2015 IEEE Symposium Series on Computational Intelligence

Prediction interval modeling tuned by an improved teaching learning algorithm applied to load forecasting in microgrids

Franka Veltman[1], Luis G. Marín[2], Doris Sáez[2], Leonel Gutiérrez[2] and Alfredo Nuñez[1]
[1]Delft University of Technology, Delft, the Netherlands
[2]Universidad de Chile, Santiago, Chile

***Abstract*—In recent years there has been growing interest in prediction models for non-conventional energy sources and demand in electrical systems because of the increasing use of renewable energy sources. In this paper the proposed prediction interval models are validated using local load data from a real-life microgrid in Huatacondo, Chile. The microgrid operates with an energy management system (EMS), which dispatches distributed generators based on unit commitment, minimizing generation costs. The relevant inputs for the EMS are predictions of the consumption and the available amount of renewable resources. In this paper a linear and a Takagi-Sugeno fuzzy model are proposed and they are used to construct a prediction interval that includes a representation of the uncertainties. The model parameters are identified such that they minimize a multi-objective cost function that not only includes the error but also the width of the prediction interval and its coverage probability. The resulting parameter identification is a complex non-convex problem. An Improved Teaching Learning Based Optimization (ITLBO) algorithm is proposed in order to solve the problem. This method is compared with a Particle Swarm Optimization procedure for a benchmark problem, showing that both algorithms find similar results. ITLBO is used to identify the load prediction models. These models are used to predict load up to two days ahead. Both models succeed in accomplishing the design objectives.**

I. Introduction

The rapidly increasing use of microgrids with high penetration of renewable non-conventional energy sources (RNCES) as an alternative for electrification has generated a growing interest in recent years in the development of forecasting strategies that are able to determine i) the energy that RNCES are able to provide, and ii) the demand which the microgrids must reach.

Energy microgrids are defined as low voltage power systems with an approximate nominal rate of hundreds of kilowatts, composed mainly of local loads, RNCES, and energy storage systems (ESS) [1]. The microgrids contribute to solve the problem of the energy industry that is to deal with depletion of conventional energy resources and the increased demand for electricity. In addition, the use of the RNCES contributes to a reduction of pollution levels and emissions of carbon dioxide, which are affecting the environment [2], [3].

The design of control systems for the efficient operation of the microgrid is necessary to guarantee a reliable, safe and economical operation. In [4] control strategies for microgrids are classified on three levels. The primary and secondary levels are associated with the efficient operation, and the tertiary level refers to the coordinated operation of the microgrid with the main grid. The secondary control is named Energy Management System (EMS) and is responsible for the reliability, security and economic operation. This task becomes particularly challenging in isolated microgrids due to the uncertainty in RNCES and highly variable loads. Therefore the coordination of energy between sources and consumers becomes a critical issue [5].

For the operation of the EMS it is necessary to generate models that anticipate the behavior within a time window of the expected value of both the demand and the energy available from the RNCES. These prediction models should include the uncertainty associated with RNCES and the demand. The RNCES uncertainty is mainly due to the changing weather conditions, and the uncertainty of demand is due to small changes in the demand of individual users. Because of this, the capability of a microgrid depends on the accuracy of the predictions of both the available energy and demand. Consequently, new techniques are necessary for the design of robust prediction models.

In recent years, computational intelligence methods, such as Artificial Neural Networks (ANN) and Fuzzy Logic Systems (FLS), proved to be beneficial in issues that involved prediction modeling for RNCES and demand [6], [7], because they are universal approximators of non-linear systems. In [8], the authors proposed a method for short term wind power prediction of a wind power plant by training neural networks based on historical data of wind speed and wind direction. In [9], a neuro-fuzzy network is presented for modeling the wind farm and for forecasting the wind power. In the aforementioned models the uncertainty has not been included, only the expected value of the available energy is obtained.

In prediction interval (PI) models, the intervals in which future observations will fall with a certain probability are predicted. This model type is proposed as an alternative to ensure the accuracy of the forecast of the available energy and demand. These prediction interval models provided both the expected resources value and the measure of variability, and they are used as inputs for controlling the operation of the microgrid, for instance in order to apply strategies of robust control. In [10] and [11] the authors proposed

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.100

two different alternatives based on fuzzy systems to derive prediction interval models. In [10] a method based on the covariance of the estimation error was proposed to determine the upper and lower bounds that define the interval, while in [11] an optimization procedure was used to define the boundaries of the interval. In [12], the authors presented a forecasting system including a representation of the uncertainties associated with renewable resources and loads from a real microgrid in Huatacondo, Chile, based on covariance methods. In the current paper, the contribution is the use of other fuzzy models and the application of the ITLBO algorithm in order to identify the parameters of the model.

In [13] and [14], different methods are presented to quantify uncertainties associated with forecasting, using a ANN. The results show that the proposed methods can construct PIs by the probabilistic concept of confidence interval for load and RNCES forecasts in a short time, first the model is obtained and afterwards the prediction intervals are created.

