Electricity Load Forecasting: Ensemble Approach

Hisashi Takeda

Doctor of Philosophy

Department of Statistical Science
School of Multidisciplinary Sciences
SOKENDAI (The Graduate University for Advanced Studies)

Electricity Load Forecasting: Ensemble Approach

Hisashi Takeda

Department of Statistical Science

SOKENDAI (The Graduate University for Advanced Studies)

This dissertation is submitted for the degree of

Doctor of Philosophy

School of Multidisciplinary Sciences

August 8, 2016

To Chiaki and Ririsa

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text.

Hisashi Takeda
August 8, 2016

Acknowledgements

I would like to acknowledge Professor Yoshiyasu Tamura, Associate Professor Genta Ueno at the Institute of Statistical Mathematics in Japan, and Associate Professor Seisho Sato at the University of Tokyo. Special thanks are due to the Tokyo Electric Power Company, Inc. for sharing their valuable knowledge.

Abstract

The purpose of this research project is to develop a modeling framework for forecasting electricity load and analyzing the structure of the electricity-load behavior. The theme "Electricity Load Forecasting: Ensemble Approach" was chosen to explore the applicability of an ensemble-based data-assimilation technique for both load forecasting and structural analysis.

The first chapter of the thesis introduces the historical background that explains why utilities need structural analysis on electricity-load behavior and then discusses the problems concerning load forecasting. The second chapter shows data sources available for forecasts of load, including photovoltaic power (PV). In the third chapter, existing state-of-the-art forecasting techniques are reviewed. The fourth chapter illustrates the theoretical background regarding state-space models (SSMs), ensemble filtering methods, and model performance metrics. In the fifth chapter, SSMs for load forecasting are developed and compared to existing methods. Shrinkage or multiple linear regression methods are introduced to further enhance accuracy. In the sixth chapter, SSMs for PV generation are developed. In the final chapter, a summary and some conclusions are provided. This thesis demonstrates that the forecasting performance of the proposed models is significantly better than the performance of existing models; therefore, the proposed modeling framework is a promising technique.

The original contribution to knowledge is that the methodology of making ensemble-based structural models suitable for load forecasting is developed for the first time, and the effectiveness of using an ensemble-based method is clearly demonstrated through numerical experiments. The developed framework opens the door to more flexible and accurate modeling with the capability of load analysis, an advantage that existing methods do not provide. The framework has the potential for remarkable economic impacts on utilities. It helps solve emerging problems such as the low accuracy of load forecasts caused by the rapid increase in PV generation; hence, it minimizes the generation cost of thermal units and reduces imbalance charges for electric-power disparities between forecasts and their physical deliveries.

Contents

List of Figures

List of Tables

Listings

Nomenclature

Roman Symbols

ah_t binary variable for day following a holiday or weekend at hour t

AR_t autoregressive component at hour t

C_m total photovoltaic power capacity in month m

c_t total photovoltaic power capacity at hour t

$date_t$ calendar date at hour t

$\Delta tp_{2,t}$ difference between 2-hour-mean temperature and ST_t at hour t [°C]

$\Delta tp_{48,t}$ difference between 48-hour-mean temperature and ST_t at hour t [°C]

e_m forecasting/ filtering error in month m

HM_t humidity effect component at hour t

hm_t relative humidity at hour t [%]

ho_t binary variable for holiday at hour t

$INDAY_t$ intra-daily (24-hour periodic) component at hour t

$INDAYS_t$ intra-daily (24-hour periodic) component after smoothing at hour t

$INWK_t$ intra-weekly (168-hour periodic) component at hour t

$INWKS_t$ intra-weekly (168-hour periodic) component after smoothing at hour t

Y_t electricity load at hour t

MIN_t minimum value of the $INDAY_t$ at hour t

N_{kx} number of state variables

N_m number of members in ensemble

N_t number of hours predicted

$p_{i,t}$ photovoltaic power from the ith area at hour t

p_t total photovoltaic power at hour t

PV_t photovoltaic effect component at hour t

L_i, L trade-off parameter for system noise variance

R covariance matrix of observation noise

RA_t solar radiation effect component at hour t

ra_t amount of global solar radiation from hour $t-1$ to t [MJ/(m^{2}h)]

$r_{i,t}$ mean global solar radiation of the i^{th} observatory from hour $t-1$ to t [W/m^2]

ST_t cooling/heating switch-off temperature at hour t

TP_t temperature effect component at hour t

$tp_{x,t}$ x-hour mean temperature at hour t [°C]

$tp_{2,t}$ 2-hour mean temperature at hour t [°C]

$tp_{48,t}$ 48-hour mean temperature at hour t [°C]

$TREND_t$ trend component at hour t

$v_t^{(i)}$ the i^{th} system noise at hour t

$v_m^{(i)}$ the i^{th} system noise in month m

W_t observation noise for electricity load model at hour t

W_m observation noise for photovoltaic monthly purchase volume model in month m

$w_{i,m}$ weight for the total capacity to make the ith local capacity in month m

DAY_t effect of the day of the week (day-effect component) at hour t

tu_t, we_t, th_t, fr_t, sa_t, and su_t binary variables for days of the week at hour t

WS_t wind effect component at hour t

ws_t wind speed at hour t [m/s]

YR target year that includes the one-week forecasting term

Greek Symbols

$\alpha_{t,j(t)}$ coefficient for MIN_t at hour t on the day of the week $j(t)$

β coefficient vector for a design matrix

$\beta^{(hm)}$ humidity response indicator

$\beta^{(pv)}$ photovoltaic power response indicator

$\beta^{(ra)}$ solar radiation response indicator

$\beta^{(ws)}$ wind speed response indicator

δ_t instantaneous temperature response factor at hour t

δ_m photovoltaic conversion factor in month m

η_m photovoltaic conversion coefficient in month m

γ_t instantaneous temperature response indicator at hour t

γ'_t cumulative temperature response indicator hour t

$\kappa_{i,t}$ photovoltaic cell temperature factor from ith area at hour t

$\phi_{1,t}$, $\phi_{2,t}$ the first and the second coefficients of AR(2) at hour t

Superscripts

$(\cdot)$ identification number of a particular component or parameter

Subscripts

i, j, k indexes for general purpose

$j(t)$ index for day of the week

m index for month

t index for hour

Other Symbols

f function that gives the relationship between intra-daily and day effects

$[\cdot]_+$ hinge function

h map function

$\mathcal{L}_t$ likelihood function at hour t

$\mathcal{L}_m$ likelihood function in month m

$\mathcal{N}$ normal distribution

Acronyms / Abbreviations

4D-Var Four-Dimensional Variational Data-Assimilation Algorithm

AE_t Absolute Error at hour t

ANNs Artificial Neural Networks

ARIMA Auto-Regressive Integrated Moving Average

DM Diebold-Mariano (test)

EAKF Ensemble Adjustment Kalman Filter

EKF Extended Kalman Filter

EnKF Ensemble Kalman Filter

EnKF/PO Ensemble Kalman Filter with Perturbed Observations

EnKF/SR Ensemble Kalman Filter with Square-Root Algorithm

EnSRF Ensemble Square-Root Filter

ETKF Ensemble Transform Kalman Filter

HP Hourly Periodic (state-space models with Kalman filter)

KF Kalman Filter

Lasso Least Absolute Shrinkage and Selection Operator

LHS Left-Hand Side of an equation

MAE Mean Absolute Error

$MAPE$ Mean Absolute Percentage Error

MBE Mean Bias Error

MLR Multiple Linear Regression

PCT Photovoltaic Cell Temperature

PE_t Percentage Error at hour t

PE_m Percentage Error in month m

PF Particle Filter

PV Photovoltaics

RegARIMA Auto-Regressive Integrated Moving Average with external Regressors

RHS Right-Hand Side of an equation

$RMSE$ Root Mean Squared Error

RW weekly Random-Walk model

$SD_{\{PE\}}$ Standard Deviation of percentage error

SRF Square-Root Filter

SSMs State-Space Models

TEPCO Tokyo Electric Power Company

UC Unobserved Component (models)

UKF Unscented Kalman Filter

Chapter 1

Introduction

1.1 Motivations

1.1.1 Need for Structural Analysis on Electricity-Load Behavior

Following the Great East Japan Earthquake of 2011, most nuclear power plants in Japan were shut down due to safety concerns. Consequently, this caused an unprecedented tightening of the supply-demand balance for electricity. The earthquake also caused the public to be more energy conscious, and this has accelerated the widespread use of energy-saving appliances, such as light emitting diodes (LEDs). To obtain environmentally friendly power supplies, several incentives have been introduced to facilitate the installation of renewable energy supplies; hence, the number of these installations is growing rapidly. These changes affect the electricity load on various timescales—days, weeks, and years. Under these circumstances, it becomes increasingly important for utilities to properly monitor changes in the electricity load in order to secure a stable power supply and make a proper plan for investing in power facilities. For covering peak loads with a limited power source, it is necessary to accurately plan pumped-storage hydropower operations at least a week in advance, and this requires accurate load forecasting. When accurate forecasts are needed, most utilities [e.g., 40] use statistical methods, such as multiple linear regression (MLR) or artificial neural networks (ANNs). However, these are not suitable for analyzing the load, since they tend not to provide any insight into the cause of a structural change; for example, regression coefficients estimated using highly correlated explanatory variables are usually very large positive or negative values, and they offer no information on cause and effect.

1.1.2 Emerging Problems Concerning Renewable Energy

Since 2012, electric utilities in Japan have been obligated to purchase excess renewable energies at a fixed price through a government-guaranteed period. Subsequently, the installed capacity of photovoltaic (PV) generation has increased rapidly. Compared with other renewable energies, the feed-in-tariff rate for PV systems is relatively high (e.g., ¥42/kWh for 20 years). In addition, the installment cost and environmental requirements for the system have been low in comparison. These advantages have led to a boom in investment in PV systems. The wide variation of PV power generation, which is dependent on the weather, necessitates short-term PV power forecasting in order to maintain supply-demand balance in a power system. This balance is maintained by system operators through short-term electricity load forecasting. However, the difficulty involved in hourly PV power estimation lowers the accuracy of load forecasting. This problem is described in detail as follows. Figure 1.1 shows the relationship between electricity load and PV power. PV self-consumption, which is power consumption within houses or firms of PV suppliers, is shown above the load curve, which is indicated by the thick black line. Although PV self-consumption is not a part of the load, on cloudy days, it decreases, and the load curve increases to compensate for the electricity shortfall in houses. The remainder of the PV power, more than 85% of the total PV power generated, is sold to a utility as a power source, which is shown as the area just below the load curve. Thus, both sold and self-consumed PV power affect utilities, and due to the influence of weather, PV power as a power source is virtually uncontrollable. Since the target of load forecasting is a load that contains such PV power, it is also important to accurately forecast PV power generation on an hourly basis.

Hourly PV power forecasting is not an easy task for major utilities, especially those without a remote monitoring system for power-consumption, also referred to as a smart-meter system.[1] The difficulty in proper PV forecasting is that utilities without a smart-meter system cannot measure both the hourly PV power generation, which inflows to the power grid, and PV self-consumption. Instead, only reported monthly PV purchase volumes and hourly weather information are available (observational and two-week-ahead forecasts). Therefore, we must estimate hourly PV power generation based on these data. Major utilities in Japan have used physics-based models for PV forecasting. Since these models do not have a process of model-fitting to observational data, a severe bias problem arises that directly leads

[1] All households in Japan will be equipped with smart meters that are capable of reporting PV power by 2020 (earliest estimate).

to a large imbalance charge.[2]

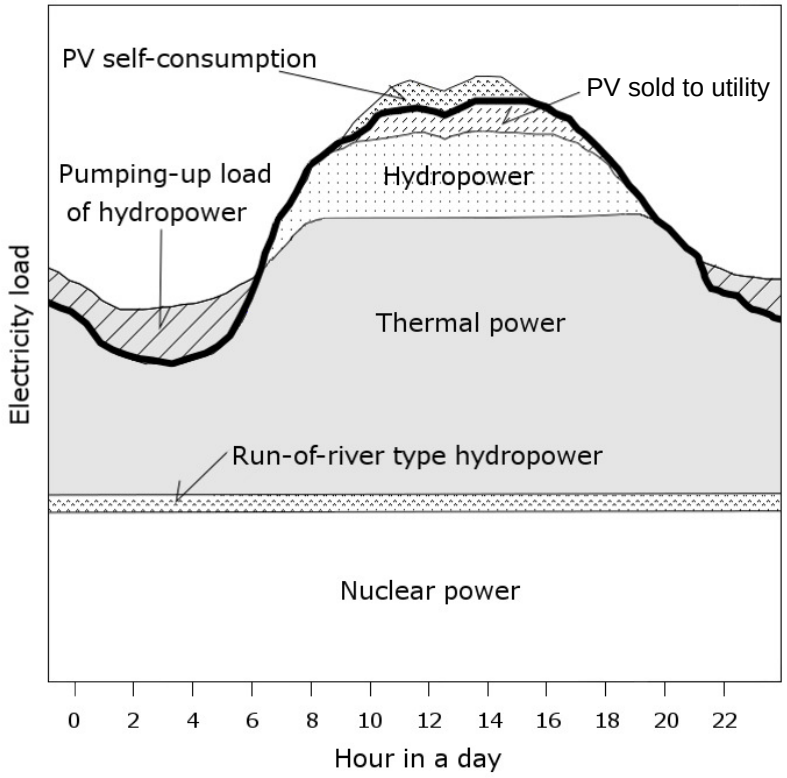

Fig. 1.1 Electric power supply from various power sources over the course of a typical sunny day. The bold line indicates the intra-daily load curve. Note that nuclear power is not included due to the forced shutdown of nuclear power plants following the Great East Japan Earthquake of 2011 at the time of writing.

1.2 Objectives

To solve the above-mentioned problems, our goal is to develop a modeling framework that can be used for both forecasts and analysis on the load including PV power. Utilities require analytical frameworks that can be used to explain the physical or economic rationales behind load changes or inaccurate forecasts to management or system supervisory organizations, such as the Organization for the Cross-regional Coordination of Transmission Operators, Japan. At the same time, the load forecasting should be accurate enough that it can be used

[2]Imbalance charges of 53.21 ¥/kWh (summer), 47.03 ¥/kWh (other seasons), and 28.84 ¥/kWh (at night) ¥/kWh for forecasting errors greater than 3%, and 15.44 ¥/kWh for forecasting errors within 3%. (http://www.tepco.co.jp/corporateinfo/provide/engineering/wsc/yakkan2604-j.pdf)

in daily operations. Thus far, different methods have been used for each purpose, since in most cases, they are incompatible. A typical load structural analysis is performed by estimating the electricity consumption based on the penetration of electrical appliances, the response to the weather, the stay-at-home rate, and other economic statistics. The models used for the analysis are more focused on accountability than accuracy.

1.3 Outline of the Thesis

In this thesis, we proposed an hourly short-term load forecasting method and an hourly PV physics-based model which effectively assimilates with monthly PV-purchase data.

The remainder of the thesis is structured as follows.

Chapter 2 describes data sources used for the forecasts of both electricity load and PV power.

Chapter 3 reviews the literature and investigates existing state-of-the-art methods in the forecasting research fields.

Chapter 4 provides the theoretical background for state-space models (SSMs) and ensemble Kalman filter (EnKF), which will be used in the following two chapters.

Chapter 5 illustrates methods for the load forecasting and provides performance evaluation in comparison with the existing state-of-the-art methods. Our aim is to develop SSMs with enough load forecasting accuracy to ensure that the accountability assigned by the load analysis is correct. Compared with the current state-of-the-art methods, the proposed method significantly improves the forecasting accuracy. For load structural analysis, weather-response indicators which are needed for official reports and require additional analyses (e.g., simple regression analysis) in existing methods are directly estimated in our method; this was not discussed in any of the studies that we reviewed.

Chapter 6 illustrates methods for the PV power forecasting and provides a performance evaluation as well. The proposed PV method solved the severe bias problem by drastically reducing forecasting bias. Moreover, although the proposed model is simple, it outperformed the results of a benchmark model currently in operational use. PV system parameters such as the coefficient and the factor of PV conversion can be directly estimated using the proposed method.

The final chapter provides a summary and the conclusions of the thesis. We have successfully developed a unique modeling framework that can be used for load forecasting and analysis, and thus our goal has been achieved.

Chapter 2

Data Sources

2.1 Electricity Load

An electricity load model was developed using hourly load data available from the Tokyo Electric Power Company (TEPCO), which covers metropolitan Tokyo and the surrounding area. Load data are downloadable in the CSV (comma-separated values) file format from the TEPCO website.[1] Figure 2.1 shows the sources of electric power supplied over the course of a typical day. The thick line shows the target load, which is also known as the *intra-daily load curve*. The electricity demand in the service area includes PV self-consumption; this appears at the top of the curve. Note that TEPCO's actual load includes the pumping-up load of hydropower. The load shows an overall trend, as well as intra-weekly and intra-daily periodic variations. These features can be seen in Fig. 2.2: the load decreases at night and increases during the day; a small dip usually occurs around lunch time (i.e., between 12:00 and 13:00); and there are marked drops on weekends (Sa and Su).

[1] TEPCO Electricity Forecast: http://www.tepco.co.jp/en/forecast/html/index-e.html

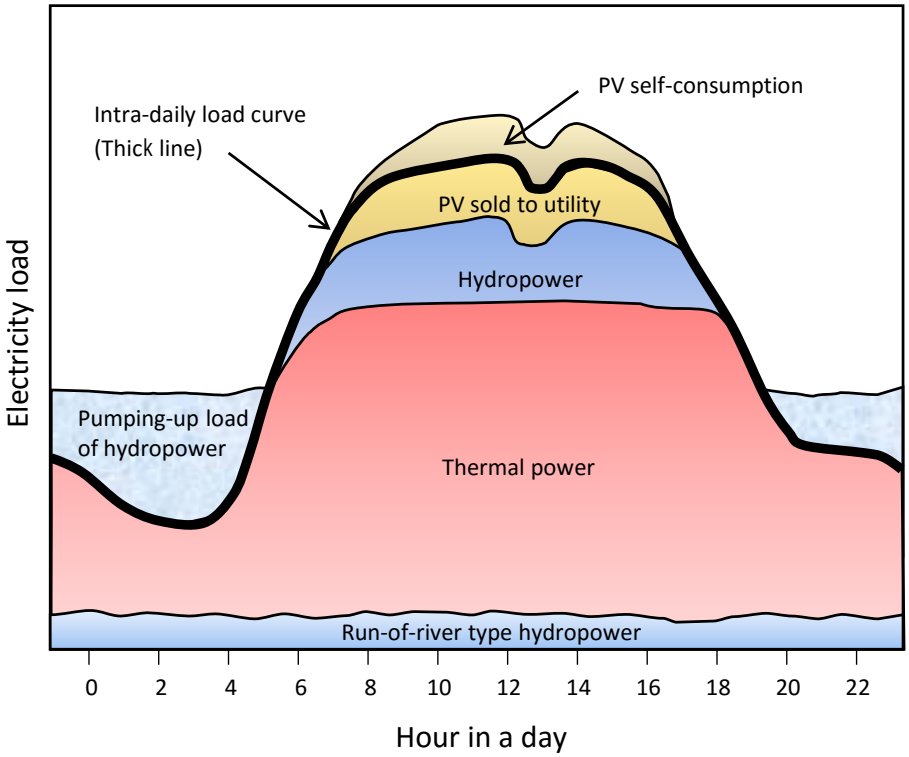

Fig. 2.1 Intra-daily load curve

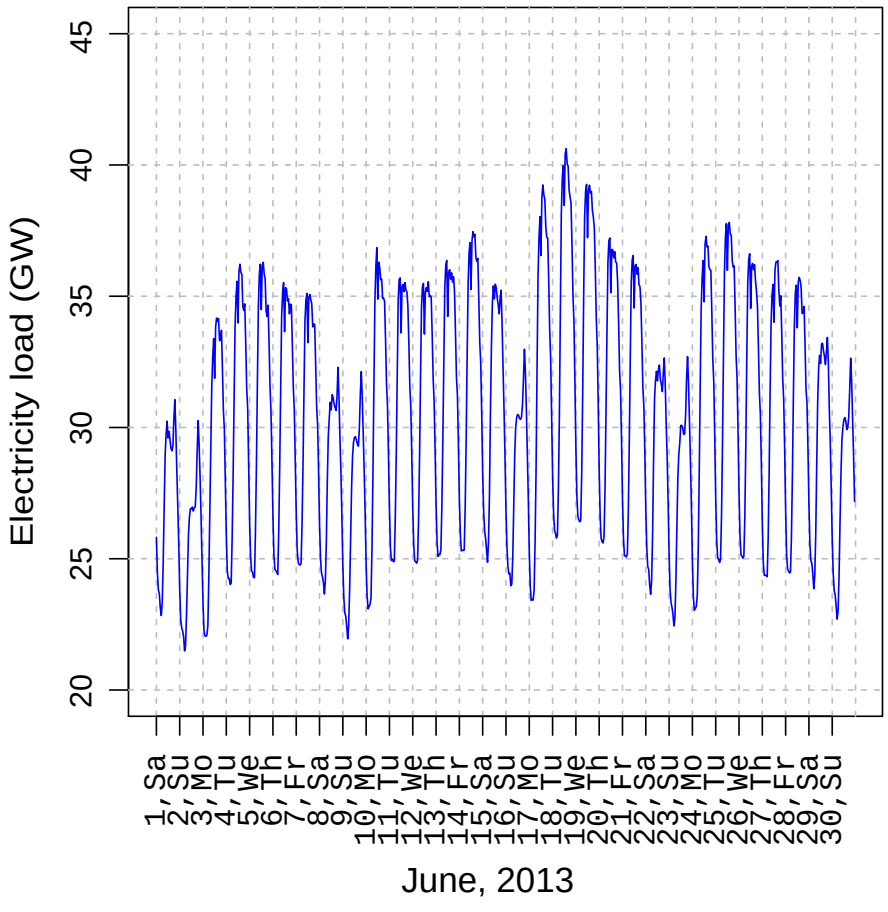

Fig. 2.2 Intra-daily load curve for June 2013

2.2 Photovoltaic (PV) Power

The following data were used for the PV power forecast in Chapter 6. For confidentiality reasons, in the following figures, we will use sequential numbers rather than real time stamps for PV data.

2.2.1 PV Purchase Volume

There are two types of PV-supplier categorized by capacity: capacity less than 10 kW (type I), and 10 kW or above (type II). Type-I suppliers can sell only excess power which equals to the total generated power less self-consumption. On the other hand, type-II suppliers can sell the whole generated power. Figure 2.3 shows monthly PV purchase volume of both suppliers. The PV purchase volume from the type-II supplier has increased at by far faster pace than that from the type-I supplier.

2.2.2 Installed PV Capacity

We used historical records of installed PV capacity of each type in the utility service area. These records are routinely reported to utility companies every month. The installed PV capacities of both suppliers are plotted in Fig. 2.4. The data can be downloaded in the CSV file format from the Agency for Natural Resources and Energy. [2]

Area Rate of Installed PV Capacity by Individual Supplier

The target PV power for forecasting is an aggregated power from large numbers of PV systems spread over the utility service area. Figures 2.5 and 2.6 show the area rates of installed PV capacity of type-I and type-II suppliers, respectively. The two supplier types have very different installation patterns. The majority of type-I suppliers are households. As such, population-dense areas like Tokyo largely have type-I suppliers. In contrast, type-II suppliers are primarily PV firms. Therefore, suburban areas have mostly type-II suppliers. This suggests that separate estimation of PV power depending on supplier type be a reasonable strategy.

[2] http://www.fit.go.jp/statistics/public_sp.html

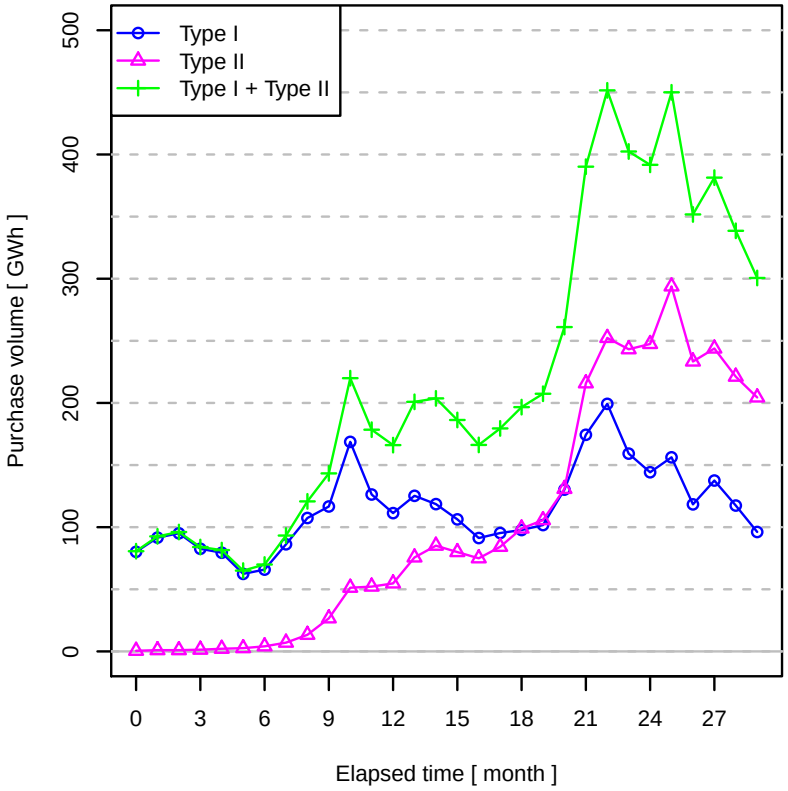

Fig. 2.3 PV purchase volume for 2012–2014

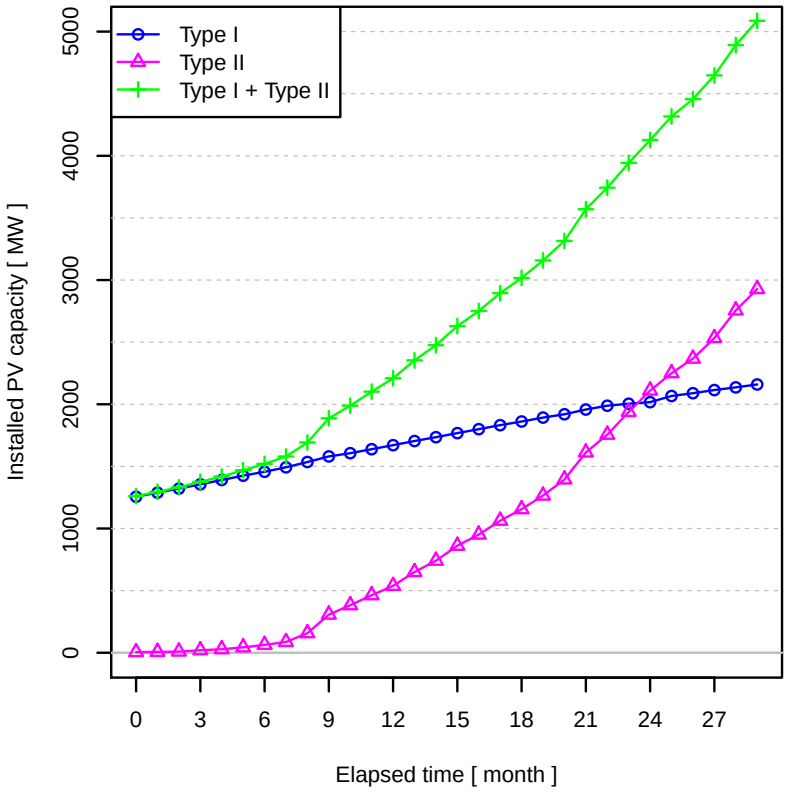

Fig. 2.4 Installed PV capacities of type-I and type-II suppliers

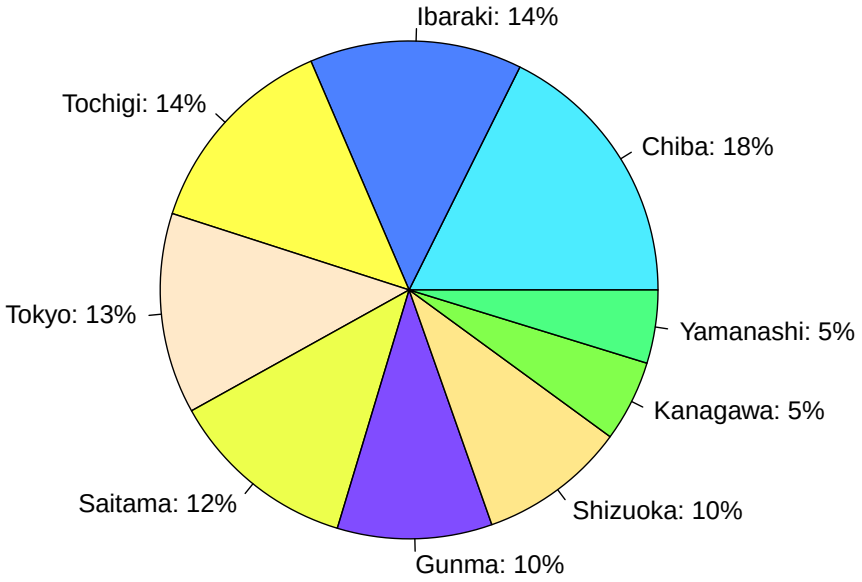

Fig. 2.5 Installed PV capacity rates for type-I suppliers by prefecture in 2014

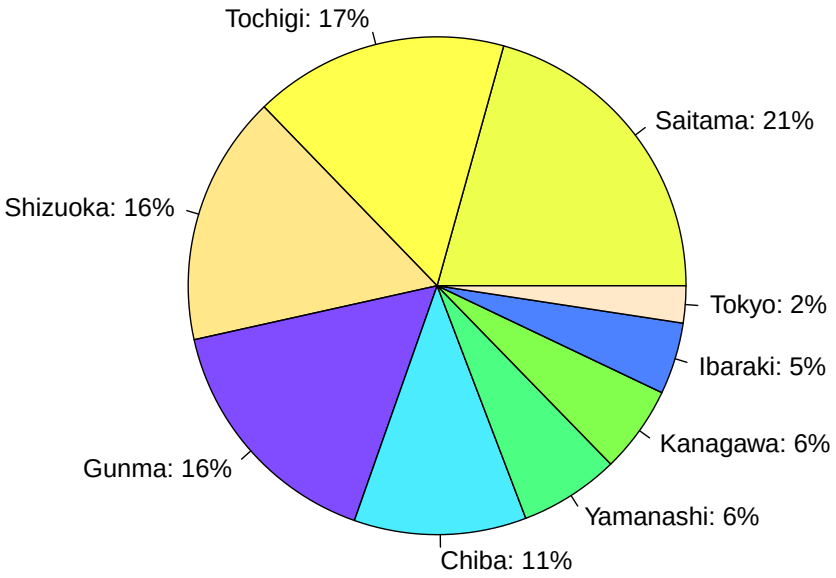

Fig. 2.6 Installed PV capacity rates for type-II suppliers by prefecture in 2014

2.3 Weather and Calendar Information

For model fitting and evaluation, we used weather observations obtained from the website of the Japan Meteorology Agency.[3] Calendar information (such as day of the week and holidays) was also used. There are nine geographically separated observatories located in the utility service area as shown in Fig. 2.7. From the observatories, every 30 minutes,[4] the following observations were available for our study:

- Air temperature [°C]
- Relative humidity [%]
- Global solar radiation [MJ/(m^{2}h)] or [W/m^2]
- Wind speed [m/s]

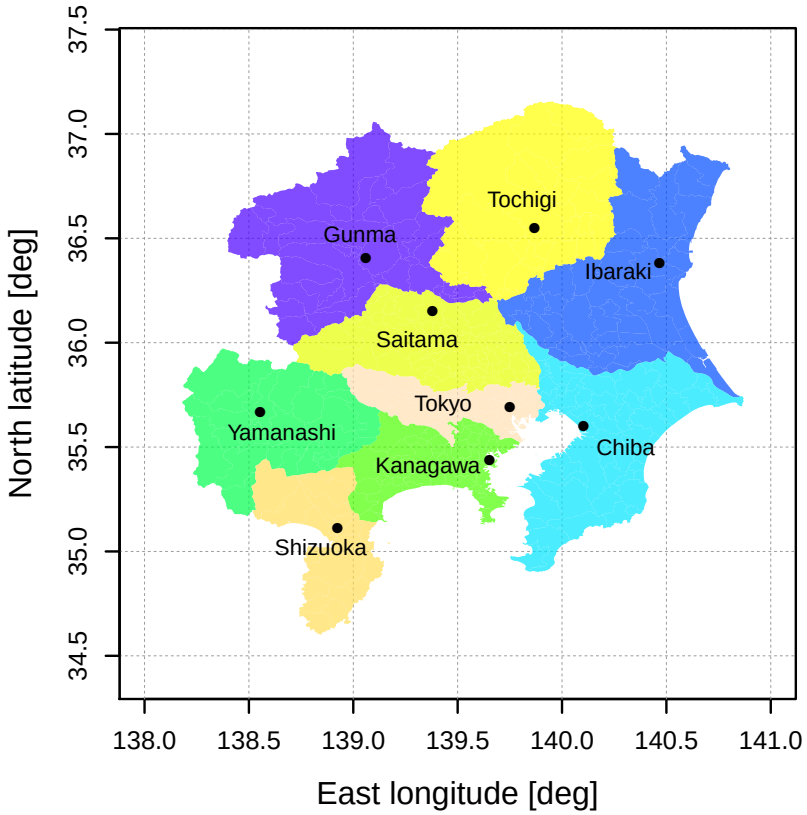

Fig. 2.7 Weather observatories in the utility service area. With a prefecture name, each dot (•) shows a geographic location of a observatory.

