Load Forecasting with Support Vector Machines and Semi-Parametric Method

J.A Jordaan[1] and A. Ukil[2]

[1]Tshwane University of Technology
Staatsartillerie Road, Pretoria, 0001, South Africa
jordaan.jaco@gmail.com
[2]ABB Corporate Research
Segelhofstrasse 1K, Baden Daettwil, CH-5404, Switzerland
abhiukil@yahoo.com

Abstract. A new approach to short-term electrical load forecasting is investigated in this paper. As electrical load data are highly non-linear in nature, in the proposed approach, we first separate out the linear and the non-linear parts, and then forecast using the non-linear part only. Semi-parametric spectral estimation method is used to decompose a load data signal into a harmonic linear signal model and a non-linear trend. A support vector machine is then used to predict the non-linear trend. The final predicted signal is then found by adding the support vector machine predicted trend and the linear signal part. The performance of the proposed method seems to be more robust than using only the raw load data. This is due to the fact that the proposed method is intended to be more focused on the non-linear part rather than a diluted mixture of the linear and the non-linear parts as done usually.

1 Introduction

Short-term load forecasting (STLF) is used to estimate the electrical power demand. Accurate STLF has a significant impact on a power system's operational efficiency. Many decisions, such as spinning reserve allocation, real time generation control and security analysis, are based on STLF [1]. This also means that accurate STLF has economic and security related advantages. This allows electrical companies to commit their own production resources in order to optimise energy prices, which leads to cost savings and to increased power system security and stability [2].

In the last few years, several techniques for short- and long-term load forecasting have been discussed, such as Kalman filters, regression algorithms, artificial neural networks (ANN) [2,3] and fuzzy neural networks [1]. Another method of load forecasting is to use support vector machines (SVM). SVM is a powerful methodology for solving problems in non-linear classification, function estimation and density estimation [4]. Load forecasting is an application of function estimation (regression). In the SVM solution method one solves convex optimisation problems, typically quadratic programs with a unique solution, compared to neural network multi-layer perceptrons (MLP) where the cost function could have multiple local minima.

In [5] the authors used ANN for STLF and the training time for the ANN was quite long compared to that of SVM. For some cases the ANN performed very poorly. For the MLP, it is hard to estimate the optimal number of neurons needed for a given task [6]. This often results in over- or underfitting. This is because for MLP we choose an appropriate structure, the number of hidden layer neurons in the MLP. Keeping the confidence interval fixed in this way, we minimise the training error, i.e., we perform the empirical risk minimisation (ERM). These can be avoided using the SVM and the structural risk minimisation (SRM) principle [7]. In SRM, we keep the value of the training error fixed to zero or some acceptable level and minimise the confidence level. This way, we structure a model of the associated risk and try to minimise that. The result is the optimal structure of the SVM.

In this paper we introduce a new approach to load forecasting using SVM. Other load forecasting approaches using SVM include [7] where genetic algorithms were used in combination with SVM. The genetic algorithms were used to determine proper values for the free parameters of the SVM. In [8] the authors used regression trees to select the important input variables and to partition the input variable space for use in the SVM.

The layout of the paper is as follows: in section 2 we introduce the new proposed method of treating the load prediction problem, section 3 shows the numerical results obtained, and the paper ends with a conclusion.

2 Semi-Parametric Method

When a model is fitted to the data taken from a power system, we many time have components in the data that are not directly part of the process we want to describe. If a model is fit to the data as it is, then the model parameters will be biased. We would have better estimates of the model parameters if the unwanted components (nuisance, bias, or non-linear components) are first removed. This method has been used successfully in the field of spectral estimation in power systems when we analyse the measured signals on power transmission lines [9].

