Hybrid Kalman Algorithms for Very Short-term Load Forecasting and Confidence Interval Estimation

Che Guan, *Student Member, IEEE,* Peter B. Luh, *Fellow, IEEE*, Laurent D. Michel, Matthew A. Coolbeth and Peter B. Friedland

***Abstract*--Very short-term load forecasting predicts the load over one hour into the future in five-minute steps and performs the moving forecast every five minutes. To quantify forecasting accuracy, the confidence interval is estimated in real-time. An effective prediction with a small associated confidence interval is important for area generation control and resource dispatch, and can help the operator further make good decisions. We previously presented a multi-level wavelet neural network method, but it cannot produce a good confidence interval due to the model itself. This paper presents a method of multiple wavelet neural networks trained by hybrid Kalman algorithms. The prediction, however, is difficult, since one effective model is not able to capture complex load features at different frequencies. Appropriate transformations on load components also result in a complicated derivation in order to estimate an accurate variance. The key idea is to use neural network trained by extended Kalman filter for the low frequency component which has a near linear input-output function relationship; and use neural networks trained by unscented Kalman filter for high frequency components which have nonlinear input-output function relationships. Forecasts for load components from individual networks are then transformed back and derived, and combined to form the final load prediction with the good confidence interval. Numerical testing demonstrates significant value for load component predictions via hybrid Kalman filter-based algorithms for training neural networks and the derivation for confidence interval, and shows that our method provides the accurate prediction.**

Index Terms-- *Confidence interval estimation, extended Kalman filter, multilevel wavelet neural networks, unscented Kalman filter, very short-term load forecasting.*

I. Introduction

VERY short-term load forecasting (VSTLF) predicts the load over one hour into the future in five minute steps and performs a moving forecast every five minutes. In order to indicate the forecasting reliability, a confidence interval (CI) is estimated in real-time. Effective load forecast is essential for area generation control and resource dispatch. A load forecasting accuracy within 5% is probably adequate (Ranaweea, Karady and Farmer, 1997). With a help of the smaller CI for VSTLF, the operator will make a low-risk decision. The framework multi-level wavelet neural networks (MWNN) has been presented for VSTLF and is briefly reviewed in Section II.B, but it cannot produce a good CI due to the incapable of filtering out uncertainties with respect to model and data. Therefore, MWNN has to be further developed for VSTLF with an accurate CI estimation. However, it is difficult to accurately capture components in terms of complex load features and adopt a good model with respect to model's capacity. It is also hard to derivate a right estimation of standard deviation when appropriate transformations are applied on load component input.

Methods for VSTLF and CI estimation are limited. Existing approaches for VSTLF can be roughly classified into two categories: the nonparametric including extrapolation and regression; and the parametric consisting of fuzzy inference, Kalman filters and neural networks. The former often use most recent data to approximate the load in the near future, whereas the latter use recent and statistical data to update system parameters and then perform the prediction. Existing approaches for CI estimation mainly include five techniques: error output, maximum likelihood, estimation equations, resampling and Bayesian inference. These will be briefly reviewed in Section II.A. Although these methods are practical and applicable, a few papers presented a combined method for both VSTLF and CI estimation. Furthermore, few papers developed effective forecasters for load frequency components with different features, and derivate an accurate form of standard deviation around VSTLF when necessary transformations were applied on load inputs.

In this paper, a method of multilevel wavelet neural networks trained by hybrid Kalman algorithms (MWNNHK) is developed to forecast next hour's load in five-minute steps and generate a moving prediction every five minutes, around which a good CI is estimated at the same time. To accurately capture the feature of individual load component, the idea is to use the neural network trained by extended Kalman filter (EKFNN) to predict low frequency component and neural networks trained by unscented Kalman filter (UKFNN) for high frequency components because data analysis shows that the low frequency forecast as a function of its input is near linear whereas high frequencies' are nonlinear. Furthermore, the former propagates mean and covariance through linearization of the near linear model whereas the latter propagates and recovers them through the non-linear function. Forecasts from individual networks are then transformed back and combined to form the overall forecast. The details are described in Section III.

This work was supported by a grant from ISO New England.

Che Guan and Peter B. Luh are with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269-2157.

Laurent D. Michel and Matthew A. Coolbeth are with the Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269-2155.

Peter B. Friedland is with ISO New England, Holyoke, MA 01040.

To estimate a good CI, sub-variances from networks are further derived with respect to different transformations applied on load components. The overall variance is a sum of sub-variances from networks since the output sub-variances are orthogonal, which originates from the orthogonality of input components determined by wavelet decomposition property. The final CI is obtained by taking the standard deviation of the derived variance. The details are given in Section IV. Numerical testing results for a simple example and prediction of ISO-NE in Section V demonstrate values of MWNNHK for VSTLF and CI estimation.

II. Literature Review

A. VSTLF and Confidence Interval Estimation Methods

Methods for VSTLF have been briefly described in Section I. Among forecasting models, neural networks have been widely used, and most of them adopt multilayer preceptron with back-propagation learning algorithm. Their inputs usually include time index, load of previous hour, load of the yesterday and previous week with same hour and weekday index to the target hour. The relative increment, logarithmic difference or normalization in load may be applied on the load input in order to improve data stationary and to emphasize the look-ahead feature of training data (Charytoniuk and Chen, 2000; Shamsollahi and at al., 2001; Chen and York, 2008). The output may have different definitions in forecasting horizons and resolutions, e.g., 5 minute ahead (Shamsollahi and at al., 2001), 1 to 30 minute ahead in 1 minute step (Chryssolouris, Lee and Ramsey, 1996), 5 to 30 minute ahead in 5 minute step (Daneshi and Daneshi, 2008), 20 minute ahead by set 1, 30 minute ahead by set 2 and etc. (Charytoniuk and Chen, 2000), 5-60 minute ahead in 5 minute steps and with moving forecast every 5 minutes (Guan and at al., 2009). Furthermore, several neural networks may be used to predict different periods of day with unique pattern in load dynamics, e.g., summer and autumn. However, real-time load contain frequency components with complicated features, it's difficult to capture all properties within one general model.

Few papers provide CI methods for VSTLF, thus references related to the general prediction with CI estimation is reviewed and summarized. It is found that most approaches are developed based on neural networks. The most direct way may be the error output technique, which used augmented network (with error as an additional output) to perform the load and error forecast and the latter will be used for CI estimation. This method cannot cope with multiple steps ahead forecasting (Alves da Silva and Moulin, 2000).

The maximum likelihood was presented to produce an input-dependent estimate of variance for data noise uncertainty. However, the data noise variance was likely to be underestimated as estimate fitted some data noise, particularly in regions of low data density where over-fitting was more probable (Papadopoulos, Edwards and Murrray, 2001).

