FUZZY CONTROL OF STATE ESTIMATION ROBUSTNESS

Jorge Pereira
INESC Porto and FEP
jpereira@inescporto.pt

Vladimiro Miranda
INESC Porto and FEUP
vmiranda@inescporto.pt

J. Tomé Saraiva
INESC Porto and FEUP
jsaraiva@inescporto.pt

INESC Porto- Instituto de Engenharia de Sistemas e Computadores do Porto, P. República 93, 4050-497 Porto, PORTUGAL
FEUP – Faculty of Engineering of University of Porto
FEP - Faculty of Economics of University of Porto

Abstract – This paper reports the main results from the application of Fuzzy Inference Systems – FIS - to tackle the problem of selecting the most adequate set of weights to use in State Estimation algorithms directed to distribution networks. These networks have distinctive characteristics regarding transmission ones turning the migration of software packages from EMS to DMS systems not immediate. In previous papers, the authors described a Fuzzy State Estimation algorithm that has the flexibility to adequately address these problems. However, this algorithm requires fine tuning of several weights. The authors solved this problem by training a FIS system using a set of rules derived from a large number of exercises run for small networks. This approach, as it will be illustrated in the paper, proved to be very successful in the sense that the FIS displayed a remarkable capacity of generalization since it extremely improved the convergence pattern of the Fuzzy State Estimation algorithm when analyzing larger networks, differently from the ones used to build the training set.

Keywords: State Estimation, Fuzzy Control, Fuzzy Inference, System Operation.

1 INTRODUCTION

The development of a State Estimation approach especially shaped for distribution networks demanded new models and techniques, to take in account the characteristics of such systems that cannot be found in classical transmission systems. Among such characteristics, one must rank: the reduced number of telemetered measurements, the existence of switching as major decision variables and the increasing importance of distributed generation.

Recently, the authors have reported the development and practical implementation in utilities of a new model [1]-[2], called Fuzzy State Estimation (FSE) that is able to represent all those characteristics. In particular, in the absence of measurements, one has adopted a fuzzy model for loads and injections, and one has represented the classical binary variables associated to switches by continuous functions whose error must be minimized. It was also developed an approach aiming at allowing the representation of islanding, with local distributed generators assuring the energization of segments of a network disconnected from the main system.

The results of a FSE computer implementation including all these features have been extremely good. However, the convergence of the method depended on the fine tuning of a set of weights or parameters, a task that had to be done on a case-by-case basis and by trial an error, for each system.

This paper presents an elegant solution for the tuning of the algorithm. The trial-and-error method is replaced by the use of a set of fuzzy controllers composing a Fuzzy Inference System (FIS), whose output are the necessary weights to assure a good convergence of the State Estimation algorithm. The FIS is composed of sets of Fuzzy Rules, which may have a linguistic translation in such terms that make the rules understandable by the users. This model has been developed to allow the simultaneous determination of the correct topology of a network and the calculation of the most fit set of values for state variables, minimizing errors in measurements.

The paper shows that both Mamdani and Takagi-Sugeno controllers [3]-[4], work well, for this purpose. The success of this approach derives from the fact that it was possible to generate a training set having as basis a small network. For a small network, it was possible to define a coherent set of cases with enough diversity, a task almost impossible for a large network, or for the network that one wanted to analyze. This definitely was a task to avoid, since a major objective was to build a learning set and to conduct the training process of the FIS independently of each specific network to analyze. Another important advantage of this approach corresponded to the possibility of running the analysis of a small network for a large number of cases, since computation time was reduced in this case. This enabled building a training set very rich in terms of the diversity of situations – topology, generations, loads – that were considered.

With trained controllers, one has then been able to successfully run the State Estimation procedure for large systems, without having any concern for tuning weights or parameters. Having learned rules from a small system, the fuzzy inference controllers were able to generalize and determine the right set of weights that assured the global algorithm convergence.

This is, in a way, a remarkable result with great industrial interest: a numerical algorithm (such as the State Estimation) has been tuned to convergence by a fuzzy inference system – or, better said, one was able to give robustness to numerical algorithms (of the Newton type) by using a fuzzy control of algorithm parameters. The interest of this approach goes in fact beyond state estimation.

