Congreso Nacional de Control Automático 2013
Ensenada, Baja California, Mexico, Octubre 16-18, 2013

Power Production Forecasting for Photovoltaic Generation Systems via Neural Networks with Particle Swarm Optimization Kalman Learning

Luis J. Ricalde[1], Eduardo E. Ordoñez[1], Lifter O. Ricalde[1] and Alma Y. Alanis[2]
1) Universidad Autonoma de Yucatan, Av. Industrias no Contaminantes por Periferico Norte, Apdo. Postal 150 Cordemex, Merida, Yucatan, Mexico, e-mail: lricalde@uady.mx.
2) CUCEI, Universidad de Guadalajara, Apartado Postal 51-71, Col. las aguilas, C.P. 45080, Zapopan, Jalisco, Mexico, e-mail: almayalanis@gmail.com

***Resumen*— This paper focusses on applications of neural networks for forecasting in photovoltaic arrays. The Particle Swarm Optimization technique is applied for tuning the parameters of Extended Kalman Filter training algorithm for data modeling in smart grids. The length of the regression vector is determined using the Cao methodology. The applicability of this architecture is illustrated via simulation using real data values for Photovoltaic modules.**

Keywords: Recurrent neural networks, Particle Swarm Optimization, Photovoltaic Module, Power generation forecasting, Kalman training.

I. Introduction

The limited existing reserves of fossil fuels and the harmful emissions associated with them have lead to an increased focus on renewable energy applications in recent years. The first steps on integrating renewable energy sources began with hybrid wind and solar systems as complementing sources and as solution for rural applications and weak grid interconnections. Further research have implemented hybrid systems including several small scale renewable energy sources as solar thermal, biomass, fuel cells and tidal power. Since the production costs for photovoltaic and wind turbine applications have considerably reduced, they have become the primary choice for hybrid energy generation systems.

Photovoltaic systems convert light energy from the sun into electrical energy. The most basic system includes photovoltaic panels, wiring lines, inverter and system electrical energy storage. It typically includes a monitoring system to measure and record weather data and system performance. In order to predict the generated power from an array of PV modules, several author have proposed the use of mathematical models to characterize photovoltaic modules based on electrical circuits with different topologies which model both photocurrent source and losses through connections in the system elements. Many of these models are based on complex mathematical expressions to estimate the behavior with the solar cells and the parameters used in these models are determined experimentally or extraction techniques using analytical or numerical (Sandrolini et al, 2010).

Through the use of predictive models, it is possible to calculate the power output of photovoltaic systems when a set of entries is added to the system weather. The data used in predictive models estimate the system output parameters such as DC power, AC power and module temperature. Predictive models applied to PV can be used in several ways. They can be used before purchasing a full system to compare the desired output of a particular system with other potential designs. They can also be used to determine whether a system works as expected, allowing operators to determine maintenance schedules. Finally can be used to predict the performance of an existing system and calculate the expected return.

Most available predictive schemes for photovoltaic systems apply algebraic formulas to estimate and model the performance of each system component, then using docking models for components the entire system performance is estimated. Some models use relatively simple formulas and typical system parameters for estimating the performance (Marion and Anderberg, 2000). Other models use data from the specification sheets to generate component parameters (De Soto et al, 2000). More complex models require specific parameters are generated through rigorous testing of the components of photovoltaic systems (King et al , 2004),(King et al , 2007).

Artificial Neural Networks (ANN) have been considered as a convenient analysis tool for energy systems forecasting and control applications due to the simplicity of the model and the accuracy of the results for nonlinear and stochastic models and have been implemented in several practical applications (Senjyu et al, 2006). ANN have been previously implemented for short term predictions, outperforming other classical methods due to the fast learning algorithm which enables on-line implementations and the versatility to vary the prediction horizon . Due to their nonlinear modeling characteristics, neural networks have been successful applied in control systems, pattern classification, pattern recognition, and time series forecasting problems. There are several previous works that use artificial neural networks to predict time series for energy generation systems (Bonanno et al, 2012), (Amoudi and Zang, 2000).