Estimating equations assumed that errors have some distributions, based on which CI was derived. The simplest way is to assume that errors were stationary and normal, thus confidence interval was centered around the forecast. This approach did not coniser the uncertainty (model or data uncertainty) and required a large sample size (Charytoniuk and Niebrzydowski, 1998). A derived student t-distribution was presented for CI. However, normality assumption for error or weights may not be satisfied when network function was nonlinear (Chryssolouris, Lee and Ramsey, 1996 and Alves da Silva and Moulin, 2000). A probabilistic load model was also presented (Charytoniuk and Niebrzydowski, 1998). It highly assumed load and weather had some distributions.

The resampling method derived CI by randomly selecting data points with replacement from an original output data set to form multiple sample data sets. The CI of the mean can then be calculated from means of the sample data sets (Zhang and Luh, 2005). Generally, bootstrap is one of popular resampling methods and is widely adopted. In addition, a resampling based a binomial destruction was derived (Alves da Silva and Moulin 2000). However, the method resampled output thus cannot effectively consider input uncertainties. The inferring on CIs treated an error as a random variable which was often a correlated and non-stationary process.

The Bayesian approach for networks started with a prior distribution of network weights and then maximized a posterior distribution from historical data in order to produce optimized weights (Zhang and Luh, 2005). To avoid computing high dimensional functions for posterior, Markov Chain Monte Carlo method drew from some complex distribution (Walsh, 2004; Papadopoulos, Edwards and Murray, 2001), and it gave better results than alternative non-Bayesian methods in all case problems, and best models were those with less restrictive priors (Lampinen and Vehtari, 2001). The method was impractical for a practical network with high dimensional weights. Therefore, the prediction distribution conditioned on a new input and weights can be derived and approximated as Gaussian via linearizing network (Wright, 1999; Zhang, Luh and Kasiviswanathan, 2003).

It can be concluded that multilayer preceptron network with back-propagation learning is a potential model for the prediction. However, the general network cannot estimate a good CI due to its incapable of filtering out data and model uncertainties. Also, back-propagation learning operates on basis of first-order gradients with respect to weights (Haykin and Kailath, 2002). To produce a good CI and accelerate rate of convergence, the feed-forward networks were trained with the EKF by treating weight as state (Singhal, and Wu, 1989). To speed up the computation, EKF was extended to the decoupled EKF by ignoring the interdependence of mutually exclusive groups of weights (Puskorius and Feldkamp, 1991). The numerical stability and accuracy of decoupled EKF was further improved by U-D factorization (Zhang and Luh, 2005). If the network function was highly nonlinear, the EKFNN gives poor performance because mean and covariance are propagated via linearization of the underlying non-linear model. UKFNN showed a superior performance for nonlinear functions (Ilyas and at al., 2008).

B. *Multilevel Wavelet Neural Networks*

A multilevel wavelet neural network method was presented previously to address VSTLF (Che and at al., 2009) as depicted in Figure 1. The real-time load had different frequency components and each with a unique feature. To capture the feature for each component, the Daubechies 2 wavelet was chosen to decompose the load of last hour into three components: low-low (LL), low-high (LH) and high (H) frequency components. The numbers of wavelet and decomposed levels were chosen based on testing results and would make sure the model provides the best prediction.

Data analysis showed the relative increment helped capture the change of LL component and thus was applied on it. Transformed components were next normalized, which together with other proper time indices (hour and weekday) were fed into low and high frequency networks so that each component can be accurately captured. Forecasts from individual networks were then transformed back and combined to form hourly forecast in five minute steps. In order to take full use of up-to-date load and provide a good forecasting resolution, twelve MWNN structures were further applied to form a moving forecast, and each handled a unique one-hour load input. These load inputs to different structures were sequentially 5-minute value shifted in an hour window.

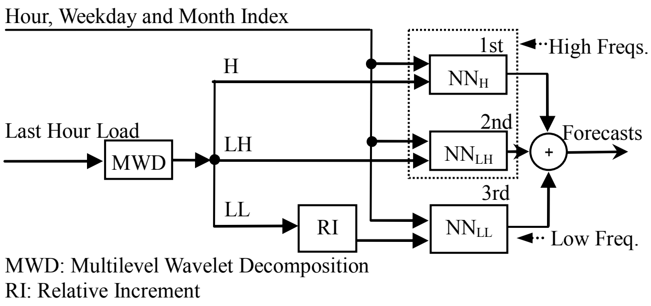

Fig. 1. Structure of Multilevel Wavelet Neural Networks (MWNN)

III. Multilevel Wavelet Neural Networks Trained by Hybrid Kalman Algorithms

The general neural network adopts multilayer preceptron with back-propagation learning algorithm to perform the prediction. This network prediction exhibits uncertainty due to inaccuracies of the training data (data noise) and to the limitations of the model (model uncertainty) (Papadopoulos, Edwards and Murray, 2001). The network in MWNN thus cannot estimate a good CI for load forecasting. The Kalman filter is a well-known method for recursive state estimation of linear dynamic system, and is a minimum mean-square-error estimator. To eliminate the uncertainty, it replaces back propagation learning and is used to train neural network by treating network weight as state and output as measurement in the system with dynamic. Through linearization, EKF has been widely adopted for state estimation of nonlinear system. These EKFNN methods show the fast convergence and provide accurate predictions (Zhang and Luh, 2005; Puskorius and Feldkamp, 1991 and Singhal, and Wu, 1989). If the network function is highly nonlinear, UKFNN shows a superior performance (Yu and at al., 2007). Data analysis for VSTLF shows that the load has multiple components and each with a unique feature. Furthermore, Q-Q plot depicted in Figure 2 shows that the network input-output function for low frequency component is linear related, whereas the one for high frequency component's is highly non-linear related. In order to accurately predict each frequency component, the structure of multilevel wavelet neural networks trained by hybrid Kalman algorithms is presented as depicted in Figure 3. In the framework, EKFNN is adopted to forecast the low frequency component, as will briefly described in A; UKF is developed to forecast high frequency components, as will briefly described in B. Forecasts from separate networks are added up to form the final load forecast. Meanwhile, estimated sub-variances from networks will be further derived in order to produce an accurate CI, as described in Section IV.

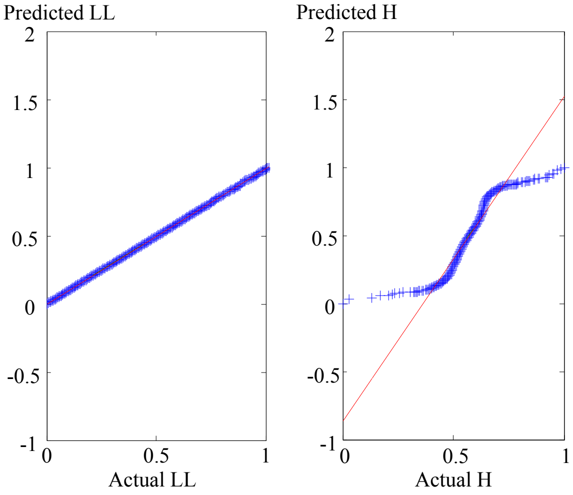

Fig. 2. Q-Q plot for actual input and predicted output of load freq.

