Training a FIS with EPSO under an Entropy Criterion for Wind Power prediction

V. Miranda, *Fellow IEEE*, C. Cerqueira and C. Monteiro

***Abstract*—This paper summarizes efforts in understanding the possible application of Information Theoretic Learning Principles to Power Systems. It presents the application of Renyi's Entropy combined with Parzen windows as a measure of information content of the error distribution in model parameter estimation in supervised learning. It illustrates the concept with an application to the prediction of power generated in a wind park, made by Takagi-Sugeno Fuzzy Inference Systems, whose parameters are discovered with an EPSO – Evolutionary Particle Swarm Optimization algorithm.**

***Index Terms*—Information theoretic learning, fuzzy inference systems, power systems.**

I. Introduction

This paper reports research on the application of Information Theoretic Learning (ITL) to the development of Takagi-Sugeno Fuzzy Inference Systems, and its application in the prediction of wind power from wind parks.

ITL [1][2] is a recent approach to modeling linear or non linear mappers that associate input data with output. A mapper is taken as a function whose analytic form is unknown, and the objective is to use data to discover a set of parameters or weights W that build the adequate input-output transfer function. This discovery is classically made by training – classical mappers may be Artificial Neural Networks or Fuzzy Inference Systems. In particular, we will be interested in the latter in this paper and, in particular, in supervised training.

ITL addresses the problem of extracting information directly from data. The basic objective is to find a model that may allow the maximal amount of information to be extracted from known input-output data and used in setting the weights W. Because information is the issue, ITL focuses on the evaluation of information Entropy, when assessing the equivalence of information between the desired response and the output of a mapper.

The most widely adopted measure of this "target vs. output" equivalence is correlation, in many cases represented by MSE – the Minimum Square Error criterion. It seems so "natural" that in many cases it has been adopted without challenge. However, it is a criterion related only with the second order moment of the distribution of errors (variance) and does not take advantage higher order moments. It is known that only Gaussian distributions contain all information in the first two moments of the distribution. If a distribution (of errors) is not Gaussian, training a system by optimizing variance neglects information contained in higher order moments.

The application of Information Entropy and Entropy optimization concepts has been tried in many areas related with machine learning. However, in many cases there principles have been applied using mostly Gaussian assumptions for the data distribution, which may be far from represented in real data and not correct when adapting nonlinear systems. Most of all, Shannon Entropy is the usual Entropy measure used and it presents practical difficulties. Instead, ITL uses Renyi's Entropy definition combined with Parzen windows (to estimate the pdf – probability density function of data) in a manageable procedure.

This paper presents the concepts of ITL and illustrates its usefulness with an application to wind park generation prediction, based on average wind speed and direction. Other factors can influence the power output of a wind park such as air density and turbulence intensity but speed and direction have been recognized as the main explanatory variables and those than can be more easily measured.

Wind power prediction for a wind park is more difficult than just wind prediction because:

a) A wind park is a geographically distributed structure.
b) Wind speed may vary from generator to generator.
c) Tail or shadow effects that reduce the energy of the wind behind a turbine become more or less important depending on the layout of the park and on the direction of wind.
d) A complicated terrain produces unexpected effects.
e) The non-linear characteristic of the curve of power vs. wind speed of generators adds further complexity to the problem.

The prediction of power output from a wind park is a necessary phase in methods of wind forecasting that rely on a wind forecast as an intermediate step. This is the case, for instance, of methods like *Prediktor*[3] – that has at the origin a model of fluid dynamics equations, and converts it to wind as seen by a wind park and then derives power from theoretical power curves – but also of *eWind*[4]. But it is also the case for pure statistical methods [5] and for methods based on computational intelligence techniques[6]. This prediction is highly important presently in Europe, where the growing penetration of wind generation will reach heavy percentages

V. Miranda is with INESC Porto and also with FEUP, Faculty of Engineering of the University of Porto, Portugal (e-mail: vmiranda@inescporto.pt).

Cristina Cerqueira is with INESC Porto and also with FCUP, Faculty of Sciences of the University of Porto, Portugal (email: caac@inescporto.pt).

Claudio Monteiro is is with INESC Porto and also with FEUP (email: cmonteiro@inescporto.pt)

in some countries in the coming years (like Germany, Spain or Portugal), because of the collective effort in the European Union in complying with the Kyoto protocol.

Some methods convert wind predictions into wind power predictions by using an empirical *power curve* that tries to represent the non-linear behavior of wind generators. Lange [7], for instance, derived a model for uncertainties in power prediction by relating the standard deviation of their errors with the standard deviation of errors in wind predictions and the local slope of such power curve. However, the input-output relation between wind and power is much more complex than represented by a single non-linear function, which will introduce unnecessary noise and therefore mapping methods such as neural networks or fuzzy inference systems are better suited to emulate such relation.

The reason for investigating the application of an Entropy criterion instead of the classical MSE, in training a mapper for wind power prediction, lies in the fact that errors in wind park power output predictions are far from being Gaussian. Even if wind predictions had Gaussian errors, the non-linearity of the characteristic curve of wind turbines causes predictions to display non-Gaussian characteristics. This has been shown, for instance, in [8], for 20 sites in Germany over a period of 3 years. Typically, error distributions from wind power prediction models are right skewed and have positive excess of kurtosis, meaning that: they are asymmetrical, they present a higher frequency of errors to the left of the mean and are flatter than the Gaussian distribution.