Apart from this initial Introductory Section, Section 2 describes the Fuzzy State Estimation algorithm including some of the recent advances as the treatment of topology problems and the possibility of system splitting. Section 3 gives some insight to the main theoretical concepts about Fuzzy Inference Systems. Section 4 details the process adopted to built the FIS recalling that it was possible to obtain a sufficiently general FIS in the sense it was successively applied to large systems although it was trained considering results from small systems. Section 5 includes results obtained with the application of the whole developed methodology – issues as quality of the convergence for several topologic structures and several sets of input data - to large systems and in Section 6 the main conclusions will be drawn.

2 FUZZY STATE ESTIMATION

2.1 Traditional State Estimation

State Estimation became a traditional software package in Energy Management System, EMS. This module has an essential role in achieving a coherent set of voltage magnitudes and phases, as well as any other value computed from them, considering that there is a number of telemetered values, eventually affected by errors, read in a widespread way all through the network. In this sense, State Estimation aims at identifying the values of a set of state variables that most adequately fit to the telemetered available data.

The most well-known State Estimation model considers that each measurement Z can be expressed in terms of state variables X using a function h(X) and will generally be affected by an error ε (1). The State Estimation formulation aims at identifying the set of X – traditionally voltage magnitudes and phases – so that the sum of the weighted squares of the errors is minimized (2). In this approach one considers that the errors are represented by random variables having zero mean, variance σ_i^2 and that they are not correlated. This leads to a diagonal co-variance matrix R that is traditionally used in (2) to weight the errors.

$$Z = h(X) + \varepsilon \qquad (1)$$

$$\min\ \varepsilon^T R^{-1} \varepsilon \qquad (2)$$

After substituting ε obtained from (1) in (2) one obtains (3). The optimization of this non-linear continuous function – provided topology, transformer taps and capacitor sections are considered fix – can be achieved by formulating the stationary optimality conditions. These can be expressed by (4) in terms of a set of non-linear equations function of the state variables X. In (4) H is the Jacobean matrix of the measurement vector h.

$$\min\ [Z - h(X)]^T R^{-1} [Z - h(X)] \qquad (3)$$

$$H(X)^T R^{-1} [Z - h(X)] = 0 \qquad (4)$$

The set of non-linear equations (4) can be solved by an iterative approach as the Newton Raphson. In this approach the solution in iteration k+1 is given by (5) and is obtained from the solution in iteration k plus a deviation namely depending on the gain matrix G (6). This matrix must be computed and inverted in each iteration of the Newton Raphson algorithm and is responsible by itself for a significant part of the computation time.

$$X^{k+1} = X^k + \left(G^k\right)^{-1} \left[H\left(X^k\right)\right]^T R^{-1} \left[Z - h\left(X^k\right)\right] \qquad (5)$$

$$G^k = \left[H\left(X^k\right)\right]^T R^{-1} \left[H\left(X^k\right)\right] \qquad (6)$$

After getting the set of voltage magnitudes and phases that better explain the available measurements, one can compute all other variables as power flows and currents, losses and nodal injections.

This traditional approach relies on three basic assumptions:

- it assumes that the number of telemetered measurements available in real time in the Control Centers is large enough not only to enable running the algorithm but also to achieve redundancy in terms of reducing the impact of eventual large errors;
- it considers that the topology of the network is known beyond any doubt and is therefore an input to the problem;
- following the previous point, it assumes that the network in operation remains connected. This means that splitting is not considered, even if it may lead to a configuration more adapted to the available set of measurements.

These basic assumptions are not always valid in distribution networks. To cope with these problems the authors developed an integrated approach [1]-[2] flexible enough to integrate fuzzy assessments for nodal powers, to deal with topologic variables in a novel and efficient way and to allow system splitting if that is more adequate given the set of measurements. These three issues will be addressed in the next three sub-sections.