Additionally, evolutionary algorithms are good options for optimization problems with multiple objective, non-linear objective, non-differentiable functions or in general non-deterministic polynomial-time hard (NP-hard) optimization problems. In [15], [16] and [17] different heuristic methods are presented such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), for the constructions of the load and RNCES prediction models. In [18], the Teaching Learning Based Optimization (TLBO) algorithm is proposed. An advantage of this algorithm is that it does not depend on problem specific tuning parameters, only on the common control parameters, population size and number of generations. In [19] a number of technical adjustments to the classic algorithm are proposed, resulting in the Improved Teaching Learning Based (ITLBO). Furthermore it is shown that this algorithm can perform better than the common well established algorithms like PSO and GA on a range of standard unconstrained benchmark functions.

In this paper a methodology is presented for interval modeling of the demand of a microgrid based on linear and fuzzy models. Moreover, since the resulting problem is NP-hard, the model parameters are found by using the ITLBO algorithm in order to minimize a cost function. In the methodology, the normalized width of the interval as well as the coverage probability are used as metrics for the training of the interval model. In this way, it is possible to guarantee the desired coverage probability with the minimum interval width. Furthermore, the center of the interval, considered as the expected value, is computed in such a way that the error with respect to the measured value is also minimized.

The main contribution of this work is the novel method to construct prediction intervals. New model structures are proposed and the model parameters are found by minimizing a multi-objective cost function with heuristic optimization. The prediction intervals provide a framework to represent uncertainty and they are therefore suitable for the load forecast in microgrids.

The rest of this paper is organized as follows: Section II presents the structures of the linear and fuzzy models and the cost function that is minimized by optimizing the parameters of the models. Section III provides a description of the ITLBO algorithm, an explanation of the PSO algorithm with which ITLBO is compared and the application to a benchmark problem of both algorithms. In Section IV the results of the load prediction with the proposed models are presented. The last section is devoted to conclusions and future studies.

II. Prediction using interval modeling

A. Prediction interval based on linear model

A linear model is used considering that it is a model with few parameters to identify. The following prediction interval based linear model is defined, like proposed in [20] :

$$\overline{y}(x) = \sum_{i=1}^{p} g_i x_i + g_0 + \sum_{i=1}^{p} s_i |x_i| + s_0 \tag{1}$$

$$\underline{y}(x) = \sum_{i=1}^{p} g_i x_i + g_0 - \sum_{i=1}^{p} s_i |x_i| - s_0 \tag{2}$$

where $\overline{y}$ is the upper-bound of the interval, $\underline{y}$ is the lower-bound, and $x = [x_1, ..., x_p]$ the inputs. The parameters of the interval model $g = [g_0, g_1 ... g_p]$ and $s = [s_0, s_1, ..., s_p]$ need to be identified.
In the PI based fuzzy model that is to be proposed in the next section this linear model is used in the consequences.

B. Prediction interval based on TS fuzzy model

A Takagi-Sugeno (TS) fuzzy model is a model that combines different local models and is given in the form of fuzzy rules. Thus, each rule is presented as:

$$R_j : \text{if } x_1 \text{ is } F_{j,1} \text{ and...and } x_p \text{ is } F_{j,p}, \text{ then} \tag{3}$$

$$y_j = \sum_{i=1}^{p} g_{i,j} x_i + g_{0,j} \tag{4}$$

Where p is the number of inputs. The rules $(j = 1, 2, ..., R)$ are activated according to normalized activation degree $\beta_j(x)$:

$$y(x) = f^{\mathrm{TS}}(x) = \sum_{j=1}^{R} \beta_j(x) y_j(x) \tag{5}$$

$$\beta_j(x) = \frac{A_j(x)}{\sum\limits_{j=1}^{R} A_j(x)} \tag{6}$$

$A_j(x) \in [0, 1]$ is the activation degree of rule j.
In this research Gaussian membership functions $(F_{j,i})$ are used. The activation degree per rule is given by:

$$A_j(x) = \prod_{i=1}^{p} F_{j,i}(x_i) = \prod_{i=1}^{p} e^{-0.5(a_{i,j}(x_i - b_{i,j}))^2} \tag{7}$$

where $a_{i,j}$ and $b_{i,j}$ are respectively the inverse of the standard deviation and the centers of the Gaussian curves per rule per input. The prediction interval based on fuzzy modeling is given by:

$$\overline{y}(x) = \sum_{j=1}^{R} \beta_j(x)\overline{y_j}(x) \qquad \underline{y}(x) = \sum_{j=1}^{R} \beta_j(x)\underline{y_j}(x) \tag{8}$$

with

$$\overline{y_j}(x) = \sum_{i=1}^{p} g_{i,j}x_i + g_{0,j} + \sum_{i=1}^{p} s_{i,j} \mid x_i \mid + s_{0,j} \tag{9}$$

$$\underline{y_j}(x) = \sum_{i=1}^{p} g_{i,j}x_i + g_{0,j} - \sum_{i=1}^{p} s_{i,j} \mid x_i \mid - s_{0,j} \tag{10}$$

The parameters of the fuzzy prediction interval model that should be identified are $a_j = [a_{j,1}, ..., a_{j,p}]$, $b_j = [b_{j,1}, ..., b_{j,p}]$, $g_j = [g_{j,0}, g_{j,1}, ..., g_{j,p}]$ and $s_j = [s_{j,0}, s_{j,1}, ..., s_{j,p}]$ for $j = 1, .., R$.

