Decision Tree based Oscillatory Stability Assessment for Large Interconnected Power Systems

S. P. Teeuwsen, *Student Member, IEEE*, I. Erlich, *Member, IEEE*,

and M. A. El-Sharkawi, *Fellow, IEEE*

***Abstract* – This paper deals with a new method for eigenvalue prediction of critical stability modes of power systems based on decision trees. Special interest is focused on inter-area oscillations of large-scale interconnected power systems. The existing methods for eigenvalue computations are time-consuming and require the entire system model that includes an extensive number of states. However, using decision trees, the oscillatory stability can be predicted based on a few selected inputs. Hereby, the outputs of the tree are assigned to the damping ratio of the critical inter-area eigenvalues. Decision trees are fast, easy to train and provide high accuracy for eigenvalue prediction.**

***Index Terms*– Artificial Intelligence, Decision Tree, Oscillatory Stability Assessment, Large Power Systems**

I. Introduction

Inter-area oscillations in large-scale power systems are becoming more common especially for the European Interconnected Power System UCTE/CENTREL. The system has grown very fast in a short period of time due to the recent east expansion. This extensive interconnection alters the stability region of the network, and the system experiences inter-area oscillations associated with the swinging of many machines in one part of the system against machines in other parts. Moreover, for certain load flow conditions, the system damping changes widely [1], [2]. The deregulation of electricity markets in Europe aggravated the situation further due to the increasing number of long distance power transmissions. Moreover, the network is becoming more stressed also by the transmission of wind power.

In fact, the European power system is designed rather as a backup system to maintain power supply in case of power plant outages. The system is operated by several independent transmission utilities, joined by a large meshed high voltage grid. Because of the increasing long distance transmissions, the system steers closer to the stability limits. Thus, the operators need computational real time tools for assessing system stability. Of main interest in the European power system is hereby the oscillatory stability assessment (OSA). The use of on-line tools is even more complicated because TSOs exchange only restricted information. Each TSO controls a particular part of the power system, but the exchange of data between different parts is limited to a small number because of the high competition between the utilities.

However, the classical small-signal stability computation requires the entire system model and is time consuming for large power systems [3]. Therefore, this paper proposes using Artificial Intelligence (AI) methods for a fast on-line OSA based only on a small set of data. In literature, one can find various concepts for AI based automatic learning techniques in power systems [4]. The focus in this paper is on decision trees, which is only one of many AI methods, however it is close to engineering based decisions. The trees are structured physically and show some analogies to power transmission operator decisions.

II. 16-Machine Dynamic Test System

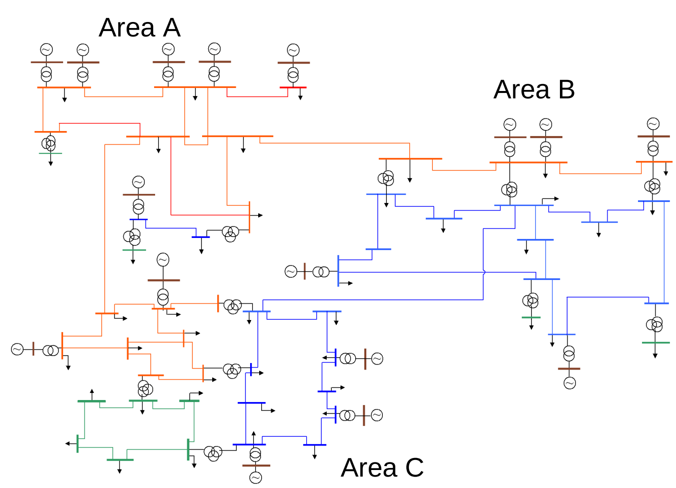

Fig. 1 One-Line Diagram of the PST 16-Machine Dynamic Test System

The PST16 System (Figure 1) used in this study is a 400/220 kV 16-machine dynamic test system. The network consists of 3 strongly meshed areas, which are connected by long distance transmission lines. Therefore, it allows studying different kinds of stability problems, especially inter-area oscillations, and has been developed based on characteristic parameters of the European power system [5]. Since the