2.2 Fuzzy State Estimation

Fuzzy State Estimation admits that, at least, one input value is represented by a fuzzy number, as the triangular one depicted in Figure 1. Such a concept can be used to translate a natural language assessment such as:
"*The load in node k will most likely be 3MW but values from 3 to 2 and from 3 to 4 are still possible*".

In this sense, 3 is assigned a maximum compatibility degree with this assessment – membership degree 1.0 – while values from 3 to 2 and from 3 to 4 are assigned decreasing compatibility degrees from 1.0 to 0.0.

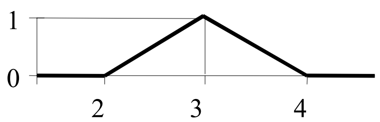

Figure 1 : Membership function of a triangular fuzzy number.

Having in mind this concept, the traditional State Estimation algorithm was enlarged to accommodate fuzzy indications for nodal powers or voltages and it is fully detailed in [1]-[2]. Briefly, it works as follows:

i) Build the fuzzy vector of measurements $\tilde{Z}$;
ii) Build the vector of central values of the measurements, *Z*. The central value of a fuzzy number is the average of the values having membership degree 1.0;
iii) Run a crisp traditional State Estimation study according to the formulation in sub-section 2.1. The output is the state vector X_1 ;
iv) Get the fuzzy deviations between $\tilde{Z}$ and *Z* by performing the fuzzy operation (7).

$$\Delta\tilde{Z} = \tilde{Z} - h(X_1) \quad (7)$$

v) Obtain the fuzzy deviations of the state variables using (8). Add these fuzzy deviations to the crisp values of state variables obtained in iii) according to (9).

$$\Delta\tilde{X} = (G^{-1}H^T R^{-1})\Delta\tilde{Z} \quad (8)$$

$$\tilde{X} = X_1 + \Delta\tilde{X} \quad (9)$$

vi) For all other variables (power flows, currents, losses, nodal injections), consider a non linear function depending on voltage magnitudes and phases as, for instance (10) for a line flow in branch ij. In this expression J_{FL} integrates the linear terms of the Taylor Series development of F_{ij} expressed in terms of voltage magnitudes and phases, that is terms of state variables-

$$\Delta F_{ij} = J_{FL}\Delta X \quad (10)$$

These in turn depend on the deviations on measurements through expression (11). This comes (5) recognizing that (11) multiplied by $Z - h(X)$ gives the deviation of state variables. This way it is possible to obtain expressions directly relating an active power flow F_{ij} , for instance, with input data (12).

$$\Delta X = (G)^{-1}[H(X)]^T R^{-1}\Delta Z \quad (11)$$

$$\Delta\tilde{F}_{ij} = \left[J_{FL}(X_1)\left(G^{-1}H^T R^{-1}\right)\right]\Delta\tilde{Z} \quad (12)$$

vii)The final value of $\tilde{F}_{ij}$ is given by the fuzzy addition of $\Delta\tilde{F}_{ij}$ with the crisp value F_{ij} obtained in iii).

2.3 Topological variables

Topology issues can be represented by 0-1 variables leading to non-continuous formulations. Several techniques have been developed within State Estimation problems to avoid the combinatorial problems arising from this binary nature. References [5]-[6] are examples of models of this type. In our formulation, we addressed this problem by including topology real valued variables having the particular feature of being constrained to a binary behavior. Assuming that d_{ij} is the variable representing the status of a branch of a switching device, we will include it in the set of state variables, and d_{ij} will be constrained by an equation as (13). This equation has two real roots (0 and 1) thus allowing enforcing the binary result of these state variables while not loosing the continuity of the formulation.

$$x^2 - x = 0 \quad (13)$$

In practice, a value can be available for d_{ij} in the database of the SCADA system. In this case, if certainty is not absolute about this value, expression (14) shall be considered indicating that the final output d_{ij} can be different from the input d_{ij}^{mes} if that is more adequate to explain the whole set of measurements. If no data is available for d_{ij} in the database, expression (15) is used.

$$d_{ij}^{mes} = d_{ij}^2 + \varepsilon_k \quad (14)$$

$$0 = d_{ij} - d_{ij}^2 + \varepsilon_k \quad (15)$$

The expressions of the flows in branches affected by this kind of uncertainty must include d_{ij} to ensure the coherency of the final results as in (16) for the active flow from node i to node j.

