Blind Kalman Filtering for Short-term Load Forecasting

Shalini Sharma, Angshul Majumdar, Víctor Elvira, Emilie Chouzenoux

Blind Kalman Filtering for Short-term Load Forecasting

Shalini Sharma, *Student member, IEEE*, Angshul Majumdar, *Senior member, IEEE*, Víctor Elvira, *Senior member, IEEE*, Émilie Chouzenoux, *Senior member, IEEE*

***Abstract*—In this work we address the problem of short-term load forecasting. We propose a generalization of the linear state-space model where the evolution of the state and the observation matrices is unknown. The proposed blind Kalman filter algorithm proceeds via alternating the estimation of these unknown matrices and the inference of the state, within the framework of expectation-maximization. A mini-batch processing strategy is introduced to allow on-the-fly forecasting. The experimental results show that the proposed method outperforms the state-of-the-art techniques by a considerable margin, both on load profile estimation and peak load forecast problems.**

***Index Terms*—load forecasting, state-space model, Kalman filtering, expectation-minimization algorithm.**

I. Introduction

In this letter, we address the classical problem of short term (day ahead) load forecasting [1]. Classical signal processing techniques like stochastic time series analysis, Kalman filter, multiple linear regression, and exponential smoothing were used initially for this problem [2]. These linear techniques, with fixed and empirically set linear operators, presented low accuracy and flexibility, paving the way for non-linear neural network prediction paradigm since the 1990's [3]. After the saturation of the initial studies on neural networks (since the 2000's), non-linearity in short-term load forecasting was modelled in terms of support vectors [4]. The rise of deep learning led to the modern residual neural networks (ResNet) [5] and long short-term memory (LSTM) networks [6] in electricity load forecasting. One advantage of classical signal processing models (e.g., Kalman filter and its non-linear versions [7]) is their interpretability. Moreover, they can explicitly quantify the uncertainty in the estimate. The main issue with these models (both linear [2] and non-linear [8]) is that the state and the observations functions need to be known. Modelling short term (weekly) and long term (seasonal) fluctuations by a single function leads to oversimplification and consequently poor performance. This is the likely reason why neural network approaches, known for their function approximation capability, improve over classical signal processing techniques based on state-space models. However, the improved performance is at the expense of losing both interpretability and uncertainty quantification. In this work, we aim at retaining advantages from both approaches. Our methodology is based on the classical state-space model. However, we do not require specification of the state and observation matrices, which are instead learnt progressively from the data. While we assume a linear-Gaussian model, the state and observation matrices that we estimate can change over time. Hence, the method works by assuming temporally local (and unknown) linearity, which can be seen as an approximation of an underlying non-linearity, generalizing the standard linear-Gaussian model with static parameters. In this framework, we propose an inference method for the joint estimation of these unknown linear operators and the hidden state, all in a sequential manner. In this work we focus on the application of forecasting building level loads, which has generated much interest this last decade [9], [10]. We specifically address two problems: next day hourly (profile) load forecast and next day peak load forecast.

The paper is organized as follows. In Section II, we first focus on the problem of profile estimation, and introduce our blind Kalman filter method to address it. Then we show how to extend the latter to the problem of peak load forecast. Section III presents our numerical results, along with comparisons with state-of-the-art methods. Section IV concludes this paper.

S. Sharma and A. Majumdar are with Indraprastha Institute of Information Technology, Delhi, India (e-mail: shalinis, angshul@iiitd.ac.in). Emilie Chouzenoux is with CVN, Inria Saclay, CentraleSupélec, Univ. Paris Saclay, France (e-mail: emilie.chouzenoux@centralesupelec.fr). Víctor Elvira is with the School of Mathematics, University of Edinburgh, United Kingdom (e-mail: victor.elvira@ed.ac.uk).

II. Proposed Approach: Blind Kalman Filtering

A. Profile load forecast

The profile estimation problem can be formulated, using the following linear state-space model, for time $k = 1, \ldots, K$:

State Evolution:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{u}_k, \tag{1}$$

