Development of Data Assimilation System for Toroidal Plasmas

MORISHITA Yuya

2023

Contents

Chapter 1

Introduction

1.1 Fusion energy

In today's world, with problems of fossil fuel depletion and CO_2 emissions, power generation via nuclear fusion is attracting attention as a next-generation electricity source. Nuclear fusion is a reaction in which a new nucleus is produced by nuclear force when light nuclei approach each other, overcoming electrostatic repulsive force. As a result, the reaction reduces the total mass, and the corresponding energy is extracted as the kinetic energy of the charged particles and neutrons produced in the reaction. Fusion power generation converts this energy into electricity and has the following advantages compared with conventional power generation methods:

- Since we can extract the fuels from seawater (described below), they can be a semi-permanent source of electricity.

- Runaway reactions that can occur in fission reactors do not happen in fusion reactors because the core temperature must be very high to cause fusion reactions.

- Fusion reactors do not produce high-level long-lived radioactive wastes and do not emit CO_2.

The fusion reaction between the hydrogen isotopes: deuterium (D) and tritium (T),

$$\mathrm{D} + \mathrm{T} \rightarrow \mathrm{He}^4 + \mathrm{n} + 17.6\ \mathrm{MeV}, \tag{1.1}$$

is considered as the most promising reaction for the power generation in terms of reaction cross section. Deuterium can be obtained from seawater, whereas tritium is rarely found in nature. However, we can produce tritium from the lithium isotopes, Li^6 and Li^7, through the reactions:

$$\begin{aligned} \mathrm{Li}^6 + \mathrm{n} &\rightarrow \mathrm{He}^4 + \mathrm{T} + 4.8\ \mathrm{MeV}, && (1.2) \\ \mathrm{Li}^7 + \mathrm{n} + 2.5\ \mathrm{MeV} &\rightarrow \mathrm{He}^4 + \mathrm{T} + \mathrm{n}. && (1.3) \end{aligned}$$

Fortunately, seawater contains large amounts of these lithium isotopes; thus, we can obtain the fuels of the power generation from seawater semi-permanently.

Many scientists worldwide are still working on various approaches to realize power generation by fusion reactors. Magnetic confinement fusion is one of the prevailing

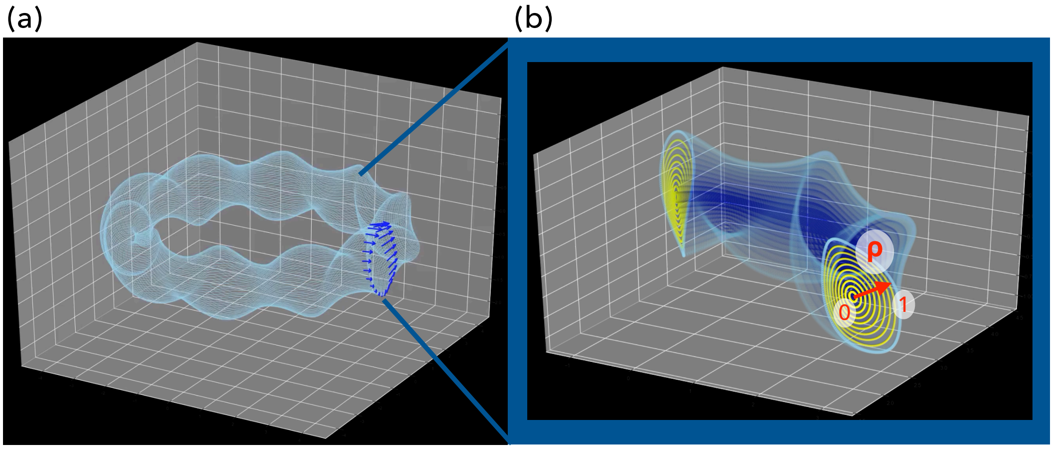


Figure 1.1: A magnetic surface (a) and a nested structure of magnetic surfaces (b) in LHD.

approaches and utilizes the fusion reaction caused by the thermal motion of the fuels. To sustain a fusion reaction, we must maintain the fuels at high temperatures and densities. The fusion reaction's temperature is extremely high (exceeding 100 million degrees Celsius), and the fuel enters a plasma state. Since no containers can withstand the ultra-high temperature, the plasma is confined using a magnetic field. The following section will briefly mention the magnetically confined plasma (fusion plasma).

1.2 Magnetically confined plasma

The basic idea of the magnetic confinement is to confine plasma using closed surfaces made of magnetic field lines, taking advantage of the property of charged particles to wind around magnetic field lines. This surface is called "magnetic surface", and is created by single magnetic field line orbiting a torus as shown in Fig. 1.1(a). These magnetic surfaces create a multilayered nested structure toward the center of the torus, as shown in Fig. 1.1(b). As a variable representing the radial position, the normalized minor radius ρ is often used. Here, ρ is determined by the magnetic surface, where 0 and 1 correspond to the center and edge of the plasma, respectively.

Magnetically confinement devices can be broadly classified into two types based on the way to create the confinement magnetic field. One is the helical device, in which the confinement magnetic field is created only by the external coils, and the other is the tokamak, in which the field is created by the toroidal current flowing in the plasma in addition to the external coils. This thesis focuses specifically on Large Helical Device (LHD) [1, 2], which is classified as a helical device.

LHD is one of the world's largest helical devices using superconducting coils and is located in Toki city, Japan. The magnetic configuration of LHD is produced by two superconducting helical coils and three pairs of superconducting poloidal coils. The magnetic field strength of LHD is 3 T and the typical major and minor radii are 3.6 m and 0.6 m, respectively, as shown in Fig. 1.2. Helical devices can be operated in

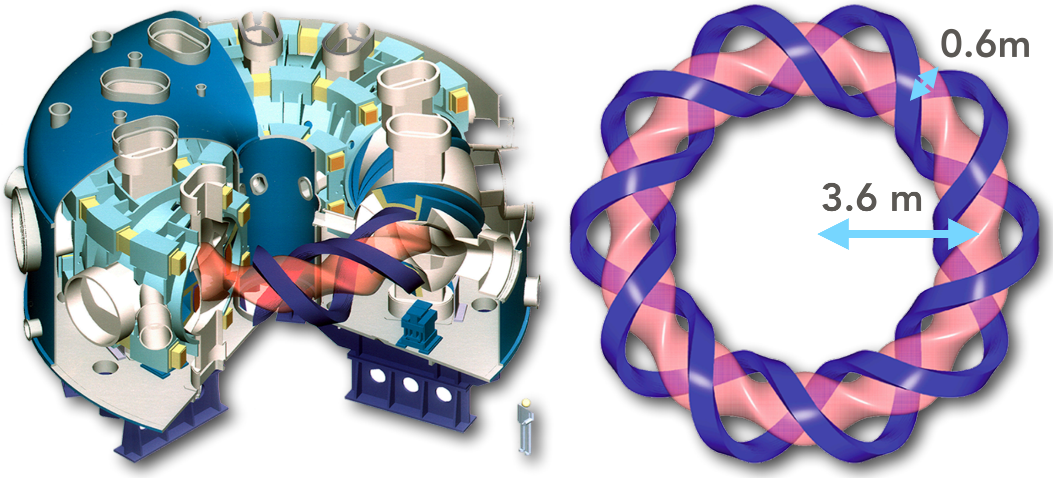


Figure 1.2: Basic concept of LHD [3].

steady-state compared to tokamaks, which require toroidal current. LHD has a long operating record of 54 minutes.

1.3 Integrated transport simulation

Fusion plasma, which is confined in a magnetic field and should be maintained under extremely high temperature and density conditions, is a typical complex system containing physical phenomena with various spatiotemporal scales [4]. To simulate the behavior of the entire fusion plasma, integrated simulation is often employed. Integrated simulation is a method of simulating the behavior of an entire system by combining modules that represent each physical phenomenon. Integrated simulation codes for fusion plasmas are being developed around the world for the prediction and analysis of entire fusion plasmas, operation scenario development, and reactor design, e.g., [5–9]. In addition, operation of future fusion reactors will require a system that continuously monitors and controls the state of the fusion plasma. Model predictive control, in which control decisions are based on predictions by simulations, is considered necessary for the control, and the integrated codes are also important as the predictive model.

Although the integrated simulation codes are expected to be utilized in various situations, they face some severe problems as follows.

- Physical models comprising the codes contain uncertainties, and their entanglements significantly reduce the predictive and analytical capabilities. Especially for turbulent transport, a model with high prediction accuracy has not been established, and the large uncertainty exists.

- To accurately predict fusion plasma behavior, the models must consider conditions outside the plasma (e.g., wall condition [10]) as well as inside the plasma, further increasing the uncertainty of the simulation [11].

- More than one simulation model cannot be optimized simultaneously. When we try to estimate a turbulent model parameter to reproduce experimental observations, other employed models in the simulation are assumed to be validated and fixed.

- Present integrated simulation codes have difficulty achieving fast and accurate calculations required for real-time prediction and control.

To overcome these problems and realize a numerical system that can analyze, predict, and control the behavior of fusion plasma with high accuracy, we introduce data assimilation techniques into the integrated simulation.

1.4 Data assimilation

In general, data assimilation is a technology that integrates observations and simulation models by optimizing state variables from limited observed information [12]. It finds the optimum combination of a numerical model and observations to estimate the state of a system more accurately. It allows us to optimize variables by quantifying the uncertainties using probability distributions. Data assimilation enhances the predictive and analytical capabilities of simulation models and is generally used for the following applications.

- Highly accurate and fast prediction.

- Model optimization.

- Estimation of unobserved quantities and state reconstruction.

- Inverse problem.

- Uncertainty quantification.

Data assimilation is mainly employed for large complex systems such as weather forecasting and ocean analysis [13–18].

The purpose of this study is to develop a numerical system that can predict and control the behavior of fusion plasma with high accuracy employing the data assimilation. Data assimilation can also improve simulation models involving turbulence models. We have been developing a data assimilation system, referred to as ASTI, as a comprehensive system for analysis, prediction, and control of fusion plasma based on an integrated code as shown in Fig. 1.3. ASTI stands for Assimilation System for Toroidal plasma Integrated simulation. ASTI is built independently of the system model code (simulation code) and can perform data assimilation on any experimental devices based on the observed data and simulation code. In this thesis, we employ the integrated transport simulation code TASK3D [19] as the system model, which is developed to predict and analyze the behavior of helical fusion plasmas [20–22].

The main feature of ASTI is that it attempts to realize predictive control of complex behavior of fusion plasmas by applying data assimilation. For this purpose, we also develop a new data assimilation framework, named as "Data Assimilation and Control

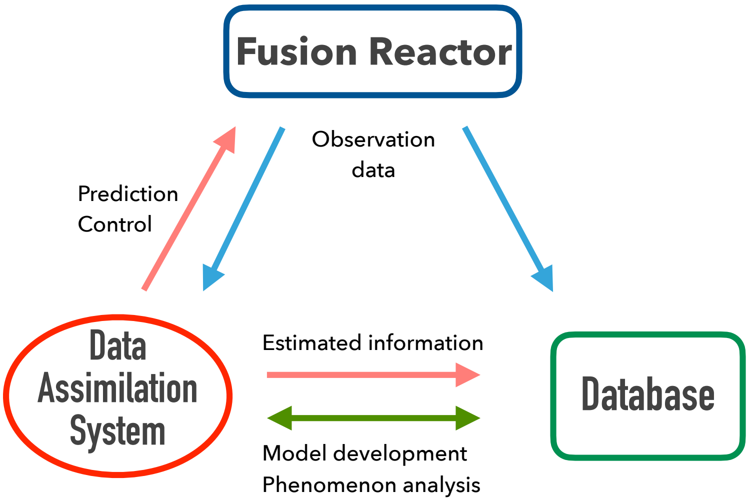


Figure 1.3: Concept of a data assimilation system.

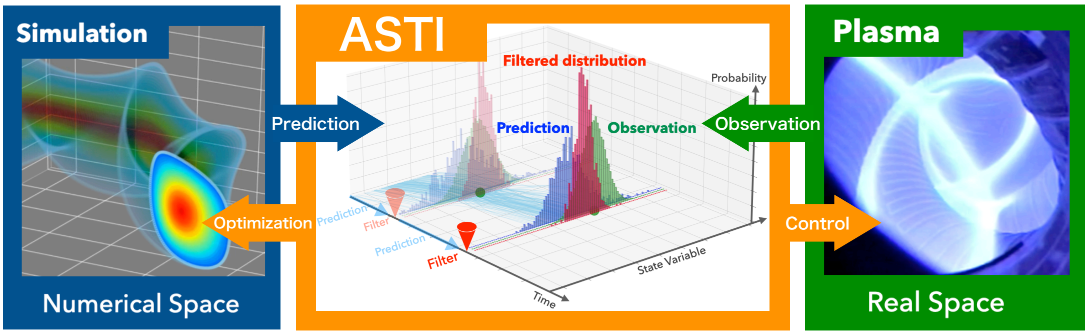


Figure 1.4: ASTI.

System" (DACS), which includes the bidirectional interaction between the data assimilation system and the real system (fusion reactor). Of course, ASTI also works as an analysis system of fusion plasma, which enables us to develop higher performance models by using a wide variety of measurement data accumulated in experimental fusion devices. ASTI is expected to become a comprehensive system that connects numerical space and real space as Fig. 1.4.

In this thesis, the data assimilation method and system model underlying ASTI are described, and the analysis and control of LHD plasmas using ASTI are discussed in detail.

1.5 Contents of this thesis

There are two major objectives of this thesis. The first is to introduce typical data assimilation methods into the integrated simulation of fusion plasmas and to investigate the effectiveness. This corresponds to the development of ASTI as an analysis system. The second is to develop a data assimilation framework including control processes to provide a foundation of flexible and powerful control systems for fusion reactors. The developed framework allows ASTI to be extended to a control system of fusion plasmas. These topics are described in the first half (Chapter 2 and 3) and the second half (Chapter 4 and 5), respectively.

In Chapter 2, the sequential Bayesian filter, a basic data assimilation framework, and data assimilation methods based on it (ensemble Kalman filter and smoother) are explained. In addition, details of the ASTI implementation and the system model (TASK3D) will be discussed. In the last part of Chapter 2, we also present a rapid calculation code of neutral beam injection (NBI) heating developed for ASTI.

In Chapter 3, The effectiveness and validity of the estimations by ASTI are investigated by several experimental data sets of NBI heated plasmas in LHD. Two cases are shown: assimilation of the experimental data into the integrated simulation of only heat transport and heat and particle transport. This chapter also discusses fast and accurate prediction and analysis by combining data assimilation and simplified models (fast models).

We propose a new data assimilation framework, DACS, developed for control of complex systems in Chapter 4. DACS integrates system model updates and optimal control-input estimation and can be applied to fusion plasma control and a wide range of other control problems in which system models can be prepared even in a simplified manner. This chapter also discusses the construction of a specific control algorithm and the implementation in ASTI.

In Chapter 5, we demonstrate the effectiveness of DACS through numerical experiments in controlling virtual (numerically created) fusion plasma. In addition, we reveal the characteristics of the control performance related to the choice of hyper parameters and the discrepancies between the system model and the real system.

In the final chapter, we summarize this thesis and discuss the future prospects of this study.

Chapter 2

Data Assimilation System, ASTI

In recent years, with advances in computer performance, numerical simulation has been developed as a major research method in natural science in addition to experimental and theoretical approaches. Computer simulation enables us to understand and predict complex phenomena. In addition, data-driven science (statistical science), in which inductive inferences are made from data, has also made great progress, and machine learning including neural networks has achieved significant results. Data assimilation is a technique that combines simulation science (deductive inference) and data-driven science (inductive inference). Data assimilation estimates the state of a target system based on information from observation data and simulation models, and was originally developed for large-scale integrated simulations in meteorology and oceanography. In recent years, data assimilation has been used in a variety of fields, partly due to improvements in computer performance [13–18, 23, 24].

Data assimilation methods are roughly divided into two types. The first type, represented by Kalman filter (KF) [25], is based on sequential Bayesian filter (estimation) [26, 27]. The second type, represented by the four-dimensional variational method [28], is based on the maximum *a posteriori* (MAP) estimation. ASTI as an observation system is based on the former, sequential Bayesian filter, and the integrated transport simulation code for helical fusion plasmas, TASK3D is employed as the system model.

In this chapter, we explain the sequential Bayesian filter and data assimilation methods based on it (ensemble Kalman filter and smoother). In addition, details of the system model (TASK3D) and the models developed for high-speed calculations are discussed.

2.1 Data assimilation for state estimation

In this section, we introduce the sequential Bayesian filter, one of the typical data assimilation methods. In ASTI, the data assimilation methods for state estimation based on the sequential Bayesian filter: ensemble Kalman filter and smoother [29] and particle filter [27] are implemented.

2.1.1 State-space model

In general, simulation models are developed to compute the time evolution of a single state of the system. In other words, the time evolution of the system is calculated deterministically by the simulation model. However, in order to sequentially optimize simulation models by observation, it is necessary to consider uncertainty in the simulation to assimilate the observed data. In data assimilation, we consider a probabilistic distribution of a state vector (state distribution). Here, state vector is a vector whose components are variables that define a state of a target system (state variable). We calculate the time evolution of the probability distribution in the state space using the system model. In other words, we follow multiple solutions in the form of probability distribution. Note that a general deterministic simulation corresponds to taking the state distribution as a delta function.

The framework of the sequential Bayesian filter assumes the state-space model consisting of a system model and an observation model shown as Eqs. (2.1) and (2.2), respectively.

$$\mathbf{x}_t = f_t(\mathbf{x}_{t-\Delta t}, \mathbf{v}_t), \tag{2.1}$$

$$\mathbf{y}_t = h_t(\mathbf{x}_t, \mathbf{w}_t). \tag{2.2}$$

The vectors $\mathbf{x}_t$ and $\mathbf{y}_t$ denote the state vector and the observation vector at a time t. The system model represents the temporal evolution of state vector from time $t - \Delta t$ to t by the nonlinear operator f_t. Since the operator f_t (simulation model) is generally incomplete, the uncertainty of $\mathbf{x}$ should increase with time evolution. This uncertainty associated with time evolution is considered as the system noise $\mathbf{v}_t$. In this thesis, TASK3D is employed as the nonlinear operator f_t. The observation model, Eq. (2.2), represents the relationship between the state vector $\mathbf{x}_t$ and the observation vector $\mathbf{y}_t$ considering the observation noise $\mathbf{w}_t$. The observation model allows the state vector and the observation vector to be compared. It is noted that the observation noise is not only determined by measurement errors caused by the instruments, but also includes errors caused by phenomena that cannot be represented by the simulation model (f_t). The observation model is often written as a linear model:

$$\mathbf{y}_t = H_t \mathbf{x}_t + \mathbf{w}_t. \tag{2.3}$$

This observation model projects the state vector on the observation space by the matrix H_t called the observation matrix, and the observation noise $\mathbf{w}_t$ is added to the projected vector. This thesis also assumes the linear observation model.

