Bus1;
        datin >> ws;
        datin >> Bus_Type;
        datin >> ws;
        datin >> Pd;
        datin >> ws;
        datin >> Qd;
        datin >> ws;
```

```
        datin >> Gs;
        datin >> ws;
        datin >> Bs;
        datin >> ws;
        datin >> Area;
        datin >> ws;
        datin >> Vm;
        datin >> ws;
        datin >> Va;
        datin >> ws;
        datin >> baseKV;
        datin >> ws;
        datin >> zone;
        datin >> ws;
        datin >> Vmax;
        datin >> ws;
        datin >> Vmin;
        datin >> ws;

        loop = loop + 1;                        // increasing loop = loop + 1

// Checking for Load Bus
        if (Bus_Type == 1)
        {
                Pd = Pd * increment;                    // increasing Pd
                Qd = Qd * increment;                    // increasing Qd
                goto output;
        }
```

```
        else
        {
                goto output;
        }

// Transfering into output file.

output:         out << Bus1;
                out << "    ";
                out << Bus_Type;
                out << "    ";
                out << Pd;
                out << "    ";
                out << Qd;
                out << "    ";
                out << Gs;
                out << "    ";
                out << Bs;
                out << "    ";
                out << Area;
                out << "    ";
                out << Vm;
                out << "    ";
                out << Va;
                out << "    ";
                out << baseKV;
                out << "    ";
                out << zone;
                out << "    ";
                out << Vmax;
                out << "    ";
```

```
        out << Vmin;
        out << "    ";
        out << endl;
}

while (!datin.eof())
{
getline (datin, header);              // Transfering the rest to the output file
out << header << endl;
}

datin.close ();
out.close ();