Observation:

$$\mathbf{y}_k = \mathbf{B}\mathbf{x}_k + \mathbf{v}_k, \tag{2}$$

where $\mathbf{x}_k$ and $\mathbf{y}_k$ are the hidden state and observed hourly quantities (e.g., power, temperature, humidity), respectively, for $k = 1, \ldots, K$, $\mathbf{u}_k$ and $\mathbf{v}_k$ are additive white Gaussian noises (AWGN) with zero mean and covariances $\mathbf{Q}$ and $\mathbf{R}$, respectively. $\mathbf{A}$ is the state transition matrix and $\mathbf{B}$ is the observation matrix. The first state $\mathbf{x}_0$ is a random variable with normal distribution $\mathcal{N}(\mathbf{x}_0; \bar{\mathbf{x}}_0, \mathbf{P}_0)$. When the matrices $\mathbf{A}$ and $\mathbf{B}$ are known, the filtering and smoothing solutions are given by the Kalman filter and Rauch-Tung-Striebel (RTS) smoother i.e., computing the distributions $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ and $p(\mathbf{x}_k|\mathbf{y}_{1:K})$ respectively [11]. However, in practical situations, $\mathbf{A}$ and $\mathbf{B}$ are unknown, and hence they must be estimated.

We propose here the so-called blind Kalman filter (BKF) algorithm. The solution, relying on the expectation-maximization framework [13], proceeds in two parts; it alternates between the (i) estimation of the states, assuming $\mathbf{A}$ and $\mathbf{B}$ to be fixed, and (ii) estimation of $\mathbf{A}$ and $\mathbf{B}$, assuming the states to be fixed.

1) Filtering-smoothing step: For the first step, we fix the parameters $\mathbf{A}$ and $\mathbf{B}$. We use the aforementioned Kalman filter / RTS smoother.[1] The Kalman recursions are as follows:
Initialize: $\bar{\mathbf{x}}_0$, $\mathbf{P}_0$
For $k = 1, \ldots, K$
Predict State:

$$\left\{ \begin{array}{ll} \mathbf{x}_k^- & = \mathbf{A}\bar{\mathbf{x}}_{k-1}, \\ \mathbf{P}_k^- & = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^\top + \mathbf{Q}. \end{array} \right. \tag{3}$$

Update state:

$$\left\{ \begin{array}{ll} \mathbf{z}_k & = \mathbf{y}_k - \mathbf{B}\mathbf{x}_k^-, \\ \mathbf{S}_k & = \mathbf{B}\mathbf{P}_k^-\mathbf{B}^\top + \mathbf{R}, \\ \mathbf{K}_k & = \mathbf{P}_k^-\mathbf{B}^\top\mathbf{S}_k^{-1}, \\ \overline{\mathbf{x}}_k & = \mathbf{x}_k^- + \mathbf{K}_k\mathbf{z}_k, \\ \mathbf{P}_k & = \mathbf{P}_k^- - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^\top. \end{array} \right. \tag{4}$$

The Kalman filter provides the normal filtering distribution as

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \mathcal{N}(\mathbf{x}_k; \bar{\mathbf{x}}_k, \mathbf{P}_k). \tag{5}$$

One can also compute the distribution of the hidden state conditioned to all data (i.e., including future data when possible). This is the smoothing distribution given by

$$p(\mathbf{x}_k|\mathbf{y}_{1:K}) = \mathcal{N}(\mathbf{x}_k; \mathbf{x}_k^s, \mathbf{P}_k^s), \tag{6}$$

where mean and variance are computed by the RTS smoother that re-uses some of the results of the Kalman filter:
For $k = K, \ldots, 1$
Backward Recursion (Bayesian Smoothing):

$$\left\{ \begin{array}{ll} \mathbf{x}_{k+1}^- & = \mathbf{A}\overline{\mathbf{x}}_k, \\ \mathbf{P}_{k+1}^- & = \mathbf{A}\mathbf{P}_k\mathbf{A}^\top + \mathbf{Q}, \\ \mathbf{G}_k & = \mathbf{P}_k\mathbf{A}^\top[\mathbf{P}_{k+1}^-]^{-1}, \\ \mathbf{x}_k^s & = \overline{\mathbf{x}}_k + \mathbf{G}_k[\mathbf{x}_{k+1}^s - \mathbf{x}_{k+1}^-], \\ \mathbf{P}_k^s & = \mathbf{P}_k + \mathbf{G}_k[\mathbf{P}_{k+1}^s - \mathbf{P}_{k+1}^-]\mathbf{G}_k^\top. \end{array} \right. \tag{7}$$

Using the RTS smoother requires processing the full sequence of data, which prevents from an online prediction and a responsive estimation of the hidden state. In practice, we will make use of a windowing strategy, applying the RTS to data minibatches.

2) State matrices update step: The next step is to update the matrices $\mathbf{A}$ and $\mathbf{B}$ through a pointwise maximum likelihood procedure. As explained in [11], [12], this step can be viewed as the M-step in a generic EM method, while the E-step corresponds to the step of Section II-B (see also [14], [15], [16] for applications of a similar framework in other application fields).