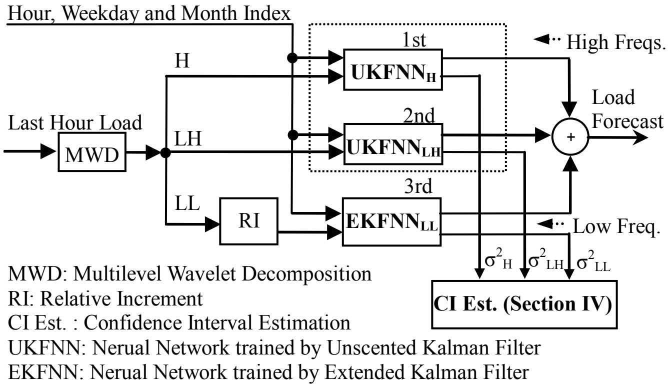
Hour, Weekday and Month Index
H
1st
High Freqs.
$UKFNN_H$
Last Hour Load
LH
2nd
Load Forecast
MWD
$UKFNN_{LH}$
+
3rd
LL
RI
$EKFNN_{LL}$
Low Freq.
σ^2_H
σ^2_{LH}
σ^2_{LL}
CI Est. (Section IV)
MWD: Multilevel Wavelet Decomposition
RI: Relative Increment
CI Est. : Confidence Interval Estimation
UKFNN: Nerual Network trained by Unscented Kalman Filter
EKFNN: Nerual Network trained by Extended Kalman Filter

Fig. 3. Overall Structure of Multilevel Wavelet Neural Networks Trained by Hybrid Kalman Algorithms (MWNNHK)

A. *EKFNN for Low Frequency Component*

Data analysis shows that low frequency network function between input and output is near linear. Through linearization, EKF replaces the back propagation learning of the low frequency neural network in MWNN and trains the network by treating its weight as state and desired output as measurement in the system with dynamics. In the process of training, weights are adjusted by using a set of input-output observations {u(t), z(t) t=1,…,T}, where T is the number of samples, u(t) is a n_ux1 input vector representing the normalized relative increment in low frequency component, and $\hat{z}(t)$ is corresponding n_zx1 estimated output for observation z(t).

To prepare a proper input to EKFNN, the relative increment (Charytoniuk and Chen, 2000) is applied on low frequency component l_t as denoted by (1), which is then normalized by (2) in order to produce the element u_t for the input vector u(t) = $\{u_t, u_{t+1}, \ldots\}^T$

$$l_t^{RI} = \left(l_t - l_{t-1}\right)/l_{t-1} \tag{1}$$

$$u_t = \left(l_t^{RI} - l_{\min}^{RI}\right)/\left(l_{\max}^{RI} - l_{\min}^{RI}\right). \tag{2}$$

To produce a right form of forecasting component with respect to the transformation on network input, the de-normalization is taken on the element $\hat{z}_t$ of the network output vector $\hat{z}(t) = \left\{\hat{z}_t, \hat{z}_{t+1}, \cdots\right\}^T$ by (3), which is then inverse-transformed with respect to relative increment by (4) in order to produce the element $\hat{l}_t$ for the vector of low frequency load prediction $\hat{L}(t) = \left\{\hat{l}_t, \hat{l}_{t+1,\cdots}\right\}^T$ with dimension n_zx1

$$\hat{z}_t' = \hat{z}_t \cdot \left(l_{\max}^{RI} - l_{\min}^{RI}\right) + l_{\min}^{RI} \tag{3}$$

$$\hat{l}_t = \left(\hat{z}_t' + 1\right) \cdot l_{t-1},\quad \hat{l}_{t+1} = \left(\hat{z}_{t+1}' + 1\right) \cdot \hat{l}_t \cdots . \tag{4}$$

The formulation of training network via EKF (Zhang and Luh, 2005) is described via state and measurement functions by (5) and (6)

$$w(t+1) = w(t) + \varepsilon\left(\mathrm{t}\right) \tag{5}$$

$$z(t) = \mathrm{h}\left(u(t-1), w(t)\right) + v\left(\mathrm{t}\right). \tag{6}$$

where h(•) is the input-output function of the network, ε(t) and ν(t) are the process and observation noises which are both assumed to be zero mean Gaussian noised with covariance Q(t) and R(t), and w(t) is an n_wx1 state vector and n_w is determined by numbers of inputs, hidden neurons and outputs (n_w, n_h and n_z) produced by (7)

$$n_w = \left(\mathrm{n_x} + 1\right) \times \mathrm{n}_h + \left(\mathrm{n_h} + 1\right) \times n_z . \tag{7}$$

With input u(t), weight w(t) and output $\hat{z}(t)$ vectors in framework of EKF, key steps of derivation (Bar-Shalom, Li and Kirubarajan, 2001) are summarized below

$$\widehat{w}(t+1 \mid t) = \widehat{w}(t \mid t) \tag{8}$$

$$P\left(t+1 \mid t\right) = P(t \mid t) + Q(t) \tag{9}$$

$$\hat{z}(t+1 \mid t) = \mathrm{h}\left(u(t), \widehat{w}(t+1 \mid t)\right) \tag{10}$$

$$S\left(t+1\right) = H(t+1) \cdot P(t+1 \mid t) \cdot H(t+1)^T + R(t+1) \tag{11}$$

$$where \quad H(t+1) = \left(\partial h(u,w)/\partial w\right)\Big|_{\substack{u=u(t) \\ w=\widehat{w}(t+1|t)}} \tag{12}$$

$$K(t+1) = P\left(t+1 \mid t\right) H\left(t+1\right)^T S\left(t+1\right)^{-1} \tag{13}$$

$$\widehat{w}(t+1 \mid t+1) = \widehat{w}(t+1 \mid t) + K\left(t+1\right) \cdot \left(z(t+1) - \hat{z}\left(t+1 \mid t\right)\right)$$

$$P(t+1 \mid t+1) = P(t+1 \mid t) - K(t+1) S(t+1) K(t+1)^T . \tag{14}$$

where H(t+1) is the partial derivative of h(•) with respect to w(t) at the estimated weights with dimension n_z x n_w, K(t+1) is the Kalman gain, P(t+1|t) is the prior weight covariance matrix and is updated to posterior weight covariance matrix P(t+1|t+1) based on the Bayesian formula, and S(t+1) is the measurement covariance matrix and its diagonal $\sigma^2_{LL}(t) = \{\sigma^2_t, \sigma^2_{t+1}, \ldots\}$will be used to derive CI later.

B. UKFNN for High Frequency Components

High frequency neural networks (NN_{LH} and $NN_{H)}$ are separately used to capture high frequency features in MWNN. Based on Figure 2, the input-output function for high frequency is nonlinear. When the Jacobian function H for high frequency is highly nonlinear, the EKF can give poor performance because the mean and covariance are propagated through linearization of the underlying non-linear model. UKF uses a sampling technique known as the unscented transform to pick a minimal set of sample points (called sigma points) around the mean. These sigma points are then propagated through the non-linear function, from which mean and covariance of estimate are recovered. The result is a filter which more accurately captures true mean and covariance. It also avoids the need to calculate Jacobian function H. Therefore, networks for high frequencies are replaced by $UKFNN_H$ and $UKFNN_{LH}$ as described in the following.