[3] Japan Meteorological Agency, *Past observation data*: http://www.data.jma.go.jp/obd/stats/etrn/index.php
[4] At the time of this study, only hourly observations were provided.

Chapter 3

Literature Review

3.1 Electricity Load Forecasting

Internationally, there have been hundreds of studies about load forecasting, and these have considered the use of many different statistical techniques. Although it is impractical to list all of these techniques here, the following ones are those that are commonly used. The most widely used technique for load forecasting is multiple liner regression (MLR) [10], although machine learning techniques have gained in popularity in recent years; examples include fuzzy inference [59], support vector machines [45], and particle swarm optimization [6]. Singular value decomposition has been used for robust estimations and dimension reduction [39], and the Gaussian process has been used for nonlinear modeling [47]. A large number of neural-network-based methods [e.g., 24] have been studied; their main purpose is to handle nonlinearity in a system. State-space models (SSMs) and the Box-Jenkins autoregressive integrated moving average (ARIMA) [e.g., 50] have been used since the early days of load-forecasting research.

For load analysis, structural time-series models are commonly used, and these are often used to forecast yearly growth in the load [e.g., 73]. However, recent methods use both weather and economic indicators in an attempt to create a forecast that is seamless from the short term to the long term [57]. In this study, we consider a forecasting horizon of one week, which is considered to be short-term load forecasting. We have thoroughly reviewed all the papers that propose methods for forecasts of up to several weeks. Harvey and Koopman [27] used time-varying splines to model periodic changes in the load, and they showed the necessity of incorporating an evolutionary process in a forecasting model. Taylor [62] developed a scenario-based forecasting model that used 51 different weather

ensemble members. Several exponential smoothing techniques using SSMs have been developed [e.g., 63]. SSMs have been developed for the national load in France [14] and the regional load in the UK [53], and in Section 5.3.3, we use these results to evaluate the accuracy of our SSMs. We note that one of the advantages of SSMs is that individually created models (submodels or components) can easily be incorporated into a single model; for example, a nonlinear model for temperature effects can be easily incorporated as a part of the load model. Another advantage is that SSMs can be updated recursively, and this is appropriate for modeling the natural evolution mechanism of the load components. SSMs have a long history and have been extensively studied; however, thus far, only a few attempts have been made to use them to model nonlinearity. For flexible nonlinear modeling, we use the ensemble Kalman filter (EnKF) as the algorithm for estimating the SSMs. Generally speaking, forecasts obtained from SSMs tends to be less accurate than those produced by the black-box methods that are used by many utilities. The accuracy of our method was improved by using a shrinkage method, the least absolute shrinkage and selection operator (Lasso) [68], and MLR. In any method, for increasing the forecasting accuracy and stability, it is important to select the proper explanatory variables; for example, using correlation analysis to select input variables has been shown to increase forecasting accuracy [24]. We used the Lasso to select the variables, since it has the additional advantage of reducing overfitting, as compared with the step-wise methods that are commonly used in practice.

3.2 Photovoltaic Power Forecasting

A thorough review of existing PV forecasting methods revealed that the problem mentioned in Section 1.1.2. has not yet been considered in past research. Since the history of PV forecasting research is very short compared with that of load forecasting, PV forecasting techniques are less diverse.

In the following, we present an overview of PV forecasting technology. Satellite images with cloud motion are commonly used for short-term (within several hours) forecasting [12], whereas physics-based models are usually used for longer-term (more than six hours) forecasting. Most PV forecasting techniques preliminarily predict solar irradiance using widely available numerical weather prediction [38]. For forecasting periods of more than one year, classical seasonal decomposition models are used to decompose time series data into seasonal components, trend components, and irregular components [56]. As an example, the Kalman filter has been successfully used to remove the bias of solar irradiance forecasts [7].

We herein focus on short-term forecasting, which is our primary interest. Artificial intelligence (AI) methods, such as ANNs, have been most commonly used in hourly PV forecasting. For example, several ANNs with distinct topologies have been used for PV forecasting, and two solar modules produced by major manufacturers have been tested [48]. A recurrent neural network has been successfully applied to several hour-ahead PV power forecasting [76]. For other AI methods, hybrid hourly forecasting using a genetic algorithm to combine ARIMA and three artificial intelligence methods have been proposed [77]. The hybrid model outperformed these four models, and solar radiation and empirical PV hourly power data are the only input data for the model. Note that some studies have used actual hourly PV power data as training data. However, these studies considered only a small amount of aggregated power from experimental residential areas or from a few PV firms, which is in contrast to the present study which considers the total PV power for an entire utility service area.

Forecasting methods that do not require knowledge of PV systems are gaining in popularity. The hourly quantile regression model is used for one-day-ahead forecasting [2]. Forecasting techniques that do not consider solar radiation have been accessed, and ANNs have been demonstrated to outperform ARIMA and k-nearest-neighbors algorithms [54]. A reforecasting technique to remove systematic bias has also been developed [11]. Feature extraction from solar irradiance and weather pattern recognition [74] and regularized linear/nonlinear models [1] have also been developed.

Two basic types of strategies are usually used for PV forecasting of total power: bottom-up strategies, which aggregate locally forecasted PV power generation, and direct strategies, which directly forecast the total PV power generation [78]. The mean absolute error (MAE) has been reported to be reduced by more than 3% by using a bottom-up strategy, as compared to a direct strategy. In addition to this accuracy advantage, only the bottom-up strategy is capable of providing precise information about local PV power, which would contribute to solving over-voltage problems that occur in power distribution networks. Therefore, we adopted a bottom-up strategy; that is, we first forecast local PV power generation, followed by the total PV power.

3.3 Ensemble Methods

Most load forecasts that use structural time series models [e.g., 53] are based on the Kalman filter (KF) [37], though the KF has a high computational cost and is not capable of implementing nonlinear system dynamics [69]. To handle nonlinear modeling, the extended KF which uses the Taylor series expansion for nonlinear terms was developed; however, it also has a high computational cost when approximating the error covariance. Evensen (initial work [19], comprehensive work [20]) developed the EnKF, which overcomes both problems by using an ensemble representation for the error covariance. The EnKF adopts a Monte Carlo approximation to the KF, and the result is that the sample mean and covariance matrix are asymptotically the same as those of the KF. The EnKF consists of a linear observation model with Gaussian noise and a linear or nonlinear system model with any type of noise distribution. The nonlinear formulation affords much greater flexibility than does the KF, which can handle only linear models. In addition, the ensemble approximation technique drastically reduces the computational cost, and this allows us to assimilate data into systems that are too large for previous methods. Since the revolutionary success of Evensen, the EnKF as well as the four-dimensional variational data-assimilation algorithm (4D-Var) have become the most widely used algorithms for the assimilation of meteorological or oceanographic data. For example, the EnKF has been successfully applied to forecasting ozone concentrations [16], assimilating snow [46] and land surface temperature [23], and building a coupled atmosphere-ocean model [71]. Although the electricity load has a very close relationship with meteorological phenomena, studies using either the EnKF or the 4D-Var have been strangely neglected by scientists. We apply the EnKF to load forecasting and demonstrate its effectiveness for the first time. Using EnKF, which can deal with a nonlinear model, it becomes possible to easily enhance an elaborate physics-based model by incorporating observed data. Moreover, it is very easy to add uncertainty information, such as quantiles, to the point estimate, since ensemble members obtained by EnKF represent a prediction distribution. Most existing forecasting methods provide only point estimates [18]. For unit commitment for thermal plants, utilities use the forecasted load curve, which fluctuates with the PV power. Therefore, interval estimation of the load is more useful than point estimates for system operators.

There are several variants of the EnKF, including the EnKF with perturbed observations (EnKF/PO), which was the first to be introduced and is widely used in many practical applications. However, it is known that perturbed observations increase the forecasting error to some extent. To reduce this error, the ensemble square-root filter (EnSRF) filter was devel-

oped [75]; the EnSRF does not require perturbed observations. The ensemble transform KF (ETKF) [9] and the ensemble adjustment KF (EAKF) [4] are also similar square-root filters. Particle filter or Sequential Monte Carlo [25, 42] is another promising ensemble technique which no longer requires linearity assumption for both the observation model and the system model. In the present study, we use the EnSRF with Andrews' matrix formulation [5] since it is easily implemented and performs better than others. We will use the term "EnKF" to refer to the EnSRF in this thesis. Major data-assimilation methods are summarized in Table 3.1.

Table 3.1 Summary table of the major data-assimilation methods

Method	Acronym	Citation	Model type [a]		Noise type [b]		Explanation [c]
			Obs.	Sys.	Obs.	Sys.	
Kalman Filter	KF	[37]	L	L	G	G	· The most fundamental algorithm for data-assimilation · Computationally expensive
Square-Root Filter	SRF	[66], [8]	L	L	G	G	· KF with decomposed state-covariance matrix · Robust for round-off error · Computationally expensive
Extended KF	EKF	[61], [49]	NL	NL	G	G	· KF with Taylor approximation of a nonlinear model · Computationally expensive
Unscented KF	UKF	[36]	NL	NL	G	G	· Sigma points approximation of a state distribution · The points are selected with a deterministic algorithm
Ensemble KF with Perturbed Observations	EnKF/PO	[19], [20]	L	NL	G	NG	· Ensemble approach; PO method · Not robust for sampling error
Ensemble SRF	EnSRF	[75]	L	NL	G	NG	· Ensemble approach; SRF(F)-based method · Easy implementation
Ensemble KF with Square-Root Algorithm	EnKF/SR	[21]	L	NL	G	NG	· Ensemble approach; SRF(B)-based method · Easy implementation
Ensemble Adjustment KF	EAKF	[4]	L	NL	G	NG	· Ensemble approach; SRF(F)-based method · Computational advantage when $N_m > N_{kx}$
Ensemble Transform KF	ETKF	[9]	L	NL	G	NG	· Ensemble approach; SRF(B)-based method · Computational advantage when $N_m < N_{kx}$
Particle Filter	PF	[43], [25]	NL	NL	NG	NG	· Ensemble approach · Easy implementation; Similarity to genetic algorithm
Four-Dimensional Variational Data-Assimilation	4D-Var	[79]	NL	NL	NG	NG	· Variational approach · Difficult implementation

[a] L: Linear model, NL: Nonlinear model
[b] G: Gaussian noise, NG: Non-Gaussian noise
[c] F: Forward-multiplication type, B: Backward-multiplication type, N_{kx}: Number of state variables, N_m: Number of ensemble members

Chapter 4

Theoretical Background

4.1 State-Space Models

Models for a time series or a controlling system in the form of Eqs. 4.1 and 4.2 are called state-space models (SSMs) [28]. The state process is given by Eq. 4.1, and the observation process is given by Eq. 4.2. Estimates can be obtained as a sum of the separate components in a linear observation model, and thus, it is easy to modify the model. Another advantage of using SSMs is that we can use a recursive algorithm such as the KF or its variants to estimate the states. A long-term (more than one step) forecast can be obtained by repeating a one-step-ahead prediction using Eqs. 4.1 and 4.2. These equations are as follows:

$$\mathbf{x}_t = f_t(\mathbf{x}_{t-1}, \mathbf{v}_t) \,, \tag{4.1}$$

$$\mathbf{y}_t = g_t(\mathbf{H}_t, \mathbf{x}_t, \mathbf{w}_t) \,, \tag{4.2}$$

where $\mathbf{x}_t$ is the state vector, $\mathbf{y}_t$ is the observation vector, $\mathbf{H}_t$ is the observation matrix, $\mathbf{w}_t$ is the observation noise vector, $\mathbf{v}_t$ is the system noise vector, $f_t(\cdot)$ is the system model, and $g_t(\cdot)$ is the observation model.

4.2 Ensemble Filtering Methods for Data Assimilation

4.2.1 Common Notations

Notations which are commonly used in the following subsections are explained here.
First, by Eq. 4.3, we define the predicted state vector of size N_{kx} of the i^{th} ensemble member. Likewise, the filtered state vector by Eq. 4.4. Each entry is a realization drawn from the

corresponding distribution (i.e., the prediction distribution or the filtering distribution).

$$\hat{\mathbf{x}}^{(i)} := \left[\hat{x}_1^{(i)}, \cdots, \hat{x}_{N_{kx}}^{(i)}\right]' \qquad \in \mathbb{R}^{N_{kx}\times 1}, \tag{4.3}$$

$$\mathbf{x}^{(i)} := \left[x_1^{(i)}, \cdots, x_{N_{kx}}^{(i)}\right]' \qquad \in \mathbb{R}^{N_{kx}\times 1}. \tag{4.4}$$

Using $\hat{\mathbf{x}}^{(i)}$ and $\mathbf{x}^{(i)}$, by Eq. 4.5, we define the predicted state matrix $\hat{\mathbf{X}}$ with N_m state vectors. Likewise, the filtered state matrix $\mathbf{X}$ by Eq. 4.6.

$$\hat{\mathbf{X}} := \left[\hat{\mathbf{x}}^{(1)}, \cdots, \hat{\mathbf{x}}^{(N_m)}\right] \qquad \in \mathbb{R}^{N_{kx}\times N_m}, \tag{4.5}$$

$$\mathbf{X} := \left[\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N_m)}\right] \qquad \in \mathbb{R}^{N_{kx}\times N_m}. \tag{4.6}$$

Secondly, we define the mean matrices for $\hat{\mathbf{X}}$ and $\mathbf{X}$ by Eqs.4.7 and 4.8, respectively, then the deviation matrix by Eq. 4.9.

$$\underline{\hat{\mathbf{X}}} := \hat{\mathbf{X}}\mathbb{1}/N_m \qquad \in \mathbb{R}^{N_{kx}\times N_m}, \tag{4.7}$$

$$\underline{\mathbf{X}} := \mathbf{X}\mathbb{1}/N_m \qquad \in \mathbb{R}^{N_{kx}\times N_m}, \tag{4.8}$$

$$\hat{\mathbf{D}} := \hat{\mathbf{X}} - \underline{\hat{\mathbf{X}}} \qquad \in \mathbb{R}^{N_{kx}\times N_m}, \tag{4.9}$$

where $\mathbb{1} \in \mathbb{R}^{N_m\times N_m}$ is the matrix in which all elements are unity (i.e., 1.0). Note that the elements of each row of $\underline{\hat{\mathbf{X}}}$ or $\underline{\mathbf{X}}$ have the same value (i.e., the ensemble mean).

Thirdly, using the observation matrix $\mathbf{H} \in \mathbb{R}^{N_y\times N_{kx}}$, the prediction matrix $\hat{\mathbf{Y}}$ and its mean matrix $\underline{\hat{\mathbf{Y}}}$ are defined by the following equations:

$$\hat{\mathbf{Y}} = \mathbf{H}\hat{\mathbf{X}} \qquad \in \mathbb{R}^{N_y\times N_m}, \tag{4.10}$$

$$\underline{\hat{\mathbf{Y}}} = \mathbf{H}\underline{\hat{\mathbf{X}}} \qquad \in \mathbb{R}^{N_y\times N_m}. \tag{4.11}$$

Finally, we define $\tilde{\mathbf{R}} := (N_m - 1)\mathbf{R} \in \mathbb{R}^{N_y\times N_y}$ for efficient matrix operations, where $\mathbf{R}$ is the covariance matrix for observation noise $\mathbf{W} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \in \mathbb{R}^{N_y\times N_m}$.

In the following subsections, we will chronologically explain the procedures for various types of ensemble filtering methods, based on Ueno's formulations [70] which are modified for efficient matrix operations.

4.2.2 Ensemble Kalman Filter with Perturbed Observations (EnKF/PO)

The first ensemble Kalman filtering algorithm was introduced by Evensen [19][20]. Using the same notations as defined previously, EnKF/PO is performed in the following procedure:

$$\mathbf{K} = \hat{\mathbf{D}}\hat{\mathbf{D}}'\mathbf{H}'\left(\mathbf{H}\hat{\mathbf{D}}\hat{\mathbf{D}}'\mathbf{H}' + \tilde{\mathbf{R}}\right)^{-1} \quad \in \mathbb{R}^{N_{kx}\times N_y}, \tag{4.12}$$

$$\mathbf{X} = \hat{\mathbf{X}} + \mathbf{K}\left(\mathbf{Y} + \mathbf{W} - \hat{\mathbf{Y}}\right) \quad \in \mathbb{R}^{N_{kx}\times N_m}, \tag{4.13}$$

where $\mathbf{K}$ is a Kalman gain matrix.

4.2.3 Ensemble Adjustment Kalman Filter (EAKF)

Anderson developed an EnKF without perturbed observations, which significantly outperformed 4D-Var and EnKF/PO for the first time [4]. Calculation performance increases when compact singular value decomposition is possible. EAKF is categorized as a square-root filter of the forward-multiplication type. It is computationally advantageous when $N_m > N_{kx}$. Using the same notations as defined previously, EAKF is performed in the following procedure. With Eq. 4.14, compact singular value decomposition of RHS is performed. Also, with Eq. 4.15, eigendecomposition of RHS is performed.

$$\mathbf{U}\mathbf{G}\mathbf{V}' = \hat{\mathbf{D}} \quad \in \mathbb{R}^{N_{kx}\times N_m}, \tag{4.14}$$

$$\mathbf{Z}\mathbf{B}\mathbf{Z}' = (\mathbf{H}\mathbf{U}\mathbf{G})'\tilde{\mathbf{R}}^{-1}\mathbf{H}\mathbf{U}\mathbf{G} \quad \in \mathbb{R}^{N_r\times N_r}, \tag{4.15}$$

$$\mathbf{K} = \mathbf{U}\mathbf{G}\mathbf{Z}(\mathbf{I}+\mathbf{B})^{-1}\mathbf{Z}'(\mathbf{H}\mathbf{U}\mathbf{G})'\tilde{\mathbf{R}}^{-1} \quad \in \mathbb{R}^{N_{kx}\times N_y}, \tag{4.16}$$

$$\underline{\mathbf{X}} = \hat{\underline{\mathbf{X}}} + \mathbf{K}(\mathbf{Y} - \hat{\mathbf{Y}}) \quad \in \mathbb{R}^{N_{kx}\times N_m}, \tag{4.17}$$

$$\mathbf{D} = \mathbf{U}\mathbf{G}\mathbf{Z}(\mathbf{I}+\mathbf{B})^{-1/2}\mathbf{G}^{+}\mathbf{U}'\hat{\mathbf{D}} \quad \in \mathbb{R}^{N_{kx}\times N_m}, \tag{4.18}$$

$$\mathbf{X} = \underline{\mathbf{X}} + \mathbf{D} \quad \in \mathbb{R}^{N_{kx}\times N_m}, \tag{4.19}$$

where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices with left- and right-singular vectors for the corresponding singular values, respectively. $\mathbf{G}^{+}$ is a Moor-Penrose pseudoinverse with an effective rank of $N_r \leq \min(N_{kx}, N_m)$. $\mathbf{Z}$ is an orthogonal matrix whose i^{th} column is the i^{th} eigenvector of RHS. $\mathbf{B}$ is a diagonal matrix whose entries are the eigenvalues of RHS.

4.2.4 Ensemble Square-Root Filter (EnSRF)

Similar to EAKF, but with a much simpler implementation, Whitaker also developed an EnKF without perturbed observations [75]. EnSRF is categorized as a square-root filter of

the forward-multiplication type. Using the same notations as defined previously, EnSRF is performed in the following procedure. With each Eq. 4.20 and 4.21, Cholesky decomposition of RHS is performed.

$$\mathbf{U}\mathbf{U}^{'} = \tilde{\mathbf{R}} + \mathbf{H}\hat{\mathbf{D}}\hat{\mathbf{D}}^{'}\mathbf{H}^{'} \qquad \in \mathbb{R}^{N_y \times N_y}, \tag{4.20}$$

$$\mathbf{V}\mathbf{V}^{'} = \tilde{\mathbf{R}} \qquad \in \mathbb{R}^{N_y \times N_y}, \tag{4.21}$$

$$\mathbf{K}_m = \hat{\mathbf{D}}\hat{\mathbf{D}}^{'}\mathbf{H}^{'}(\mathbf{U}^{'})^{-1}\mathbf{U}^{-1} \qquad \in \mathbb{R}^{N_{kx} \times N_y}, \tag{4.22}$$

$$\mathbf{K}_d = \hat{\mathbf{D}}\hat{\mathbf{D}}^{'}\mathbf{H}^{'}(\mathbf{U}^{'})^{-1} + (\mathbf{U}+\mathbf{V})^{-1} \qquad \in \mathbb{R}^{N_{kx} \times N_y}, \tag{4.23}$$

$$\underline{\mathbf{X}} = \underline{\hat{\mathbf{X}}} + \mathbf{K}_m(\mathbf{Y} - \underline{\hat{\mathbf{Y}}}) \qquad \in \mathbb{R}^{N_{kx} \times N_m}, \tag{4.24}$$

$$\mathbf{D} = (\mathbf{I} - \mathbf{K}_d\mathbf{H})\hat{\mathbf{D}} \qquad \in \mathbb{R}^{N_{kx} \times N_m}, \tag{4.25}$$

$$\mathbf{X} = \underline{\mathbf{X}} + \mathbf{D} \qquad \in \mathbb{R}^{N_{kx} \times N_m}, \tag{4.26}$$

where $\mathbf{U}$ and $\mathbf{V}$ are upper triangular matrices with positive diagonal real entries, respectively. $\mathbf{K}_m$ and $\mathbf{K}_d$ are Kalman gains for the means and the deviations, respectively.

4.2.5 Ensemble Kalman Filter with Square-Root Algorithm (EnKF/SR)

Evensen developed another EnKF that is also based on the square-root filtering scheme of the backward-multiplication type [21]. It is demonstrated that EnKF/SR effectively overcomes the slow convergence of EnKF/PO, which is due to sampling errors introduced by perturbed observations, and a significant reduction in computing time has been achieved. Using the same notations as defined previously, EnKF/SR is performed in the following procedure. With Eq. 4.27, eigendecomposition of RHS is performed. Also, with Eq. 4.28, singular-value decomposition of RHS is performed.

$$\mathbf{Z}\mathbf{B}\mathbf{Z}^{'} = \tilde{\mathbf{R}} + \hat{\mathbf{Y}}\hat{\mathbf{Y}}^{'} \qquad \in \mathbb{R}^{N_y \times N_y}, \tag{4.27}$$

$$\mathbf{U}\mathbf{G}\mathbf{V}^{'} = \mathbf{G}^{-1/2}\mathbf{Z}^{'}\hat{\mathbf{Y}} \qquad \in \mathbb{R}^{N_y \times N_m}, \tag{4.28}$$

$$\mathbf{K}_m = \hat{\mathbf{D}}\hat{\mathbf{D}}^{'}\mathbf{H}^{'}(\mathbf{U}^{'})^{-1}\mathbf{U}^{-1} \qquad \in \mathbb{R}^{N_{kx} \times N_y}, \tag{4.29}$$

$$\underline{\mathbf{X}} = \underline{\hat{\mathbf{X}}} + \mathbf{K}_m\left(\mathbf{Y} - \hat{\mathbf{Y}}\right) \qquad \in \mathbb{R}^{N_{kx} \times N_m}, \tag{4.30}$$

$$\mathbf{D} = \hat{\mathbf{D}}\mathbf{V}\sqrt{\mathbf{I} - \mathbf{G}^{'}\mathbf{G}}\mathbf{V}^{'} \qquad \in \mathbb{R}^{N_{kx} \times N_m}, \tag{4.31}$$

$$\mathbf{X} = \underline{\mathbf{X}} + \mathbf{D} \qquad \in \mathbb{R}^{N_{kx} \times N_m}, \tag{4.32}$$

where $\mathbf{Z}$ is an orthogonal matrix whose i^{th} column is the i^{th} eigenvector of RHS. $\mathbf{B}$ is a diagonal matrix whose entries are the eigenvalues of RHS. $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices

with left- and right-singular vectors for the corresponding singular values, respectively.

4.2.6 Ensemble Transform Kalman Filter (ETKF)

Bishop developed an EnKF that is based on the square-root filtering algorithm, of the backward-multiplication type [9]. It is computationally advantageous when $N_m < N_{kx}$. Using the same notations as defined previously, the ETKF is performed in the following procedure. In Eq. 4.33, eigendecomposition of RHS is performed.

$$\mathbf{ZBZ}^{'} = \mathbf{I} + (\mathbf{H\hat{D}})^{'}\tilde{\mathbf{R}}^{-1}\mathbf{H\hat{D}} \qquad \in \mathbb{R}^{N_m \times N_m}, \tag{4.33}$$

$$\mathbf{K}_m = \mathbf{\hat{D}ZB}^{-1}\mathbf{Z}^{'}(\mathbf{H\hat{D}})^{'}\tilde{\mathbf{R}}^{-1} \qquad \in \mathbb{R}^{N_y \times N_m}, \tag{4.34}$$

$$\underline{\mathbf{X}} = \underline{\hat{\mathbf{X}}} + \mathbf{K}_m\left(\mathbf{Y} - \underline{\hat{\mathbf{Y}}}\right) \qquad \in \mathbb{R}^{N_{kx} \times N_m}, \tag{4.35}$$

$$\mathbf{D} = \mathbf{\hat{D}ZB}^{-1/2}\mathbf{Z}^{'} \qquad \in \mathbb{R}^{N_{kx} \times N_m}, \tag{4.36}$$

$$\mathbf{X} = \underline{\mathbf{X}} + \mathbf{D} \qquad \in \mathbb{R}^{N_{kx} \times N_m}, \tag{4.37}$$

where $\mathbf{Z}$ is an orthogonal matrix whose ith column is the ith eigenvector of RHS. $\mathbf{B}$ is a diagonal matrix whose entries are the eigenvalues of RHS.

4.2.7 Particle Filter (PF)

Particle filter or Sequential Monte Carlo method was introduced by Kitagawa [42] and Gordon *et al.*[25] in the same year. Using the same notations as defined previously, PF is performed in the following procedure. Several types of resampling techniques have been developed intended to prevent degeneration of particles. Douc compared several resampling techniques for PF [15]. A low variance resampling technique [67] is used for the following procedure since it is computationally simple ($\mathcal{O}(N_m)$ for sampling N_m particles) and its superior performance was reported.
For $i = 1, \ldots, N_m$, the i^{th} particle weight or likelihood $w^{(i)}$ is calculated by Eq. 4.38, then it is normalized to $\tilde{w}^{(i)}$ by Eq. 4.39.

$$w^{(i)} = (2\pi)^{-N_y/2}\,|\mathbf{R}|^{-1/2}\exp\left[-\frac{1}{2}\left(\mathbf{Y}^{(i)} - \hat{\mathbf{Y}}^{(i)}\right)^{'}\mathbf{R}^{-1}\left(\mathbf{Y}^{(i)} - \hat{\mathbf{Y}}^{(i)}\right)\right] \qquad \in \mathbb{R}, \tag{4.38}$$

$$\tilde{w}^{(i)} = \frac{w^{(i)}}{\sum_{j=1}^{N_m} w^{(j)}} \qquad \in \mathbb{R}, \tag{4.39}$$

Resample particle $\hat{\mathbf{x}}^{(i)} \in \hat{\mathbf{X}}$ with the probability $\tilde{w}^{(i)}$, with replacement, to obtain $\mathbf{x}^{(i)} \in \mathbf{X}$.

4.2.8 EnSRF for Univariate Time Series Model

In the following chapters, we illustrate the filtering procedure of the EnSRF for electricity load forecasting, which was specially modified for fast processing of a univariate time series. Although the EnKF itself is an algorithm, in the following sections, we will use the term EnKF in a broader context to refer to the use of SSMs with this EnSRF.

$$\mathbf{m} := \hat{\mathbf{D}}\hat{\mathbf{D}}'\mathbf{H}' \qquad \in \mathbb{R}^{N_m \times 1}\,, \tag{4.40}$$

$$u := \sqrt{\mathbf{H}\mathbf{m} + \tilde{\mathbf{R}}} \qquad \in \mathbb{R}\,, \tag{4.41}$$

$$v := \sqrt{\tilde{\mathbf{R}}} \qquad \in \mathbb{R}\,, \tag{4.42}$$

$$\mathbf{k}_m = \frac{\mathbf{m}}{u^2} \qquad \in \mathbb{R}^{N_m \times 1}\,, \tag{4.43}$$

$$\mathbf{k}_d = \frac{\mathbf{m}}{u(u+v)} \qquad \in \mathbb{R}^{N_m \times 1}\,, \tag{4.44}$$

$$\underline{\mathbf{X}} = \underline{\hat{\mathbf{X}}} - \mathbf{k}_m(\hat{\mathbf{Y}} - \mathbf{Y}) \qquad \in \mathbb{R}^{N_{kx} \times N_m}\,, \tag{4.45}$$

$$\mathbf{D} = \hat{\mathbf{D}} - \mathbf{k}_d\mathbf{H}\hat{\mathbf{D}} \qquad \in \mathbb{R}^{N_{kx} \times N_m}\,, \tag{4.46}$$

$$\mathbf{X} = \underline{\mathbf{X}} + \mathbf{D} \qquad \in \mathbb{R}^{N_{kx} \times N_m}\,. \tag{4.47}$$

4.3 Estimating Variance of Noise

For the electricity load model that is a univariate time series model, our procedure for estimating the variances of the observation noise and the system noises was as follows. First, we assumed the observation noise was $\{W_t\} \sim$ i.i.d. $\mathcal{N}(0,\ R)$, where R is the variance of the observation noise, and we assumed the i^{th} system noise was $\{v_t^{(i)}\} \sim$ i.i.d. $\mathcal{N}(0,\ R/L_i)$, where the L_i is the trade-off parameter. A model with this setup is called a linear Gaussian state-space model. The variances of the system noises were initialized to approximately zero; this corresponds to setting the trade-off parameters large enough that the components are stable. Once these variances were fixed, the variance R of the observation noise $\{W_t\}$ was estimated; it was based on the maximum-likelihood estimation whose likelihood function is shown in Eq. 4.48. The estimate $\hat{R}$ was obtained from a grid search with Eq. 4.49 [43, Eq. 6.45]. After that, we slightly adjusted the variance $\hat{R}/L_i$ of the system noise $\{v_t^{(i)}\}$ in order to obtain a more accurate forecast over an evaluation term. These equations are as follows:

$$\mathcal{L}_t(R) := \frac{1}{N_m} \sum_{i=1}^{N_m} \frac{1}{\sqrt{2\pi R}} \exp\left(-\frac{(y_{i,t|t-1} - y_t)^2}{2R} \right), \tag{4.48}$$

$$\hat{R} = \arg\max_R \sum_{t=1}^{N_t} \log \mathcal{L}_t(R), \tag{4.49}$$

where y_t is the load, $y_{i,t|t-1}$ is the one-step-ahead load predicted by the i^{th} ensemble member, N_m is the number of ensemble members, and N_t is the number of time steps predicted.

