

## Learning a wind farm power curve with a data-driven approach

Antonino Marvuglia<sup>1,\*</sup>, Antonio Messineo<sup>2</sup>

<sup>1</sup> CRP Henri Tudor/CRTE, 66 rue de Luxembourg, L-4002 Esch/Alzette, Luxembourg

<sup>2</sup> Faculty of Engineering & Architecture, Kore University of Enna, Italy

\*Corresponding author. Tel: +352 425991652, Fax: +352 425991555, E-mail: antonino.marvuglia@tudor.lu

**Abstract:** Improving the performance of prediction algorithms is one of the priorities in the wind energy research agenda of the scientific community. In a very simplistic approach, short-term predictions of wind power production at a given site could be generated by passing forecasts of meteorological variables (namely wind speed) through the so-called wind farm power curve, which links the wind speed to the power that is produced by the whole wind farm. However, the estimation of this conversion function is indeed a challenging task, because it is nonlinear and bounded, in addition to being non-stationary due for example to changes in the site environment and seasonality. Even for a single wind turbine the measured power at different wind speeds is generally different to the rated power, since the operating conditions on site are generally different to the conditions under which the turbine was calibrated (the wind speed on site is not uniform horizontally across the face of the turbine; the vertical wind profile and the air density are different than during the calibration; the wind data available on site are not always measured at the height of the turbine's hub).

The recent developments in data mining and evolutionary computation (EC) offer promising approaches to modelling the power curves of turbines. In this paper we use a self-supervised neural network called GMR (Generalized Mapping Regressor) to learn the relationship between the wind speed and the generated power in a whole wind farm. GMR is an incremental self-supervised neural network which can approximate every multidimensional function or relation presenting any kind of discontinuity. The approach used is a data driven one, in the sense that the relationship is learned directly from the data, without using any explicit physical or mathematical relationship between input and output space. The model is potentially applicable to any site, provided that a statistically consistent amount of wind and power data is available. The methodology allows the creation of a non-parametric model of the power curve that can be used as a reference profile for on-line monitoring of the power generation process, as well as for power forecasts.

The results obtained with the proposed approach are compared with another state-of-the-art data mining algorithm (namely, a feedforward Multi Layer Perceptron) showing that the algorithm provides fair performances if a suitable pre-processing of the input data is accomplished.

**Keywords:** Wind farm, Power curve, Data-driven, Neural network, Machine learning

### 1. Introduction

An interesting aspect of current research on wind power generation is the definition and implementation of accurate models to predict the energy output of a whole wind farm, rather than single wind turbines. In power systems a traditional generator is usually described as 'dispatchable', whereas wind generation is often referred to as 'non dispatchable'. Even though wind power generation has reached maturity in power systems, it is often still considered as a negative load because it is less predictable than thermal generation [1; 2].

Accurate wind power forecasting and prediction reduces the risk of uncertainty and allows for better grid planning and integration of wind into power systems.

There has been much debate and discussion in relation to the costs associated with wind power integration. A common conclusion is that as the levels of wind power penetration increase additional system balancing is required. Wind power forecasting and prediction tools are therefore invaluable because they enable better dispatch, scheduling and unit commitment of thermal generators, hydro plant and energy storage plant and more competitive market trading as wind power ramps up and down. Overall they reduce the financial and technical risk of uncertainty of wind power production for all electricity market participants.

Researchers have applied different methodologies in studying wind farms. In [3] the author built a model to predict the power produced by a wind farm using the data from the weather prediction model (HIRLAM) and the local weather model (WASP). In [4] a regression and a neural network (NN) model are compared in order to estimate a turbine's power curve. A novel approach for the analysis and modelling of wind vector fields was introduced by Goh et al. [5] and developed by Mandic et al. [6]. In these papers the wind vector is represented as a complex-valued quantity and wind speed and direction are modelled simultaneously. In [7] a new algorithm based on fuzzy logic was applied to estimate wind turbine power curve. In [8] a variety of different approaches have been used to build prediction models and characterize power curves of a wind farm by a nonlinear parametric model. In [9] parametric and non-parametric models have been applied for the same task and results have been compared. An extensive review of the existing wind speed and related generated power forecasting approaches can be found in [10].