Gaussian distributions are the only ones that contain information in their first two moments. When dealing with error distributions as if they were Gaussian, we miss what may be important information that could be used to build a better predictor. This is the case when we use a variance criterion such as MSE as a criterion to train a mapper: we pass to the parameters of the mapper just a fraction of the information contained in the input set and leave behind useful information in the error distribution.

Instead, if one adopts an Entropy criterion, we aim at extracting all possible information from data leaving behind an error distribution with as little information content as possible.

In the work reported in this paper, power predictions will be produced by a 1st-order Takagi-Sugeno Fuzzy Inference System (TS-FIS) [9]. Instead of training the system with classical backpropagation methods, we have opted to use an special evolutionary algorithm called EPSO – Evolutionary Particle Swarm Optimization [10], to find the optimal weights that minimize a performance function of the predictor. Traditionally, this function would be the Minimum Square Error (MSE) of the output (compared with targets in a training set). Applying ITL concepts, we will instead minimize the Entropy of the error distribution.

Our results show that the new concept achieves error distributions narrower than with the MSE criterion, denoting that most of the times the error is smaller that the one achieved with the MSE criterion[11]. As a bonus, we also show that EPSO is a suitable method to discover the weights of a TS-FIS system, whether under MSE or an Entropy criterion.

II. Training Mappers

A mapper is a word used to designate a Neural Network, a Fuzzy Inference System or, in general, any system that emulates an input-output transfer function and whose performance depends on the tuning of internal weights or parameters.

We can divide a mapper in three basic modules: its internal structure, the performance criterion and the mechanism of training. Once defined the type of mapper to work with (class, structure), we have the power to act on each module in the following way (Figure 1) :

a) in the internal structure, by modifying the weights
b) in the performance criterion, by selecting an adequate measure of performance
c) in the training mechanism, by choosing an algorithm or procedure to close a feedback loop that updates the weights as a function of the performance criterion.

These are actions that can be taken independently to optimize a mapper.

In the work reported in this paper, we have made the following choices:

a) structure: a Fuzzy Inference System of the Takagi-Sugeno Type (TS-FIS or just FIS)
b) performance criterion: Renyi's Entropy minimization, as opposed to Mean Square Error minimization
c) training mechanism: an EPSO – Evolutionary Particle Swarm Optimization algorithm, as opposed to the classical back-propagation gradient algorithm.

The most important aspect dealt with in this paper is the performance criterion. For the ITL approach, using Entropy as a measure of performance in a supervised learning context (i.e., where one has a defined target T for the mapper output), the basic idea is the following: if one could discover a set of weights W that would model a mapper whose output would present a distribution of (Target-Output) errors as a Dirac function (meaning that all errors would be equal, see Figure 2), we would have reached a machine whose output would reproduce exactly the real data – by just adding to the results a bias corresponding to the mean of the pdf of the errors, i.e., the deviation from zero.

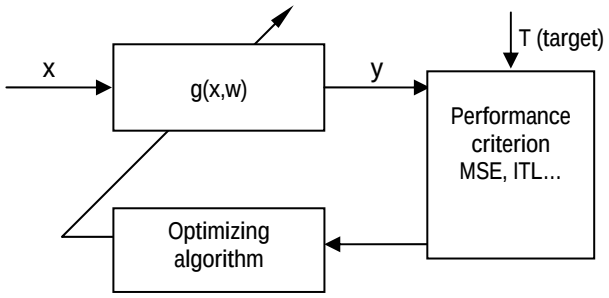

Figure 1 – Basic arrangement of a mapper identifying its three main modules

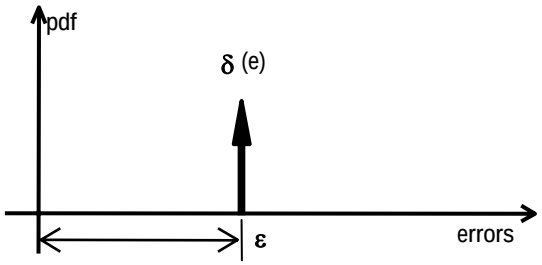

Figure 2 – A mapper producing a systematic error ε for all inputs will display a error density function like a Dirac function

Therefore, the objective of model development should be to discover weights W that lead to a pdf of errors as much approximated as possible to a Dirac function. This may be achieved by minimizing the Entropy of the error distribution – considering that the Dirac function has minimum Entropy. The success of ITL is in having discovered a cost function representing this objective and having set up a manageable procedure to compute the solution.

III. Entropy

Entropy is a concept developed in information theory that formalizes the notion of information content. The less predictable a message is, the larger is its information content; a message perfectly know *a priori* has a zero information content.