$$P_{ij} = \left[\left(g_{ij} + \frac{g_{shij}}{2}\right)V_i^2 - V_i V_j\left(g_{ij}\cos(\theta_{ij}) + b_{ij}\sin(\theta_{ij})\right)\right]d_{ij} \quad (16)$$

2.4 System splitting

Splitting is another major feature designed to turn the model more flexible and to reduce the gap between mathematic formulations and reality. In fact, uncertainties regarding the topology in operation can be adequately addressed using the ideas in 2.3 provided that the network remains connected in a single island. In this sense, changes in the status of a switching device are not yet admitted if they ultimately lead to network islanding. Islanding requires the possibility of defining a phase reference in each possible island of the network. The number and the constitution of the islands are not known a priori but the formulation should have the flexibility to adapt itself along the algorithm, namely taking into account the set of available measurements. This means that islanding should be interpreted as an additional resource of the State Estimation algorithm to provide an output that is more adequate and better explains the available measurements.

From a mathematical point of view splitting was not considered because a single phase reference was selected all along the network. Thus, if splitting was considered, some matrixes would become singular.

In our formulation the ability to dynamically consider a phase reference for each island was included by admitting a phase pseudo measurement on all generation nodes, to which it was assigned the value of 0 (17).

$$\theta_i = 0 + \varepsilon_k \qquad (17)$$

In the beginning of the algorithm, a large weight is typically assigned to the phase measurement in one of the generation nodes, to the largest generation, for instance. Smaller weights will be assigned to all other phase measurements. This means that the algorithm may use this possibility but it will preferably try to compute the state variables for a single island configuration. This comes from the fact that there is a larger weight assigned to a particular reference thus having a larger influence in the final result. In this case, changes in the phases in the remaining generation nodes will lead to residuals that have a minor impact on the final result since their weights are reduced. However, if the available measurements are more adequately explained by admitting splitting, the algorithm can use a phase reference in each island, provided there is at least one generator in each of them.

3 FUZZY INFERENCE SYSTEMS

Instead of using a trial-and-error approach or trying to adjust parameters for each different particular system, we used Fuzzy Inference Systems (FIS) to define a set of weights leading to good algorithm performance and convergence. The main problem was to correctly identify a set of *influence factors* that would, when input to the FIS, generate correct responses. As we shall see, this identification was successful.

A FIS can be defined as a system that transforms or maps fuzzy or crisp values to another collection of fuzzy or crisp values. At its core there is a Rule Set of fuzzy rules and an Inference Mechanism, which calculates the level to which rule is activated for a given input pattern.

The most important FIS are of the Mamdani type (M-FIS) [3] and of the Takagi-Sugeno type (TS-FIS) [4]. Both types of FIS can be represented in a neuro--fuzzy form, and learning procedures may be used to set their parameters. We used an ANFIS (Adaptive Neuro-Fuzzy Inference System) [7] to train the TS-FIS and NEFPROX (Neuro-Fuzzy Function Approximation) [8] to train the M-FIS.

Let us consider a generic inference system with *n* input variables and one output. In the case of the M-FIS type the rules are of the type (18).

$$\textit{if } \mathbf{x_1} \textbf{ is } \mathbf{A_{1j}} \textit{ and } \ldots \textit{ and } \mathbf{x_n} \textbf{ is } \mathbf{A_{nj}} \textit{ then } \mathbf{y} \textbf{ is } \mathbf{B_j} \qquad (18)$$

In rule (18) $(x_1,\ldots,x_n)$ is the vector of input variables, y is the output variable and A_{ij} and B_j are fuzzy numbers. These fuzzy numbers are represented by membership functions and are associated to the input variables x_i and to the output variable y in rule j

The fuzzy numbers *A* and *B* can be associated with linguistic labels. The final output of the controller is obtained by the defuzzification of the result or it is obtained with an ***or*** operation over the rules results. To do this a number of methods can be used in practice, namely the Center-of-Gravity or the Mean-of-Maxima (see reference [9]).