S. P. Teeuwsen, University of Duisburg-Essen, Germany, (teeuwsen@uni-duisburg.de)

I. Erlich, University of Duisburg-Essen, Germany, (erlich@uni-duisburg.de)

M. A. El-Sharkawi, University of Washington, Seattle, USA, (elsharkawi@ee.washington.edu)

operating point in real power systems changes continuously, 5 different operating conditions are considered. These operating conditions include high and low load situations like in winter and in summer loads, and changes of the network topology when transmission lines are switched.

To generate learning samples for decision tree construction, various load flow scenarios under the 5 operating conditions are considered. These scenarios are generated by real power exchange between 2 areas. The different load flow scenarios result in generating 5,360 patterns for learning of the decision trees.

III. Feature Selection

Stability assessment methods require input information to make reliable predictions. These inputs should characterize the system sufficiently. Beside this, the variables must be measured and offered by the utilities. Large power systems include many state information such as transmission line flows, generated powers and demands, voltages, and voltage angles. However, any effective AI method, like decision trees, requires a small number of inputs, which must be selected first from the full set of variables. When used with too many inputs, AI methods will lead to long processing times and cannot be managed well [6]. According to the AI literature, the input variables are characterized as "features". In general, the selected features must represent the entire system, since a loss of information in the reduced set results in loss of both performance and accuracy in the AI methods. In large interconnected power systems, it is difficult to develop exact relationships between features and the targeted oscillatory stability. This is because the system is highly nonlinear and complex. For this reason, feature selection cannot be performed by engineering judgment or physical knowledge only, but it must be implemented according to a mathematical procedure or algorithm. In the AI science different methods are developed for this purpose. The selection technique applied in this study is based on physical pre-selection, followed by the Principal Component Analysis (PCA) and the k-Means cluster algorithm. Finally, the input features are selected from the clusters constructed by k-means. The selection procedure is shown as block diagram in Figure 2.

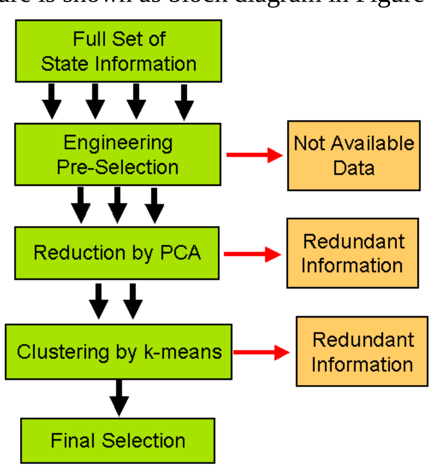

Fig. 2 Selection Procedure

A. Feature Pre-Selection

The key idea of the pre-selection is collecting as many data from the power system, which are assumed to be of physical interest for OSA. The focus is hereby on those features, which are both measurable in the real power system and available from the power utilities. Besides the typical power system features like real and reactive power or voltages, the rotating generator energy is used as feature as well. The last one is defined as the installed MVA of running blocks multiplied by the inertia constant. Since the number of running blocks is adjusted during the computation depending on the generated power, this feature provides information about the rotating mass in the system. The rotating generator energy is computed for both individual machines and all machines in the same area. The pre-selected features are listed in Table I. Note that the pre-selected features contain also voltage angles. Voltage angles are not measured in power systems nowadays, but this paper will point out, that voltage angles provide important information about the system stability. Moreover, they can be measured if necessary since the necessary equipment already exists.