In (Bonanno, 2012) the authors investigated the application of radial basis neural networks for prediction of the electrical characteristics of a PV module taking into

account the change in all parameters at different operating conditions. The applied inputs parameters considered for training were irradiation and module temperature. For the training of the network is employed a large amount of experimental data and the error backpropagation algorithm. The results are validated by simulations.

The best well-known training approach for recurrent neural networks (RNN) is the back propagation through time . However, it is a first order gradient descent method, and hence its learning speed could be very slow. Another well-known training algorithm is the Levenberg–Marquardt one; its principal disadvantage is that is not guarantee it will find the global minimum and its learning speed could be slow too, this depends on the initialization. In past years, Extended Kalman Filter (EKF) based algorithms has been introduced to train neural networks (Alanis et al, 2007) . With the EKF based algorithm, the learning convergence is improved. The EKF training of neural networks, both feedforward and recurrent ones, has proven to be reliable for many applications over the past ten years . However, EKF training requires the heuristic selection of some design parameters which is not always an easy task (Alanis, 2007) .

During the past decade, the use of evolutionary computation in engineering applications has increased. Evolutionary algorithms apply adaptation and stochastically in optimization problems in schemes as evolutionary programming, genetic algorithms and evolution strategies (Parsopoulos and Vrahatis, 2010). Particle Swarm Optimization (PSO) technique, which is based on the behavior of a flock of birds or school of fish, is a type of evolutionary computing technique . The PSO algorithm uses a population of search points that evolve in a search space using a communication method to transfer the acquired experience from best solutions. This algorithm has several advantages like the simplicity of the updating law, faster convergence time and less complexity on the reorganization of the population. The PSO methods also have emerged as an excellent tool to improve the performance of Neural Network learning process. In (Lin et al, 2009), the PSO algorithm is extended to multiple swarms in a neuro-fuzzy network with good results in forecasting applications. It has been shown that the PSO training algorithm takes fewer computations and is faster than the BP algorithm for neural networks to achieve the same performance.

In this paper we propose the use of PSO for tuning the parameters of EKF training algorithm for data modeling in smart grids. The length of the regression vector is determined using the Cao methodology which is an improvement to the false neighbors approach (Cao, 1997). The applicability of this architecture is illustrated via simulation using real data values for Photovoltaic modules and arrays in order to show the potential applications in forecasting for energy generation in smart grid schemes.

II. Neural Identification

In this paper for the neural model identification the Recurrent Multi-Layer Perceptron is chosen, then the neural model structure problem reduces to dealing with the following issues: selecting the inputs to the network and 2) selecting the internal architecture of the network.

The structure selected in this paper is NNARX (Norgaard et al, 2000) (acronym for Neural Network AutoRegressive eXternal input); the output vector for the artificial neural network is defined as the regression vector of an AutoRegressive eXternal input linear model structure (ARX) (Alanis, 2007).

It is common to consider a general nonlinear system; however, for many control applications is preferred to express the model in an affine form, which can be represented by the following equations

$$y\left(k+1\right)=f\left(y\left(k\right),y\left(k-1\right),...,y\left(k-q+1\right)\right) \quad (1)$$

where q is the dimension of the regression vector. In other words, a nonlinear mapping f exists, for which the present value of the output $y\left(k+1\right)$ is uniquely defined in terms of its past values $y\left(k\right),...,y\left(k-q+1\right)$ and the present values of the input $u\left(k\right)$.

Considering that it is possible to define

$$\phi\left(k\right)=\left[y\left(k\right),...,y\left(k-q+1\right)\right]^{T}$$

which is similar to the regression vector of a ARX linear model structure (Norgaard, 2000), then the nonlinear mapping f can be approximated by a neural network defined as

$$y\left(k+1\right)=\varphi\left(\phi\left(k\right),w^{*}\right)+\varepsilon$$

where w^{*} is an ideal weight vector, and ε is the modeling error; such neural network can be implemented on predictor form as

$$\widehat{y}\left(k+1\right)=\varphi\left(\phi\left(k\right),w\right) \quad (2)$$

where w is the vector containing the adjustable parameters in the neural network.