Training a M-FIS using the NEFPROX means finding the parameters of the membership functions related with each one of the fuzzy numbers A_{ij} and B_j and the rules that must be considered by the FIS.

On the other hand, in the case of a 1st-order TS-FIS the rules are of the type (19). In this case, the output of each rule y_j is a crisp value evaluated using a linear combination of the input values with coefficients a_{ij}.

$$\begin{gathered}\textit{if } \mathbf{x_1} \textbf{ is } \mathbf{A_{1j}} \textit{ and } \ldots \textit{ and } \mathbf{x_n} \textbf{ is } \mathbf{A_{nj}} \textit{ then} \\ \mathbf{y_j = a_{0j} + a_{1j}x_1 + \ldots + a_{nj}x_n}\end{gathered} \qquad (19)$$

The output y of the TS-FIS is obtained by the sum of individual rule's outputs, weighted by rule firing strengths g_j (20), which result from an ***and*** operation on rule's outputs. In expression (20) n_a represents the number of rules activated by the input values.

$$y = \frac{\sum_{j=1}^{n_a} g_j y_j}{\sum_{j=1}^{n_a} g_j} \qquad (20)$$

Training a M-FIS using the ANFIS corresponds to find the parameters of the membership functions related to each one of the fuzzy numbers A_{ij}, the rules that must be considered by the FIS and the coefficients a_{ij} of the output values of each rule.

For ***and*** operations, T-Norms should be used. In M-FIS it is usual to adopt the *min* or the *product* operator while in TS-FIS, the most usual T-Norm used is *product*. If the T-Norm *product* is used in TS-FIS the firing strength g_j for rule j is given by (21), where $f_{A_{ij}}(x_i)$ is the membership function of the input fuzzy number A_{ij} evaluated for the input value x_i.

$$g_j = f_{A_{1j}}(x_1) \times f_{A_{2j}}(x_2) \times \cdots \times f_{A_{nj}}(x_n) \qquad (21)$$

For ***or*** operations, T-Conorms should be used. The most widely used T-Conorm in M-FIS is the *Max* operator.

4 TUNNING THE WEIGHTS WITH FIS

4.1 General aspects

The FIS concepts briefly presented in the previous section will now be applied to the FSE problem in order to find a set of weights to be assigned to topological variables. In this section we will detail the set of *influence factors* used as input and the procedure that was adopted to obtain the FIS parameters.

4.2 Inputs and outputs

As we have seen, the inputs to the FIS may be interpreted as representing the *influence factors* that condition FIS response. We have selected the following influence factors, associated only with nodes and branches but also having some relation with topological variables (representing switching devices or branches):

- Connectivity of the branch (CNB) - ratio between branch conductance and the average of conductances in the two extreme nodes of the branch (values taken from the bus admittance matrix);
- Physical characteristic of the branch (PCB) - the symmetrical of the ratio of branch conductance over branch susceptance;
- Voltage deviation of the area (VDA) - average of voltage magnitudes in the two extreme nodes of the branch;
- Load Level of branch nodes (LLB) - average of active power consumption in the two extreme nodes of the branch;
- Significance of the load (SGL) - ratio between the LLB variable and the total active power consumption of the possible network island where the branch is located.

Each of these variables was assigned 5 linguistic levels - Very Small, Small, Medium, Large and Very Large - represented each one by a Gaussian-type membership function (22) with μ median and σ spread. The median parameter is the point for which the membership function has the maximum value (equal to 1.0) and the spread is a positive value related with the width of the membership function. As an example, Figure 2 represents the membership functions of the five linguistic levels that can be assigned to variable CNB. In the case of linguistic level Medium the parameters μ and σ the values 0.504845 and 0.243606, respectively.

$$f_{A_{ij}}(x_i) = e^{-\left(\frac{x_i - \mu}{\sigma}\right)^2} \qquad (22)$$

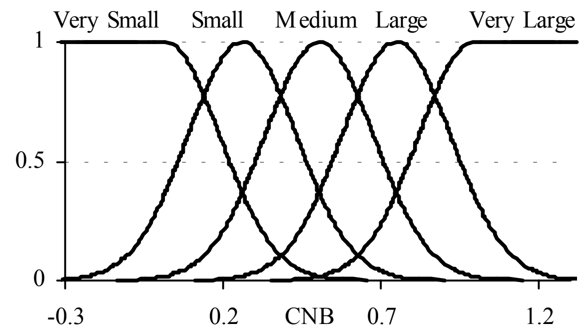

Figure 2 : Membership values of CNB.