Shannon [12] defined the Entropy of a probability distribution $P = (p_1, p_2, \ldots, p_n)$ as

$$H_S(P) = \sum_{k=1}^{N} p_k \log \frac{1}{p_k} \quad \text{with} \quad \sum_{k=1}^{N} p_k \quad \text{and} \quad p_k \geq 0$$

Although this definition has been widely applied, namely in communication systems, other definitions are possible. Renyi's Entropy [13] is defined as

$$H_{R\alpha} = \frac{1}{1-\alpha} \log \sum_{k=1}^{N} p_k^{\alpha} \quad \text{with} \quad \alpha > 0, \alpha \neq 1$$

In fact, Renyi's Entropy is a family of functions $H_{R\alpha}$ depending on a parameter α. There is a relation between Shannon's and Renyi's definitions:

$$H_{R\alpha} \geq H_S \geq H_{R\beta} \quad \text{if} \quad \beta > 1 > \alpha > 0$$

$$\lim_{\alpha \to 1} H_{R\alpha} = H_S$$

When $\alpha = 2$, we have what is called quadratic Entropy

$$H_{R2} = -\log \sum_{k=1}^{N} p_k^2$$

This definition can be generalized for a continuous random variable Y with pdf $f_Y(z)$:

$$H_{R2} = -\log \int_{-\infty}^{+\infty} {f_Y}^2(z)dz$$

We can see that Renyi's Entropy, with its sum of probabilities, is much more amenable to algorithmic implementation than Shannon's Entropy with its sum of weighted logarithms of probability.

IV. Parzen windows

The estimation of the pdf of data from a sample constituted by discrete points $\mathbf{y}_i \in R^M$, i = 1,…,N in a M-dimensional space, may be done by the Parzen window method [14]. This technique uses a kernel function centered on each point; it looks at a point as being locally described by a probability density Dirac function, which is replaced or approximated by a continuous set whose density is represented by the kernel. If a Gaussian kernel is used, the expression of the estimation $\hat{f}_Y$ for the real pdf f_Y of a set of N points is a summation of individual contributions

$$\hat{f}_Y(\mathbf{z}) = \frac{1}{N} \sum_{i=1}^{N} G(\mathbf{z} - \mathbf{y}_i, \sigma^2 \mathbf{I})$$

where G(.,.) is the Gaussian kernel and $\sigma^2\mathbf{I}$ is the covariance matrix (here assumed with independent and equal variances in all dimensions). In each dimension, we have

$$G(z_k - y_{ik}, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(z_k - y_{ik})^2}$$

It is easy to understand that the "size" of the window, here defined by the value of σ, is important in obtaining a smoother (for larger values) or more "spiky" estimate for f_Y.

V. ITL criterion

Combining Renyi's definition of the Entropy of a pdf with an estimate of the pdf by the Parzen window method, we reach an Entropy estimator for a discrete set of data points {**y**} as

$$H_{R2}(\mathbf{y}) = -\log \int_{-\infty}^{+\infty} {\hat{f}_Y}^2(\mathbf{z})d\mathbf{z} = -\log V(\mathbf{y}) \quad \text{, where}$$

$$V(\mathbf{y}) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \int_{-\infty}^{+\infty} G(\mathbf{z} - \mathbf{y}_i, \sigma^2 \mathbf{I}) G(\mathbf{z} - \mathbf{y}_i, \sigma^2 \mathbf{I}) d\mathbf{z}$$

In this expression we recognize the convolution of Gaussian functions, which has the following interesting result:

$$V(\mathbf{y}) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} G(\mathbf{y}_i - \mathbf{y}_j, 2\sigma^2 \mathbf{I})$$

This means that, in order to calculate Entropy, we do not have to calculate any integrals but simply the Gaussian function values of the vector distances between pairs of samples. In ITL vocabulary, V(**y**) is called the *information potential (IP)* of the data set. As the objective is to minimize H, one can instead maximize the information potential V. So, Max V becomes the cost function for optimizing a trainable mapper with minimum output Entropy [15].

The discovery of weights in a mapper may be done by applying a suitable optimization method that will discover the weights **w** that minimize the objective function

$$\min \; H_{R2}(\mathbf{w})$$

such as an evolutionary algorithm like EPSO.

VI. APPLICATION TO TS-FIS

Takagi-Sugeno Fuzzy Inference Systems (TS-FIS) may be viewed as neuro-fuzzy mappers and are commonly viewed as being optimized via supervised training. This means that one has a training and a test set with target values **T** known beforehand and the training task has the objective of leading the output of the system to become as similar as possible to the target. In this case, we are dealing not so much with the information content of the output **y**, but with the information content of the errors $\boldsymbol{\varepsilon} = \mathbf{T} - \mathbf{y}$. From an ITL point of view, one should therefore try to minimize the Entropy of error distribution, leaving only residual information in the errors and making use of maximum information in data to build the weight-based model or mapper.

In the following paragraphs we will describe the application of ITL criterion to 0- and 1st-order TS-FIS; its generalization to n-order TS-FIS of polynomial form is straightforward.

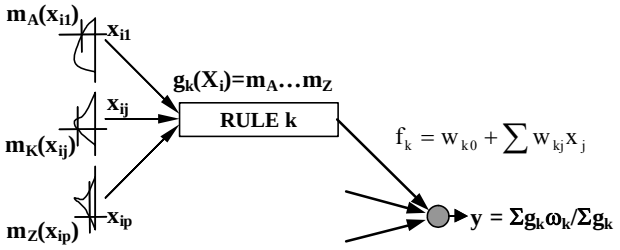

Figure 3 – TS-FIS scheme. Each input pattern X activates some membership functions; the combination of these fires a rule k with strength g_k. The weighted combination of rule firing strengths gives the output of the system.