The neural network structure, used in this work is depicted in Fig. 1, which contains sigmoid units only in the hidden layer; the output layer is a linear one. The used sigmoid function $S\left(\bullet\right)$ is defined as a logistic one, as follows:

$$S\left(\varsigma\right)=\frac{1}{1+\exp\left(-\beta\varsigma\right)},\qquad\beta>0 \quad (3)$$

where ς is any real value variable.

A. EKF Training Algorithm

Kalman filter (KF) estimates the state of a linear system with additive state and output white noise. Kalman filter algorithm is developed for a linear, discrete-time dynamical system. For KF-based neural network training, the network weights become the states to be estimated. Due to the fact that the neural network mapping is nonlinear, an EKF-type is required (Alanis et al, 2012).

Consider a nonlinear dynamic system described by the next model in state space

$$\begin{aligned} w(k+1) &= f(k,w(k))+v_1(k) \\ y(k) &= h(k,w(k))+v_2(k) \end{aligned} \quad (4)$$

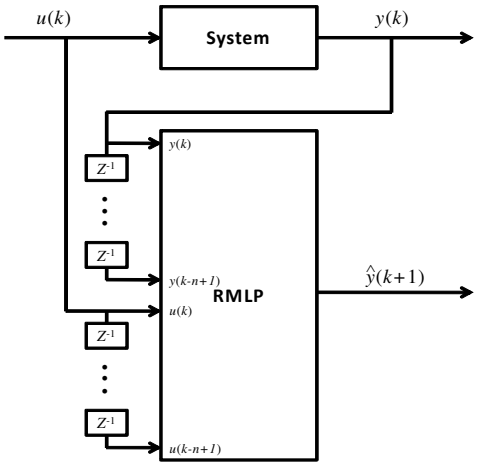

Figura 1. Neural network structure

where $v_1(k)$ and $v_2(k)$ are zero-mean, white noises with covariance matrices given by $Q(k)$ and $R(k)$, respectively. $f(k, w(k))$ denotes the nonlinear transition matrix function.

The basic idea of the extended Kalman filter is to linearize the state space model 4 at each time instant around the most recent state estimate, which is taken to be $\hat{w}(k)$. The training goal is to find the optimal weight values which minimize the prediction error. The modified Extended Kalman Filter (EKF) algorithm is defined by:

$$\begin{aligned} \hat{w}(k+1) &= \hat{w}(k) + K(k)[y(k) - \widehat{y}(k)] \quad (5) \\ K(k) &= P(k) H^T(k) M(k) \\ P(k+1) &= P(k) - K(k) H(k) P(k) + Q(k) \end{aligned}$$

with

$$\begin{aligned} M(k) &= \left[R(k) + H(k) P(k) H^T(k)\right]^{-1} \\ e(k) &= y(k) - \widehat{y}(k) \quad (6) \end{aligned}$$

where $e(k) \in \Re$ is the respective approximation error, $P \in \Re^{L \times L}$ is the prediction error associated covariance matrix at step k, $w \in \Re^L$ is the weight (state) vector, L is the respective number of neural network weights, y is the system output, $\widehat{y}$ is the neural network output, $K \in \Re^L$ is the Kalman gain vector, $Q \in \Re^{L \times L}$ is the state noise associated covariance matrix, $R \in \Re$ is the measurement noise associated covariance; $H \in \Re$ is a vector, in which each entry (H_{ij}) is the derivative of one of the neural network output, $(\widehat{y})$, with respect to one neural network weight, (w_j) defined as follows

$$H_{ij}(k) = \left[\frac{\partial \widehat{y}(k)}{\partial w_j(k)}\right]^T_{w_i(k)=\hat{w}_i(k)}$$

where $i = 1, ..., m$; $j = 1, ..., L$. Usually P, Q and R are initialized as diagonal matrices, with entries $P(0)$, $Q(0)$ and $R(0)$, respectively. It is important to note that for the EKF training algorithm $P(0)$, $Q(0)$ and $R(0)$ are considered as design parameters that typically are heuristically determined, however in this paper we propose the use of Particle Swarm Optimization for determining such parameters (Alanis, 2012).