The output of the M-FIS is defined by 7 levels, represented by Gaussian-type functions defined as a function of $\log_{10}$(weight). The output of the TS-FIS was also defined as the $\log_{10}$(weight). The FIS is then applied to find the most convenient weight to be associated to each topological variable (see Figure 3).

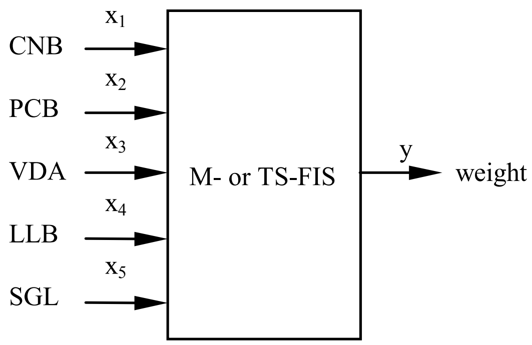

Figure 3 : FIS generating the weight to be assigned to a topological variable

4.3 Training the FIS

To build a training set for the FIS, we used a small network considering realistic data, and then we generated a large number of different and significative cases by changing loads, characteristics of lines, voltage level in the reference buses, and changes in the topology in operation. For each case, we ran seven Fuzzy State Estimation procedures. In each one, we fixed the weight value of the topological variables. The weights were fixed in the set {-6,-5,-4,-3,-2,-1,0} defining the $\log_{10}$(weight). From each of these seven runs, we selected the case leading to the best performance of the state estimation of the algorithm and we included it in the training set. At the end of this procedure, we built a training set incorporating 14993 points, each of them defined by 5 values for input variables (influence factors) and by a value for the output variable (weight).

Using this training set, we proceeded to train a M-FIS (with 256 rules) and a TS-FIS (also with 256 rules), with success. In both cases, the training process was conducted aiming at minimizing the mean square error in the training set. As an illustration, in (23) it is represented a rule obtained for the M-FIS and in (24) it is represented a rule obtained for the TS-FIS.

if ***x_1(CNB) is Small*** *and* ***x_2(PCB) is Medium*** *and* ***x_3(VDA) is Large*** *and* ***x_4(LLB) is Small*** *and* ***x_5(SGL) is Medium*** *then* ***weight is 10^{-4}*** (23)

if ***x_1(CNB) is Very Large*** *and* ***x_2(PCB) is Large*** *and* ***x_3(VDA) is Small*** *and* ***x_4(LLB) is Large*** *and* ***x_5(SGL) is Medium*** *then* (24)

$$y = -0.916422 - 0.013741x_1 - 0.036931x_2 - 0.01297x_3 + 0.028395x_4 - 0.038957x_5$$

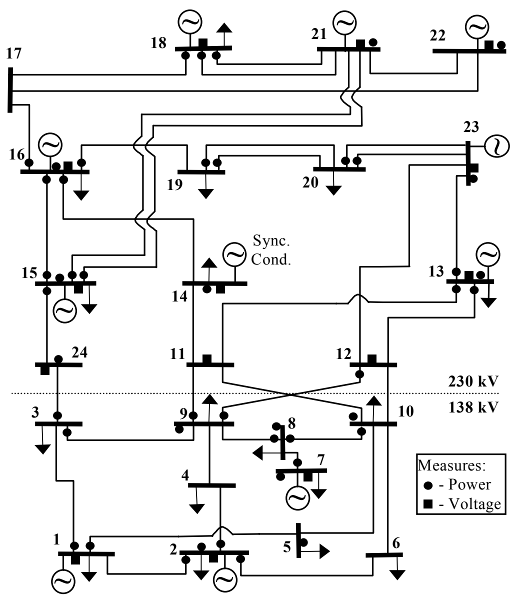

Figure 4 : IEEE 24 bus Test System.