Table I
Description of Pre-Selected Features

#	Feature Description	Symbol
1	Sum Generation per Area	P, Q
2	Individual Generator Power	P, Q
3	Rotating Generator Energy	E
4	Sum Rotating Generator Energy per Area	E
5	Power on Transmission Lines	P, Q
6	Power Exchanged between Areas	P, Q
7	Bus Voltages	V
8	Bus Voltage Angles	φ

B. Dimensionality Reduction

In the next selection step, the data are reduced in dimension by PCA, which is characterized by a high reduction rate and a minimal loss of information. The PCA technique is fast and can be applied to large data sets. However, a simple projection onto the lower dimensional space transforms the original features into new ones without physical meaning. Because feature selection techniques do not have this disadvantage, the PCA is not used for transformation but for dimensional reduction only, followed by a feature selection method. The PCA projections onto the lower dimensional space provide information which features are projected onto which principal axes. This information can be used for selection since similar features will be projected onto the same principal axes. Therefore, not the features but the projections are clustered in the lower dimensional space using the k-means cluster algorithm. The dimensionality reduction before clustering is necessary when processing large amounts of data because the k-means cluster algorithm leads to best results for small and medium size data sets.

Let **F** be a feature matrix of dimension, $p \times n$ where *n* is the number of the original feature vectors and *p* is the number of patterns. The empirical covariance matrix **C** of the normalized **F** is computed by

$$\mathbf{C} = \frac{1}{p-1} \cdot \mathbf{F}^{\mathrm{T}} \cdot \mathbf{F} \tag{1}$$

Let $\mathbf{T}$ be a $n \times n$ matrix of the eigenvectors of **C,** and the diagonal variance matrix $\mathbf{\Sigma}^2$ is given by

$$\mathbf{\Sigma}^2 = \mathbf{T}^{\mathrm{T}} \cdot \mathbf{C} \cdot \mathbf{T} \tag{2}$$

$\mathbf{\Sigma}^2$ includes the variances σ_x^2. Notice that the eigenvalues λ_k of the covariance matrix **C** are equal to the elements of the variance matrix $\mathbf{\Sigma}^2$. The standard deviation σ_k is also the singular value of **F**:

$$\sigma_k^2 = \lambda_k \qquad (1 \le k \le n) \tag{3}$$

The n eigenvalues of **C** can be determined and sorted in descending order $\lambda_1 \ge \lambda_2 \ge \ldots \ge \lambda_n$. While **T** is an n-dimensional matrix whose columns are the eigenvectors of **C**, $\mathbf{T}_q$ is a $n \times q$ matrix including q eigenvectors of **C** corresponding to the q largest eigenvalues of **C**. The value of q determines the size of the new dimension of the features and is smaller than n. It also determines the retained variability of the features, which is the ratio between the first q eigenvalues and the sum of all n eigenvalues [7].

$$\upsilon = \frac{\sum_{i=1}^{q} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \tag{4}$$

The n rows of $\mathbf{T}_q$ are vectors $\mathbf{v}_i^{\mathrm{T}}$, which represent the projection of the i-th feature of **F** onto the lower q-dimensional space. The q elements of $\mathbf{v}_i^{\mathrm{T}}$ correspond to the weights of the i-th feature of **F** on the q axes of the subspace. Thus, the n vectors $\mathbf{v}_i^{\mathrm{T}}$ can be interpreted as new feature vectors corresponding to the original ones. However, the dimension of theses vectors is lower than the dimension of the original feature vectors. Therefore, the computational time is reduced.

C. Clustering and Final Selection

Features, which are highly correlated, have similar absolute vectors $\mathbf{v}_i^{\mathrm{T}}$. In other words, two independent features will show a high divergence of their corresponding vectors. On the other side, two identical features will lead to identical vectors. Once the vectors $\mathbf{v}_i^{\mathrm{T}}$ are computed, they can be clustered to p groups, whereby p is independent of q. Because of the similarity between the features within a cluster, one can be selected and the others can be treated as redundant information. In the final step, the features are selected by engineering knowledge and thus, a group of p features will be maintained. The key question in selection is not only the selection itself but also to find the best number of features. If there are too many features selected, the remaining redundancy is high and additional features not only provide no useful information but also affect the OSA results negatively. On the other side, too few features do not provide enough information for practical OSA. However, the best number of features depends on the problem and can be determined by experience with different data sets. In this study, we show that OSA based on 5 – 50 selected input features is applicable. Therefore, we generated 6 data sets, each with 5, 10, 20, 30, 40, and 50 input features.