cout << "Thank you.... ";
}
```

Appendix K

Attribute limit modification tests

Version of FDT	FDT ver 2
KS technique	Using KS formulation based upon the FFDT concept using the first bus voltage value as reference.
Partitioning technique	Using maximum KS score location as partitioning limit.
Stop split rule	The rules for the stop split stage are based upon minimum of the test node. If the population of the test node is within the minimum population range, the node will stop partitioning and label the test node as a leaf node. The population variations for this test are 60, 100, 120, 150, 200, 250 and 300.
Pruning technique	Manual pruning technique
Test system used	IEEE 300 bus system using all load bus at loading factor of 1.02x
Computer used	Intel PIII 1GHz

Results

Table K(1): Results of varying FDT and its effects on the FDT performance

Attribute limit	Total nodes generated by FDT	Total leaf nodes generated by FDT	Execution time (s)
60	41	22	4
100	31	17	4
120	23	13	3
150	23	13	3
200	11	7	1
250	19	11	3
300	3	2	Less than 1

Table K(1) shows that at a lower attribute limit value, the FDT will partition 41 nodes in total with 22 leaf nodes. In short, the FDT partitions 300 bus voltage values into 22 sets where some may contain only a few attributes. In terms of the FDT performance, what needed are fewer leaf nodes but better partitioning accuracy, i.e. the nodes should contain either majority stable buses or majority unstable buses. In terms of execution time, the more nodes it needed to partition, the more execution time required. This can be seen at the execution time column shown. As the attribute limit is increased, fewer nodes are generated hence less execution time. But if the attribute limit is increased too much, say 250 and 300 as an attribute limit, the significance of the leaf nodes produced will be lost but with better execution time. Comparing the FDT performance and taking into account the execution time, an attribute limit of 150 is chosen.

Appendix L

Investigating the effect of setting different new limit element into the stop split stage

Version of FDT	FDT ver 3.0
KS technique	Using KS formulation based upon the FFDT concept using bus voltage values (of the IEEE 300 bus test system at a loading factor of 1.0x at all load buses) as reference
Partitioning technique	Comparing the location of the maximum and minimum KS score. • If theses scores are located at both ends of the test nodes, partitioning will stop. • If the maximum KS score is on the extreme ends of the test node, and the location of the minimum KS score is not located at the extreme ends of the test node, the partitioning limit will be based upon the location of the minimum KS score. • If the maximum KS score is not at extreme ends, partitioning will be based upon the maximum KS location as a partitioning limit. The location of the KS scores are considered to be on the extreme ends if is located within the range of 10 attributes from each end.

Stop split rule	The rules for the stop split stage are based upon the purity of the node and the minimum population (minimum population value is 50 attributes) of the test node. The summary of the rules are: • If the test node consist of all strong buses (bus value should be more than the minimum value limit) stop splitting the node and label is as a leaf node. • If the test node consist of a combination of strong buses and weak buses. Check the total population of the test node. If it exceeds the minimum population limit continue partitioning. If not, stop partitioning and label the test node as a leaf node. This test investigates 2 types of minimum value limit that are 0.8 and 0.9.