B. PSO improvement for EKF Training Algorithm

Particle swarm optimization (PSO) is a swarm intelligence technique developed by Kennedy and Eberhart in 1995 (Kennedy and Eberhart, 1995). In the basic PSO technique proposed by Kennedy and Eberhart (Kennedy J. and Eberhart, 1995), great number of particles moves around in a multi-dimensional space and each particle memorizes its position vector and velocity vector as well as the time at which the particle has acquired the best fitness. Furthermore, related particles can share data at the best-fitness time. The velocity of each particle is updated with the best positions acquired for all particles over iterations and the best positions are acquired by the related particles over generations.

To improve the performance of the basic PSO algorithm, some new versions of it have been proposed. In this paper the algorithm proposed in (Kiran et al, 2006) is used in order to determine the design parameters for the EFK-Learning algorithm. Initially a set of random solutions or a set of particles are considered. A random velocity is given to each particle and they are flown through the problem space. Each particle has memory which is used to keep track of the previous best position and corresponding fitness. The best value of the position of each individual is stored as p_{id}. In other words, p_{id} is the best position acquired by an individual particle during the course of its movement within the swarm. It has another value called the p_{gd}, which is the best value of all the particles p_{id} in the swarm. The basic concept of the PSO technique lies in accelerating each particle towards its p_{id} and p_{gd} locations at each time step. The PSO algorithm used in this paper is defined as follows (Kiran, 2006):

1) Initialize a population of particles with random positions and velocities in the problem space.
2) For each particle, evaluate the desired optimization fitness function.
3) Compare the particles fitness evaluation with the particles p_{id} if current value is better than the p_{id} then set p_{id} value equal to the current location.
4) Compare the best fitness evaluation with the population's overall previous best. If the current value is better than the p_{gd}, then set p_{gd} to the particle's array and index value.
5) Update the particle's velocity and position as follows:
   The velocity of the ith particle of d dimension is given by:

$$\begin{aligned} v_{id}(k+1) &= c_0 v_{id}(k) \\ &\quad + c_1 \,\mathrm{rand}_1 (p_{id}(k) - x_{id}(k)) \\ &\quad + c_2 \,\mathrm{rand}_2 (p_{gd}(k) - x_{id}(k)) \end{aligned}$$

   The position vector of the ith particle of d dimension is updated as follows:

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k)$$

   where c_0 is the inertia weight, c_1 is the cognition acceleration constant and c_2 is the social acceleration constant.

6) Repeat the step 2 until a criterion is met, usually a sufficiently good fitness or a maximum number of iterations or epochs.

In case the velocity of the particle exceeds V_{max} (the maximum velocity for the particles) then it is reduced to V_{max}. Thus, the resolution and fitness of search depends on the V_{max}. If V_{max} is too high, then particles will move in larger steps and so the solution reached may not be the as good as expected. If V_{max} is too low, then particles will take a long time to reach the desired solution (Kiran, 2006). Due the above explained PSO are very suitable models of noisy problems, as the one we are considering.

C. Regressor Structure

We now discuss the choice of an appropriate number of delayed signals to be used in the training phase, due do a wrong number of delayed signals, used as regressors, could have a substantially negative impact on the training process, while a too small number implies that essential dynamics will not be modeled. Additionally, large number of regression terms increases the required computation time. Also, if too many delayed signals are included in the regression vector, it will contain redundant information. For a good behavior of the model structure, it is necessary to have both a sufficiently large lag space and an adequate number of hidden units. If the lag space is properly determined, the model structure selection problem is substantially reduced. There have been many discussions of how to determine the optimal embedding dimension from a scalar time series based in Takens' theorem (Cao, 1997). The basic methods, which are usually used to choose the minimum embedding dimension, are: (1) computing some invariant on the attractor, (2) singular value decomposition and (3) the method of false neighbors. However, a practical method to select the lag space is the one proposed by Cao (Cao, 1997) to determine the minimum embedding dimension; it overcomes most of the shortcomings of the above mentioned methodologies, like high dependence from design parameters and high computational cost, among others (Cao, 1997). In this paper, we adopt this technique for determination of the optimal regressor structure.