[1]More details regarding the probabilistic interpretation of all involved variables can be found in [11, Chapters 4-8].

More precisely, using the outputs of the RTS, the E-step defines the following quantities:

$$\left\{ \begin{array}{ll} \mathbf{\Sigma} & = \frac{1}{K}\sum_{k=1}^K \mathbf{P}_k^s + \mathbf{x}_k^s(\mathbf{x}_k^s)^\top, \\ \mathbf{\Phi} & = \frac{1}{K}\sum_{k=1}^K \mathbf{P}_{k-1}^s + \mathbf{x}_{k-1}^s(\mathbf{x}_{k-1}^s)^\top, \\ \mathbf{\Gamma} & = \frac{1}{K}\sum_{k=1}^K \mathbf{y}_k(\mathbf{x}_k^s)^\top, \\ \mathbf{\Lambda} & = \frac{1}{K}\sum_{k=1}^K \mathbf{P}_k^s\mathbf{G}_{k-1}^\top + \mathbf{x}_k^s(\mathbf{x}_{k-1}^s)^\top. \end{array} \right. \tag{8}$$

Then, a lower bound of the marginal log-likelihood $\varphi(\mathbf{A}, \mathbf{B}) = \log p(\mathbf{y}_{1:K}|\mathbf{A}, \mathbf{B})$ under this model is built (see [11, Theorem 12.3], in particular [11, Eq. (12.38)], for obtaining recursively the log-marginal likelihood within the Kalman framework, and [11, Theorem 12.4] for the full derivation lower bound). One derives the M-step by maximizing this minorizing bound to update $\mathbf{A}$ and $\mathbf{B}$, which yields the closed form expressions:

$$\left\{ \begin{array}{ll} \mathbf{A} & = \mathbf{\Lambda}\mathbf{\Phi}^{-1}, \\ \mathbf{B} & = \mathbf{\Gamma}\mathbf{\Sigma}^{-1}. \end{array} \right. \tag{9}$$

Our proposed method referred later as *blind Kalman filter* (BKF) by applying (3)-(4)-(5) then (6)-(7), until stabilization of function φ. We processed each data minibatches sequentially. At the time k, the parameters are estimated by applying the EM iterates only on the past N observations, in a sliding-window manner. A warm start strategy is employed for the Kalman iterations initialization for the next window.

B. Peak load forecast

In order to address the problem of peak load forecast, we modify the observation model integrating the following label consistency constraint for an observed peak load value as

$$c_k = \mathbf{w}^\top\mathbf{x}_k + n_k, \tag{10}$$

where n_k is zero-mean AWGN with variance σ^2, and $\mathbf{w}$ is an unknown vector. It is easy to show that this new equation can actually be incorporated to the observation equation (2) by setting $\widetilde{\mathbf{y}}_k = \widetilde{\mathbf{B}}\mathbf{x}_k + \widetilde{\mathbf{v}}_k$ with

$$\widetilde{\mathbf{y}}_k = \left[\begin{array}{c} \mathbf{y}_k \\ c_k \end{array} \right], \tag{11}$$

$$\widetilde{\mathbf{B}} = \left[\begin{array}{c} \mathbf{B} \\ \mathbf{w}^\top \end{array} \right], \tag{12}$$

$$\widetilde{\mathbf{v}}_k = \left[\begin{array}{c} \mathbf{v}_k \\ n_k \end{array} \right]. \tag{13}$$

Once we have modified the model in this way, the previously described blind Kalman algorithm can be applied.

III. Experimental Evaluation

The dataset used here is the I-BLEND dataset which is collected for 52 months at the Indraprastha Institute of Information Technology, New Delhi, India. The dataset contains power consumption of student hostels, academic buildings, and administrative buildings. The data is available at 10 minutes intervals, but we have used aggregated readings at the hourly level for our experiments. On top of that, we collected the corresponding hourly weather information (temperature and humidity) at the city level. The input for each day k therefore