5 APPLICATION TO LARGE SYSTEMS

5.1 Network data

The IEEE 24 bus system was used to evaluate the performance of the described approach. The topology of the system and the location of the measurements are detailed in Figure 4. System data is detailed [10]-[11]. Reference [11] also includes the description of the system with all substations modeled at the bus section level. We focused our attention on the substations related to buses 14, 15, 16 and 24 of the original system in Figure 4. Figure 5 and Figure 6 show the detailed representation at the section level of the substation for buses 14 and 15 and for buses 16 and 24, respectively.

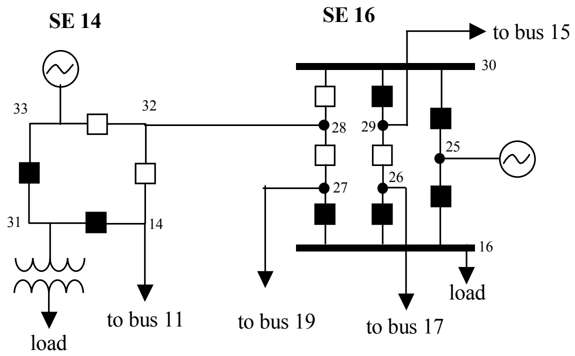

Figure 5 : Substations 14 and 16 modeled at section level.

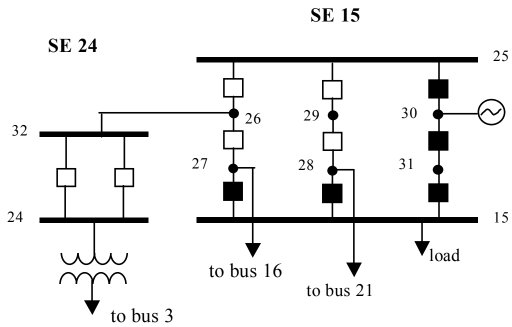

Figure 6 : Substations 15 and 24 modeled at section level.

5.2 Simulations

Different types of topology errors were simulated:

- Inclusion errors, which occur when the status of switching devices responsible for connecting / disconnecting a line to the system are reported to the estimator as closed while they are actually open (that is, the line is erroneously considered to be in operation in the system model);
- Exclusion errors, when the line is erroneously considered to be disconnected in the system model, which occur when the respective switching devices are reported as opened while in fact they are closed.

Table 1 presents the number of iterations of the State Estimation algorithm for several situations in which the status of some devices are suspect. In the first place, and for comparison purposes, it should be said that the algorithm takes 4 iterations to converge if no topology errors are present in the input data. Regarding the results in Table 1, when topological errors are considered the number of iterations performed by the unique state estimation run in each case increases. For the two analyzed substations, the Inclusion type errors are the most difficult to address. For this network, the Inclusion errors lead to several alternative connections turning it more difficult to get convergence.

Substation	Error type	Switching device suspected	Number iterations
(Without topological errors)			4
14 - 16	Inclusion	28-30; 14-32	8
	Exclusion	28-30; 14-32	4
15 - 24	Inclusion	25-26; 24-32	10
	Exclusion	25-26; 24-32	4

Table 1 : Number of iterations in each simulated case

In the two cases, a unique suspected branch replaces the two suspected switching devices, because these two switching devices make the electrical connection of branch to the other network equipment. If the switching devices are closed the branch is in service and it has power flow, otherwise the branch is out of service.

Table 2 shows the weights associated to each one of these four State Estimation problems. These weights are obtained using TS-FIS and the input variables have the values related with the respective branch. This is the branch between nodes 28 and 32 (see Figure 5) in the first case and the branch between nodes 26 and 32 (see Figure 6) in the other case.