Pruning technique	No pruning technique is used
Test system used	IEEE300 bus test system with loading factor of 1.05x on all load buses.
Computer used	Intel PIII 1GHz

Results

Table L(1): Results of varying element limit in the stop split stage and its effects on the FDT performance

Stop split attribute limit	Total nodes generated by FDT	Total leaf nodes generated by FDT	Execution time (s)
0.8	7	4	1
0.9	17	9	1

The partitioning capabilities are as shown:

Table L(2): A table showing attribute range for leaf nodes created by the FDT with attribute level limit of 0.8

Leaf nodes (Node No.) of FDT results for system loading of 1.05x at all load buses with attribute level limit of 0.8	Attribute range in leaf node (Bus No.)
2	1 – 247
4	248 – 526
6	528 – 9005
7	9006 – 9533

The data range shown in table L(2) is plotted against the bus voltage profile as shown.

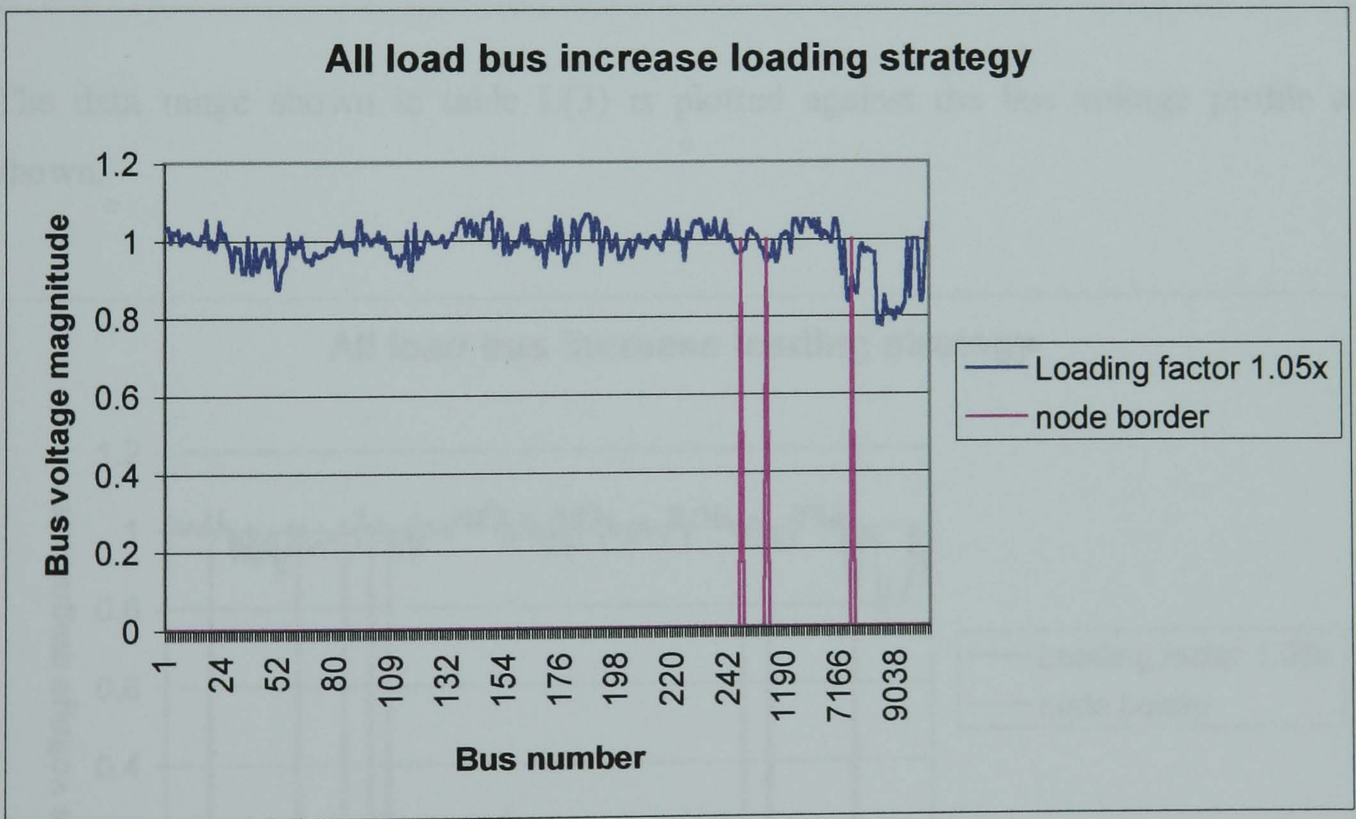


Figure L(1): Partitioning performance for attribute value limit of 0.8

Note that the vertical lines in figure L(1) denote the node borders of bus 1 to 247, 248 to 526, 528 to 9005 and 9006 to 9533 respectively.

Table L(3): A table showing attribute range for leaf nodes created by the FDT with attribute level limit of 0.9

Leaf nodes (Node No.) of FDT results for system loading of 1.05x at all load buses with attribute level limit of 0.9	Attribute range in leaf node (Bus No.)
16	1 – 19
17	20 – 60
15	61 – 85
13	86 – 98
9	99 – 107
5	108 – 247
6	248 – 526
10	528 – 9005
11	9006 – 9533

The data range shown in table L(3) is plotted against the bus voltage profile as shown.

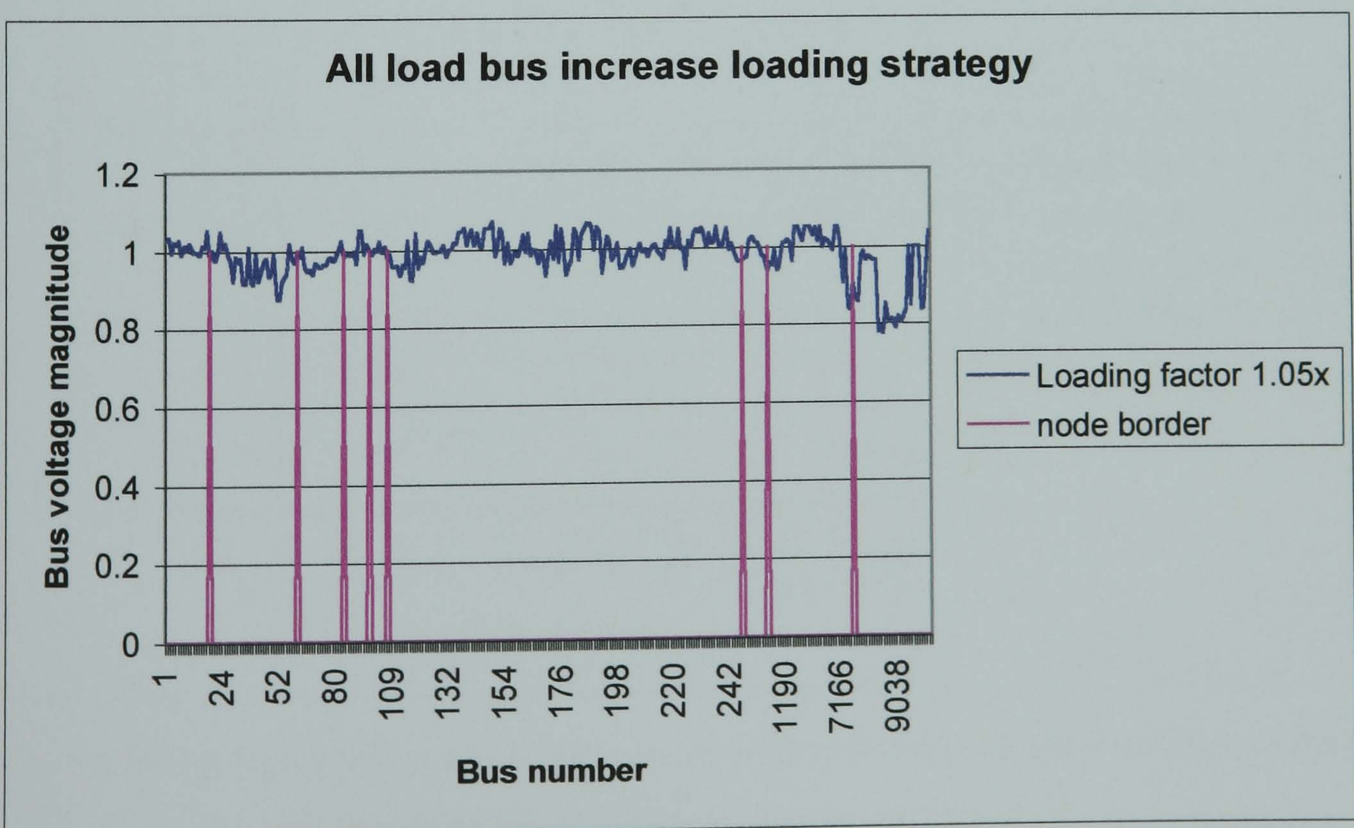


Figure L(2): Partitioning performance for attribute value limit of 0.9

Note that the vertical lines in figure L(2) denote the node borders as listed in table L(3) respectively.

Discussion

The performance of the FDT in general has improved since the amount of leaf nodes without pruning is small and the execution time is very fast. Using an attribute value limit of 0.8, the results shown in figure L(2) shows good classification properties where the strong buses are classified in nodes 4 and 6 whereas node 2 contains a majority of strong buses with a small amount of weak buses included. Node 7 contains a majority of weak buses. As the attribute value limit is increased to 0.9, accuracy of classification is improved where the weak buses are classified at nodes 11 and 17 and the rest of the leaf nodes contain relatively strong buses. Comparing the two limits, using a limit of 0.9 will give more leaf nodes but improved classification of weak busses. Therefore using an attribute value limit of 0.9 in the stop split stage improved the FDT significantly.

Appendix M

Bus voltage angle variation analysis with different load strategies

<u>Overview</u>

System	: Standard IEEE 300 bus system
Power flow software used	: MatPower operating under MatLab.
Type of test made	: (a) Load increment on all load buses
	(b) Load increment on selected heavily loaded buses
	(c) Load increment on selected lightly loaded buses.

Approach to analysis:

(a) Run power flow software using different load increase on all or selected load busses.

(b) Use output results for analysis

Results:

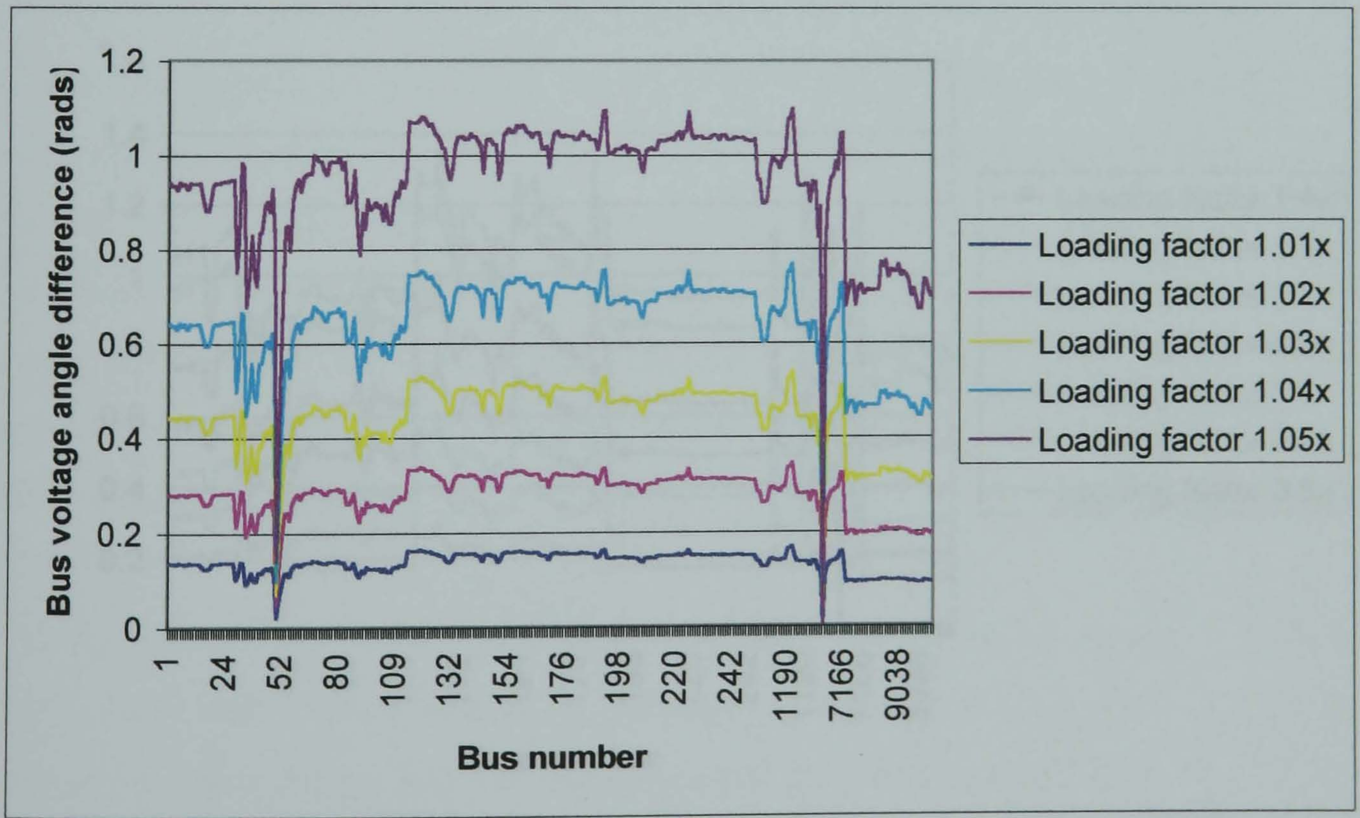


Figure M(1): Bus voltage angle difference with reference to the nominal loading factor at all load centres loading strategy