Building	Window	MAE			RMSE			MAPE		
		LSTM	ResNet	BKF	LSTM	ResNet	BKF	LSTM	ResNet	BKF
Lecture	1 week	0.423	0.339	**0.131**	0.721	0.581	**0.232**	27.5	20.3	**2.8**
	2 weeks	0.500	0.418	**0.224**	0.890	0.620	**0.267**	28.9	21.6	**2.9**
	4 weeks	0.921	0.811	**0.340**	1.366	1.003	**0.355**	28.1	24.3	**3.5**
Academics	1 week	0.387	0.287	**0.152**	0.714	0.603	**0.263**	29.8	26.3	**3.1**
	2 weeks	0.428	0.309	**0.159**	0.809	0.697	**0.286**	30.6	29.9	**3.3**
	4 weeks	0.876	0.733	**0.223**	1.311	1.020	**0.370**	38.9	37.0	**4.1**
Facilities	1 week	0.903	0.804	**0.237**	1.287	1.038	**0.326**	23.3	19.1	**2.1**
	2 weeks	0.967	0.829	**0.327**	1.441	1.173	**0.391**	24.8	22.3	**2.7**
	4 weeks	1.026	0.943	**0.439**	1.517	1.246	**0.430**	30.6	28.7	**3.2**
Girls Hostel	1 week	0.432	0.340	**0.149**	0.831	0.644	**0.262**	23.8	23.9	**3.2**
	2 weeks	0.592	0.487	**0.154**	0.944	0.709	**0.295**	29.2	25.2	**3.5**
	4 weeks	1.036	0.874	**0.170**	1.083	0.937	**0.375**	32.3	34.3	**4.2**
Boys Hostel	1 week	0.321	0.243	**0.107**	0.873	0.639	**0.199**	28.3	22.8	**3.1**
	2 weeks	0.469	0.391	**0.126**	0.961	0.714	**0.217**	29.8	24.1	**3.4**
	4 weeks	0.893	0.548	**0.223**	1.320	1.001	**0.286**	31.9	29.1	**4.2**

TABLE I
COMPARISON OF FORECASTING RESULTS FOR PROFILE ESTIMATION.

Building	Window	MAE			RMSE			MAPE		
		LSTM	ResNet	BKF	LSTM	ResNet	BKF	LSTM	ResNet	BKF
Lecture	1 week	0.195	0.120	**0.0062**	0.237	0.229	**0.081**	26.3	17.7	**2.5**
	2 weeks	0.195	0.120	**0.066**	0.244	0.229	**0.086**	26.8	19.3	**2.7**
	4 weeks	0.219	0.218	**0.091**	0.374	0.319	**0.098**	28.5	21.9	**2.9**
Academics	1 week	0.199	0.107	**0.066**	0.228	0.210	**0.098**	27.6	25.6	**3.0**
	2 weeks	0.201	0.108	**0.075**	0.237	0.210	**0.102**	30.7	29.5	**3.1**
	4 weeks	0.291	0.206	**0.083**	0.316	0.309	**0.115**	37.6	36.7	**3.2**
Facilities	1 week	0.117	0.103	**0.068**	0.220	0.205	**0.084**	19.4	16.6	**1.9**
	2 weeks	0.120	0.103	**0.069**	0.227	0.205	**0.087**	21.9	19.5	**2.1**
	4 weeks	0.217	0.200	**0.075**	0.322	0.294	**0.094**	28.4	26.3	**2.8**
Girls Hostel	1 week	0.184	0.105	**0.071**	0.207	0.208	**0.077**	22.0	21.6	**3.1**
	2 weeks	0.189	0.106	**0.081**	0.212	0.208	**0.086**	28.2	22.5	**3.1**
	4 weeks	0.282	0.201	**0.092**	0.306	0.301	**0.126**	31.8	31.5	**3.3**
Boys Hostel	1 week	0.165	0.103	**0.060**	0.183	0.204	**0.082**	22.5	20.1	**2.8**
	2 weeks	0.167	0.104	**0.064**	0.186	0.205	**0.085**	25.5	22.0	**2.9**
	4 weeks	0.274	0.202	**0.071**	0.295	0.296	**0.092**	31.4	28.7	**3.3**

TABLE II
COMPARISON OF FORECASTING RESULTS FOR PEAK LOAD ESTIMATION.

consists of a vector $\mathbf{y}_k$ of length 72 (24 hourly power readings, 24-hourly temperature readings, and 24-hourly humidity readings) in case of the profile estimation problem, while $\widetilde{\mathbf{y}}_k$ is of size 73 for the peak load forecasting. Note that the model could easily include weather forecasts if they were available (either perfect or imperfect) by simply adding these features in the input vectors.