Finally, the data are normalized by their mean and standard deviation before used in the following OSA methods. Figure 3 shows the selected features for 5 inputs plot into the one-line diagram of the test system.

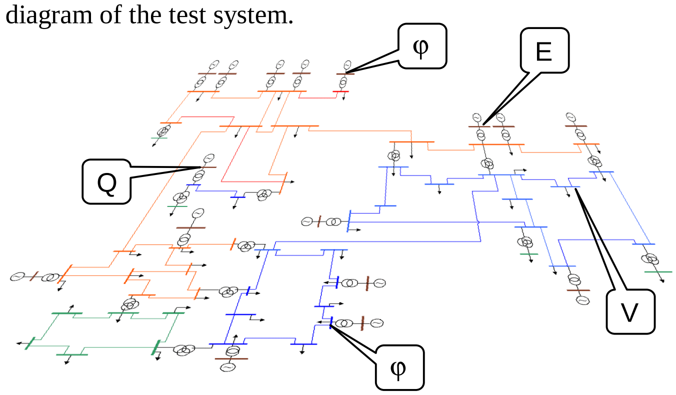

Fig. 3 One-Line Diagram of the PST 16-Machine Dynamic Test System with 5 Selected Input Features for OSA

IV. Decision Trees

Decision tree techniques belong to the AI methods and became highly popular in the age of modern computers. They base on a sequence of questions that can be answered with yes or no. Each question asks whether a predictor satisfies a given condition, whereby the predictors can be both continuous and discrete. Depending on the answer to each question, one can either proceed to another question or arrive at a response value.

A. Tree Growing

To construct a tree, the data is divided into two sets. One set is used to learn the tree and the other set is used to test it afterwards. For tree growing, there are different algorithms available depending on the kind of tree desired. Regardless of the algorithm, the first task is to find the root node for the tree. The root node is the first node splitting the entire data set into two parts. Therefore, the root must do the best job in separating the data. The initial split at the root creates two new nodes, called branch nodes. The algorithm searches at both branch nodes again for the best split to separate the sub sets, and following this procedure, the algorithm will continue to split all branch nodes until either a branch node contains only pattern of one kind or the diversity cannot be increased by splitting the node. The nodes, where the tree is not further split, are labeled as leaf nodes. When the entire tree is split until only leaf nodes remain, the full tree size is obtained.

The different algorithms depend on the criteria used to split the nodes. The separability at a branch node can be determined using the entropy model [4], whereby the feature showing the least entropy gives the most information about the separability at this branch node. The entropy bases on pattern distribution and probability. The feature, which provides most information about the separability, is used to split the branch node. If all patterns at one node belong to the same class, the entropy is zero and the node becomes a leaf node. In literature, various splitting rules are proposed and discussed [8], [9]; the most famous are the Gini's diversity index, the twoing rule, and the maximum deviance reduction.

The process of learning is fast and applied once in the beginning. When the tree is grown, it can be used to predict an output depending on the presented inputs. It has been observed, that large decision trees usually do not retain their accuracy over the whole space of instances and therefore tree pruning is highly recommended [10].

B. Tree Pruning

Pruning is the process of reducing a tree by turning some branch nodes into leaf nodes, and removing the leaf nodes under the original branch. Since less reliable branches are removed, the pruned decision tree often gives better results over the whole instance space even though it will have a higher error over the training set. To prune a tree, the training set can be split into the growing set (for learning the tree) and the pruning set (for tree pruning). Different pruning approaches use the testing data for pruning. However, pruning is necessary to improve the tree capability and reduce the error cost. Moreover, large trees are specialized on the used growing data and some lower branches might be affected by outliers.