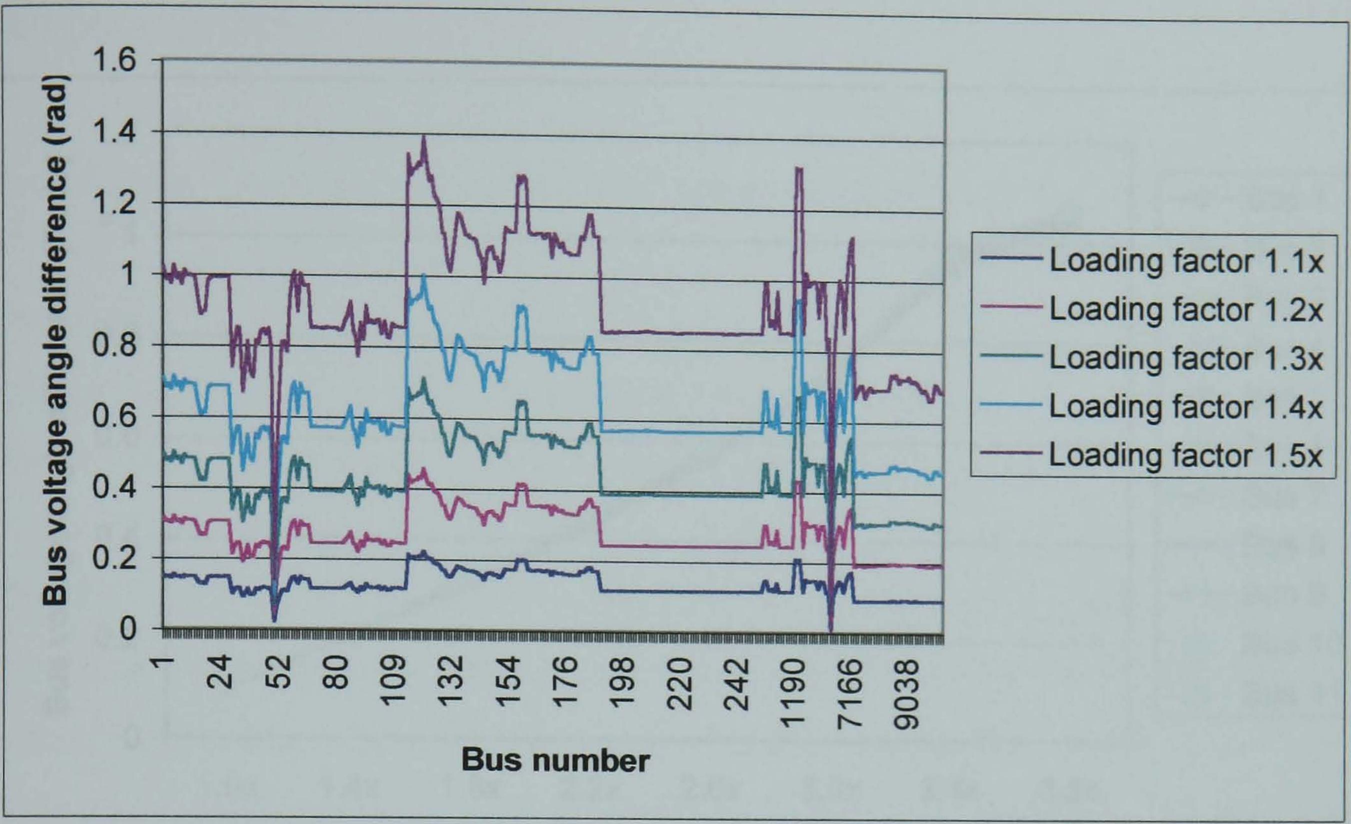


Figure M(2): Bus voltage angle difference with reference to the nominal loading factor at heavily loaded buses loading strategy

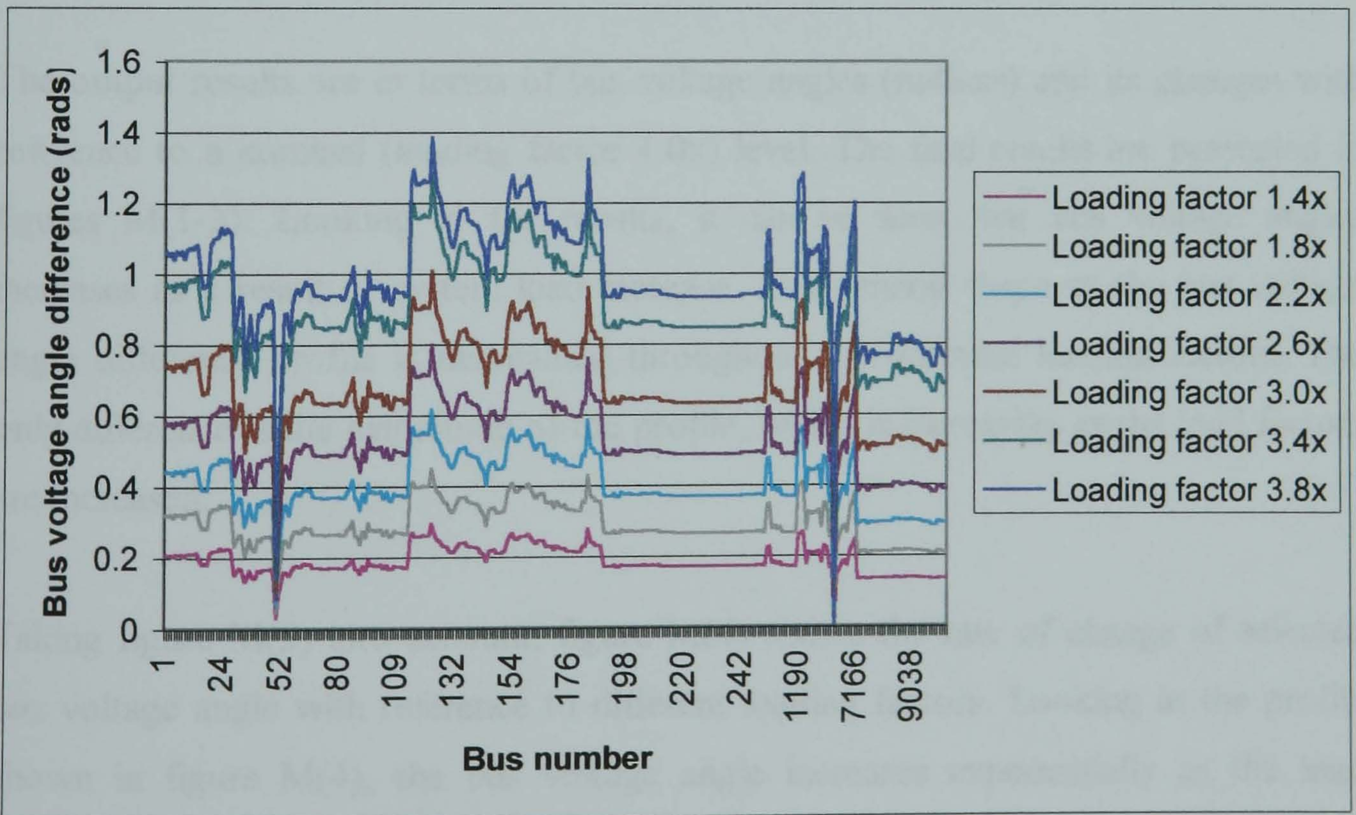


Figure M(3): Bus voltage angle difference with reference to the nominal loading factor at lightly loaded buses loading strategy

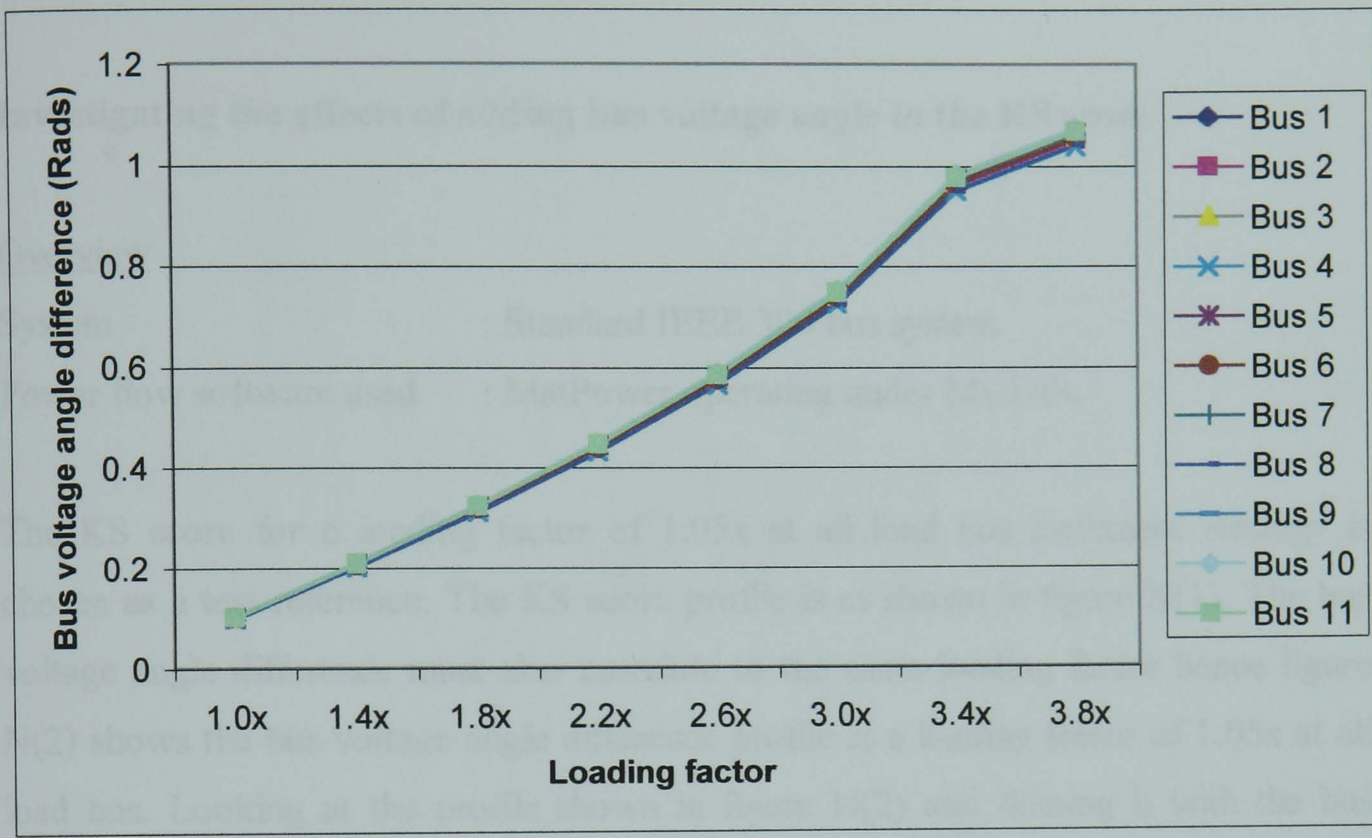


Figure M(4): Selected bus voltage angle with reference to nominal loading factor at lightly loaded buses loading strategy.

The output results are in terms of bus voltage angles (radians) and its changes with reference to a nominal (loading factor 1.0x) level. The final results are presented in figures M(1-3). Looking at the results, it can be seen that bus voltage angles increases as a result of system load increase. The general shape of the bus voltage angle difference profile is maintained throughout the different loading factors. The only difference is the magnitude of the profile, which is increasing as the load factors are increased.

Taking figure M(3) into account, figure M(4) shows the rate of change of selected bus voltage angle with reference to different loading factors. Looking at the profile shown in figure M(4), the bus voltage angle increases exponentially as the load factor is increased. Therefore a conclusion can be made that bus voltage angle does show a significant change as the system loading is increased.

Appendix N

Investigating the effects of adding bus voltage angle to the KS score

Overview

System : Standard IEEE 300 bus system

Power flow software used : MatPower operating under MatLab.

The KS score for a loading factor of 1.05x at all load bus increment strategy is chosen as a test reference. The KS score profile is as shown in figure N(1). The bus voltage angle difference must also correlate to the same loading factor hence figure N(2) shows the bus voltage angle difference profile at a loading factor of 1.05x at all load bus. Looking at the profile shown in figure N(2) and relating it with the bus voltage profile at similar loading level, it can be seen that low magnitude angle difference is observed at around the weak bus areas. A probable step is to include this by inverting the bus voltage angle hence the low valued angle difference will be at a maximum. But in doing so, one point in the profile where the bus voltage angle is 0 radian (slack bus) will be rendered to be infinite. By setting the inverted value to 0 radian, the inversion of bus voltage angle is as shown in figure N(3). Multiplying this, the KS score shown in figure N(1) will give the overall hybrid KS score as shown in figure N(4). Looking at the new hybrid KS profile and relating it to the bus voltage profile at the same loading factor, it is shown that the KS score is minimum at the weak bus area. In theory, if this KS score is applied back into the FDT process, better accuracy can be obtained.

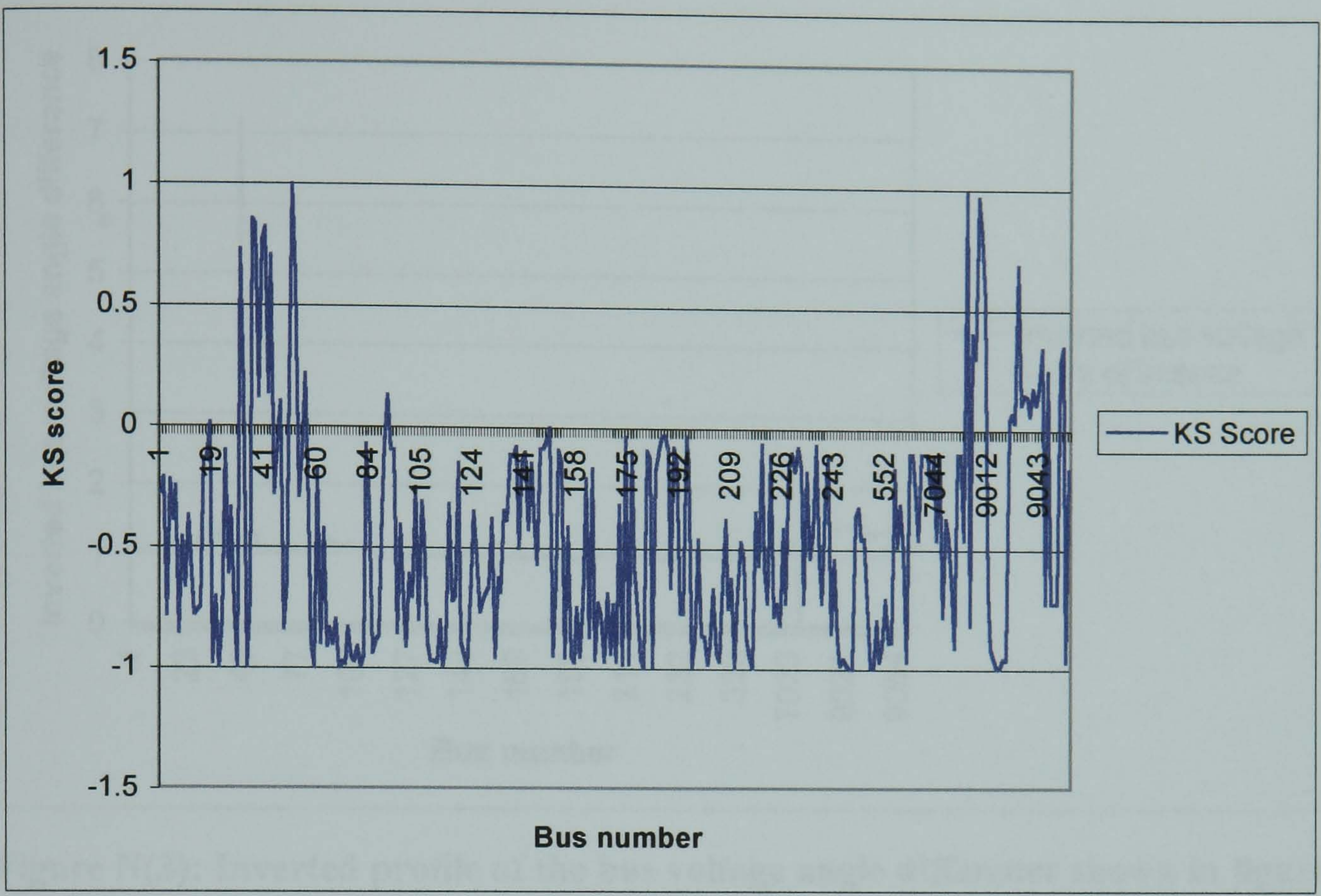


Figure N(1): KS score profile at 1.05x loading factor on all load bus strategy

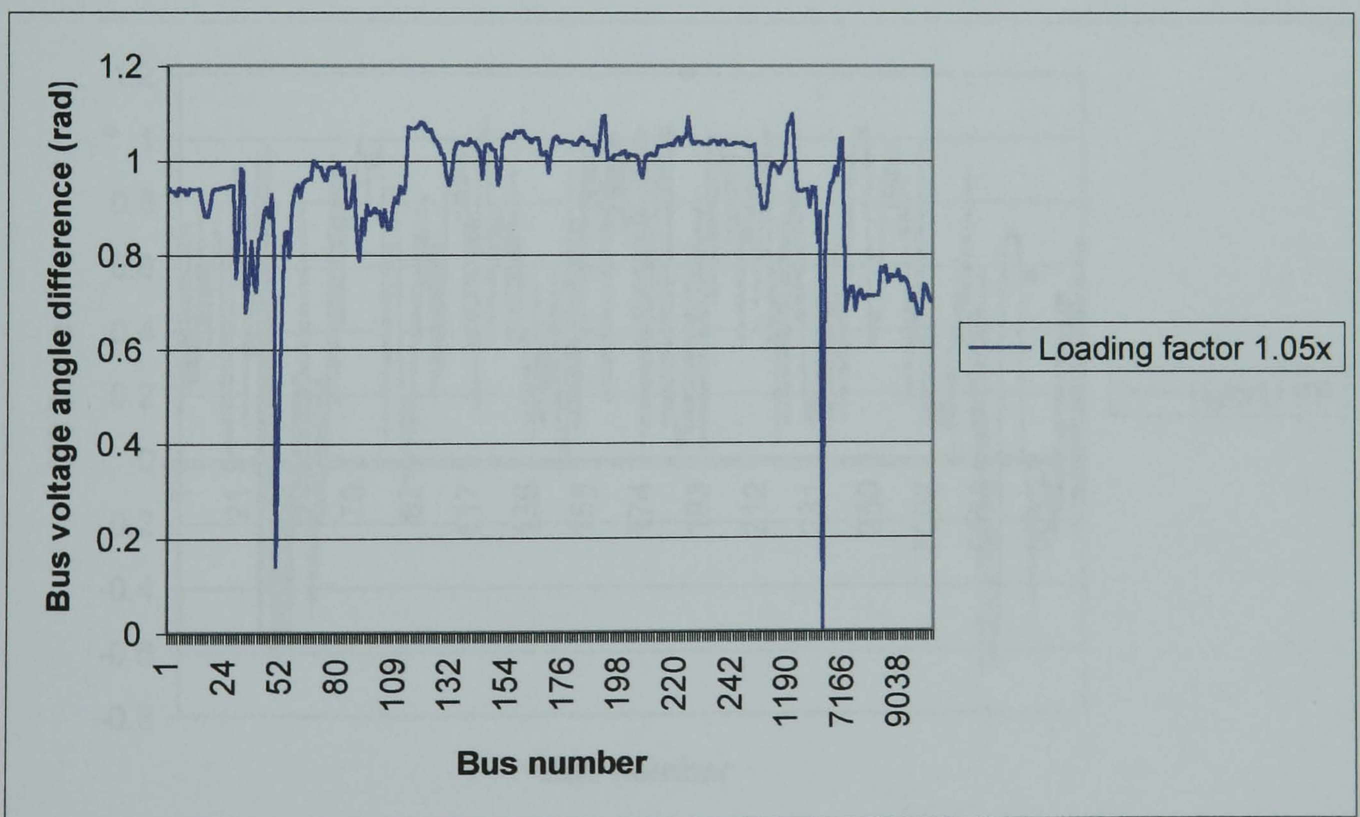


Figure N(2): Bus voltage angle difference with reference to nominal reference value (obtained at loading factor 1.0x) for 1.05x loading factor on all load bus strategy

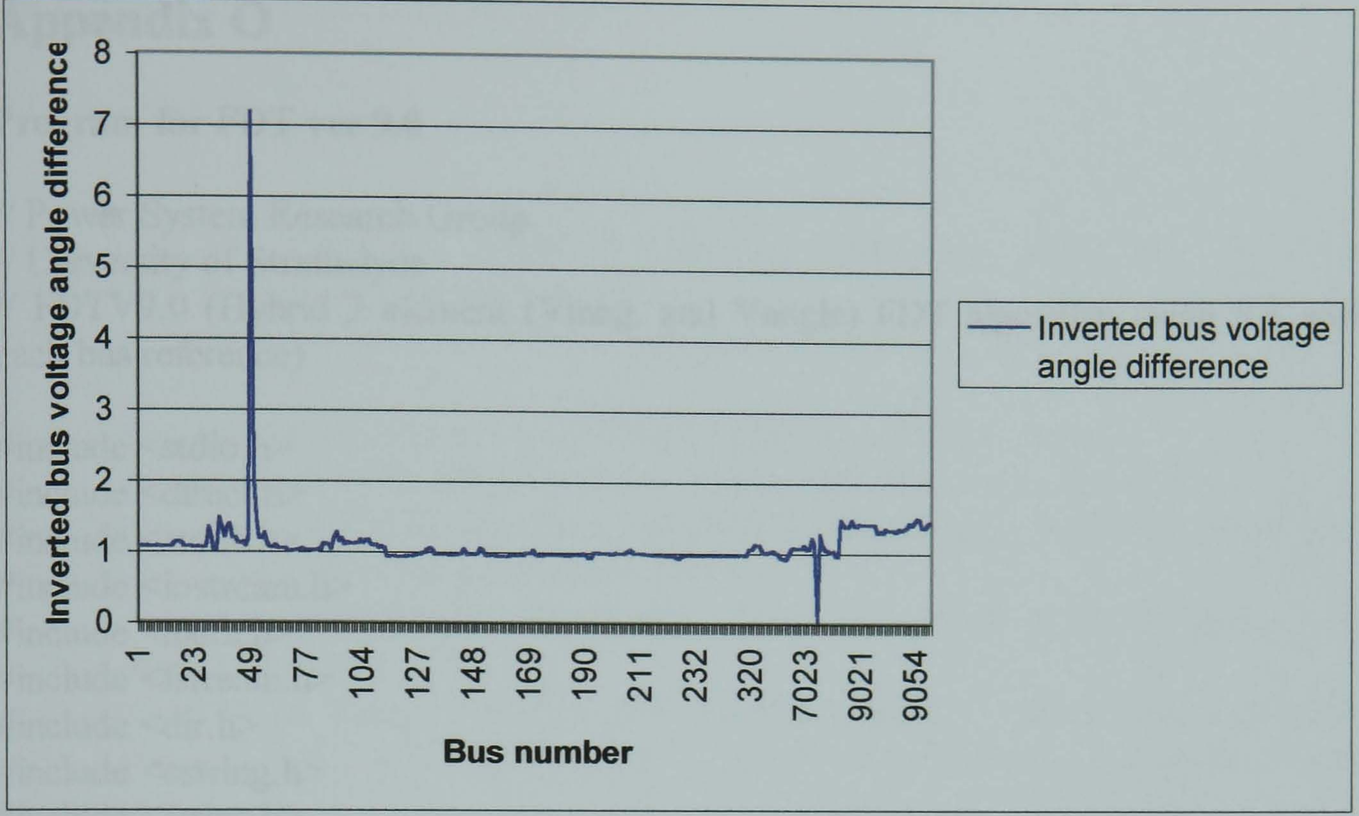


Figure N(3): Inverted profile of the bus voltage angle difference shown in figure N(2)

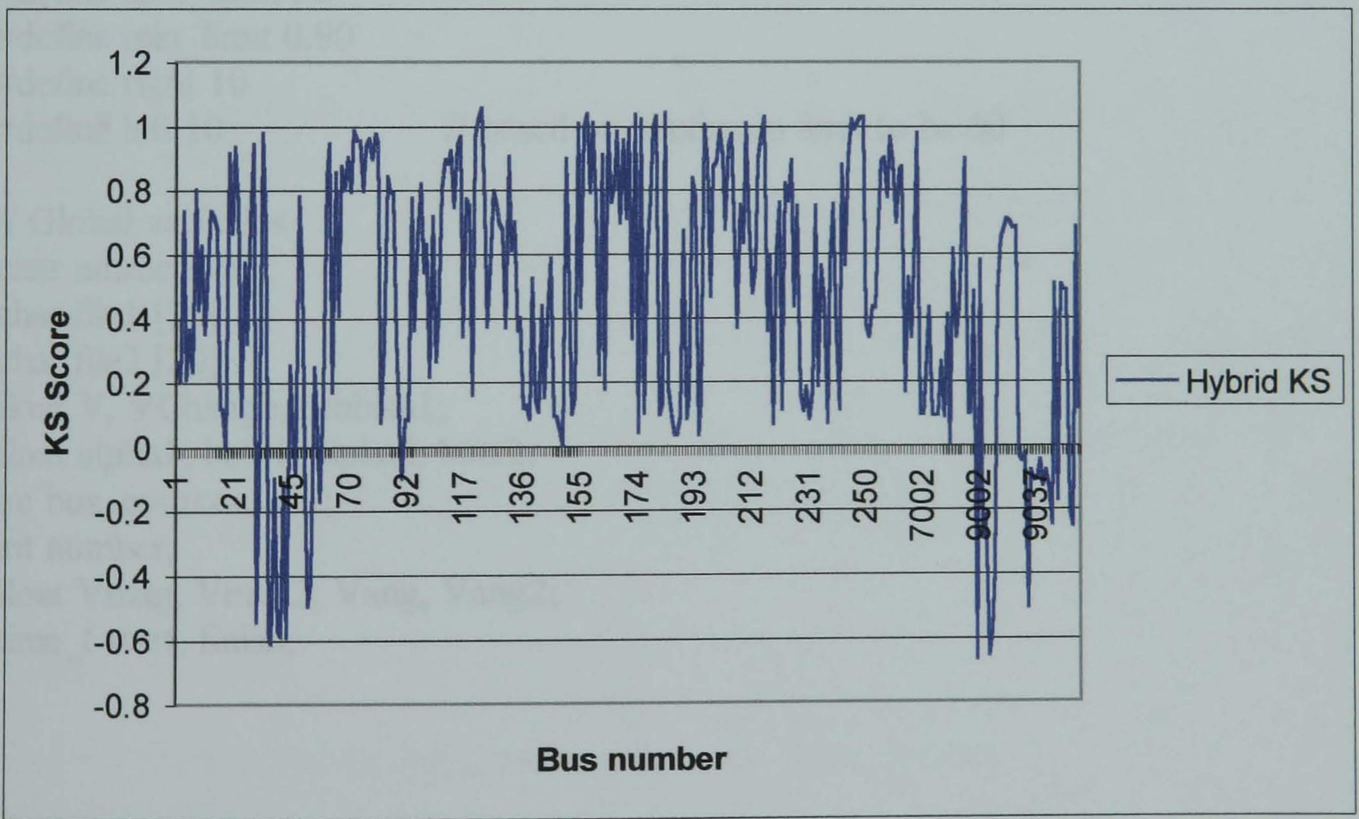


Figure N(4): Hybrid KS profile

Appendix O

Program for FDT ver 9.0