In typical data assimilation frameworks, the state-space model is defined by coupling the system model and observation model. The cooperation of these two models provides a foundation for optimizing the state vector by combining the observed data with the simulation. To perform data assimilation using the state-space model, the probability distributions of the initial state ($p(\mathbf{x}_0)$), system noise ($p(\mathbf{v}_t)$), and observation noise ($p(\mathbf{w}_t)$) must be determined in advance. The specific data assimilation process is described in the next section and beyond.

2.1.2 Sequential Bayesian filter

In this chapter, we assume that observation data are available at time t_i $(i \geq 1)$. When a time point is written in a subscript, it is denoted as A_i, which indicates variable A at time t_i. In addition, we use the multiple time notation $A_{1:i} \equiv \{A_1, A_2, \cdots, A_i\}$.

The state distributions at a given time t_i are divided into the following three types, depending on up to which time the observation information was assimilated.

- Predicted distribution, $p(\mathbf{x}_i \mid \mathbf{y}_{1:i-1})$
- Filtered distribution, $p(\mathbf{x}_i \mid \mathbf{y}_{1:i})$
- Smoothed distribution, $p(\mathbf{x}_i \mid \mathbf{y}_{1:I})$ $(I > i)$

The predicted and filtered distributions appear in the filter procedure shown below. The smoothed distribution is discussed in the next subsection.

The procedure of the sequential Bayesian filter comprises two steps on the state distribution, i.e., prediction and filtering (assimilation of observation into the distribution). The change in the probability distribution of state vector at each step can be summarized as follows.

- Prediction
$$p(\mathbf{x}_i \mid \mathbf{y}_{1:i}) \to p(\mathbf{x}_{i+1} \mid \mathbf{y}_{1:i}). \tag{2.4}$$
- Filter
$$p(\mathbf{x}_{i+1} \mid \mathbf{y}_{1:i}) \to p(\mathbf{x}_{i+1} \mid \mathbf{y}_{1:i+1}). \tag{2.5}$$

Here, $\mathbf{y}_{1:i} \equiv \{\mathbf{y}_1, \cdots, \mathbf{y}_i\}$, and $p(\mathbf{x}_i \mid \mathbf{y}_{1:i})$ means the distribution of $\mathbf{x}_i$ including the observed information from $t = 1$ to $t = i$. Given an initial distribution $p(\mathbf{x}_0)$, the data assimilation process can proceed by repeating the prediction step and filtering step alternately, as shown in Fig. 2.1.

Prediction step

In this step, we predict the state distribution $p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i})$ from $p(\mathbf{x}_i \mid \mathbf{y}_{1:i})$ using the system model, Eq. (2.1). The state distribution $p(\mathbf{x}_{i+1} \mid \mathbf{y}_{1:i})$ is calculated by

$$\begin{aligned} p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i}) &= \int p(\mathbf{x}_{i+1}, \mathbf{x}_i|\mathbf{y}_{1:i})\mathrm{d}\mathbf{x}_i \\ &= \int p(\mathbf{x}_{i+1}|\mathbf{x}_i, \mathbf{y}_{1:i})p(\mathbf{x}_i|\mathbf{y}_{1:i})\mathrm{d}\mathbf{x}_i \\ &= \int p(\mathbf{x}_{i+1}|\mathbf{x}_i)p(\mathbf{x}_i|\mathbf{y}_{1:i})\mathrm{d}\mathbf{x}_i. \end{aligned} \tag{2.6}$$

Here, we assumed the Markov property attached to the system model, Eq. (2.1):

$$p(\mathbf{x}_{i+1}|\mathbf{x}_i, \mathbf{y}_{1:i}) = p(\mathbf{x}_{i+1}|\mathbf{x}_i). \tag{2.7}$$

This first-order Markov property is just a condition imposed on the state vector, and higher-order Markov chains can be realized by including state variables at multiple

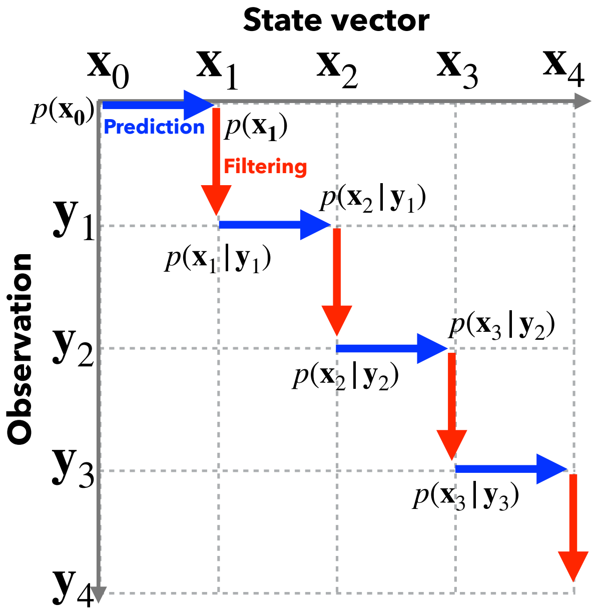


Figure 2.1: Data assimilation process of sequential Bayesian filter.

time points in the state vector. Using the system model, the conditional distribution $p(\mathbf{x}_{i+1}|\mathbf{x}_i)$ in Eq. (2.6) is given as

$$\begin{aligned} p(\mathbf{x}_{\mathbf{i+1}}|\mathbf{x}_i) &= \int p(\mathbf{x}_{i+1}, \mathbf{v}_{i+1}|\mathbf{x}_i)\mathrm{d}\mathbf{v}_{i+1} \\ &= \int \delta(\mathbf{x}_{i+1} - f_{i+1}(\mathbf{x}_i, \mathbf{v}_{i+1}))p(\mathbf{v}_{i+1})\mathrm{d}\mathbf{v}_{i+1}. \end{aligned} \tag{2.8}$$

Filtering step

Once the predicted distribution $p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i})$ is computed and the observation data at time $i+1$ is available, the filtered distribution can be computed from Bayes' theorem as follows:

$$\begin{aligned} p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i+1}) &= p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i}, \mathbf{y}_{i+1}) \\ &= \frac{p(\mathbf{x}_{i+1}, \mathbf{y}_{i+1}|\mathbf{y}_{1:i})}{p(\mathbf{y}_{i+1}|\mathbf{y}_{1:i})} \\ &= \frac{p(\mathbf{y}_{i+1}|\mathbf{x}_{i+1}, \mathbf{y}_{1:i})p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i})}{p(\mathbf{y}_{i+1}|\mathbf{y}_{1:i})} \\ &= \frac{p(\mathbf{y}_{i+1}|\mathbf{x}_{i+1})p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i})}{p(\mathbf{y}_{i+1}|\mathbf{y}_{1:i})} \\ &= \frac{p(\mathbf{y}_{i+1}|\mathbf{x}_{i+1})p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i})}{\int p(\mathbf{y}_{i+1}, \mathbf{x}_{i+1}|\mathbf{y}_{1:i})\mathrm{d}\mathbf{x}_{i+1}} \\ &= \frac{p(\mathbf{y}_{i+1}|\mathbf{x}_{i+1})p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i})}{\int p(\mathbf{y}_{i+1}|\mathbf{x}_{i+1})p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i})\mathrm{d}\mathbf{x}_{i+1}}. \end{aligned} \tag{2.9}$$

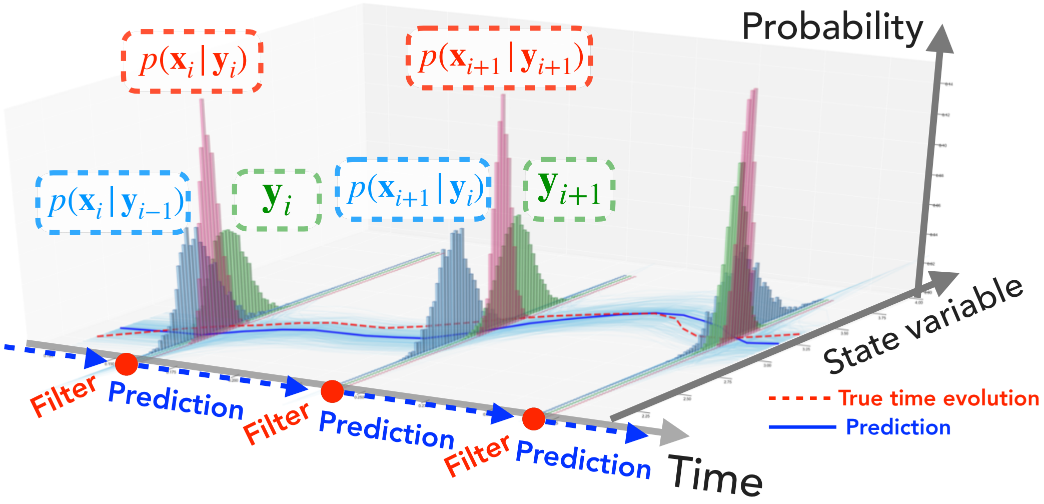


Figure 2.2: Changes in state distribution in the sequential Bayesian filter.

Here, we assumed the Markov property attached to the observation model, Eq. (2.2):

$$p(\mathbf{y}_{i+1}|\mathbf{x}_{i+1}, \mathbf{y}_{1:i}) = p(\mathbf{y}_{i+1}|\mathbf{x}_{i+1}). \tag{2.10}$$

The numerator at the rightmost side of Eq. (2.9) is given by the predicted distribution $p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i})$ multiplied by the likelihood function $p(\mathbf{y}_{i+1}|\mathbf{x}_{i+1})$.

We start the data assimilation procedure from the initial distribution $p(\mathbf{x}_0)$ and repeat the prediction and filtering steps. This procedure allows us to perform the system state prediction while optimizing the simulation model so that the prediction is always successful. Figure 2.2 illustrates the changes in state distribution in the procedure of sequential Bayesian filter.

2.1.3 Smoother

The filter optimizes the state vector to enhance the prediction capability of the simulation model based on the observation data. The filtered estimate includes information that should be considered in the simulation model. However, it is known that there is a time difference (delay) between the filtered estimate and observation, because the filtered estimate is based on the observation data prior to the time of filtering [30]. To estimate the state vector in a reasonable manner both temporally and spatially, smoothing procedure is required. The smoother corrects the filtered distribution using observation data posterior to the time of filtering (future data).

There are several methods of the smoothing, here we describe the fixed-interval smoothing, implemented in ASTI. This method optimizes the state distributions backward from time I ($I > i$) in sequence like $p(\mathbf{x}_I|\mathbf{y}_{1:I}) \to p(\mathbf{x}_{I-1}|\mathbf{y}_{1:I}) \to \cdots \to p(\mathbf{x}_i|\mathbf{y}_{1:I})$. This method uses the observed data up to time I to estimate all smoothed distributions up to time I as shown in Fig. 2.3.

Assume that the filtering up to time I has been completed and the predicted and filtered distributions up to time I have been obtained. The smoothed distribution at time

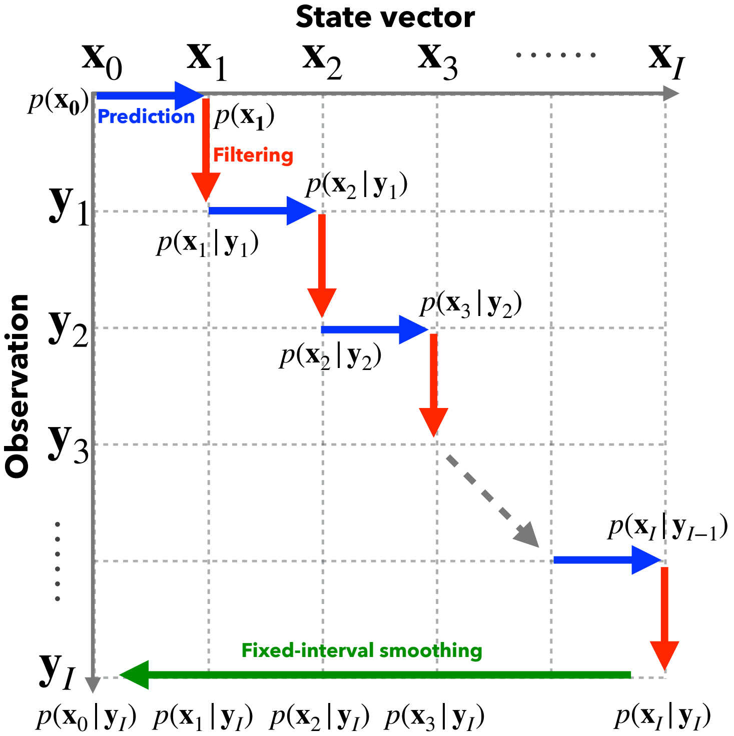

State vector
$\mathbf{x}_0$
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_3$
......
$\mathbf{x}_I$
Observation
$\mathbf{y}_1$
$\mathbf{y}_2$
$\mathbf{y}_3$
$\mathbf{y}_I$
$p(\mathbf{x}_0)$
Prediction
$p(\mathbf{x}_1)$
Filtering
$p(\mathbf{x}_2|\mathbf{y}_1)$
$p(\mathbf{x}_1|\mathbf{y}_1)$
$p(\mathbf{x}_3|\mathbf{y}_2)$
$p(\mathbf{x}_2|\mathbf{y}_2)$
$p(\mathbf{x}_I|\mathbf{y}_{I-1})$
Fixed-interval smoothing
$p(\mathbf{x}_0|\mathbf{y}_I)$
$p(\mathbf{x}_1|\mathbf{y}_I)$
$p(\mathbf{x}_2|\mathbf{y}_I)$
$p(\mathbf{x}_3|\mathbf{y}_I)$
$p(\mathbf{x}_I|\mathbf{y}_I)$

Figure 2.3: Process of fixed-interval smoothing.

i, $p(\mathbf{x}_i|\mathbf{y}_{1:I})$, can be written using the predicted distribution $p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i})$, the filtered distribution $p(\mathbf{x}_i|\mathbf{y}_{1:i})$, and the smoothed distribution one time ahead, $p(\mathbf{x}_{i+1}|\mathbf{y}_{1:I})$, as follows.

$$\begin{aligned}
p(\mathbf{x}_i|\mathbf{y}_{1:I}) &= \int p(\mathbf{x}_i, \mathbf{x}_{i+1}|\mathbf{y}_{1:I})\mathrm{d}\mathbf{x}_{i+1} \\
&= \int p(\mathbf{x}_i|\mathbf{x}_{i+1}, \mathbf{y}_{1:i})p(\mathbf{x}_{i+1}|\mathbf{y}_{1:I})\mathrm{d}\mathbf{x}_{i+1} \\
&= \int \frac{p(\mathbf{x}_i, \mathbf{x}_{i+1}|\mathbf{y}_{1:i})}{p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i})}p(\mathbf{x}_{i+1}|\mathbf{y}_{1:I})\mathrm{d}\mathbf{x}_{i+1} \\
&= \int \frac{p(\mathbf{x}_i|\mathbf{y}_{1:i})p(\mathbf{x}_{i+1}|\mathbf{x}_i)}{p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i})}p(\mathbf{x}_{i+1}|\mathbf{y}_{1:I})\mathrm{d}\mathbf{x}_{i+1} \\
&= p(\mathbf{x}_i|\mathbf{y}_{1:i}) \int \frac{p(\mathbf{x}_{i+1}|\mathbf{x}_i)p(\mathbf{x}_{i+1}|\mathbf{y}_{1:I})}{p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i})}\mathrm{d}\mathbf{x}_{i+1}. \qquad (2.11)
\end{aligned}$$

Here, we assumed the Markov property,

$$\begin{aligned}
p(\mathbf{x}_i|\mathbf{x}_{i+1},\mathbf{y}_{1:I}) &= p(\mathbf{x}_i|\mathbf{x}_{i+1},\mathbf{y}_{1:i},\mathbf{y}_{i+1:I}) \\
&= \frac{p(\mathbf{y}_{i+1:I}|\mathbf{x}_i,\mathbf{x}_{i+1},\mathbf{y}_{1:i})p(\mathbf{x}_i|\mathbf{x}_{i+1},\mathbf{y}_{1:i})}{p(\mathbf{y}_{i+1:I}|\mathbf{x}_{i+1},\mathbf{y}_{1:i})} \\
&= \frac{p(\mathbf{y}_{i+1:I}|\mathbf{x}_{i+1},\mathbf{y}_{1:i})p(\mathbf{x}_i|\mathbf{x}_{i+1},\mathbf{y}_{1:i})}{p(\mathbf{y}_{i+1:I}|\mathbf{x}_{i+1},\mathbf{y}_{1:i})} \\
&= p(\mathbf{x}_i|\mathbf{x}_{i+1},\mathbf{y}_{1:i}). \qquad (2.12)
\end{aligned}$$

From Eq. (2.11), the smoothing procedure can be interpreted as correcting the filtered distribution $p(\mathbf{x}_i|\mathbf{y}_{1:i})$ with future observations. Equation (2.11) propagates observed information from the future to the past. Combination of the prediction, filter, and smoother allows us to estimate the state distribution at any given time based on observation data up to an arbitrary time.

Up to here, we have discussed the data assimilation procedure without assuming any specific form of probability distribution or any specific state-space model. In the next subsection, we introduce specific algorithms of the sequential Bayesian filter.

2.1.4 Kalman filter and smoother

Assuming that the system model and observation model are linear models, and the noises are white noises following Gaussian distributions, Eqs. (2.1) and (2.2) are written as

$$\begin{aligned}
\mathbf{x}_{i+1} &= F_{i+1}\mathbf{x}_i + \mathbf{v}_{i+1},\ \mathbf{v}_{i+1} \sim N(\mathbf{0}, Q_{i+1}), \qquad (2.13) \\
\mathbf{y}_i &= H_i\mathbf{x}_i + \mathbf{w}_i,\ \mathbf{w}_i \sim N(\mathbf{0}, R_i), \qquad (2.14)
\end{aligned}$$

where the matrices F_{i+1} and H_i are a time evolution matrix of $\mathbf{x}_i \to \mathbf{x}_{i+1}$ and an observation matrix, respectively. Matrices Q and R are covariance matrices of the system noise and observation noise, respectively, where $N(\mathbf{0}, \Sigma)$ denotes the Gaussian distribution with zero mean and covariance matrix Σ. The sequential Bayesian filter under the state-space model, Eqs. (2.13) and (2.14), is specifically called "Kalman filter" (KF). A major feature of KF is that the sate distribution remains Gaussian throughout the data assimilation processes.

The prediction step of the sequential Bayesian filter can be calculated as follows:

$$\begin{aligned}
\mathbf{x}_{i+1}|\mathbf{y}_{1:i} &\sim N(\mathbf{x}_{i+1|i}, V_{i+1|i}), \qquad (2.15) \\
\mathbf{x}_{i+1|i} &= F_{i+1}\mathbf{x}_{i|i}, \qquad (2.16) \\
V_{i+1|i} &= F_{i+1}V_{i|i}F_{i+1}^{\mathrm{T}} + Q_{i+1}, \qquad (2.17)
\end{aligned}$$

where the matrix V is the covariance matrix of state distribution and the superscript T denotes the matrix transposition. The variables with subscript $a|b$ indicate the parameters related to the distribution $p(\mathbf{x}_a|\mathbf{y}_{1:b})$.