The new method we propose is to separate the load data into linear and non-linear (trend) components. This method is called the Semi-Parametric method for harmonic content identification. We assume that there is an underlying linear part of the load data that could be represented with a sum of n damped exponential functions

$$y_L(k) = \sum_{i=1}^{n} A_i e^{j\theta_i} e^{(j2\pi f_i + d_i)Tk} , \tag{1}$$

where $y_L(k)$ is the k-th sample of the linear part of the load signal, A is the amplitude, θ is the phase angle, f is the frequency, d is the damping and T is the sampling period. Since we work only with real signals, the complex exponential functions come in complex conjugate pairs (see eq. (16)). The equivalent Auto Regressive (AR) model of (1) is given by

$$y_L(k) = -\sum_{i=1}^{n} x_i y_L(k-i), \quad k = n+1 \ldots n+m, \tag{2}$$

with model parameters x_i, model order n, and $n+m$ number of samples in the data set. The model parameters x_i and model order n has to be estimated from the data.

We propose the following model to separate the linear and non-linear parts [9,10]:

$$y_L(k) = y(k) + \mathit{\Delta} y(k), \tag{3}$$

where $y(k)$ is the measured signal sample, $\mathit{\Delta} y(k) = E[\mathit{\Delta} y(k)] + \epsilon(k)$ is the residual component consisting of a non-zero time varying mean $E[\mathit{\Delta} y(k)]$ (nuisance or bias component) and noise $\epsilon(k)$. The mean of the residual component is represented by a Local Polynomial Approximation (LPA) model [11]. y_L is then the required linear signal that can be represented with a sum of damped exponentials (1). The LPA model is a moving window approach where a number of samples in the window are used to approximate (filter) one of the samples in the window (usually the first, last or middle sample). The LPA filtering of data was made popular by Savitsky and Golay [12,13].

By substituting eq. (3) in (2) we obtain

$$y(k) + \mathit{\Delta} y(k) = -\sum_{i=1}^{n} x_i \left[y(k-i) + \mathit{\Delta} y(k-i)\right]. \tag{4}$$

In matrix form, for $n+m$ samples, the model is

$$\mathbf{b} + \mathit{\Delta}\mathbf{b} = -\mathbf{A}\mathbf{x} - \mathit{\Delta}\mathbf{A}\mathbf{x}, \tag{5}$$

where

$$\mathbf{b} = \begin{bmatrix} y(n+1) \\ y(n+2) \\ \vdots \\ y(n+m) \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} y(n) & y(n-1) & \cdots & y(1) \\ y(n+1) & y(n) & \cdots & y(2) \\ \vdots & \vdots & \ddots & \vdots \\ y(n+m-1) & y(n+m-2) & \cdots & y(m) \end{bmatrix}, \tag{6}$$

$$\mathit{\Delta}\mathbf{b} = \begin{bmatrix} \mathit{\Delta} y(n+1) \\ \mathit{\Delta} y(n+2) \\ \vdots \\ \mathit{\Delta} y(n+m) \end{bmatrix}, \mathit{\Delta}\mathbf{A} = \begin{bmatrix} \mathit{\Delta} y(n) & \mathit{\Delta} y(n-1) & \cdots & \mathit{\Delta} y(1) \\ \mathit{\Delta} y(n+1) & \mathit{\Delta} y(n) & \cdots & \mathit{\Delta} y(2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathit{\Delta} y(n+m-1) & \mathit{\Delta} y(n+m-2) & \cdots & \mathit{\Delta} y(m) \end{bmatrix}. \tag{7}$$

The matrix signal model (5) can be rewritten in a different form and represented as

$$\mathbf{A}\mathbf{x} + \mathbf{b} + [\mathit{\Delta}\mathbf{b}\, \mathit{\Delta}\mathbf{A}] \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} = \mathbf{0}, \tag{8}$$

or

$$\mathbf{A}\mathbf{x} + \mathbf{b} + \mathbf{D}(\mathbf{x})\,\Delta\mathbf{y} = \mathbf{0}, \tag{9}$$

where the following transformation has been used:

$$[\Delta\mathbf{b}\,\Delta\mathbf{A}]\begin{bmatrix}1\\ \mathbf{x}\end{bmatrix} = \mathbf{D}(\mathbf{x})\,\Delta\mathbf{y} \tag{10}$$