```
// Power System Research Group
// University of Strathclyde
// FDTV9.0 (Hybrid 2 element (Vmag. and Vangle) FDT algorithm using KS with
each bus reference)

#include <stdio.h>
#include <direct.h>
#include <stdlib.h>
#include <iostream.h>
#include <math.h>
#include <fstream.h>
#include <dir.h>
#include <cstring.h>
#include <string.h>
#include <time.h>
#include <iomanip.h>

#define direct "c:\\testbed"        // defining working directory
#define split_limit 50
#define min_limit 0.90
#define right 10
#define left 10                    // based on maximum limit to be 50

// Global variables
char ndirect[80];
char file1 [20];
char file2 [20];
float V, VChange, rubbish1;
float alpha1, beta1, alpha2, beta2;
int bus, counter, stat;
int number;
float Vmag, Vmag2, Vang, Vang2;
time_t start, finish;
```

```
// --------------------------- Training Set ----------------------------

void training()
{

float MaxCaP, MaxCaQ, MinCaQ, MaxOnP, MaxOnQ, MinOnQ, MaxCuP, MaxCuQ; // ref online genaration variable
float SysPload, SysQload, SysPloss, SysQloss, SysBchrg, SysSchrg; // ref system summary values
float Bus, Vmag, Vang, Pgen, Qgen, Pload, Qload; // ref Bus Summary
float FBus, TBus, FP, FQ, TP, TQ, Ploss, Qloss; // ref Branch Summary
float MaxCaP2, MaxCaQ2, MinCaQ2, MaxOnP2, MaxOnQ2, MinOnQ2, MaxCuP2, MaxCuQ2; // current online genaration variable
float SysPload2, SysQload2, SysPloss2, SysQloss2, SysBchrg2, SysSchrg2; // current system summary values
float Bus2, Vmag2, Vang2, Pgen2, Qgen2, Pload2, Qload2; // current Bus Summary
float FBus2, TBus2, FP2, FQ2, TP2, TQ2, Ploss2, Qloss2; // current Branch Summary
char reference [20];
char load_flow_in [20];

// Variable initialisation

rubbish1 = 0;