From the assumption of Gaussian distribution, the filtering step is given by

$$\begin{aligned} \mathbf{x}_{i+1}|\mathbf{y}_{1:i+1} &\sim N(\mathbf{x}_{i+1|i+1}, V_{i+1|i+1}), & (2.18)\\ \mathbf{x}_{i+1|i+1} &= \mathbf{x}_{i+1|i} + K_{i+1}(\mathbf{y}_{i+1} - H_{i+1}\mathbf{x}_{i+1|i}), & (2.19)\\ V_{i+1|i+1} &= V_{i+1|i} - K_{i+1}H_{i+1}V_{i+1|i}, & (2.20)\\ K_{i+1} &= V_{i+1|i}H_{i+1}^{\mathrm{T}}(H_{i+1}V_{i+1|i}H_{i+1}^{\mathrm{T}} + R_{i+1})^{-1}. & (2.21) \end{aligned}$$

Here, the matrix K, which plays an important role in this calculation, is called "Kalman gain". The modification of the mean vector by the filter depends on K_{i+1} and the difference between prediction and observation, $\mathbf{y}_{i+1} - H_{i+1}\mathbf{x}_{i+1|i}$, as shown in Eq. (2.19). The variables without observations are also optimized by the filter through K. As can be seen from the calculations of prediction and filtering, The KF procedure is closed with the mean and covariance matrix of the Gaussian distribution. See [26, 31] for details of the formula transformations.

Similarly, the smoother can be calculated as follows:

$$\begin{aligned} \mathbf{x}_i|\mathbf{y}_{1:I} &\sim N(\mathbf{x}_{i|I}, V_{i|I}), & (2.22)\\ \mathbf{x}_{i|I} &= \mathbf{x}_{i|i} + A_i(\mathbf{x}_{i+1|I} - \mathbf{x}_{i+1|i}), & (2.23)\\ V_{i|I} &= V_{i|i} + A_i(V_{i+1|I} - V_{i+1|i})A_i^{\mathrm{T}}, & (2.24)\\ A_i &= V_{i|i}F_{i+1}^{\mathrm{T}}V_{i+1|i}^{-1}. & (2.25) \end{aligned}$$

This smoothing calculation is called "Kalman smoother" (KS).

Although KF is a powerful state estimation technique, its target systems are limited to linear systems. In general, simulation models are rarely linear, and systems in which data assimilation is introduced often require nonlinear and complex simulation models. The ensemble Kalman filter (EnKF), which will be discussed in the next section, is an extension of KF and performs data assimilation to nonlinear systems using an ensemble approximation.

2.1.5 Ensemble Kalman filter and smoother

The ensemble Kalman filter (EnKF) [29] approximates the state distribution by an ensemble and constructs the ensemble sequentially so that its sample mean and covariance matrix asymptotically match the conditional mean and covariance matrix expected by the KF. Since the prediction step is approximated by computing the time evolution of each ensemble member using the simulation model, the EnKF can perform the sequential Bayesian filter even for nonlinear system models, as shown in Fig. 2.4. In addition, the smoothing process can also be performed by ensemble Kalman smoother (EnKS).

EnKF

The EnKF assumes a state-space model consisting of a nonlinear system model and a linear observation model shown as Eqs. (2.26) and (2.27), respectively.

$$\begin{aligned} \mathbf{x}_{i+1} &= f_{i+1}(\mathbf{x}_i, \mathbf{v}_{i+1}),\ \mathbf{v}_{i+1} \sim N(\mathbf{0}, Q_{i+1}), & (2.26)\\ \mathbf{y}_i &= H_i\mathbf{x}_i + \mathbf{w}_i,\ \mathbf{w}_i \sim N(\mathbf{0}, R_i). & (2.27) \end{aligned}$$

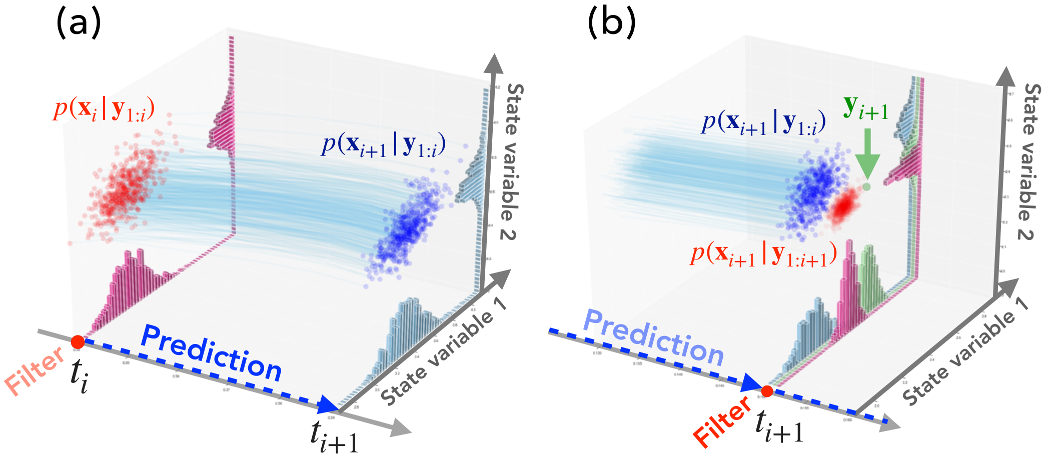

(a)
$p(\mathbf{x}_i|\mathbf{y}_{1:i})$
$p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i})$
State variable 2
State variable 1
Filter
t_i
Prediction
t_{i+1}
(b)
$p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i})$
$\mathbf{y}_{i+1}$
$p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i+1})$
State variable 2
State variable 1
Prediction
Filter
t_{i+1}

Figure 2.4: Changes in ensemble members at each step of the EnKF: prediction (a) and filter (b).

The EnKF works even for system noise that is not Gaussian distributed. The state distribution $p(\mathbf{x})$ is approximated by an ensemble consisting of $\{\mathbf{x}^{(n)}\}_{n=1}^{n=N}$ as follows:

$$p(\mathbf{x}) \simeq \frac{1}{N}\sum_{n=1}^{N}\delta\left(\mathbf{x}-\mathbf{x}^{(n)}\right), \tag{2.28}$$

where N is the number of ensemble members, and n is the index of them. The prediction is executed by calculating the time evolution of each ensemble member based on the system model:

$$\mathbf{x}_{i+1|i}^{(n)} = f_{i+1}(\mathbf{x}_{i|i}^{(n)}, \mathbf{v}_{i+1}^{(n)}). \tag{2.29}$$

It is noted that $\mathbf{x}_{t|s}^{(n)}$ is the n-th ensemble member that approximates the state distribution at the time t given $\mathbf{y}_{1:s} \equiv \{\mathbf{y}_1, \cdots, \mathbf{y}_s\}$, $p(\mathbf{x}_t|\mathbf{y}_{1:s})$. Therefore, $\mathbf{x}_{i+1|i}^{(n)}$ is an ensemble member approximating the predicted state distribution $p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i})$, and $\mathbf{x}_{i|i}^{(n)}$ is an ensemble member approximating the filtered state distribution at the previous time step, $p(\mathbf{x}_i|\mathbf{y}_{1:i})$. The vector $\mathbf{v}_{i+1}^{(n)}$ is the n-th sample drawn from the system noise distribution $p(\mathbf{v}_{i+1})$, i.e., $N(\mathbf{0}, Q_{i+1})$. In this study, the system noise, $\mathbf{v}_{i+1}^{(n)}$, is added before the time evolution, i.e., $\mathbf{x}_{i+1|i}^{(n)} = f_{i+1}(\mathbf{x}_{i|i}^{(n)} + \mathbf{v}_{i+1}^{(n)})$. Each ensemble member examines a time evolution with different initial values and model parameters.

The current filtered distribution is obtained in the filtering step by calculating Eq. (2.30) for each ensemble member [29].

$$\begin{aligned}\mathbf{x}_{i+1|i+1}^{(n)} &= \mathbf{x}_{i+1|i}^{(n)} + \hat{K}_{i+1}(\mathbf{y}_{i+1} - H_{i+1}\mathbf{x}_{i+1|i}^{(n)} + \mathbf{w}_{i+1}^{(n)}), &(2.30)\\ \hat{K}_{i+1} &= \hat{V}_{i+1|i}H_{i+1}^{\mathrm{T}}(H_{i+1}\hat{V}_{i+1|i}H_{i+1}^{\mathrm{T}} + R_{i+1})^{-1}. &(2.31)\end{aligned}$$

Here, $\mathbf{w}_{i+1}^{(n)}$ is the n-th sample drawn from the observation noise distribution $p(\mathbf{w}_{i+1})$, and $\hat{V}_{i+1|i}$ is a sample covariance matrix of the ensemble for the predicted distribution:

$$\begin{aligned}
\hat{V}_{i+1|i} &= \frac{1}{N-1}\sum_{n=1}^{N}(\mathbf{x}_{i+1|i}^{(n)} - \hat{\mathbf{x}}_{i+1|i})(\mathbf{x}_{i+1|i}^{(n)} - \hat{\mathbf{x}}_{i+1|i})^{\mathrm{T}}, && (2.32)\\
\hat{\mathbf{x}}_{i+1|i} &= \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_{i+1|i}^{(n)}. && (2.33)
\end{aligned}$$

The matrix R_{i+1} is a covariance matrix of the observation noise, i.e., $\mathbf{w}_{i+1} \sim N(\mathbf{0}, R_{i+1})$. Since the matrix $\hat{K}_{i+1}$ is determined by the covariance matrix of the predicted distribution, $\hat{V}_{i+1|i}$, and the covariance matrix of the observation noise, R_{i+1}, in Eq. (2.31), the EnKF optimizes the entire state vector using limited observation information based on the correlation between the state variables generated through the time evolution by system model. This mechanism allows the EnKF to enhance the prediction capability of the simulation model using the data observed sequentially.

The filtering can be performed by matrix calculation. We introduce the following matrices to derive the matrix representation of the filtering:

$$\begin{aligned}
X_{i|*} &= \left(\mathbf{x}_{i|*}^{(1)}, \mathbf{x}_{i|*}^{(2)}, \cdots \mathbf{x}_{i|*}^{(N)}\right),\\
\check{X}_{i|*} &= \left(\mathbf{x}_{i|*}^{(1)} - \hat{\mathbf{x}}_{i|*},\ \mathbf{x}_{i|*}^{(2)} - \hat{\mathbf{x}}_{i|*},\ \cdots,\ \mathbf{x}_{i|*}^{(N)} - \hat{\mathbf{x}}_{i|*}\right),\\
W_i &= \left(\mathbf{w}_i^{(1)}, \mathbf{w}_i^{(2)}, \cdots, \mathbf{w}_i^{(N)}\right),\\
Y_i &= \left(\mathbf{y}_i, \mathbf{y}_i, \cdots, \mathbf{y}_i\right),
\end{aligned}$$

where, $\hat{\mathbf{x}}_{i|*}$ is the sample mean of $\mathbf{x}_{i|*}^{(n)}$, and the matrix Y_i has N columns. The filtering calculation given by Eq. (2.30) can be written as follows:

$$\begin{aligned}
X_{i+1|i+1} &= X_{i+1|i}Z_{i+1}, && (2.34)\\
Z_{i+1} &= I_N + \check{X}_{i+1|i}^{\mathrm{T}}H_t^{\mathrm{T}}(H_{i+1}\hat{V}_{i+1|i}H_{i+1}^{\mathrm{T}} + R_{i+1})^{-1}\\
&\quad \times(Y_{i+1} + W_{i+1} - H_{i+1}X_{i+1|i}). && (2.35)
\end{aligned}$$

Here, I_N is an $N \times N$ identity matrix.

EnKS

The EnKS corrects the filtered estimate by the EnKF using observation data posterior to the time of filtering (future data). The EnKS can estimate the state vector more reasonably, not only spatially but also temporally. The smoothing calculation by the EnKS can be easily performed by storing a matrix that reconstructs the ensemble members at the filtering step.

The smoothed ensemble at time t_i reflecting the observation data until time t_{i+1}, $X_{i|i+1}$, can be obtained by Eq. (2.36) using the filtered ensemble at time t_i, $X_{i|i}$, and the reconstruction matrix given by Eq. (2.34) at time t_{i+1}, Z_{i+1} [29].

$$X_{i|i+1} = X_{i|i}Z_{i+1}. \qquad (2.36)$$

Using Eq. (2.36), the recursion formula of the smoothed ensemble is obtained as

$$X_{i|k+1} = X_{i|k} Z_{k+1}, \tag{2.37}$$

where $k > i$. Using the recursion formula, we can obtain the smoothed estimate at time t_i, based on the observation data up to an arbitrary time $t_k (> t_i)$ as follows:

$$X_{i|k} = X_{i|i} Z_{i+1} Z_{i+2} \cdots Z_k. \tag{2.38}$$

This smoothing calculation can be executed only by storing the reconstruction matrix Z_i at every filtering step.

When the time-series observation data $\mathbf{y}_{1:I}$ are given, the smoothed ensemble matrices can be calculated efficiently by the following procedure (fixed-interval smoother) [32].

(a) Determine the initial ensemble matrix $X_{0|0}$ and store this matrix.

(b) Execute the EnKF procedure until the time t_I storing the filtered ensemble matrix $X_{i|i}$ and the reconstruction matrix Z_i at every filtering step ($i = 1, 2, \cdots, I$).

(c) Initialize a matrix A to an identity matrix and set $i \leftarrow I - 1$.

(d) At time t_i, set $A \leftarrow Z_{i+1} A$ and calculate the smoothed ensemble matrix at time t_i by $X_{i|I} = X_{i|i} A$.

(e) Set $i \leftarrow i - 1$ and repeat step 4 until all smoothed ensemble matrices are obtained.

2.1.6 Comparison of ensemble Kalman filter and particle filter

The EnKF assumes the state distribution to be Gaussian (i.e., uses up to the second-order moments of the approximated distribution: mean and covariance matrix), and the filter can be performed by matrix calculations. If the target system is highly nonlinear and the relationships between state variables are far from linear, or if the observation model is nonlinear, the estimation by the EnKF does not always work.

The particle filter (PF) [27] is also a powerful data assimilation technique for nonlinear systems based on the sequential Bayesian filter. Like the EnKF, the PF approximates a state distribution with a finite number of ensemble members. The PF performs the filtering step by random sampling from the ensemble members approximating the predicted distribution. The ensemble members are weighted by a likelihood function determined by the observation. Thus, the PF can optimize state distributions of arbitrary shape by preferentially sampling ensemble members with high likelihood weights. However, the PF requires more ensemble members than the EnKF because it is more sensitive to the approximation of the distribution with a finite number of ensemble members

Although both the EnKF and the PF are implemented in ASTI, we have confirmed that EnKF is sufficient in many cases. In this thesis, we focus on the estimation results employing the EnKF.

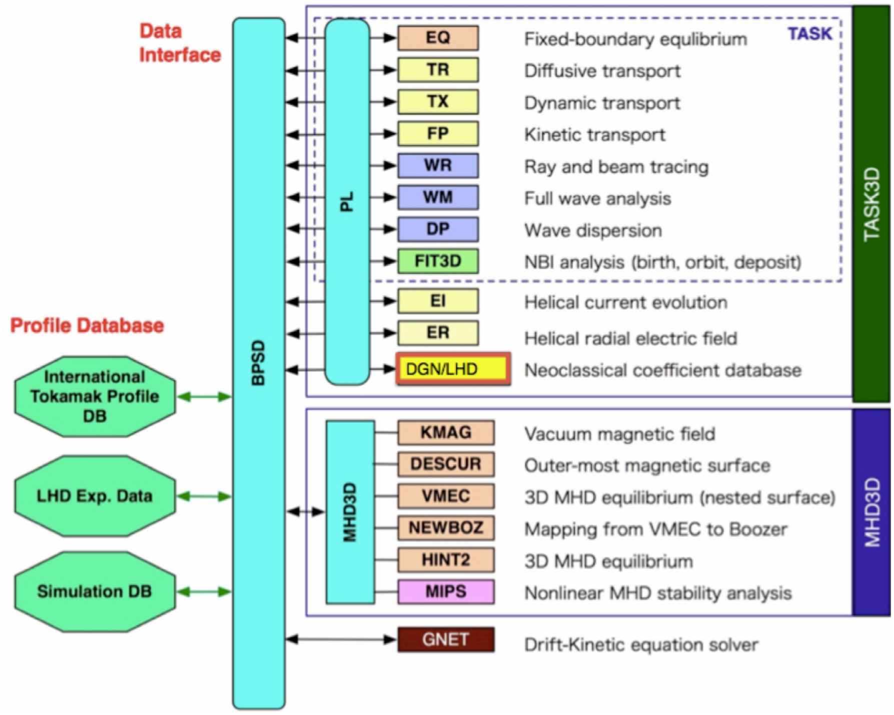

Data Interface
Profile Database
International Tokamak Profile DB
LHD Exp. Data
Simulation DB
BPSD
PL
TASK
EQ
Fixed-boundary equlibrium
TR
Diffusive transport
TX
Dynamic transport
FP
Kinetic transport
WR
Ray and beam tracing
WM
Full wave analysis
DP
Wave dispersion
FIT3D
NBI analysis (birth, orbit, deposit)
EI
Helical current evolution
ER
Helical radial electric field
DGN/LHD
Neoclassical coefficient database
TASK3D
MHD3D
KMAG
Vacuum magnetic field
DESCUR
Outer-most magnetic surface
VMEC
3D MHD equilibrium (nested surface)
NEWBOZ
Mapping from VMEC to Boozer
HINT2
3D MHD equilibrium
MIPS
Nonlinear MHD stability analysis
MHD3D
GNET
Drift-Kinetic equation solver

Figure 2.5: The module structure of TASK3D.

2.2 System model of fusion plasma

Fusion plasma is a complex system containing physical phenomena with various spatial and temporal scales, and it is virtually impossible to simulate the behavior of the entire fusion plasma with a single model. Thus, the simulations require integrated simulation codes, which integrate elemental codes (modules) describing physical phenomena in limited time and space scales.

TASK3D is an integrated simulation code for helical fusion plasmas and has been developed based on the TASK code [33, 34], which is applicable for tokamak configurations. We employ the TASK3D code as the system model of ASTI. TASK3D is composed of a variety of modules as shown in Fig. 2.5. Since TASK and TASK3D have an interface to exchange simulation results with external codes, various codes can be connected to TASK3D depending on the simulation target. In this section, we discuss the system model, TASK3D, and external codes for neutral beam injection heating: HFREYA and GNET.