Pruning is basically an estimation problem. The best tree size is estimated based on the error cost. When only training data are used to prune a tree, the estimation is called internal estimation. Accuracy is computed based on the misclassifications at all tree nodes. Then, the tree is pruned by computing the estimates following the bottom-up approach (post-pruning). The resubstitution estimate of the error variance for this tree and a sequence of simpler trees are computed. Because this estimation probably under-estimates the true error variance, the cross-validation estimation is computed next. The cross-validation estimate provides an estimate of the pruning level needed to achieve the best tree size. Finally, the best tree is the one that has a residual variance that is no more than one standard error above the minimum values along the cross-validation line [8].

V. OSCILLATORY STABILITY ASSESSMENT

Decision trees can be used for nonlinear regression (Regression Tree) when using continuous variables or they can be used for classification (Classification Tree) when using discrete classes [8].

This paper will introduce decision tree strategies for the oscillatory stability assessment using both tree methods.

A. Classification Tree

When the tree is constructed as a classifier, the tree output is assigned to a class. The most common classification in OSA is the decision between stable and instable situations. However, the power system is characterized by its dominant eigenvalues resulting from the system state matrix [3]. The damping ratio corresponding to one particular mode is computed as

$$\zeta = \frac{-\sigma}{\sqrt{\sigma^2 + \omega^2}} \quad (5)$$

whereby the complex pair of system eigenvalues for this mode is written in the form $\lambda = \sigma \pm j\omega$.

Thus, we can characterize the system as stable, when the damping ratio of the least damped eigenvalue for one given case is above a threshold. In other words, if at least one of the eigenvalues shows a damping below the threshold value, the system is considered as instable. The basic idea of a classification tree for the 2-class case is shown in Figure 4.

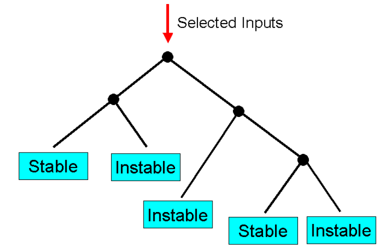

Fig. 4 Classification Tree for Oscillatory Stability Assessment

The 2-class case can be expanded to a multiple class classifier easily using damping ranges for classification. The ranges are defined first by their limits for upper and lower damping border. Then, the classifier is trained on if the pattern falls into a given range or not. The advantage is the increased resolution, which results in a more defined damping range instead of the rough division between stable and instable cases.

B. Regression Tree

Further accuracy can be obtained using regression trees instead of classification trees. As opposed to pre-defined classes, the decision tree can be learned on the exact value for the minimal damping ratios. The result can be more accurate than a class, which distinguishes only between predefined classes, but not between patterns within the same class. Figure 5 shows the basic idea of a regression tree.

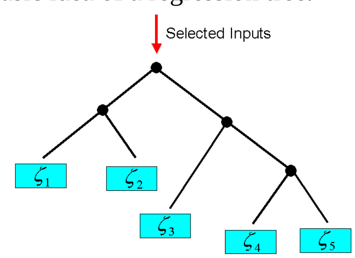

Fig. 5 Regression Tree for Oscillatory Stability Assessment

VI. RESULTS

A. Classification

In this study, we assume a system damping of larger than 4% for any system mode as sufficiently damped and therefore as stable. Conversely, if there is at least one mode with a damping ratio smaller than 4%, the system is considered to be instable. This leads to 1,468 patterns in the stable class and 3,892 patterns in the instable class. Then, the entire set is split into these two classes and the decision tree is grown. After learning the tree, it is tested with a different data set. About 90% of the data are used for growing and 10% for testing. If the tree output and the system state match each other, the classification is correct. Otherwise, the classification is false. The percentage of misclassifications for learning and testing is shown in Table II for varying number of input features between 5 – 50. Since the testing data are picked randomly from the entire set, the results vary for with each run. Therefore, the errors in Table II are averaged over 20 computations.