Very similar to EKF, UKF replaces the back propagation learning and trains the high frequency network by treating the weight as state and the output as measurement. The input u(t) is a n_ux1 vector representing the normalized high frequency and its element is produced by (15), and $\hat{z}(t)$ is the corresponding n_zx1 estimated output and its element is de-normalized by (16)

$$u_t = \left(l_t - l_{\min}\right)/\left(l_{\max} - l_{\min}\right) \tag{15}$$

$$\hat{l}_t = \hat{z}_t \cdot \left(l_{\max} - l_{\min}\right) + l_{\min} . \tag{16}$$

UKF adopts similar predict and update equations (Julier, Uhlmann and Durrant-Whyte, 1995) as EKF does. For predict equations, estimated state and covariance are written in augmented forms by (17) and (18). A set of 2N+1 sigma points χ is derived by (19) and (20). These points are then weighted to obtain prior state and covariance by (21) and (22)

$$\widehat{w}^a\left(t \mid t\right) = [\widehat{w}^T\left(t \mid t\right) \quad E(\varepsilon^T)]^T \tag{17}$$

$$P^a\left(t \mid t\right) = \begin{bmatrix} P\left(t \mid t\right) & 0 \\ 0 & Q \end{bmatrix} \tag{18}$$

$$\chi^i(t \mid t) = \widehat{w}^a(t \mid t) + \left(\sqrt{\left(L+\lambda\right) \cdot P^a(t \mid t)}\right)_i, i = 1 \ldots N$$

$$\chi^i(t \mid t) = \widehat{w}^a(t \mid t) - (\sqrt{\left(L+\lambda\right) \cdot P^a(t \mid t)})_{i-N}\, i = N+1 \ldots 2N$$

$$\chi^0(t \mid t) = \widehat{w}^a(t \mid t) \tag{19}$$

$$\chi^i(t+1\,|\,t)=\chi^i(t\,|\,t) \tag{20}$$

$$\widehat{w}(t+1\,|\,t)=\sum_{i=0}^{2N} W_s^i\cdot\chi^i(t+1\,|\,t)$$

$$where\, W_s^0=\lambda/(N+\lambda), W_s^i=.5/(N-\lambda), \lambda=\alpha^2(N)-N \tag{21}$$

$$P(t+1\,|\,t)$$
$$=\sum_{i=0}^{2N} W_c^i\cdot\left[\chi^i(t+1\,|\,t)-\widehat{w}(t+1\,|\,t)\right]\cdot\left[\chi^i(t+1\,|\,t)-\widehat{w}(t+1\,|\,t)\right]^T$$

$$where \quad W_c^0=\lambda/(N+\lambda)+\left(1-\alpha^2+\beta\right), W_c^i=.5/(N-\lambda)$$
$$\alpha=.001, \beta=2\,. \tag{22}$$

For update equations, the predicted state $\widehat{w}(t+1\,|\,t)$ and covariance P(t+1|t) are augmented with mean and covariance of measurement noise. The augmented mean $\widehat{w}^a(t+1\,|\,t)$ and covariance $P^a(t+1|t)$ denoted by (23) and (24) will derive another set of 2N+1 sigma points χ in (25), which are then projected via network function h to calculate γ points by (26)

$$\widehat{w}^a(t+1\,|\,t)=[\widehat{w}^T(t+1\,|\,t) \quad E(v^T)]^T \tag{23}$$

$$P^a(t+1\,|\,t)=\begin{bmatrix} P(t+1\,|\,t) & 0 \\ 0 & R \end{bmatrix} \tag{24}$$

$$\chi^i(t+1\,|\,t)=\widehat{w}^a(t+1\,|\,t)+\left(\sqrt{(N+\lambda)\cdot P^a(t+1\,|\,t)}\right)_i, i=1...N$$

$$\chi^i(t+1\,|\,t)$$
$$=\widehat{w}^a(t+1\,|\,t)-\left(\sqrt{(N+\lambda)\cdot P^a(t+1\,|\,t)}\right)_{i-N} \quad i=N+1...2N$$

$$\chi^0(t+1\,|\,t)=\widehat{w}^a(t+1\,|\,t) \tag{25}$$

$$\gamma^i(t+1)=h(\chi^i(t+1\,|\,t)), \quad i=0...2N\,. \tag{26}$$

These γ points are then weighted to produce the posterior state $\widehat{w}(t+1\,|\,t+1)$ and covariance P(t+1|t+1), gain K(t+1), prediction $\hat{z}(t+1\,|\,t)$ and innovation covariance S(t+1)

$$\hat{z}(t+1\,|\,t)=\sum_{i=0}^{2N} W_s^i\cdot\gamma^i(t+1) \tag{27}$$

$$S(t+1)$$
$$=\sum_{i=0}^{2N} W_c^i\left[\gamma^i(t+1)-\hat{z}(t+1\,|\,t)\right]\cdot\left[\gamma^i(t+1)-\hat{z}(t+1\,|\,t)\right]^T \tag{28}$$

$$P_{w(t+1)z(t+1)}$$
$$=\sum_{i=0}^{2N} W_c^i\left[\chi^i(t+1\,|\,t)-\widehat{w}(t+1\,|\,t)\right]\cdot\left[\gamma^i(t+1)-\hat{z}(t+1\,|\,t)\right]^T \tag{29}$$

$$K(t+1)=P_{w(t+1)z(t+1)}\cdot S(t+1)^{-1} \tag{30}$$

$$\widehat{w}(t+1\,|\,t+1)=\widehat{w}(t+1\,|\,t)+K(t+1)\cdot\left(z(t+1)-\hat{z}(t+1\,|\,t)\right) \tag{31}$$

$$P(t+1\,|\,t+1)=P(t+1\,|\,t)-K(t+1)S(t+1)K(t+1)^T\,. \tag{32}$$

To make propagation reasonable, the sum W_s and W_c have to be unit one respectively so that the mean and covariance of propagated sigma points are equal to mean and covariance of the state as described by (33)

$$\sum_{i=0}^{2N} W_s^i=\sum_{i=0}^{2N} W_c^i=1, \chi(t+1\,|\,t)=\widehat{w}(t+1\,|\,t), P_\chi=P(t+1\,|\,t)\,. \tag{33}$$

There are parameters are used to produce W_s and W_c. λ is a scaling parameter, α determines the spread of the sigma points around w̄ , and β is used to incorporate prior knowledge of the distribution of w (Ilyas and at al., 2008).