We consider a time series $x_1, x_2, ..., x_n$ and define a set of time-delay vectors as

$$\begin{aligned} y_i &= \left[\begin{array}{cccc} x_i & x_{i+\tau} & ... & x_{i+(d-1)\tau} \end{array} \right] \\ i &= 1, 2, ..., N-(d-1)\tau \end{aligned}$$

where d is the embedding dimension. This dimension is determined from the evolution of a function $E(d)$ defined as

$$\begin{aligned} E(d) &= \frac{1}{N-d\tau} \sum_{i=1}^{N-d\tau} \frac{\left\| y_i(d+1) - y_{n(i,d)}(d+1) \right\|}{\left\| y_i(d) - y_{n(i,d)}(d) \right\|} \\ i &= 1, 2, ..., N-d\tau \end{aligned}$$

where $n(i,d)$ is an integer such that $y_{n(i,d)}(d)$ is the nearest neighbor of $y_i(d)$ (Kennel et al, 1992). The minimum embedding dimension d_0+1 is determined when $E(d)$ stops changing for any d_0.

III. Characterization of Photovoltaic Power Generation

A photovoltaic module is composed from a number of solar cells connected in series; the modules can be connected in a series-parallel configuration to compose a PV array. The characterization of a photovoltaic module is given by its Voltage-Current curve under 1 sun of radiation (1000 W/m^2).

The photovoltaic module behavior will be predicted using neural networks trained with the Extended Kalman Filter. The module will be characterized by variable electrical output of a photovoltaic system such as electrical power, voltage and electric current depending on the input variables irradiance, temperature module and wind speed. We use a Recurrent Neural Network composed of 10 hidden neurons and 4 regressors and 1 external input.

Figura 2. Photovoltaics modules in the FI-UADY

Once we establish the main characteristics that influence

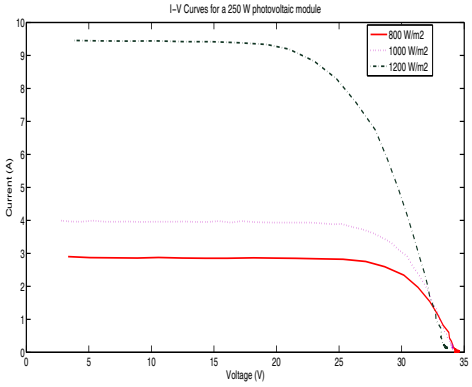

Figura 3. I-V curves for a 250 photovoltaic module under different irradiance conditions.

the generation of electricity in photovoltaic installations

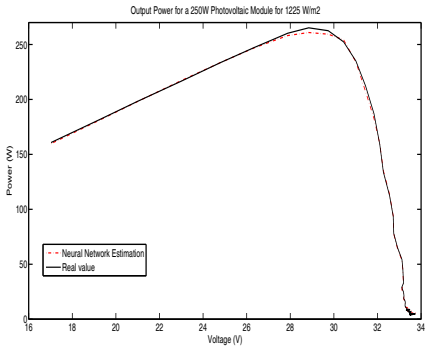

Figura 4. Neural Network approximation for Power curve for a 250 W PV module.

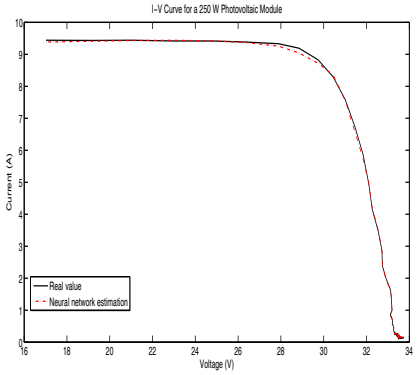

Figura 5. Neural Network approximation for I-V curve for a 250 W PV module.

we implement a Recurrent Neural Network to predict the generated power. Meteorological and electrical data that have been collected through measuring instruments installed in the PV system during the period of five years with readings every minute, this data will be subjected to signal processing in Matlab software to have the time series as a function of hours or minutes and get average values for each ten minutes. The time series corresponding to measurements taken every minute five days of physical quantities irradiance, module temperature, wind speed, voltage and DC current in a 1.6 kW installed photovoltaic array in the Energy Laboratory at the School of Engineering at the Autonomous University of Yucatan (Figure 2).