The key issue addressed in this paper is the estimation of the relationship between the wind speed and a wind farm power output. This relationship is expressed as a power curve, which has a logistic function shape. However, as discussed in [8], the experimental power curve of a wind turbine (and of an entire wind farm as well) is not an ideal logistic function. All regions outside of the logistic curve represent either power losses or power gains. The presence of outliers and abnormal values might be due to several reasons: the presence of values of wind speed close to the cut-in or the cut-out speed of the installed wind turbines; environmental issues (blades affected by dirt, bugs and ice); shut-down due to maintenance or energy curtailment; control system issues; sensors malfunctions; pitch control malfunctions; unsuitable blade pitch angle setting; blade damage [8; 11]. Finally, the measured powers at different wind speeds are generally different than the rated power also because the operating conditions on site are generally different than the conditions under which the turbine was calibrated (the wind speed on site is not uniform horizontally across the face of the turbine; the vertical wind profile and the air density are different during the operation phase than during the calibration; the wind data on site are not always measured at the height of the turbine's hub). Moreover, due to the nonlinearity of the wind farm power curve, the uncertainty in its estimation dramatically amplifies the uncertainty contained in the wind speed forecasts. If the model of a power curve reflecting a normal status was available, the abnormal status of a turbine could be monitored and detected by this model.

The main motivation of this paper lies in the detection of the abnormalities of the wind turbines (and as a consequence also wind farms as a whole) power curve. The recent developments in data mining and evolutionary computation (EC) offer promising approaches to modelling turbines power curves. In this paper we use a self-supervised neural network called GMR (Generalized Mapping Regressor) [12; 13] to learn the relationship between the wind speed and the power generated in a wind farm. GMR is an incremental self-supervised neural network which can approximate every multidimensional function or relation presenting any kind of discontinuity. The approach used is a data driven approach, in the sense that the relationship is learned directly from the data, without using any explicit physical or mathematical relationship between input and output space. The methodology allows the creation of a non-parametric model of the power curve that can be used as a reference profile for on-line monitoring of the power generation process, as well as for power forecasts.

## **2. Methodology**

GMR is an incremental self-organizing neural network with chains (second layer weights) among neurons. The basic idea of GMR is to transform the function approximation problem

into a pattern recognition problem under an unsupervised framework. Hence, a coarse-to-fine covering strategy of the mapping is used. Suppose we want to model the mapping relationship between a set of inputs  $\mathbf{x}$  and a set of outputs  $\mathbf{y}$ . GMR algorithm works transforming the data mapping problem  $f: \mathbf{x} \rightarrow \mathbf{y}$  into a pattern recognition problem in the augmented space  $Z$  represented by vectors  $\mathbf{z} = [\mathbf{x}^T \mathbf{y}^T]^T$ , which are the inputs of GMR. In this space, the branches of the mapping become clusters which have to be identified.

The algorithm comprises four phases (training, linking, merging, and recalling) which are schematized in Fig. 1.

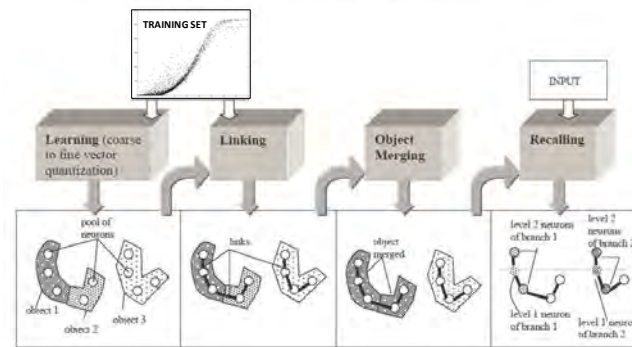


Fig. 1. The four phases of the GMR algorithm.