We run our BKF algorithm for matrices $\mathbf{A}$ and $\mathbf{B}$ of size 24×24 and 72×24 (or 73×24, in case of peak load problem) respectively, initialized from a uniform random distribution. Note that this dimensionality of 24 for the hidden state, also corresponding to the number of hours considered per observation, was observed empirically to yield the best results. $\bar{\mathbf{x}}_0$ is initialized as a zero vector, and $\mathbf{P}_0$, $\mathbf{Q}$, and $\mathbf{R}$ are set as multiple of identity matrix with scale values 10^{-5}, 10^{-2} and 10^{-2}, respectively. For the peak load problem, we initialized $\mathbf{w}$ with all ones. In both cases, we provide the results for 5 iterations of EM, which appears enough here to reach convergence, and three different window sizes of $N \in \{1, 2, 4\}$ weeks, using sliding windowing with an overlap of 1 day. We have compared with two recent state-of-the-art deep learning techniques (ResNet [4] and LSTM [5]) whose parameters were taken from the corresponding papers. Both methods are trained using the first half of the total number of days, as a training set. After that from the first window of the test data, the trained model is used for prediction. For comparing the prediction accuracy of all the methods, we have used standard performance metrics: mean absolute error (MAE), root mean squared error (RMSE) and mean absolute percentage error (MAPE). All the codes are run in Python 3.6 and Pytorch environment.

We first present, in Table I, the results for profile estimation. BKF is used to predict the entire output vector (of length 72) for the next day, provided the records for past data within the considered time window. The prediction accuracy is evaluated by means of MAE, RMSE, and MAPE, on the power readings (i.e., the first 24 values of the output vector). In Table II we show the results of the BKF in the peak load estimation problem. Here, accuracy is computed only for the prediction of future peak load values (i.e., last value of the output vector). In both cases, we consider the predicted mean given the past observations, given by $\widehat{\mathbf{y}}_{K+1} = \mathbf{B}\mathbf{x}_K^-$.

From both tables, we find out that our proposed method yields the best results by a large margin. We reduce the

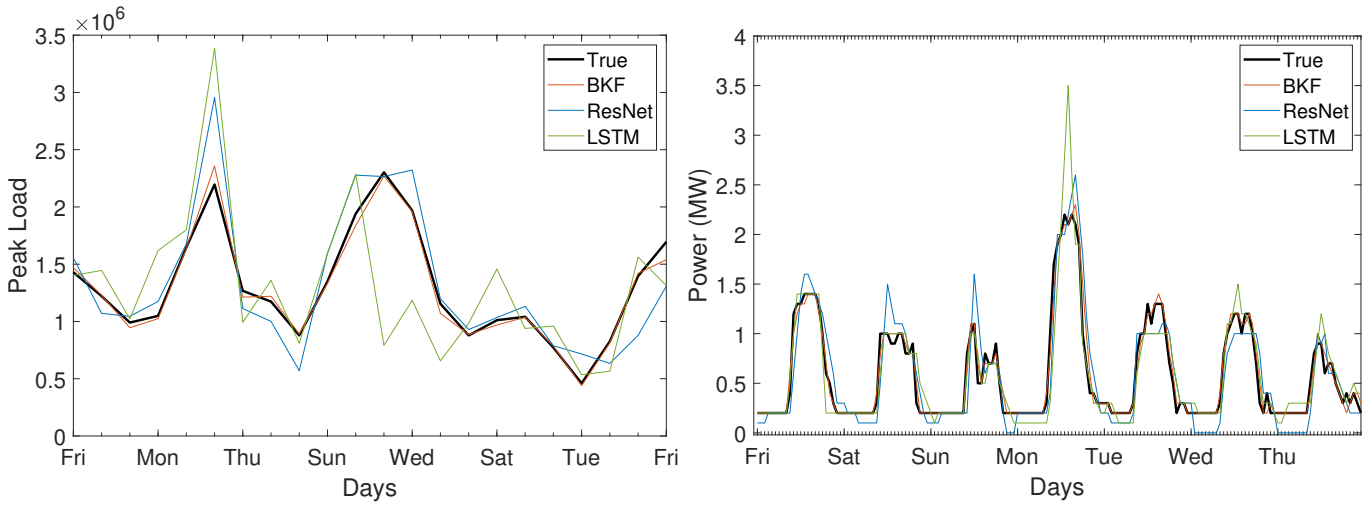

Fig. 1. Results of the prediction of daily peak load (left) and hourly power readings (right) for BKF, ResNet and LSTM, in case of 1 week window for `Academics` dataset.

error in terms of MAE and RMSE by half, compared to the next best performing technique. In terms of MAPE, we reduce the error by almost an order of magnitude. Note that when we increase the window size, the performance of all the techniques deteriorates. This may be because the data is non-stationary, so that keeping a long memory of the past reduces the performance.