4.4 Constraints for Component

There are many occasions where constraints (upper and lower limits) for components are required, especially when models are designed based on physics. In order to obtain the constrained components, a sigmoid function (Fig. 4.1) is commonly used [35]. A definition of the function is given by Eq. 4.50. It maps $\eta \in \mathbb{R} : -\infty < \eta < +\infty$ onto $\xi \in \mathbb{R} : 0 < \xi < 1$. In the recursive calculation of the EnKF, provided that the constraint of a coefficient is $0 < \xi < 1$, we use a state variable η whose relationship with ξ is described as Eq. 4.51.

$$\xi = S(\eta) := \frac{1}{1+\exp(-\eta)} , \tag{4.50}$$

$$\eta = S^{-1}(S(\eta)) = \log \frac{S(\eta)}{1-S(\eta)} = \log \frac{\xi}{1-\xi} . \tag{4.51}$$

We modified the above formula to a more generic form with the constraint $a < \xi < b$ ($a, b \in \mathbb{R}$), where $\eta : -\infty < \eta < +\infty$ can be mapped onto ξ by applying one of the following three cases:

(i) $a < \xi < b$

Define a map function $T : \eta \mapsto \xi : a < \xi < b$ by Eq. 4.52, then a state variable η can be obtained by Eq. 4.53.

$$\xi = T(\eta) := (b-a)S(\eta) + a = \frac{b + a\exp(-\eta)}{1+\exp(-\eta)} , \tag{4.52}$$

$$\eta = T^{-1}(T(\eta)) = \log \frac{T(\eta)-a}{b-T(\eta)} = \log \frac{\xi - a}{b-\xi} . \tag{4.53}$$

(ii) $0 < \xi < \infty$

Define a map function $U : \eta \mapsto \xi : 0 < \xi < \infty$ by Eq. 4.54, then a state variable η can be obtained by Eq. 4.55.

$$\xi = U(\eta) := \exp(\eta) , \tag{4.54}$$

$$\eta = \log(\xi) . \tag{4.55}$$

(iii) $-\infty < \xi < 0$

Define a map function $U' : \eta \mapsto \xi : -\infty < \xi < 0$ by Eq. 4.56, then a state variable η can

be obtained by Eq. 4.57.

$$\xi = U'(\eta) := -\exp(\eta)\,, \tag{4.56}$$

$$\eta = \log(-\xi)\,. \tag{4.57}$$

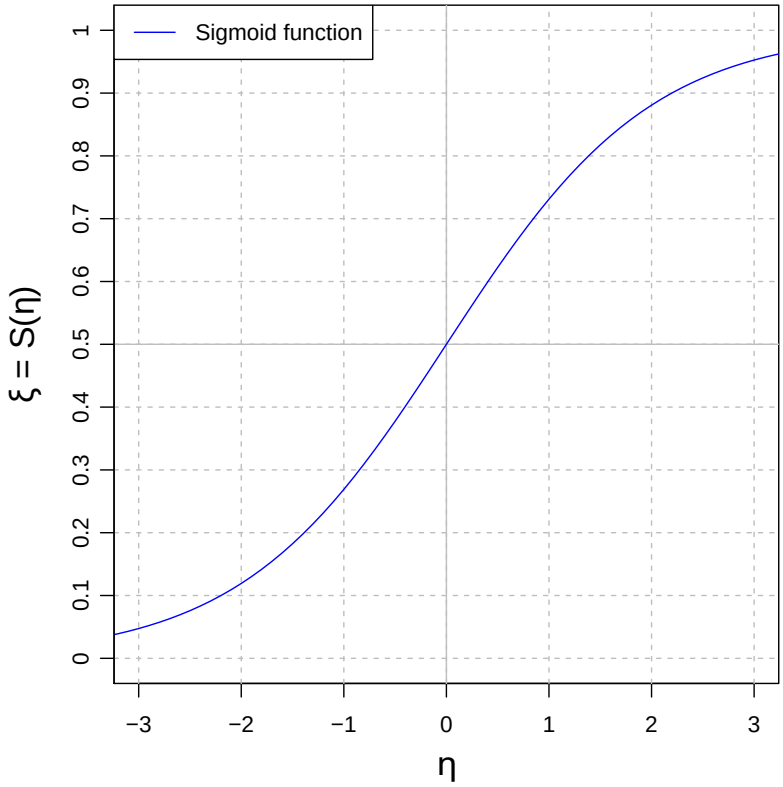

Fig. 4.1 Sigmoid function

4.5 Initial Ensembles

In order to capture unexpected structural changes of the load, initial ensemble members representing an initial distribution of a component were generated in such a way that the dispersion of the distribution was as wide as possible.

Based on the procedures explained in the preceding subsection, some components may have their upper and/or lower limits. As an initial distribution of such a constrained component, either the uniform distribution or the truncated normal distribution is supposed to be most suitable among others. In this study, initial ensembles for a constrained component

were generated so that the distribution of the component becomes a uniform distribution.

4.6 Simulation

To better understand the basic properties of structural time series modeling (e.g., SSMs), a numerical simulation using true models is conducted in this section.

The first step of the simulation is to generate a time series for each latent component using a true model. The true model should be as simple as possible without losing intrinsic features of the latent phenomena. Then, the simulation data, which represent electricity load observations in our case, are obtained by aggregating those components. The next step is to propose models for the latent components and check if those models for the load and the components properly converged to the simulated true time series.

The procedure stated above is sometimes referred to as the twin experiment in the data-assimilation context [46]. There are two types of twin experiment: one is the identical-twin experiment which uses the exact same model for both the true model and the proposed; and the other is the fraternal-twin experiment which does not use the exact same model. For flexible modeling intended for electricity load forecasting, we adopted the latter one; that is, the proposed models are different from the true models.

4.6.1 True Models

The observation Y_t is decomposed into the five latent components: $TREND_t$, $INWK_t$, $INDAY_t$, AR_t, and an irregular component which is observation noise $W_t \sim \mathcal{N}(0, 2^2)$, where $t = 1, 2, \cdots, 24 \times 7 \times 4$ [h]. The latent components are explained in detail in the following paragraphs.

$$Y_t = TREND_t + INWK_t + INDAY_t + AR_t + W_t \,. \tag{4.58}$$

True Trend Component Model

The true trend component is a straight line with a slope which increases at the rate of 3 [GW/week] and the intercept is 35 [GW], mimicking an increasing trend of load during the spring season in the metropolitan area in Japan.

$$TREND_t = 35 + 3t/(24 \times 7) \,. \tag{4.59}$$

True Intra-Weekly Component Model

The true intra-weekly component is a sinusoidal curve with the amplitude of 4 [GW] and the time period of a week (i.e., 24×7 [h]).

$$INWK_t = 4\sin(2\pi t/(24 \times 7)) \ . \tag{4.60}$$

True Intra-Daily Component Model

The true intra-daily component is also a sinusoidal curve with the amplitude of 4 [GW] and the time period of a day (i.e., 24 [h]).

$$INDAY_t = 4\sin(2\pi t/24) \ . \tag{4.61}$$

True Autoregressive Component Model

The true autoregressive component is the second-order autoregressive model, AR(2), whose the first and the second coefficients are ϕ_1 and ϕ_2 respectively. We used $\phi_2 = -1.0$ which causes a stationary oscillation, and $\theta = \pi/6$ which is the intermediate parameter with the time period of 12 [h](= $2\pi/\theta$). The relationship expressed in Eq. 4.63 will be explained in the following section.

$$AR_t = \phi_1 \, AR_{t-1} + \phi_2 \, AR_{t-2} \ , \tag{4.62}$$

$$\phi_1 = 2\sqrt{-\phi_2} \, \cos(\theta) \ . \tag{4.63}$$

The true models which represent the electricity load structure, explained in the preceding subsection, are drawn in Fig. 4.2.

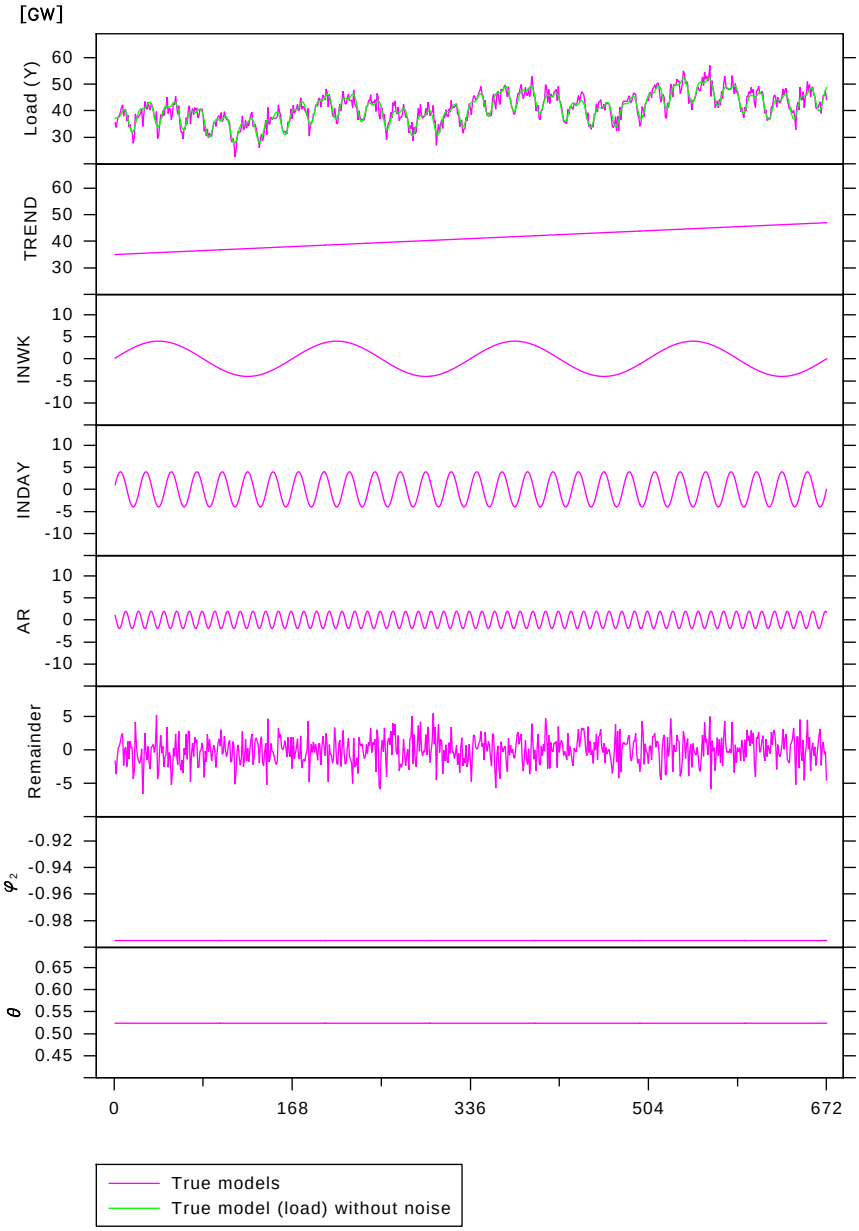

Fig. 4.2 Electricity load structure model (true models). Simulated load contains true component models; that is, the load is the summation of those models. On the top panel, load (magenta) and load without observation noise (green) are drawn. From the second panel, $TREND_t$, $INWK_t$, $INDAY_t$ and AR_t are drawn, followed by *Remainder* (or innovation) which is observation noise, then AR_t paramters ϕ_2 and θ.

4.6.2 Proposed Models (Auto-projective Component Models)

System Noise

Most of the components in the following sections have a term for system noise, which is denoted as $v_t^{(\cdot)}$, where the superscript is the identification number of a particular component or parameter. The system noise increases the uncertainty of prediction of each component.

Trend Component Model

The local trend model was used for modeling a trend of the simulated electricity load since, as in the Holt-Winters method, it ensures that the slopes within a trend are very similar.

$$TREND_t = 2\,TREND_{t-1} - TREND_{t-2} + v_t^{(1)}. \tag{4.64}$$

Intra-Weekly Component Model

The seasonal component model of Kitagawa [44] was used for the intra-weekly and intra-daily components. It is referred to as the normalized seasonal model [31, Ch. 8] since it requires that the average of the seasonal components is equal to zero:

$$\Sigma_{i=0}^{167}\, INWK_{t-i} = v_t^{(2)}. \tag{4.65}$$

Intra-Daily Component Model

The intra-daily component is defined in the same way as $INWK_{t-i}$.

$$\Sigma_{i=0}^{23}\, INDAY_{t-i} = v_t^{(3)}\ . \tag{4.66}$$

Autoregressive Component Model

The autoregressive component, AR(2), given by Eqs. 4.67 and 4.68, represents a periodic variation within 24 hours. With restricted coefficients of AR(2), a trigonometric curve (i.e., sine or cosine curve) can be expressed in the recursive equations. Another role of AR(2) is to increase the accuracy of very short-term (e.g., 1–3 hours) forecasts, which is much more common usage. The coefficients of AR(2), $\phi_{1,t}$ and $\phi_{2,t}$, must satisfy the stationary condition. In addition, an oscillatory condition is required to model the small fluctuations that often occur when an intra-daily or intra-weekly component is apparently over-fitted. We used Doménech's method [13] to restrict the parameters of AR(2), and we formulated the

conditions ($\phi_{1,t}^2 + 4\phi_{2,t} < 0 \wedge \phi_{2,t} > -1$) in Eqs. 4.68–4.70, using the intermediate parameter $\theta_t : 0 < \theta_t < \pi$. These equations are as follows:

$$AR_t = \phi_{1,t}\, AR_{t-1} + \phi_{2,t}\, AR_{t-2} + v_t^{(4)} \,, \tag{4.67}$$

$$\phi_{1,t} = 2\sqrt{-\phi_{2,t}}\cos\theta_t \,, \tag{4.68}$$

$$\log\left(\frac{1+\phi_{2,t}}{-\phi_{2,t}}\right) = \log\left(\frac{1+\phi_{2,t-1}}{-\phi_{2,t-1}}\right) + v_t^{(5)} \,, \tag{4.69}$$

$$\log\left(\frac{\theta_t}{\pi-\theta_t}\right) = \log\left(\frac{\theta_{t-1}}{\pi-\theta_{t-1}}\right) + v_t^{(6)} \,. \tag{4.70}$$

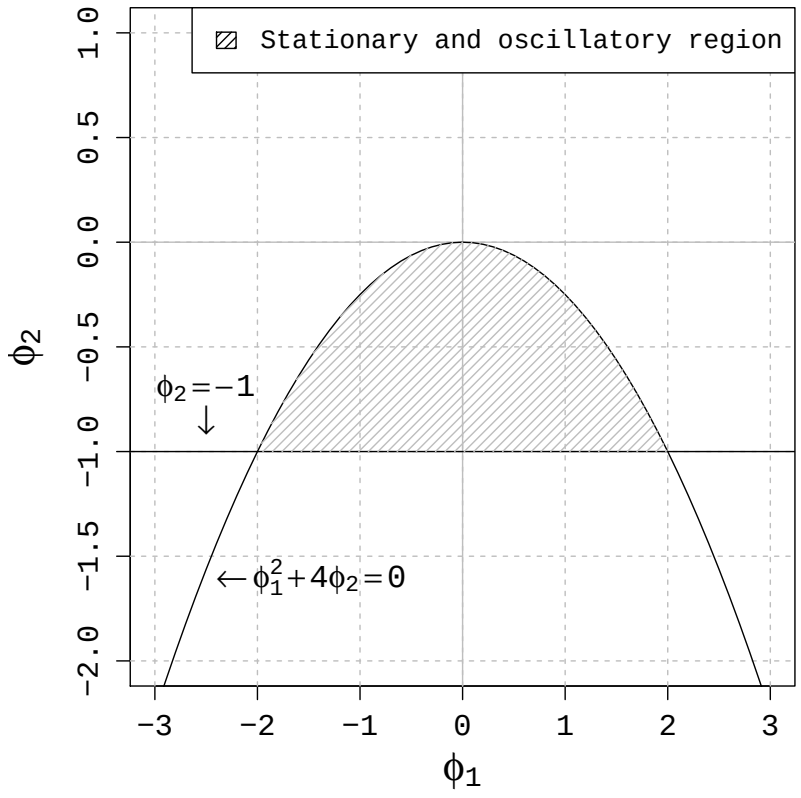

Fig. 4.3 Stationary and oscillatory parameter space of AR(2) process

4.6.3 Data Assimilation Result of Proposed Models

Figure 4.4 shows the results of data assimilation with EnSRF. Only $TREND_t$ successfully converged to the true trajectories immediately after one week (168 hours). All the components were stabilized; however, small fluctuations were observed in both $INWK_t$ and $INDAY_t$. This can be explained by smoothness, which is not assumed in these modelings. AR component as well as its two parameters failed to converge to the true trajectory and

values.

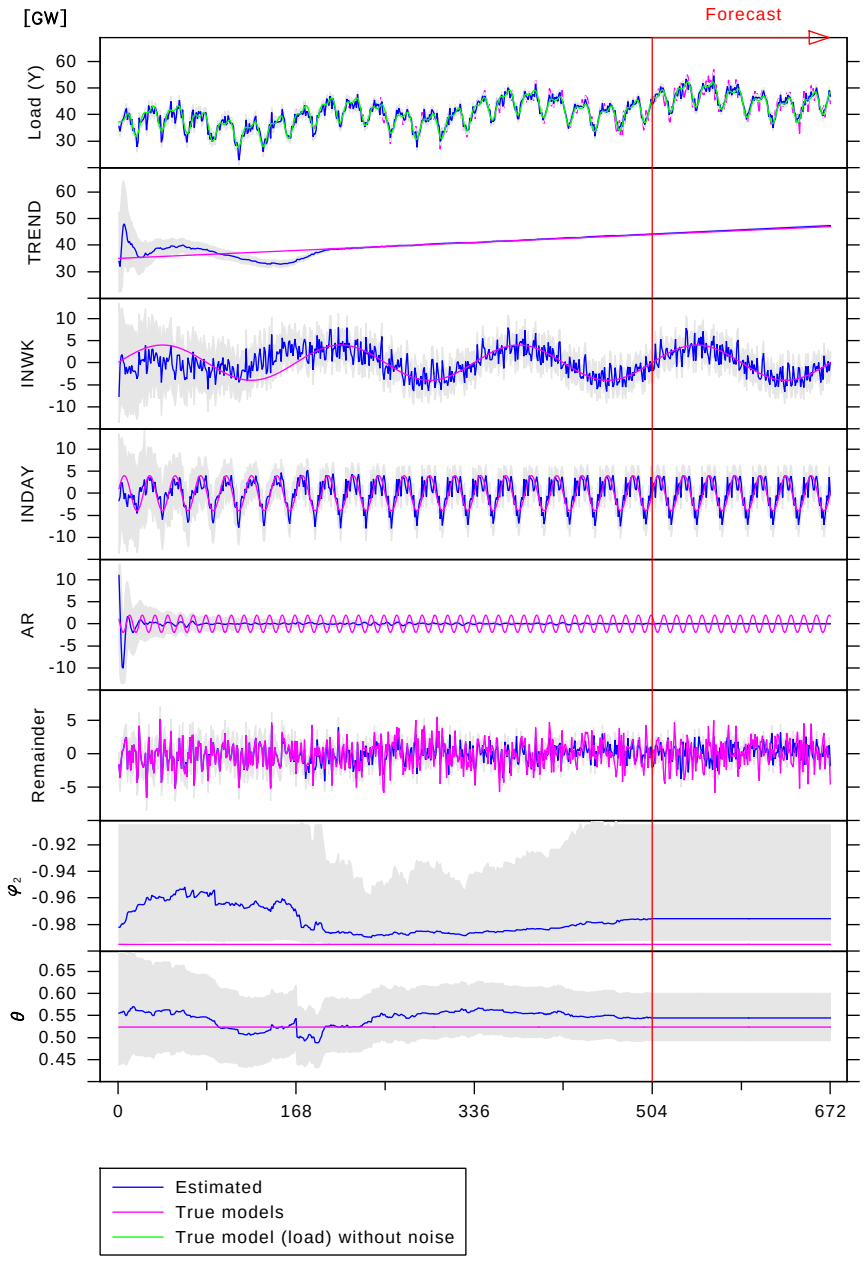

Fig. 4.4 Twin experiment. Simulated load was decomposed by SSMs with EnSRF (ensemble size is 300). The magenta line and the blue line in each panel denote a true component/parameter and an estimated one, respectively. The green line in the top panel shows the true (simulated) load without observation noise. From the third week (504 hours), one-week forecasts were performed.

4.6.4 Enhancement of $INWK_t$ and $INDAY_t$

The two-sided exponentially weighted smoother was used for both seasonal components, $INWK_t$ and $INDAY_t$, to subdue small fluctuations.

Smoothing for $INWK_t$

In order to increase the quality of the data assimilation, we applied the two-sided exponentially weighted smoother function $s(\cdot)$ with argument $\{x_t \mid t = 0, 1, \cdots, N; x_{-j} = x_{N-j}, x_{N+j} = x_j\}$, which is a periodic time series. It updates each state at each step rather than every 168 steps. The smoothed intra-weekly component is

$$\{INWKS_{t-i}\}_{i=0}^{167} = s\left(\{INWK_{t-i}\}_{i=0}^{167}\right) , \tag{4.71}$$

$$s(x_t) = (\Sigma_{j=-p}^{p} \lambda^{|j|} x_{t+j})/\eta , \tag{4.72}$$

$$\eta = \Sigma_{j=-p}^{p} \lambda^{|j|} , \tag{4.73}$$

where the notation $\{a_i\}_{i=0}^{N}$ signifies $\{a_0, a_1, \cdots, a_N\}$. $N = 167$ [h] is the time length, and $p = 6$ [h] is a one-side time length for smoothing weights. η is the normalization parameter that ensures the total smoothing weight is equal to 1.0.

Smoothing for $INDAY_t$

The intra-daily component model $INDAY_t$ is also smoothed by the following equation:

$$\{INDAYS_{t-i}\}_{i=0}^{23} = s\left(\{INDAY_{t-i}\}_{i=0}^{23}\right) . \tag{4.74}$$

4.6.5 Results (Graphics) of Data Assimilation by Ensemble Approaches

In the following subsections, we will show the results of data assimilation performed by the following ensemble approaches:

- Ensemble Square-Root Filter
- Ensemble Kalman Filter with Square-Root Algorithm
- Ensemble Transform Kalman Filter
- Ensemble Adjustment Kalman Filter
- Ensemble Kalman Filter with Perturbed Observations
- Particle Filter

Ensemble Square-Root Filter (EnSRF)

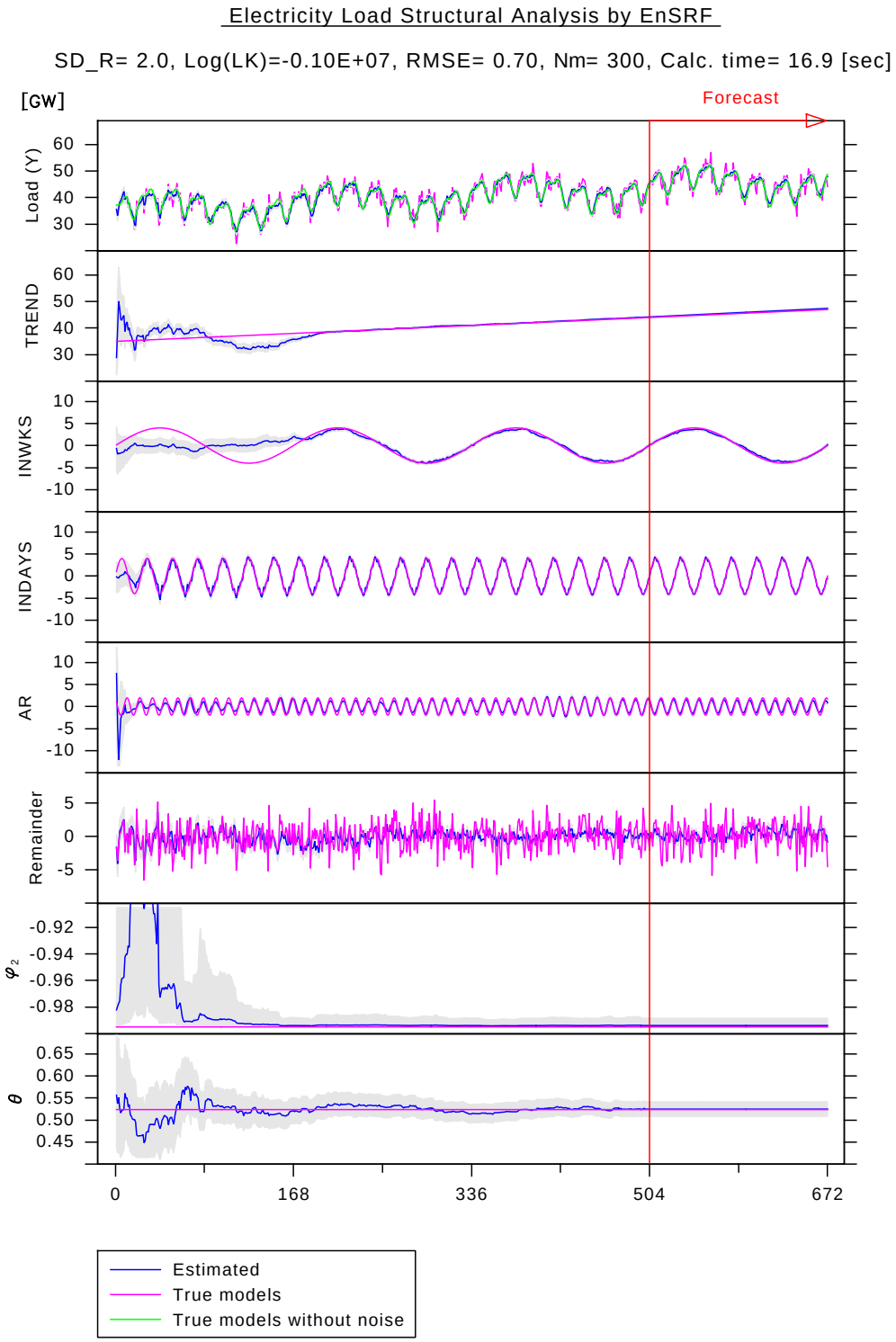

Fig. 4.5 Twin experiment with smoothing of *INWK* and *INDAY*. Simulated load was decomposed by SSMs with EnSRF. ϕ_2 and θ are parameters of the AR component. The magenta line and the blue line in each panel denote a true component/parameter and an estimated one, respectively. The green line in the top panel shows a true (simulated) load with observation noise.

Ensemble Kalman Filter with Square-Root Algorithm (EnKF/SR)

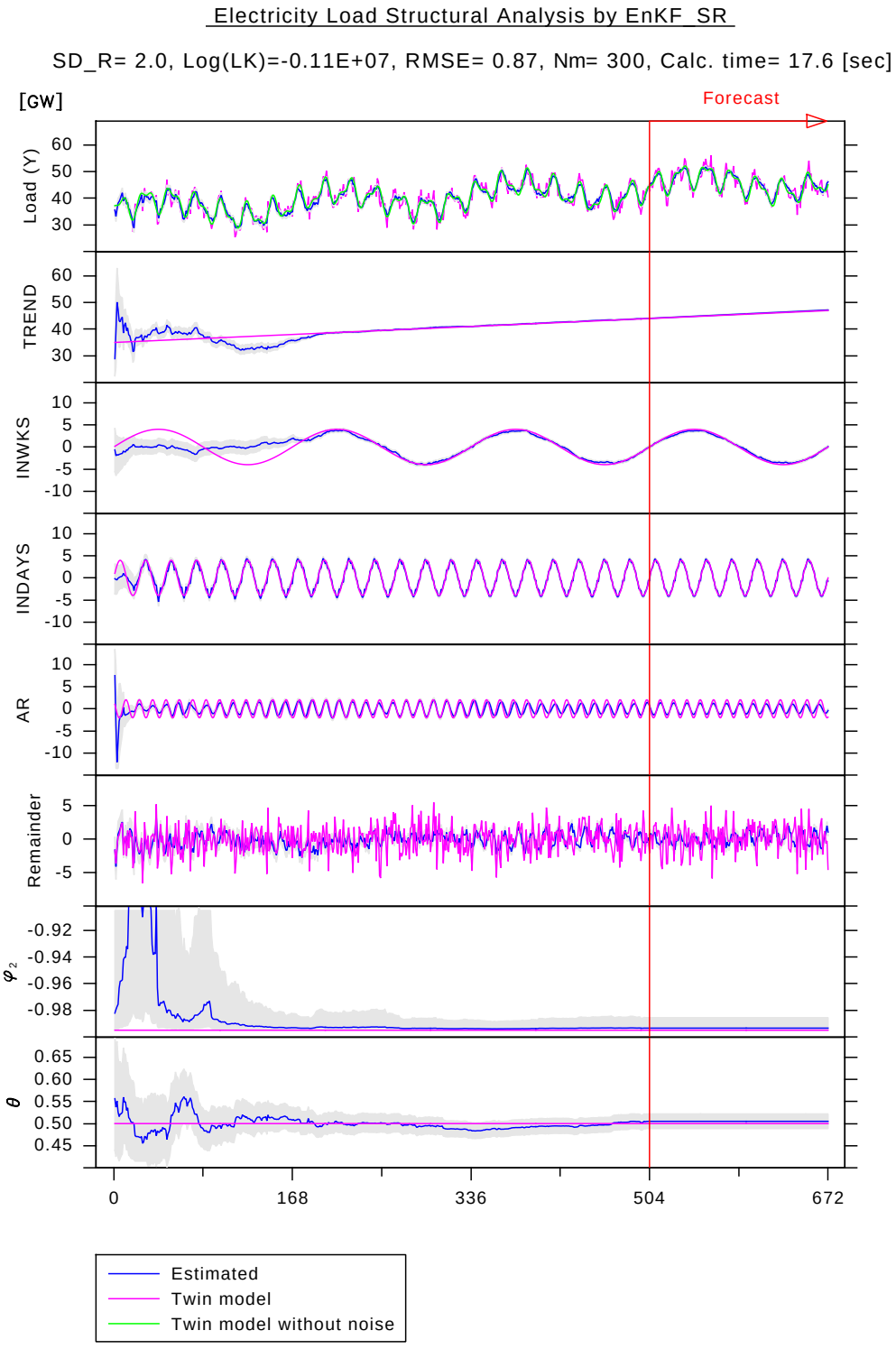

Fig. 4.6 Ensemble Kalman Filter with Square-Root Algorithm (EnKF/SR)

Ensemble Transform Kalman Filter (ETKF)

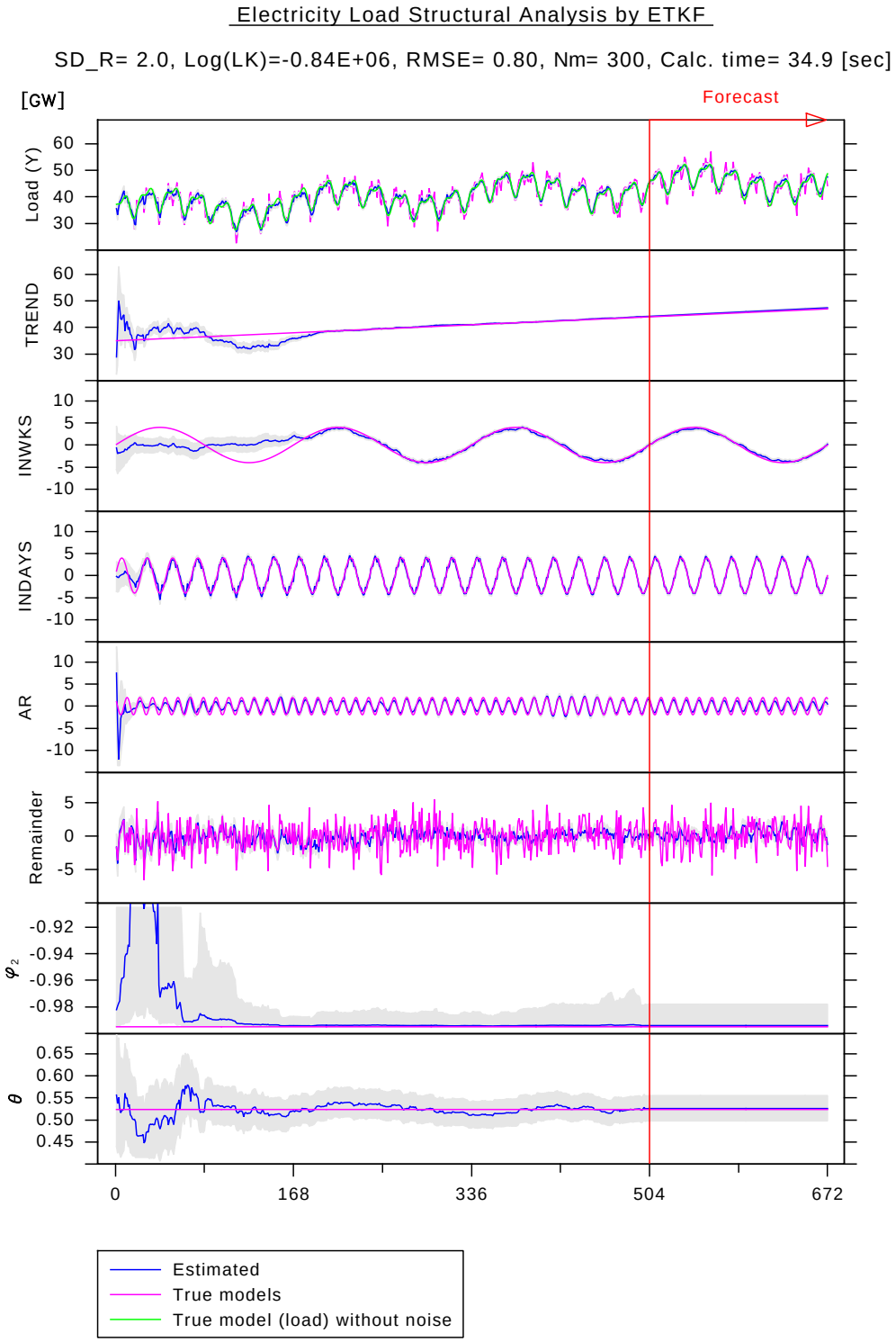

Fig. 4.7 Ensemble Transform Kalman Filter (ETKF)