The **training phase** concerns the *vector quantization* of the  $Z$  space. The aim of vector quantization is, in practice, the reduction of the original data set to a small representative set of prototype vectors that are easier to manage [14]. During training, the augmented space is recovered by either creating neurons or adapting their weights according to the novelty of the input data. At the presentation of each input belonging to the training data set (TS), there are two possibilities: either creation of a new neuron (whose weight vector is equal to the input vector) or adaptation of the weight vector of the closest neuron (in input/weight space). Given a threshold  $\rho$  (*vigilance threshold*), a new neuron is created if the hyperspheres of radius  $\rho$ , centered in the already created weight vectors, do not contain the input. The parameter  $\rho$  is very important: it determines the resolution of training. Learning can be divided into two sub-phases: *coarse quantization* and *fine quantization*. The vigilance threshold used in the coarse quantization phase ( $\rho_1$ ) is higher than the one used in the fine quantization,  $\rho_2$ . In general, the first epoch (i.e. presentation of the entire TS) defines the number of neurons needed for mapping and the others adapt their weights for a better approximation. The neurons thus obtained identify the *objects*, which are compact sets of data in  $Z$ . The resulting neurons are called *object neurons*. In the second sub-phase, at first a pre-processing is required for labelling each neuron with the list of the input data which had the neuron as winner (i.e. whose weight vector is the closest to the input vector); it can be accomplished by presenting all data (*production phase*) to GMR and recording the corresponding winning neurons. At the end, for each neuron a list of the inputs for which it won is stored. This list represents the *domain* of the object neuron. Every list is considered as the TS for a subsequent training which takes place separately (and in parallel) for each object domain. At the end, the neural network is made up of the neurons generated by the secondary learning phases (*final neurons*), labelled as belonging to an object by the corresponding object neuron.

The next phase is the **linking phase**. Neurons' linking is accomplished by computing the second layer weights, which are discrete and equal to zero in the absence of a link. A link is computed at the presentation of each piece of data from the TS. The technique used in this paper to perform the linking phase exploits the direction of the principal component of the domain data (i.e. the direction corresponding to the highest eigenvalue of the autocorrelation matrix of the domain data) [12]. This direction is here referred to as *domain principal*

*direction* (PD). For each data point, the weights are sorted according to the Euclidean distance from it, and the winning neuron is determined. It is then linked to another neuron chosen in a subset of neurons (*candidate neurons*). The subset was here determined by defining in advance a number  $k$  of nearest neighbours of the input. Then, for each candidate, the absolute value of the scalar product between its PD and the winner's PD is evaluated. The winner is linked to the candidate yielding the maximum scalar product (i.e., the candidate whose PD is closest in direction to the winner's PD). This approach is justified by the fact that clusters with similar shapes have to be connected.

In the *merging phase*, GMR checks whether different objects are linked. If they are, the objects are merged. The *recalling phase* replaces the neurons in the reduced manifold with Gaussians representing the domain. Their parameters are estimated by the maximum likelihood (ML) technique. This phase is essentially a process of Gaussian labelling followed, if required, by an interpolation step. A more detailed description of GMR is contained in [12].

### 3. Results

The data used for the case study refer to a wind farm (whose location is not disclosed for confidentiality reasons) and the available data set comprises a wind speed time series (one whole year) measured by an anemometer located at 50 m above the ground level (a.g.l.), as well as the power produced by each of the wind turbines, collected by a SCADA (Supervisory Control and Data Acquisition) systems at the wind farm. Since the aim of the study was modelling the power curve of the wind farm as a whole, the sum of the power produced by all the turbines was used for the application of the proposed algorithm. A training and a test set containing respectively 90% and 10% of the entire time series available (randomly selected across the whole data set) were created. In order to speed up the training, a sampling of one datum out of three was then made on the training set (TS), so to select a smaller subset (that was used as the actual TS). Before starting the training phase, the data which had been clearly recorded erroneously, due to a malfunctioning of the SCADA system, were filtered out (namely, data with an abscissa greater than the cut-in speed of the wind turbines and with a null ordinate). The remaining data were normalized in the interval  $[-1; +1]$  and denoised using the Kernel Principal Component Analysis (KPCA) technique with a bandwidth of the Gaussian kernel  $\sigma = 0.4$ . KPCA is a *generically nonlinear* signal processing technique; it is often used for denoising in image applications [15]. The denoised TS used to train GMR contains 2828 observations. A test data set containing 876 observations has been used in the recalling phase. Figure 2 shows the data before (a) and after (b) the normalization and denoising procedure. The data showed in Fig. 2(a) have been previously rescaled, only for visualization purposes, in order to protect data confidentiality.