TABLE II
ERRORS FOR FULL AND PRUNED CLASSIFICATION TREE FOR VARYING NUMBER OF INPUTS

# of Tree Inputs	Full Tree		Pruned Tree	
	Learning	Testing	Learning	Testing
5	0.9 %	3.4 %	2.1 %	3.4 %
10	0.3 %	1.5 %	0.6 %	1.5 %
20	0.3 %	2.1 %	1.0 %	2.1 %
30	0.5 %	1.9 %	1.2 %	2.1 %
40	0.4 %	1.9 %	1.1 %	1.9 %
50	0.3 %	1.9 %	0.8 %	2.1 %

Apparently, the tree using 10 inputs leads to the best results and is therefore most suitable for OSA, but nevertheless all of them are applicable for OSA.

In the next step, the 2-class classifier is expanded to a multiple class classifier to increase the accuracy of the prediction. Hereby, the decision tree outputs are assigned to the damping ratios of the least damped eigenvalues. For all computed patterns, the minimal damping ratio is in the range between –3% and 8%. When the exact damping value is rounded to integer, the patterns can be distributed into 12 different classes of 1% damping ratio width. The distribution of the patterns over the 12 classes is shown in Figure 6.

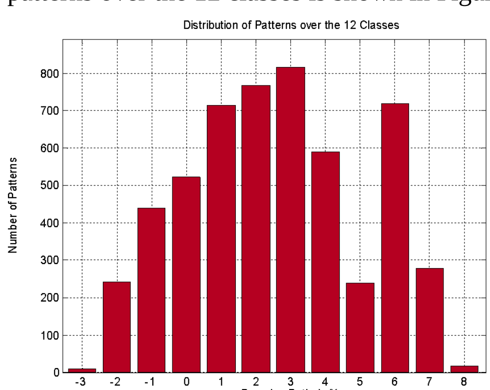

Fig. 6 Distribution of Patterns over the 12 Classes

After defining the classes, the decision tree is grown to assign each pattern to one of the classes. The error distribution for learning and testing is computed and shown in the following Table III, separately for varying input numbers and averaged over 10 computations. The errors are divided into different categories depending on how much the classifier failed. If the classifier gives a neighboring class, the prediction is still accurate enough for OSA, but when a class far away from the real one is predicted, the result is useless.

TABLE III
ERROR DISTRIBUTION FOR FULL AND PRUNED CLASSIFICATION TREE FOR VARYING NUMBER OF INPUTS

# of Inp.	Error Distribution	Full Tree		Pruned Tree	
		Learning	Testing	Learning	Testing
5	± 1 Class	3.7 %	12.8 %	6.4 %	13.3 %
	± 2 Classes	1.3 %	3.9 %	2.0 %	3.7 %
	± *n* Classes	0.5 %	1.3 %	0.8 %	1.6 %
10	± 1 Class	2.2 %	10.7 %	3.2 %	11.0 %
	± 2 Classes	0.4 %	1.0 %	0.5 %	1.1 %
	± *n* Classes	0.1 %	0.4 %	0.2 %	0.4 %
20	± 1 Class	2.2 %	11.4 %	3.5 %	11.8 %
	± 2 Classes	0.3 %	1.0 %	0.4 %	0.8 %
	± *n* Classes	0.2 %	0.8 %	0.3 %	0.8 %
30	± 1 Class	2.3 %	11.5 %	4.0 %	12.1 %
	± 2 Classes	0.4 %	0.8 %	0.5 %	0.9 %
	± *n* Classes	0.3 %	0.5 %	0.3 %	0.5 %
40	± 1 Class	2.3 %	12.2 %	4.1 %	12.4 %
	± 2 Classes	0.4 %	1.1 %	0.5 %	1.1 %
	± *n* Classes	0.3 %	0.4 %	0.4 %	0.4 %
50	± 1 Class	1.8 %	11.0 %	4.6 %	11.9 %
	± 2 Classes	0.3 %	0.6 %	0.4 %	0.7 %
	± *n* Classes	0.2 %	0.3 %	0.2 %	0.3 %