or

$$\left[\begin{bmatrix}\Delta y(n+1)\\ \Delta y(n+2)\\ \vdots\\ \Delta y(n+m)\end{bmatrix}\begin{bmatrix}\Delta y(n) & \Delta y(n-1) & \cdots & \Delta y(1)\\ \Delta y(n+1) & \Delta y(n) & \cdots & \Delta y(2)\\ \vdots & \vdots & \ddots & \vdots\\ \Delta y(n+m-1) & \Delta y(n+m-2) & \cdots & \Delta y(m)\end{bmatrix}\right]\begin{bmatrix}1\\ x_1\\ x_2\\ \vdots\\ x_n\end{bmatrix} =$$

$$\begin{bmatrix}x_n & \cdots & x_1 & 1 & 0 & \cdots & 0\\ 0 & x_n & \cdots & x_1 & 1 & \ddots & \vdots\\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0\\ 0 & \cdots & 0 & x_n & \cdots & x_1 & 1\end{bmatrix}\begin{bmatrix}\Delta y(1)\\ \Delta y(2)\\ \vdots\\ \Delta y(n+m)\end{bmatrix}. \tag{11}$$

If the number of parameters in vector $\mathbf{x}$, (model order n) is not known in advance, the removal of the nuisance component and noise from the signal $y(k)$ is equivalent to estimating the residual $\Delta y(k)$ and the model order n while fulfilling constraints (9). To solve the semi-parametric model, the second norm of the noise, plus a penalty term which puts a limit on the size of vector $\mathbf{x}$ is minimised. The following optimisation problem should be solved:

$$\begin{aligned}\underset{\mathbf{x},\Delta\mathbf{y}}{min}\left\{\frac{1}{2}\|\epsilon\|_2^2 + \frac{\mu}{2}\mathbf{x}^T\mathbf{x}\right\} &= \underset{\mathbf{x},\Delta\mathbf{y}}{min}\left\{\frac{1}{2}\left(\Delta\mathbf{y} - E[\Delta\mathbf{y}]\right)^T\left(\Delta\mathbf{y} - E[\Delta\mathbf{y}]\right) + \frac{\mu}{2}\mathbf{x}^T\mathbf{x}\right\}\\ &= \underset{\mathbf{x},\Delta\mathbf{y}}{min}\left\{\frac{1}{2}\Delta\mathbf{y}^T\mathbf{W}\Delta\mathbf{y} + \frac{\mu}{2}\mathbf{x}^T\mathbf{x}\right\} \end{aligned} \tag{12}$$

subject to the equality constraints $\mathbf{A}\mathbf{x} + \mathbf{b} + \mathbf{D}(\mathbf{x})\,\Delta\mathbf{y} = \mathbf{0}$,

where

$$\mathbf{W} = (\mathbf{I} - \mathbf{S})^T(\mathbf{I} - \mathbf{S}), \tag{13}$$

$\mathbf{I}$ is the identity matrix, $\mathbf{S}$ is the LPA smoothing matrix used to estimate $E[\Delta y(k)]$ as $\mathbf{S}\Delta\mathbf{y}$, and μ is the Ridge regression factor used to control the size of vector $\mathbf{x}$ [14,15].

2.1 Estimation of the Harmonic Components

The next step is then to calculate the parameters of the harmonic components in eq. (1). We do this as follows [16,17]:

1. The coefficients x_i are those of the polynomial

$$\underline{H}\left(z\right) = 1 + \sum_{i=1}^{n} x_i \underline{z}^{-i}, \tag{14}$$

where $\underline{z}$ is a complex number

$$\underline{z} = e^{(j2\pi f + d)T}. \tag{15}$$

By determining the n roots, $\underline{z}_i$, $i = 1, 2, \ldots, n$, of eq. (14), and using eq. (15) for $\underline{z}$, we can calculate the values of the n frequencies and dampings of the harmonic components. It should be noted that we are using complex harmonic exponentials to estimate the input signal's linear component. However, the signals we measure in practice are real signals of the form

$$y_L\left(k\right) = \sum_{i=1}^{n/2} 2A_i e^{d_i Tk} cos\left(2\pi f_i Tk + \theta_i\right), \tag{16}$$

where A_i, θ_i, f_i and d_i are the same as defined for the complex harmonics in eq. (1). Therefore if we expect to have $\frac{n}{2}$ components in our real signal, there will be n complex harmonic exponentials, and thus will the AR model order be n. The complex harmonic exponentials will then always come in $\frac{n}{2}$ complex conjugate pairs.

2. To determine the n amplitudes A_i and phase angles θ_i, we substitute the linear component $y\left(k\right) + \mathit{\Delta} y\left(k\right)$, and the estimated frequencies and dampings into eq. (1). We obtain an overdetermined system of linear equations of $N \times n$ that can be solved using the least squares method:

$$y\left(k\right) + \mathit{\Delta} y\left(k\right) = \sum_{i=1}^{n} A_i e^{j\theta_i} e^{(j2\pi f_i + d_i)Tk}, \; k = 1, 2, \ldots, N. \tag{17}$$

2.2 Non-Linear Part

The non-linear part (plus the noise), which could represent trends or other non-linearities in the power system, is then given by

$$y_N(k) = y(k) - y_L(k)\,, \tag{18}$$

where $y_N(k)$ is the k-th non-linear signal sample and $y(k)$ is the measured load sample. This non-linear part is then used to train a support vector machine. After the training is complete, the SVM could be used to predict the non-linear part. The linear part is calculated from the signal model (1), which is then added to the non-linear part to obtain the final predicted load values.

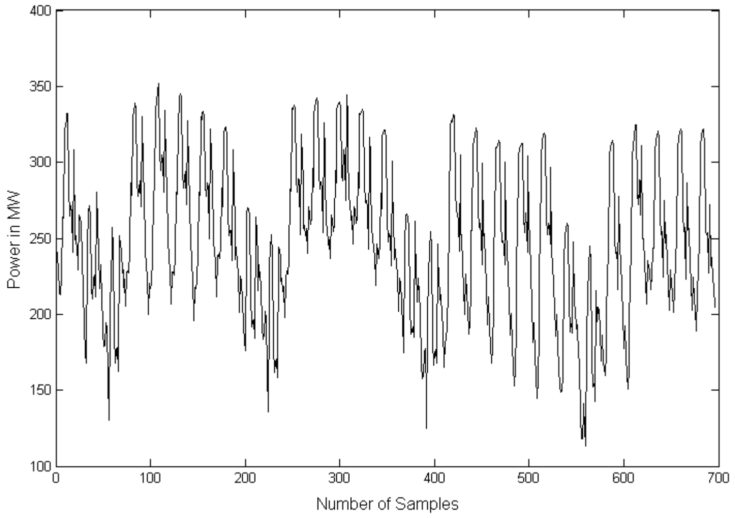

Fig. 1. Load of a Town

3 Numerical Results

For this experiment we tested many linear and non-linear SVM regression methods. For the results we show only the non-linear Radial Basis function kernel SVM. For the implementation we used MATLAB [18] and the Least Squares Support Vector Machines toolbox from [4].

Before the support vector machine is trained with the load data, some preprocessing is done on the data. First the data is normalised by dividing by the median of the data. This is because statistically the median is least affected by data variabilities. Therefore, after prediction, the signal must be de-normalised by multiplying it again with the median. Then the normalised data is separated into a linear and a non-linear part.

The test data, shown in Fig. 1, contained 29 days of load values taken from a town at one hour intervals. This gives a total number of 696 data samples. We removed the last 120 data samples from the training set. These samples would then be used as testing data. Each sample is also classified according to the hour of the day that it was taken, and according to which day. The hours of the day are from one to 24, and the days from one (Monday) to seven (Sunday).