2.2.1 TASK3D

Generally, actual fusion plasma has a torus shape and is confined in the nested magnetic flux surface, as shown in Fig. 1.1. In integrated simulations of toroidal plasmas, the time evolution of spatial distributions of plasma density, temperature, current density, etc. is governed by transport processes whose characteristic time is the confinement time of particles, momentum, energy, magnetic flux, etc. Since the characteristic times of many phenomena other than transport are much shorter than those of the transport process, we can say that the long-time behavior of the plasma is described by the transport phenomena. In addition, the flow of heat and particles across magnetic field lines (magnetic surfaces) is slower than the flow along the field; thus, we can assume that plasma temperature T and density n are constant on a magnetic surface, i.e., $T(\rho)$ and $n(\rho)$. In TASK3D, the radial heat transport and particle transport are treated as issues in a torus-shaped plasma as one-dimensional (1D) problems for the normalized minor radius ρ.

The core of TASK3D is the following particle and heat transport equations for each electron and ion species [35]:

$$\begin{aligned} \frac{\partial}{\partial t}\left(n_s \mathcal{V}'\right) &= -\frac{\partial}{\partial \rho}\left(\mathcal{V}'\Gamma_s\right) + S_s\mathcal{V}', &(2.39)\\ \frac{\partial}{\partial t}\left(\frac{3}{2}n_s T_s \mathcal{V}'^{5/3}\right) &= -\mathcal{V}'^{2/3}\frac{\partial}{\partial \rho}\left(\mathcal{V}'Q_s\right) + P_s\mathcal{V}'^{5/3}, &(2.40)\end{aligned}$$

where

$$\begin{aligned} \Gamma_s &= \langle|\nabla\rho|\rangle n_s V_s - \langle|\nabla\rho|^2\rangle D_s \frac{\partial n_s}{\partial \rho}, &(2.41)\\ Q_s &= \langle|\nabla\rho|\rangle n_s T_s\left(V_{K_s} + \frac{3}{2}V_s\right) - \langle|\nabla\rho|^2\rangle n_s \chi_s \frac{\partial T_s}{\partial \rho} - \langle|\nabla\rho|^2\rangle \frac{3}{2} D_s T_s \frac{\partial n_s}{\partial \rho}. &(2.42)\end{aligned}$$

Here, n_s and T_s are the density and temperature of the s-species. In addition, $\langle\ \rangle$ represents the magnetic flux surface average, $\mathcal{V}$ denotes the plasma volume, and $\mathcal{V}' = \mathrm{d}\mathcal{V}/\mathrm{d}\rho$. The equilibrium magnetic field is calculated by VMEC [36], which is a code to calculate the three-dimensional magnetohydrodynamics equilibrium. TASK3D integrates various plasma phenomena (modules) through the transport parameters (V_s, D_s, V_{K_s}, and χ_s) and the source terms (S_s and P_s), as shown in Fig. 2.6.

Coefficients D_s and χ_s are the particle and thermal diffusivities, which are assumed to be given by the sum of turbulent (TB) transport and neoclassical (NC) transport (collisional transport including the effect of magnetically trapped orbits):

$$\begin{aligned} D_s &= D_s^{\mathrm{TB}} + D_s^{\mathrm{NC}}, &(2.43)\\ \chi_s &= \chi_s^{\mathrm{TB}} + \chi_s^{\mathrm{NC}}. &(2.44)\end{aligned}$$

The convection velocities V_s and V_{K_s} are the particle and heat pinch velocities, respectively, and these velocities are assumed to be determined via only neoclassical transport. The neoclassical components of particle and heat fluxes, Γ_s^{NC} and Q_s^{NC}, are written by

$$\Gamma_s^{\mathrm{NC}} = -n_s D_s^{\mathrm{NC}}\left\{\frac{1}{n_s}\frac{\partial n_s}{\partial r} - \frac{q_s E_r}{T_s} + \left(\frac{D_s^{\mathrm{od}}}{D_s^{\mathrm{NC}}} - \frac{3}{2}\right)\frac{1}{T_s}\frac{\partial T_s}{\partial r}\right\}, \tag{2.45}$$

$$Q_s^{\mathrm{NC}} = -n_s T_s D_s^{\mathrm{od}}\left\{\frac{1}{n_s}\frac{\partial n_s}{\partial r} - \frac{q_s E_r}{T_s} + \left(\frac{\chi_s^{\mathrm{NC}}}{D_s^{\mathrm{od}}} - \frac{3}{2}\right)\frac{1}{T_s}\frac{\partial T_s}{\partial r}\right\}, \tag{2.46}$$

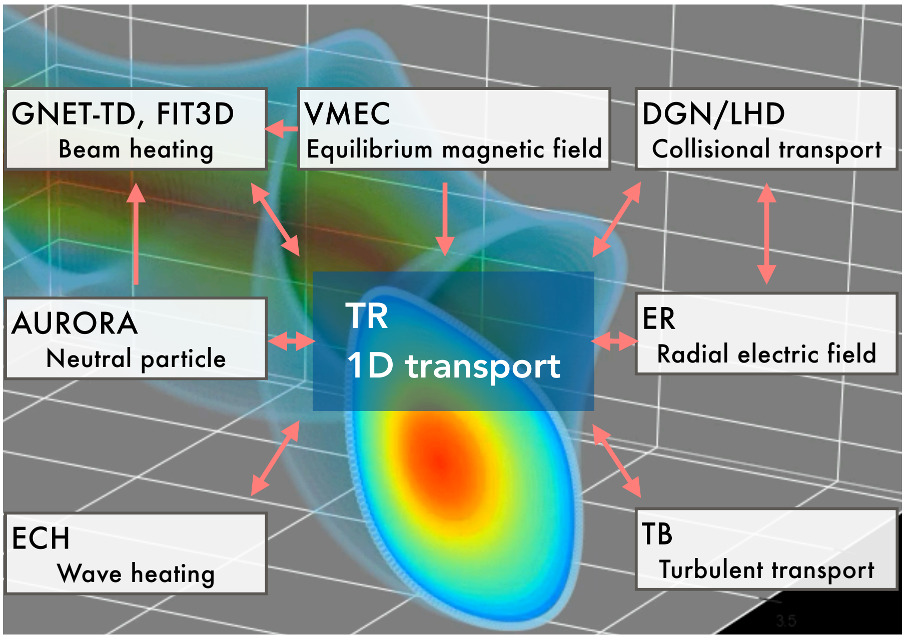


Figure 2.6: An example of module relationships in TASK3D.

where $D_s^{\rm od}$ is the off-diagonal coefficient, and E_r is the radial electric field. Thus, the convection velocities V_s and V_{K_s} are given as follows.

$$V_s = \frac{1}{n_s}\left(\Gamma_s^{\rm NC} + D_s^{\rm NC}\frac{\partial n_s}{\partial r}\right), \tag{2.47}$$

$$V_{K_s} = \frac{1}{n_s T_s}\left\{Q_s^{\rm NC} + n_s D_s^{\rm od}\frac{\partial T_s}{\partial r}\left(\frac{\chi_s^{\rm NC}}{D_s^{\rm od}} - \frac{3}{2}\right)\right\}. \tag{2.48}$$

The neoclassical transport coefficients are evaluated using the DGN/LHD module [37]. This module enables fast estimation of the neoclassical transport parameters using a neural network which regressed a database of the parameters evaluated by the DCOM [38] and GSRAKE [39] codes. In addition, the radial electric field E_r in Eqs. (2.45) and (2.46) is determined by the ambipolar condition in the Er module [40],

$$\Gamma_{\rm e}^{\rm NC} - \sum_{\rm i} Z_i \Gamma_i^{\rm NC} = 0, \tag{2.49}$$

where i denotes ion species, and Z_i is the charge number of ion i.

On the other hand, the turbulent components involve large uncertainties and simple functional models are employed. For NBI heated plasmas in LHD, we employ the constant model for particle diffusion, the gyro-Bohm model for electron thermal diffusion, and the gyro-Bohm+grad$T$ model for ion thermal diffusion, based on the previous study for NBI plasmas in LHD [21,41]:

$$D_s^{\mathrm{TB}} = D_{\mathrm{const}}, \tag{2.50}$$

$$\chi_{\mathrm{e}}^{\mathrm{TB}} = C_{\mathrm{e}} \frac{T_{\mathrm{e}}}{eB} \frac{\rho_{\mathrm{i}}}{a}, \tag{2.51}$$

$$\chi_{\mathrm{i}}^{\mathrm{TB}} = C_{\mathrm{i}} \frac{T_{\mathrm{i}}}{eB} \frac{\rho_{\mathrm{i}}}{a} \left(\frac{\nabla T_{\mathrm{i}}}{T_{\mathrm{i}}} a \right), \tag{2.52}$$

where B, ρ_{i}, and a are the magnetic field strength, the ion Larmor radius, and the plasma minor radius, respectively. Here, the gyro-Bohm model is the heuristic model based on the gyro-Bohm scaling of energy confinement [42, subsection 4.15 and 4.16]. The parameters D_{const}, C_{e}, and C_{i} are constant factors estimated by the best fitting of the experimental results. Based on the previous studies [21,41], we employ the constant factors: $D_{\mathrm{const}} = 0.60$, $C_{\mathrm{e}} = 1.44$, and $C_{\mathrm{i}} = 0.57$.

The S_s and P_s terms are the particle and heat source terms, respectively. The particle source S_s is primarily determined by the ionization of neutral particles evaluated by the AURORA module [21,43] of TASK3D. AURORA calculates the component of S coming from the ionization of neutral particles using the plasma profiles and the density and temperature of neutral particles at the plasma edge. In the case of NBI heated plasma, we should consider the component from the beam ions which slowed down to the thermal velocity. The heat source term P_s comprises the heating power, the power exchange between particle species, and the loss term by interaction with neutrals. In the next subsection, we discuss simulation codes to evaluate the particle source and the heating power in the case of NBI heating.

2.2.2 HFREYA and GNET

Neutral Beam Injection (NBI) heating [42] is a heating method by injecting neutral beams into the plasma. The beam particles slip through the magnetic field and ionize in the plasma. The beam ions collide with thermal particles and heat the plasma. Finally, the beam ions slow down to the thermal velocity and become part of the bulk plasma as thermal ions.

The simulation code of NBI heating [44–48] is indispensable for the transport analysis and prediction of the behavior of NBI-heated plasma [20,49,50]. HFREYA [47] and GNET [48] are simulation codes to simulate the NBI heating and have been developed to analyze energetic particles. These two codes correspond to two physics processes of NBI heating, i.e., the beam ion birth process and the slowing-down process. The HFREYA code computes the birth position and velocity of the beam ions resulting from the beam ion birth process. The GNET code simulates the slowing down process of the beam ions. These codes have been applied to the NBI heating analysis for helical fusion plasmas including LHD [51–53].

The HFREYA code, which is a part of the FIT3D code [47], is used to calculate the NBI beam ion birth profile in a plasma using a Monte Carlo algorithm. The HFREYA code follows the test particles generated at the ion source in the beam injector along their trajectories until they are ionized or go through the plasma. The plasma geometry is introduced using the MHD equilibrium with the VMEC code. Figure 2.7(a) shows an ion birth profile calculated by HFREYA.

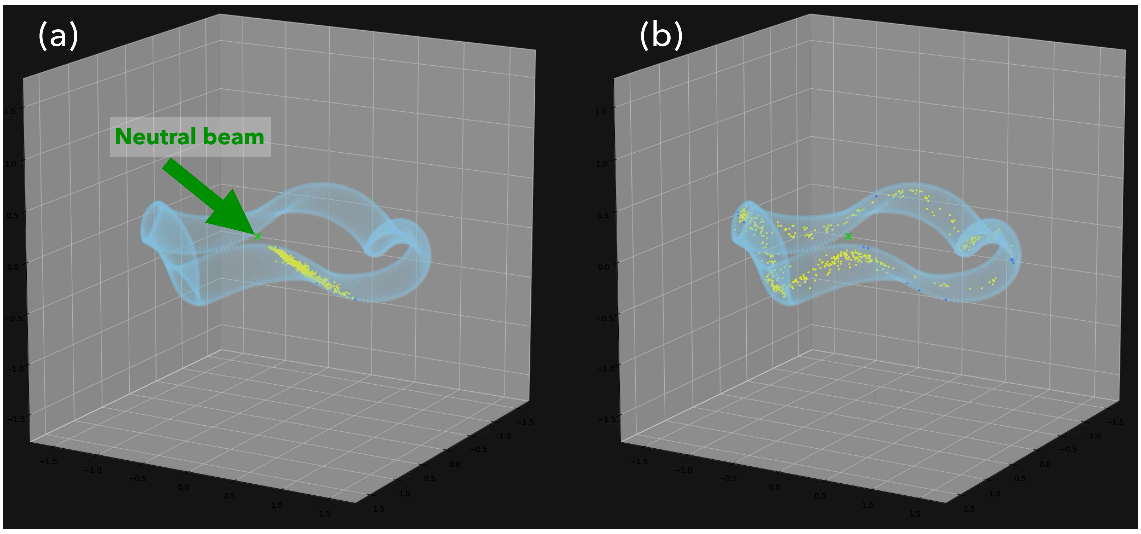


Figure 2.7: (a) An ion birth profile calculated by HFREYA and (b) test particles in the slowing-down process calculated by GNET.

GNET is used to solve the drift kinetic equation in a five-dimensional phase space (3-D in real and 2-D in velocity space),

$$\frac{\partial f}{\partial t} + (\mathbf{v}_{\parallel} + \mathbf{v}_{\mathrm{D}}) \cdot \nabla f + \dot{\mathbf{v}} \cdot \nabla_{\mathbf{v}} f = C(f) + L + S, \tag{2.53}$$

where f is the fast ion distribution function, $\mathbf{v}_{\parallel}$ is the velocity parallel to the magnetic field line, and $\mathbf{v}_{\mathrm{D}}$ is the perpendicular drift velocity. The term $C(f)$ is the linear Coulomb collision operator, L is the particle loss term consisting of the orbit loss and charge exchange loss, and S is the particle source term calculated using the HFREYA code. Similarly to HFREYA, the GNET code is based on the Monte Carlo method where a large number of test particles are followed until they go out of the plasma (the real space boundary) or their kinetic energy reaches the thermal energy (the velocity space boundary). Figure 2.7(b) shows test particles in the slowing-down process calculated by GNET. GNET can evaluate various physical quantities including the power deposition of NBI heating. Furthermore, GNET has been extended to take into account the time evolution of the plasma (GNET- TD [54]).

HFREYA and GNET have made great contributions to the analysis of NBI heating, but has the issue of high computational costs. It takes tens of hours to evaluate the radial profiles of NBI power deposition for a second in actual plasma. Therefore, it is difficult to employ GNET for the analysis of many discharges in a short time. In Section 2.4, we introduce a rapid simulation model of NBI heating.

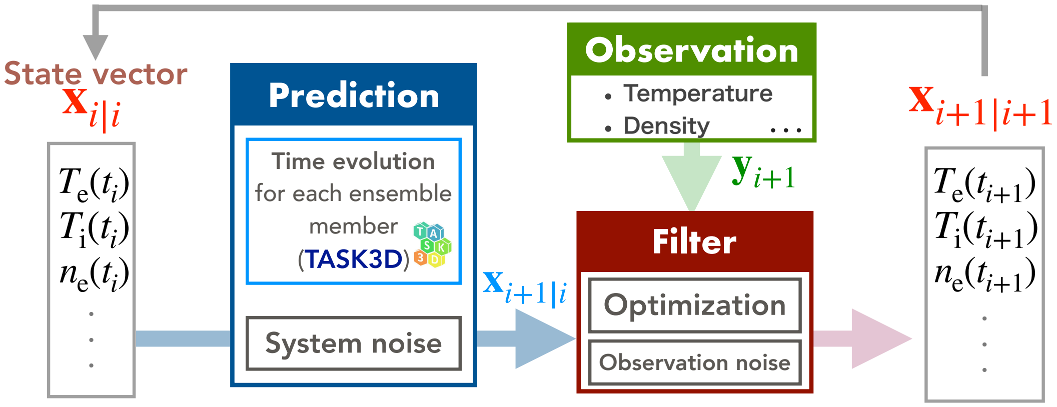


Figure 2.8: Data assimilation process based on the EnKF and TASK3D.

2.3 ASTI as an analysis system

The analysis and prediction by ASTI are based on the sequential Bayesian filter. In this thesis, we focus on the EnKF and EnKS employing TASK3D as the system model as shown in Fig. 2.8. Simpler techniques based on Bayesian methods or KFs have been applied for the analysis of tokamak plasmas [55–57] to make the integrated simulation codes more useful. The EnKF has several advantages for the application to the fusion plasma integrated simulation. For example, the EnKF does not need the linear approximation and is strong for nonlinear systems, while the extended Kalman filter, which is employed in [56,57], requires that and does not always work well for nonlinear systems. Furthermore, the EnKF allows non-gaussian noise distribution. Although the computational cost of EnKF often matters, parallel computing and modern high-performance computing can overcome this problem. In this section, we examine the performance of the EnKF and EnKS in ASTI using simple examples through estimations of the factors of turbulent heat transport models [32].

2.3.1 State-space model

To demonstrate the estimation performance of ASTI, we estimate the factors of the turbulent heat transport models, which can reproduce a time series data-set of plasma temperature, using the EnKF and EnKS. We assume an NBI heated plasma in LHD and consider the uncertainties of temperature T_s, density n_s, the constant factor in the turbulent transport models, C_s, and NBI heat deposition term P_s^{NBI}. Here, we consider two particle species: electron (s = e) and light hydrogen ion (s = i). The state vector is defined as follows:

$$\mathbf{x}_t = (\mathbf{T}_{\mathrm{e},t}^{\mathrm{T}}, \mathbf{T}_{\mathrm{i},t}^{\mathrm{T}}, \mathbf{n}_{\mathrm{e},t}^{\mathrm{T}}, \mathbf{n}_{\mathrm{i},t}^{\mathrm{T}}, \mathbf{C}_{\mathrm{e},t}^{\mathrm{T}}, \mathbf{C}_{\mathrm{i},t}^{\mathrm{T}}, \mathbf{k}_{\mathrm{e},t}^{\mathrm{T}}, \mathbf{k}_{\mathrm{i},t}^{\mathrm{T}})^{\mathrm{T}}. \tag{2.54}$$

Every state variable is defined on 60 computational grid points (radial direction, 1D) and has a similar structure as follows:

$$\mathbf{T}_{\mathrm{e}} = (T_{\mathrm{e}}^1, T_{\mathrm{e}}^2, \cdots, T_{\mathrm{e}}^{60})^{\mathrm{T}}. \tag{2.55}$$

Table 2.1: Rates of the standard deviation of the initial distribution and system noise.