We also provide in Figure 1 an example of forecasting results of daily peak load as well as hourly power readings, for our BKF method, as well as for LSTM and ResNet, in the dataset `academics`. Here we use a 1-week sliding-window with overlap of 1 day. The hourly power reading profile shows a rather interesting pattern throughout the day. Let us remind that this dataset is associated to an academic blocks where faculty members and PhD students sit and classes are held. As expected, the power consumption is lower at night and gradually ramps up from morning (between 9am and noon). One can then notice a slight dip in consumption between 1pm and 2pm explained by lunch hour so that many faculty member's rooms and classrooms are unused. Late in the evening, one can observe a slight rise in consumption, mostly because graduate students tend to work after hours (being a residential campus). One can also notice on the daily peak load profile that the peak consumption for the weekdays is higher than that of the weekend. We observe a great predictive performance of our BKF method in both situations.

IV. Conclusion

In this work, we have proposed a novel method for short-term load forecasting. It is based on the linear state-space model with unknown state and observation matrices that are sequentially estimated from the data. Our method allows to predict the next day load, given past observed data within a given time window. It operates on a small segment of the entire time series and assumes that segment to be linear. This letter shows that the strategy is effective when compared to other state-of-the-art approaches. In the future, we can consider an unequal and adaptive choice of the window length.

References

[1] Y. Chen et al., "Short-Term Load Forecasting: Similar Day-Based Wavelet Neural Networks," IEEE Transactions on Power Systems, vol. 25, no. 1, pp. 322-330, Feb. 2010.
[2] I. Moghram and S. Rahman, "Analysis and evaluation of five short-term load forecasting techniques," IEEE Transactions on Power Systems, vol. 4, no. 4, pp. 1484-1491, 1989.
[3] K. Y. Lee, Y. T. Cha and J. H. Park, "Short-term load forecasting using an artificial neural network," IEEE Transactions on Power Systems, vol. 7, no. 1, pp. 124-132, 1992.
[4] B. J. Chen, M. W. Chang and C. J. Lin, "Load forecasting using support vector Machines: a study on EUNITE competition 2001," IEEE Transactions on Power Systems, vol. 19, no. 4, pp. 1821-1830, 2004.
[5] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu and J. He, "Short-term load forecasting with deep residual networks," IEEE Transactions on Smart Grid, vol. 10, no. 4, pp. 3943-3952, 2019.
[6] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," IEEE Transactions on Smart Grid, vol. 10, no. 1, pp. 841-851, 2019.
[7] R. E. Kalman, "A new approach to linear filtering and prediction problems," Journal of Basic Engineering, vol. 82, pp. 35–45, 1960.
[8] T. Launay, A. Philippe and S. Lamarche, "On particle filters applied to electricity load forecasting," arXiv preprint arXiv:1210.0770, 2012.
[9] Y. Liu, W. Wang and N. Ghadimi, "Electricity load forecasting by an improved forecast engine for building level consumers," Energy, vol. 139, pp.18-30, 2017.
[10] J. G. Jetcheva, M. Majidpour and W. P. Chen, "Neural network model ensembles for building-level electricity load forecasts," Energy and Buildings, vol. 84, pp. 214-223, 2014.
[11] S. Sarkka, Bayesian Filtering and Smoothing, 3rd edition, 2013.
[12] R. H Shumway and D. S. Stoffer, "An approach to time series smoothing and forecasting using the EM algorithm," Journal of Time Series Analysis, vol. 3, no. 4, pp. 253–264, 1982.
[13] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," Journal of the Royal Statistical Society: Series B, vol. 39, no. 1, pp.1–22, 1977.
[14] L. Frenkel and M. Feder, "Recursive Expectation-Maximization (EM) algorithms for time-varying parameters with applications to multiple target tracking," IEEE Transactions on Signal Processing, vol. 47, no. 2, pp. 306–320, 1999.
[15] M. E. Khan and D. N. Dutt, "An expectation-maximization algorithm based Kalman smoother approach for event-related desynchronization (ERD) estimation from EEG," IEEE Transactions on Biomedical Engineering, vol. 54, no. 7, pp. 1191–1198, 2007.
[16] G. W. Pulford and B. F. La Scala, "Map estimation of target manoeuvre sequence with the expectation-maximization algorithm," IEEE Transactions on Aerospace and Electronic Systems, vol. 38, no. 2, pp. 367–377, 2002.