As can be seen from the errors in the Table III, most errors occur in the class next to the target class. Classes with more distance to the correct class show much lower errors. Another interesting result concerns the number of inputs, which is necessary for a good classification. The errors decrease noticeably from 5 to 10 inputs. But when more then 10 inputs are used, the errors neither show an increasing nor a decreasing behavior. They almost remain constant with a slight spread around their values. This leads to the conclusion, that 10 inputs are sufficient for classification tree application in the investigated power system, both in the multiple-class case and the 2-class case as shown before.

B. Regression

Finally, the tree is implemented as regression tree. The error is defined as the difference between the damping percentage of the tree output and the real damping value. The mean of the errors is nearly zero and therefore only the standard deviations are listed in Table IV for learning and testing under varying input numbers and averaged over 10 computations.

TABLE IV
STANDARD DEVIATIONS OF ERRORS FOR FULL AND PRUNED REGRESSION TREE FOR VARYING NUMBER OF INPUTS

# of Tree Inputs	Full Tree		Pruned Tree	
	Learning	Testing	Learning	Testing
5	0.22 %	0.52 %	0.30 %	0.54 %
10	0.13 %	0.34 %	0.18 %	0.35 %
20	0.12 %	0.35 %	0.17 %	0.37 %
30	0.10 %	0.26 %	0.15 %	0.28 %
40	0.11 %	0.33 %	0.16 %	0.35 %
50	0.10 %	0.30 %	0.16 %	0.32 %

The results listed in Table IV show small standard deviations for the difference between decision tree outputs and real damping values. The following Figure 7 shows the errors versus the learning patterns for the full regression tree based on 30 inputs. It can be noticed that the error is low and only some single outliers affect the result. Figure 8 shows the errors versus the testing patterns for the same regression tree.

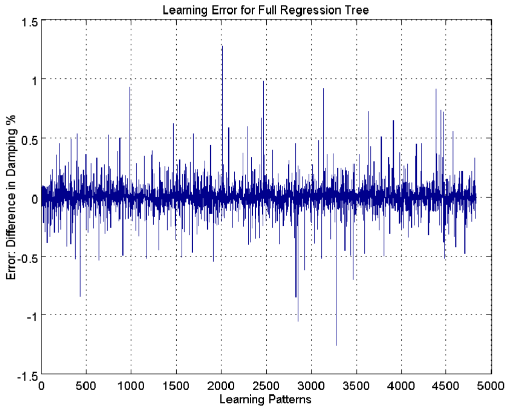

Fig. 7 Learning Error for Full Regression Tree with 30 Inputs

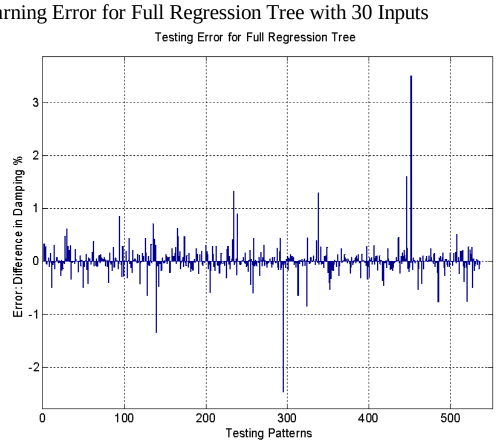

Fig. 8 Testing Error for Full Regression Tree with 30 Inputs

VII. Conclusion

This study proposes the use of decision tree for oscillatory stability assessment. The trees can be implemented easily as classifier for stability or as predictor for the system damping. Both types of tree show high accuracy when tested by a set of independent testing data. Decision trees can be constructed in a short period of time and used on-line since the tree evaluation does not require any time-consuming computation.