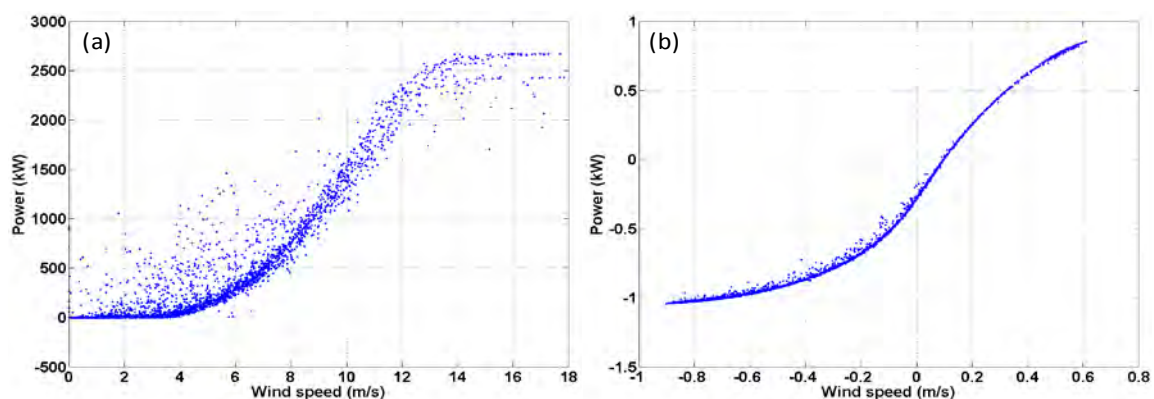


Fig. 2. Training set data before (a) and after (b) normalization and denoising.

A number of nearest neighbours  $k=4$  was used during the linking phase. In the rough phase the vigilance threshold was set to  $\rho_1 = 0.5$  and in the fine tuning phase  $\rho_2 = 0.025$  was used. The number of final neurons is 101 and the number of object neurons is 26. During the recalling phase, GMR was used both without interpolation and with Gaussian interpolation. Figure 3 shows the results of the linking and merging phases of GMR. A zoom of the part of the curve contained in the dashed box is showed in the upper left and lower right corners of Fig. 3(b). In the lower right corner different objects are represented with different symbols to improve readability.

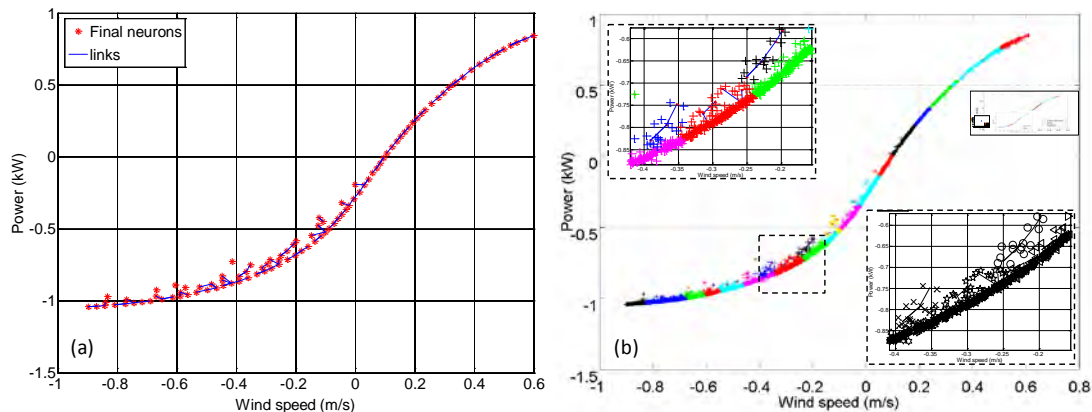


Fig. 3. Results of the linking (a) and merging (b) phases of GMR.

The same TS used for the GMR model was also used to train a feedforward Multi Layer Perceptron (MLP). The learning algorithm used is backpropagation with momentum [16]. The network has two layers of neurons, the hidden layer with 10 and the output layer with 1 neuron (since the output is mono-dimensional). A logistic sigmoid function and a linear function were respectively used as the activation function of the first and the second layer. Figure 4 shows the results of the modelling obtained both on the TS and test set data, with the two models, while Table 1 summarizes their prediction accuracy.