The data fed into the support vector machine could be constructed as follows: to predict the load of the next hour, load values of the previous hours are used. We can additionally also use the day and hour information. For example, this means that as inputs to the SVM, we could have k consecutive samples, and two additional input values representing the hour and day of the predicted $k+1-th$ sample. The SVM will then predict the output of the $k+1-th$ sample. We can also call the value of k : a delay of k samples.

To evaluate the performance of the different SVMs, we define a performance index, the Mean Absolute Prediction Error (MAPE):

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|t_i - p_i|}{t_i} \times 100\,, \tag{19}$$

where t_i is the $i-th$ sample of the true (measured) value of the load, p_i is the predicted load value of the SVM, and N is the total number of predicted samples. For this experiment, the last 24 hours of the 120 removed samples in the load set was used to test the different SVMs. Different values of delay was used, from six until 96. We also tested the prediction method without splitting the data into linear and non-linear parts, and compared it to the proposed new method. The results of the performance index for each of the methods are shown in Table 1. From the statistics in the Table it seems that the method without separating the data into different components performs slightly better than the method separating the data. In general both methods performed well, but there were occasions where the method without splitting the data had a very bad performance, eg. from delay 60 until 96. This can be seen in Fig. 2 where the bad performance is illustrated for a delay of 78. Also for a delay of 78, in Fig. 3 the method with separating the data is shown. It can be seen that this method produces better results. It was found that the method without separating the