C. Problem statement

Let us assume that the state $x(k)$ and the output $y^*(k)$ of the system are available ($k = 1, ..., N$). The identification problem of the model is designed to minimize a multi-objective cost function considering the following three components: the Mean Squared Error (MSE), the Prediction Interval Normalized Averaged Width (PINAW) and the Prediction Interval Coverage Probability (PICP).

The MSE is the mean squared error between the actual output $y^*(k)$ and the predicted mean signal of upper- and lower-bound of the prediction interval:

$$\text{MSE} = \frac{1}{N}\sum_{k=1}^{N}\left(y^*(k) - \frac{1}{2}\left(\overline{y}(k) + \underline{y}(k)\right)\right)^2 \tag{11}$$

The PINAW is used to minimize the width of the prediction interval and it is mathematically defined as follows:

$$\text{PINAW} = \frac{1}{N \cdot R_o}\sum_{k=1}^{N}\left(\overline{y}(k) - \underline{y}(k)\right) \tag{12}$$

where R_o is the range of the output:

$$R_o = \max_k\left(y^*(k)\right) - \min_k\left(y^*(k)\right)$$

The PICP gives the ratio of points that fall within the prediction interval:

$$\text{PICP} = \frac{1}{N}\sum_{k=1}^{N}\delta(k) \tag{13}$$

$$\text{with} \quad \delta(k) = \begin{cases} 1 & \text{if} \quad \overline{y}(k) \leq y^*(k) \leq \underline{y}(k) \\ 0 & \text{otherwise} \end{cases}$$

The multi-objective cost function is defined as follows:

$$V = \beta_1 \cdot \text{MSE} + \beta_2 \cdot \text{PINAW} + \beta_3 \cdot (\eta(\text{PICP} - \text{PICP}_\text{D}))^2 \tag{14}$$

where PICP_D is the desired PICP. This term steers the PICP towards a chosen value, η regulates the size of the allowed PICP error and β_1, β_2 and β_3 are weighing factors. The proposed cost function is non-linear, furthermore the prediction interval models are not strictly linear. In the next section population based optimization methods PSO and ITLBO are used to minimize (14).

III. Improved teaching learning based algorithm

A. Algorithm description

The Teaching Learning Based Optimization algorithm is a procedure in which the individual solutions that form the population are called "learners". The best learner is always chosen as a teacher and more teachers can be added in the population. The teachers try to increase the knowledge of their class during the "teaching phase". The students are also allowed to communicate with each other in order to improve their knowledge. In the "learning phase" the emphasis is put on student-student interaction but the teacher participates as well, in general this algorithm minimizes cost function $f(X)$. The elements of the input are called "subjects" $i = 1, 2, ...p$, i.e. $X = [X^1, ..., X^p]$.

Step 1: Choose optimization parameters
Choose the number of learners (N_l) and the number of teachers (N_t) .

Step 2: Initialize population
The population (i.e. learners $k = 1, 2, ..., N_\text{l}$), is randomly initialized within a chosen interval. Evaluate the fitness $f(X_k)$ of the initial population.

Step 3: Teacher selection
Select the teachers out of the students $(X_{T_1}, X_{T_2}, ..., X_{T_{N_\text{t}}})$. The best student is always a teacher. The other teachers are randomly selected out of the remaining students and sorted based on their fitness (i.e. teacher $X_{T_{N_\text{t}}}$ is always lowest in rank).

Step 4: Assign learners to teachers
Each learner is assigned to one teacher, the level of the student is always lower than the level of his teacher and higher than the level of the teacher of the group next in rank with lower results. i.e.

$$f(X_{T_s}) \geq f(X_k) > f(X_{T_{s+1}})$$

when student X_k is assigned to teacher X_{T_s}

Step 5: Teaching phase
The adaptive teaching factor is given by

$$(T_F)_k = \frac{f(X_k)}{f(X_{T_s})}$$

for learner k in group s.
The difference mean $(D_M)_s^i$ evaluates the difference between the current mean of the group and the result of the corresponding teacher in a specific subject:

$$(D_M)_s^i = r(X_{T_s}^i - (T_F)_k(M_s^i))$$

where r is a random number on interval $[0, 1]$ and M_s^i is the mean of group s in subject i.
Per subject another learner in the group X_j^i is randomly selected and the knowledge of learner k of group s in subject i

is updated by the knowledge of his teacher and the knowledge of the other learner j according to

$$X_k^{i\,\prime} = \begin{cases} X_k^i + (D_M)_{i,s} + r(X_j^i - X_k^i) & \text{if } f(X_j) \leq f(X_k) \\ X_k^i + (D_M)_{j,s} + r(X_j^i - X_k^i) & \text{if } f(X_k) < f(X_j) \end{cases}$$

in all subjects, for all students and all groups.