State variable	Initial distribution	System noise
T_e	5%	10%
T_i	5%	10%
n_e	5%	10%
n_i	5%	10%
C_e	20%	20%
C_i	20%	20%
k_e	5%	5%
k_i	5%	5%

The superscript on each element in Eq. (2.55) indicates the grid point. In Eq. (2.54), $\mathbf{T}, \mathbf{n}, \mathbf{C}$, and $\mathbf{k}$ denote the temperature, density, numerical factor of the turbulence models in Eqs. (2.51) and (2.52), and the factor for the NBI heat deposition, with i and e for ion and electron, respectively.

Although TASK3D can solve the time evolution of the particle density and the temperature, in this section, TASK3D solves only the time evolution of the temperature. In other words, we investigate cases where the temporal variation of density is relatively small. The observations of electron and ion densities have the same values, that is, $n_\mathrm{e} = n_\mathrm{i}$. The density is assumed to be stationary in the prediction step but is updated in the filtering step. Since factor C_s is assumed to be grid-dependent, as Eq. (2.55), C is allowed to have spatial variation. The factor k_s^i is defined to adjust the NBI heat deposition evaluated by GNET-TD. The following P_s^* is used instead of P_s in Eq. (2.40):

$$P_\mathrm{s}^{*i} = k_\mathrm{s}^i P_\mathrm{s}^i. \tag{2.56}$$

The initial ensemble mean is set to the observed data for T_s and n_s, and to the conventional values for C_s and k_s in Eqs. (2.51) and (2.52) ($k_\mathrm{e} = 1$, and $k_\mathrm{i} = 1$). The initial standard deviation of the ensemble and the system noise are set to be proportional to the ensemble mean, and their rates are listed in Table 2.1. The noise of k_s is assumed to be smaller than that of C_s. This means that the NBI heat deposition model (GENT-TD) is more reliable than the turbulent transport models. This assumption distinguishes the roles of k_s and C_s, and allows stable estimation of these terms. The standard deviation of observation noise is assumed to be proportional to the difference between the prediction and the observation data [58] (See Section 3.1 for details). The covariance matrix of observation noise, R_t, is estimated before every filtering step. We employ the rate of 0.8.

2.3.2 Settings of observation data

For the assessment of the estimation by the EnKF and EnKS, assimilation experiments are performed for four time series data-sets of temperature and density simulated by TASK3D (Cases 1 through 4). The simulated observation data are generated by TASK3D for four different time-space distributions of factor C_e and C_i. The noise

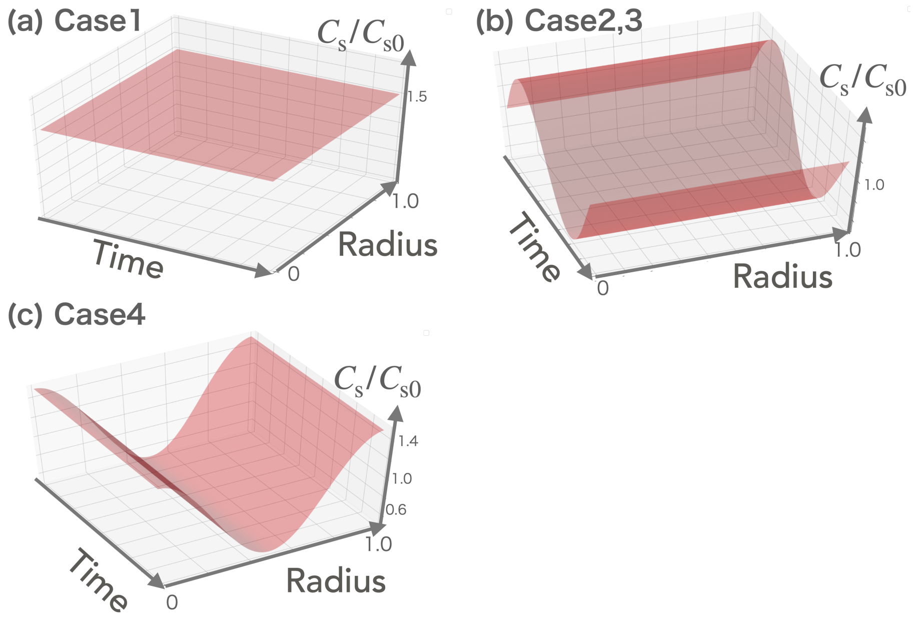


Figure 2.9: Time-space distributions of $C_\mathrm{e}/C_\mathrm{e0}$ and $C_\mathrm{i}/C_\mathrm{i0}$ for Cases 1 through 4 in Table 2.2.

Table 2.2: Setup of data assimilation experiments

Case	Observation data	True $C_\mathrm{s}/C_\mathrm{s0}$	τ_DA[ms]
1	Simulated data	constant : 1.5	80
2	Simulated data	$1 + 0.5 \times \sin(2\pi t/1.0)$	80
3	Simulated data	$1 + 0.25 \times \sin(2\pi t/0.1)$	160
4	Simulated data	$1 + 0.5 \times \cos(2\pi\rho)$	80

generated from the Gaussian distribution, whose mean is zero and standard deviation is 5% of the simulated value, is added to the simulated data as a measurement error. In all cases, the NBI heat deposition precalculated by GNET-TD for LHD discharge (No. 114053) is used.

In Case 1, C_e and C_i are constants and $C_s/C_{s0} = 1.5$, as shown in Fig. 2.9(a), where C_{s0} denotes the conventional values of C_e and C_i. In Case 2, C_e and C_i have the temporal variation $C_s/C_{s0} = 1 + 0.5 \times \sin(2\pi t/1.0)$, as in Fig. 2.9(b), where t is the time in seconds. This case corresponds to slow temporal variation compared with the cycle of assimilation $\tau_\mathrm{DA} = 80$ ms. In Case 3, C_e and C_i have the temporal variation $C_s/C_{s0} = 1 + 0.25 \times \sin(2\pi t/0.1)$. This case corresponds to fast temporal variation compared with the cycle of assimilation $\tau_\mathrm{DA} = 160$ ms. In Case 4, C_e and C_i have the spatial variation $C_s/C_{s0} = 1 + 0.5 \times \cos(2\pi\rho)$ as in Fig. 2.9(c), where ρ is the normalized minor radius. These assimilations are performed with 2000 ensemble members for $\tau_\mathrm{DA} = 80$ ms (Cases 1, 2, and 3) and $\tau_\mathrm{DA} = 160$ ms (Case 4). A summary of these assimilations are shown in Table 2.2.

2.3.3 Assimilation results

The temporal changes of filtered estimates and smoothed estimates of C_s/C_{s0} in Cases 1 through 4 are shown in Fig. 2.10. In Case 1, the filtered estimates approach the true values within the first few times of assimilation and stay around the true values. In this case, it is found that the EnKS estimates do not differ much form those of the EnKF, because the true values of C_e and C_i are stationary. In Case 2, it seems that the EnKF and EnKS can follow the slow temporal change of C_s, however, the estimates of EnKF have 0.04-0.08 s delays from the true values (for example, at $t = 0.96$, 1.04, and 1.12 in both $C_\mathrm{e}/C_\mathrm{e0}$ and $C_\mathrm{i}/C_\mathrm{i0}$). The EnKS correcting the time delays, almost all the smoothed estimates are close to the true curve lines. In Case 3, the estimates, especially by the EnKS, are around $C_s/C_{s0} = 1$. The EnKS corrects the filtered estimates toward the average of the C_s/C_{s0} temporal change, $C_s/C_{s0} = 1$. This indicates that the EnKS follows the temporal trend of C_s in the time scale longer than the assimilation cycle. In Case 4, the smoothed estimates reproduce the true spatial variation of C_s/C_{s0} and stay around the true values. We have confirmed that the estimates of EnKS do not differ much form those of the EnKF for the same reason as Case 1.

Figure 2.11 shows the TASK3D simulation results of T_e and T_i using the smoothed estimates of C_s and k_s (dashed lines labeled 'TASK3D*') for Cases 1 to 4. In all cases, the results of both T_e and T_i agree reasonably with the observations. These results indicate the validity of the estimation by the EnKS.

Figure 2.12 shows the smoothed estimates of the radial profiles of C_s/C_{s0} and χ_s^TB in Cases 1 and 4 at time 1.0 s. The smoothed estimates around the center ($\rho < 0.2$) have larger uncertainties than in the other region ($\rho > 0.2$). This indicates that the influence of C_s around the center region on the radial profiles of temperature is smaller than that of C_s in the other region because the temperature gradients around the center tend to zero. We can also see the large errors between the true values and the smoothed estimates of $C_\mathrm{i}/C_\mathrm{i0}$ around $\rho = 1$ in Cases 1 and 4. It is considered that the boundary condition in TASK3D affects the EnKF and EnKS estimate. Since the Dirichlet boundary condition is employed to solve Eq. (2.40) in TASK3D, the temperature around $\rho = 1$ is determined independently of χ_s. However, the estimates of C_s and χ_s^TB in $0.2 < \rho < 0.8$ agree well with the true values.

The smoothed estimate of C_s includes information that should be considered in turbulent transport models to predict the temperature of plasma more accurately. If the smoothed estimates of C_s for various plasmas can be reproduced by a regression model, a more valid turbulent transport model can be obtained with relevant physical interpretation.

We confirmed that the EnKS can estimate the radial profile of C_s with high accuracy for $0.2 < \rho < 0.8$ and that temporal trend of C_s, in a time scale longer than the assimilation cycle, from the assimilation of the simulated data sets. Furthermore, the TASK3D simulation using the smoothed estimates has reproduced the experimental temperature data with high accuracy. These results indicate the effectiveness and validity of the EnKF and EnKS approach for accurate estimation of plasma parameters and the possibility of advanced turbulence modeling using the smoothed estimates of C_s for various plasmas.

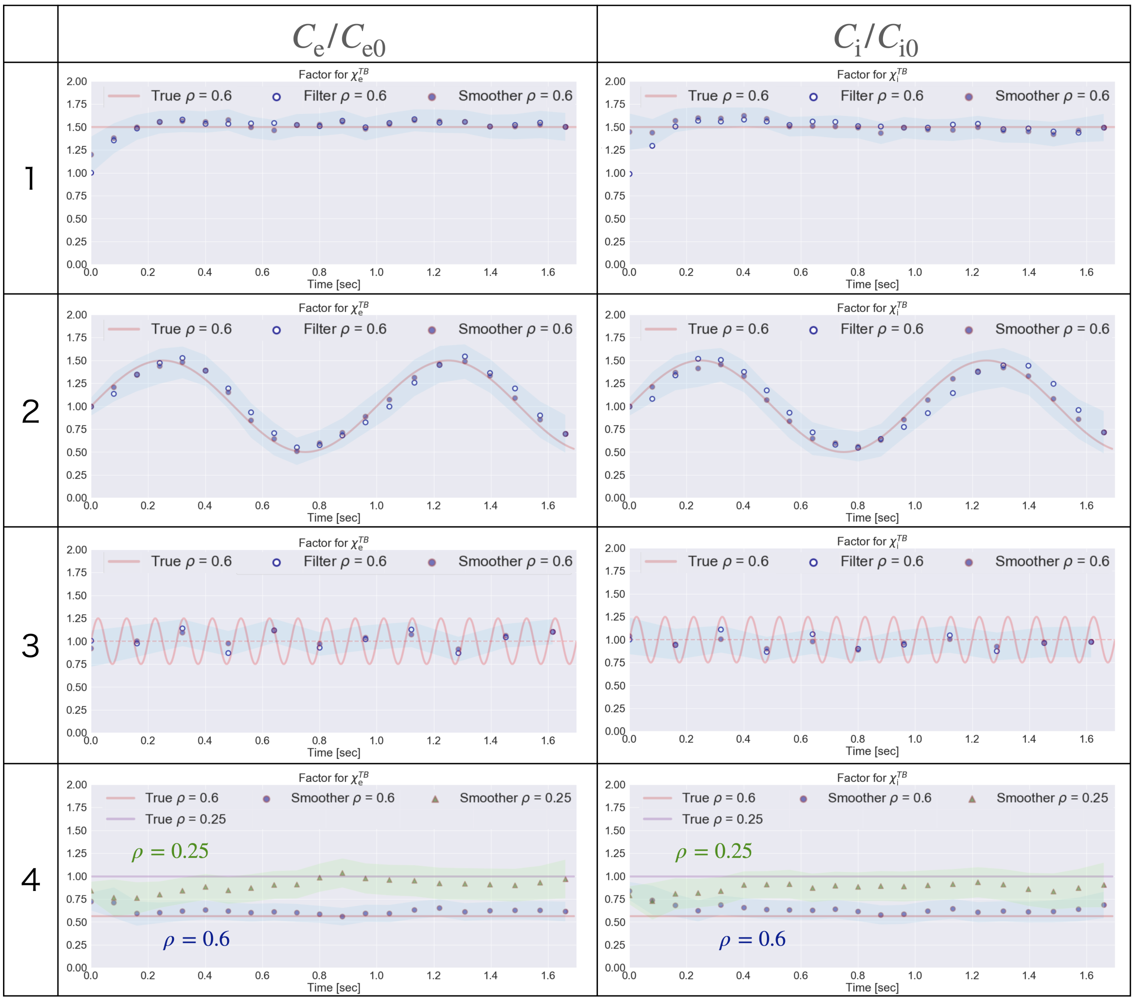


Figure 2.10: Assimilation results of temporal change of C_e/C_{e0} and C_i/C_{i0} in Cases 1 through 4. The highlighted areas around the smoothed estimates represent the standard deviations of the smoothed ensemble.

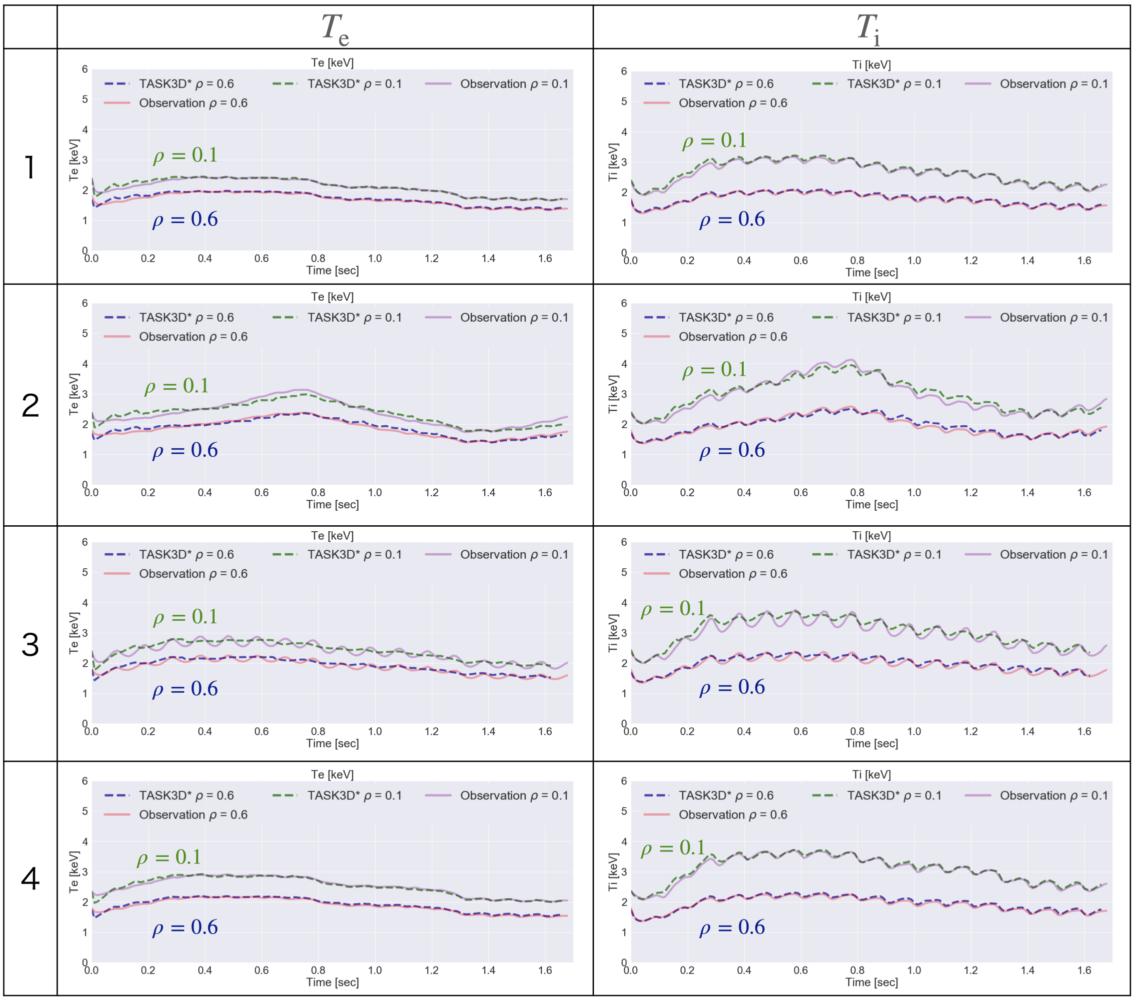


Figure 2.11: TASK3D simulation results of T_e and T_i using the smoothed estimates of C_s and k_s for $\rho = 0.1$ and $\rho = 0.6$ in Cases 1 through 4.

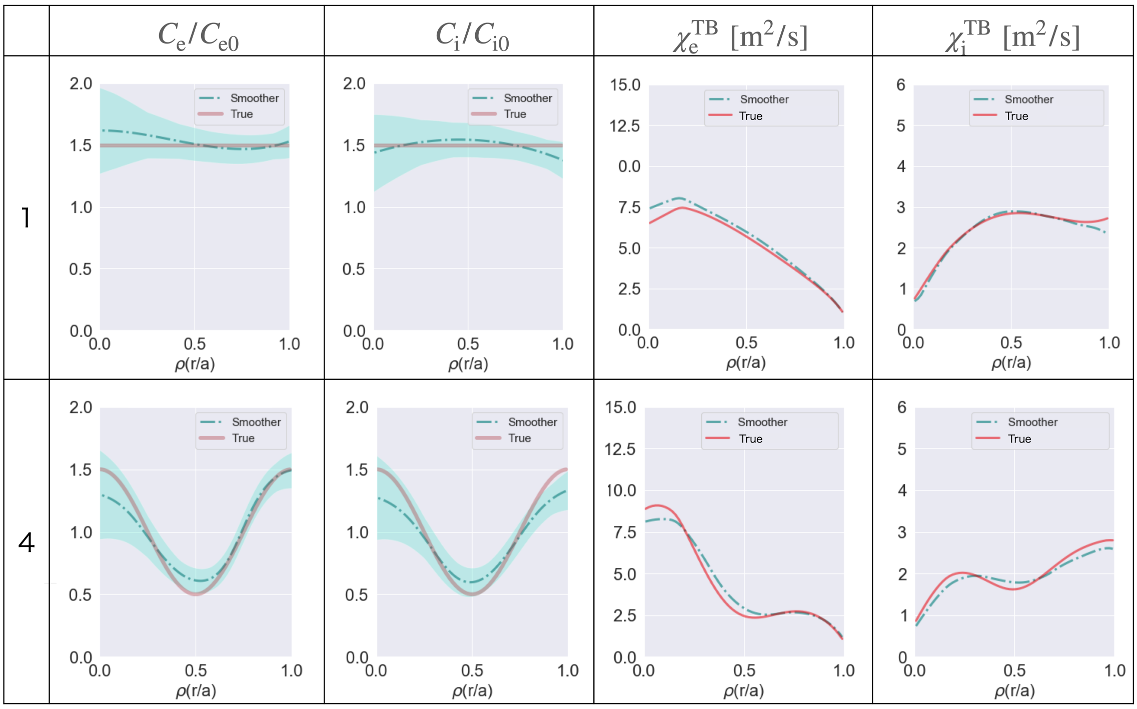


Figure 2.12: Smoothed estimates of radial profile of C_e/C_{e0},C_i/C_{i0},χ_e^{TB} and χ_i^{TB} at time 1.0 s for Cases 1 and 4. The highlighted areas around the smoothed estimates represent the standard deviations of the smoothed ensemble.