In a TS FIS, one has rules that are fuzzy in their antecedent and crisp in their consequent. A general form of a rule k with output y_k is

IF (x_1 is A and … and x_p is Z) THEN $y_k = y(\mathbf{x}, \mathbf{w})$

The antecedent of rule k is a fuzzy set whose membership function g_k is the intersection of fuzzy sets describing conditions A,…,Z. Usually, the T-norm used to represent intersection is the product (of the membership values of each input variable).

The consequent of a rule k is a function f_k of inputs. In 0-order TS-FIS, f_k is constant and, therefore, $f_k = w_k$. In 1st-order TS-FIS, f_k is a linear combination of inputs such as in

$f_k = w_k + w_{k1}x_1 + \ldots + w_{kp}x_p$

The output of a TS-FIS is a weighted sum of the responses of all the rules (see Figure 3):

$$y_i = \frac{\sum_{k=1}^{R} g_k w_k}{\sum_{k=1}^{R} g_k} = \sum_{k=1}^{R} \bar{g}_k w_k \quad , \quad \text{with } \bar{g}_k = \frac{g_k}{\sum_{k=1}^{R} g_k}$$

In supervised training mode, an input pattern $\mathbf{X}_i$ with response y_i will generate an error relative to a desired target T_i as $\varepsilon_i = T_i - y_i$. Under an ITL criterion, one would seek to maximize the information potential associated with the distribution of errors

To optimize the performance criterion, we have to discover the adequate weights **w** of the consequents of rules. Other parameters may be adjusted by training in TS-FIS. If the membership functions of the inputs are Gaussian functions, one can also calculate updates on central variance and spread of these functions. However, this is not convenient in many cases, because the inputs are associated with linguistic expressions and the change in the shape or location of the membership functions creates dissociation with the linguistic labels they are supposed to represent.

To calculate weights, improvements have been proposed in parameter-search algorithms [16] in the line of the back-propagation principle. However, in this work we have opted for a different approach.

VII. EPSO AS THE OPTIMIZER

EPSO – Evolutionary Particle Swarm Optimization, is a hybrid in concepts of Evolutionary Algorithms and Particle Swarm Optimization, first proposed in [10] and with applications in Power Systems [17]. The reader is referred to these publications because space constraints do not allow its fully developed description. It is an Evolutionary Algorithm (close to the family of Evolution Strategies and Evolutionary Programming) where the mutation operator is only applied to strategic parameters and the recombination operator is non-conventional: it is, in fact, the "movement rule" of PSO (Particle Swarm Optimization) methods.

Recombination is an operation that produces new offspring from some form of combination of parent individuals, chosen in the population (the classical recombination operator, in GA, is called crossover). The movement rule of PSO generates a new individual as a weighted combination of parents, which are: a given individual in the population, the best ancestor of this individual and the best ancestor of the present generation. This may be seen as a form of intermediary recombination. In this type of recombination in evolutionary algorithms, a new individual is formed from a weighted mix of ancestors, and this weighted mix may vary in each space dimension.

The recombination rule for EPSO is the following: given a particle $\mathbf{X}_i$, a new particle $\mathbf{X}_i^{new}$ results from

$$\mathbf{X}_i^{(k+1)} = \mathbf{X}_i^{(k)} + \mathbf{V}_i^{(k+1)}$$

$$\mathbf{V}_i^{(k+1)} = w_{i0}^{*}\mathbf{V}_i^{(k)} + w_{i1}^{*}(\mathbf{b}_i - \mathbf{X}_i) + Cw_{i2}^{*}(\mathbf{b}_g^{*} - \mathbf{X}_i)$$

where the symbol * indicates that these parameters will undergo evolution under a mutation process, and

$\mathbf{b}_i$ – best point found by the line of ancestors of individual i up to the current generation

$\mathbf{b}_g$ – best overall point found by the swarm of particle in their past life up to the current generation

$\mathbf{b}_g^{*} = \mathbf{b}_g + w_{i4}^{*}N(0,1)$ - it is an individual in the neighborhood of $\mathbf{b}_g$.

$\mathbf{X}_i^{(k)}$ – location of particle i at generation k

$\mathbf{V}_i^{(k)} = \mathbf{X}_i^{(k)} - \mathbf{X}_i^{(k-l)}$ – is the "velocity" of particle i at generation k

w_{i1} – weight of the *inertia* term (a new particle is created in the same direction as its previous couple of ancestors)

w_{i2} – weight of the *memory* term (the new particle is attracted to the best position occupied by its ancestors)

w_{i3} – weight of the *cooperation* or *information exchange* term (the new particle is attracted to the overall best-so-far found by the swarm).

w_{i4} – weight affecting dispersion around the best-so-far

C – a diagonal matrix with each element in the main diagonal being a binary variable equal to 1 with a given communication probability p and 0 with probability (1-p); in basic models, p = 1 but in advanced models p = 0.2 has proven to be more effective in assuring the progress of the algorithm, by limiting communication among the particles of the swarm – yet another means of shaping the recombination operator.

EPSO is a self-adaptive algorithm because the weights that regulate recombination are taken as strategic parameters and are mutated and allowed to evolve. Selection acts on the recombination operator weights and, from generation to generation, a better (adaptive) recombination operation evolves.