MaxCaP = MaxCaQ = MinCaQ = MaxOnP = MaxOnQ = MinOnQ = MaxCuP = MaxCuQ = 0;
SysPload = SysQload = SysPloss = SysQloss = SysBchrg = SysSchrg = 0;
Bus = Vmag = Vang = Pgen = Qgen = Pload = Qload = 0;
FBus = TBus = FP = FQ = TP = TQ = Ploss = Qloss = 0;
MaxCaP2 = MaxCaQ2 = MinCaQ2 = MaxOnP2 = MaxOnQ2 = MinOnQ2 = MaxCuP2 = MaxCuQ2 = 0;
SysPload2 = SysQload2 = SysPloss2 = SysQloss2 = SysBchrg2 = SysSchrg2 = 0;
Bus2 = Vmag2 = Vang2 = Pgen2 = Qgen2 = Pload2 = Qload2 = 0;
FBus2 = TBus2 = FP2 = FQ2 = TP2 = TQ2 = Ploss2 = Qloss2 = 0;


// Start conversion

cout << "Please give the reference load flow file (filename.dat) : ";
gets(reference);

cout << "Please give the current load flow file (filename.dat) : ";
gets(load_flow_in);

start = time(NULL);           // start timer
```

```
// cout << endl << "Reading data file" << endl;

ifstream ref;
ref.open(reference);            // reference file

ifstream lfin;
lfin.open (load_flow_in);       // current load flow file

// Reference files

ofstream sysref;
sysref.open ("sysref.dat");

ofstream busref;
busref.open ("busref.dat");

ofstream branchref;
branchref.open ("brnchref.dat");

// Current files

ofstream syscur;
syscur.open ("syscur.dat");

ofstream buscur;
buscur.open ("buscur.dat");

ofstream branchcur;
branchcur.open ("brnchcur.dat");

// put in node 1 sequence

strcpy (ndirect, direct);
strcat (ndirect, "\\");
strcat (ndirect, "node1");
mkdir (ndirect);

chdir(ndirect);                      // setting train.dat into node 1

ofstream treedata;
treedata.open ("train.dat");          // input for fuzzification and Beta/Alpha test
```

```
// System Header file

// cout << "Reading reference & current result header" << endl;

while (!ref.eof() || !lfin.eof())
{

// System Reference header file

        ref >> MaxCaP;
        ref >> ws;
        if (MaxCaP == -999)                     // to stop at end points in data file
                {
                        lfin >> rubbish1;        // getting rid of rubbish end no: -999
                        lfin >> ws;
                        goto bus;
                }
        ref >> MinCaQ;
        ref >> ws;
        ref >> MaxCaQ;
        ref >> ws;
        ref >> MaxOnP;
        ref >> ws;
        ref >> MaxOnQ;
        ref >> ws;
        ref >> MinOnQ;
        ref >> ws;
        ref >> MaxCuP;
        ref >> ws;
        ref >> MaxCuQ;
        ref >> ws;
        ref >> SysPload;
        ref >> ws;
        ref >> SysQload;
        ref >> ws;
        ref >> SysPloss;
        ref >> ws;
        ref >> SysQloss;
        ref >> ws;
        ref >> SysBchrg;
        ref >> ws;
        ref >> SysSchrg;
        ref >> ws;
```

```
// reading current system header file

        lfin >> MaxCaP2;
        lfin >> ws;
        lfin >> MinCaQ2;
        lfin >> ws;
        lfin >> MaxCaQ2;
        lfin >> ws;
        lfin >> MaxOnP2;
        lfin >> ws;
        lfin >> MaxOnQ2;
        lfin >> ws;
        lfin >> MinOnQ2;
        lfin >> ws;
        lfin >> MaxCuP2;
        lfin >> ws;
        lfin >> MaxCuQ2;
        lfin >> ws;
        lfin >> SysPload2;
        lfin >> ws;
        lfin >> SysQload2;
        lfin >> ws;
        lfin >> SysPloss2;
        lfin >> ws;
        lfin >> SysQloss2;
        lfin >> ws;
        lfin >> SysBchrg2;
        lfin >> ws;
        lfin >> SysSchrg2;
        lfin >> ws;

// printing current system file

syscur << MaxCaP2 << endl << MinCaQ2 << endl << MaxCaQ2 << endl <<
MaxOnP2 << endl << MaxOnQ2 << endl << MinOnQ2 << endl << MaxCuP2 <<
endl << MaxCuQ2 << endl << SysPload2 << endl << SysQload2 << endl <<
SysPloss2 << endl << SysQloss2 << endl << SysBchrg2 << endl << SysSchrg2 <<
endl;

// printing reference system file.

sysref << MaxCaP << endl << MinCaQ << endl << MaxCaQ << endl << MaxOnP
<< endl << MaxOnQ << endl << MinOnQ << endl << MaxCuP << endl <<
MaxCuQ << endl << SysPload << endl << SysQload << endl << SysPloss << endl
<< SysQloss << endl << SysBchrg << endl << SysSchrg << endl;
```

```
        // printing tree data
/*

possible tree data inputs

*/

//      treedata   <<   setw(5)   <<   (fabs(SysQload-SysQload2)/fabs(MaxCuQ-
MaxCuQ2)) << endl;   to be looked later   - agie -
}

// bus results

bus:
// cout << "Reading reference & current bus results" << endl;

while (!ref.eof() || !lfin.eof())
{

// reading system bus file

        ref >> Bus;
        ref >> ws;
        if (Bus == -999)
                {
                        lfin >> rubbish1;
                        lfin >> ws;
                        goto branch;
                }
        ref >> Vmag;
        ref >> ws;
        ref >> Vang;
        ref >> ws;
        ref >> Pgen;
        ref >> ws;
        ref >> Qgen;
        ref >> ws;
        ref >> Pload;
        ref >> ws;
        ref >> Qload;
        ref >> ws;
```

```
// reading current bus file

        lfin >> Bus2;
        lfin >> ws;
        lfin >> Vmag2;
        lfin >> ws;
        lfin >> Vang2;
        lfin >> ws;
        lfin >> Pgen2;
        lfin >> ws;
        lfin >> Qgen2;
        lfin >> ws;
        lfin >> Pload2;
        lfin >> ws;
        lfin >> Qload2;
        lfin >> ws;

// printing current bus file

buscur << setw (5) << Bus2 << " " << setw (7) << Vmag2 << "    " << setw (7) <<
Vang2 << "    " << setw (7) << Pgen2 << "    " << setw(7) << Qgen2 << "    " <<
setw(7) << Pload2 << "  " << setw(7) << Qload2 << endl;

// printing reference system file.

busref << setw (5) << Bus << " " << setw (7) << Vmag << "    " << setw (7) <<
Vang << "   " << setw (7) << Pgen << "    " << setw(7) << Qgen << "    " << setw(7)
<< Pload << "  " << setw(7) << Qload << endl;

// printing tree data.

/*

The tree data will consist of current and reference voltage magnitude and voltage
angles values
only. This is used for KS score analysis

*/
treedata << setw (5) << Bus2 << "   " << setw(5) << Vmag << "    " << setw (10) <<
Vang*(3.142/180) << "      " << setw(10) << Vmag2 << "      " << setw(10) <<
Vang2*(3.142/180) << endl;
}
```

```
// reading ref branch values

branch:
// cout << "Reading reference & current branch results" << endl;

while (!ref.eof() || !lfin.eof())
{

        // reading reference branch file

        ref >> FBus;
        ref >> ws;
        ref >> TBus;
        ref >> ws;
        ref >> FP;
        ref >> ws;
        ref >> FQ;
        ref >> ws;
        ref >> TP;
        ref >> ws;
        ref >> TQ;
        ref >> ws;
        ref >> Ploss;
        ref >> ws;
        ref >> Qloss;
        ref >> ws;

        // reading current branch file

                lfin >> FBus2;
                lfin >> ws;
                lfin >> TBus2;
                lfin >> ws;
                lfin >> FP2;
                lfin >> ws;
                lfin >> FQ2;
                lfin >> ws;
                lfin >> TP2;
                lfin >> ws;
                lfin >> TQ2;
                lfin >> ws;
                lfin >> Ploss2;
                lfin >> ws;
                lfin >> Qloss2;
                lfin >> ws;
```

```
        // printing current branch file

branchcur << setw (5) << FBus2 << "    " << setw(5) << TBus2 << "    " << setw(7)
<< FP2 << "      " << setw(7) << FQ2 << "      " << setw(7) << TP2 << "     " <<
setw(7) << TQ2 << "     " << setw(7) << Ploss2 << "     " << setw(7) << Qloss2 <<
endl;

        // printing reference branch file

branchref << setw (5) << FBus << "    " << setw(5) << TBus << "    " << setw(7) <<
FP << "     " << setw(7) << FQ << "     " << setw(7) << TP << "     " << setw(7) <<
TQ << "    " << setw(7) << Ploss << "    " << setw(7) << Qloss << endl;
        }

ref.close ();
sysref.close ();
busref.close ();
branchref.close ();
treedata.close ();

// cout << "Finish setting the data " << endl;
}


// -------------------------- Alpha Beta Search ------------------------------

void fuzzyinit()
{
/*
This is the fuzzy initialisation sub program. Currently, the values where we
need to find the value of alpha and beta is for:

(a) Current bus voltage values only

The bus voltage angle will be used later in the program

*/

float max1, max2, min1, min2;

//FFDT Initialisation....

alpha1 = alpha2 = beta1 = beta2 = 0;
max1 = max2 = min1 = min2 =0;
bus = 0;
```

```
// Determining suitable Beta value to cover the whole data set

        ifstream betain;
        betain.open("train.dat");

        chdir (direct);          // place Va-b in main directory.. easier for calculations
        ofstream V_data;
        V_data.open("Com_A-B.dat");

        min1 = min2 = 2;

        while (!betain.eof())
        {
                betain >> bus;
                betain >> ws;
                betain >> Vmag;
                betain >> ws;
                betain >> Vang;
                betain >> ws;
                betain >> Vmag2;
                betain >> ws;
                betain >> Vang2;
                betain >> ws;