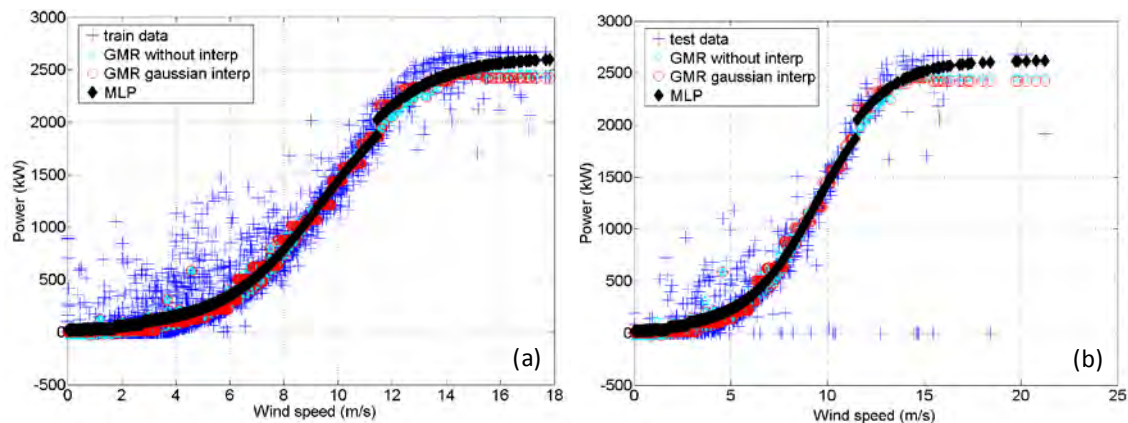


Fig. 4. Results of the GMR and MLP predictions on the training set (a) and the test set (b).

Their performances have been assessed computing the mean absolute error (MAE) and the symmetric mean absolute percentage error (sMAPE), using the test set data. The respective standard deviations (Std dev, in Table 1) have also been computed.

The sMAPE measure computes the absolute error in percent between the absolute value of the actual observation  $y_t$  and the absolute value of the forecast  $\hat{y}_t$  across all observations  $t$  of the test set of size  $n$  [17]:



$$sMAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2} * 100 \quad (1)$$

The MAE of MLP is worse than the one of the two GMR models, because GMR recognizes that the “thickness” of the data cloud around the average logistic sigmoid trend actually contains some information content and it is not just due to noise (see the links in Fig. 3(a)), whereas the MLP follows the average trend of the data, thus finding a smooth curve profile.

Table 1. Prediction accuracy of the models applied (GMR and MLP).

	MLP	GMR gaussian interp	GMR without interp
MAE (kW)	1990.60	436.11	458.63
Std dev of AE (kW)	2515.40	918.34	906.40
sMAPE (%)	68.03	64.19	68.35
Std dev of sMAPE (%)	76.25	73.73	76.40
minimum AE	71.79	1.02	0.47
maximum AE	9159.40	8585.10	8585.10

### 3.1. On-line monitoring by residual approach and control charts

Once the two non-parametric models mentioned above have been trained, they can be used to characterize the wind farm power in normal conditions, and therefore they can serve as an on-line wind farm power generation profile. The residual control chart techniques (statistical quality control) [18] are used to analyze residuals between model predicted power and observed power. The control chart approach allows the residuals and their variations to be monitored, thus detecting abnormal conditions of a turbine.

The means of the residuals obtained on the TS ( $\mu_{Train}$ ) and the test set ( $\mu_{Test}$ ), as well as their standard deviations ( $\sigma_{Train}$ ;  $\sigma_{Test}$ ) were computed. Once  $\mu_{Train}$  and  $\sigma_{Train}$  are known, the upper and lower control limits of the control chart can be computed and used to detect the anomalies. Control limits for the control chart can then be derived using Eq. (2) [18]:

$$UCL_1 = \mu_{Train} + \eta \frac{\sigma_{Train}}{\sqrt{N_{Test}}}; \quad LCL_1 = \mu_{Train} - \eta \frac{\sigma_{Train}}{\sqrt{N_{Test}}} \quad (2)$$