Step 6: Evaluation teaching phase

Calculate new fitness for all learners. Update population according to:

$$X_k{}' = \begin{cases} X_k{}' & \text{if} \quad f(X_k') < f(X_k) \\ X_k & \text{otherwise} \end{cases}$$

Step 7: Learning phase

In each group update the knowledge of the learners in a specific subject by the knowledge of another learner l of the same group and the teacher of the group according to

$$X_k^{i\,\prime\prime} = X_k^{i\,\prime} + r_1(X_k^{i\,\prime} - X_l^{i\,\prime}) + r_2(X_{T_s}^i - E_F X_k^{i\,\prime})$$
$$\text{if} \quad f(X_k') \leq f(X_l')$$

$$X_k^{i\,\prime\prime} = X_k^{i\,\prime} + r_1(X_l^{i\,\prime} - X_k^{i\,\prime}) + r_2(X_{T_s}^i - E_F X_k^{i\,\prime})$$
$$\text{if} \quad f(X_l') < f(X_k')$$

where r_1 and r_2 are random numbers in interval $[0, 1]$ and the exploration factor E_F is chosen randomly out of $\{1, 2\}$.

Step 8: Evaluation learning phase

Calculate new fitness for all learners. Update population according to:

$$X_k{}'' = \begin{cases} X_k{}'' & \text{if} \quad f(X_k{}'') < f(X_k{}') \\ X_k{}' & \text{otherwise} \end{cases}$$

otherwise choose $X_k'' = X_k'$.

Step 9: Remove worst solution

Replace worst solution of group by best solution of the group of past iteration.

Step 10: Combine groups

Combine all groups and replace duplicate solutions by best (unique) solutions of past iteration.

Step 11: Iterate

Repeat from *Step 3* until minimization objective is fulfilled or maximum number of iterations is reached.

B. PSO algorithm

The ITLBO optimization is compared with another population based optimization: Particle swarm optimization [21]. The population existing out of N_p particles is initialized with a location and a speed. For all subjects j every iteration i the velocity v_j and the location x_j of every particle are updated according to:

$$\begin{aligned}(v_j)_i =& \omega(v_j)_{i-1} + r_1 c_1(p_j - (x_j)_{i-1}) + ... \\ & r_2 c_2(g - (x_j)_{i-1}) + r_3 c_3(n - (x_j)_{i-1}) \\ (x_j)_i =& (x_j)_{i-1} + (v_j)_i\end{aligned}$$

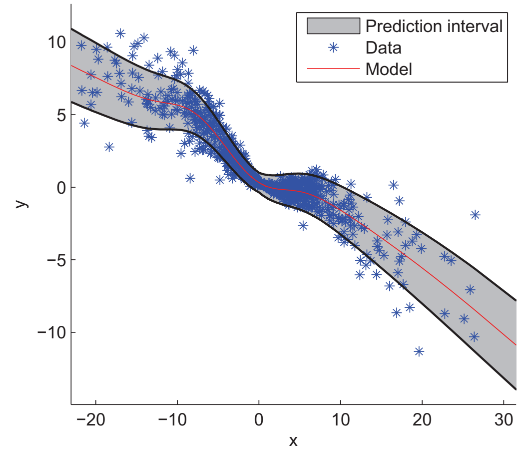

Fig. 1. 1-input data set with model and prediction interval

where p is the best previous solution of the particle, g is the best solution of the swarm so far, and n is the best neighbor of the particle. r_1, r_2, r_3 are random numbers on interval $[0, 1]$ and c_1, c_2 are constants, called cognitive and social scaling acceleration, c_3 is the neighborhood acceleration. The particles are arranged in a circle and learn from their best neighbor selected out of N_h particles of both left and right of the particle. The neighbors help the algorithm escape from local optima as explained in [22].

C. Benchmark

In this section data sets are created with a fuzzy model as in (3)-(7), with the adjustment that the consequent of the fuzzy rules includes a noise term, i.e. (4) is replaced by:

$$y_j = \sum_{i=1}^{p} g_{i,j} x_i + g_0 + e \left(\sum_{i=1}^{p} s_{i,j} x_i + s_0 \right) \tag{15}$$

where e is white noise and p the number of inputs. It is chosen to create 3 datasets with different numbers of inputs ($p = 1, 5$ and 10), all with 3 fuzzy rules. The parameters g, s, a and b were randomly chosen. The input is chosen as three sets of 300 points. Each set corresponds to a rule and has the center and standard deviation that correspond to this rule. A second set of data is created in the same way and used as validation data. The one dimensional ($p = 1$) training data set is depicted in Figure 1.

An ITLBO and a PSO algorithm are used in order to identify a linear and a fuzzy model that minimize (14) with $\beta_1 = 30$, $\beta_2 = 100$, $\beta_3 = 1$, $\eta = 100$ and $\mathrm{PICP_D} = 0.9$.

In this paper we have performed an offline optimization over the parameters. The ITLBO algorithm is applied with 50 learners and 4 teachers. The PSO algorithm also has a population size of 50 and parameters $c_1 = 1.5$, $c_2 = 1.5$, $c_3 = 1$, $N_\mathrm{h} = 2$, the w runs from $0.9 - 0.4$ during the optimization.