        // Analysing Vref

                if (Vmag >= max1)
                        max1 = Vmag;
                else if (Vmag <= min1)
                        min1 = Vmag;
                else min1 = min1;

        // Analysing Vcur

                if (Vmag2 >= max2)
                        max2 = Vmag2;
                else if (Vmag2 <= min2)
                        min2 = Vmag2;
                else min2 = min2;


    }
```

```
// Beta and alpha calculations



// Beta alpha for Vref

        beta1 = ((max1 - min1) /2) + min1;
        alpha1 = 0.5/(max1-beta1);
        V_data << setw(5) << alpha1 << "   " << setw(5) << beta1 << endl;

// Beta alpha for Vcur

        beta2 = ((max2 - min2) /2) + min2;
        alpha2 = 0.5/(max2-beta2);
        V_data << setw(5) << alpha2 << "   " << setw(5) << beta2 << endl;



V_data.close();
betain.close();

}

// ---------------Data Reading & Fuzzification template---------------------------

void fuzzy()
{

float fuzzyVmag, fuzzyVmag2, a1, a2;
int bus;

        chdir (direct);
        ifstream V_ab;
        V_ab.open("Com_A-B.dat");           // common A-B, in this case V A-B

        V_ab >> alpha1;
        V_ab >> ws;
        V_ab >> beta1;
        V_ab >> ws;
        V_ab >> alpha2;
        V_ab >> ws;
        V_ab >> beta2;
        V_ab >> ws;

        V_ab.close();
```

```
        chdir (ndirect);
        ifstream fuzzyin;
        fuzzyin.open("train.dat");

// FFDT template sequence ......................

ofstream fuzzyout;
fuzzyout.open ("fuzzyout.dat");

while (!fuzzyin.eof())
{
                fuzzyin >> bus;
                fuzzyin >> ws;
                fuzzyin >> Vmag;
                fuzzyin >> ws;
                fuzzyin >> Vang;
                fuzzyin >> ws;
                fuzzyin >> Vmag2;
                fuzzyin >> ws;
                fuzzyin >> Vang2;
                fuzzyin >> ws;


        // fuzzyfying Vcur magnitude values

                a1 = alpha1*(Vmag - beta1);
                if (a1 <= -0.5)
                        fuzzyVmag = 0;
                else if (a1 >= 0.5)
                        fuzzyVmag = 1;
                else
                        fuzzyVmag = 0.5 + a1;

        // fuzzyfying Vref magnitude values

                a2 = alpha2*(Vmag2 - beta2);
                if (a2 <= -0.5)
                        fuzzyVmag2 = 0;
                else if (a2 >= 0.5)
                        fuzzyVmag2 = 1;
                else
                        fuzzyVmag2 = 0.5 + a2;
```

```
// printing fuzzyfied elements


fuzzyout << setw(5) << bus << "  " << setw(12) << fuzzyVmag << "  " << setw(12)
<< Vang << "     " << setw (5) << fuzzyVmag2  << "     " << setw (5) << Vang2 <<
endl;

}

fuzzyout.close();
fuzzyin.close();
}

// -------------------- Kolomogorov-Smirnov Analysis --------------------------

void ks ()
{
        int bus;
        float KS, Input1, Input2, zeta;


        zeta = 2;         // basic assumption refer to wehenkel paper for futher info

        ifstream ksin;
        ksin.open("fuzzyout.dat");

//      cout << "Calculating KS score " << endl;

        ofstream ksout;
        ksout.open ("ksout.dat");

        while (!ksin.eof())
        {
                ksin >> bus;
                ksin >> ws;
                ksin >> Input1;
                ksin >> ws;
                ksin >> Vang;
                ksin >> ws;
                ksin >> Input2;
                ksin >> ws;
                ksin >> Vang2;
                ksin >> ws;
```

```
// Calculating KS score

KS = pow(2.0,1.0/zeta)*(Input2 - Input1)/pow((pow((2.0*Input1 - 1.0), zeta) +
pow((2.0*Input2-1.0), zeta)),1.0/zeta);


// Printing out result of Hybrid KS score

        ksout << setw(5) << bus << "   " << setw(12) << KS*(Vang-Vang2) << endl;

        }

ksout.close();
ksin.close();
}

// ---------------------- Partitioning sequence ----------------------------

void partition ()
{
        float KS_in, maxKS, minKS;
        int maxKS_cord, minKS_cord, total_count, count, location, limit;

// Maximum value and location search ......

        ifstream maxsearch;
        maxsearch.open("ksout.dat");

        ofstream status;
        status.open ("status.dat");

        KS_in = 0;
        maxKS_cord = 0;
        maxKS = 0;
        minKS = 2;

        total_count = 1;
        stat = 0;

while (!maxsearch.eof())
        {
                maxsearch >> bus;
                maxsearch >> ws;
                maxsearch >> KS_in;
                maxsearch >> ws;
```

```
// Max min KS search

                    if (KS_in >= maxKS)
                    {
                            maxKS = KS_in;
                            maxKS_cord = total_count;     /* Internal Coordinate
for maxKSVmag    */
                    }
                    else if (KS_in <= minKS)
                    {
                            minKS = KS_in;
                            minKS_cord = total_count;     /* Internal Coordinate
for minKSVmag    */
                    }
                    else
                            minKS = minKS;

                    total_count = total_count + 1;
        }

// cout << "KSVmag = " << KSVmag_cord << endl << "KSVChange = " <<
KSVChange_cord << endl << "KSVAngChange = " << KSVAngChange_cord <<
endl;

maxsearch.close();

// checking the location of the KS score

location = total_count - minKS_cord;        // minKS location

if (maxKS_cord <= left)             // max KS score located at extreme left of test node
        {
                if (location <= right)        // min KS score located at extreme right
                        {
                                status << "3";
                                stat = 3;
                        }
                else if (location <= left)
                        {
                                status << "3";
                                stat = 3;
                        }
```

```
                else
                        {
                        limit = minKS_cord;
                        stat = 2;
                        }
        }

else if (maxKS_cord <= right)          // KS score located at extreme right of test node
        {
                if (location <= right)         // min KS score located at extreme right
                        {
                                status << "3";
                                stat = 3;
                        }
                else if (location >= left)
                        {
                                status << "3";
                                stat = 3;
                        }
                else
                        {
                        limit = minKS_cord;      // setting min KS as partitioning limit
                        stat = 2;
                        }
        }
else
        {
                limit = maxKS_cord;            // setting min KS as partitioning limit
        stat = 2;
        }

// Partitionining Algorithm.........

if (stat != 3)
        {
                ifstream partition;
                partition.open("fuzzyout.dat");

                ofstream upper;
                upper.open ("upper.dat");

                ofstream lower;
                lower.open ("lower.dat");

                count = 1;
```

```
        while (!partition.eof())
        {
                if (count < limit)
                        {
                                partition >> bus;
                                partition >> ws;
                                partition >> Vmag;
                                partition >> ws;
                                partition >> Vang;
                                partition >> ws;
                                partition >> Vmag2;
                                partition >> ws;
                                partition >> Vang2;
                                partition >> ws;

                                upper << setw(5) << bus << "  " << setw(12)
<< Vmag << "  " << setw(12) << Vang << "    " << setw (10) << Vmag2 << "   " <<
setw (10) << Vang2 << endl;
                        }
                else
                        {
                                partition >> bus;
                                partition >> ws;
                                partition >> Vmag;
                                partition >> ws;
                                partition >> Vang;
                                partition >> ws;
                                partition >> Vmag2;
                                partition >> ws;
                                partition >> Vang2;
                                partition >> ws;

                                lower << setw(5) << bus << "  " << setw(12)
<< Vmag << "    " << setw(12) << Vang << "     " << setw (10) << Vmag2 << "     "
<< setw (10) << Vang2 << endl;

                        }
                count = count + 1;
        }
        partition.close();
        upper.close();
        lower.close();
        status << "1";
        status.close();
    }
```

```
else
        {
        status.close();
        }

}

// ----------------- Dfuzzification process -----------------------------

void dfuzzy ()
{
        int bus;
        float FVmag, FVmag2;

        chdir (direct);
        ifstream data;
        data.open("Com_A-B.dat");

while (!data.eof())
{
                data >> alpha1;
                data >> ws;
                data >> beta1;
                data >> ws;
                data >> alpha2;
                data >> ws;
                data >> beta2;
}

// Defuzzification sequence ..............

chdir (ndirect);
ofstream defuzzy;
defuzzy.open(file2);

ifstream fuzzy;
fuzzy.open(file1);

while (!fuzzy.eof())
{
                fuzzy >> bus;
                fuzzy >> ws;
                fuzzy >> FVmag;
                fuzzy >> ws;
                fuzzy >> Vang;
                fuzzy >> ws;
```

```
                fuzzy >> FVmag2;
                fuzzy >> ws;
                fuzzy >> Vang2;
                fuzzy >> ws;

/*


*/


// defuzzy Vmag values

                if (FVmag >= 3.0)
                        Vmag = 1.5;
                else if (FVmag < 0)
                        Vmag = 0.2;
                else
                Vmag = ((FVmag-0.5)/alpha1)+beta1;

// defuzzy VMag2 values

                if (FVmag2 >= 3.0)
                        Vmag2 = 1.5;
                else if (FVmag2 < 0)
                        Vmag2 = 0.2;
                else
                Vmag2 = ((FVmag2-0.5)/alpha2)+beta2;

// Printing defuzzy results

defuzzy << setw(5) << bus << "   " << setw(10) << Vmag << "     " << setw(10) <<
Vang << "     " << setw (5) << Vmag2 << "      " << setw (5) << Vang2 << endl;

                }
defuzzy.close();
fuzzy.close();