$N_{Test}$  is the number of points in the test data set, but it can be adjusted to make the control chart less sensitive to the data variability and thus reduce the risk of false alarms. The parameters in Eq. (2) could be adjusted dynamically, based on operations of individual turbines. In the application described in this paper,  $N_{Test}$  was set equal to 10, while  $\eta$  was set to the widely used value  $\eta = 3$  [9]. If  $\mu_{Test}$  is above  $UCL_1$  or below  $LCL_1$ , the power generation process at the generic sampling time  $y_{TestSet} = [y(i), \hat{y}(i)]$  is considered to be deficient (or “out-of-control”), otherwise it is not considered abnormal, i.e. it is considered “in-control”. Similarly, the control limits for  $\sigma_{Test}^2$  can also be calculated to detect out-of-control points, using control limits defined as a function of the variance of  $\sigma_{Train}^2$  [8; 9; 18]. Figure 5 shows the out-of-control points detected in the TS by GMR and the MLP on the basis of the limits defined by Eq. (2). Out of the 876 points of the test set, GMR detected 183 out-of-control points, while the MLP network detected 136 of them. It is possible to notice that GMR in this case is more conservative in the part of the curve closer to the cut-off wind speed. This is due to the choice of the parameters  $\eta$  and  $N_{Test}$ , but also to the fact that the MLP model tends to find a smooth solution, passing between the two data clouds in the last part of

the curve, whereas GMR remains closer to a constant value of the output power for wind speeds equal to or higher than the cut off speed, as it should be in case of ideal functioning of the turbines. It is also possible to notice that GMR detects some of the points above the left tail of the curve as “in-control” points, while MLP labels them as “out-of-control”. This is due to the fact that GMR finds some links and, as a consequence, some branches of the mapping, also in this zone of the space (see Fig. 3). This is reasonable and in line with the results found in [9] using a parametric model and a non-parametric model based on the k-nearest neighbour (k-NN) algorithm. In fact, one of the main advantages of GMR, compared with feedforward neural networks, lays into the fact that, while for many functional (i.e. single-valued) approximation problems feedforward network can work well by minimizing a sum-of-squares error function, they can actually give rise to high imprecision in presence of multi-valued mappings, or mappings whose structure varies for different regions of the input space. The prominent feature of GMR is its capability to output all the solutions and their corresponding mapping branches.

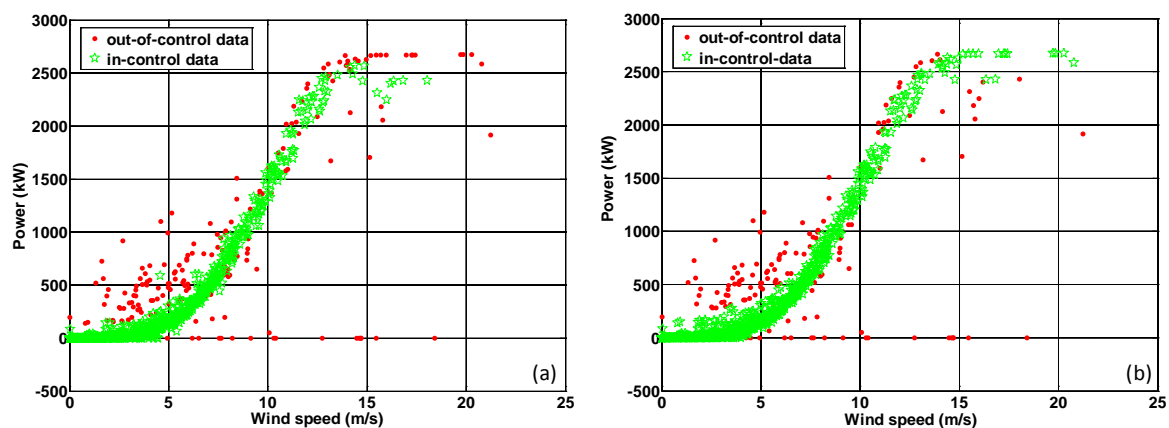


Fig. 5. Data (within the test set) detected as “in-control” and “out-of-control” according to the limits defined in Eq. (2), using GMR with Gaussian interpolation (a) and MLP (b).