The parameters of the linear model are initialized by a linear model that minimizes only the MSE, this model can be found by linear least squares method. g and s are to be identified, resulting in a total number of optimization parameters $N_{\mathrm{o}} = 2(p+1)$.
The initial parameters of the fuzzy model are found by the Gustafson-Kessel clustering algorithm [23]. For the fuzzy model a, b, g and s need to be identified for every rule, resulting in: $N_{\mathrm{o}} = R(2p + 2(p+1))$. In this case $R = 3$.

The number of iterations for ITLBO is set to 25.000. Since ITLBO executes two function evaluation per learner every iteration and PSO only one, the number of iterations for PSO is set to 50.000. In this way the two algorithms perform the same total number of function evaluations. All optimizations are executed five times in order to test the robustness of the algorithms. The final values of the cost function and its components of the linear and fuzzy models are reported in Table I for both training and validation data. For the linear model the results are similar for the two optimization techniques, the standard deviations of the solutions are small so the algorithms converge to the same solution every trial.

Furthermore it can be seen that the PICP term in the cost function is successful in keeping the PICP close to 90%. The results of the training and validation data are similar.

Looking at the fuzzy models it can be seen that the found cost function values are all lower than for the linear model, as expected because the system structure is non-linear since the benchmark is based on fuzzy rules. The true MSE and PINAW are significantly smaller than for the linear model in all cases.

Looking at the standard deviations it can be concluded that both algorithms do not converge to the same solution every trial for the fuzzy model. The differences in performance between the training and validation data is quite significant. This is probably because of the large number of optimization parameters that are considered in fuzzy modelling, which allows some over tuning.

When the two optimization methods are compared it occurs that the results are rather similar. ITLBO however has a slightly lower computation time and less optimization parameters that require tuning. Since PSO and ITLBO resulted comparable in terms of capability of finding the optimum, only the results of ITLBO are analysed on the real data case.

In the next section ITLBO is used to find the model parameters in a load forecasting application. Since for the fuzzy model convergence is not guaranteed, the optimization is executed several times and the best solution is selected.

IV. Application for load forecasting

Load forecasting is an important issue in the operation of microgrids. In order to use robust model predictive control it is desirable to predict a reasonable interval in which the following data point will fall. In this section data from the Huatacondo microgrid in the North of Chile is used to train and validate a linear and a fuzzy prediction interval model with an ITLBO algorithm. The Huatacondo microgrid is composed of two photo-voltaic systems (24 kW); a wind turbine (5 kW); the existing diesel generator unit in the village which is typical of isolated grids; an ESS composed of a lead-acid battery bank connected to the grid through a bidirectional inverter; a water pump; and loads (28 kW).

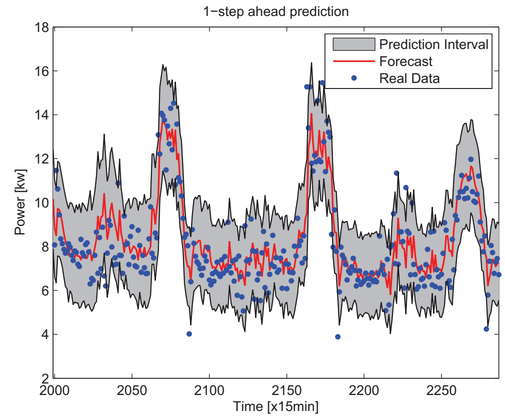

Fig. 2. 15 minutes ahead prediction interval of fuzzy model, validation data

The load is measured during a period of 147 days, with a sample time of 15 minutes. It is assumed that for both the fuzzy and linear model the maximum possible number of regressors involved in the one step ahead prediction is 100 (1 day and 1 hour).

For the linear model the number of inputs is reduced with a step wise fit method, resulting in model structure:

$$\begin{aligned} p_l(k) = f(p_l(k\text{-}1), p_l(k\text{-}2), p_l(k\text{-}3), p_l(k\text{-}4), \ldots \\ p_l(k\text{-}11), p_l(k\text{-}40), p_l(k\text{-}42), p_l(k\text{-}79), p_l(k\text{-}86), \ldots \\ p_l(k\text{-}92), p_l(k\text{-}93), p_l(k\text{-}95), p_l(k\text{-}96)) \end{aligned} \quad (16)$$

This structure of 13 inputs results in 28 optimization parameters. The number of rules and the regressors to be used in the fuzzy model are selected using a sensitivity analysis: a process in which various R are evaluated while eliminating inputs. Finally the input-rulenumber combination resulting in the lowest MSE is selected. The optimal structure has 3 rules and the following inputs:

$$\begin{aligned} p_l(k) = f^{\mathrm{TS}}(p_l(k\text{-}1), p_l(k\text{-}2), p_l(k\text{-}3), p_l(k\text{-}4), \ldots \\ p_l(k\text{-}92), p_l(k\text{-}93), p_l(k\text{-}95), p_l(k\text{-}96), p_l(k\text{-}100)) \end{aligned} \quad (17)$$

The 9 inputs and 3 rules results in 152 optimization parameters. The parameters used for the ITLBO algorithm are the same as in the benchmark problem, 50.000 iterations are done per optimization. The data is divided in 75% training data and 25% validation data.

In this case problem parameters a and b were fixed on the values found by the Gustafson-Kessel algorithm, reducing the number of optimization parameters to 60. The optimization for the fuzzy model is done 3 times and the model with the lowest cost function value is selected.