Ensemble Adjustment Kalman Filter (EAKF)

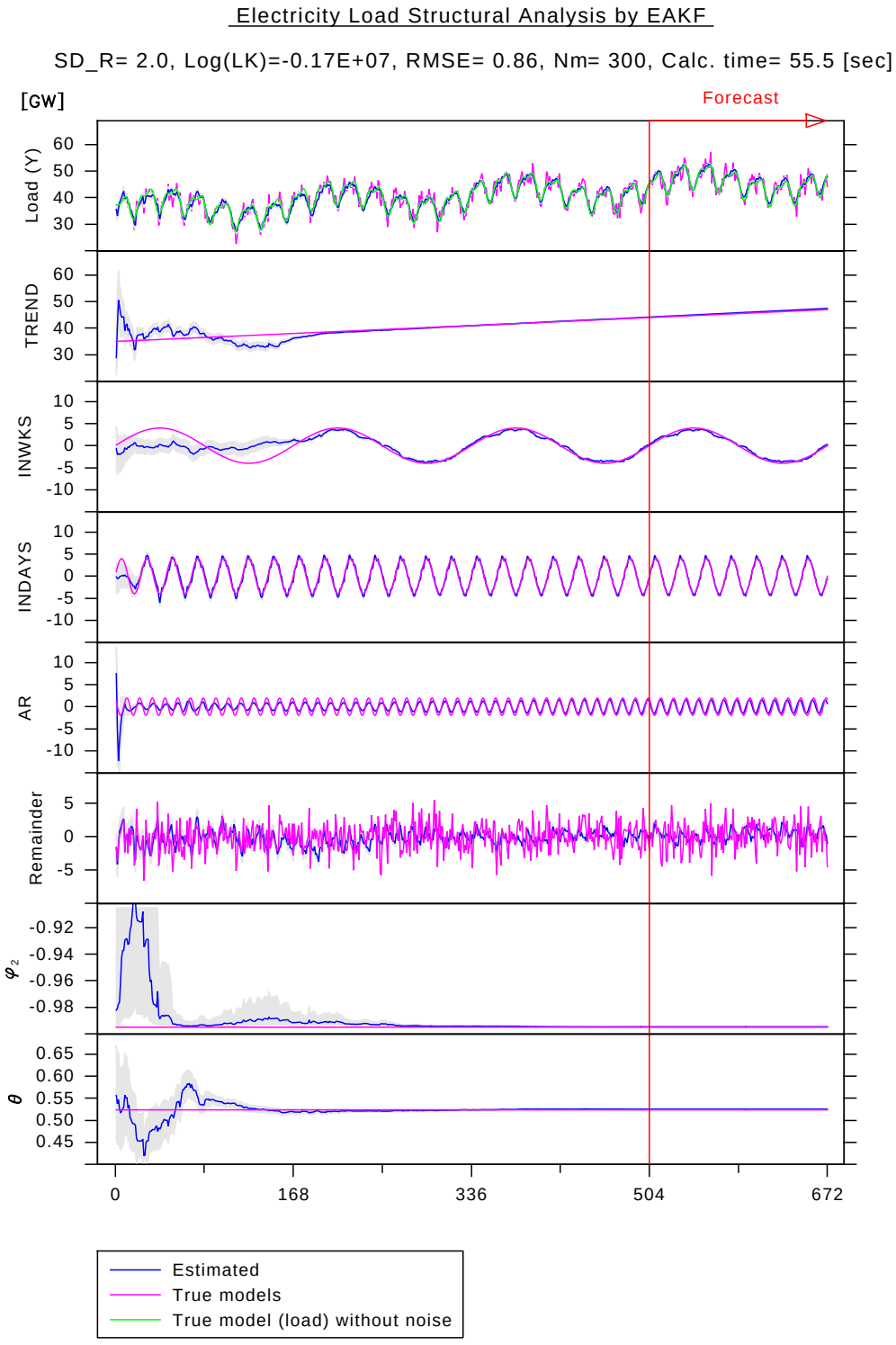

Fig. 4.8 Ensemble Adjustment Kalman Filter (EAKF)

Ensemble Kalman Filter with Perturbed Observations (EnKF/PO)

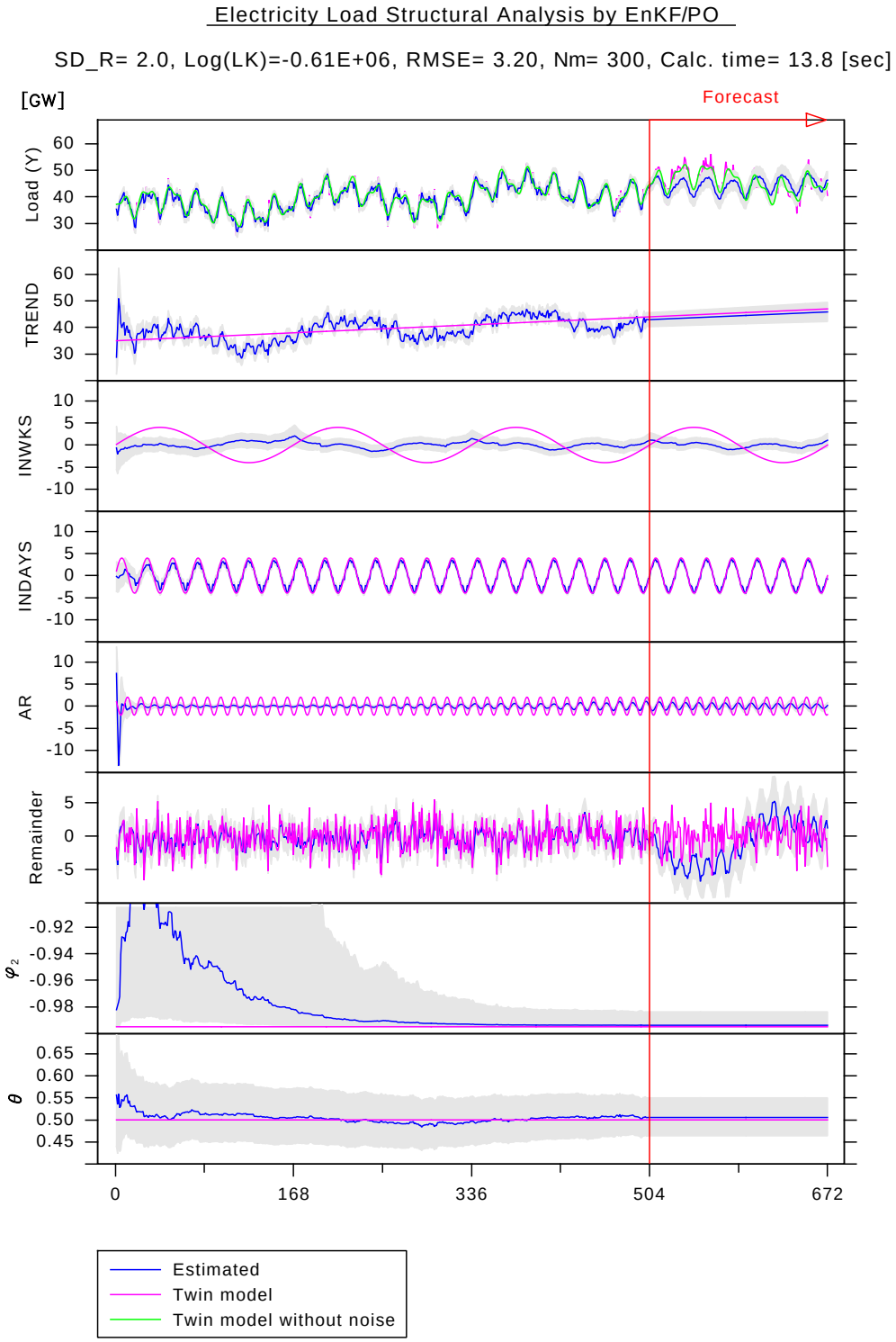

Fig. 4.9 Ensemble Kalman Filter with Perturbed Observations (EnKF/PO)

Particle Filter (PF)

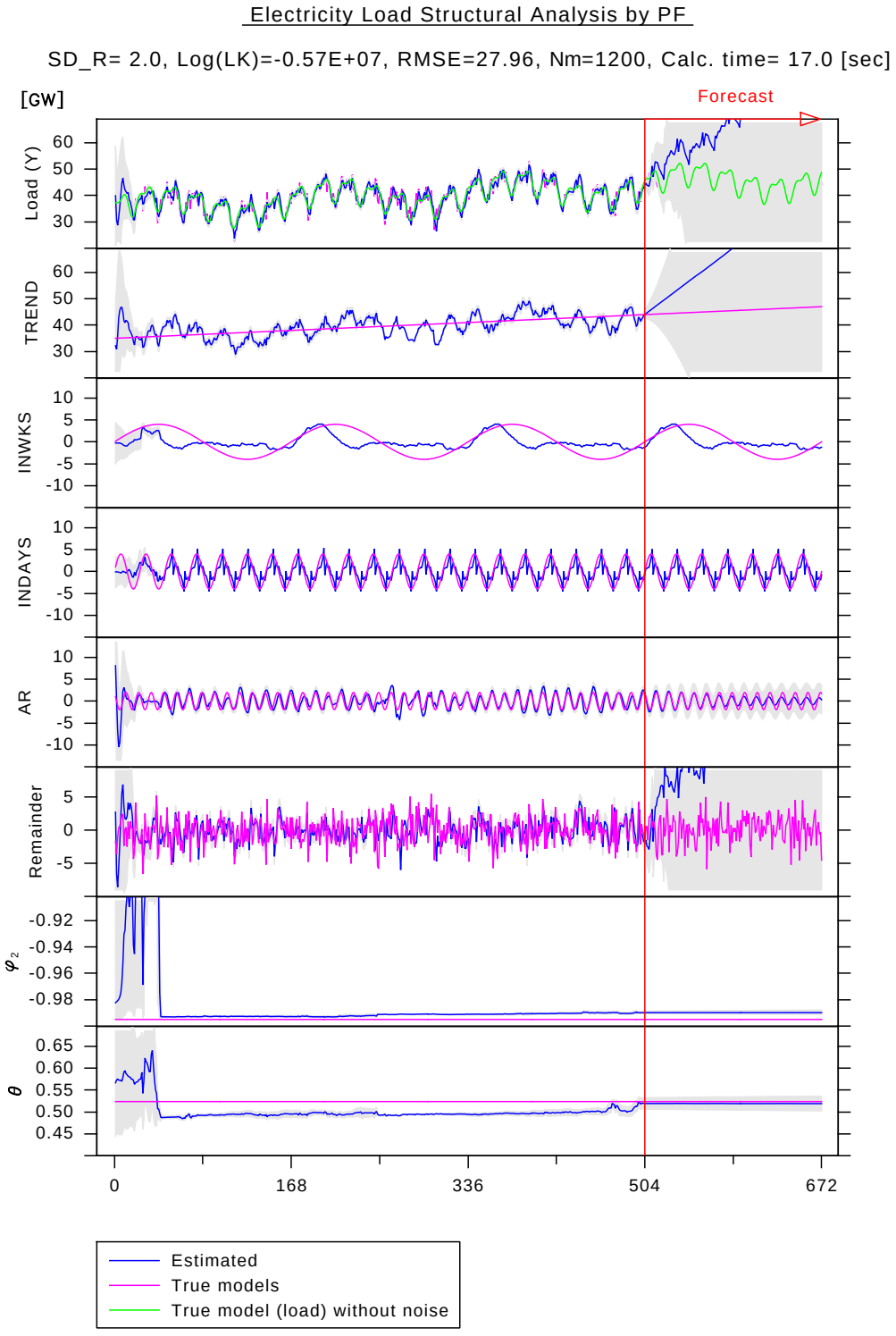

Fig. 4.10 Particle Filter (PF)

4.6.6 Summary of Implemented Ensemble Approaches

Those filters below were implemented solely by the author (see Appendix A.1 for Fortran sample code); therefore, it is not necessarily the case that performances of filters implemented by others are exactly the same as the below mentioned.

Ensemble Square-Root Filter (Fig. 4.5) proves to be the most promising candidate for data assimilation of electricity load forecasting. Immediately after the first week (168 h), all components successfully converge to the true trajectories and the calculation speed was excellent (16.9 s).

Ensemble Kalman Filter with Square-Root Algorithm (Fig. 4.6) produced very similar results to EnSRF and the calculation speed was also excellent (17.6 s).

Ensemble Transform Kalman Filter (Fig. 4.7) produced very similar results to EnSRF and the calculation speed was moderate (34.9 s).

Ensemble Adjustment Kalman Filter (Fig. 4.8) produced similar results to EnSRF; however, small fluctuations appeared in $INWKS_t$. The calculation was very time-consuming (55.5 s).

Ensemble Kalman Filter with Perturbed Observations (Fig. 4.9) failed in data assimilation of $TREND_t$ and $INWKS_t$. In addition, AR_t along with its parameters $\phi_{2,t}$ and θ_t converged very slowly. The calculation speed was the fastest of all (13.8 s).

Particle Filter (Fig. 4.10) also failed in data assimilation of most of the components. To avoid a well-known problem inherent to particle filter, the so-called degeneration of the particles, we used ensemble members three times as large as used for other ensemble approaches. The calculation speed was (17.0 s) though this was not a like-for-like comparison to others since the ensemble size was different. To properly assimilate the observations into components, countermeasures such as random particle addition should be implemented in this filter.

From the reasons mentioned above, we selected EnSRF as the filtering method for electricity load forecasting.

4.7 Model Performance Metrics

4.7.1 Measurement of Accuracy

The forecasting accuracy was measured in terms of the mean-absolute-error (MAE), the mean-absolute-percentage-error ($MAPE$) and the root-mean-squared-error ($RMSE$). We used the standard deviation ($SD_{\{PE\}}$) of the percentage-error (PE_t) to express variability of the accuracy. $RMSE$ tend to be similar to MAE; however, $RMSE$ is more focused on large errors. For comparison with a physics-based benchmark model that has the severe bias problem, we used the mean-bias-error (MBE).

In context of load forecasting, $MAPE$ is the most widely used accuracy measure, since it is easy to understand, and an imbalance charge[1] is assessed based on the absolute percentage error (APE). Furthermore, the $MAPE$ is scale independent, which enables us to compare accuracy for different load levels.

Those accuracy measures are defined by the following equations:

$$e_t := \hat{y}_t - y_t \qquad \text{[GW]}\,, \tag{4.75}$$

$$MBE := \frac{1}{N}\sum_{t=1}^{N} e_t \qquad \text{[GW]}\,, \tag{4.76}$$

$$MAE := \frac{1}{N}\sum_{t=1}^{N} |e_t| \qquad \text{[GW]}\,, \tag{4.77}$$

$$RMSE := \sqrt{\frac{1}{N}\sum_{t=1}^{N} e_t^2} \qquad \text{[GW]}\,, \tag{4.78}$$

$$PE_t := \frac{e_t}{y_t} \times 100 \qquad [\%]\,, \tag{4.79}$$

$$MAPE := \frac{1}{N}\sum_{t=1}^{N} |PE_t| \qquad [\%]\,, \tag{4.80}$$

$$SD_{\{PE\}} := \sqrt{\frac{1}{N}\sum_{t=1}^{N} (PE_t - \overline{PE})^2} \qquad [\%]\,, \tag{4.81}$$

where e_t is the error at time t, y_t is the observed value, $\hat{y}_t$ is the estimate, and N is the number of time steps in an evaluation term.

[1] This occurs when the power producer and supplier (PPS) is unable to maintain a balance between supply and demand; for each collective 30-minute load that is unbalanced, the PPS must pay a penalty to the system operator.

4.7.2 Statistical Test for Significance of Accuracy

The Diebold-Mariano (DM) test is the most sophisticated statistical tool to compare the accuracy of each set of two forecasts. In practice, tests for equality of variance, such as the F-test, Levene's test, Bartlett's test, and the Brown-Forsythe test, are also used to compare accuracy. Compared to those tests, however, the DM test requires fewer assumptions about the forecasting errors; these tests are also known to have an inflation problem with the type-I error, and this was not reported for the DM test. The null hypothesis of the DM test is that the two forecasts have the same accuracy. Let us explain the test procedure.

First, let us denote two competing h time-step ahead forecasts from the present time t as $\hat{y}^{(1)}_{t+h|t}$ and $\hat{y}^{(2)}_{t+h|t}$, and the observation as y_{t+h} ($t = 1, 2, \cdots, N$). Then, the forecast errors from the two models can be expressed by the following equations:

$$e^{(1)}_{t+h|t} = \hat{y}^{(1)}_{t+h|t} - y_{t+h} \,, \tag{4.82}$$

$$e^{(2)}_{t+h|t} = \hat{y}^{(2)}_{t+h|t} - y_{t+h} \,. \tag{4.83}$$

Secondly, we introduce loss functions from which forecasting accuracy can be measured. The two types of loss function commonly used are:

$$\text{Absolute-error loss function:} \quad L(e) := |e| \,, \tag{4.84}$$

$$\text{Squared-error loss function:} \quad L(e) := e^2 \,, \tag{4.85}$$

The absolute-error loss function $L(e) := |e|$ was used in this study since the absolute value of the load-forecasting error is usually evaluated. The DM test is performed based on the following error loss differential:

$$d_t = L\left(e^{(1)}_{t+h|t}\right) - L\left(e^{(2)}_{t+h|t}\right) \,. \tag{4.86}$$

The null and alternative hypotheses for equal predictive accuracy are :

$$H_0 : E[d_t] = 0 \quad \forall t \,, \tag{4.87}$$

$$H_1 : E[d_t] \neq 0 \,. \tag{4.88}$$

Thirdly, as a variance of the error loss differential d_t, the spectral density at frequency 0 is

used:

$$f_d(0) := \frac{1}{2\pi}\left(\Sigma_{j=-\infty}^{\infty}\gamma_d(j)\right) , \tag{4.89}$$

where $\gamma_d(j)$ is the autocorrelation of the error loss differential at lag j. In real situations, however, the length of forecasts is limited. The consistent estimator of $f_d(0)$ is used instead.

$$\hat{f}_d(0) := \frac{1}{2\pi}\left\{\Sigma_{j=-(N-1)}^{N-1} I\left(\frac{j}{h-1}\right)\hat{\gamma}_d(j)\right\} , \tag{4.90}$$

$$\hat{\gamma}_d(j) := \frac{1}{N}\Sigma_{t=|j|+1}^{N}(d_t - \bar{d})(d_{t-|j|} - \bar{d}) , \tag{4.91}$$

$$I\left(\frac{j}{h-1}\right) := \begin{cases} 1 & \text{for } \left|\frac{j}{h-1}\right| \leq 1 \\ 0 & \text{otherwise} \end{cases} , \tag{4.92}$$

$$\bar{d} := \frac{1}{N}\Sigma_{t=1}^{N} d_t . \tag{4.93}$$

Finally, we obtain the DM statistic S:

$$S = \frac{\bar{d}}{\sqrt{\frac{2\pi\hat{f}_d(0)}{N}}} \sim \mathcal{N}(0,1) . \tag{4.94}$$

Chapter 5

Electricity Load Forecasting

5.1 Method

5.1.1 Electricity Load Model

The electricity load pattern as shown in Fig. 2.2 includes the auto-projective components as explained in the preceding chapter, and the load is certainly affected by weather conditions such as temperature; thus, it also includes weather-related exogenous components. As a result, our proposed model expressed by Eq. 5.1 contains those components and an irregular component which is observation noise:

$$\begin{aligned} Y_t = \; & TREND_t + INWKS_t + f(DAY_t, INDAYS_t) + AR_t \\ & + TP_t + RA_t + HM_t + WS_t + PV_t + W_t \,, \end{aligned} \tag{5.1}$$

where t is the elapsed time in hours, Y_t is the load that was illustrated with the thick line in Fig. 1.1 or Fig. 2.1, $TREND_t$ is the trend component, $INWKS_t$ is the smoothed intra-weekly (168-hour periodic) component, $INDAYS_t$ is the smoothed intra-daily (24-hour periodic) component, DAY_t is the effect of a day of the week (day-effect component), AR_t is the autoregressive component, TP_t is the temperature effect, RA_t is the solar radiation effect, HM_t is the humidity effect, WS_t is the wind effect, PV_t is the photovoltaic effect, W_t is the irregular component (observation noise), and $f(\cdot)$ is a function that gives the relationship between the smoothed intra-daily and the day effects. These components will be explained in detail in the following sections.

Trend Component Model

Several models have been developed to represent a trend or seasonal variation of the load [31]. Of these, the local trend model is one of the most commonly used; it is defined as $TREND_t = 2\,TREND_{t-1} - TREND_{t-2} + v_t^{(1)}$. This was the model we used in the preceding chapter; however, in a preliminary study, we found that when forecasting the load one week or two in advance, the following trend model (without a slope) produced better results:

$$TREND_t = TREND_{t-1} + v_t^{(1)}. \tag{5.2}$$

Intra-Weekly Component Model

The recursive equation given by Eq. 5.3 represents the intra-weekly or 168-hour periodic variation of the load; it is based on the load of industrial factories or offices that work on an intra-weekly schedule. This component also plays a role as a combined intercept for the exogenous components. To increase the accuracy of the forecast, we applied Friedman's variable-span smoother function [22], which is a smoother with a locally linear fitting. For reducing the error, this smoother performs better than the two-sided exponentially weighted smoother. The smoothed intra-weekly component is expressed in Eq. 5.4.

$$\Sigma_{i=0}^{167}\, INWK_{t-i} = v_t^{(2)}, \tag{5.3}$$

$$\{INWKS_{t-i}\}_{i=0}^{167} = s\left(\{INWK_{t-i}\}_{i=0}^{167}\right), \tag{5.4}$$

where $s(\cdot)$ is Friedman's variable-span smoother function. The notation $\{a_i\}_{i=0}^{N}$ signifies $\{a_0, a_1, \cdots, a_N\}$.

Intra-Daily and Day-Effect Component Models

In the load, a pattern can be seen that is due to intra-daily routines, such as scheduled factory operations. There are also daily gaps during a week, as we observed in Fig. 2.2. These variations are expressed in the combined formula given by Eq. 5.5. The recursive equations given by Eqs. 5.8 and 5.9 represent the intra-daily load pattern and the smoothed, respectively, denoted as $INDAY_t$ and $INDAYS_t$. Equations 5.6 and 5.7 represent the daily gap denoted as $DAY_{t,j(t)}$, which functions as a 24-hour constant coefficient for $INDAYS_t$. For modeling the load drops on holidays, we incorporated the empirical knowledge of TEPCO forecasters in Eq. 5.6; that is, the intra-daily load level on a holiday is usually close to the mean level of the nearest Saturday and Sunday. For parsimonious modeling, we sim-

ply summed the Saturday and Sunday levels and divided it by two; however, this could be replaced by a weighted mean if a new state variable were introduced for the weight. $\alpha_{t,j(t)}$ and MIN_t in Eqs. 5.10 and 5.11, respectively, were added to ensure the intra-daily load was positive so that it did not decrease during the night. These equations are as follows:

$$f(DAY_{t,j(t)}, INDAYS_t) = DAY_{t,j(t)} \times (INDAYS_{t-i} - \bar{\alpha}_{t,j(t)} MIN_t) \,, \tag{5.5}$$

$$DAY_{t,j(t)} = \begin{cases} (DAY_{t-1,0} + DAY_{t-1,6}) \;/\; 2 + v_t^{(3)} & \text{if } date_t \text{ is holiday} \\ DAY_{t-1,j(t)} + v_t^{(3)} & \text{otherwise} \,, \end{cases} \tag{5.6}$$

$$j(t) = \begin{cases} 0 & \text{if } date_t \text{ is Sunday} \\ 1 & \text{if } date_t \text{ is Monday} \\ \quad \vdots & \\ 6 & \text{if } date_t \text{ is Saturday} \,, \end{cases} \tag{5.7}$$

$$\Sigma_{i=0}^{23}\, INDAY_{t-i} = v_t^{(4)} \,, \tag{5.8}$$

$$\{INDAYS_{t-i}\}_{i=0}^{23} = s\left(\{INDAY_{t-i}\}_{i=0}^{23}\right) \,, \tag{5.9}$$

$$\alpha_{t,j(t)} = \alpha_{t-1,j(t)} + v_t^{(5)} \,, \tag{5.10}$$

$$MIN_t = \min\{INDAYS_{t-i} \,,\; i = 0, 1, \cdots, 23\} \,, \tag{5.11}$$

where $date_t$ is the calendar date of the elapsed time t. Both $DAY_{t,j(t)}$ and $\alpha_{t,j(t)}$ are imposed positive constraints (i.e., > 0).

Autoregressive Component Model

As explained in the preceding chapter, the autoregressive component model, AR(2) is given by Eqs. 5.12 and 5.13, with the conditions ($\phi_{1,t}^2 + 4\phi_{2,t} < 0 \;\wedge\; \phi_{2,t} > -1$) in Eqs. 5.14 and 5.15, using the intermediate parameter $\theta_t : 0 < \theta_t < \pi$.

$$AR_t = \phi_{1,t}\, AR_{t-1} + \phi_{2,t}\, AR_{t-2} + v_t^{(6)} \,, \tag{5.12}$$

$$\phi_{1,t} = 2\sqrt{-\phi_{2,t}} \cos\theta_t \,, \tag{5.13}$$

$$\log\left(\frac{1+\phi_{2,t}}{-\phi_{2,t}}\right) = \log\left(\frac{1+\phi_{2,t-1}}{-\phi_{2,t-1}}\right) + v_t^{(7)} \,, \tag{5.14}$$

$$\log\left(\frac{\theta_t}{\pi-\theta_t}\right) = \log\left(\frac{\theta_{t-1}}{\pi-\theta_{t-1}}\right) + v_t^{(8)} \,. \tag{5.15}$$

Temperature-Effect Component Model

Temperature is the most important exogenous variable, since it directly affects the use of electric air conditioners. The relationship between temperature and load can be observed in Fig. 5.1, which shows the daily maximum and minimum loads. People use coolers in summer and heaters in winter, and they usually do not use either of these in spring or autumn. Temperature affects the short-term (1–2 hours) load due to the automatic response or manual control of air conditioners, and it affects the long-term (1–3 days) load due to the cumulative cooling/heating effect on buildings. In general, people turn on air conditioners when the temperature rises above a particular temperature T_t (e.g., 23 °C) in summer, and they turn them off if it drops below this. On the other hand, in winter, people turn on heaters when the temperature drops below a particular temperature T'_t (e.g., 15 °C), and they turn them off if it rises above this. We will use the common notation ST_t for both the cooling switch-off temperature T_t and the heating one T'_t.

The component model for temperature effect TP_t is given by the following equations:

$$TP_t = \gamma_t \Delta tp_{2,t}^{\delta_t} + \gamma'_t \Delta tp_{48,t} \,, \tag{5.16}$$

$$\Delta tp_{2,t} = \begin{cases} [tp_{2,t} - ST_t]_+ & \text{if } date_t \in \{\text{May 1–Oct. 15}\} \\ [ST_t - tp_{2,t}]_+ & \text{otherwise} \,, \end{cases} \tag{5.17}$$

$$\Delta tp_{48,t} = \begin{cases} tp_{48,t} - ST_t & \text{if } date_t \in \{\text{May 1–Oct. 15}\} \\ ST_t - tp_{48,t} & \text{otherwise} \,, \end{cases} \tag{5.18}$$

$$\gamma_t = \gamma_{t-1} + v_t^{(9)} \,, \tag{5.19}$$

$$\gamma'_t = \gamma'_{t-1} + v_t^{(10)} \,, \tag{5.20}$$

$$\delta_t = \delta_{t-1} + v_t^{(11)} \,, \tag{5.21}$$

$$ST_t = ST_{t-1} + v_t^{(12)} \,, \tag{5.22}$$

where δ_t is the temperature-response factor, γ_t is the temperature-response indicator for the short-term effect in the load, γ'_t is the cumulative temperature-response indicator for the long-term effect in the load, $tp_{2,t}$ and $tp_{48,t}$ are the 2- and 48-hour mean temperatures [°C], respectively, and $\Delta tp_{2,t}$ and $\Delta tp_{48,t}$ are the difference between the temperature and ST_t [°C] in the short and long term, respectively. The notation $[x]_+$ means the hinge function: $\max\{0, x\}$. Note that the forms of Eqs. 5.17 and 5.18 depend on the season (the hot season in Japan is May 1–October 15). All indicators, including ST_t, were subject to constraints as explained in Section 4.4.

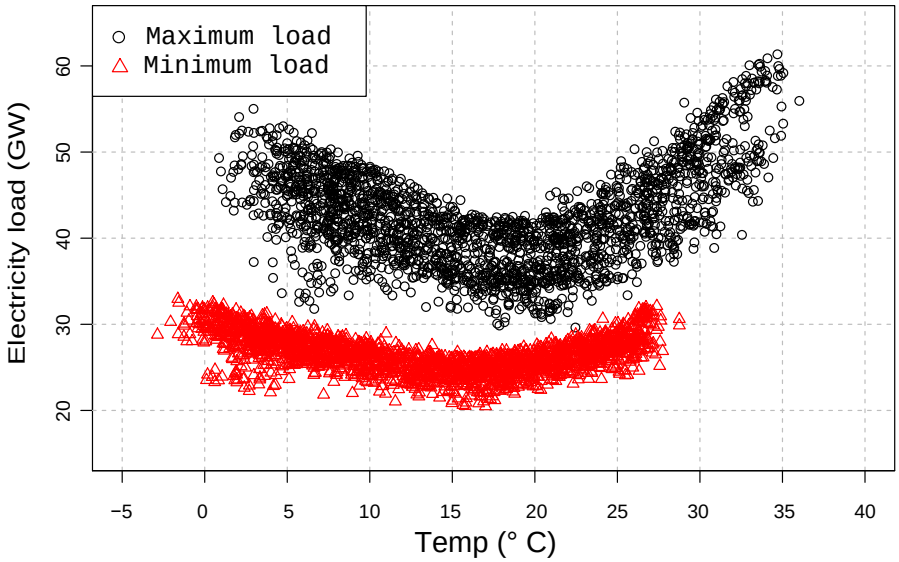

Fig. 5.1 Daily maximum (peak) and minimum (valley) loads in 2012.

Other Weather-Effect Component Models

Other weather-effect components are introduced in the following subsections. In these subsections, each component has an exogenous variable and a coefficient $\beta_t^{(\cdot)}$. $\beta_t^{(\cdot)}$ is a weather response indicator with the constraint $a^{(\cdot)} < \beta_t^{(\cdot)} < b^{(\cdot)}$. We have changed the constraints $a^{(\cdot)}$ and $b^{(\cdot)}$ for each calendar month. For instance, $\beta_t^{(ws)}$, the coefficient of wind speed, should be positive in winter, but it should be negative in summer, because of the cooling effect.

Solar-Radiation Effect Model Solar radiation causes direct heating and lighting effects on buildings throughout the year. A component for the effect of solar radiation is given in Eq. 5.23.

$$RA_t = \beta_t^{(ra)} ra_t \,, \tag{5.23}$$

where ra_t is the amount of global solar radiation [MJ/(m$^2\cdot$ h)].

Humidity Effect Model Humidity interacts with temperature and affects the load from June to September (the season with high temperatures and high humidity). Our model does not use a dummy variable for rain, since nearly 100% humidity is assumed to indicate

rainfall. A component for the effect of humidity is given in Eq. 5.24.

$$HM_t = \beta_t^{(hm)} hm_t \ , \tag{5.24}$$

where hm_t is the degree of relative humidity [%].