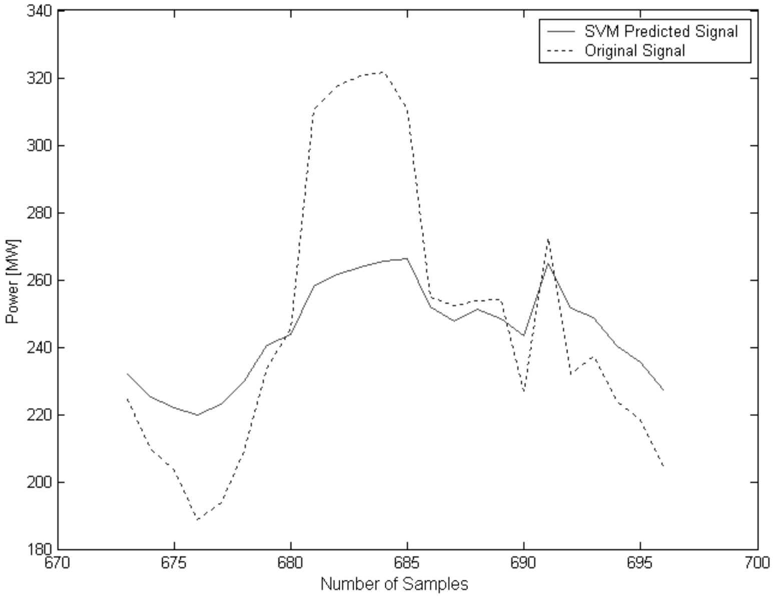

Fig. 2. Bad Performance of Method without Separating Data: Delay of 78

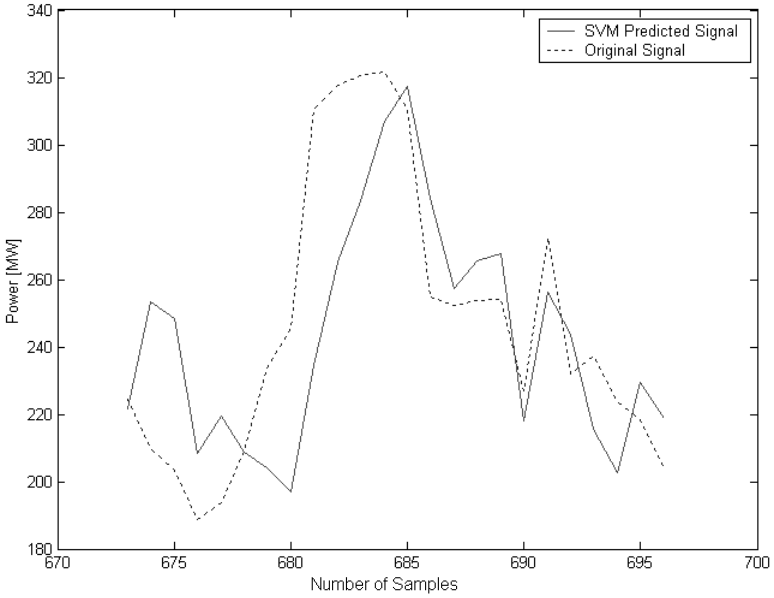

Fig. 3. Performance of Method with Separated Data: Delay of 78

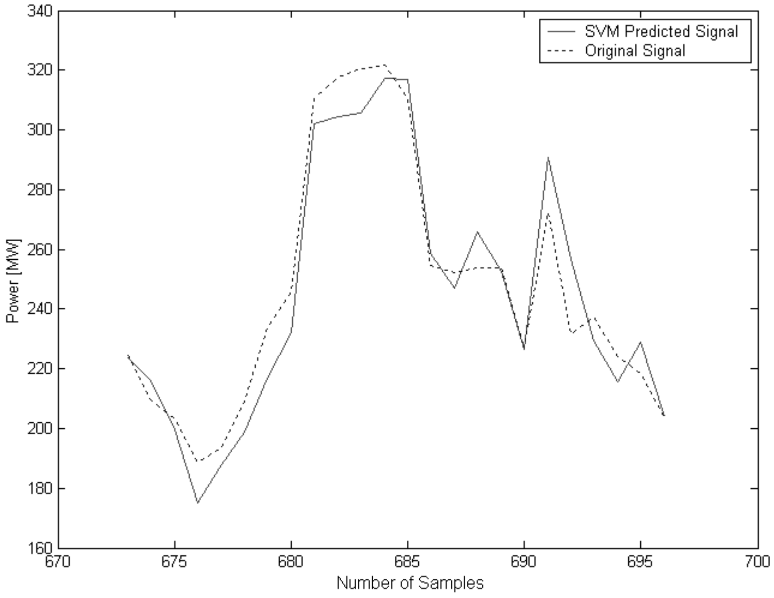

Fig. 4. Best Performance of Method with Separated Data: Delay of 15

Table 1. MAPE for the Two Methods

Delay	Separated into	Non-Separated
	Linear / Non-Linear	
6	3.7767	2.9794
15	3.5924	3.4402
24	4.9158	4.7659
33	5.3157	5.8301
42	5.4615	5.5794
51	5.3634	4.5498
60	6.0711	5.4701
69	7.3251	6.9728
78	9.5568	8.5627
87	11.1496	10.0967
96	14.3506	11.7320
Average MAPE	6.9890	6.3617
Median MAPE	5.4615	5.5794

data was more sensitive to the parameters of the SVM training algorithm, than the method separating the data. The best network without splitting the data has a delay of 15 and is shown in Fig. 4.

4 Conclusion

The Semi-Parametric method for separating the electric load into a linear and non-linear trend part was introduced. A support vector machine was then used to do load forecasting based only on the non-linear part of the load. Afterwards the linear part was added to the predicted non-linear part of the support vector machine. We compared this method to the usual method without splitting the data. On average the method without splitting the data gave slightly better results, but there were occasions where this method produced very bad results and it was also very sensitive to the SVM training parameters. The newly introduced method generally performed very well (even in the situations where the method without separating the data produced very bad results) and it was much more stable and more robust to the SVM training parameters.

This is probably due to the fact that the most important factor, the underlying non-linearities, are extracted out using the semi-parametric method and modeled using the SVM. This approach is more streamlined from the point of view of capturing the true non-linear nature of the load data by focusing only on the non-linear parts without getting diluted by taking into consideration the linear parts as is usually done.