TABLE I

BENCHMARK WITH LINEAR AND FUZZY MODEL MINIMIZED BY ITLBO AND PSO ALGORTIHM

Linear model with ITLBO							
		$p=1,\ N_\text{o}=6$		$p=5,\ N_\text{o}=22$		$p=10,\ N_\text{o}=42$	
		Mean	std	Mean	std	Mean	std
Training	V	55.69	0.01	184231.74	0.12	17871.20	0.61
	MSE	1.49	0.00	6140.44	0.00	594.94	0.00
	PICP (%)	89.87	0.05	89.58	0.21	89.84	0.10
	PINAW (%)	10.83	0.02	18.40	0.32	22.83	0.62
Validation	V	60.21	0.49	170457.81	0.83	17307.17	1.65
	MSE	1.48	0.00	5681.42	0.01	576.11	0.04
	PICP (%)	88.07	0.13	90.44	0.33	90.11	0.62
	PINAW (%)	12.01	0.02	15.01	0.26	23.60	0.62
Duration 100 iterations (s)		0.33		0.37		0.41	
Linear model with PSO							
		$p=1,\ N_\text{o}=6$		$p=5,\ N_\text{o}=22$		$p=10,\ N_\text{o}=42$	
		Mean	std	Mean	std	Mean	std
Training	V	55.70	0.01	184233.75	1.16	17870.53	0.37
	MSE	1.49	0.00	6140.44	0.00	594.94	0.00
	PICP (%)	89.89	0.00	89.62	0.26	89.73	0.06
	PINAW (%)	10.84	0.01	20.43	1.32	22.11	0.35
Validation	V	60.04	0.53	170460.37	1.36	17306.83	2.53
	MSE	1.48	0.00	5681.44	0.01	576.12	0.07
	PICP (%)	88.11	0.14	90.29	0.45	89.69	0.35
	PINAW (%)	12.02	0.01	17.04	1.23	22.89	0.35
Duration 200 iterations (s)		0.22		0.24		0.24	
Fuzzy model with ITLBO							
		$p=1,\ N_\text{o}=18$		$p=5,\ N_\text{o}=66$		$p=10,\ N_\text{o}=126$	
		Mean	std	Mean	std	Mean	std
Training	V	39.69	1.07	32512.65	233.53	8819.48	713.50
	MSE	1.05	0.04	1083.56	7.79	293.53	23.78
	PICP (%)	89.93	0.06	89.93	0.06	89.80	0.09
	PINAW (%)	8.17	0.08	5.89	0.83	13.45	0.62
Validation	V	46.41	1.51	36507.74	770.12	13463.93	1411.16
	MSE	1.08	0.03	1216.59	25.54	446.60	46.66
	PICP (%)	87.82	0.19	88.11	1.54	82.98	1.87
	PINAW (%)	9.15	0.13	4.77	0.64	13.84	0.64
Duration 100 iterations (s)		4.23		9.45		17.33	
Fuzzy model with PSO							
		$p=1,\ N_\text{o}=18$		$p=5,\ N_\text{o}=66$		$p=10,\ N_\text{o}=126$	
		Mean	std	Mean	std	Mean	std
Training	V	40.59	1.46	35269.41	5244.89	9919.30	257.69
	MSE	1.08	0.04	1175.44	174.82	330.12	8.59
	PICP (%)	89.89	0.08	89.84	0.06	89.60	0.13
	PINAW (%)	8.12	0.39	6.12	0.57	15.45	0.29
Validation	V	48.47	2.49	42082.37	5991.52	12750.13	1283.24
	MSE	1.11	0.02	1402.39	199.60	423.96	42.78
	PICP (%)	87.58	0.62	87.84	1.15	86.20	1.01
	PINAW (%)	9.05	0.50	4.98	0.44	16.00	0.35
Duration 200 iterations (s)		5.05		10.88		18.01	