Substation	Branch suspected	Error type	Weight
14 - 16	Between nodes 28 and 32	Inclusion	0.00009486
		Exclusion	0.00016846
15 - 24	Between nodes 26 and 32	Inclusion	0.00016630
		Exclusion	0.00030740

Table 2 : Weights associated to each topological variable

The influence of the weights assigned to the topological variables is large. As an example, in the case of the inclusion error of the substation 14 - 26:

- If the weight is 1.0 or 0.1, the FSE algorithm doesn't identify the topological error;
- If the weight is 0.01, the FSE algorithm identifies the topological error in 11 iterations;
- If the weight is 0.001, the FSE algorithm identifies the topological error in 10 iterations;
- If the weight is 0.00009486, as indicated in Table 2, the algorithm takes 8 to converge.

6 CONCLUSIONS

In this paper we reported an important set of results in terms of controlling the convergence and the performance of a numerical iterative procedure using a set of weights obtained off-line. The success and potential applicability of our approach in DMS systems derives from the fact that we were able to generate a training set having as basis a small network. For a small network, it is possible to define a coherent but diversified set of cases, a task almost impossible for a large network. This feature is specially important because an adequate set of weights can be used not only to deal with traditional State Estimation algorithms, but also to ensure the identification of the correct topology in operation. Both types of tested FIS – Mamdani and Takagi-Sugeno - have exhibited a remarkable generalization capacity, in terms of running State Estimation for large networks. This turns it not necessary to train the FIS for each specific large network to be analyzed.

The complete procedure has, in our view, a large potential since it covers several hot issues in distributions networks as the lack of measurements, the integration of fuzzy assessments, topology problems and the possibility of splitting. Therefore, we believe it can be widely used in DMS systems contributing to reduce the gap that often exists between reality and software models.

REFERENCES

[1] Jorge Pereira, J. Tomé Saraiva, V. Miranda, "Combining Fuzzy and Probabilistic Data in Power System State Estimation", in *Proceedings of PMAPS´97 - Probabilistic Methods Applied to Power Systems*, pp. 151-157, Vancouver, B.C., Canada, Sep. 1997.

[2] Jorge Pereira, V. Miranda, J. Tomé Saraiva, "A Comprehensive State Estimation Approach for EMS/DMS Applications", in *Proceedings of IEEE Budapest PowerTech*, paper BPT99-358-16, Budapest, Aug. 1999.

[3] E.H. Mamdani, S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller", International Journal of Man-Machine Studies, Vol.7, pp. 1-13, 1975.

[4] T. Takagi, M. Sugeno, "Fuzzy Identification of Systems and Its Application to Modeling and Control", *IEEE Transactions on Systems, Man and Cybernetics*, Vol.15, pp. 116-132, Jan 1985.

[5] A. Monticelli, "Modeling Circuit Breakers in Weighted Least Squares State Estimation", *IEEE Transactions on Power Systems*, Vol.8, No.3, pp. 1143-1149, Aug. 1993.

[6] A. Abur, H. Kim, M. K. Çelik, "Identifying the Unknown Circuit Breaker Statuses in Power Networks", *IEEE Transactions on Power Systems*, Vol.10, No.4, pp. 2029-2037, Nov. 1995.

[7] J.-S. Roger Jang, "ANFIS: Adaptive-network-based fuzzy inference systems", *IEEE Transactions on Systems, Man and Cybernetics*, Vol.23, pp. 665-685, May/June 1993.

[8] D. Nauck, R. Kruse, "Neuro-Fuzzy System for Function Approximation", *Fuzzy Sets and Systems*, Elsevier Science, Vol.101, No.2, pp. 261-171, Jan. 1999.

[9] G. J. Klir, B. Yuan, *Fuzzy Sets and Fuzzy Logic - Theory and Applications*, Prentice Hall, ISBN 0-13-101171-5.

[10] IEEE RTS Task Force of APM Subcommittee, "IEEE Reliability Test System", *IEEE PAS*, Vol.98, No.6, pp 2047-2054, Nov/Dec. 1979.

[11] R. Billinton, P.K. Vohra and Sudhir Kumar, "Effect of Station Originated Outages in a Composite System Adequacy Evaluation of the IEEE Reliability Test System", *IEEE PAS*, Vol.104, No.10, pp 2649-2656, Oct. 1985.