In a diversity of problems, EPSO has been showing better performance than other meta-heuristics such as Genetic Algorithms or the classical Particle Swarm Optimization algorithm [18][19]. It tends to escape from local optima and is robust, i.e., generates results with a narrow variance in a series of runs for a problem with random initialization.

VIII. Prediction of Generation from a Wind Park

In this section, we put together a real world problem (predicting power generation in a wind park from speed and direction of wind), a mapper (a TS-FIS) and two criteria to optimize the mapper: the classic MSE and the ITL Entropy.

The data for this exercise have been gathered in a region of northern Portugal; they are composed of three time series: wind speed, wind direction and power output of a wind park, collected every ten minutes. We dealt with data collected from January 1, 2004 to February 20, 2005. For confidentiality reasons, actual power output has been transformed into a percentage of maximum available capacity of the park, which has a considerable number of generators of close to 1 MW each, spread over mountain tops, in a total installed capacity of about 40 MW.

The objective is to show that the application of the ITL criterion produces a better mapper than the application of the classical Mean Square Error. To demonstrate this, we will train two 0-order TS-FIS using an EPSO algorithm; the MSE model will find weights that minimize the Mean Square Error, and the ITL model will find weights that minimize Renyi's Entropy of the error distribution. We have selected 5000 points to train and test the FIS and divided them in a training set of 1000 points and a test set wit the rest of the points.

In Figure 4 we present a plot of untreated data, as collected from the SCADA system, showing 9993 measurements of wind speed vs. wind park power output. This set presents odd values and had to be cleaned up – for instance, you will notice points with high wind speed and no power output, due to park disconnections.

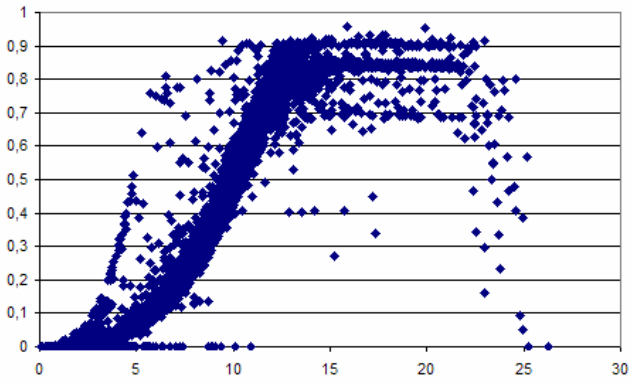

Figure 4 – Plot of wind speed (x axis) vs. power output of the wind park (y axis). Power output is represented in p.u. relative to the total installed capacity. Data untreated.

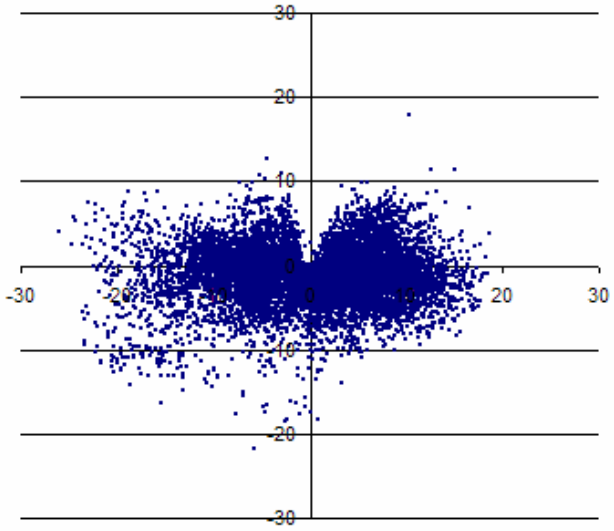

Figure 5 – Distribution of wind speed (in m/s) and direction at the measuring point near the wind park. Each point is the tip of a vector whose size is proportional to wind speed and angle is related to wind direction.

In Figure 5 we plot the same data showing wind speed and direction, as measured at a point close to the wind park.

To test the Entropy performance criterion, we have pre-defined a 0-order TS-FIS, with the following characteristics:

a) Two input variables: wind speed S, in m/s, and wind direction D, in degrees
b) The range of S is between 0 and 30, and the range of D is between 0 and 360
c) The range of power output P is between 0 and 1
d) The universe of discourse of S was partitioned in 5 fuzzy sets with Gaussian membership function

e) The universe of discourse of D was partitioned in 2 fuzzy sets with Gaussian membership function

We have maintained these membership functions fixed and only optimized the weights **w** of the 10 fuzzy rules of the system. To find optimal weights, we used a simple EPSO algorithm with 20 individuals (particles) and replication factor r = 2 (each parent gives birth to two descendants). We used Gaussian mutations with learning rate $\tau = 0.5$.

As stated before, we trained two models with EPSO: MSE and ITL. The same stopping criterion was used for both models and they only differed in the fitness function used. For the ITL model, we present results from using Gaussian Parzen windows with fixed size ($\sigma = 0.01$).

Figure 6 clearly shows that the errors from predictions of the MSE model have fewer values close to 0 than the distribution of errors resulting from the ITL model.