Moreover, the trees are robust and can be used with only a small number of input features. This study points out, that only 10 inputs are sufficient for accurate OSA. Decision trees as introduced in this study lack only the fact that they don't provide information about the number of dominant eigenvalues and their exact position in the complex plain. The output is limited to a classification result or the minimum damping for the actual system state. However, in oscillatory stability assessment, the system operator needs intelligent tools that support fast decisions he has to make and not detailed information about critical eigenvalues. Therefore, the proposed methods are very useful for modern power system management and operation.

VIII. References

[1] U. Bachmann, I. Erlich and E. Grebe, "Analysis of interarea oscillations in the European electric power system in synchronous parallel operation with the Central-European networks", IEEE PowerTech, Budapest 1999

[2] H. Breulmann, E. Grebe, et al., "Analysis and Damping of Inter-Area Oscillations in the UCTE/CENTREL Power System", CIGRE 38-113, Session 2000

[3] P. Kundur, Power System Stability and Control, McGraw-Hill, New York, 1994

[4] L.A. Wehenkel, Automatic Learning Techniques in Power Systems, Kluwer Academic Publishers, Boston, 1998

[5] S.P. Teeuwsen, I. Erlich, M.A. El-Sharkawi, "Neural Network based Classification Method for Small-Signal Stability Assessment", IEEE PowerTech, Bologna, Italy, June, 2003

[6] M. Mendel, K.S. Fu, Adaptive, learning and pattern recognition systems: theory and applications, New York, Academic Press, 1972

[7] I.T. Jolliffe, Principal Component Analysis, Springer-Verlag, New York, 1986

[8] Breiman, et al., Classification and Regression Trees, Wadsworth Int., Calif., 1984

[9] J.R. Quinlan, C4.5 Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, Calif., 1993

[10] H. Kim, G.J. Koehler, "Theory and Practice of Decision Tree Induction", Omega, Vol. 23, No. 6, pp.637-652, 1995

IX. Biographies

Simon P. Teeuwsen (1976) is presently PhD student in the Department of Electrical Power Systems at the University of Duisburg-Essen/Germany. He started his studies at the University of Duisburg in 1995. In 2000, he went as exchange student to the Unversity of Washington, Seattle, where he performed his Diploma Thesis. After his return to Germany in 2001, he received his Dipl.-Ing. degree at the University of Duisburg. He is a member of VDE, VDI, and IEEE.

Istvan Erlich (1953) received his Dipl.-Ing. degree in electrical engineering from the University of Dresden/Germany in 1976. After his studies, he worked in Hungary in the field of electrical distribution networks. From 1979 to 1991, he joined the Department of Electrical Power Systems of the University of Dresden again, where he received his PhD degree in 1983. In the period of 1991 to 1998, he worked with the consulting company EAB in Berlin and the Fraunhofer Institute IITB Dresden respectively. During this time, he also had a teaching assignment at the University of Dresden. Since 1998, he is Professor and head of the Institute of Electrical Power Systems at the University of Duisburg-Essen/Germany. His major scientific interest is focused on power system stability and control, modelling and simulation of power system dynamics including intelligent system applications. He is a member of VDE and IEEE.

Mohamed A. El-Sharkawi received the B.Sc. degree in electrical engineering in 1971 from Cairo High Institute of Technology, Egypt, and the M.A.Sc. and Ph.D. degrees in electrical engineering from the University of British Columbia, Vancouver, B.C., Canada, in 1977 and 1980, respectively. In 1980, he joined the University of Washington, Seattle, as a Faculty Member. He served as the Chairman of Graduate Studies and Research and is presently a Professor of Electrical Engineering. He is the Vice President for Technical Activities of the Neural Networks Society. He organized and taught several international tutorials on intelligent systems applications, power quality and power systems, and he organized and chaired numerous sessions in IEEE and other international conferences. He is a member of the editorial board and Associate Editor of several journals, including the IEEE TRANSACTIONS ON NEURAL NETWORKS.