IV. Confidence Interval Estimation

The output sub-covariance diagonal from each network has been produced in the last section. Because input load components are orthogonal based on wavelet decomposition property, the estimated sub-variances from networks are orthogonal accordingly. Then sub-variances will be further derived with respect to the normalization and relative increment on the input components. Considering time requirement and computation complexity, the derived form may be approximated by ignoring some small terms. The overall variance is the sum of inverse transformed sub-variance from each network. The final CI is then obtained by deriving standard deviation of variance as shown in Figure 4.

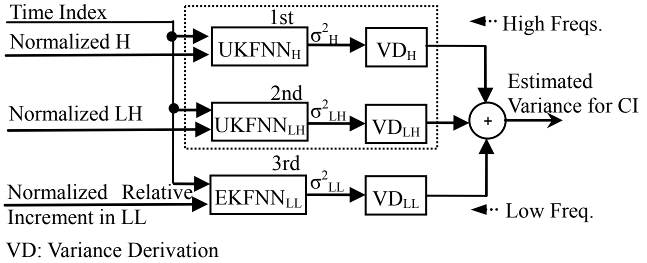

Fig. 4 Structure of Confidence Interval Estimation for MWNNHK

The overall variance equals the sum of three derived terms

$$\widehat{\sigma}^2_{Final}(t)=\widehat{\sigma}^2_{LL}(t)+\widehat{\sigma}^2_{LH}(t)+\widehat{\sigma}^2_{H}(t)\,. \tag{34}$$

The high frequency networks directly predict high frequency load components. To produce a right standard deviation form, the estimated covariance diagonal has to be scaled since the input was normalized before fed to network

$$\widehat{\sigma}^2_{H\,or\,LH}(t)=(l_{\max}-l_{\min})^2\cdot diag(S_{H\,or\,LH}(t))\,. \tag{35}$$

The low frequency network applies relative increment to improve data stationary and emphasize the look-ahead feature of training data and then feeds the normalized form to network. Because relative increment is a non-linear transformation whereas the normalization is nonlinear, a derivation has to be carried out in order to produce an accurate estimated variance. Let us extend the equation (4) involving with prediction for low frequency to the equation (36) so that the corresponding variance can be derived next

$$\hat{l}_t=\left[\hat{z}_t{}'+1\right]\cdot l_{t-1},$$
$$\hat{l}_{t+1}=\left[\hat{z}'_{t+1}+1\right]\cdot\hat{l}_t=\left[\hat{z}'_{t+1}+1\right]\cdot\left[\hat{z}'_t+1\right]\cdot l_{t-1},$$
$$\hat{l}_{t+2}=\left[\hat{z}'_{t+2}+1\right]\cdot\hat{l}_{t+1}=\left[\hat{z}'_{t+2}+1\right]\cdot\left[\hat{z}'_{t+1}+1\right]\cdot\left[\hat{z}'_t+1\right]\cdot l_{t-1},$$
$$\cdots. \tag{36}$$

Data analysis shows that $\hat{z}'^2(t)=\left\{\hat{z}'^2_t,\hat{z}'^2_{t+1},\cdots\right\}^T$ are uncorrelated, and its corresponding variance can be denoted as $\sigma'^2(t)=\{\sigma'^2_t, \sigma'^2_{t+1},\ldots\}$, which is $(L^{RI}_{max}-L^{RI}_{max})^2$ times of estimated variance of network output $\sigma^2(t)=\{\sigma^2_t, \sigma^2_{t+1},\ldots\}$. Thus variances for first two in (36) is derived to (37), and following components will be derived in a similar way by (38)

$$\overset{\frown}{\sigma}^2_t=Var\left[\hat{z}'_t+1\right]\cdot l^2_{t-1}=\sigma'^2_t\cdot l^2_{t-1}$$

$$\overset{\frown}{\sigma}^2_{t+1}=\left(Var\left[\hat{z}'_t\cdot\hat{z}'_{t+1}\right]+Var\left[\hat{z}'_t\right]+Var\left[\hat{z}'_{t+1}\right]\right)\cdot l^2_{t-1}$$

$$=\left\{E\left(\left[\hat{z}'_t\cdot\hat{z}'_{t+1}\right]^2\right)-\left[E\left(\hat{z}'_t\cdot\hat{z}'_{t+1}\right)\right]^2+\sigma'^2_t+\sigma'^2_{t+1}\right\}\cdot l^2_{t-1}$$

$$=\left\{\left(\hat{z}'^2_t+\sigma'^2_t\right)\cdot\left(\hat{z}'^2_{t+1}+\sigma'^2_{t+1}\right)-\hat{z}'^2_t\cdot\hat{z}'^2_{t+1}+\sigma'^2_t+\sigma'^2_{t+1}\right\}\cdot l^2_{t-1}$$

$$=\left\{\hat{z}'^2_t\cdot\sigma'^2_{t+1}+\hat{z}'^2_{t+1}\cdot\sigma'^2_t+\sigma^2_t\cdot\sigma'^2_{t+1}+\sigma'^2_t+\sigma'^2_{t+1}\right\}\cdot l^2_{t-1} \tag{37}$$

$$\begin{aligned}\overset{\frown}{\sigma}^2_{t+2}=[&\sigma'^2_t\cdot\sigma'^2_{t+1}\cdot\sigma'^2_{t+2}+\sigma'^2_t\cdot\sigma'^2_{t+1}+\sigma'^2_{t+1}\cdot\sigma'^2_{t+2}\\ &+\hat{z}'^2_{t+2}\cdot\sigma'^2_{t+1}+\hat{z}'^2_{t+1}\cdot\sigma'^2_{t+2}+\sigma'^2_t\cdot\sigma'^2_{t+2}+\hat{z}'^2_{t+2}\cdot\sigma'^2_t\\ &+\hat{z}'^2_{t+2}\cdot\sigma'^2_t\cdot\sigma'^2_{t+1}+\hat{z}'^2_{t+1}\cdot\sigma'^2_t\cdot\sigma'^2_{t+2}+\hat{z}'^2_{t+1}\cdot\hat{z}'^2_{t+2}\cdot\sigma'^2_t\\ &+\hat{z}'^2_t\cdot\sigma'^2_{t+2}+\hat{z}'^2_t\cdot\sigma'^2_{t+1}\cdot\sigma'^2_{t+2}+\hat{z}'^2_{t+1}\cdot\sigma'^2_t+\hat{z}'^2_t\cdot\hat{z}'^2_{t+2}\cdot\sigma'^2_{t+1}\\ &+\hat{z}'^2_t\cdot\hat{z}'^2_{t+1}\cdot\sigma'^2_{t+2}+\hat{z}'^2_t\cdot\sigma'^2_{t+1}+\sigma'^2_t+\sigma'^2_{t+1}+\sigma'^2_{t+2}]\cdot l^2_{t-1}\end{aligned}$$

$$\ldots. \tag{38}$$

Testing shows that elements of $\hat{z}'^2(t)$ are at 10^{-4} level, the elements of $\overset{\frown}{\sigma}'^2(t)$ are at 10^{-6} level. It means only elements of $\hat{z}'^2(t)\cdot\overset{\frown}{\sigma}'^2(t)^T$ and $\overset{\frown}{\sigma}'^2(t)$ will determine the final variance. Therefore, (37) and (38) is further approximated by (39)