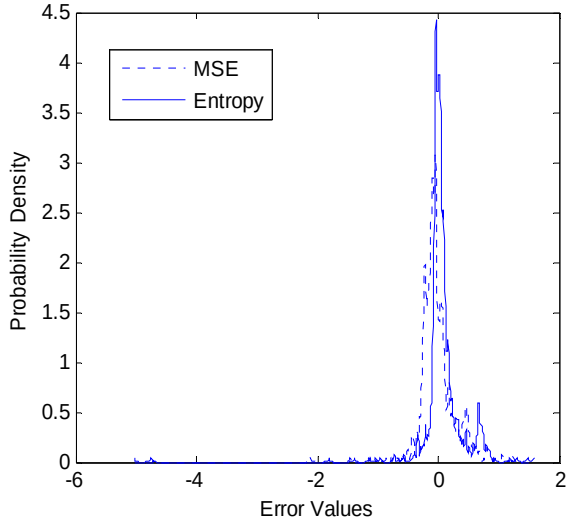

Figure 6 – Probability density functions of TS-FIS prediction errors, for both models, estimated with Parzen windows, for the training set. The more "spiky" shape of the pdf of errors associated with the training under Entropy criterion indicates a better prediction error overall.

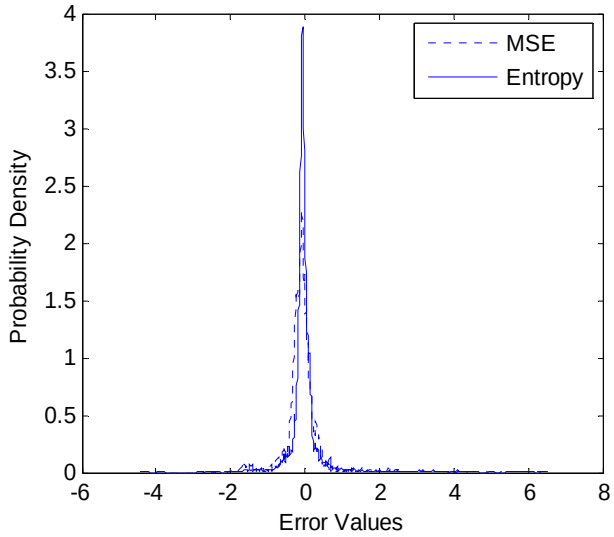

Figure 7 - Probability density functions of TS-FIS prediction errors, for both models, estimated with Parzen windows, for the test set.

This was an expected result – the Entropy of the ITL model error distribution is smaller and it is more close to a Dirac function. Of course, we have added the necessary bias to obtain a mean value of zero, before generating the predictions from the ITL model. Applying these Fuzzy Inference Systems to the test set, we obtained the result shown in Figure 7, for the probability density functions of the prediction errors. Again it is evident that predictions from the ITL model are more accurate in a larger number of cases than with the MSE model, that uses as performance criterion the classical Mean Square error.

In these figures the error distributions have been estimated also using Parzen windows with $\sigma = 0.01$. We could have presented histograms instead but felt that results would be clearer in this form.

It is interesting to notice that Renyi's Entropy, for the error distribution in the test set, generated by the application of the MSE model, is of -0.6648, while with the application of the ITL model it is of -1.1782 – naturally, a smaller value.

To be able to appreciate the impact of these results on the time domain, we plot on Figure 8 and Figure 9 two sequences of values from the test set, including the actual power measure at the SCADA and the predictions produced by the MSE and the ITL models.

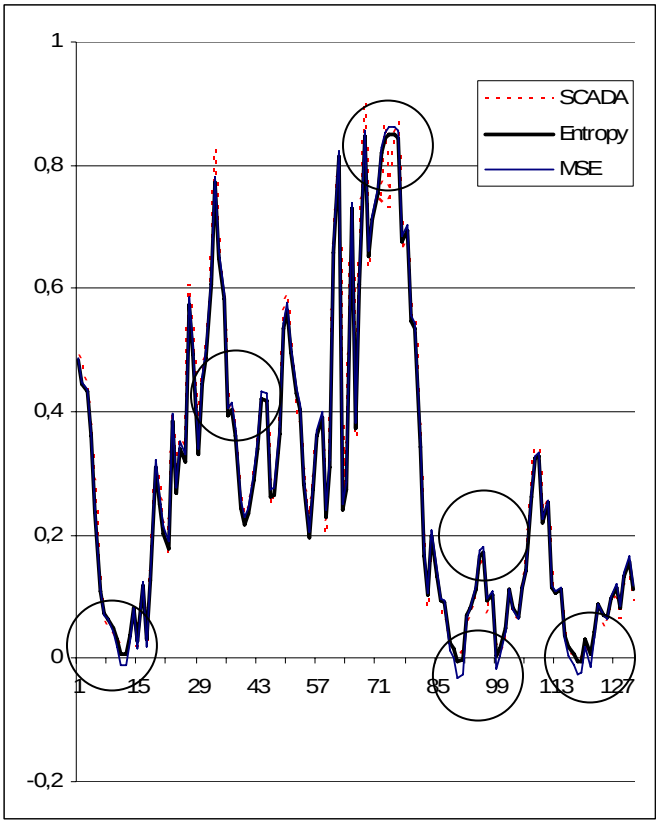

Figure 8 – Comparison, on a subset of the test set (19 weeks), of the performance of the two models. The x axis unit is days, but the plotted values are hour values. The y axis is in p.u. of nominal installed capacity. Circles identify zones where it is clear that the MSE criterion did not perform as well as the ITL Entropy criterion. In this figure, one may notice only small improvements, but the overall improvement is more clearly depicted in the pdf representation.