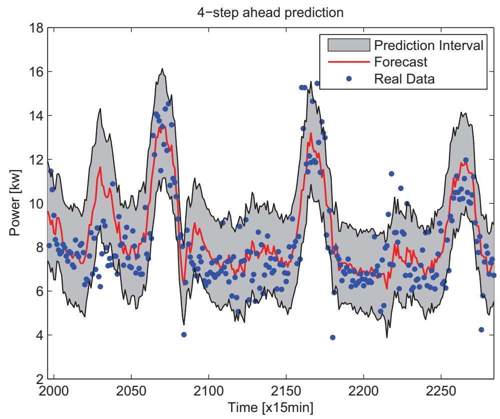

Fig. 3. 1 hour ahead prediction interval of fuzzy model, validation data

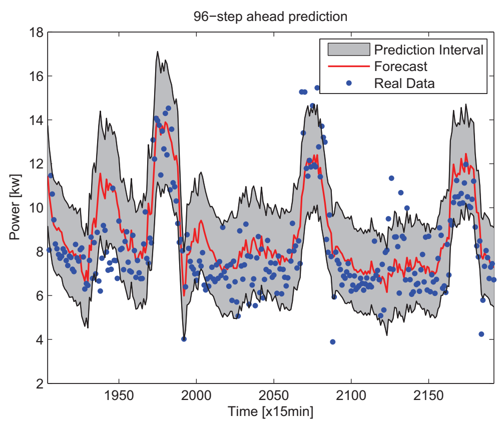

Fig. 4. 1 day ahead prediction interval of fuzzy model, validation data

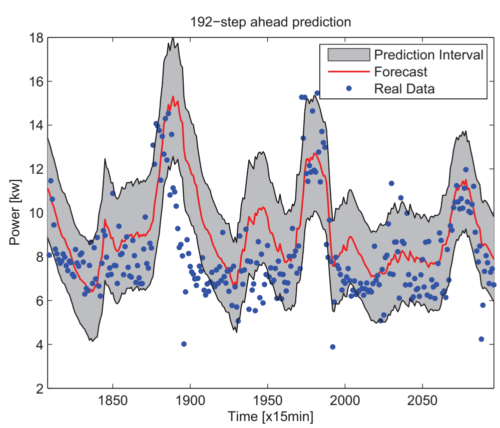

Fig. 5. 2 day ahead prediction interval of fuzzy model, validation data

The cost function values and its elements of both proposed models for a one step ahead prediction are reported in Table II. It can be seen that the fuzzy model finds a lower cost function value although the PINAW is higher than for the linear model. A larger prediction interval width obviously provides a better coverage probability. Since the models have two minimization objectives that are enforced in an indirect way (by the weight factors in the cost function) and the PICP constraint is not strict, it is hard to make a qualitative comparison between the two models. Due to the larger number of optimization parameters and the more complex computation of the output of the fuzzy model, this model is more computational expensive than the linear model.

TABLE II
COST FUNCTION FOR TRAINING AND VALIDATION DATA

		Linear	Fuzzy
Training	V	72.17	71.62
	MSE	1.84	1.79
	PICP (%)	89.69	89.47
	PINAW (%)	16.90	17.78
Validation	V	83.22	82.96
	MSE	2.17	2.21
	PICP (%)	88.34	89.53
	PINAW (%)	15.37	16.40
Duration 100 iterations (s)		4.2	19.3

TABLE III
PREDICTION INTERVAL COMPARISON FOR VALIDATION DATA

		Linear	Fuzzy
15 min	V	83.22	82.96
	MSE	2.17	2.21
	PICP (%)	88.34	89.53
	PINAW (%)	15.37	16.40
1 hour	V	130.21	114.45
	MSE	2.87	2.92
	PICP (%)	84.63	86.77
	PINAW (%)	15.27	16.42
1 day	V	164.12	143.21
	MSE	3.35	3.46
	PICP (%)	83.04	85.21
	PINAW (%)	15.18	16.47
2 days	V	236.94	192.95
	MSE	4.13	4.28
	PICP (%)	80.10	83.07
	PINAW (%)	15.03	16.53

Simulations of the validation data with the fuzzy model for 1, 4, 92 and 192 step (corresponding with respectively 15 minutes, 1 hour, 1 day, 2 days) ahead predictions are respectively depicted in Figure 2-5, only 3 days of the 36.75 validation days are shown. It can be seen that for larger prediction windows the ratio of data within the prediction interval (PICP) decreases. This can also be seen in Table III, in which multiple step ahead predictions are performed with the identified models. For the linear and fuzzy model can be seen that the PINAW barely changes for larger prediction horizons. The fuzzy model performs better in terms of PICP preservation for multiple step ahead predictions and the prediction interval width is a little bigger. The cost function value rises quickly when the prediction horizon increases (for both models), this is due to the PICP term. The PICP value that was trained to be within bounds for a one step ahead prediction and not for a higher step ahead prediction.

V. Conclusion and future work

In this paper an ITLBO algorithm is implemented and used to identify the parameters of a linear and a TS fuzzy prediction interval model. The parameters of the models are determined based on a multi-objective cost function that does not only minimize the prediction error but also the width of the prediction interval while maintaining a chosen coverage probability.

Different from standard prediction interval methods, the parameters of the prediction intervals of the models proposed in this paper are identified intermediately without creating a forecast first.

With a benchmark problem it is demonstrated that for this non-convex problem with many parameters the ITLBO and PSO algorithm find similar results. For the fuzzy model, which has many parameters to identify, an advantage of ITLBO over PSO is that the computation time is slightly smaller, moreover this algorithm has less optimization parameters to tune.

When the two models were applied in load forecasting for the Huatacondo microgrid it appeared that the cost function structure is able to find models with the desired coverage probability that also minimize the mean squared error and the band width of the prediction interval. Note that with the fuzzy modeling a slightly higher coverage probability is obtained, for both one and higher step ahead predictions.

Future work should include a Pareto analysis for the proposed multi-objective cost function in order to be able to perform a better comparison between the models.

References

[1] R. Lasseter, "Microgrids," *Power Engineering Society Winter Meeting, IEEE. New York, NY, USA, Jan. 27-31*, vol. 1, pp. 305–308, 2002.

[2] J. Hu, J. Zhu, Y. Qu, and J. Guerrero, "A new virtual-flux-vector based droop control strategy for parallel connected inverters in microgrids," *ECCE Asia Downunder (ECCE Asia). Melbourne, Australia, Jun. 3-6*, pp. 585–590, 2013.