2.4 Rapid simulation of NBI heating

Neutral beam injection (NBI) heating is one of the effective heating methods for fusion devices including LHD. The simulation code of NBI heating is indispensable for the transport analysis and prediction of the behavior of NBI-heated plasma, as described in Section 2.2.2. However, in general, the simulation codes (for example, GNET) require high computational costs to evaluate various physical quantities related to NBI heating including the power deposition.

For easy simulation of NBI heating, as a simplified version of GNET, FIT3D [19,47, 59] has also been developed. This code calculates the beam ion birth profile considering the prompt orbit effect by the Monte Carlo method and evaluates the power deposition profiles in the steady state using the simple analytical solution of the Fokker-Planck equation. FIT3D has been extended to evaluate the time evolution considering the beam slowing-down effect (conv-FIT3D [60]). The conv-FIT3D code can calculate the time evolution of the power deposition profiles for a second of actual plasma in tens of minutes. However, even applying conv-FIT3D, it is difficult to predict the plasma heating at the real-time level and to perform many NBI heating simulations for the heating scenario optimization.

In the system model of ASTI, GNET and FIT3D have been employed as the NBI heating model. However, even FIT3D takes at least 10 minutes for a second in actual plasma. Thus, a faster calculation model of the NBI heating is required to simulate plasma behavior at the real-time level. Of course, the uncertainty contained in the NBI heating model can increase as the calculation is simplified. However, the uncertainty inherent in the NBI heating model can be compensated by adjusting the model parameters through data assimilation. It is expected that high-speed and accurate simulation of NBI-heated plasma is achieved by combining the reduced model and data assimilation.

In this section, we introduce a rapid simulation code of NBI heating, named ”FIT3D-RC” [61], to evaluate the NBI power deposition. This code estimates the beam ion birth profile using the Gaussian process regression (GPR) model applied to precomputation results by the Monte Carlo simulation (HFREYA and MCNBI [47]) and calculates the slowing-down process of the beam ion using the analytical solution of the Fokker-Planck equation. Figure 2.13 shows an overview of the calculation process in FIT3D-RC. We describe the details of the developed code, FIT3D-RC, in Section 2.4.1 and demonstrate the high-speed calculation of the NBI power profile while keeping the accuracy level of the previous code (FIT3D) in Section 2.4.2.

2.4.1 FIT3D-RC

FIT3D-RC is based on the FIT3D models. The high computational cost of the previous codes is mainly due to using the Monte Carlo method throughout or in part of the simulation. The Monte Carlo calculation for the beam ion birth profile in FIT3D, which accounts for most of the computational cost in FIT3D, is replaced by the GPR prediction in FIT3D-RC. FIT3D-RC consists of two parts: one is the part of evaluating the beam ion birth profile, called ”ion-birth part”, and another is the part of calculating the slowing-down process of the beam ion, called ”slowing-down part”.

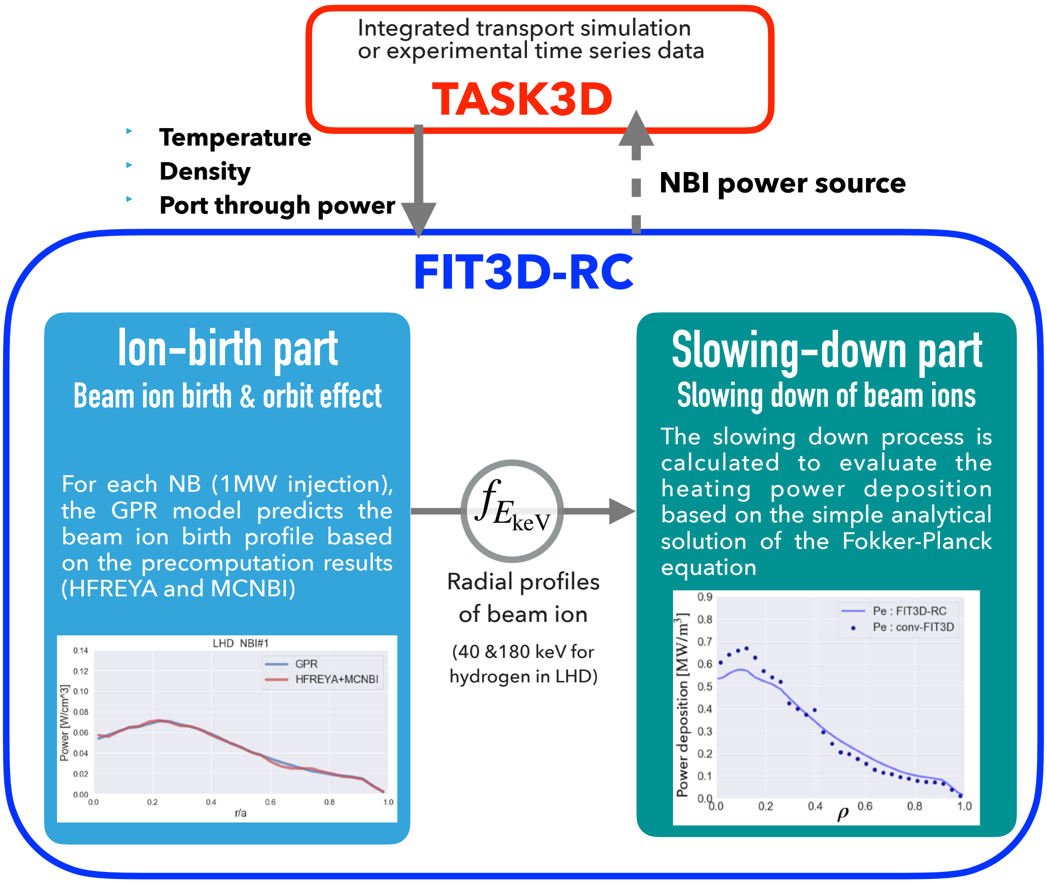

Integrated transport simulation
or experimental time series data
TASK3D
Temperature
Density
Port through power
NBI power source
FIT3D-RC
Ion-birth part
Beam ion birth & orbit effect
For each NB (1MW injection), the GPR model predicts the beam ion birth profile based on the precomputation results (HFREYA and MCNBI)
LHD NBI#1
GPR
HFREYA+MCNBI
Power [W/cm^3]
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0.0
0.2
0.4
0.6
0.8
1.0
r/a
$f_{E_{\mathrm{keV}}}$
Radial profiles
of beam ion
(40 &180 keV for
hydrogen in LHD)
Slowing-down part
Slowing down of beam ions
The slowing down process is calculated to evaluate the heating power deposition based on the simple analytical solution of the Fokker-Planck equation
Pe : FIT3D-RC
Pe : conv-FIT3D
Power deposition [MW/m^3]
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.0
0.2
0.4
0.6
0.8
1.0
ρ

Figure 2.13: An overview of the calculation process in FIT3D-RC.

Beam ion birth (ion-birth part)

The ion-birth part evaluates the beam ion birth profile for NBI using the GPR [62]. HFREYA and MCNBI calculate the beam ion birth process and the prompt orbit effect, respectively. These codes have been validated with the LHD experimental results, e.g., the neutron emission data [63]. We assume that the birth profile is determined depending on the radial profiles of electron density and temperature. Representing the density and temperature radial profiles in polynomial form:

$$g(\rho) = (g_0 - g_1)(1 - \rho^a) + g_1, \tag{2.57}$$

the sample data used for the regression are generated by the HFREA and the MCNBI code for 360 patterns of electron density and temperature profiles (20 (n_{e}) × 18 (T_{e}) = 360): $n_{\mathrm{e},0} \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ $[10^{19}\mathrm{m}^{-3}]$, $n_{\mathrm{e},1} = 0.5$ $[10^{19}\mathrm{m}^{-3}]$, $a^{n_{\mathrm{e}}} \in \{4, 8\}$, $T_{\mathrm{e},0} \in \{1, 2, 3, 4, 5, 6\}$ [keV], $T_{\mathrm{e},1} = 0.5$ [keV], and $a^{T_{\mathrm{e}}} \in \{2, 4, 8\}$ as shown in Fig. 2.14. It seems that using a regression method other than GPR, such as neural network, also works well. Since the beam ion birth profile does not change in a complicated manner with change in the radial profile of plasma density and temperature, we employ GPR, which is a relatively simple regression method. The cost of predictive calculation by

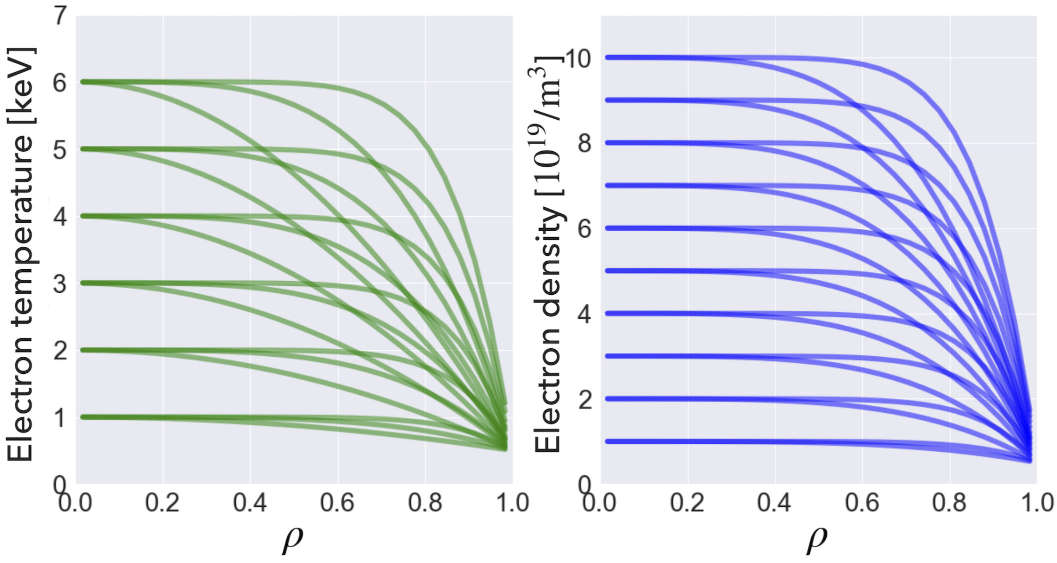


Figure 2.14: The profiles of electron temperature and density employed to generate the sample data for the GPR.

the GPR often matters; however, the cost with 360 samples and an input dimension of about 10 is negligible. The GPR predicts a Gaussian distribution, i.e., can estimate the uncertainties of the prediction and compatible with data assimilation.

The GPR model predicts the beam ion birth profile (in this study, 30 radial grid points) due to 1 MW of injection (port-through power), and the input vector $\mathbf{x}$ consists of electron density and temperature profiles defined on 6 radial points (ρ=0, 1/6, 2/6, 3/6, 4/6, 5/6):

$$\mathbf{x} = (\tilde{n}_{\mathrm{e},\rho=0}, \cdots, \tilde{n}_{\mathrm{e},\rho=\frac{5}{6}}, \tilde{T}_{\mathrm{e},\rho=0}, \cdots, \tilde{T}_{\mathrm{e},\rho=\frac{5}{6}}). \tag{2.58}$$

The input dimension is 12. The components with the superscript $\tilde{}$ denote the values normalized by the mean and standard deviation of the samples, i.e., $\tilde{x}_i^j = (x_i^j - \bar{x}_i)/\sigma_i$, where x_i^j is the i-th component of the j-th sample, $\bar{x}_i$ is the mean of x_i^j, and σ_i is the standard deviation of x_i^j. The output is also normalized by the same way. The kernel function is defined by the radial basis function:

$$k(\mathbf{x}, \mathbf{x}') = V \exp\left(-\frac{(\mathbf{x} - \mathbf{x}')^2}{2L^2}\right), \tag{2.59}$$

where $\mathbf{x}$ and $\mathbf{x}'$ are the input vectors, and V and L are the parameters that control the kernel function. Considering the Gaussian noise with zero mean and variance σ^2 on the observed values (the beam ion birth profiles calculated by HFREYA-MCNBI), the GPR model has three hyper parameters, V, L, and σ^2. These parameters are optimized by the Scaled Conjugate Gradient (SCG) method [64].

In LHD, three negative ion based NBI heating systems (NB#1-3, beam energy of 180 keV) and two positive-ion ones (NB#4 and 5, beam energy of 40 keV for hydrogen) are installed [1]. Therefore, for each beam, we build the GPR model to predict the birth profile for 1 MW injection.

For each beam energy, the beam ion birth profile caused by the beam injection at time t' can be evaluated as

$$f_{E_\mathrm{b}}^{t'}(\rho) = \sum_k H_k^{t'} W_k^{t'}(\rho), \tag{2.60}$$

where the superscript t' denotes the injection timing and k is the index of beam whose energy is E_b. In addition, $f_E(\rho)$ is the birth profile of beam ion with energy E_b, H_k is the port-through power [MW] of NB#k, and $W_k(\rho)$ is the birth profile for 1 MW injection of NB#k which the GPR model predicts. In the case of LHD, Eq. (2.60) is written as

$$f_{180\mathrm{keV}}^{t'}(\rho) = \sum_{k\in\{1,2,3\}} H_k^{t'} W_k^{t'}(\rho), \tag{2.61}$$

$$f_{40\mathrm{keV}}^{t'}(\rho) = \sum_{k\in\{4,5\}} H_k^{t'} W_k^{t'}(\rho). \tag{2.62}$$

Slowing down of the beam ions (slowing-down part)

In this part, the slowing down of beam ions is calculated to evaluate the heating power deposition based on the simple analytical solution of the Fokker-Planck equation [65] in common with FIT3D and conv-FIT3D. The slowing down of the beam ion in a plasma from $v(t)$ to $v(t+\Delta t)$ is deduced from

$$v(t+\Delta t) = \left\{ v^3(t) e^{-\frac{3\Delta t}{\tau_s}} - v_\mathrm{c}^3 \left(1 - e^{-\frac{3\Delta t}{\tau_s}}\right)\right\}^{\frac{1}{3}}, \tag{2.63}$$

$$\tau_s\ [\mathrm{sec}] = 0.12 \frac{(T_\mathrm{e}[\mathrm{keV}])^{\frac{3}{2}}}{Z_\mathrm{f}^2 n_\mathrm{e}[10^{19}\mathrm{m}^{-3}]} \frac{m_\mathrm{f}}{m_\mathrm{p}}, \tag{2.64}$$

where $v(t)$ is the beam ion (beam ion) velocity at time t, and τ_s is the slowing-down time. The quantities Z_f, m_f, and m_p are the charge and mass of the beam ion, and the proton mass, respectively. In addition, v_c is the critical velocity calculated from the critical energy E_c by $v_\mathrm{c} = \sqrt{(2E_\mathrm{c})/m_\mathrm{f}}$. The critical energy is given by

$$E_\mathrm{c}[\mathrm{keV}] = T_\mathrm{e}[\mathrm{keV}] \left(\frac{9\pi}{16}\frac{m_\mathrm{f}}{m_\mathrm{e}}\right)^{\frac{1}{3}} \left(\frac{m_\mathrm{f}}{m_\mathrm{p}} Z_1\right)^{\frac{2}{3}}, \tag{2.65}$$

and

$$Z_1 = \sum_i \frac{z_i^2 n_i}{A_i n_\mathrm{e}}, \tag{2.66}$$

where m_e is the mass of electron, z_i is the ion charge, and A_i is the ion mass number. The power deposition to electron, $P_\mathrm{e}^\mathrm{NBI}$, and that to ion, $P_\mathrm{i}^\mathrm{NBI}$, at time t can be calculated by

$$P_\mathrm{e}^\mathrm{NBI}(\rho) = \sum_{E_\mathrm{b}} \sum_{t'\in S_{E_\mathrm{b},\rho}} G_{\mathrm{e},E_\mathrm{b}}^{t'}(\rho) \times f_{E_\mathrm{b}}^{t'}(\rho) \tag{2.67}$$

and

$$P_\mathrm{i}^\mathrm{NBI}(\rho) = \sum_{E_\mathrm{b}} \sum_{t'\in S_{E_\mathrm{b},\rho}} G_{\mathrm{i},E_\mathrm{b}}^{t'}(\rho) \times f_{E_\mathrm{b}}^{t'}(\rho). \tag{2.68}$$

Here, $S_{E_\mathrm{b},\rho}$ is the set of the injection times of beam ions with higher energy than E^* at time t. E^* is the threshold at which the beam ion is considered to have finished slowing down to the thermal velocity. The fraction of power deposition added to the electron, $G^{t'}_{\mathrm{e},E_\mathrm{b}}$, and that to the ion, $G^{t'}_{\mathrm{i},E_\mathrm{b}}$, during Δt are calculated from

$$G^{t'}_{\mathrm{e},E_\mathrm{b}}(\rho) = \frac{2}{v^2_{0,E_\mathrm{b}}} \int^{v_{t',E_\mathrm{b}}(t,\rho)}_{v_{t',E_\mathrm{b}}(t+\Delta t,\rho)} \frac{v^4}{v^3 + v^3_\mathrm{c}(\rho)} p_\mathrm{cx}(v)\mathrm{d}v \tag{2.69}$$

and

$$G^{t'}_{\mathrm{i},E_\mathrm{b}}(\rho) = \frac{2}{v^2_{0,E_\mathrm{b}}} \int^{v_{t',E_\mathrm{b}}(t,\rho)}_{v_{t',E_\mathrm{b}}(t+\Delta t,\rho)} \frac{v^3_\mathrm{c} v}{v^3 + v^3_\mathrm{c}(\rho)} p_\mathrm{cx}(v)\mathrm{d}v, \tag{2.70}$$

where $v_{t',E_\mathrm{b}}(t,\rho)$ is the beam ion velocity injected at time t' ($t > t'$), and v_{0,E_b} is the velocity corresponding to the energy E_b, i.e., $v_{t',E_\mathrm{b}}(t',\rho) = v_{0,E_\mathrm{b}}$. The $p_\mathrm{cx}(v)$ is the probability that a beam ion will slow down to velocity v without charge exchange on the background neutral density n_n, written as

$$p_\mathrm{cx}(v) = \left(\frac{v^3_{0,E_\mathrm{b}} + v^3_\mathrm{c}}{v^3 + v^3_\mathrm{c}}\right)^{-\tau_\mathrm{s}/3\tau_\mathrm{cx}}, \tag{2.71}$$

$$\tau_\mathrm{cx} = \frac{1}{n_\mathrm{n} v_{0,E_\mathrm{b}} \sigma_\mathrm{cx}}, \tag{2.72}$$

where σ_cx is the charge exchange cross section at the initial beam ion velocity, v_{0,E_b}.