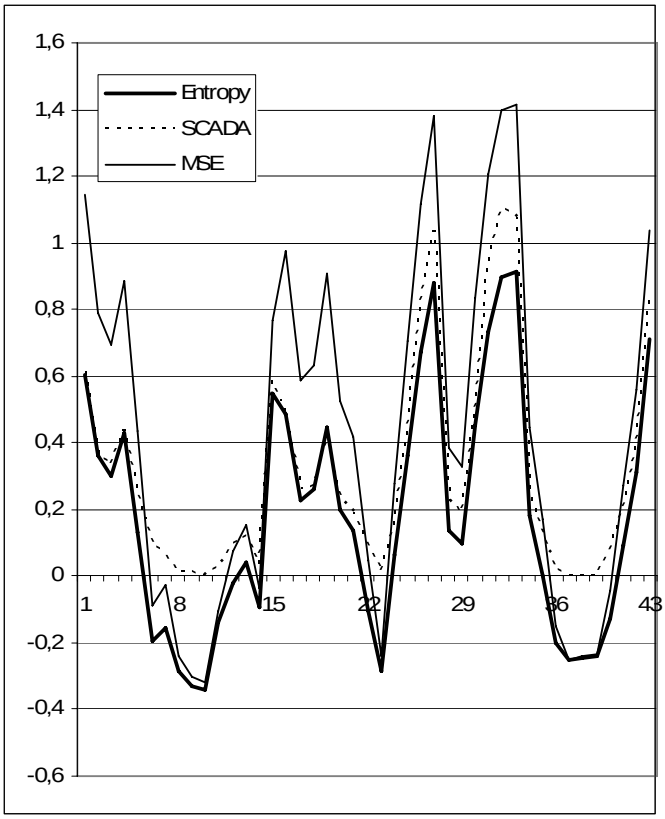

Figure 9 – Comparison, on a subset of the test set (7 weeks), of the performance of the two models. The x axis unit is days, but the plotted values are hour values. The y axis is in p.u. of nominal installed capacity. It is clear that the MSE criterion did not perform as well as the ITL Entropy criterion.

Of course, examining the 5000 points we will find some places where the MSE model produced a smaller error. But, as the error distributions show, the ITL model, based on Entropy or information content, produces a higher number of errors close to zero.

IX. CONCLUSIONS

It must be stressed that the objective of this paper is not to present a mature highly accurate prediction model for wind park power prediction – but to show that the Entropy concept, in the manageable form achieved by the Information Theoretic Learning approach, is a powerful tool with the potential to lead to the development of better prediction systems. We have proved this by producing better results for wind power prediction than by adopting the classic MSE criterion.

This means that researchers and developers should question the blind application of the Mean Square Error, as a measure of performance of Fuzzy Inference Systems or Artificial Neural Networks, or any other model of reality depending on parameters adjusted with training. The MSE criterion takes in account variance but is not sensitive to information contained in moments of higher order in the distribution of errors – and, in practice, error distributions are not well behaved nor symmetrical or Gaussian (which are the ones that contain all information in their first two moments – mean and variance).

Using Entropy as a performance criterion was not really manageable until an approach – Information Theoretic Learning – combined it, in the form of Renyi's Entropy, with Parzen windows. Nonetheless, computing this criterion is more expensive than computing the MSE criterion, because the latter only depends on (the square of) errors and the former depends on (the Gaussian of) the differences of errors. For off-line systems, however, this extra effort is worthwhile in the development phase, if it indeed leads to better predictions.

We have used the occasion to also show that we can train Fuzzy Inference Systems with a meta-heuristic such as EPSO – Evolutionary Particle Swarm Optimization. We could have used a platform such as ANFIS for the application of the MSE criterion, but this would not provide weight calculation for the Entropy criterion. To put all simulations under the same conditions, and not make the comparisons dependent on the method used, we have applied the same algorithm (EPSO) to both models and built Takagi-Sugeno Fuzzy Inference Systems from there. If anything, the Entropy criterion was in a disadvantage, because this objective function is not as well behaved as the MSE, it may display considerable number of local optima. It is also, by the way, the first time EPSO is used for such an application, with success.

The example presented belongs to the intensive research efforts presently done in the area of wind prediction and wind power forecasting. Any technique that helps in extracting more information from data will help, because the problems are very difficult, especially in medium and long term prediction. Entropy is a measure of information content and, therefore, performance criteria using Entropy will certainly show useful in the context of building models of reality.

X. ACKNOWLEDGMENT

Work partially developed while visiting the University of Florida, Gainesville, FL, USA, in April 2005. V. Miranda gratefully acknowledges the scientific support of Prof. J. Principe and also the financial support of FCT – Foundation for Science and Technology, Portugal.