Wind Effect Model Wind causes direct cooling on buildings. A component for the effect of wind is given in Eq. 5.25.

$$WS_t = \beta_t^{(ws)} ws_t \ , \tag{5.25}$$

where ws_t is the wind speed [m/s].

PV Power Component Model Beginning in May 2013, TEPCO has added hourly estimates of the PV power to the total supply; hence, it is necessary to incorporate a PV component in our SSMs. A component for the effect of PV power is given in Eq. 5.26.

$$PV_t = \beta_t^{(pv)} \hat{P}_t \ , \tag{5.26}$$

where $\hat{P}_t$ is the hourly estimates of PV power [GW] obtained using the method that will be introduced in Chapter 6. Since $\hat{P}_t$ is itself an estimate of load, we used $\beta_t^{(pv)} \approx 1.0$.

5.1.2 SSMs Representation for Electricity Load Model

We used nonlinear system models for $f_t(\cdot)$ and a linear observation model for $g_t(\cdot)$. Technically, the observation model can be nonlinear [20], and this could increase the accuracy of the forecast; however, a linear model is more appropriate for load analysis, since the reports on load structural analysis may go to nonspecialists, and thus, they should be easy to

understand. The state vector $\mathbf{x}_t$ in our model setup is as follows:

$$\mathbf{x}_t \coloneqq \begin{bmatrix} TREND_t \\ (INWK_t, INWK_{t-1}, \cdots, INWK_{t-167})' \\ (INWKS_t, INWKS_{t-1}, \cdots, INWKS_{t-167})' \\ (INDAY_t, INDAY_{t-1}, \cdots, INDAY_{t-23})' \\ (INDAYS_t, INDAYS_{t-1}, \cdots, INDAYS_{t-23})' \\ (DAY_{t,0}, DAY_{t,1}, \cdots, DAY_{t,6})' \\ f(DAY_{t,j(t)}, INDAYS_t) \\ (AR_t, AR_{t-1})' \\ (TP_t, RA_t, HM_t, WS_t, PV_t)' \\ [h(\beta_1, a_1, b_1), h(\beta_2, a_2, b_2), \cdots, h(\beta_{11}, a_{11}, b_{11})]' \end{bmatrix}, \tag{5.27}$$

$$h(x, a, b) \coloneqq \log\left(\frac{x-a}{b-x}\right), \tag{5.28}$$

$$\beta \coloneqq \left(\alpha_{t,j(t)}, \theta_t, \phi_{2,t}, \gamma_t, \gamma_t', \delta_t, ST_t, \beta_t^{(ra)}, \beta_t^{(hm)}, \beta_t^{(ws)}, \beta_t^{(pv)}\right)', \tag{5.29}$$

where a and b are the lower and upper limits for a state variable x, respectively. The i^{th} element of β in Eq. 5.29 is denoted by β_i in Eq. 5.27. Note that the elements $INWK_{t-167}, INWKS_{t-167}$, $INDAY_{t-23}$ and $INDAYS_{t-23}$ are not required for a one-step-ahead prediction; however, they are included for smoothing purposes (such as with the variable span smoother in Eq. 5.4).

5.1.3 Lasso

For further enhancement of the accuracy, we used data from past years as well from the target year. In addition, we added several new variables which should be integrated into SSMs in the future; for example, temperature-dependent polynomials were used to represent rapid changes in the load, and dummy variables were used to represent weekdays with unusual loads (such as a day following a holiday or weekend).

We used the Lasso and MLR. The Lasso is a regression method that simultaneously shrinks the coefficients and selects the variables. We used it to reduce over-fitting and to clarify the essential explanatory variables for further refining component models of SSMs. The objective function of the Lasso that we used is given by Eq. 5.30. Note that the target of the Lasso modeling is the innovation (i.e., $y_i - \hat{y}_i$) which has been labeled as "Remainder"

in the figures in Section 4.6.

$$P_\lambda(\beta) = \Sigma_{i=1}^{N} w_i^{(yr)} w_i^{(tp)} (y_i - \hat{y}_i - \mathbf{x}_i\beta)^2 + \lambda \, \Sigma_{j=1}^{p} |\beta_j| \,, \tag{5.30}$$

where y_i is the i^{th} load in the training terms; $\hat{y}_i$ is the estimate from the EnKF; $\mathbf{x}_i = (1, x_{i,1}, \cdots, x_{i,p})$ is the standardized design vector with p explanatory variables; N is the total time in hours; λ is the regularization parameter; $\beta = (\beta_0, \beta_1, \cdots, \beta_p)'$ is a vector of the regression coefficients; $w_i^{(yr)}$ is the i^{th} regression weight which exponentially increases as the year of the i^{th} record gets closer to the target year; and $w_i^{(tp)}$ is the i^{th} regression weight, which exponentially increases as the i^{th} temperature in the training term gets closer to the forecast temperature for the target hour. The proper λ for each hour was chosen by a tenfold cross-validation with the mean-squared-error criterion.

Enhanced model

Twenty-four hourly innovation models, each given by Eq. 5.31, were used for the enhanced model, since the hourly models (i.e., 24 models) compared favorably to sequential time series models in a preliminary study [e.g., 58]. The resulting model is:

$$\begin{aligned} y_t - \hat{y}_t = {} & tp_{1,t} + tp_{1,t}^2 + tp_{1,t}^3 + tp_{6,t} + tp_{12,t} + tp_{24,t} + tp_{48,t} \\ & + hm_t + ra_t + ws_t + tp_{1,t} \cdot hm_t + tp_{1,t} \cdot ra_t \\ & + tu_t + we_t + th_t + fr_t + sa_t + su_t \\ & + ah_t + ho_t \quad \text{(regression coefficients are abbreviated)} \,, \end{aligned} \tag{5.31}$$

where y_t is the load; $\hat{y}_t$ is the estimate from the EnKF; $tp_{1,t}$, $tp_{6,t}$, $tp_{12,t}$, $tp_{24,t}$, and $tp_{48,t}$ are the 1-, 6-, 12-, 24-, and 48-hour mean temperatures [°C], respectively; hm_t is the relative humidity [%]; ra_t is the amount of global solar radiation [MJ/(m^{2}h)]; and ws_t is the wind speed [m/s]. The following notation was used for the days of the week and those of special significance: tu_t Tuesday, we_t Wednesday, th_t Thursday, fr_t Friday, sa_t Saturday, su_t Sunday, ah_t a day following a holiday or weekend, and ho_t a holiday; note that these are binary variables.

Training terms

The forecasting accuracy was enhanced using training data from the target year (YR) and the past five consecutive years (YR-1, ..., YR-5). The resulting terms are shown in Fig.

5.2. Suppose we would like to obtain a one-week forecast in the YR. The training and forecasting procedure would be as follows:

Step 1 Perform a one-week ensemble Kalman prediction for all terms of YR, YR-1, ..., YR-5 without overlapping terms.

Step 2 Fit the Lasso/MLR with the data in the training terms of YR, YR-1, ..., YR-5.

Step 3 With the fitted Lasso/MLR, perform a one-week prediction for the forecasting term of YR.

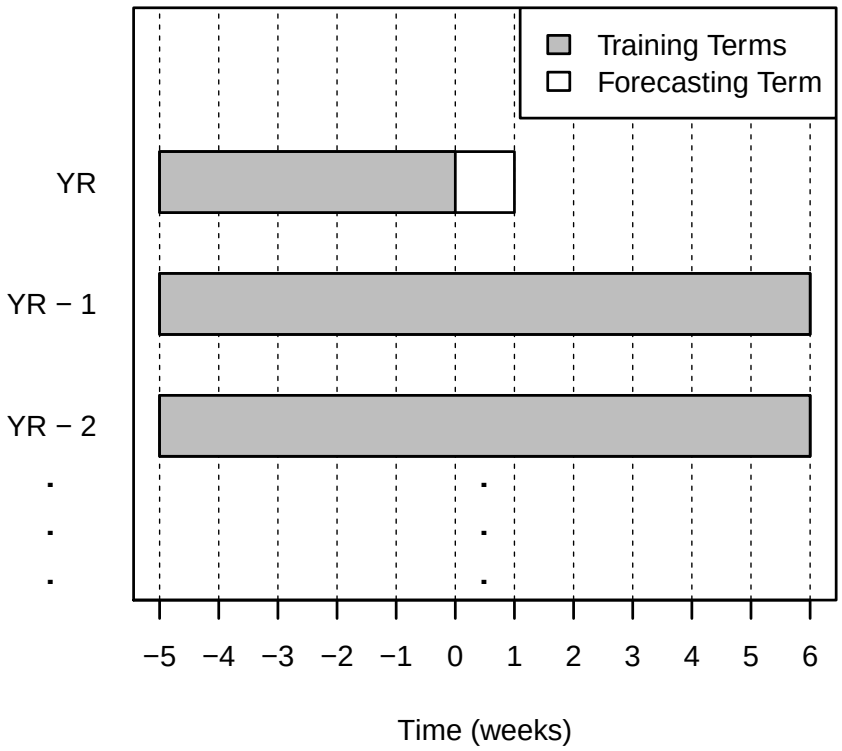

Fig. 5.2 Training terms used for enhancing the accuracy of a one-week forecast; the x-axis shows the time in weeks, and the y-axis shows the year, where YR is the target year that includes the one-week forecasting term, and YR-*i* is *i* years previous.

5.1.4 Flowchart of Proposed Ensemble Method

The flowchart of the proposed ensemble method illustrated in this section is shown in Fig. 5.3. The matrix $\mathbf{X}_{t|t-1}$ expresses the prediction distributions of states at time t, given states at time $t-1$. The subscript $_{t\,|\,t-1}$ is used to notate the prediction distribution. Likewise, the subscript $_{t\,|\,t}$ is used to notate the filtering distribution.

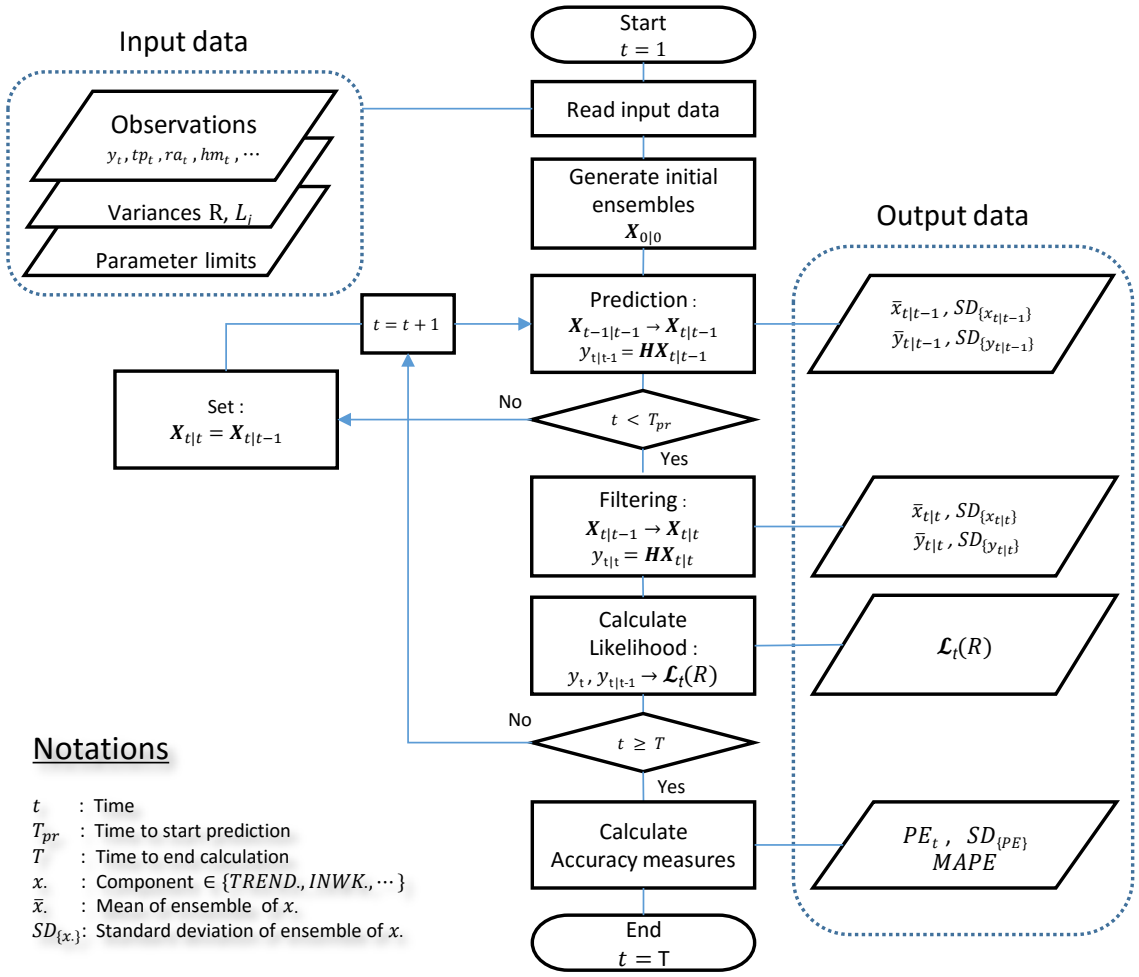

Fig. 5.3 Flowchart of the proposed ensemble method. One-week ahead forecasts ($\{\overline{y}_{t|T_{pr}-1}\}_{t=T_{pr}}^{T}$) can be obtained by repeating the one-step ahead prediction from the time T_{pr}. Filtered components ($\{\overline{x}_{t|t}\}_{t=1}^{T_{pr}-1}$) are used for load analysis (see Section 5.3.4).

5.2 Experiment

In this experiment, we used observations of the weather instead of forecasts to avoid possible effects from forecasting errors. Weather forecasts are usually calculated by supercomputers, and the major utilities in Japan purchase these forecasts. Therefore, a study of weather forecasting lies beyond the scope of this thesis. The local observations were averaged using weights that correspond to the ratio of the amount of local electricity sales; this was done because the load arises from an area that is geographically diverse.

There were marked drops in the load on particular days, such as the end-of-the-year and New Year holidays. Special treatments are usually required for forecasting those periods. Therefore, we did not use the load data in those periods, nor did we use it for the month following the 2011 earthquake.

In practice, load forecasting will be completed at hourly or half-hourly intervals, and this requires large computational resources and time. Our estimates were calculated on a rolling daily basis. For instance, if the first one-week forecast begins at 0:00 AM, then the next forecast will begin at 0:00 AM on the following day. The target years were 2012 and 2013, since complete annual data were required, and at the time of writing, of the years following the earthquake, those had the best available data.

We compared the performance of other six load forecasting models to that of our proposed models: EnKF, EnKF+Lasso, and EnKF+MLR. Four models were actually implemented and used to make forecasts; we used the same weather and load observations as were used to evaluate our models. For two of the models, the performance was taken from the literature. Figure 5.4 shows a schematic of the models used in this experiment; the details of the models will be given in the following subsections.

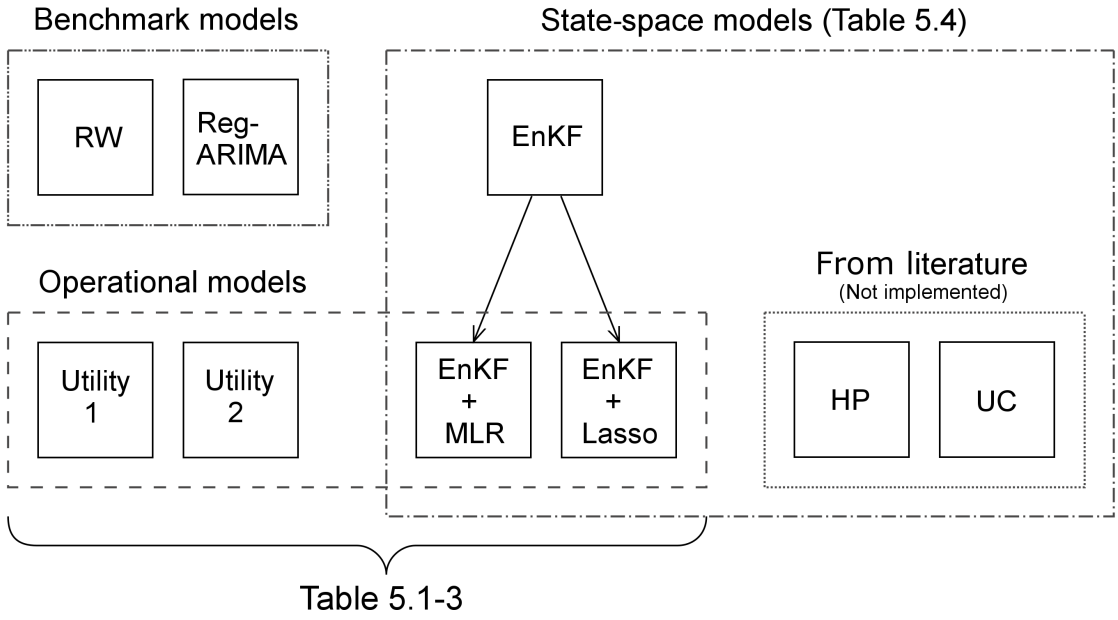

Fig. 5.4 Schematic of models implemented in the experiment. The scopes of the tables presented in Section 5.3 are also shown.

5.2.1 Benchmark Models

Benchmarks were calculated to help other researchers compare the accuracy of their models to ours. In general, benchmarks should be calculated using a well-known and accurate method that can be reproduced easily by anyone. From this perspective, we chose the following two benchmark models.

Weekly Random-Walk Model (RW) A random-walk model is the simplest and most commonly used benchmark method for time series models. Although it is a very simple model, a certain level of accuracy is achieved for a load forecast [64]. When using a horizon of only a few weeks, the load is largely explained by the day of the week, so for this study, we used a weekly random-walk model [14]. With this model (Eq. 5.32), each hourly forecast is the same as the actual hourly load during the preceding week; it is sometimes referred to as the naïve forecast.

$$\hat{Y}_t = Y_{t-168}. \tag{5.32}$$

ARIMA with External Regressors (RegARIMA) The autoregressive integrated moving average with external regressors is one of the most commonly used statistical models for time series analysis. It is very easy to reproduce the same forecasts by using *R*, a freely available statistical computing software package. The RegARIMA was calculated by the *auto.arima* function of the R package *forecast* [31], with the following settings: $allowdrift = F$, $stationary = T$, $max.P = 7$, $max.p = 24$, $max.Q = 7$, $max.q = 24$, $start.P = 7$, and $start.Q = 7$; other settings were set to the default values. Consecutive 10-week training terms, which were longer than those of EnKF (five weeks), were used to boost the forecasting accuracy, and the external regressors were the same as those used in our model.

5.2.2 Operational Models

TEPCO Model (Utility 1) Our method was compared with TEPCO's load forecasting model, which is the regression-based method that was primarily developed by Haida [26]. To compare like with like, we used an extended-time version of the model that is capable of two-week forecasting and is more accurate than the one currently in operation. The model was provided by TEPCO for the purpose of comparison under an obligation of confidentiality.

US Utility Model (Utility 2) For an additional comparison, we implemented a short-term load-forecasting model [30] that has been deployed at a utility in the USA and has demonstrated excellent performance. The model is defined as follows:

$$
\begin{aligned}
Y_t = \; & Trend + Day \times Hour + Month && (5.33)\\
& + (Month + Hour) \times \{T(t) + T(t)^2 + T(t)^3 && (5.34)\\
& \qquad + T(t-1) + T(t-1)^2 + T(t-1)^3 && (5.35)\\
& \qquad + T(t-2) + T(t-2)^2 + T(t-2)^3 && (5.36)\\
& \qquad + T(t-3) + T(t-3)^2 + T(t-3)^3 && (5.37)\\
& \qquad + T(t)_w + T(t)_w^2 + T(t)_w^3\} && (5.38)\\
& \qquad \text{(coefficients are abbreviated)} ,
\end{aligned}
$$

where Y_t is the load at hour t; $Trend$ is the yearly trend, which is equal to the year minus 2009 (e.g., the year 2010 is 1); $Hour$ is a 24-class dummy that represents the hourly variation in a day; Day is a seven-class dummy that represents the daily variation in a week; $Month$ is a 12-class dummy that represents the monthly variation in a year; and $T(t)$ is the temperature

at hour t. $T_w(t) = \Sigma_{k=1}^{24} \alpha^{k-1} T(t-k) / \Sigma_{k=1}^{24} \alpha^{k-1}$ and α (=0.9) is the smoothing factor. The weighting factor for exponentially weighted least squares for a three-year training term is 1.00015. We used the same parameter values for the α and the weighting factor as those used in the literature. When implemented, this model has 636 terms, including the intercept.

5.3 Results

5.3.1 Comparison by Type of Day

For each of the seven models, Table 5.1 shows the $MAPE$ and $SD_{\{PE\}}$ compared by type of day; the categories include all days, weekdays, weekends, and holidays. PE_ts were obtained by repeating a one-week forecast with a rolling daily origin. The performance was best for weekdays, followed in order by all days, weekends, and holidays; the mean performance was obtained when using all days.

The EnKF reached our target $MAPE$ level of 3.0% when using all days (3.05%) or weekdays (2.72%); however, weekends (3.80%) and holidays (3.85%) did not reach the target level. Nevertheless, the results outperformed those of the benchmarks (RW and RegARIMA) more than 2 to 3 points for any type of comparison. The RW performed better than expected for weekends (6.05%) in comparison with other type of day of the RW, while other models performed significantly worse on weekends.

The enhanced model, EnKF+Lasso, and EnKF+MLR improved the accuracy of the EnKF by 0.9 to 1.8 points, and they outperformed the Utility 2 MLR by 0.6 to 0.7 points; however, they underperformed the Utility 1 MLR by 0.2 to 0.6 points. For the $SD_{\{PE\}}$, the performances were in almost the same order as for the $MAPE$; the best was Utility 1, followed in order by the enhanced models, Utility 2, EnKF, RW, and RegARIMA.

Table 5.2 shows the DM statistics for 7 PM of the last (7^{th}) forecast day in the one-week-ahead forecast (164 hours from the forecasting origin).

The DM test is applicable to multiple-step-ahead forecasting, and we chose 7 PM of the last day in the one-week-ahead forecasting, since it was the hour when peak loads most frequently occurred during the years 2000–2013.

To counteract the multiple comparison problem, a significance-level correction (e.g., the Bonferroni correction) should be used when determining if there is a significant difference in the accuracy of different forecasts.

With the forecasts for 2012 and 2013, there was no significant difference between EnKF+Lasso

and EnKF+MLR, nor was there a significant difference between the two benchmarks, RW and RegARIMA. Otherwise, all comparisons showed significance.

5.3.2 Comparison by Calendar Month

Table 5.3 shows the monthly *MAPE* for each of the seven models. The EnKF has peaks (i.e., low accuracy) in March (3.33%), May (3.23%), October (3.26%), and November (3.52%). The accuracy of the enhanced models, EnKF+Lasso and EnKF+MLR, was low (over 2.0%) in April, May, and July; this was not true for the results obtained with the operational models or the benchmark models.

5.3.3 Comparison of SSMs Found in the Literature

The performances of the SSMs of other researchers were compared with those of the EnKF. The *MAPE*s of these models were derived from the literature of Dordonnat et al. [14] and Pedregal and Trapero [53]. Dordonnat developed hourly periodic SSMs with KF (HP models) for the French national electricity load, and Pedregal developed SSMs with the KF and a fixed-interval smoother (UC models).

Table 5.1 *MAPE*s by type of day for 2012 and 2013 (%)

Model	All days		Weekdays		Weekends		Holidays	
	$MAPE$	$SD_{\{PE\}}$	$MAPE$	$SD_{\{PE\}}$	$MAPE$	$SD_{\{PE\}}$	$MAPE$	$SD_{\{PE\}}$
	(n = 112,392)		(n = 78,624)		(n = 30,408)		(n = 3,360)	
Proposed								
EnKF	**3.05**	3.93	**2.72**	3.50	**3.80**	4.71	**3.85**	4.83
EnKF + Lasso	1.87	2.43	1.81	2.36	1.97	2.54	2.15	2.82
EnKF + MLR	1.86	2.43	1.81	2.35	1.98	2.56	2.17	2.84
Operational								
Utility 1 MLR	1.34	1.76	1.23	1.63	1.55	1.98	1.92	2.43
Utility 2 MLR	2.59	3.27	2.40	2.90	2.70	3.14	NA^a	NA^a
Benchmark								
RW	6.12	8.41	6.02	8.25	**6.05**	8.36	9.05	10.95
RegARIMA	6.21	8.11	5.49	6.54	8.03	7.63	6.58	7.53

Note. Hourly predictions in the special terms are not included. Each sample size is equal to the number of forecasting hours calculated on daily rolling basis.

[a] Holiday effect is not implemented here.

Table 5.2 Diebold-Mariano statistics for 7 PM of weekdays

	EnKF+Lasso	EnKF+MLR	Utility 1	Utility 2	RW	RegARIMA
EnKF	+2814.1 ***	+2807.1 ***	+3815.4 ***	+2318.9 ***	-4.6 ***	-8.9 ***
EnKF+Lasso		**-0.1**	+10.3 ***	-495.2 ***	-5.7 ***	-9.7 ***
EnKF+MLR			+6.4 ***	-488.2 ***	-5.6 ***	-9.2 ***
Utility 1				-1496.5 ***	-6.6 ***	-11.8 ***
Utility 2					-5.4 ***	-15.7 ***
RW						**+0.1**

Note. p-values based on the two-sided test were coded with the significance codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
With the Bonferroni correction, *** indicates there remained a significant difference at the significance level of 5%.

Table 5.3 Monthly *MAPE*s of weekdays for 2012 and 2013 (%)

Month		**Proposed**			**Operational**		**Benchmark**	
	n	EnKF(1)	(1) + Lasso	(1) + MLR	Utility 1	Utility 2	RW	RegARIMA
Jan	5,712	2.46	1.68	1.66	1.53	2.12	8.58	6.43
Feb	6,720	2.29	1.58	1.59	1.10	2.07	5.54	4.97
Mar	6,888	**3.33**	1.77	1.75	1.64	3.22	7.72	4.82
Apr	6,720	2.49	**2.06**	**2.04**	1.32	4.14	3.97	4.82
May	6,048	**3.23**	**2.36**	**2.36**	1.01	2.59	3.68	5.75
Jun	6,888	1.98	1.33	1.38	0.78	2.19	3.61	5.26
Jul	7,224	2.66	**2.01**	**2.01**	1.37	2.63	9.15	6.68
Aug	5,880	2.07	1.51	1.50	1.32	1.91	8.54	5.74
Sep	6,384	2.32	1.92	1.88	1.41	1.92	8.95	4.49
Oct	7,392	**3.26**	1.77	1.76	0.93	1.72	4.25	4.96
Nov	6,888	**3.52**	1.87	1.88	1.03	1.96	4.21	6.46
Dec	5,880	2.92	1.87	1.88	1.45	2.21	4.54	5.69
Jan–Dec	78,624	2.72	1.81	1.81	1.23	2.40	6.02	5.49

Note. Hourly predictions in the special terms are not included.

Table 5.4 *MAPE*s of SSMs (%)

Target Time	**Proposed**			**HP Models**[a]				**UC Models**[b]	
	EnKF(1)	(1)+Lasso	(1)+MLR	Reg	Univ	TVR	TTR	UC	UC1/ 2
9 AM	3.49	2.11	2.10	2.77	3.07	3.06	3.09	*NA*	*NA*
12 AM	2.81	2.08	2.09	2.42	2.49	2.67	2.66	*NA*	*NA*
Every Hour	3.05	1.87	1.86	*NA*	*NA*	*NA*	*NA*	4.37	3.92

Note. Hourly predictions in the special terms are not included except UC models.
[a] Hourly periodic SSMs with KF [14] consist of the following models. Reg: regression model, Univ: univariate model, TVR: time-varying regression model, and TTR: local linear trend + time-varying regression model.
[b] SSMs with KF and a fixed interval smoother called the unobserved components models [53]. UC1/2 has a component for the annual cycle in addition to UC.

5.3.4 Load Analysis

The load analysis can be done by using the filtered load components. Figure 5.5 shows a snapshot of the load structure decomposed by the EnKF, and Fig. 5.6 shows the model parameters, which were estimated simultaneously. We can observe a gradual data-assimilation process by the EnKF from June 15th to July 19th.

The load analysis provides us with the following information:

$TREND$: Seasonal trend level

$INDAYS$: Averaged intra-daily load curve

DAY : Relative load levels for Mondays, Tuesdays, ..., and holidays

TP : Temperature effect

HM : Humidity effect

RA : Solar radiation effect

PV : Photovoltaic effect

γ, δ : Instantaneous temperature response

γ' : Cumulative temperature response

ST : Switch-off temperature for air conditioners

$\beta^{(ra)}$, $\beta^{(hm)}$, $\beta^{(ws)}$: Weather responses which represent direct heating by solar radiation, dehumidification, and direct cooling by wind

There are many applications for the information produced by the analysis. For example, the seasonal trend level ($TREND$) can be used for the seasonal adjustment routine in the planning division of a utility.

The averaged intra-daily load curve ($INDAYS$) can be used to determine constant outputs of base-load units such as coal-fired thermal power plants and nuclear power plants.

The curve with relative load levels ($f(DAY, INDAYS)$) can be used as a year-long dispatching curve for the scheduling of thermal power plants. With this curve, cost-saving fuel procurement will be possible.

The instantaneous temperature-response indicators (γ and δ) are essential for letting system operators know the amount of supplies to order; they can perform a quick forecast with the indicators in case the loads are underestimated or the weather forecast was unreliable.

The cumulative temperature-response indicator (γ') provides us with information about the insulation efficiency of buildings.

The switch-off temperature (ST) can be used as a more sophisticated reference temperature for calculating heating and cooling degree days.

Precise effect analysis regarding solar radiation is possible through the relationship of components RA and PV as shown in Fig. 5.7.

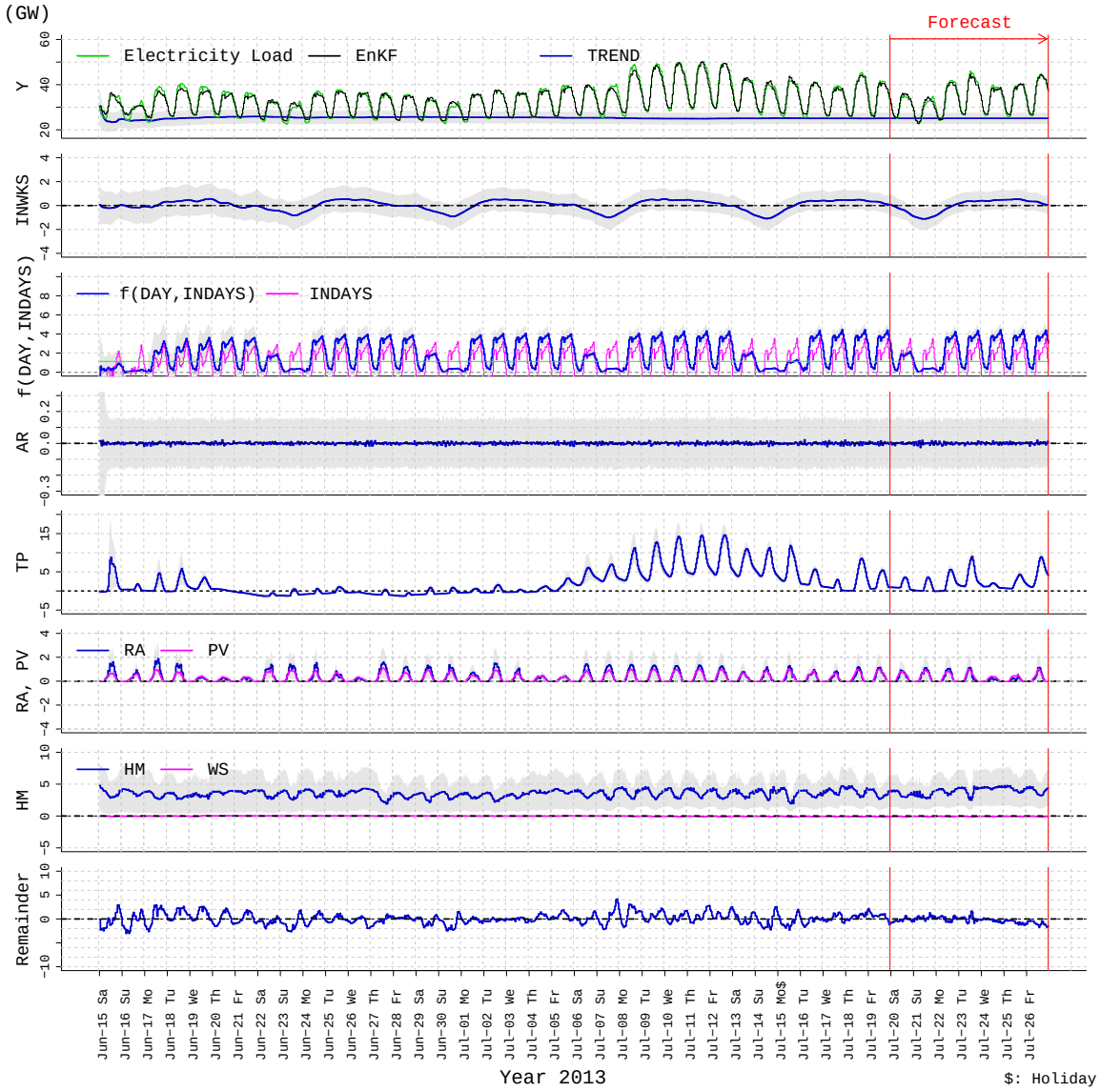

Fig. 5.5 Snapshot of the electricity load structure, as decomposed by the EnKF. The blue line in each panel shows the ensemble mean, and the gray region surrounding each line shows the domain that is within one standard deviation of the mean.