[3] L. Valverde, C. Bordons, and F. Rosa, "Power management using model predictive control in a hydrogen-based microgrid," in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society, Montreal, Canada, Oct. 25-28*, 2012, pp. 5669–5676.

[4] D. Olivares, A. Mehrizi-Sani, A. Etemadi, S. Member, C. Cañizares, R. Iravani, M. Kazerani, A. Hajimiragha, O. Gomis-Bellmunt, M. Saeedifard, S. Member, R. Palma-Behnke, G. Jiménez-Estévez, and N. Hatziargyriou, "Trends in microgrid control," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1905 – 1919, 2014.

[5] R. Palma-Behnke, C. Benavides, F. Lanas, B. Severino, L. Reyes, J. Llanos, and D. Sáez, "A microgrid energy management system based on the rolling horizon strategy," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 996–1006, 2013.

[6] A. Ghanbari, E. Hadavandi, and S. Abbasian-Naghneh, "Comparison of artificial intelligence based techniques for short term load forecasting," in *2010 Third International Conference on Business Intelligence and Financial Engineering. Hong Kong, Aug. 13-15*. IEEE, 2010, pp. 6–10.

[7] H. Alfares and M. Nazeeruddin, "Electric load forecasting: Literature survey and classification of methods," *International Journal of Systems Science*, vol. 33, no. 1, pp. 23–34, 2002.

[8] Z. Liu, W. Gao, Y.-H. Wan, and E. Muljadi, "Wind power plant prediction by using neural networks," in *Energy Conversion Congress and Exposition (ECCE). Raleigh, NC, USA, Sep. 15-20*, 2012, pp. 3154–3160.

[9] J. Xia, P. Zhao, and Y. Dai, "Neuro-fuzzy networks for short-term wind power forecasting," in *2010 International Conference on Power System Technology. Hangzhou, Zhejiang Province, China, Oct. 24-28*. IEEE, 2010, pp. 1–5.

[10] I. Škrjanc, "Fuzzy confidence interval for pH titration curve," *Applied Mathematical Modelling*, vol. 35, no. 8, pp. 4083–4090, 2011.

[11] I. Škrjanc, S. Blažič, and O. Agamennoni, "Interval fuzzy model identification using l -Norm," *IEEE Transactions on fuzzy systems*, vol. 13, no. 5, pp. 561–568, 2005.

[12] D. Sáez, F. Ávila, D. Olivares, C. Cañizares, and L. Marín, "Fuzzy prediction interval models for forecasting renewable resources and loads in microgrids," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 548–556, 2015.

[13] H. Quan, D. Srinivasan, and A. Khosravi, "Short-term load and wind power forecasting using neural network-based prediction intervals," *IEEE Transactions on Neural and Learning System*, vol. 25, no. 2, pp. 303–315, 2014.

[14] K. Methaprayoon, C. Yingvivatanapong, W.-j. Lee, and J. Liao, "An integration of ANN wind power estimation into unit commitment considering," *IEEE Transactions on Industry Applications*, vol. 43, no. 6, pp. 1441–1448, 2007.

[15] E. Pratheepraj, A. Abraham, S. Deepa, and V. Yuvaraj, "Very short term wind power forecasting using PSO-neural network hybrid system," in *Advances in Computing and communications*, Springer, Ed., 2011, ch. Soft Compu, pp. 503–511.

[16] V. Hinojosa and A. Hoese, "Short-term load forecasting using fuzzy inductive reasoning and evolutionary algorithms," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 565–574, 2010.

[17] D. Niu, Y. Wang, and D. Wu, "Power load forecasting using support vector machine and ant colony optimization," *Expert Systems with Applications*, vol. 37, no. 3, pp. 2531–2539, 2010.

[18] R. Rao, V. Savsani, and D. Vakharia, "Teachinglearning-based optimization: A novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.

[19] R. Rao and V. Patel, "An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems," *Scientia Iranica*, vol. 20, no. 3, pp. 710–720, 2012.

[20] N. Karnik and J. Mendel, "Operations on type-2 fuzzy sets," *Fuzzy Sets and Systems*, vol. 122, no. 2, pp. 327–348, 2001.

[21] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, Nov 1995, pp. 1942–1948 vol.4.

[22] D. Tran, Z. Wu, and V. Nguyen, "A new approach based on enhanced pso with neighborhood search for data clustering," *International Conference of Soft Computing and Pattern Recognition (SoCPaR). Hanoi, Vietnam, Dec. 15-18*, pp. 98–104, 2013.

[23] D. Gustafson and W. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," *IEEE Conference On Adaptive Processes. San Diego, CA, USA, Jan. 10-12*, vol. 25, no. 1, pp. 761–766, 1978.

Acknowledgment

This work has been partially supported by the Millennium Institute "Complex Engineering Systems" (ICM: P-05-004-F, CONICYT: FBO16); by Solar Energy Research Center SERC-Chile, CONICYT/FONDAP/15110019; by FONDECYT Project 1140775 and by Delft Center for Systems and Control.