#### 4. Conclusions

The paper presents an application of a data-driven technique for mapping the power curve of a whole wind farm and shows its utilization for the creation of quality control charts. The results obtained with the proposed methodology are compared with those obtained with a “traditional” non-parametric method: a MLP neural network. A few other applications have already been presented in the literature, dealing with the same topic. The originality of this paper is twofold: on one hand, its application to the modelling of the power curve of an entire wind farm, instead of a single wind turbine; on the other hand, the utilization of GMR, which is an incremental self-organizing competitive neural network with adaptive linking among neurons, able to approximate every kind of mapping (function or relation) in both senses, i.e.  $M(x, y): x \in \mathcal{H}^n \leftrightarrow y \in \mathcal{H}^n$ . The models can be used as the reference power curve (on-line profile) for monitoring the performances of a wind farm as a whole. In a real, on-line application, the models will be updated using the most current operational data.

The case study here presented demonstrated that the control chart approach produces satisfactory results in monitoring power curves. In future research, other data mining algorithms will be assessed for enhancing accuracy of non-parametric models. In particular, the application of Bayesian networks will be considered to identify links between anomalies and the specific reasons causing them, which is not possible simply using control charts.

## References

- [1] R.M.G. Castro, A.F.M. Ferreira, A comparison between chronological and probabilistic methods to estimate wind power capacity credit, *IEEE Trans Pow Syst* 16(4), 2001, pp. 904-909.
- [2] P. Børre Eriksen, T. Ackermann, H. Abildgaard, P. Smith, W. Winter, J. Rogríguez García, System operation with high wind penetration, *IEEE Pow Energ Magazine*, November/December 2005.
- [3] L. Landberg, Short-term prediction of the power production from wind farms, *Journal of Wind Engineering and Industrial Aerodynamics* 80(1-2), 1998, pp. 207-220.
- [4] S. Li, D.C. Wunsch, E. O'Hair, M.G. Giesselmann, Comparative analysis of regression and artificial neural network models for wind turbine power curve estimation, *J Solar Energ Engineering* 123(4), 2001, 327-332.
- [5] S.L. Goh, M. Chen, D.H. Popović, K. Aihara, D. Obradovic, D.P. Mandic, Complex-valued estimation of wind profile, *Renew Energ* 31, 2006, pp. 1733-1750.
- [6] D.P. Mandic, S. Javidi, S.L. Goh, A. Kuhc, K. Aihara, Complex-valued prediction of wind profile using augmented complex statistics, *Renew Energ* 34, 2009, 196-201.
- [7] T. Üstütaş, A.D Şahin, Wind turbine power curve estimation based on cluster center fuzzy logic modelling, *Journal of Wind Engineering and Industrial Aerodynamics* 96, 2008, pp. 611-620.
- [8] A. Kusiak, H. Zheng, Z. Song. Models for monitoring wind farm power, *Renew Energ* 34, 2009, pp. 583-590.
- [9] A. Kusiak, H. Zheng, Z. Song, On-line monitoring of power curves, *Renew Energ* 34, 2009, pp. 1487-1493.
- [10] M. Lei, L. Shiyang, J. Chuanwen, L. Hongling, Z. Yan, A Review on the Forecasting of Wind Speed and Generated Power, *Renew Sust Energ Rev* 13(4), 2009, pp. 915-920.
- [11] B. Bell, Individual wind turbine and overall power plant performance verification, San Diego, CA, 2008.
- [12] Cirrincione G, Cirrincione M, Lu C, Van Huffel S. A novel neural approach to inverse problems with discontinuities (the GMR neural network), *Proceedings of 2003 Int. J Conf Neural Networks (IJCNN'03)*, Portland, Oregon, pp. 3106-3111.
- [13] C. Lu, The Generalised Mapping Regressor (GMR) neural network for inverse discontinuous problems, *MSc Thesis*, 2000, Katholieke Universiteit Leuven (Belgium).
- [14] R.M. Gray, Vector quantization, *IEEE ASSP Magazine* 4(2), 1984, pp. 4-29.
- [15] A.R. Teixeira, A.M. Tomé, K. Stadlthanner, E.W. Lang. KPCA denoising and the pre-image problem revisited, *Digit Signal Process* 18, 2008, pp. 568-580.
- [16] M.T. Hagan, H.B. Demuth, M.H. Beale, *Neural network design*, PWS Publishing, 1996.
- [17] S. Haykin, *Neural Networks – A Comprehensive Foundation*, Prentice Hall, Upper Saddle River, 1999.
- [18] D.C. Montgomery, *Introduction to statistical quality control*, John Wiley & Sons, New York, 5<sup>th</sup> ed. 2005.