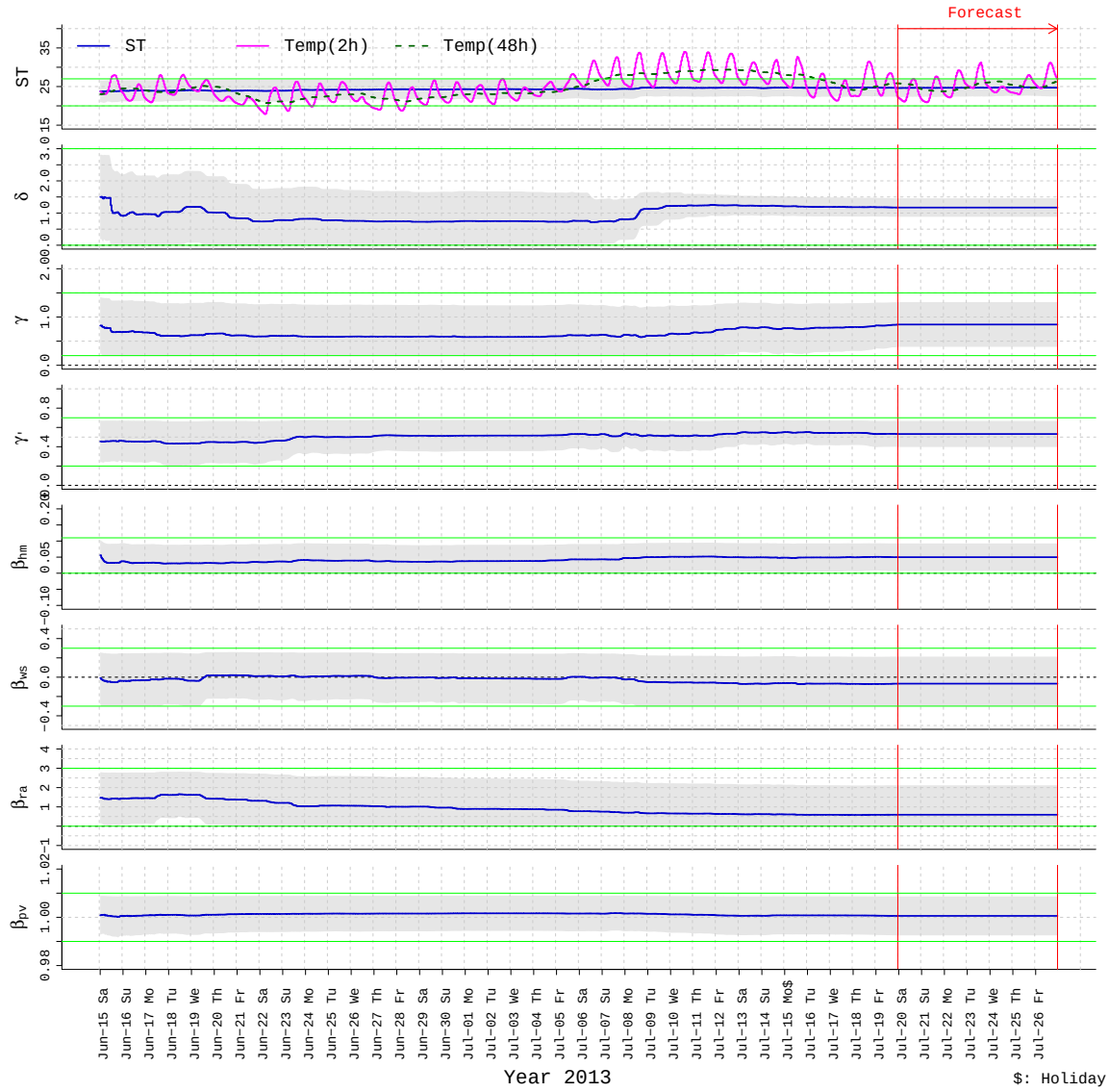

Fig. 5.6 Model parameters simultaneously estimated in the filtering process of the EnKF (*cf.* Fig. 5.5). The blue line in each panel shows the ensemble mean, the gray region surrounding each line shows the domain that is within one standard deviation of the mean, and the outer lines indicate the upper and lower limits of the ensembles. Note that Temp(2h) and Temp(48h) in the top panel are observations added for indicating the position of ST relative to those temperatures.

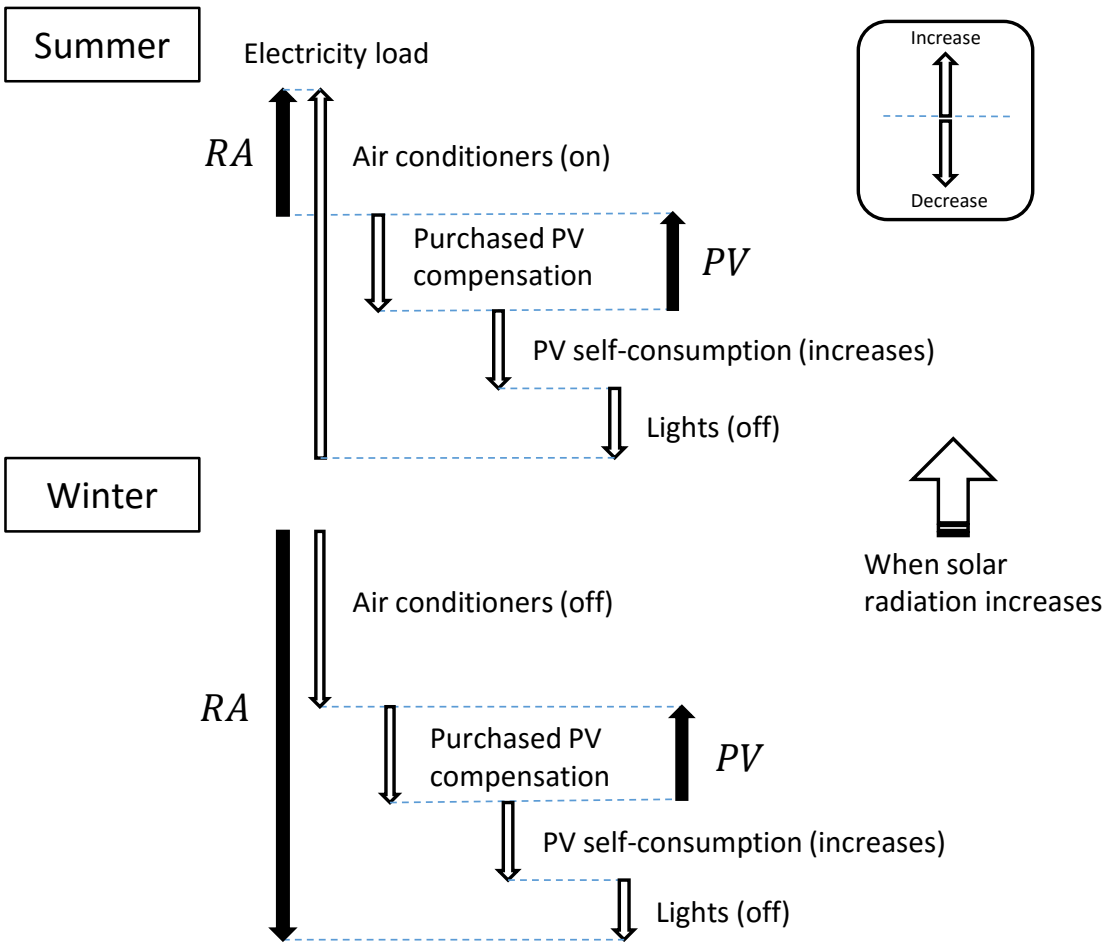

Fig. 5.7 Relationship diagram of components RA and PV. When solar radiation increases, both RA and PV increase in summer, and RA decreases while PV increases in winter. Note that electricity load decreases when PV self-consumption increases

5.4 Discussion

When comparing performance by type of day, the order is the same as that which is often found in the load-forecasting literature. This can be explained by the amount of training data available for each type of day; there is a large amount of training data for weekdays, but only a limited amount for other types of days. For this reason, the intra-daily curve, which is the base curve used for all days, is similar to that of weekdays. In our modeling, the intra-daily curve was adjusted by using the daily coefficient to express each type of day. As a result, weekdays tend to be the easiest to forecast.

Benchmarks were very useful for understanding the relative accuracy, and they also helped us to locate the weak points of the proposed models. For weekdays, the EnKF outperformed the benchmarks (RW and RegARIMA) by a large margin—over 3.0 points. In contrast, for weekends, the EnKF obviously did not perform as well, since the accuracy of the benchmark RW was almost the same for both weekdays and weekends.

There was no significant difference between the accuracies of the enhanced models EnKF+Lasso and EnKF+MLR. Therefore, instead of using MLR, we recommend that utilities use the Lasso, which has additional useful functionalities, such as variable selection and over-fitting prevention.

It is entirely fair to say that the accuracy of the enhanced models reached an operational level, since they outperformed Utility 2, which has been reported to have excellent performance in actual operation. Furthermore, the accuracy gap between Utility 1 and our model was not very large, and it can be removed by fine tuning the parameters.

When comparing accuracy by calendar month, the EnKF showed marked drops ($MAPE$ over 3.0%) in March, May, October, and November. We determined that the drop in March occurred due to the relatively high temperatures observed in 2013, and the drop in May occurred due to the lack of sufficient filtering terms for the period immediately after a long special term (the Golden Week holidays). We did not implement any remedial models for special terms; hence, the special terms were skipped by repeating the one-step-ahead prediction, and so those terms did not contribute to the data assimilation. The last two drops occurred due to a change in the temperature effect component in the model (see Eqs. 5.17 and 5.18). We conclude that countermeasures should be developed to strengthen our models for seasonal changes.

Compared with similar studies in the literature, our method can be distinguished from those of [14] and [53], which also use SSMs, by the type of filter we used. Instead of a KF, we used the EnKF, which can handle nonlinear models. It must be noted that the SSMs

used in the other studies were designed for operational use, while the EnKF is intended for analytical use. However, our enhanced models for operational use greatly outperformed those models.

Improving the accuracy of load forecasting is expected to reduce costs in the following categories:

- Unit commitment (mainly, the start-up cost for a thermal unit)
- Imbalance charges
- Emergency power exchanges
- Outages
- Investment in power facilities

It is very difficult to directly measure all the costs that will be reduced by improving the accuracy. However, in the literature, we found several attempts at estimating these costs. As an example, the economic impact of load forecasting errors was assessed by using a Monte Carlo approach to simulate the unit commitment for that forecast [72]. It was determined that the imbalance charge[1] for a power shortfall beyond 3% of the load is a few times higher than the thermal generating cost.[2] As an example of calculating the imbalance charge, note that during a summer day, a 4% underestimation that lasts for three hours will result in a loss of approximately ¥106 million. Moreover, emergency power-exchange charges are much higher than imbalance charges; the charge is doubled if it is within 3%, and it is higher by a factor of $3/2$ if it is beyond 3%. Therefore, even a difference in accuracy of 0.1% can make a big difference in the profit, especially for large utilities. In this respect, our method apparently has large economic advantages over the existing state-of-the-art methods found in the literature.

The EnKF in our method requires a relatively short training period (several weeks at most), and this is favorable for capturing rapid changes in the load structure. On the other hand, it tends to be less accurate than a method that uses longer training periods, especially if it includes terms from past years. The year-long $MAPE$ of the one-week forecast was

[1] 53.21 (summer), 47.03 (other seasons), and 28.84 (at night) [¥/kWh] for beyond 3%, and 15.44 [¥/kWh] for within 3% [65]

[2] 10.2 [¥/kWh]: the mean thermal generation cost of 12 companies over five years [34]

below 3.0%, though a few large *MAPE*s were observed; this shows that the model is sufficiently accurate to perform a precise load analysis from which trustworthy indicators can be obtained.

Planning divisions of utilities obtain the temperature-response indicator γ_t by using a simple linear regression, and then use it for canceling the temperature effect for long-term load forecasting. Also, the heating/cooling degree day is one of the most widely used explanatory variables in economic methods for load analysis; it is used to express annual energy consumption. It is a very rough estimation of the relationship between energy consumption and temperature, and it is primarily derived from the use of air conditioners. We expect that a more precise analysis will be possible by using our newly introduced response indicators, γ'_t and δ_t.

Since it is important to have an accurate load analysis in order to forecast the growth in electricity demand and make proper investment plans for electric power facilities, a comprehensive load analysis is conducted in Japan every few years to estimate the future residential electricity demand [51] and to determine its sensitivity to temperature [33]. Our results can be used for such a long-term forecast as well as for a short-term one, which is our remarkable achievement that cannot be found in any literature.

PV forecasting and load forecasting are conducted separately in most utilities. In our method, PV power was incorporated into the SSMs as one of the load components; this enables the simultaneous estimation of the model parameters, and avoids the problem of inconsistent results due to separate estimations.

Ideally, the load forecast should be made solely by the EnKF, in order to ensure that the accountability is accurate; this is because the enhanced models have redundant variables. These redundant variables should be reduced primarily by directly enhancing existing component models, or in the following way. One of the merits of using SSMs lies in their capability to unify many separate time series models; thus, we expect that the Lasso and MLR can be incorporated into SSMs relatively smoothly, compared to incorporating them into existing methods. In the case of the Lasso, this would be achieved by using a Laplacian prior [52] as the coefficient of an exogenous variable, since the EnKF produces a natural Bayesian posterior.

Chapter 6

Photovoltaic Power Forecasting

6.1 Methods

6.1.1 Monthly Installed PV Capacity Model

Installed PV capacity is an essential exogenous variable to forecast PV power. In order to estimate these hourly values, the monthly capacity for each type was modeled in advance, using the following polynomial regression model:

$$C_m = [\mathbf{X}\beta]_+ , \tag{6.1}$$

$$\mathbf{X} = \left(1, m, m^2, m^3\right) , \tag{6.2}$$

$$\beta = (\beta_0, \beta_1, \beta_2, \beta_3)' , \tag{6.3}$$

where C_m is the total capacity of the m^{th} month, $\mathbf{X}$ is the design vector of variable m, and β is the coefficient vector for $\mathbf{X}$. Note that $[\cdot]_+$ is the hinge function.

6.1.2 Module Temperature Model

Operating temperature and module performance are closely related, and this relationship has been extensively investigated [60]. Nominal operation cell temperature has been used as an indicator of PV module temperature [3]. In the present study, since the local air temperature and wind speed were available as input data, we adopted a more detailed model, namely, Sandia's module temperature model [41]. Since the typical temperature difference between the cell and the module has been reported to be 2 to 3 °C for flat-plate modules at an irradiance level of 1 kW/m^2, we assumed that the module temperature $T_{i,t}^{(module)}$ was

approximately the same as the cell temperature $T_{i,t}^{(cell)}$. The proposed model is expressed by the following equation:

$$T_{i,t}^{(cell)} \approx T_{i,t}^{(module)} = T_{i,t}^{(air)} + r_{i,t} \exp(\beta_m + \gamma_m s_{i,t}) \ , \tag{6.4}$$

where subscripts t and m indicate data related to the elapsed time in hours and in months, respectively, subscript i indicates data related to the i^{th} observatory or its coverage area, $T_{i,t}^{(air)}$ is the air temperature [°C], $r_{i,t}$ is the mean global solar radiation [W/m^2], $s_{i,t}$ is the wind speed [m/s], and β_m and γ_m are Sandia's empirically determined coefficients, which vary depending on module type.

6.1.3 Hourly PV Power Models

We made the model as simple as possible so that it could be easily understood by utility practitioners without losing the essential features of the PV generation mechanism. In most studies, conventional PV system parameters are treated as fixed values [e.g., 76] since the PV data are obtained from a limited number of PV firms with uniform PV system specifications. However, in our case, it is natural to use PV system parameters which vary monthly since the target PV power comes from various types of PV systems, and the proportions coming from the various system types vary daily. The proposed hourly PV power models, which consist of a local area model (Eq. 6.5) and a total area model (Eq. 6.6), are expressed by the following equations:

$$p_{i,t} = \eta_m \ (w_{i,m} \ c_t \ r_{i,t})^{\delta_m} \kappa_{i,t} \ , \tag{6.5}$$

$$p_t = \sum_{i=1}^{9} p_{i,t} \ , \tag{6.6}$$

where $p_{i,t}$ is the hourly PV power from the i^{th} area, c_t is the hourly total capacity obtained via spline-interpolation with monthly total capacity C_m (see Section 6.1.1), $w_{i,m}$ is the weight for the total capacity to make local capacities, and η_m and δ_m are the PV-conversion coefficient and factor, respectively. The basic modeling concept is that the hourly PV power is proportional (η_m) to the product of the installed PV capacity ($w_{i,m} \cdot c_t$) and solar radiation ($r_{i,t}$), with a slight nonlinearity ($\delta_m \approx 1$).

In order to take into consideration efficiency drops caused by temperature and wind, we

used the cell temperature factor [32]:

$$\kappa_{i,t} = 1 + \alpha_m \left(T_{i,t}^{(cell)} - 25 \right) / 100 \,, \tag{6.7}$$

where α_m is the maximum-power temperature coefficient.

6.1.4 SSMs Representation for Monthly PV Purchase Volume Model

The hourly PV power models and their parameters, described in the previous subsections, were incorporated in the following SSMs on a monthly basis. Using the EnKF, each state variable of the SSMs was represented as a distribution by ensemble members (typically 50–1000 members). The prediction distribution or one-step-ahead forecast of each state variable was calculated using the ensemble square-root filter (EnSRF) [75] as follows:

$$y_m = \mathbf{H}\mathbf{x}_m + W_m \,, \tag{6.8}$$

$$\mathbf{H} = (\underbrace{1, \cdots, 1}_{9}, 0, \cdots, 0) \,, \tag{6.9}$$

$$\mathbf{x}_m = f(\mathbf{x}_{m-1}, \mathbf{c}, \mathbf{r}, \mathbf{s}, \mathbf{T}, \mathbf{v}_m) \,, \tag{6.10}$$

$$\mathbf{x}_m = \begin{pmatrix} (P_{1,m}, \cdots, P_{9,m})' \\ h(w_{1,m}, a^{(w1)}, b^{(w1)}) \\ \vdots \\ h(w_{9,m}, a^{(w9)}, b^{(w9)}) \\ h(\eta_m, a^{(\eta)}, b^{(\eta)}) \\ h(\delta_m, a^{(\delta)}, b^{(\delta)}) \\ h(\alpha_m, a^{(\alpha)}, b^{(\alpha)}) \\ h(\beta_m, a^{(\beta)}, b^{(\beta)}) \\ h(\gamma_m, a^{(\gamma)}, b^{(\gamma)}) \end{pmatrix} \,, \tag{6.11}$$

$$h(x, a, b) \coloneqq \log\left(\frac{x-a}{b-x} \right) \,, \tag{6.12}$$

$$P_{i,m} = \sum_{t=t_m}^{t_{m+1}-1} p_{i,t} \,, \tag{6.13}$$

where y_m is the observation which is the total monthly purchase volume [GWh], $\mathbf{x}_m$ is the state vector, $\mathbf{H}$ is the observation vector. f is the system model function which transforms a filtering distribution to a prediction distribution with a first-order trend model (i.e., $x_m = x_{m-1} + v$; $x.$ and v are a state variable and system noise, respectively). Moreover, $\mathbf{c}$, $\mathbf{r}$, $\mathbf{s}$, and

T are the hourly PV capacities, solar radiation, wind speed, and air temperature in the m^{th} month, respectively. The function h is the map function such that $h : x \in (a, b) \longmapsto (-\infty, \infty)$ and a and b are the lower and the upper limits for a state variable x (i.e., $a < x < b$). The term W_m is the observation noise such that $\{W_m\} \sim$ i.i.d. $\mathcal{N}(0,\ R)$, where R is the variance of the observation noise. Also, $\mathbf{v}_m$ is the system noise vector such that each element $\{v_m^{(i)}\} \sim$ i.i.d. $\mathcal{N}(0,\ R/L_i)$, where L_i is a trade-off parameter and $i = 1, \cdots, N_v$ (size of $\mathbf{v}_m$).

$P_{i,m}$ is the monthly purchase volume from the i^{th} area, and t_m denotes the first hour of the m^{th} month. This monthly volume is calculated by using hourly power $p_{i,t}$:

$$p_{i,t} = \eta_m \, (w_{i,m} \, c_t \, r_{i,t})^{\delta_m} \kappa_{i,t} \tag{6.14}$$

$$= \eta_m \, (w_{i,m} \, c_t \, r_{i,t})^{\delta_m} \left[1 + \alpha_m \{T_{i,t}^{(cell)} - 25\}/100\right] \tag{6.15}$$

$$= \eta_m \, (w_{i,m} \, c_t \, r_{i,t})^{\delta_m} \left[1 + \alpha_m \{T_{i,t}^{(air)} + r_{i,t} \exp(\beta_m + \gamma_m s_{i,t}) - 25\}/100\right], \tag{6.16}$$

where c_t, $r_{i,t}$, $s_{i,t}$ and $T_{i,t}^{(air)}$ are elements of the hourly PV capacity vector **c**, the weather data matrices **r**, **s**, and **T**, respectively.

6.1.5 Flowchart of Proposed Ensemble Method

The proposed method is performed in accordance with the following calculation flow. This process was performed individually for each type of supplier.

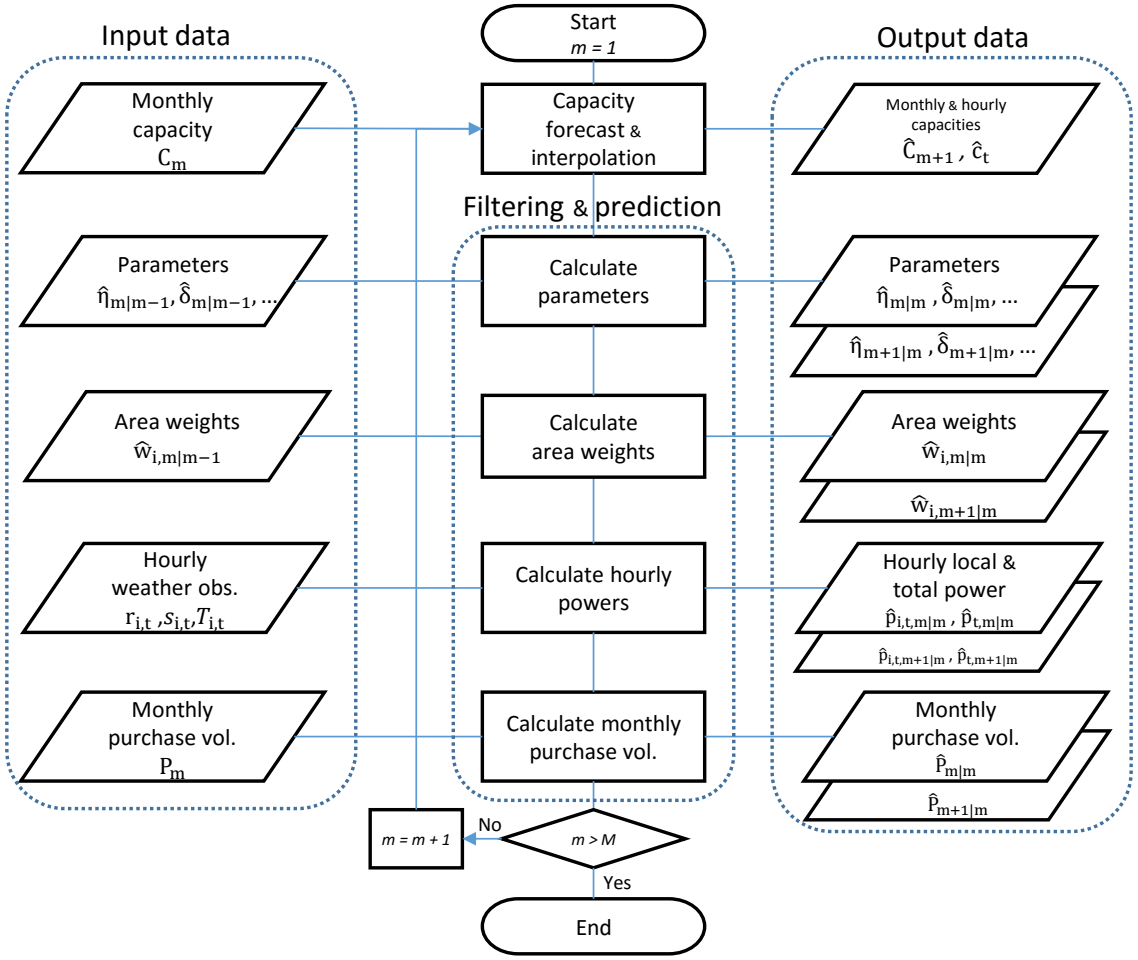

Fig. 6.1 Flowchart of the proposed ensemble method immediately after the m^{th} month. Here, m and M are the month indicator and the maximum value of m, respectively. From this process, we obtain the filtering $(m|m)$ and prediction distributions $(m+1|m)$ for all state variables. Note that $\hat{\cdot}$ expresses the estimated value.

6.2 Experiment

We applied the proposed model to PV purchase volumes of both types. Thus, the PV system parameters, namely, the local and the total powers, were estimated separately according to the type. As for the limits for $w_{i,m}$ in Eq. 6.5, we used ±20% of the capacity rates (see Figs. 2.5 and 2.6). For the other parameters, $0.0 < \eta_m < 1.0$, $0.0 < \delta_m < 2.0$, $-0.5 < \alpha_m < -0.2$, $-4.0 < \beta_m < -2.0$, and $-0.2 < \gamma_m < 0.0$. The limits for α_m, β_m, and γ_m were taken from the literature [32, 41]. The target monthly PV purchase volume does not include PV self-consumption which corresponds to about 13% of the total PV generated. Therefore, we used relatively wider ranges for the limits to simulate PV systems that virtually output only purchased PV power.

Table 6.1 shows the type of input data used to obtain a distribution of a state variable (i.e., local and total PV powers and PV system parameters) in the experiment. For a filtering distribution, given all the observations, the local and total PV powers, and the system parameters were estimated. For a prediction distribution, all given data must be forecast. However, weather observation was used to avoid weather forecasting error and precisely assess the accuracy of the proposed model.

The prediction distribution is used to estimate the local and total PV powers intended for use in utility PV forecasting operations. On the other hand, the filtering distribution, as well as the smoothing distribution, is used for system-parameter estimation and structural analysis for retrospective evaluation. Since filtered PV system parameters, as well as hourly powers, have important analytical information, the filtering errors were also examined, as shown in the tables in Section 6.3, in order to understand the quality of the analysis.

Table 6.1 Data types used in the experiment

State distribution	Purchase volume	Capacity	Weather
Filtering	Observation	Observation	Observation
Prediction	–	Forecast	Observation [a]

[a] In actual operation, weather forecast will be used.

6.2.1 Benchmark Model

As a benchmark model for accuracy comparison, we used an hourly PV power model which has been in operational use [29]. Forecasts were calculated with weather observations. The model consists of three well-known models. The hourly PV power is estimated using the

following procedure:
First, by using the Erbs model [17], the hourly global horizontal solar radiation is split into its beam and diffuse components. Second, using the Perez diffuse irradiance model [55], these components are combined into hourly irradiance on tilted planes. Finally, the hourly irradiance is converted to PV power using a simple linear regression model.

6.3 Results and Discussion

6.3.1 Hourly Interpolation of Monthly Installed PV Capacity

Figure 6.2 shows plots of monthly reported installed PV capacities and interpolated hourly capacities. Each polynomial regression model fit the reported monthly PV capacities very well, and thus, accurate forecasts were obtained.

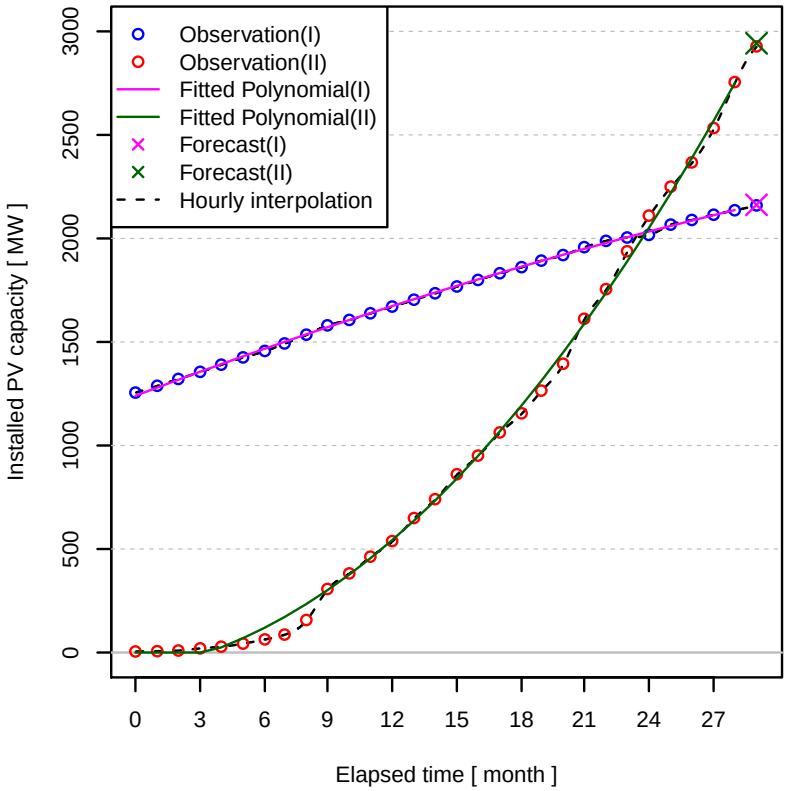

Fig. 6.2 Interpolated installed PV capacities. This figure shows data for an example of fitting a polynomial regression model, with a one-step-ahead forecast of the last month, for each supplier. Observations (circles) were monthly reported installed PV capacities. Fitted values (solid lines) and forecast values (crosses) were calculated with the polynomial regression. Hourly interpolations (dashed lines) were spline-interpolated values. Here, (I) and (II) indicate type-I and type-II suppliers, respectively.

6.3.2 Data Assimilation of Monthly PV Purchase Volumes

Figures 6.3 and 6.4 show the data-assimilation results—the one-step-ahead forecasts of states—for type-I and type-II suppliers, respectively. The area within one-standard devi-

ation of the mean of the prediction distribution is shown in gray, and we can understand how a state variable (i.e., any of local purchase volumes, weights, and parameters) converges from its behavior. The convergence pattern of each state variable was similar for the type-I and type-II suppliers. The PV conversion factor δ_m quickly converged after the sixth month. Here, η_m slowly converged, changing its mean.

On the other hand, the other weather related parameters, namely, α_m, β_m, and γ_m, exhibited poor convergence, though the parameters began to change when the PV investment boom began at the 22^{nd} month. The most likely explanation for this phenomenon is that these parameters were well adjusted from the beginning, with the narrow limits obtained from the literature and a preliminary study (see Section 6.2). Another explanation for this phenomenon is that the effects of weather on the efficiency of a PV system are limited (10% at most), which prevents the parameters from converging well. According to a previous study [32], the cell temperature factor $\kappa_{i,m}$ ranges from 91% to 94% in August, and 100% to 103% in February.

Moreover, the weights $w_{i,m}$ did not change significantly, in contrast to our expectations. The PV system penetration rate might be approximately the same in each local area for the period evaluated. Each local PV power can be affected by other local PV powers through the weights $w_{i,m}$. However, the straight lines representing the weights suggest that almost no correlation exists. Further investigation of the weights is required.