$$\overset{\frown}{\sigma}^2_t=\sigma'^2_t\cdot l^2{}_{t-1}$$

$$\overset{\frown}{\sigma}^2_{t+1}\approx\left[\left(1+\hat{z}'^2_{t+1}\right)\cdot\sigma'^2_t+\left(1+\hat{z}'^2_t\right)\cdot\sigma'^2_{t+1}\right]\cdot l^2{}_{t-1}$$

$$\overset{\frown}{\sigma}^2_{t+2}\approx\begin{bmatrix}\left(1+\hat{z}'^2_{t+1}\right)\cdot\sigma'^2_t+\left(1+\hat{z}'^2_t\right)\cdot\sigma'^2_{t+1}\\ +\left(1+\hat{z}'^2_t+\hat{z}'^2_{t+1}\right)\cdot\sigma'^2_{t+2}\end{bmatrix}\cdot l^2{}_{t-1},\qquad\cdots. \tag{39}$$

Substitute network output vector $\hat{z}(t)$ and $\sigma^2(t)$, eq. (39) is finally deducted to (40), from where the derived variance for low frequency is described as $\overset{\frown}{\sigma}^2_{LL}(t)=\left\{\overset{\frown}{\sigma}^2_t,\overset{\frown}{\sigma}^2_{t+1},\cdots\right\}$

$$\overset{\frown}{\sigma}^2_{t+k}=\sum_{j=0}^{k}\left\{\left(1+\sum_{i=0}^{k}\hat{z}'^2_{t+i}-\hat{z}'^2_{t+j}\right)\cdot\sigma'^2_{t+j}\right\}\cdot l^2{}_{t-1},\ k=0,\cdots,n_z-1$$

$$=\sum_{j=0}^{k}\left\{\left(1+\sum_{i=0}^{k}\left(a\cdot\hat{z}_{\mathrm{t+i}}+b\right)^2-\left(a\cdot\hat{z}_{t+j}+b\right)^2\right)\cdot a^2\cdot\sigma^2_{t+j}\right\}\cdot l^2{}_{t-1}$$

$$where\quad a=\left(l^{RI}_{\max}-l^{RI}_{\min}\right),b=l^{RI}_{\min}. \tag{40}$$

V. Numerical Testing Results

The MWNNHK method has been implemented in MATLAB on a personal computer with Pentium Dual Core 2.20GHz CPU. Three examples are presented below. Example 1 uses a classroom-type problem to examine the MWNNHK algorithm. Example 2 predicts 2007 New England load and demonstrates benefits of the EKFNN and UKFNN for capturing different frequency components. Example 3 predicts 2008 New England load, and shows the accurate VSTLF with small estimated CIs.

Example 1. Consider the following signal:

$$y(t)=100\sin(20\cdot t)+35\sin(400\cdot t)+3\sin(2000\cdot t)$$

which is composed of a low frequency component $100sin(20\cdot t)$ and two high frequency components $35sin(400\cdot t)$ *and* $3sin(2000\cdot t)$. The signal will be decomposed into three components and feed to EKFNN and UKFNNs. There are three process and measure noises, and each has a normal distribution with mean of $2\text{x}10^{-3}$ and variance 10^{-6} respectively. The objective is to predict $y(t)$ for $t\in$ [2001, 2002…2100]. This constructed signal has a strong resemblance to the actual in terms of amplitude and frequency.

Case 1. This example compares different ways of using EKFNN and UKFNN: using UKFNN and EKFNN to predict all the components; using EKFNN to predict high freq. and UKFNN for the low; using UKFNN to predict the high freq. and EKFNN for the low. MAE (mean absolute error) presented in Table I show first three ways degrade prediction accuracy as comparing to the last way (our presented method).

Case 2. The MAE, average interval and one sigma coverage for the estimation are list in Table II. The prediction together with the actual are plotted in Figure 5. The result shows predictions follow the actual quite well, which has small CIs close to historical average intervals, but high one σ coverage.

TABLE I
MAE FOR CLASSROOM PROBLEM

Min.	$\text{UKFNN}_{\text{H,LH,LL}}$	$\text{EKFNN}_{\text{H,LH,LL}}$	$\text{EKFNN}_{\text{H,LH}}$ UKFNN_{LL}	**$\textbf{EKFNN}_{\textbf{LL}}$** **$\textbf{UKFNN}_{\textbf{H,LH}}$**
1	3.6466	4.2533	5.9586	**1.6418**
2	9.8464	3.2470	5.1412	**1.7080**
3	7.1651	6.3217	5.1465	**2.9775**
4	14.5115	9.5686	8.4283	**4.7568**
5	8.5768	1.9385	3.6746	**1.9262**
6	5.7038	4.7614	4.0244	**2.6622**
7	5.1717	7.0685	5.2265	**3.7339**
8	11.1378	7.7590	5.9589	**5.6335**
9	2.4700	2.4416	5.2147	**1.1207**
10	10.6977	4.0791	5.2304	**4.5001**
11	16.5632	7.7090	7.3508	**3.0665**
12	11.1270	7.6954	6.0483	**5.1211**

TABLE II
MAE, AVERAGE INTERVAL AND 1-Σ COVERAGE FOR CLASSROOM PROBLEM

No.	MAE	Ave. Interval	Historical Ave. Interval	One sigma Coverage (%)
1	1.6418	4.2369	2.0345	95.6522
2	1.7080	6.1518	2.3536	100.000
3	2.9775	5.4585	3.9939	86.9565
4	4.7568	7.0894	5.9250	82.6087
5	1.9262	5.2922	2.4038	100.000
6	2.6622	4.6636	3.2963	82.6087
7	3.7339	6.4578	4.5681	82.6087

8	5.6335	9.2992	6.9613	73.9130
9	1.1207	4.1444	1.5328	100.000
10	4.5001	6.6771	5.4715	82.6087
11	3.0665	5.0214	3.6865	82.6087
12	5.1211	6.1539	6.9893	69.5652

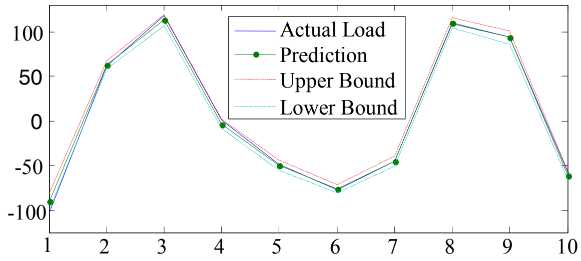

Fig. 5. Results for Example 1

Example 2. The MWNNHH method is used to predict ISO-NE load from Jan. to June, 2007. Training period is from Oct. to Dec., 2006. The example repeats the case 1 of example 1. MAEs presented in Table III and one sigma coverage in Table IV show that EKFNN for low frequency and UKFNN for high frequencies give the best estimation against other ways.