The calculation procedure to evaluate the power deposition profiles of NBI heating at time t is summarized as follows:

(a) Initialize the power deposition profiles, $P^\mathrm{NBI}_\mathrm{e}$ and $P^\mathrm{NBI}_\mathrm{i}$, to zero.

(b) Calculate the beam ion birth profiles for each beam energy, $f^t_{E_\mathrm{b}}(\rho)$, by Eq. (2.60) and add the time t into the set $S_{E_\mathrm{b},\rho}$.

(c) Update the beam ion velocities for the time step Δt, $v_{t',E_\mathrm{b}}(t,\rho) \to v_{t',E_\mathrm{b}}(t+\Delta t,\rho)$ calculating the slowing-down process represented by Eq. (2.63), for all radial grids (ρ positions), beam energies E_b, and injection times t' in the set $S_{E_\mathrm{b},\rho}$.

(d) Calculate the power deposition profiles, $P^\mathrm{NBI}_\mathrm{e}$ and $P^\mathrm{NBI}_\mathrm{i}$, using Eqs. (2.67) and (2.68).

(e) If $v_{t',E_\mathrm{b}}(t,\rho) \leq v^*$, where $v^* = \sqrt{(2E^*)/m_\mathrm{f}}$, remove the injection time t' from the set $S_{E_\mathrm{b},\rho}$.

The set $S_{E_\mathrm{b},\rho}$ and the velocities v_{t',E_b} at the start of simulation are determined by slowing-down calculation of the beam ions injected before the start, using the profiles of density and temperature at the start of simulation.

2.4.2 Performance of FIT3D-RC

We apply FIT3D-RC to an NBI-heated plasma of LHD. The major radius of the magnetic axis is $R_{\mathrm{ax}} = 3.6$ m and the field strength at the plasma centre is $B_0 = 2.85$ T. The equilibrium magnetic field is calculated by VMEC. In this study, the resulting bootstrap and beam currents are not included. In addition, it is assumed that the magnetic field and the background plasma profiles do not change in the HFREYA-MCNBI simulation.

Table 2.3 shows the values of the GPR hyper parameters (V, L, σ^2) optimized by the SCG method. In addtion, Table 2.4 shows the coefficient of determination (R^2), the root mean square error (RMSE), and the root mean square percentage error (RMSPE) of the ion-birth part. Figure 2.15 shows the comparison of the beam ion birth profile [MW/m^3] between the HFREYA-MCNBI and the GPR results for NB#1, 3, and 5. In Table 2.4, R^2 of NB#1-3 reach around 0.98, and R^2 of NB#4 and 5 get around 0.9. The regression models reproduce the simulation results with high accuracy. The RMSPE is 5∼7 % for NB#1-3 and about 10 % for NB#4 and 5. These differences include calculation errors of the Monte Carlo method by HFREYA-MCNBI. Since the values of the power deposition in NB#4 and 5 are lower than NB#1-3, it is considered that the RMSPE of NB#4 and 5 are evaluated more greatly. This difference between NB#1-3 and NB#4 and 5 also seems to be related to the difference in the beam energy and the injection angle: tangential for NB#1-3 and perpendicular for NB#4 and 5.

Table 2.3: The hyper parameters of the GPR model in the ion-birth part optimized by the SCG method.

NB#	V	L	σ^2
1	10.50	5.590	0.06970
2	14.64	5.695	0.09003
3	12.94	5.741	0.06980
4	7.779	5.305	0.1612
5	10.49	5.758	0.1782

Table 2.4: The coefficients of the determination (R^2), the root mean square error (RMSE), and the root mean square percentage error (RMSPE) of the GPR models for the beam ion birth profile.

NB#	R^2	RMSE [MW/m^3]	RMSPE [%]
1	0.976	0.00270	7.07
2	0.990	0.00359	5.08
3	0.982	0.00290	5.57
4	0.903	0.00316	9.92
5	0.910	0.00298	10.37

Figure 2.16 shows the prediction results of the birth profiles for typical radial profiles

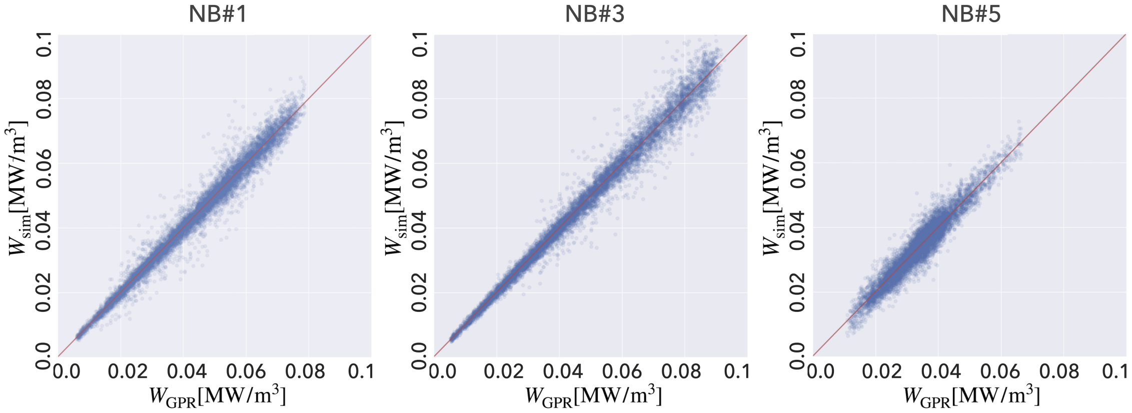


Figure 2.15: The comparison of the beam ion birth profile ($W(\rho)$ in Eq. (2.60) for all radial grid points) between the HFREYA-MCNBI simulation, W_{sim}, and the GPR prediction, W_{GPR}, for NB#1 (left), NB#3 (middle), NB#5 (right)

of temperature and density. Case 1 is a low density and high temperature case, Case 2 is a medium density and temperature case, and Case 3 is a high density and low temperature case. The profiles shown in Fig. 2.16 are $f_{180\mathrm{keV}}$ and $f_{40\mathrm{keV}}$ given by Eqs. (2.61) and (2.62) for $H_1 = H_2 = H_3 = H_4 = H_5 = 1$ [MW]. It seems that, in all cases, the predicted profiles agree well with the calculation results by HFREYA-MCNBI.

Next, we demonstrate an application of FIT3D-RC to experimental time series data of NBI-heated plasma in LHD (shot number: 117100). Figure 2.17 shows the time evolution of port-through power of NBI, and Fig. 2.18 shows the measured time-series data of density and temperature as the result of the heating. The experimental time series data are fitted by a polynomial with 8th order (only even degrees) in the radial direction and linearly interpolated in the time direction, for use as the background plasma information. This simulation starts at 4.1 s, and we set $\Delta t = 10$ [ms] and $E^* = 0$ [keV].

Figure 2.19 shows the calculated radial profiles of power deposition at 3 timings: t =4.17, 4.63, and 5.00. For comparison, the calculation results by conv-FIT3D [60] are also shown in the same figures. The conv-FIT3D code has been developed based on the FIT3D code and can evaluate the NBI power deposition and induced momentum considering beam slowing-down effect. In Fig. 2.19, it can be seen that the deposition profiles calculated by FIT3D-RC are in good agreement with that by conv-FIT3D, although there are slight errors. These errors are at a level that is not problematic for prompt transport analysis and the data assimilation.

In these cases, the calculation using FIT3D-RC takes 10 seconds for a second in actual plasma ($\Delta t = 10$ [ms] and 30 radial grids), using single core of a standard laptop (2.9 GHz Intel Core i5). The calculation time has been significantly reduced compared to the previous codes: GNET-TD taking dozens of hours, and conv-FIT3D taking dozens of minutes. Calculations at the real-time level are also possible by taking a larger time interval and a larger radial grid. FIT3D-RC is expected to be used for

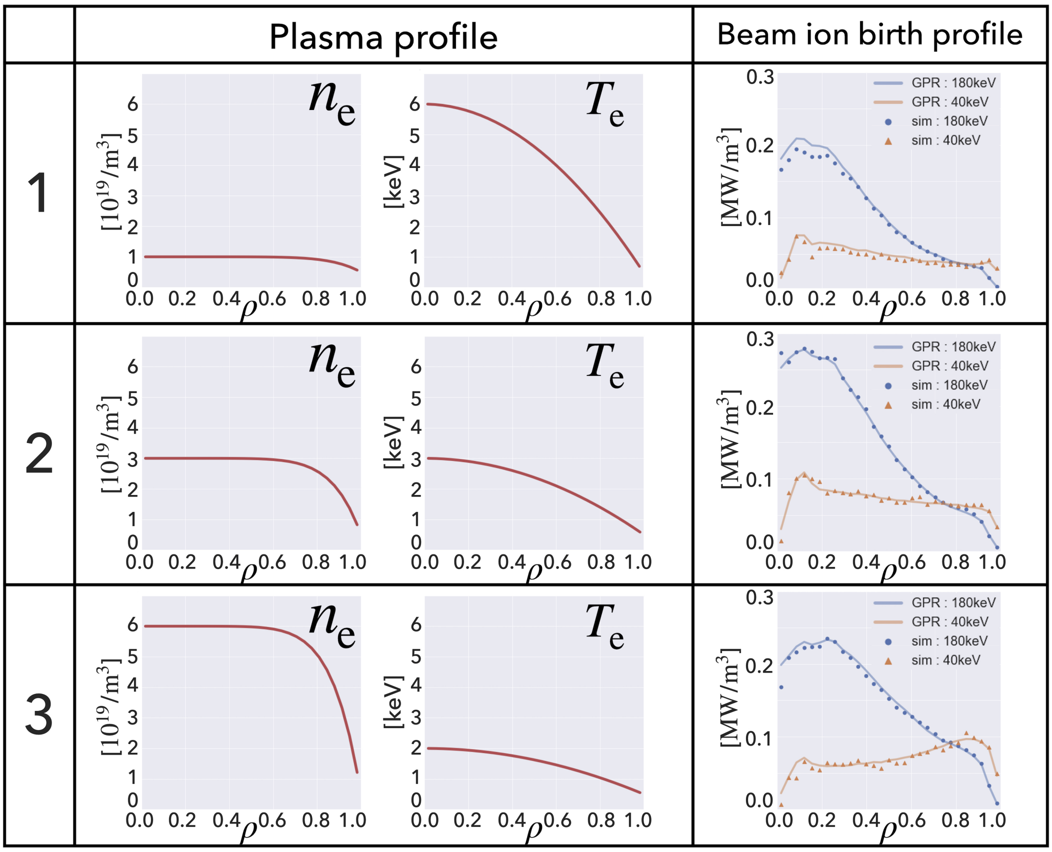


Figure 2.16: The prediction results of the beam ion birth profile by the GPR model for the 3 cases. The profiles labeled ”GPR” are the birth profiles given as Eqs. (2.61) and (2.62) for all $H_k = 1$ [MW] calculated by the GPR model. The profiles labeled ”sim” are those by HREYA and MCNBI.

the high-speed analysis and prediction of NBI-heated plasmas. We will discuss the integrated transport simulation using FIT3D-RC as the system model of ASTI in Section 3.2. Data assimilation can compensate the uncertainties inherent in the NBI heating model (including the uncertainty increased by the reduced computational cost) based on the observation data. Therefore, the data assimilation and FIT3D-RC can realize high-speed and accurate integrated transport simulation.

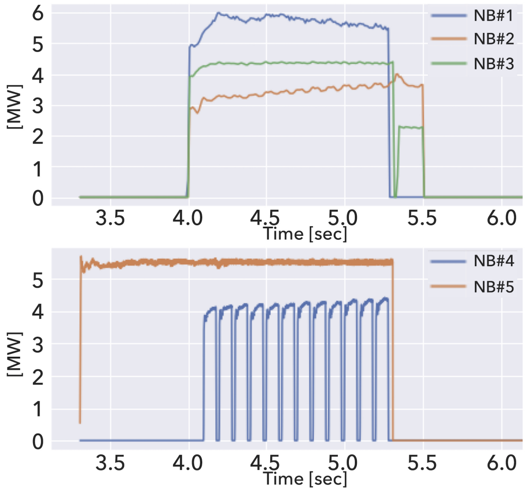


Figure 2.17: The time evolution of port-through power of NBI (shot number: 117100).

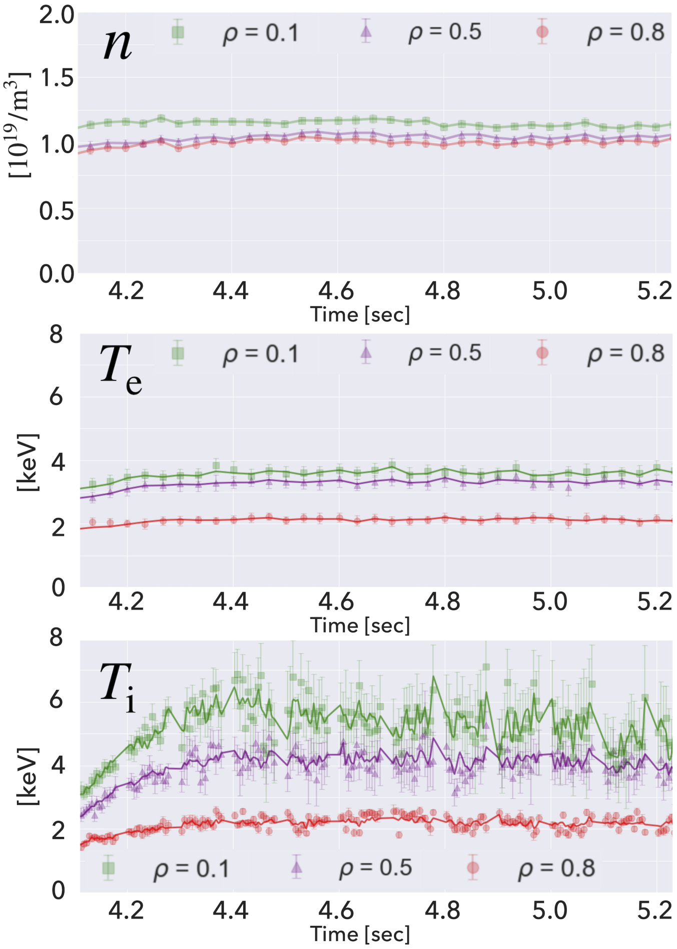


Figure 2.18: Experimental time series data of density and temperature (shot number: 117100). The solid lines are the interpolated data.

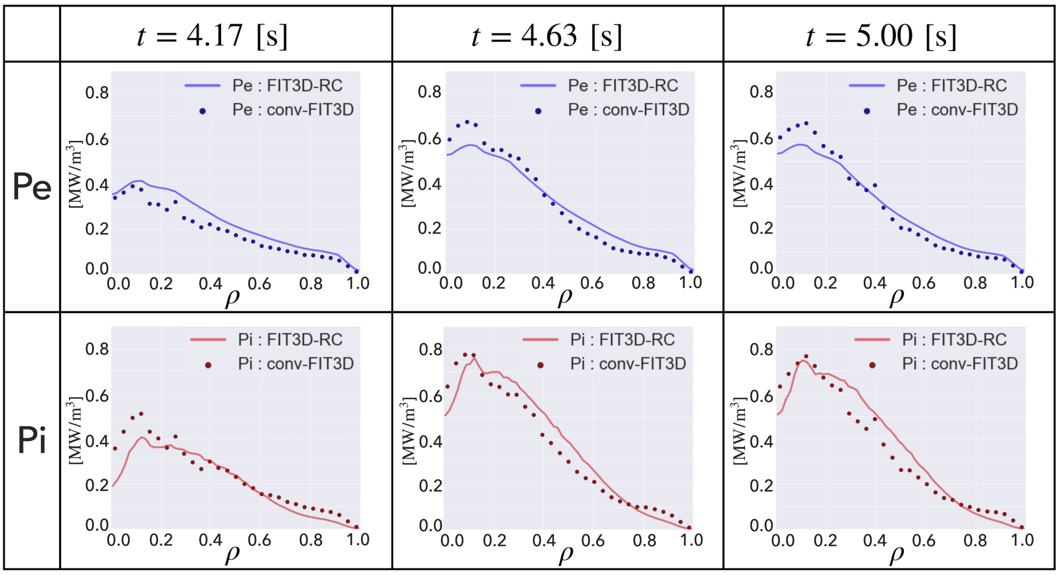


Figure 2.19: The simulation results of the power deposition of NBI heating at 3 timings. The profiles labeled "FIT3D-RC" are the deposition profiles calculated by FIT3D-RC. The profiles labeled "conv-FIT3D" are those by conv-FIT3D.

Chapter 3

Prediction and Model Parameter Estimation for NBI Heated Plasmas

In this chapter, we demonstrate applications of ASTI to experimental data of NBI-heated plasmas in LHD. Through the applications to actual discharges, we confirm the effectiveness and validity of the estimation by ASTI and investigate the prediction and analysis capabilities. In addition, we provide perspectives on transport model development by ASTI. In Section 3.1, the data assimilation using the EnKF and EnKS for heat transport in an NBI-heated plasma is discussed. Section 3.2 describes the data assimilation for particle and heat transport in LHD. In the same section, we also describe the reduction in computation time by using the rapid simulation code, FIT3D-RC, introduced in Section 2.4.

3.1 Data assimilation to heat transport simulation in LHD

We apply ASTI to a typical high ion temperature plasma in LHD (shot number 114053). The time series data of experimentally measured temperature and density profiles are assimilated into the TASK3D simulation. In this section, we discuss the prediction of electron and ion temperature and the estimation of the turbulent heat transport coefficients. This section also describes models to adjust noise parameters and optimization of hyper parameters using likelihood.