Due to random variations in weather conditions, power generation from renewable sources is constantly changing. Combining the forecast of solar irradiance, wind speed, module temperature and output power is a good way to improve the performance in scheduling of photovoltaic power. Reliability is one of the most important factors in smart grid operation, so constant monitoring and control is necessary to achieve this goal. An accurate forecast can improve the performance of intelligent controllers and management systems in the grid.

This project is implemented in the Mechatronics Building of the UADY Faculty of Engineering using the data obtained from an instrumented 1.6 kW array consisting of 16 modules as shown in Fig. 2; each module has 3 temperature sensors.To characterize the wind and solar potential, irradiance and wind speed data are collected from the meteorological station installed next to the array. The statistical values obtained from a one year analysis are applied to train the neural predictor.

To evaluate the performance of the PSO algorithm, we implement a neural network predictor for photovoltaic power generation, on the basis of Kalman filter training. As first stage we determine the optimal dimension of the regression vector; then, we select the number of hidden units for both hidden layers. The training is performed using minute data from five days. The neural network used is a RMLP trained with an PSO-EKF; the hidden layer logistic sigmoid activation functions (3), whose β was fixed in 1 and the output layer is composed of just one neuron, with a linear activation function. The initial values for the covariance matrices (R, Q, P) are determined using the PSO algorithm, with 200 as the maximum number of iterations, 4 generations, 3 particles and $c_1 = c_2 = 0.1$. The initial values for neural weights are randomly selected. The length of the regression vector is 6 because that is the order of the system, which is determined using the cao methodology.

The training is performed off-line, using a series-parallel configuration; for this case the delayed output is taken from the wind speed. The mean square error (MSE) reached in training is 5×10^{-4} in 200 iterations.

Figure 6 displays the computation of the minimum embedding dimension for each of the analyzed variables. We select 6 regressors to be included into the neural network input vector for the Photovoltaic Power. To train the HONN for each variable, we kept the following design parameters: 3 external inputs corresponding to irradiance, module temperature and wind speed, 15 units in the hidden layer, 1 neuron in the output layer, 200 iterations maximum, initial values for synaptic weights randomly selected in the range and MSE required to end the training less than 1x10^{-4}. The training was performed off-line, using a parallel configuration; for this case the delayed output is taken from the neural network output.

We used 720 samples to accomplish the network training. The results for are shown from Fig. 7; the forecasting is successfully done with a good prediction horizon.

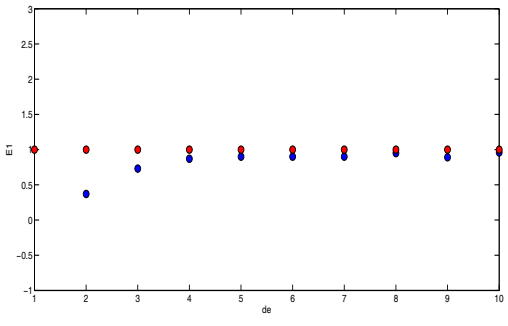

Figura 6. Embedded dimension for the generated power.

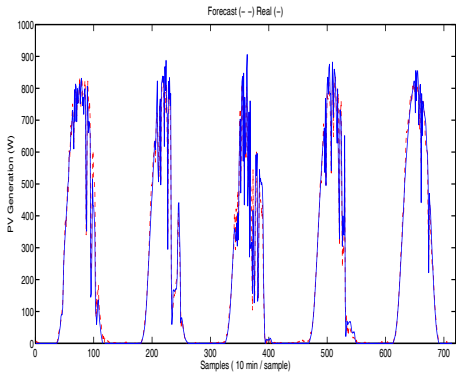

Figura 7. Photovoltaic energy generation forecasting

IV. Conclusions

This paper proposes the use of a RMLP trained with an PSO-EKF learning algorithm, to predict photovoltaic array output power with good results. The proposed method has a compact structure but taking into account the dynamic nature of the system which behavior is required to predict. The proposed neural identifier proves in our experiments to be a model that captures very well the complexity associated with important variables in smart grids operation. Future work on implementing higher order neural networks aims for the design of optimal operation algorithms for smart grids composed of wind and photovoltaic generation systems interconnected to the utility grid.