TABLE III
HOURLY MAE (MW) FOR NEW ENGLAND 2007 LOAD (JANUARY - JUNE)

Min.	UKFNN$_{H,LH,LL}$	EKFNN$_{H,LH,LL}$	EKFNN$_{H,LH}$ UKFNN$_{LL}$	**EKFNN$_{LL}$** **UKFNN$_{H,LH}$** **Presented**
5	42.9423	19.2667	43.0113	**18.1491**
10	68.2618	29.1739	68.1406	**27.4623**
15	81.9085	35.9657	82.6099	**34.3758**
20	96.2815	42.6757	96.8666	**41.3936**
25	105.3378	48.0644	105.4523	**47.6545**
30	121.0136	54.3451	120.7941	**54.1193**
35	133.1259	60.5533	133.0947	**60.2267**
40	140.9562	66.9414	141.0874	**66.9865**
45	152.1939	73.6568	152.1553	**73.7966**
50	158.1206	81.1829	158.0587	**81.1327**
55	165.3893	88.4589	165.4714	**87.2577**
60	172.3442	96.2712	172.6157	**94.5576**

TABLE IV
ONE SIGMA COVERAGE FOR NEW ENGLAND 2007 LOAD (JANUARY - JUNE)

Min.	UKFNN$_{H,LH,LL}$	EKFNN$_{H,LH,LL}$	EKFNN$_{H,LH}$ UKFNN$_{LL}$	**EKFNN$_{LL}$** **UKFNN$_{H,LH}$**
5	75.1603	89.8581	75.0687	**90.9799**
10	69.7115	83.9057	69.7344	**85.6914**
15	70.3068	82.3260	69.5742	**84.1575**
20	67.8342	79.7848	67.6969	**80.8837**
25	67.0330	78.9606	66.9643	**79.3956**
30	64.8352	77.3123	65.0870	**77.6099**
35	63.3013	75.9844	63.3242	**75.9158**
40	62.1337	74.5650	62.3397	**74.1300**
45	60.6456	72.2527	60.5311	**72.3901**
50	58.8599	70.2839	58.5852	**70.3068**
55	58.7912	68.9789	58.8141	**69.2995**
60	57.6236	66.9643	57.9899	**67.4222**

Example 3. The values of MWNNHK are examined to perform New England 2008 load from January to December. Training period is from October, 2006 to December, 2007. The prediction, average interval and one sigma coverage for load components in Table V to VII show that our method produces a very good load prediction with accurate CIs and high one–sigma coverage around load forecasting in real-time. A precise prediction for the actual with one sigma standard deviation around prediction is also depicted in figure 6.

TABLE V
MAE (MW) AND MAPE (%) FOR 2008 ISO-NE (JANUARY – DECEMBER)

	MAE (MW)				MAPE (%)
Min.	H	LH	LL	**MWNNHK**	**MWNNHK**
5	7.0587	1.7992	16.2757	**17.8461**	**0.1220**
10	4.0212	7.4561	21.4342	**26.3988**	**0.1801**
15	6.6841	5.6850	29.0596	**33.6141**	**0.2286**
20	3.9546	10.0628	38.4568	**40.2707**	**0.2732**
25	6.2506	1.6889	46.1253	**46.7610**	**0.3159**
30	3.6405	6.9822	54.5185	**53.2385**	**0.3588**
35	6.0117	5.1312	57.3525	**58.3653**	**0.3917**
40	3.5359	9.3823	63.7545	**64.6758**	**0.4333**
45	6.0793	1.9091	71.2818	**70.9390**	**0.4742**
50	3.6540	6.2274	79.0691	**77.6563**	**0.5182**
55	7.1432	5.4774	79.9938	**82.5112**	**0.5502**
60	3.4114	10.3463	83.3770	**88.4310**	**0.5895**

TABLE VI
AVERAGE INTERVAL (MW) FOR 2008 NEW ENGLAND LOAD (JAN. – DEC.)

MIN.	H	LH	LL	**MWNNHK**	Historical Ave. Interval
5	3.4565	5.2926	37.9921	**46.7412**	25.0819
10	3.4286	5.4570	51.0502	**59.9358**	37.7371
15	3.4520	5.3423	61.9938	**70.7880**	47.6900
20	3.4337	5.4054	70.6886	**79.5277**	57.7445
25	3.4263	5.2941	78.1970	**86.9174**	67.5386
30	3.4188	5.3825	85.0687	**93.8700**	77.5208
35	3.4315	5.3250	92.3165	**101.0730**	86.2294
40	3.4202	5.4616	98.5952	**107.4770**	96.6688
45	3.4328	5.3257	104.2336	**112.9921**	106.3797
50	3.4284	5.3680	109.6307	**118.4270**	117.1425
55	3.5687	5.5519	114.8600	**123.9805**	124.6536
60	3.4906	6.2501	119.7280	**129.4687**	134.2487

TABLE VII
ONE -SIGMA COVERAGE FOR ISO-NE 2008 LOAD (JAN. – DEC.)

Min.	H	LH	LL	**MWNNHK**
5	33.1731	96.3828	92.5595	**94.8031**
10	52.8846	49.3590	92.9258	**91.9185**
15	35.0733	59.1575	90.8196	**90.2930**
20	55.1969	37.4313	86.9277	**89.0568**
25	36.6758	96.7720	84.4780	**87.2024**
30	58.1731	50.5723	81.8681	**85.9203**
35	37.2482	64.6749	83.0586	**85.0504**
40	59.1346	41.3004	81.4789	**83.9515**
45	37.9350	95.2152	79.5559	**82.5092**
50	57.6007	55.5632	78.3883	**81.3187**
55	34.2949	64.3086	79.5330	**80.9524**
60	63.3471	42.3077	79.8077	**80.1969**

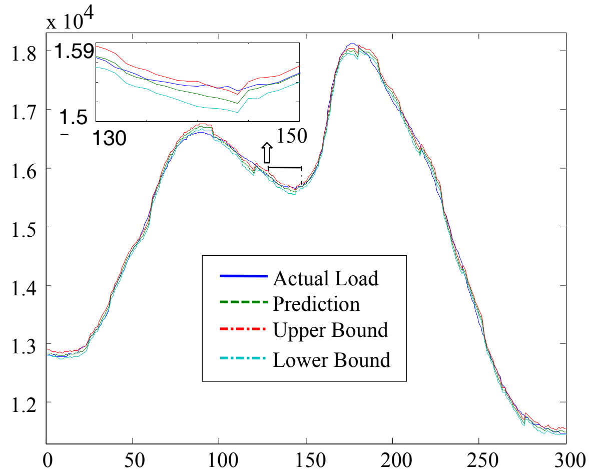

Fig. 6. The actual and prediction with associated CIs (Jan. 7st, 2008)

VI. Conclusion

A generic framework that combines wavelet decomposition neural networks trained by hybrid Kalman algorithms is presented for VSTLF and CI estimation. This paper investigated EKFNN for the low frequency component and UKFNN for high frequencies based on data analysis that the former has a near-linear input-output function, whereas the latter has a nonlinear one. Different transformations on load components are analyzed, and testing verifies the actuary of CI derivation. The resulting provides a very good prediction for ISO-NE's load and produces a small CI with the high one-sigma coverage. The method is currently being implemented and will be used in ISO-NE for their hourly used.