Acknowledgement

The authors would like to thank the National Research Foundation of South Africa for their financial support.

References

1. Papadakis, S.E., Theocharis, J.B., Kiartzis, S.J., Bakirtzis, A.G.: A novel approach to short-term load forecasting using fuzzy neural networks. IEEE Transactions on Power Systems **13** (1998) 480–492
2. Piras, A., Germond, A., Buchenel, B., Imhof, K., Jaccard, Y.: Heterogeneous artificial neural network for short term electrical load forecasting. IEEE Transactions on Power Systems **11** (1996) 397–402
3. Bitzer, B., Rösser, F.: Intelligent Load Forecasting for Electrical Power System on Crete. In: UPEC 97 Universities Power Engineering Conference, UMIST-University of Manchester (1997)
4. Pelckmans, K., Suykens, J., Van Gestel, T., De Brabanter, J., Lukas, L., Hamers, B., De Moor, B., Vandewalle, J.: LS-SVMlab Toolbox User's Guide, Version 1.5. Catholic University Leuven, Belgium, [Online], Available from: <http://www.esat.kuleuven.ac.be/sista/lssvmlab/> (2003)
5. Ukil, A., Jordaan, J.: A new approach to load forecasting: Using semi-parametric method and neural networks. In King, I., Wang, J., Chan, L., Wang, D., eds.: Neural Information Processing. Volume 4233 of LNCS., Springer (2006) 974–983
6. Suykens, J.: Support Vector Machines and Kernel Based Learning. Tutorial: IJCNN, Montreal, [Online], Available from: <http://www.esat.kuleuven.ac.be/sista/lssvmlab/ijcnn2005_4.pdf> (2003)
7. Pai, P.F., Hong, W.C.: Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms. Elsevier Electric Power Systems Research **74** (2005) 417–425
8. Yang, J., Stenzel, J.: Short-term load forecasting with increment regression tree. Elsevier Electric Power Systems Research **76** (2006) 880–888
9. Zivanovic, R.: Analysis of Recorded Transients on 765kV Lines with Shunt Reactors. In: Power Tech2005 Conference, St. Petersburg, Russia (2005)
10. Zivanovic, R., Schegner, P., Seifert, O., Pilz, G.: Identification of the Resonant-Grounded System Parameters by Evaluating Fault Measurement Records. IEEE Transactions on Power Delivery **19** (2004) 1085–1090
11. Jordaan, J.A., Zivanovic, R.: Time-varying Phasor Estimation in Power Systems by Using a Non-quadratic Criterium. Transactions of the South African Institute of Electrical Engineers (SAIEE) **95** (2004) 35–41 ERRATA: Vol. 94, No. 3, p.171-172, September 2004.
12. Gorry, P.A.: General Least-Squares Smoothing and Differentiation by the Convolution (Savitzky-Golay) Method. **62** (1990) 570–573
13. Bialkowski, S.E.: Generalized Digital Smoothing Filters Made Easy by Matrix Calculations. **61** (1989) 1308–1310
14. Draper, N.R., Smith, H.: Applied Regression Analysis. Second edn. John Wiley & Sons (1981)
15. Tibshirani, R.: Regression Shrinkage and Selection via the Lasso. Journal of the Royal Society. Series B (Methodological) **58** (1996) 267–288
16. Casar-Corredera, J.R., Alcásar-Fernándes, J.M., Hernándes-Gómez, L.A.: On 2-D Prony Methods. IEEE **CH2118-8/85/0000-0796 $1.00** (1985) 796–799
17. Zivanovic, R., Schegner, P.: Pre-filtering Improves Prony Analysis of Disturbance Records. In: Eighth International Conference on Developments in Power System Protection, Amsterdam, The Netherlands (2004)
18. Mathworks: MATLAB Documentation - Neural Network Toolbox. Version 6.5.0.180913a Release 13 edn. Mathworks Inc., Natick, MA (2002)