Acknowledgements: The authors thank the support of CONACYT Mexico, through Project 103191Y and CONACYT-Yucatan Government through Project FOMIX 170414.

Referencias

[1]

Alanis A. Y. , Sanchez E. N. and Loukianov A. G. (2007) "Discrete time adaptive backstepping nonlinear control via high order neural networks", *IEEE Transactions on Neural Networks*, vol. 18, no. 4, pp. 1185-1195 .

Alanis A. Y. , Simetti C. , Ricalde L. J. and Odone F. , "A Wind Speed Neural Model with Particle Swarm Optimization Kalman Learning", 9th International Symposium on Intelligent Automation and Control, Puerto Vallarta, Mexico, June, 2012.

Amoudi AA, Zang L., (2000) Application of radial basis function networks for solar-array modeling and maximum power-point prediction. Procedures Generation Transmission Distribution, vol. 147, pp.310-316.

Bonanno F., Capizzi G., Graditi G., Napoli C., Tina G., (2012), A radial basis function neural network based approach for the electrical characteristics estimation of a photovoltaic module, *Applied Energy* 97, pp. 956-961.

Cao L. (1997) "Practical method for determining the minimum embedding dimension of a scalar time series", *Physica D: Nonlinear Phenomena*, vol. 110, no. 1, pp. 43-50.

De Soto W., Klein S., Beckman W. (2006), Improvement and validation of a model for photovoltaic array performance, *Solar Energy*, vol. 80, pp. 78-88.

Kennedy J. and Eberhart R.C. , (1995) "Particle swarm optimization", *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 1942-1948.

Kennel M. , Brown R. and Abarbanel H. D. , "Determining the embedding dimension for phase-space reconstruction using a geometrical construction", Physical Review A, vol. 45, no. 6, pp. 3403-3411, March, 1992.

King D., Boyson W., Kratochvil J. (2004), Photovoltaic Array Performance Model, SAND2004-3535. Sandia National Laboratories.

King D., Gonzalez S., Galbraith G., Boyson W. (2007), Performance Model for Grid-Connected Photovoltaic Inverters, SAND2007-5036. Sandia National Laboratories.

Kiran R. , Jetti S. R. and Venayagamoorthy G. K. (2006)"Online Training of a Generalized Neuron with Particle Swarm Optimization", *Proceedings of the 2006 International Joint Conference on Neural Networks,* Vancouver, BC, Canada.

Lin C. J. , Chen C. H. andLin C. T. (2009) "A Hybrid of Cooperative Particle Swarm Optimization and Cultural Algorithm for Neural Fuzzy Networks and Its Prediction Applications", *IEEE Transactions on Systems, Man and Cybernetics- Part C*, vol. 39, no. 1, pp. 55-68.

Marion B., Anderberg M. (2000), PVWATTS – An Online Performance Calculator for Grid-Connected PV Systems, Procceedings SES Solar 2000 Conference, Madison, WI.

Norgaard M. , Poulsen N. K. , Ravn O. (2000) "Advances in Derivative-Free State Estimation for Nonlinear Systems", *Technical Report IMM-REP-1988-15* (revised edition), Technical University of Denmark, 2000.

Parsopoulos K. E. , Vrahatis M. N., *Particle Swarm Optimization*, IGI Global, 2010.

Sandrolini L., Artioli M., Reggiani U. (2010), Numerical method for the extraction of photovoltaic module double-diode model parameters through cluster analysis. *Applied Energy* 87, pp. 442-451.

T. Senjyu, A. Yona, N. Urasaki and T. Funabashi, "Application of recurrent neural network to long-term-ahead generating power forecasting for wind speed generator", Power Systems Conference and Exposition PSCE 2006, pp. 1260-1265, 2006.