VII. References

[1] A. P. Alves da Silva and L. S. Moulin (2000). Confidence Interval for Neural Networks Based Short-term Load Forecasting. *IEEE Transaction on Power Systems*, 15(4), 1191-1196.

[2] B. Walsh (2004) Markov Chain Monte Carlo and Gibbs sampling. Lecture Notes for EEB 581.

[3] C. Charytoniuk, J. Niebrzydowski (1998). Confidence Interval Construction for Load Forecast. *Electric Power Systems Research*, 48, 97-103.

[4] C. Guan, P. B. Luh, M. A. Coolbeth, Y. Zhao, L. D. Michel, Y. Chen, C. J. Manville, P. B. Friedland and S. J. Rourke (2009). Very Short-term Load Forecasting: Multilevel Wavelet Neural Networks with Data Pre-filtering, *Proceedings of the IEEE PES 2009 General Meeting*.

[5] D. K. Ranaweea, G. G. Karady and R. G. Farmer (1997). Economic impact Analysis of Load Forecasting, *IEEE Transaction on Power Systems, 12(3),* 1388-1392.

[6] D. Chen and M. York (2008). Neural Network Based Very Short Term Load Prediction, *Transmission and Distribution Conference and Exposition, T&D. IEEE/PES,* 1-9

[7] G. Chryssolouris, M. Lee and A. Ramsey (1996). Confidence Interval Prediction for Nerual Network Models. IEEE Transactions on Neural Networks, 7(1), 229-232.

[8] G. Papadopoulos, P. J. Edwards and A. F. Murray (2001). Confidence Estimation Methods for Neural Networks: A Practical Comparison. *IEEE Transaction on Neural Networks, 12(6),* 1278-1287.

[9] H. Daneshi and A. Daneshi (2008). Real Time Load Forecast in Power System, *3rd International Conference on Deregulation and Restructuring and Power Technologies, DRPT*, 689-695.

[10] J. Lampinen and A. Vehtari (2001). Bayesian Approach for Neural Networks – Review and Case Studies. *Neural Networks, 14,* 257-274.

[11] K. Liu, S. Subbarayan, R. R. Shoults, M. T. Manry, C. Kwan, F. L. Lewis and J. Naccarino (1996). Comparison of Very Short-Term Load Forecasting Techniques, *IEEE Tran. on Power Systems, 11(2),* 877-882.

[12] L. Zhang and P. B. Luh (2005). Neural Networks – Based Market Clearing Price Prediction and Confidence Interval Estimation With an Improved Extended Kalman Filter Method. *IEEE Transaction on Power Systems, 20(1),* 59-66.

[13] L. Zhang, P. B. Luh and K. Kasiviswanathan (2003). Energy Clearing Price Prediction and Confidence Interval Estimation with Cascaded Neural Networks. *IEEE Transaction on Power Systems, 18(1),* 99-105.

[14] M. Ilyas, J. Lim, J. G. Lee and C. G. Park (2008). Federated Unscented Kalman Filter Design for Multiple Satellites Formation Flying in LEO, *International Conference on control, Automation & Systems,* 453-458.

[15] P. Shamsollahi, K. W. Cheung, Q. Chen and E. H. Germain (2001). A Neural Network Based Very Short Term Load Forecaster for the Interim ISO New England Electricity Market System, *IEEE Power Industry Computer Applications Conference,* 217-222.

[16] S. Haykin and T. Kailath (2002). *Adaptive Filter Theory*, Fourth Edition, *Pearson Education, Inc*, 759-760.

[17] S.J., Julier, J. K. Uhlmann, H.F. Durrant-Whyte (1995). A New Approach for Filtering Nonlinear Systems, *Proceedings of the American Control Conference,* 1628-1632

[18] W. A. Wright (1999). Bayesian Approach to Neural-Network Modeling with Input Uncertainty. *IEEE Transaction on Neural Networks, 10(6),* 1261-1270.

[19] W. Charytoniuk and M. S. Chen (2000). Very Short-Term Load Forecasting Using Artificial Neural Networks, *IEEE Transaction on Power Systems, 15(1),* 263-268.

[20] Y., Bar-Shalom, X. R. Li and T. Kirubarajan (2001). Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction, J. Wiley and Sons, 200-209, and 385-386.

[21] Y. Chen, P. B. Luh, C. Guan, Y. Zhao, L. D. Michel, M. A. Coolbeth, P. B. Friedland and S. J. Rourke (2010). Short-term Load Forecasting: Similar Day-based Wavelet Neural Networks, *IEEE Transaction on Power Systems.*

VIII. Biographies

Che Guan received his Bachelor degree from Electrical and Information Engineering at Changchun University of Science and Technology in 2004, and Master degree from Mechanical and Electrical Engineering at Chinese Academy of Sciences in 2007. He is currently a graduate student in the Electrical and Computer Engineering Department at the University of Connecticut.

Peter B. Luh received his Ph.D. degree in Applied Mathematics from Harvard University in 1980, and has been with the University of Connecticut since then. Currently he is the SNET Professor of Communications & Information Technologies. He is interested in planning, scheduling, and coordination of design, manufacturing, and supply chain activities; configuration and operation of elevators and HVAC systems for normal and emergency conditions; schedule, auction, portfolio optimization, and load/price forecasting for power systems. He is a Fellow of IEEE, Vice President of Publication Activities for the IEEE Robotics and Automation Society, the founding Editor-in-Chief of the new IEEE Transactions on Automation Science and Engineering (2003-2008), and was the Editor-in-Chief of IEEE Transactions on Robotics and Automation (1999-2003).

Laurent D. Michel received his Ph.D. in Computer Science from Brown University in 1999 and is currently holding an Associate Professor position in the Computer Science & Engineering Department at the University of Connecticut. He specializes is Combinatorial Optimization with a particular emphasis on Constraint Programming. He has co-authored 2 monographs, more than 60 papers and sits on the Editorial Board of Constraints, Mathematical Programming Computation and Constraint Letters.

Matthew A. Coolbeth graduated with a Bachelor degree in computer science from the University of Connecticut in 2007. He is currently a graduate student at the same university in the department of computer science and engineering.

Peter Friedland has been working in the electric utility industry for the past 22 years including work with an EMS Vendor, Transmission Operator and ISO/RTO. For the past 8 years, he has been with ISO New England in varying management capacities including Standard Market Design (SMD) Project Manager, IT-EMS Manager and Operations Manager. Peter earned a BSEE from University of Connecticut in 1986. Peter also attended University of Washington and UCONN as an Electrical Engineering Graduate Student. Peter is a member of the UCONN Electrical Engineering Industrial Advisory Board.