XI. REFERENCES

[1] J. C. Principe and Dongxin Xu, "Introduction to information theoretic learning", *Proc. International Joint Conference on Neural Networks (IJCNN'99)*, Washington DC, USA, 10-16 July 1999, pp. 1783-1787

[2] J. C. Principe and D. Xu "Information-theoretic learning using Renyi's quadratic Entropy", in J.-F. Cardoso, C. Jutten, and P. Loubaton, editors, *Proceedings of the First International Workshop on Independent Component Analysis and Signal Separation*, Aussois, France, pages 407-412, 1999.

[3] L. Landberg, "Short term prediction of the power production of wind farms", Journal of Wind Eng. and Ind. Aerodynamics, no.80, pp. 207-220, 1999

[4] B. Bailey, M. C. Brower and J. C. Stack "Short term wind forecasting – development and application of a mesoscale model!, Proceedings of the 1999 European Wind Energy Conference EWEC'99, Nice, France, pag. 1062-1065, March 1999

[5] G. Giebel, L. Landberg and T. S. Nielsen, "The ZEPHYR project: the next generation prediction tool", Proceedings of the 2001 European Wind Energy Conference EWEC'01, pp. 777-781, Copenhagen, Denmark, June 2001

[6] S. Li, D. C. Wunsch E. A. O'Hair, "Using neural networks to estimate wind power turbine generation", IEEE Transactions on Energy Conversion, vol. 13, no.3, pp. 276-282, September 2001

[7] M. Lange, “Analysis of the uncertainty of wind power predictions”, PhD Thesis, Carl von Ossietzky University, Oldemburg, Germany, 2003
[8] M. Lange, “On the uncertainty of wind power predictions - Analysis of the forecast accuracy and statistical distribution of errors”, Transactions of the ASME, Journal of Solar Energy Engineering, no. 127, v.2, pag. 177-194, May 2005
[9] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its application to modeling and control”, IEEE Transactions on Systems, Man and Cybernetics, v. 15, pp. 116-132, 1985
[10] V. Miranda and N. Fonseca, "EPSO – Best-of-Two-Worlds Meta-Heuristic Applied To Power System Problems ", *Proceedings of WCCI 2002 - World Congress on Computational Intelligence - CEC - Conference on Evolutionary Computing*, Honolulu, Hawaii, U.S.A., May, 2002
[11] D. Erdogmus and J. C. Principe, “Comparison of Entropy and mean square error criteria in adaptive system training using higher order statistics”, in P. Pajunen andJ. Karhunen, editors, Proceedings of the Second International Workshop on Independent Component Analysis and Blind Signal Separation, Helsinki, Finland, Otamedia, Espoo, Finland, 2000, , pages75-80
[12] C.E. Shannon, “A Mathematical Theory of Communications”, Bell Systems Technical Journal, vol. 27, pp. 379-423, pp. 623-656, 1948.
[13] A. Renyi, “Some Fundamental Questions of Information Theory”, *Selected Papers of Alfred Renyi*, vol 2, pp. 526-552, Akademia Kiado, Budapest, 1976.
[14] E. Parzen, “On the estimation of a probability density function and the mode”, *Annals Math. Statistics*, v. 33, 1962, p. 1065
[15] D. Erdogmus and J. C. Principe, “Generalized Information Potential Criterion for Adaptive System Training”, *IEEE Transactions on Neural Networks*, vol. 13, no. 5, September 2002, pp. 1035-1044
[16] R. A. Morejon and J. C. Principe, “Advanced search algorithms for information-theoretic learning with kernel-based estimators”, *IEEE Transactions on Neural Networks*, vol. 15, no. 4, July 2004, pp. 874-84
[17] V. Miranda and N. Fonseca, “EPSO - Evolutionary Particle Swarm Optimization, a New Algorithm with Applications in Power Systems”, Proceedings of the IEEE Transmission and Distribution Asia-Pacific Conference 2002, Yokohama, Japan, Oct 2002
[18] H. Mori and Y. Komatsu, “A Hybrid Method of Optimal Data Mining And Artificial Neural Network for Voltage Stability Assessment”, *Proceedings of IEEE St. Petersburg PowerTech Conference*, Russia, June 2005
[19] N. W. Oo and V. Miranda, “Multi-energy Retail Market Simulation with Intelligent Agents”, *Proceedings of IEEE St. Petersburg PowerTech Conference*, Russia, June 2005

XII. BIOGRAPHIES

Vladimiro Miranda received his Licenciado, Ph.D. and Agregado degrees from the Faculty of Engineering of the University of Porto, Portugal (FEUP) in 1977, 1982 and 1991, all in Electrical Engineering. In 1981 he joined FEUP and currently holds the position of Professor Catedrático. He is also currently Director of INESC Porto. He has authored many papers and been responsible for many projects in areas related with the application of Computational Intelligence to Power Systems.

Cristina Cerqueira is an undergraduate student in the course of Mathematics Applied to Technology, from the Faculty of Sciences of the University of Porto, Portugal, in her final year. She joined INESC Porto as a young researcher in 2005, in the Power Systems Unit, working in evolutionary methods of optimization.

Cláudio Monteiro was born in France, on March 14th, 1968. He received his Licenciado and MSc. and Ph.D degrees from FEUP in 1993, 1996 and 2003, in Electrical Engineering and Computers. In 1993 he joined INESC Porto as a researcher in the Power Systems Unit.