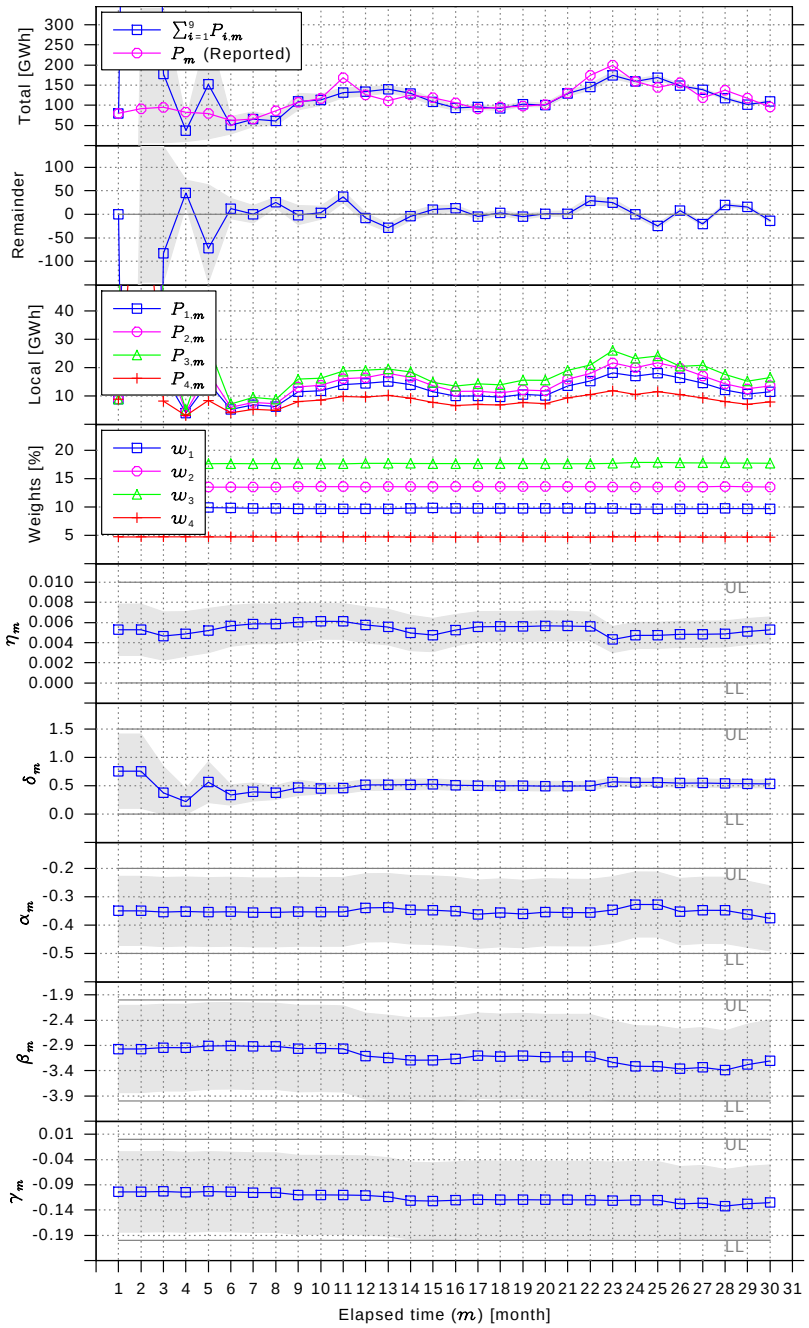

Fig. 6.3 One-step-ahead forecasts for type-I supplier. Each solid line shows the mean of prediction ensemble members. The region (gray area) surrounding the line indicates the area within one-standard deviation of the mean. UL and LL lines indicate the upper and the lower limits, respectively. In the top panel, P_m shows the reported PV purchase volumes, and $\sum_{i=1}^{9} P_{i,m}$ shows the forecasts. The second panel shows the remainder which is $P_m - \sum_{i=1}^{9} P_{i,m}$. The third panel shows the forecasts of the local PV purchase volumes. The fourth panel shows the weights for the total capacity to make the local capacity. Since the graphic space is limited, only the first four local volumes and corresponding weights are shown. The following five panels show model parameters.

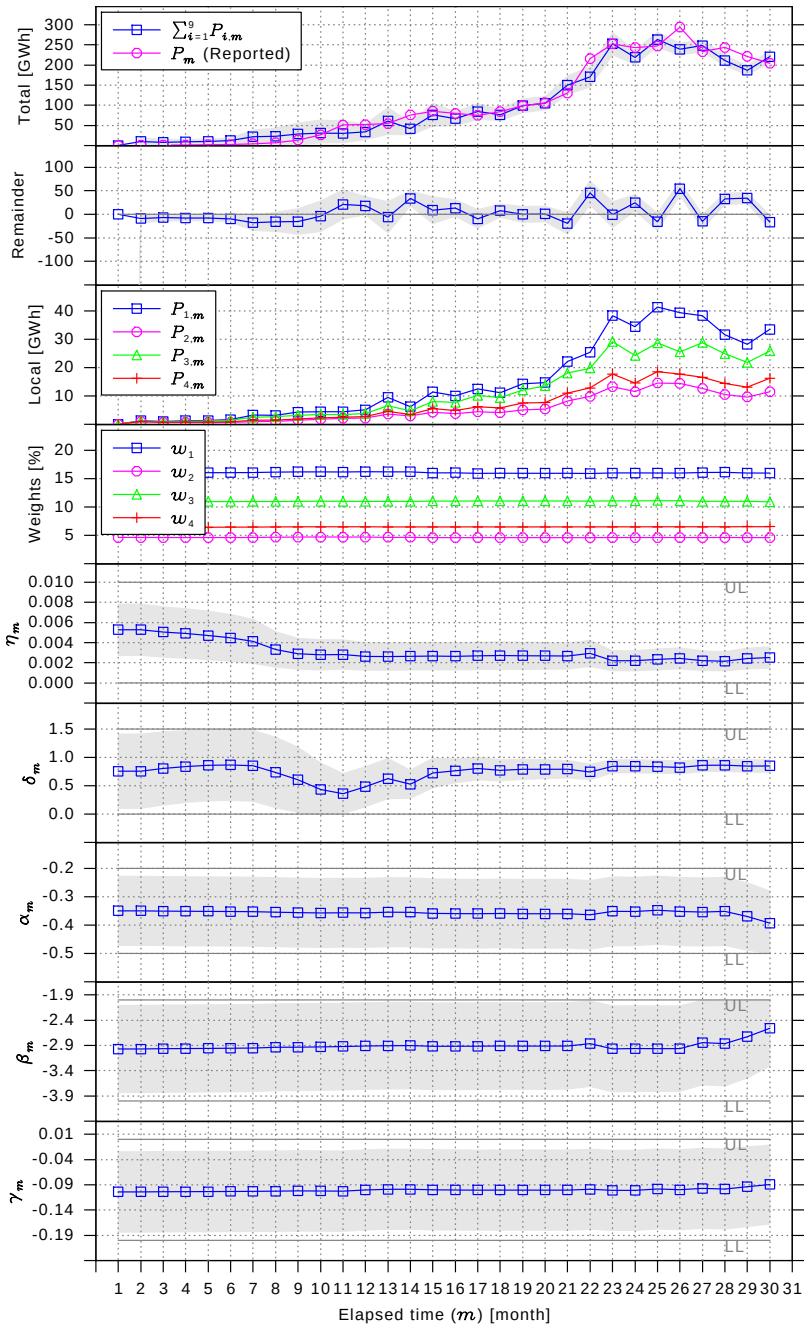

Fig. 6.4 One-step-ahead forecasts for type-II supplier.

6.3.3 PV Efficiency Indicators

It is natural to use the PV conversion coefficient η_m that is time-varying on a monthly basis since the target PV power comes from a mixture of all types of PV systems, and the mixture rate has been changing day by day. Using the PV-conversion factor δ_m, we expressed the nonlinearity of PV power caused by efficiency change as mostly due to the variability of photovoltaic cell temperatures; however, as another enhanced model, one can use the cell temperature factor $(1+\kappa)(PCT-25)$ [32], where PCT is the photovoltaic cell temperature and κ is a negative coefficient ($-0.5 < \kappa < -0.2$).

6.3.4 Hourly PV Power

Using the estimated PV system parameters ($\hat{\eta}_m$, $\hat{\delta}_m$, $\hat{\alpha}_m$, $\hat{\beta}_m$, and $\hat{\gamma}_m$) and weights ($\hat{w}_{i,m}$) along with the exogenous variables (c_t, $r_{i,t}$, $s_{i,t}$, and $T_{i,t}^{(air)}$), the hourly PV power can be calculated using the PV power models (see Eqs. 6.5 and 6.6). Figure 6.5 shows examples of hourly PV power forecasts using the proposed model and the benchmark model for a fortnight in May 2014. Although curve shapes of these forecasts were similar, the forecasts obtained using the proposed model were smoother than those provided by the benchmark model. The forecasts obtained using the proposed model for the type-I suppliers were relatively lower than those of the benchmark model on these specific days.

Since no true curve of hourly PV power was available, we compared the curve for the proposed model with that for the benchmark model, the shape of which was verified by utility experts to be very similar to the true curve in a preliminary experiment. With this visual curve-shape comparison, validation of the proposed model as an hourly PV power model was performed.

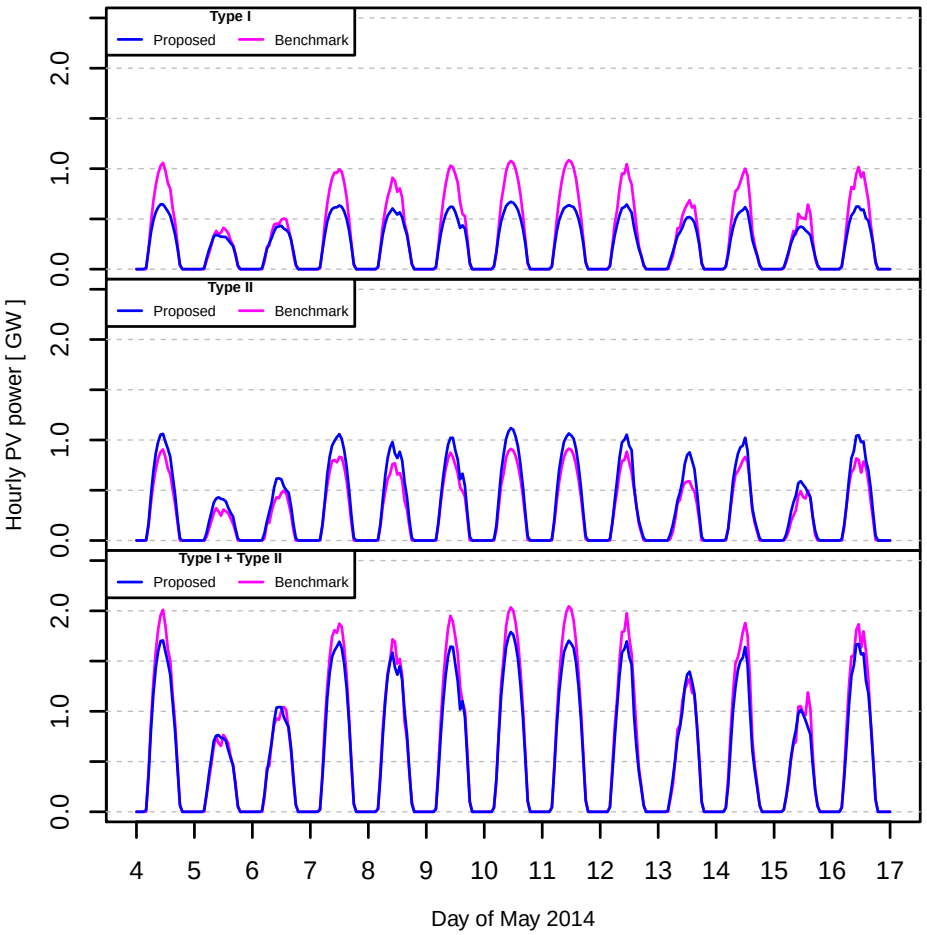

Fig. 6.5 Forecasts of hourly PV power. From top to bottom, the PV powers for type-I suppliers, type-II suppliers, and both type-I and type-II suppliers for the proposed and benchmark models.

6.3.5 Accuracy Evaluation

As a target purchase volume for accuracy evaluation, we used the total purchase volume of both types (i.e., type I + type II), since the total PV power is our primary interest in terms of load forecasting. Figures 6.6, 6.7, and 6.8 show the forecasts, forecasting errors, and percentage errors, respectively. Table 6.2 shows the $MAPE$ and $SD_{\{PE\}}$, and Table 6.3 shows the MAE, MBE, and $RMSE$. Table 6.4 shows statistical test results for accuracy improvement based on the Diebold-Mariano test.

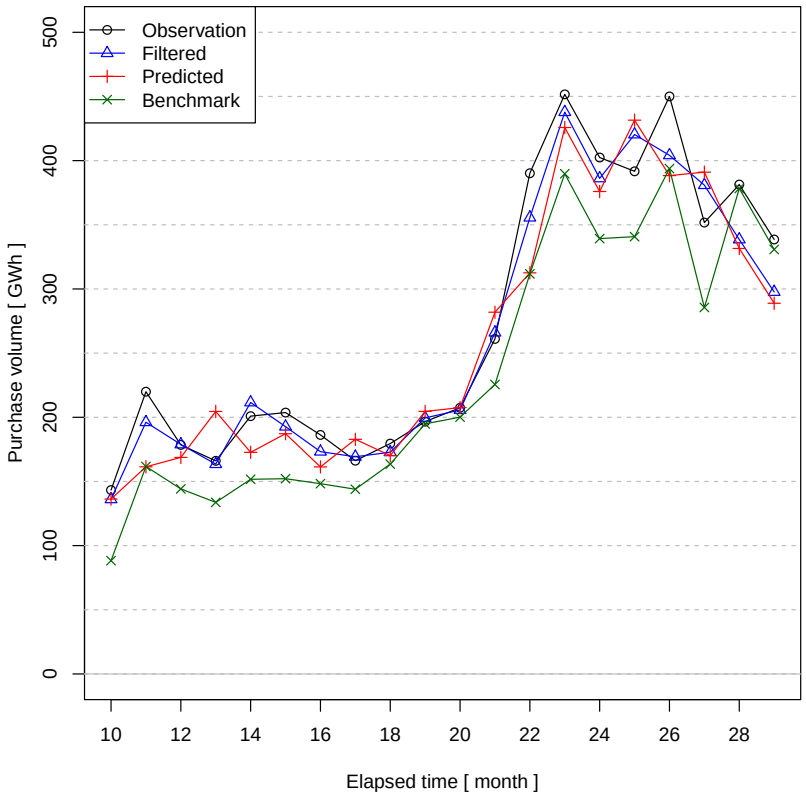

Fig. 6.6 Forecasts of total PV purchase volumes for 2013 and 2014. “Observation” indicates the reported PV purchase volumes (type I + type II), “Filtered” indicates the means of the filtering distribution, and “Predicted” indicates the means of the prediction distribution.

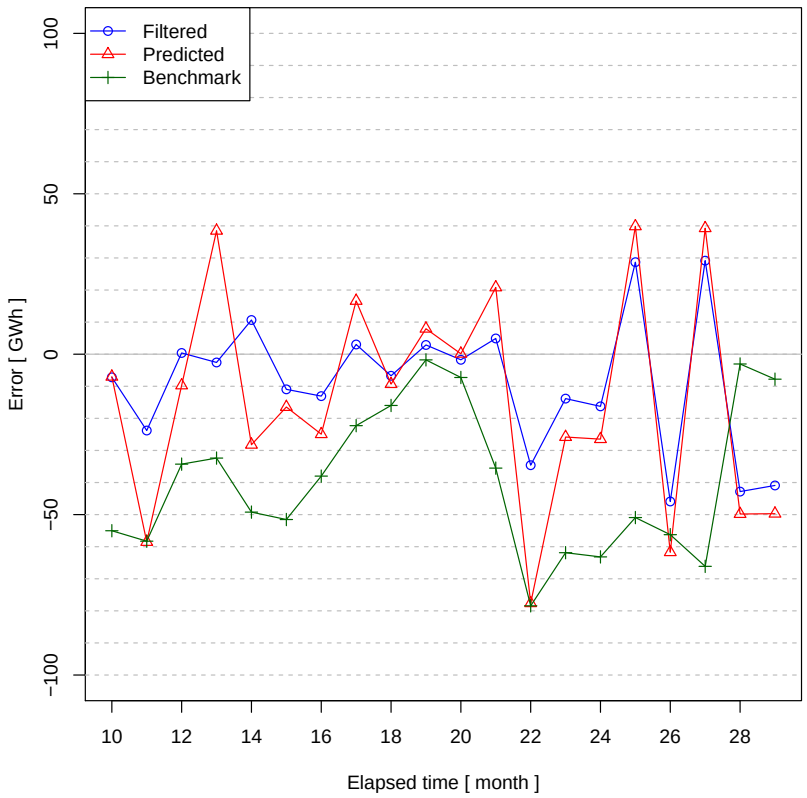

Fig. 6.7 Forecasting errors of total PV purchase volumes for 2013 and 2014

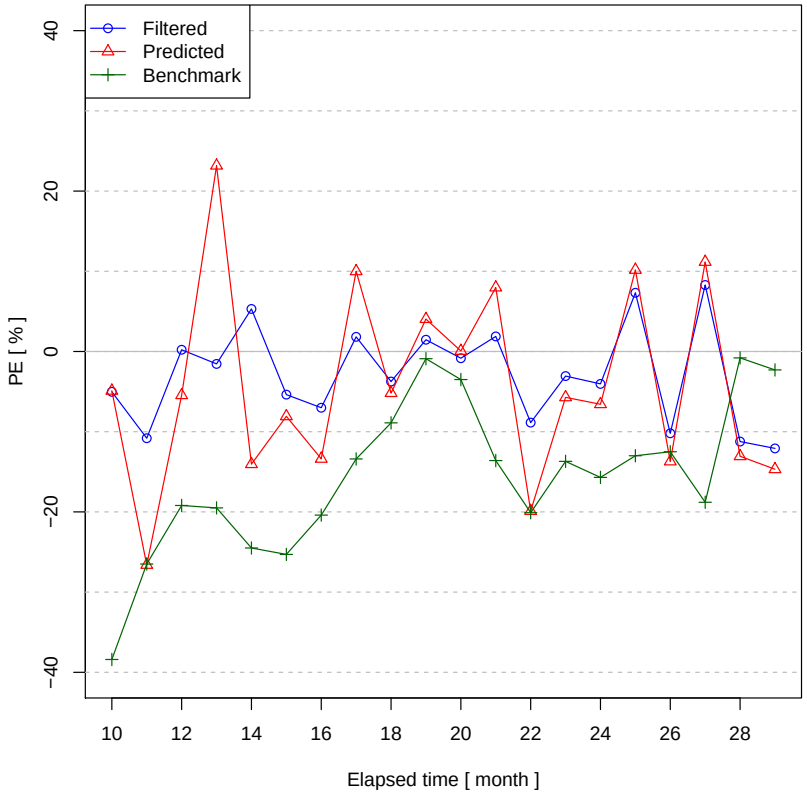

Fig. 6.8 Percentage errors of total PV purchase volumes for 2013 and 2014

Table 6.2 *MAPE*s and *SD*s for 2013 and 2014

Term (Year)	*MAPE* [%]			$SD_{\{PE\}}$ [%]		
	Filt. ($n = 20$)	Pred. ($n = 20$)	Bench. ($n = 20$)	Filt. ($n = 20$)	Pred. ($n = 20$)	Bench. ($n = 20$)
2013	4.54	12.32	21.79	4.90	14.35	8.43
2014	6.30	9.73	10.45	7.21	10.99	7.22
2013–2014	**5.51**	**10.90**	15.55	**6.12**	**12.27**	9.53

Note. 'Filt.' denotes filtering errors. 'Pred.' denotes forecasting errors. 'Bench.' denotes forecasting errors of the benchmark model

Table 6.3 *MAE*s, *MBE*s, and *RMSE*s for 2013 and 2014

Term (Year)	*MAE* [GWh]			*MBE* [GWh]			*RMSE* [GWh]		
	Filt. ($n = 20$)	Pred. ($n = 20$)	Bench. ($n = 20$)	Filt. ($n = 20$)	Pred. ($n = 20$)	Bench. ($n = 20$)	Filt. ($n = 20$)	Pred. ($n = 20$)	Bench. ($n = 20$)
2013	8.71	23.26	39.66	−5.58	−11.01	−39.66	10.97	28.09	42.06
2014	23.79	36.29	39.29	−11.86	−16.65	−39.29	28.56	42.41	48.14
2013–2014	17.00	30.43	39.46	**−9.03**	**−14.11**	−39.46	22.43	36.66	45.50

Table 6.4 Diebold-Mariano statistics for 2013 and 2014

	Pred.	Bench.
Filt.	-5.1 ***	-4.0 ***
Pred.		-1.7 .

Note. p-values based on the Diebold-Mariano test were coded with the significance codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Power of the loss function = 1.0. An alternative hypothesis is that the column method is less accurate than the row method.

Forecasting Accuracy and Bias

Regarding the overall accuracy (i.e., 2013 and 2014), the proposed model significantly outperformed the benchmark model, with the exception of $SD_{\{PE\}}$, as shown by the results in Tables 6.2, 6.3, and 6.4. Compared with the benchmark model, the forecasts obtained using the proposed model had high variability ($SD_{\{PE\}} = 12.27$). On this point, there is room for further improvement. In Table 6.3, the forecasting bias ($MBE = -14.11$) was drastically reduced compared with that of the benchmark model ($MBE = -39.46$). Thus, the severe bias problem was settled.

Regarding the filtering results, the proposed model had high accuracy ($MAPE = 5.51$), low bias ($MBE = -9.03$), and low variability ($SD_{\{PE\}} = 6.12$), which ensures that analyzed PV system parameters and weights are reliable. Based on the filtering results, the precise PV system parameters and weights can be obtained. Estimation of an entire utility service area has not been possible until this time.

Error Sources for PV Forecasting

Let us consider sources of forecasting error. For physics-based error sources, we have to consider the effects of solar irradiance, air temperature, wind speed, autoregressive components of PV, cloud cover, atmospheric pressure, conversion efficiency, installation angles and dust accumulation on PV panels [77]. In addition, the PV capacity forecasting errors, and the difference between observatory locations and dense installation areas of PV systems, may be major error sources. For human-based error sources, the PV purchase volume in a calendar month always includes the errors from two adjacent months. In other words, the reported volumes were not true volumes for a calendar month. This happens because a limited number of meter-reading staff members visit different areas in turn to measure the amount of generated PV power at houses.

Although it is very difficult to take into consideration all significant error sources in modeling, we have started to develop a more elaborate physics-based model by which to improve forecasting accuracy and PV efficiency analysis.

Chapter 7

Summary and Conclusions

In this thesis, we have proposed a novel modeling framework for load forecasting and analysis. The forecasting accuracy of each of the proposed models was compared with those of the existing state-of-the-art models. Two benchmarks and two utility operational models were implemented, and their accuracies were calculated under the same conditions. Although the proposed models are within the framework of SSMs, which usually underperform compared to black-box approaches, the proposed models significantly outperformed one of the utility models (by 0.7 points for all days); this demonstrates that the proposed models reached operational accuracy. Also, the proposed models were compared with the two sets of SSMs whose results were taken from the literature. Superiority over the existing SSMs was also demonstrated by the fact that the proposed models outperformed the existing SSMs by more than 0.6, 0.3, and 2.1 points at 9 AM, 12 AM, and in every hour, respectively, while providing more enriched load-analysis information than has been obtained with the existing SSMs.

In order to solve the emerging problem caused by the recent PV investment boom, we also proposed the PV power forecasting method in the same modeling framework as the load forecasting. The proposed method can enhance the hourly PV power model by incorporating the monthly PV purchase volume, which contributed to directly solving the problem confronting most utilities. The effectiveness of the proposed method was demonstrated experimentally. In fact, the proposed method significantly outperformed the physics-based model that is currently in operational use. On these grounds, we have come to the conclusion that the proposed method is one of the most promising approaches for forecasting short-term purchased PV power and PV efficiency analysis.

The combination of SSMs and the EnKF provides us with new and precise information about the load structure and the weather response. Therefore, it is no longer necessary for a utility to perform a retrospective analysis of the load behavior; this reduces the operator's workload and provides rational explanations for the low accuracy that sometimes occurs in practice. These sophisticated indicators will drastically reduce the cost when a utility must purchase energy from other utilities or must use pumped-storage hydropower.

The EnKF has been proven to be one of the most successful techniques for modeling meteorological and oceanographic phenomena. It is noteworthy that this is the first attempt to use the EnKF for load forecasting and analysis. It was demonstrated that the EnKF holds great promise in that area as well.

This thesis illustrated many modeling techniques that are based on the empirical knowledge of utility professionals; hence, we hope that it will also be of great help to those who are interested in load forecasting in academic fields. We expect that with our method, utility practitioners will be able to efficiently analyze the load structure and perform accurate daily forecasts.

References

[1] Aggarwal, S. and Saini, L. (2014). Solar energy prediction using linear and non-linear regularization models: A study on AMS (American Meteorological Society) 2013–14 Solar Energy Prediction Contest. *Energy*, 78:247–256.

[2] Almeida, M. P., Perpiñán, O., and Narvarte, L. (2015). PV power forecast using a nonparametric PV model. *Solar Energy*, 115:354–368.

[3] Alonso García, M. C. and Balenzategui, J. L. (2004). Estimation of photovoltaic module yearly temperature and performance based on Nominal Operation Cell Temperature calculations. *Renewable Energy*, 29(12):1997–2010.

[4] Anderson, J. L. (2001). An Ensemble Adjustment Kalman Filter for Data Assimilation. *Monthly Weather Review*, 129(12):2884–2903.

[5] Andrews, A. (1968). A square root formulation of the Kalman covariance equations. *AIAA Journal*, 6:1165–1166.

[6] Bahrami, S., Hooshmand, R.-A., and Parastegari, M. (2014). Short term electric load forecasting by wavelet transform and grey model improved by PSO (particle swarm optimization) algorithm. *Energy*, 72:434–442.

[7] Barkhouse, D. A. R., Gunawan, O., Gokmen, T., Todorov, T. K., and Mitzi, D. B. (2012). Device characteristics of a 10.1% hydrazine-processed Cu2ZnSn(Se,S)4 solar cell. *Progress in Photovoltaics: Research and Applications*, 20(1):6–11.

[8] Bierman (1977). *Factorization methods for discrete sequential estimation*, volume 128. Academic Press, New York.

[9] Bishop, C. H., Etherton, B. J., and Majumdar, S. J. (2001). Adaptive Sampling with the Ensemble Transform Kalman Filter. Part I: Theoretical Aspects. *Monthly Weather Review*, 129(3):420–436.

[10] Charlton, N. and Singleton, C. (2014). A refined parametric model for short term load forecasting. *International Journal of Forecasting*, 30(2):364–368.

[11] Chu, Y., Urquhart, B., Gohari, S. M., Pedro, H. T., Kleissl, J., and Coimbra, C. F. (2015). Short-term reforecasting of power output from a 48 MWe solar PV plant. *Solar Energy*, 112:68–77.

[12] Coimbra, C. F. M., Kleissl, J., and Marquez, R. (2013). *Overview of Solar-Forecasting Methods and a Metric for Accuracy Evaluation*. Elsevier.

[13] Doménech, R. and Gómez, V. (2006). Estimating Potential Output, Core Inflation, and the NAIRU as Latent Variables. *Journal of Business & Economic Statistics*, 24(3):354–365.

[14] Dordonnat, V., Koopman, S., Ooms, M., Dessertaine, A., and Collet, J. (2008). An hourly periodic state space model for modelling French national electricity load. *International Journal of Forecasting*, 24(4):566–587.

[15] Douc, R. and Cappe, O. (2005). Comparison of resampling schemes for particle filtering. *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pages 64–69.

[16] Eben, K., Juruš, P., Resler, J., Belda, M., Pelikán, E., Krüger, B. C., and Keder, J. (2005). An ensemble Kalman filter for short-term forecasting of tropospheric ozone concentrations. *Quarterly Journal of the Royal Meteorological Society*, 131(613):3313–3322.

[17] Erbs, D., Klein, S., and Duffie, J. (1982). Estimation of the diffuse radiation fraction for hourly, daily and monthly-average global radiation. *Solar Energy*, 28(4):293–302.

[18] Espinar, B., Aznarte, J. L., Girard, R., Moussa, a. M., and Kariniotakis, G. (2010). Photovoltaic Forecasting: A state of the art. *5th European PV-Hybrid and Mini-Gird Conference*, 33:1–6.

[19] Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics.

[20] Evensen, G. (2003). The Ensemble Kalman Filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, 53(4):343–367.

[21] Evensen, G. (2004). Sampling strategies and square root analysis schemes for the EnKF. *Ocean Dynamics*, 54(6):539–560.

[22] Friedman, J. H. (1984). A Variable Span Smoother. *Stanford University Technical Report*, 5(1984).

[23] Ghent, D., Kaduk, J., Remedios, J., Ardö, J., and Balzter, H. (2010). Assimilation of land-surface temperature into the land surface model JULES with an Ensemble Kalman Filter. *Journal of Geophysical Research*, 115(July):1–16.

[24] Ghofrani, M., Ghayekhloo, M., Arabali, A., and Ghayekhloo, A. (2015). A hybrid short-term load forecasting with a new input selection framework. *Energy*, 81:777–786.

[25] Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113.

[26] Haida, T. (2009). Study on Daily Electric Load Curve Forecasting Method based on Regression Type Hourly Load Modeling with Yearly Load Trends, Day-types and Insolations. *IEEJ Transactions on Power and Energy*, 129(12):1477–1485.

[27] Harvey, A. and Koopman, S. (1993). Forecasting Hourly Electricity Demand Using Time-Varying Splines. *Journal of the American Statistical Association*, 88(424):1228–1236.

[28] Harvey, A. and Shephard, N. (1993). Structural time series models. In *Handbook of Statistics*, volume 11, pages 261–302. Elsevier B.V.

[29] Hidaka, K., Sinoda, Y., and Okamoto, T. (2014). A Study on Wide Area Output Estimation of Photovoltaic Systems. *The Institute of Electrical Engineers of Japan*, 134(6):477–483.

[30] Hong, T. (2010). *Short Term Electric Load Forecasting*. PhD thesis, North Carolina State University.

[31] Hyndman, R. J. and Khandakar, Y. (2008). Automatic time series forecasting : the forecast package for R Automatic time series forecasting : the forecast package for R. *Journal Of Statistical Software*, 27(3):1–22.

[32] Iga, A. and Ishihara, Y. (2006). Characteristics and Embodiment of the Practical Use Method of "Monthly Temperature Coefficient" of the Photovoltaic Generation System. *IEEJ Transactions on Power and Energy*, 126(8):767–775.

[33] Imamura, E. and Asano, H. (2009). Load Survey Analysis of Hot Water and Electricity Demands of Residential Sector in the Metropolitan Area - Analysis of Sensitivity to Temperature of Energy Demand and Load Profile -. Technical report, CRIEPI.

[34] Ishimura, A. (2012). Japan Energy Brief. *Power*, pages 1–13.

[35] Joukov, V., Bonnet, V., Venture, G., and Kuli, D. (2015). Constrained Dynamic Parameter Estimation using the Extended Kalman Filter. pages 3654–3659.

[36] Julier, S. J. and Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. *Int Symp AerospaceDefense Sensing Simul and Controls*, 3:182–193.

[37] Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME-Journal of Basic Engineering*, 82(Series D):35–45.

[38] Kaur, A., Nonnenmacher, L., Pedro, H. T., and Coimbra, C. F. (2016). Benefits of solar forecasting for energy imbalance markets. *Renewable Energy*, 86:819–830.

[39] Kavaklioglu, K. (2014). Robust electricity consumption modeling of Turkey using Singular Value Decomposition. *International Journal of Electrical Power and Energy Systems*, 54:268–276.

[40] Khotanzad, A., Afkhami-Rohani, R., and Maratukulam, D. (1998). ANNSTLF - Artificial Neural Network Short Term Load Forecaster Generation Three. *IEEE Transactions on Power Systems*, 13(4):1413–1422.

[41] King, D., Boyson, W., and Kratochvill, J. (2004). Photovoltaic Array Performance Model. *Sandia Report*, (December).

[42] Kitagawa, G. (1993). A Monte Carlo Filtering and Smoothing Method for Non-Gaussian Nonlinear State Space Models. *Proceedings of the 2nd U.S.-Japan Joint Seminar on Statistical Time Series Analysis*, pages 110–131.