}
```

```
// ---------------- Stop Split check --------------------

void stpsplit ()
{

float minV, maxV;
float totalV;
int no;

ifstream upper;
upper.open ("uprDfuz.dat");

ifstream lower;
lower.open ("lorDfuz.dat");

ofstream spltcode;
spltcode.open ("spltcode.dat");


// Upper partition analysis -------------------------------------------------

no = totalV = 0;

maxV =  0;              // initialising max values
minV =  5;                // initialising min values (arbitary number)

while (!upper.eof())
        {
                        upper >> bus;
                        upper >> ws;
                        upper >> Vmag;
                        upper >> ws;
                        upper >> Vang;
                        upper >> ws;
                        upper >> Vmag2;
                        upper >> ws;
                        upper >> Vang2;
                        upper >> ws;

                        no = no + 1;

// for average calculations

                        totalV = totalV + Vmag2;
```

```
// Max - min search for V

                    if (Vmag2 >= maxV)
                    {
                            maxV = Vmag2;
                    }
                    else if (Vmag2 <= minV)
                    {
                            minV = Vmag2;
                    }
                    else
                            minV = minV;

}

// Value analysis for upper partitioned values


/*
        Uprdfuz.dat split code
        Currently i am trying a splt code based upon the no of the partitioned set
*/

        if (minV <= min_limit)          // population has unstable bus
        {
                if (no > split_limit)       // check if the population count is big
                    {
                            spltcode << "2" << endl;
                    }
                else
                    {
                            spltcode << "1" << endl;
                    }
        }
        else                                                                    //
population has stable bus only
        {
                spltcode << "1" << endl;
        }
```

```
// Lower.dat analysis --------------------------------------------------

no = totalV = 0;

maxV =  0;              // initialising max values
minV =  5;                // initialising min values (arbitary number)

//cout << endl << "For Lower Partitioned Data: " << endl;
while (!lower.eof())
        {

                lower >> bus;
                lower >> ws;
                lower >> Vmag;
                lower >> ws;
                lower >> Vang;
                lower >> ws;
                lower >> Vmag2;
                lower >> ws;
                lower >> Vang2;
                lower >> ws;

                no = no + 1;

// for average calculations

                totalV = totalV + Vmag2;

// Max - min search for V

                        if (Vmag2 >= maxV)
                        {
                                maxV = Vmag2;
                        }
                        else if (Vmag2 <= minV)
                        {
                                minV = Vmag2;
                        }
                        else
                                minV = minV;

        }
```

```
// Value analysis for lower partitioned values

/*
        lordfuz.dat split code
        Currently i am trying a splt code based upon the no of the partitioned set
*/

        if (minV <= min_limit)
                {
                        if (no > split_limit)
                                {
                                        spltcode << "2" << endl;
                                }
                        else
                                {
                                        spltcode << "1" << endl;
                                }

                }

        else
                {
                        spltcode << "1" << endl;
                }

        upper.close();
        lower.close();
        spltcode.close();
}

// -------------- copy and file rename ----------------------------

void copy_rename()
{
        int code, num;
        char node [10];

// Storing current source node values

                chdir (direct);
                ofstream bin3;
                bin3.open ("srcnode.dat");
                        bin3 << direct << "\\" << "node" << number << endl;
                bin3.close ();
```

```
// Reading stop split code

        ifstream binread4;
        binread4.open ("srcnode.dat");
                binread4.getline (ndirect, 80);
        binread4.close();

        chdir (ndirect);


//      cout << "Reading stop split code  ";

        ifstream spltcode;
        spltcode.open ("spltcode.dat");

        strcpy (file1,"Uprdfuz.dat");

        while (!spltcode.eof())
        {
        spltcode >> code;
        spltcode >> ws;

        num = counter + 1;      // setting dir number - uprdfuz at node(counter + 1) &
lordfuz at node (counter + 2)

/*

        Node allocation sequence are follows the sequence below:

                (a) Node (a) = uprdefuz converted into train.dat
                (b) Node (a+1) = lordfuz converted into train.dat
*/

                // Creating new directory - for chdir () function

                ofstream bin;
                bin.open ("bin.dat");
                        bin << direct << "\\" << "node" << num << endl;
                bin.close ();

                ifstream binread;
                binread.open ("bin.dat");
                        binread.getline (ndirect, 80);
                binread.close();

                // Creating new node folder - for mkdir () function
```

```
		ofstream bin2;
		bin2.open ("bin.dat");
			bin2 << "node" << num << endl;
		bin2.close ();

		ifstream binread2;
		binread2.open ("bin.dat");
			binread2.getline (node, 10);
		binread2.close();

		// copying file and setting spltcode -------------

		ifstream copyfl;
		copyfl.open (file1);

		chdir (direct);
		mkdir (node);
		chdir (ndirect);		// Changing into new directory

	if (code == 1)	// splt check, 1 = leaf, 2 = split further
		{
		ofstream copyin1;
		copyin1.open("leaf.dat");		// just copy defuzzification results
because of leaf status

		ofstream stat1;
		stat1.open ("status.dat");

		while(!copyfl.eof())
			{
				copyfl >> bus;
				copyfl >> ws;
				copyfl >> Vmag;
				copyfl >> ws;
				copyfl >> Vang;
				copyfl >> ws;
				copyfl >> Vmag2;
				copyfl >> ws;
				copyfl >> Vang2;
				copyfl >> ws;

				copyin1 << bus << "    " << setw(10) << Vmag << "    "
<< setw(10) << Vang << "    " << setw (10) << Vmag2 << "    " << setw (10) <<
Vang2 << endl;
			}
```

```
            stat1 << "3" << endl;          // leaf stat code
            stat1.close ();
            copyin1.close();

            }
        else        // split code == 2, hence further partition
            {

            ofstream copyin2;
            copyin2.open("train.dat");

            ofstream stat2;
            stat2.open ("status.dat");

            while(!copyfl.eof())
                {
                        copyfl >> bus;
                        copyfl >> ws;
                        copyfl >> Vmag;
                        copyfl >> ws;
                        copyfl >> Vang;
                        copyfl >> ws;
                        copyfl >> Vmag2;
                        copyfl >> ws;
                        copyfl >> Vang2;
                        copyfl >> ws;

                        copyin2 << setw(5) << bus << "    " << setw(10) <<
Vmag << "    " << setw(10) << Vang << "    " << setw(10) << Vmag2 << "    " <<
setw(10) << Vang2 << endl;
                }

            stat2 << "2" << endl;          // Further partition stat code

            stat2.close ();
            copyin2.close();

            }

        copyfl.close();

        chdir (direct);

        ifstream binread3;
        binread3.open ("srcnode.dat");
```

```
                binread3.getline (ndirect, 80);
        binread3.close();

        chdir (ndirect);
        counter = counter + 1;
        strcpy (file1,"lordfuz.dat");

        }

        spltcode.close ();
}

// -------------------- Leaf search and identify -------------------------

void leaf_search ()
{
int code, number, count;
chdir (direct);

ofstream leaf;                          // leaf node list
leaf.open ("leafy.dat");

        leaf << "The following are leaf nodes for knowledge extraction" << endl;

count = number = 1;

while (number <= counter)
        {
                        chdir (direct);

                        ofstream temp_bin1;
                        temp_bin1.open ("tem_bin.dat");
                        temp_bin1 << direct << "\\node" << number;
                        temp_bin1.close();

                        ifstream temp_bin2;
                        temp_bin2.open ("tem_bin.dat");
                        temp_bin2.getline(ndirect, 80);
                        temp_bin2.close();

                        chdir (ndirect);

                        ifstream code_read;
                        code_read.open ("status.dat");
                                code_read >> code;
                                code_read >> ws;
```

```
                    code_read.close();


                    if (code == 3)          // code 3 => leaf node
                    {
                            leaf << "Node no: " << number << endl;
                            count = count + 1;

                    }

                    number = number + 1;
        }

leaf << "Total leaf nodes: " << count << endl;
leaf.close ();

}

// --------------------- Main program ----------------------------------

void main ()
{

int code, limit;
float time_difference;

_chdrive (3);           // change active drive to drive c:
chdir (direct);

number = 1;
counter = 1;

cout << " FDT version 9.0 - Hybrid FDT " << endl;

training ();

// Partitining starts

                chdir (ndirect);
fdt:
                fuzzyinit();
                fuzzy ();
                ks ();
                partition ();
                if (stat != 3)
                {
```

```
                strcpy(file2, "uprDfuz.dat");
                strcpy(file1,  "upper.dat");
                dfuzzy();
                strcpy(file2, "lorDfuz.dat");
                strcpy(file1, "lower.dat");
                dfuzzy();
                strcpy(file2, "dfuzout.dat");
                strcpy(file1, "fuzzyout.dat");
                dfuzzy();

                stpsplit();
                copy_rename();
                limit = counter;
        }
        else
        {
                limit = counter;
        }

// Auto searching for split code

        while (number <= limit)
        {
                chdir (direct);

                ofstream temp_bin1;
                temp_bin1.open ("tem_bin.dat");
                temp_bin1 << direct << "\\node" << number;
                temp_bin1.close();

                ifstream temp_bin2;
                temp_bin2.open ("tem_bin.dat");
                temp_bin2.getline(ndirect, 80);
                temp_bin2.close();

                chdir (ndirect);

                ifstream code_read;
                code_read.open ("status.dat");
                        code_read >> code;
                        code_read >> ws;
                code_read.close();

        if (code == 2)                // status code = 2 => further split is required
                {
                        goto fdt;
```

```
                }

                number = number + 1;
        }

// Searching for leaf nodes and create leaf node list for use of analysis.

leaf_search ();

finish = time(NULL);                    // stop timer
time_difference = difftime(finish, start);    // calculating time difference

cout << "Time taken : " << time_difference << " secs " << endl;
cout << "Thank you for using fdt ver 9.0 auto" << endl;

}
```