[43] Kitagawa, G. (1996). Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics*, 5(1):1–25.

[44] Kitagawa, G. and Gersch, W. (1984). A Smoothness Priors-State Space Modeling of Time Series with Trend and Seasonality. *Journal of the American Statistical Association*, 79(386):378.

[45] Ko, C.-N. and Lee, C.-M. (2013). Short-term load forecasting using SVR (support vector regression)-based radial basis function neural network with dual extended Kalman filter. *Energy*, 49:413–422.

[46] Lark, M. A. P. C., Slater, A. G., and Clark, M. P. (2006). Snow Data Assimilation via an Ensemble Kalman Filter. *Journal of Hydrometeorology*, 7(3):478–493.

[47] Lauret, P., David, M., and Calogine, D. (2012). Nonlinear models for short-time load forecasting. *Energy Procedia*, 14:1404–1409.

[48] Lo Brano, V., Ciulla, G., and Di Falco, M. (2014). Artificial Neural Networks to Predict the Power Output of a PV Panel. *International Journal of Photoenergy*, 2014:12.

[49] McElhoe, B. (1966). An assessment of the navigation and course corrections for a manned flyby of Mars or Venus. ... *and Electronic Systems, IEEE Transactions on*, 2(4):613–623.

[50] Ohtsuka, Y., Oga, T., and Kakamu, K. (2010). Forecasting electricity demand in Japan: A Bayesian spatial autoregressive ARMA approach. *Computational Statistics and Data Analysis*, 54(11):2721–2735.

[51] Otsuka, A. and Nakano, K. (2014). Structural Analysis and Simulation of Residential Electric Demand in Japan - Empirical Analysis by using 47 Prefectural Data -. Technical report, CRIEPI.

[52] Park, T. and Casella, G. (2008). The Bayesian Lasso. *Journal of the American Statistical Association*, 103(482):681–686.

[53] Pedregal, D. J. and Trapero, J. R. (2010). Mid-term hourly electricity forecasting based on a multi-rate approach. *Energy Conversion and Management*, 51(1):105–111.

[54] Pedro, H. T. C. and Coimbra, C. F. M. (2012). Assessment of forecasting techniques for solar power production with no exogenous inputs. *Solar Energy*, 86(7):2017–2028.

[55] Perez, R., Seals, R., Ineichen, P., Stewart, R., and Menicucci, D. (1987). A new simplified version of the perez diffuse irradiance model for tilted surfaces. *Solar Energy*, 39(3):221–231.

[56] Phinikarides, A., Makrides, G., Zinsser, B., Schubert, M., and Georghiou, G. E. (2015). Analysis of photovoltaic system performance time series: Seasonality and performance loss. *Renewable Energy*, 77:51–63.

[57] Pielow, A., Sioshansi, R., and Roberts, M. C. (2012). Modeling short-run electricity demand with long-term growth rates and consumer price elasticity in commercial and industrial sectors. *Energy*, 46(1):533–540.

[58] Ramanathan, R., Engle, R., Granger, C. W., Vahid-Araghi, F., and Brace, C. (1997). Shorte-run forecasts of electricity loads and peaks. *International Journal of Forecasting*, 13(2):161–174.

[59] Sadaei, H. J., Enayatifar, R., Abdullah, A. H., and Gani, A. (2014). Short-term load forecasting using a hybrid model with a refined exponentially weighted fuzzy time series and an improved harmony search. *International Journal of Electrical Power & Energy Systems*, 62:118–129.

[60] Skoplaki, E. and Palyvos, J. a. (2009). On the temperature dependence of photovoltaic module electrical performance: A review of efficiency/power correlations. *Solar Energy*, 83(5):614–624.

[61] Smith, G. L., McGee, L. A., and Schmidt, S. F. (1962). Application of Statistical Filter Theory to the Optimal Estimation of Position and Velocity on Board a Circumlunar Vehicle.

[62] Taylor, J. W. (2003). Using Weather Ensemble Predictions in Electricity Demand Forecasting Using Weather Ensemble Predictions in Electricity Demand Forecasting. *International Journal of Forecasting*, 19:57–70.

[63] Taylor, J. W. (2012). Short-term load forecasting with exponentially weighted methods. *IEEE Transactions on Power Systems*, 27(1):458–464.

[64] Taylor, J. W. and Mcsharry, P. E. (2008). Short-Term Load Forecasting Methods : An Evaluation Based on European Data. *IEEE Transactions on Power Systems*, 22:2213–2219.

[65] TEPCO (2014). Wheeling Service provisions (in Japanese).

[66] Thornton, C. L. (1976). Triangular covariance factorizations for Kalman filtering. *October*, 30:4841–9.

[67] Thrun, S. (2002). Probabilistic robotics. *Communications of the ACM*, 45(3):1999–2000.

[68] Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso Robert Tibshirani. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

[69] Tippett, M. K., Anderson, J. L., Bishop, C. H., Hamill, T. M., and Whitaker, J. S. (2003). Ensemble Square Root Filters. *Monthly Weather Review*, 131(7):1485–1490.

[70] Ueno, G. (2011). *Implimentaion of Ensemble Kalman Filters (in Japanese)*. Institute of Statistical Mathematics, Japan, Tokyo.

[71] Ueno, G., Higuchi, T., Kagimoto, T., and Hirose, N. (2010). Maximum likelihood estimation of error covariances in ensemble-based filters and its application to a coupled atmosphere-ocean model. *Quarterly Journal of the Royal Meteorological Society*, 136(650):1316–1343.

[72] Vazquez, M. A. O. (2006). *Optimizing the Spinning Reserve Requirements*. PhD thesis.

[73] Wang, C.-h., Grozev, G., and Seo, S. (2012). Decomposition and statistical analysis for regional electricity demand forecasting. *Energy*, 41(1):313–325.

[74] Wang, F., Zhen, Z., Mi, Z., Sun, H., Su, S., and Yang, G. (2015). Solar irradiance feature extraction and support vector machines based weather status pattern recognition model for short-term photovoltaic power forecasting. *Energy and Buildings*, 86:427–438.

[75] Whitaker, J. S. and Hamill, T. M. (2002). Ensemble Data Assimilation without Perturbed Observations. *Monthly Weather Review*, 130(7):1913–1924.

[76] Yona, A., Senjyu, T., Funabashi, T., Mandal, P., and Kim, C.-h. (2013). Decision Technique of Solar Radiation Prediction Applying Recurrent Neural Network for Short-Term Ahead Power Output of Photovoltaic System. 2013(September):32–38.

[77] Yuan-Kang, Wu; Chao-Rong, Chen; Hasimah Abdul, R. (2014). A Novel Hybrid Model for Short-Term Forecasting in PV Power Generation. *International Journal of Photoenergy*, 2014.

[78] Zamo, M., Mestre, O., Arbogast, P., and Pannekoucke, O. (2014). A benchmark of statistical regression methods for short-term forecasting of photovoltaic electricity production, part I: Deterministic forecast of hourly production. *Solar Energy*, 105:792–803.

[79] Zou, J. and Navon, I. (1996). The linearization and adjoint of radiation transfer processes in the NMC spectral model part I: solar radiative transfer. *Meteorology and Atmospheric Physics*, 58:193–203.

This thesis is based on the following paper.

Takeda, H., Tamura, Y., and Sato, S. (2016). Using the ensemble Kalman filter for electricity load forecasting and analysis. *Energy*, 104:184–198.

Appendix A

Implementation

A.1 Fortran Sample Code

```
!-----------------------------------------------------------------------------------------
! Ensemble Filters
!-----------------------------------------------------------------------------------------
! Input
!       Xp   : Prediction distribution matrix
!       H    : Observation matrix
!       R    : Observation noise covariance (assumed diagonal matrix)
!       y_t  : Observations at time t
!       N_Y  : Number of observed time series
!       N_KX : State vector size
!       N_M  : Ensemble size
!         t  : time t [hours] (used only for seeds of PF-resampling)
!
! Output
!       Xf : Filtering distribution matrix
!-----------------------------------------------------------------------------------------
! Copyright(C) 2016 Hisashi Takeda. All Rights Reserved.
!-----------------------------------------------------------------------------------------
! NB: Use Intel@Math Kernel Library for matrix operations
! NB: Use compiler's automatic zero-initialization for all of variables, arrays and matrices

module Ensemble_Filters

    use f95_precision, only: wp => dp
    use blas95,        only: gemm
    use lapack95,      only: potrf, potri, trtri, syevd, gesvd
    use GenRVs_mo
    implicit none
    contains

    !-----------------------------------------------------------------
    ! Ensemble Kalman Filter with Perturbed Observations (EnKF/PO)
    !-----------------------------------------------------------------
    subroutine EnKF_PO (Xp, Xf, H, R, y_t, N_Y, N_KX, N_M)

        integer,  intent(in)             :: N_Y, N_KX, N_M
        real(wp), intent(in)             :: Xp(N_KX, N_M), H(N_Y, N_KX), R(N_Y, N_Y), y_t(N_Y)
        real(wp), intent(inout)          :: Xf(N_KX, N_M)
        real(wp), dimension (N_KX, N_M)  :: mXp, dXp, K_INOV
        real(wp), dimension (N_KX, N_KX) :: Vp
        real(wp), dimension (N_KX, N_Y)  :: VpHt, K
        real(wp), dimension (N_M, N_M)   :: A1
        real(wp), dimension (N_Y, N_M)   :: Y, Yp, W0, W
        real(wp), dimension (N_Y, N_Y)   :: HVpHt, U, Ainv
        real(wp)                         :: Ninv, Ninv_ub
        integer                          :: i_y

        do i_y = 1, N_Y
             Y(i_y, :) = y_t(i_y)
            W0(i_y, :) = GenNorm (N_M, 0._wp, sqrt(R(i_y, i_y)), i_y)
```

```
51          end do
52
53          Ninv    = 1._wp / dble(N_M)
54          Ninv_ub = 1._wp / (dble(N_M) - 1._wp)
55
56          W = W0
57          call gemm (W0, A1,  W, 'n', 'n', -Ninv, 1._wp) ! Mean adjusted W
58
59          A1  = 1._wp
60          call gemm (Xp, A1, mXp, 'n', 'n', Ninv, 0._wp) ! mXp = Xp * A1 / N_M
61
62          dXp = Xp - mXp
63
64          call gemm (dXp, dXp,    Vp, 'n', 't', Ninv_u, 0._wp) ! Vp = 1/(N-1) * dXp * dXp'
65          call gemm (Vp,    H,  VpHt, 'n', 't') !
66          call gemm (H,  VpHt, HVpHt, 'n', 'n') !
67
68          ! Cholesky decomposition, UU' is positive definite
69          U = HVpHt + R ! = A = UU'
70          call potrf (U, 'U')
71          Ainv = U
72          call potri (Ainv, 'U') ! [HVpH'+R]^{-1}
73
74          call gemm (VpHt,    Ainv,      K, 'n', 'n') ! K      = VpH'[HVpH'+R]^{-1}
75          call gemm (H,         Xp,     Yp, 'n', 'n') ! Yp     = H * Xp
76          call gemm (K, Y + W - Yp, K_INOV, 'n', 'n') ! K_INOV = K * (Y + W - Yp)
77
78          Xf = Xp + K_INOV
79
80      end subroutine EnKF_PO
81
82      !-----------------------------------------------------
83      ! Ensemble Adjustment Kalman Filter (EAKF)
84      !-----------------------------------------------------
85      subroutine EAKF (Xp, Xf, H, R, y_t, N_Y, N_KX, N_M)
86
87          real(wp), parameter                   :: EPS = 1._wp-3
88          integer,  intent(in)                  :: N_Y, N_KX, N_M
89          real(wp), intent(in)                  :: Xp(N_KX, N_M), H(N_Y, N_KX), R(N_Y, N_Y), y_t(N_Y)
90          real(wp), intent(inout)               :: Xf(N_KX, N_M)
91          real(wp), dimension (N_KX, N_M)       :: mXp, mXf, dXp, dXf, U0, UGVt
92          real(wp), dimension (N_M, N_M)        :: A1
93          real(wp), dimension (N_Y, N_M)        :: Y, mYp
94          real(wp), dimension (N_Y, N_Y)        :: Rinv
95          real(wp), dimension (N_KX, N_Y)       :: mK
96          real(wp), dimension (N_KX, N_KX)      :: A
97          real(wp), allocatable, dimension(:, :) :: G, Ginv, GinvUt, U, UG, HUG
98          real(wp), allocatable, dimension(:, :) :: Z, UGZ, I_Binv, I_Binvsq
99          real(wp), allocatable, dimension(:, :) :: UGZ_I_Binv, UGZ_I_Binvsq, UGZ_I_Binv_Zt
100         real(wp), allocatable, dimension(:, :) :: GtUtHtRinv, ZBZt
101         real(wp), allocatable, dimension(:)    :: sigma, lambda
102         real(wp)                               :: sigma_sum
103         integer                                :: i_m, i, n_min, N_R, i_N_R
```

```
104
105         do concurrent (i_m = 1:N_M)
106             Y(:, i_m) = y_t
107         end do
108
109         A1  = 1._wp
110         call gemm (Xp, A1, mXp, 'n', 'n', 1._wp / dble(N_M), 0._wp) ! mXp = Xp * A1 / N_M
111
112         mXf = mXp
113         dXp = Xp - mXp
114
115         n_min = min(N_KX, N_M)
116         allocate ( sigma(n_min) )
117
118         UGVt = dXp
119         call gesvd (UGVt, sigma, U0) ! Singular value decomposition
120
121         sigma_sum = sum(sigma)
122         do i = 1, n_min
123             if (sigma(i) / sigma_sum > EPS) N_R = N_R + 1 ! To be compact SVD
124         end do
125
126         allocate( G(N_R, N_R), Ginv(N_R, N_R), U(N_KX, N_R), UG(N_KX, N_R), HUG(N_Y, N_R) )
127         allocate( GtUtHtRinv(N_R, N_Y), ZBZt(N_R, N_R), Z(N_R, N_R), lambda(N_R) )
128
129         U = U0(1:N_KX, 1:N_R)
130
131         do concurrent (i = 1:N_R)
132             G   (i, i) = sigma(i)         ! Diagonal matrix with singular value entries
133             Ginv(i, i) = 1._wp / sigma(i) ! Moor-Penrose pseudoinverse
134         end do
135
136         Rinv = 1._wp / ((dble(N_M) - 1._wp) * R) ! [(N-1)R]^{-1}
137         call gemm (U,            G,         UG, 'n', 'n') ! UG
138         call gemm (H,           UG,        HUG, 'n', 'n') ! HUG
139         call gemm (HUG,       Rinv, GtUtHtRinv, 't', 'n') ! (HUG)'[(N-1)R]^{-1}
140         call gemm (GtUtHtRinv, HUG,       ZBZt, 'n', 'n') ! ZBZ' = (HUG)'[(N-1)R]^{-1}HUG
141
142         ! Eigendecomposition
143         Z = ZBZt
144         call syevd (Z, lambda, 'V')
145
146         allocate ( I_Binv(N_R, N_R), I_Binvsq(N_R, N_R) )
147
148         ! B: Diagonal matrix with eigenvalue entries(Lambda)
149         do concurrent (i_N_R = 1:N_R)
150             I_Binv  (i_N_R, i_N_R) = 1._wp / (1._wp + lambda(i_N_R))     ! [I+B]^{-1}
151             I_Binvsq(i_N_R, i_N_R) = 1._wp / sqrt(1._wp + lambda(i_N_R)) ! [I+B]^{-1/2}
152         end do
153
154         allocate ( UGZ(N_KX, N_R), UGZ_I_Binv(N_KX, N_R), UGZ_I_Binvsq(N_KX, N_R), GinvUt(N_R, N_KX) )
155         allocate ( UGZ_I_Binv_Zt(N_KX, N_R) )
156
```

```
157         call gemm (UG,                     Z,           UGZ, 'n', 'n') ! UGZ
158         call gemm (UGZ,              I_Binv,    UGZ_I_Binv, 'n', 'n') ! UGZ[I+B]^{-1}
159         call gemm (UGZ_I_Binv,           Z, UGZ_I_Binv_Zt, 'n', 't') ! UGZ[I+B]^{-1}Z'
160         call gemm (UGZ_I_Binv_Zt, GtUtHtRinv,           mK, 'n', 'n') ! mK
161         call gemm (UGZ,           I_Binvsq,  UGZ_I_Binvsq, 'n', 'n') ! UGZ[I+B]^{-1/2}
162         call gemm (Ginv,                   U,        GinvUt, 'n', 't') ! G^{+}U'
163         call gemm (UGZ_I_Binvsq,      GinvUt,             A, 'n', 'n')               ! A
164         call gemm (H,                    mXp,           mYp, 'n', 'n')               ! mYp
165         call gemm (mK,                 Y - mYp,           mXf, 'n', 'n', 1._wp, 1._wp) ! mXf
166         call gemm (A,                    dXp,           dXf, 'n', 'n')               ! dXf
167
168         Xf = mXf + dXf
169
170     end subroutine EAKF
171
172     !---------------------------------------------------------------
173     ! Ensemble Square-Root Filter (EnSRF)
174     !---------------------------------------------------------------
175     subroutine EnSRF (Xp, Xf, H, R, y_t, N_Y, N_KX, N_M)
176
177         integer,  intent(in)             :: N_Y, N_KX, N_M
178         real(wp), intent(in)             :: Xp(N_KX, N_M), H(N_Y, N_KX), R(N_Y, N_Y), y_t(N_Y)
179         real(wp), intent(inout)          :: Xf(N_KX, N_M)
180         real(wp), dimension (N_KX, N_M)  :: mXp, mXf, dXp, dXf
181         real(wp), dimension (N_M, N_M)   :: A1
182         real(wp), dimension (N_KX, N_KX) :: dK_H
183         real(wp), dimension (N_Y, N_M)   :: mYp, Y
184         real(wp), dimension (N_M, N_Y)   :: dXpt_Ht
185         real(wp), dimension (N_KX, N_Y)  :: dXp_dXpt_Ht, dXp_dXpt_Ht_Utinv, mK, dK
186         real(wp), dimension (N_Y, N_Y)   :: UUt, U, Uinv, Utinv, U_Vinv, V
187         integer i, j
188
189         do concurrent (j = 1:N_M)
190             Y(:, j) = y_t
191         end do
192
193         A1  = 1._wp
194         call gemm (Xp, A1, mXp, 'n', 'n', 1._wp / dble(N_M), 0._wp) ! mXp = Xp * A1 / Nm
195
196         dXp = Xp - mXp
197         mXf = mXp
198         dXf = dXp
199
200         call gemm (dXp, H, dXpt_Ht, 't', 't')
201         call gemm (dXp, dXpt_Ht, dXp_dXpt_Ht)
202
203         ! Cholesky decomposition for a diagonal matrix (N-1)R
204         do i = 1, N_Y
205             V(i, i) = sqrt(dble(N_M - 1) * R(i, i))
206         end do
207
208         UUt = dble(N_M - 1) * R
209         call gemm (H, dXp_dXpt_Ht, UUt, 'n', 'n', 1._wp, 1._wp)
```

```
210
211         ! Cholesky decomposition, UU' is positive definite
212         U = UUt
213         call potrf (U, 'U')
214
215         ! Set 0 in the lower triangle
216         do j = 1, (N_Y - 1)
217             U((j + 1):N_Y, j) = 0._wp
218         end do
219
220         Utinv = transpose(U)
221         call trtri (Utinv, 'L')! U'^{-1}
222
223         Uinv = U
224         call trtri (Uinv, 'U')! U^{-1}
225
226         U_Vinv = U + V
227         call trtri (U_Vinv, 'U') ! (U + V)^{-1}
228
229         call gemm (dXp_dXpt_Ht, Utinv, dXp_dXpt_Ht_Utinv)
230         call gemm (dXp_dXpt_Ht_Utinv, Uinv,   mK)              ! mK
231         call gemm (dXp_dXpt_Ht_Utinv, U_Vinv, dK)              ! dK
232         call gemm (H, mXp, mYp)                                ! mYp = H * mXp
233         call gemm (mK, Y - mYp,  mXf, 'n', 'n',  1._wp, 1._wp) ! mXf = mXp + mK * (Y - mYp)
234         call gemm (dK,       H, dK_H)                          ! dK * H
235         call gemm (dK_H,   dXp,  dXf, 'n', 'n', -1._wp, 1._wp) ! dXf = dXp - dK * H * dXp
236
237         Xf = mXf + dXf
238
239     end subroutine EnSRF
240
241     !---------------------------------------------------------------
242     ! EnSRF for Univariate Time Series
243     !---------------------------------------------------------------
244     subroutine EnSRF_uni(Xp, Xf, H, R, y_t, N_K, N_M)
245
246         integer,  intent(in)                         :: N_M, N_K
247         real(wp), intent(in)                         :: y_t, R
248         real(wp), intent(in),    dimension (  1, N_K) :: H
249         real(wp), intent(in),    dimension (N_K, N_M) :: Xp
250         real(wp), intent(inout), dimension (N_K, N_M) :: Xf
251         real(wp),                dimension (N_K, N_M) :: mXp, mXf, dXp, dXf
252         real(wp),                dimension (N_M, N_M) :: A1
253         real(wp),                dimension (N_K, N_K) :: dK_H
254         real(wp),                dimension (  1, N_M) :: mYp
255         real(wp),                dimension (N_M, 1)   :: dXpt_Ht
256         real(wp),                dimension (N_K, 1)   :: dXp_dXpt_Ht, mK, dK
257         real(wp),                dimension (  1, 1)   :: UUt
258         real(wp)                                      :: U, V
259
260         A1  = 1._wp
261         call gemm(Xp, A1, mXp, 'n', 'n', 1._wp / dble(N_M), 0._wp)
262
```

```
263         dXp = Xp - mXp
264         mXf = mXp
265         dXf = dXp
266
267         call gemm(dXp, H, dXpt_Ht,           't', 't')
268         call gemm(dXp, dXpt_Ht, dXp_dXpt_Ht, 'n', 'n')
269
270         UUt = dble(N_M - 1) * R
271         call gemm(H, dXp_dXpt_Ht, UUt, 'n', 'n', 1._wp, 1._wp)
272
273         U = sqrt(UUt(1, 1))
274         V = sqrt(dble(N_M - 1) * R)
275
276         mK = dXp_dXpt_Ht / UUt(1, 1)
277         dK = dXp_dXpt_Ht / U / (U + V)
278
279         call gemm(H, mXp, mYp)                                  ! mYp = H * mXp
280         call gemm(mK, y_t - mYp,  mXf, 'n', 'n', 1._wp, 1._wp)  ! mXf = mXp + mK * (y_t - mYp)
281         call gemm(dK,         H, dK_H)                          ! dK_H = dK * H
282         call gemm(dK_H,     dXp,  dXf, 'n', 'n', -1._wp, 1._wp) ! dXf = dXp - dK * H * dXp
283
284         Xf = mXf + dXf
285
286     end subroutine EnSRF_uni
287
288     !-------------------------------------------------------------
289     ! Ensemble Kalman Filter with Square-Root Algorithm (EnKF/SR)
290     !-------------------------------------------------------------
291     subroutine EnKF_SR (Xp, Xf, H, R, y_t, N_Y, N_KX, N_M)
292
293         integer,  intent(in)            :: N_Y, N_KX, N_M
294         real(wp), intent(in)            :: Xp(N_KX, N_M), H(N_Y, N_KX), R(N_Y, N_Y), y_t(N_Y)
295         real(wp), intent(inout)         :: Xf(N_KX, N_M)
296         real(wp), dimension (N_KX, N_M) :: mXp, mXf, dXp, dXf
297         real(wp), dimension (N_M, N_M)  :: A1, Vt, I_StS, I_StS_Vt, dT
298         real(wp), dimension (N_Y, N_M)  :: Y, mYp, dYp, UGVt, G
299         real(wp), dimension (N_Y, N_Y)  :: ZBZt, Z, Binv, Binv_Zt, Z_Binv_Zt, Binvsq, Binvsq_Zt, U
300         real(wp), dimension (N_KX, N_Y) :: mK, dXpdYpt
301         real(wp)                        :: sigma(N_KX), lambda(N_Y), sigma_sum
302         integer                         :: i_y, i_m, i, n_min
303
304         do concurrent (i_m = 1:N_M)
305             Y(:, i_m) = y_t
306             I_StS(i_m, i_m) = 1._wp ! Identity matrix
307         end do
308
309         A1  = 1._wp
310         call gemm (Xp, A1, mXp, 'n', 'n', 1._wp / dble(N_M), 0._wp) ! mXp = Xp * A1 / N_M
311
312         mXf = mXp
313         dXp = Xp - mXp
314
315         call gemm (H, dXp, dYp, 'n', 'n') ! dYp = HdXp
```

```
316
317         ZBZt = (dble(N_M) - 1._wp) * R
318         call gemm (dYp, dYp, ZBZt, 'n', 't', 1._wp, 1._wp) ! ZSZ' : = dYpdYp' + (N-1)R
319
320         Z = ZBZt
321         call syevd (Z, lambda, 'V') ! Eigendecomposition
322
323         ! B: Diagonal matrix with eigenvalue entries(Lambda)
324         do i_y = 1, N_Y
325             Binv  (i_y, i_y) = 1._wp / lambda(i_y)       ! Lambda^{-1}
326             Binvsq(i_y, i_y) = 1._wp / sqrt(lambda(i_y)) ! Lambda^{-1/2}
327         end do
328
329         call gemm (Binv,        Z,   Binv_Zt, 'n', 't') ! Lambda^{-1} * Z'
330         call gemm (Z,     Binv_Zt, Z_Binv_Zt, 'n', 'n') ! Z * Lambda^{-1} * Z'
331         call gemm (Binvsq,      Z, Binvsq_Zt, 'n', 't') ! Lambda^{-1/2} * Z'
332         call gemm (Binvsq_Zt, dYp,      UGVt, 'n', 'n') ! UGVt = Lambda^{-1/2} * Z'dYp
333
334         call gesvd (UGVt, sigma, U, Vt)! Singular value decomposition
335
336         n_min = min(N_Y, N_M)
337         sigma_sum = sum(sigma)
338         do concurrent (i = 1:n_min)
339             G(i, i) = sigma(i) ! Diagonal matrix with singular value entries
340         end do
341
342         call gemm (Xp,            dYp,  dXpdYpt, 'n', 't') ! dXpdYp'
343         call gemm (dXpdYpt, Z_Binv_Zt,       mK, 'n', 't') ! mK  = dXpdYp'Z*Lambda^{-1}*Z'
344         call gemm (H,             mXp,      mYp, 'n', 'n') ! mYp = H * mXp
345         call gemm (mK,        Y - mYp,      mXf, 'n', 'n',  1._wp, 1._wp) ! mXf = mXp + mK*(Y - mYp)
346         call gemm (G,               G,    I_StS, 't', 'n', -1._wp, 1._wp) ! I - G'G
347         call gemm (sqrt(I_StS),    Vt, I_StS_Vt, 'n', 'n') ! sqrt(I - G'G) * V'
348         call gemm (Vt,       I_StS_Vt,       dT, 't', 'n') ! dT  = V * sqrt(I - S'S) * V'
349         call gemm (dXp,           dT,      dXf, 'n', 'n') ! dXf = dXp * dT
350
351         Xf = mXf + dXf
352
353     end subroutine EnKF_SR
354
355     !------------------------------------------------
356     ! Ensemble Transform Kalman Filter (ETKF)
357     !------------------------------------------------
358     subroutine ETKF (Xp, Xf, H, R, y_t, N_Y, N_KX, N_M)
359
360         integer,  intent(in)            :: N_Y, N_KX, N_M
361         real(wp), intent(in)            :: Xp(N_KX, N_M), H(N_Y, N_KX), R(N_Y, N_Y), y_t(N_Y)
362         real(wp), intent(inout)         :: Xf(N_KX, N_M)
363         real(wp), dimension (N_KX, N_M) :: mXp, mXf, dXp, dXf, dXp_dS
364         real(wp), dimension (N_M, N_M)  :: A1, I_MM, Z, Binv, Binvsq, Binv_Zt, Binvsq_Zt, dS, dT, ZBZt
365         real(wp), dimension (N_Y, N_M)  :: Y, mYp, dYp
366         real(wp), dimension (N_M, N_Y)  :: dYpt_Rinv
367         real(wp)                        :: Rinv(N_Y, N_Y), lambda(N_M), mK(N_KX, N_Y)
368         integer                         :: i_m
```

```
369
370         do concurrent (i_m = 1:N_M)
371             Y(:, i_m) = y_t
372             I_MM(i_m, i_m) = 1._wp ! Identity matrix of rank N_M
373         end do
374
375         A1  = 1._wp
376         call gemm (Xp, A1, mXp, 'n', 'n', 1._wp / dble(N_M), 0._wp) ! mXp = Xp * A1 / N_M
377
378         mXf = mXp
379         dXp = Xp - mXp
380
381         Rinv = 1._wp / ((dble(N_M) - 1._wp) * R)   ! [(N-1)R]^{-1}
382         call gemm (H,    mXp,       mYp, 'n', 'n') ! mYp = H * mXp
383         call gemm (H,    dXp,       dYp, 'n', 'n') ! dYp = H * dXp
384         call gemm (dYp, Rinv, dYpt_Rinv, 't', 'n') ! dYp'*[(N-1)R]^{-1}
385
386         ZBZt = I_MM
387         call gemm (dYpt_Rinv, dYp, ZBZt, 'n', 'n', 1._wp, 1._wp)
388
389         ! Eigendecomposition
390         Z = ZBZt
391         call syevd (Z, lambda, 'v')
392
393         ! B: Diagonal matrix with eigenvalue entries(Lambda)
394         do concurrent (i_m = 1:N_M)
395             Binv  (i_m, i_m) = 1._wp / lambda(i_m)       ! Lambda^{-1}
396             Binvsq(i_m, i_m) = 1._wp / sqrt(lambda(i_m)) ! Lambda^{-1/2}
397         end do
398
399         call gemm (Binv,           Z,   Binv_Zt, 'n', 't') ! Lambda^{-1} * Z'
400         call gemm (Z,        Binv_Zt,        dS, 'n', 'n') ! Z * Lambda^{-1} * Z'
401         call gemm (Binvsq,         Z, Binvsq_Zt, 'n', 't') ! Lambda^{-1/2} * Z'
402         call gemm (Z,      Binvsq_Zt,        dT, 'n', 'n') ! Z * Lambda^{-1/2} * Z'
403         call gemm (dXp,           dS,    dXp_dS, 'n', 'n') ! dXp * Z * Lambda^{-1} * Z'
404         call gemm (dXp_dS, dYpt_Rinv,        mK, 'n', 'n') ! dXp*Z*Lambda^{-1}*Z'*dYp'*[(N-1)R]^{-1}
405         call gemm (dXp,           dT,       dXf, 'n', 'n') ! dXf = dXp * dT
406         call gemm (mK,       Y - mYp,       mXf, 'n', 'n', 1._wp, 1._wp) ! mXf = mXp + mK*(Y - mYp)
407
408         Xf = mXf + dXf
409
410     end subroutine ETKF
411
412     !---------------------------------------------------------------
413     ! Particle Filter
414     !---------------------------------------------------------------
415     subroutine PF (Xp, Xf, H, R, Y, N_L, N_KX, N_M, t)
416
417         real(wp), parameter       :: PI = 3.14159265358973238462643383279502884_wp
418         integer,  intent(in)      :: N_L, N_KX, N_M, t
419         real(wp), intent(in)      :: Y(N_L), R(N_L, N_L), H(N_L, N_KX), Xp(N_KX, N_M)
420         real(wp), intent(inout)   :: Xf(N_KX, N_M)
421         real(wp), dimension (N_M) :: loglike_mem, like, wgt
```

```
422         real(wp)                  :: rv(1), c, Un, Ninv
423         integer                   :: i_m, i_w
424
425         ! Weights for resampling
426         call LogLikelihood (Xp, H, R, Y, loglike_mem)
427         like = exp(loglike_mem - maxval(loglike_mem))
428         wgt = like / sum(like)
429
430         ! Low-variance resampling
431         Ninv = 1._wp / dble(N_M)
432         rv = GenUnif (1, 0._wp, Ninv, t)
433
434         c = wgt(1)
435         i_w = 1
436         do i_m = 1, N_M
437             Un = rv(1) + (i_m - 1) * Ninv
438             do while (Un > c)
439                 i_w = i_w + 1
440                 c = c + wgt(i_w)
441             end do
442             Xf(:, i_m) = Xp(:, i_w)
443         end do
444
445     end subroutine PF
446
447 end module Ensemble_Filters
```

Listing A.1 Fortran Sample Code