3.1.1 State vector

The state vector employed in this section is almost the same as the one employed in Section 2.3:

$$\mathbf{x}_t = (\mathbf{T}_{\mathrm{e},t}^{\mathrm{T}}, \mathbf{T}_{\mathrm{i},t}^{\mathrm{T}}, \mathbf{n}_{\mathrm{e},t}^{\mathrm{T}}, \mathbf{n}_{\mathrm{i},t}^{\mathrm{T}}, \mathbf{c}_{\mathrm{e},t}^{\mathrm{T}}, \mathbf{c}_{\mathrm{i},t}^{\mathrm{T}}, \mathbf{k}_{\mathrm{e},t}^{\mathrm{T}}, \mathbf{k}_{\mathrm{i},t}^{\mathrm{T}})^{\mathrm{T}}. \tag{3.1}$$

Every state variable is defined on 60 computational grid points (radial direction, 1D) and has a similar structure as follows:

$$\mathbf{T}_{\mathrm{e}} = (T_{\mathrm{e}}^{1}, T_{\mathrm{e}}^{2}, \cdots, T_{\mathrm{e}}^{60})^{\mathrm{T}}. \tag{3.2}$$

Table 3.1: The rates of the standard deviation of initial distribution and system noise.

State variable	Initial distribution	System noise
T_e	3%	8%
T_i	3%	6%
n_e	3%	4%
n_i	3%	4%
c_e	15%	10%
c_i	15%	8%
k_e	5%	4%
k_i	5%	3%

The superscript on each element in Eq. (3.2) indicates the grid point. In this section, TASK3D solves only the time evolution of the temperature, and the observations of electron and ion densities have the same values, that is, $n_\mathrm{e} = n_\mathrm{i}$. The density is assumed to be stationary in the prediction step but is updated in the filtering step. The state variables to adjust the turbulent heat diffusivities: c_e and c_i are defined as

$$\chi_\mathrm{e}^{\mathrm{TB}*} = c_\mathrm{e}\chi_\mathrm{e}^{\mathrm{TB}},\ \chi_\mathrm{i}^{\mathrm{TB}*} = c_\mathrm{i}\chi_\mathrm{i}^{\mathrm{TB}}. \tag{3.3}$$

These diffusivities with * are used instead of χ_s^{TB} in Eq. (2.44). The factor c_s is assumed to be grid-dependent, i.e., has spatial variation. The factor k_s is defined to adjust the NBI heat deposition evaluated by GNET-TD. The following P_s^* is used instead of P_s in Eq. (2.40):

$$P_\mathrm{e}^* = k_\mathrm{e}P_\mathrm{e},\ P_\mathrm{i}^* = k_\mathrm{i}P_\mathrm{i}. \tag{3.4}$$

The factor k_s also has spatial variation. The system model is optimized to reproduce temperature profiles through both the NBI heat deposition term and transport coefficients.

3.1.2 Initial distribution and noise

The initial ensemble mean is set to the observed data for T_s and n_s, and to the conventional values for c_s and k_s, i.e., $c_\mathrm{e} = c_\mathrm{i} = k_\mathrm{e} = k_\mathrm{i} = 1$. The initial standard deviation of the ensemble and the system noise are set to be proportional to the ensemble mean, and their rates are listed in Table 2.1. Here, we assume that the uncertainty associated with k_s is smaller than the uncertainty associated with c_s. If we assume that k_s has uncertainty as large as c_s, k_s and c_s cannot be uniquely optimized. This means that the NBI heat deposition model (GENT-TD) is more reliable than the turbulent transport models. This assumption distinguishes the roles of k_s and c_s, and allows stable estimation of these terms. In addition, the noise of the variables about electron is assumed to be larger than that about ion because the temporal variation and measurement error of ion temperature are larger than those of electron temperature in the NBI-heated plasma.

The standard deviation of observation noise is assumed to be proportional to the difference between the prediction and the observation data. The covariance matrix of

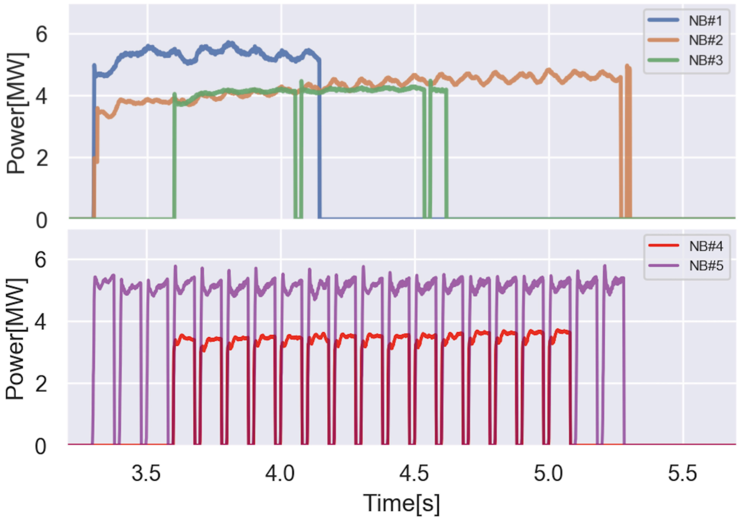


Figure 3.1: The time evolution of power through port of NBI (shot number: 114053).

observation noise, R_t, is estimated before every filtering step. This assumption keeps the variance of filtered ensemble in an adequate magnitude. For large prediction error, this assumption works to enlarge the variance and leads to more flexible optimization for state variables. On the other hand, for small prediction error, this assumption works to reduce the variance and prevents the variance from becoming excessively large. In addition, we set the upper limit of the standard deviation of observation noise to that of the predicted distribution, and the lower limit to 5% of the observation.

The rate which determines the standard deviation of observation noise can be optimized by likelihood. When using an ensemble approximation, the log-likelihood for the hyper parameters θ can be approximated as follows:

$$\begin{aligned} l(\theta) &= \sum_{i=1}^{M} \log p(\mathbf{y}_i|\theta, \mathbf{y}_{1:i-1}) \\ &\simeq \sum_{i=1}^{M} \log \left(\sum_{n=1}^{N} \alpha_i^{(n)} \right) - M \log N, \end{aligned} \tag{3.5}$$

where

$$\alpha_i^{(n)} = \frac{1}{\sqrt{(2\pi)^l |R_i|}} \exp \left\{ -\frac{1}{2} (\mathbf{y}_i - H\mathbf{x}_{i|i-1}^{(n)})^{\mathrm{T}} R_i^{-1} (\mathbf{y}_i - H\mathbf{x}_{i|i-1}^{(n)}) \right\}. \tag{3.6}$$

Here, $t_1, \cdots, t_M$ indicate the times assimilating observation data, and l is the dimension of $\mathbf{y}$. The rate of the observation model is determined to be 0.8, at which the maximum likelihood is obtained among the values: 0.5, 0.8, 1.0 and 1.2.

3.1.3 Assimilation results

We apply ASTI to a typical high ion temperature plasma in LHD (shot number: 114053). The time evolution of NBI heating (port through power) is shown in Fig. 3.1.

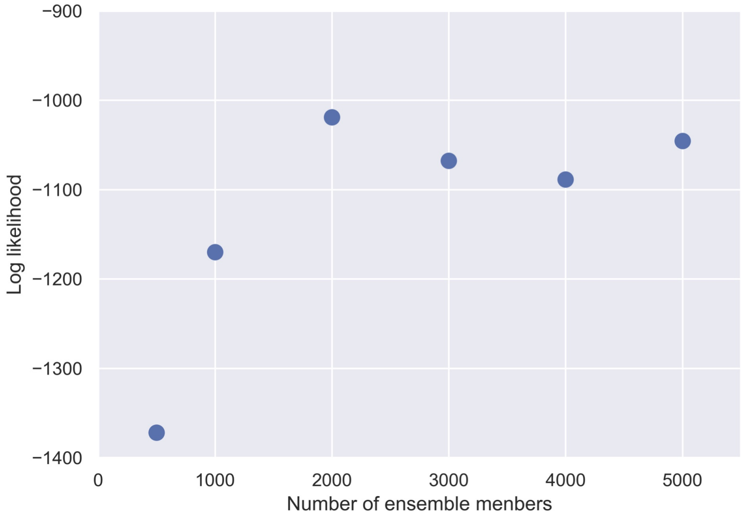


Figure 3.2: The log likelihood for the number of ensemble members

NB#1-#3 are injected tangentially to the magnetic field line with the beam energy $E_{\mathrm{NB}} \sim 180$ keV, and NB#4, #5 are injected perpendicularly to the magnetic field line with the beam energy $E_{\mathrm{NB}} \sim 40$ keV. NB#4 and #5 undergo power modulation to obtain the plasma background information for the ion temperature measurement. To make the observed data easy to be implemented, the time series of observed radial profiles are fitted by a polynomial with 6th order (only even degrees) in the radial direction and linearly interpolated in the time direction. Thus, we can define the observation vector as

$$\mathbf{y} = (\mathbf{T}_{\mathrm{e}}^{\mathrm{T}}, \mathbf{T}_{\mathrm{i}}^{\mathrm{T}}, \mathbf{n}_{\mathrm{e}}^{\mathrm{T}}, \mathbf{n}_{\mathrm{i}}^{\mathrm{T}})^{\mathrm{T}}, \tag{3.7}$$

and the observation matrix as

$$H_t = (I_{240\times240} \; O_{240\times240}). \tag{3.8}$$

where, $I_{240\times240}$ is a 240×240 identity matrix and $O_{240\times240}$ is a 240×240 zero matrix. This assimilation starts at 3.4 sec in the discharge, and this time is set to 0.0 sec in this simulation.

Prediction by the EnKF

We performed this assimilation with 2000 ensemble members for 2 cycles of assimilation (τ_{DA}): $\tau_{\mathrm{DA}} = 80$ msec (Case 1) and $\tau_{\mathrm{DA}} = 160$ msec (Case 2). The number of the ensemble members is determined by the log-likelihood. Figure 3.2 shows the log-likelihood for various numbers of ensemble members ($\tau_{\mathrm{DA}} = 80$ msec and M=10 in Eq. (3.5)). From Fig. 3.2, the number of ensemble members is found to be sufficient for 2000. Figure 3.3 shows the assimilation results of T_{e} and T_{i} and the case without the data assimilation (simple simulation). Since the T_{i} measurement starts at 0.3 sec, the T_{e} data are assimilated instead of T_{i} data (gray hatching in Fig. 3.3). In the

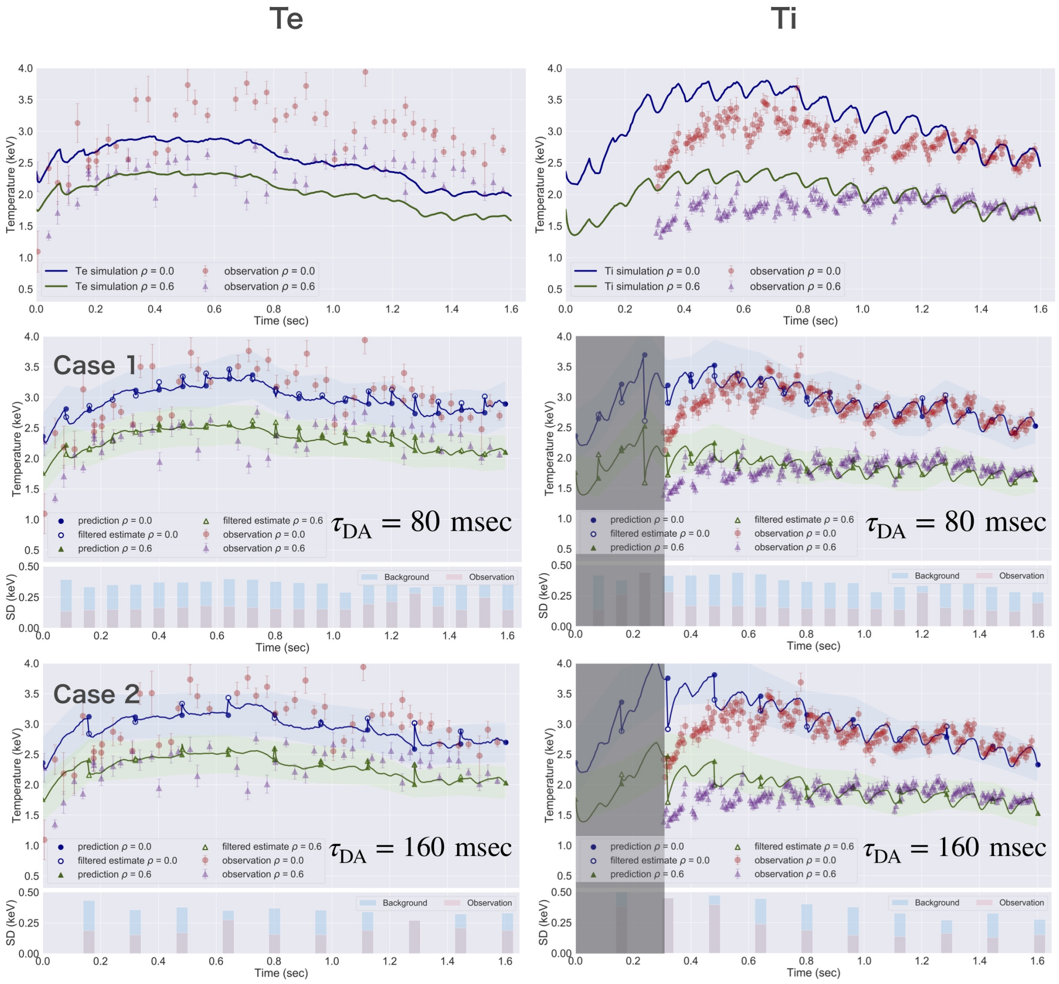


Figure 3.3: Prediction and filtered estimate of temporal change of T_{e} and T_{i} for $\rho = 0.0$ and $\rho = 0.6$. The two graphs in the first row are the simulation result without data assimilation, and the graphs in the second and third rows are the simulation results of case 1 and case 2. The bar graphs show the estimated standard deviations of background error and observation noise before filtering.

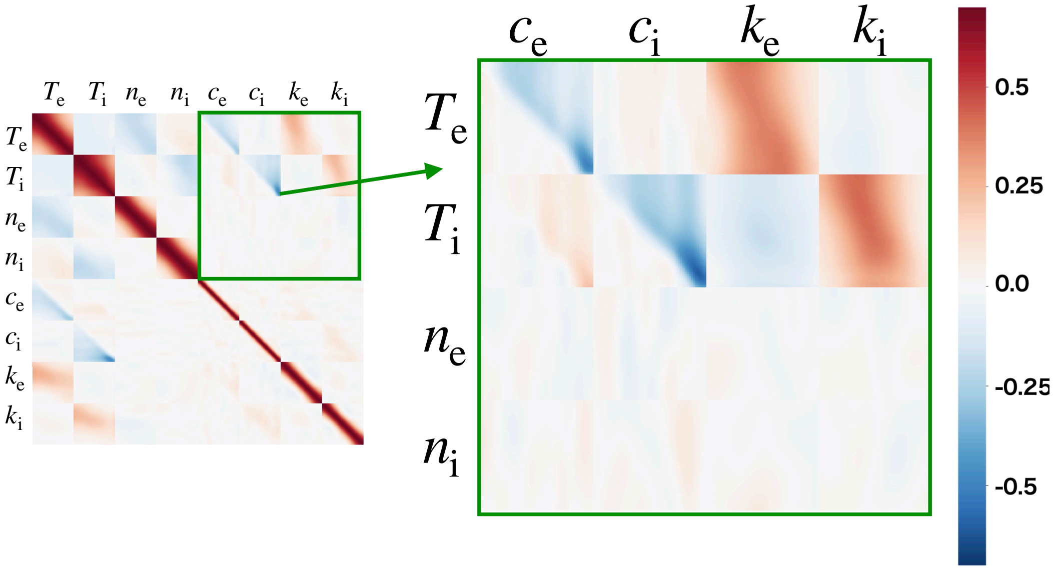


Figure 3.4: Ensemble correlation matrix of state variables at 160 msec in case 1.

simple simulation case, we use the time series of observed density data as the temporal evolution of electron and ion density in TASK3D.

As shown in Fig. 3.3, in both T_e and T_i, the prediction results by ASTI correspond reasonably with the observations better than those of simple simulation. The predictions agree well with the observations even in Case 2. This indicates that ASTI can follow the temporal change of numerical coefficients of the turbulent transport model in this plasma even for $\tau_\mathrm{DA} = 160$ msec. In other words, the nature of c_e and c_i behaviors do not largely change in 160 msec in this plasma.

Figure 3.4 shows an ensemble correlation matrix of state variables in Case 1. We can see a negative correlation between c_s and T_s, and a positive correlation between k_s and T_s. The correlation represents the relationship between the state variables in the simulation model (heat transport model). In the T_e-c_e and T_i-c_i parts of Fig. 3.4, we can see strong negative correlations in the upper triangular region. This means that the outer transport coefficients have a strong influence on the inner temperature. The correlation enables us to optimize unobserved variables as well as observed variables. The correlation information affects the optimization in Eq. (2.30) through the Kalman gain term, $\hat{K}_t$.

Figure 3.5 shows the filtered estimate of C_e and C_i, and the values of χ^TB as a result of the c adjustments are shown in Fig. 3.6. In addition, Fig. 3.7 shows the temperature profiles of prediction using data assimilation and only TASK3D at 960 msec in case 1. We can see that the temporal change of c_e is smooth, and the radial profile have been optimized to reproduce the observed profile of T_e, whose gradients are smaller around $\rho = 0.5$, and larger around $\rho = 0.8$ as in Fig. 3.7. The filtered estimate of c_i tends to become larger in the central region of the plasma. This is because that the T_i predictions without data assimilation are higher than the observation as in Fig. 3.3, and c_i is optimized to lower the ion temperature considering that the NBI heat deposition

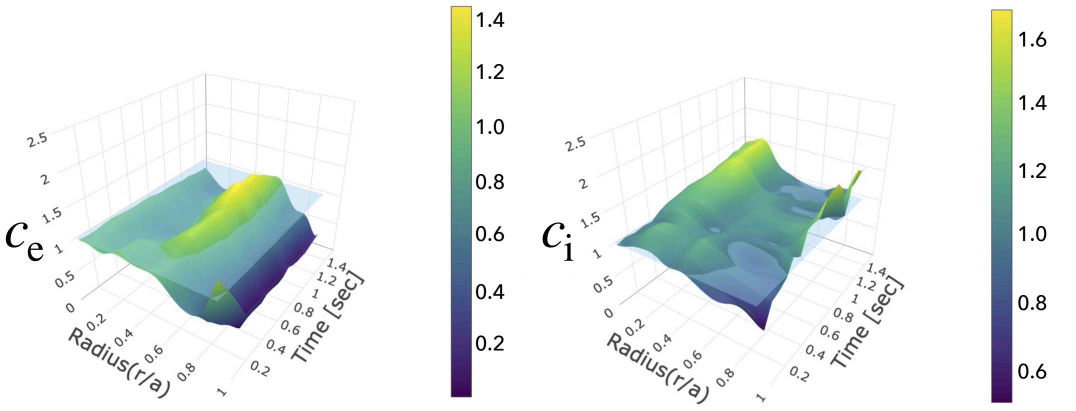


Figure 3.5: Filtered estimates of $c_{\rm e}$ and $c_{\rm i}$ in case 1, which are divided by the conventional values. Blue planes ($c = 1$) correspond to the conventional values in Eqs. (2.51) and (2.52).

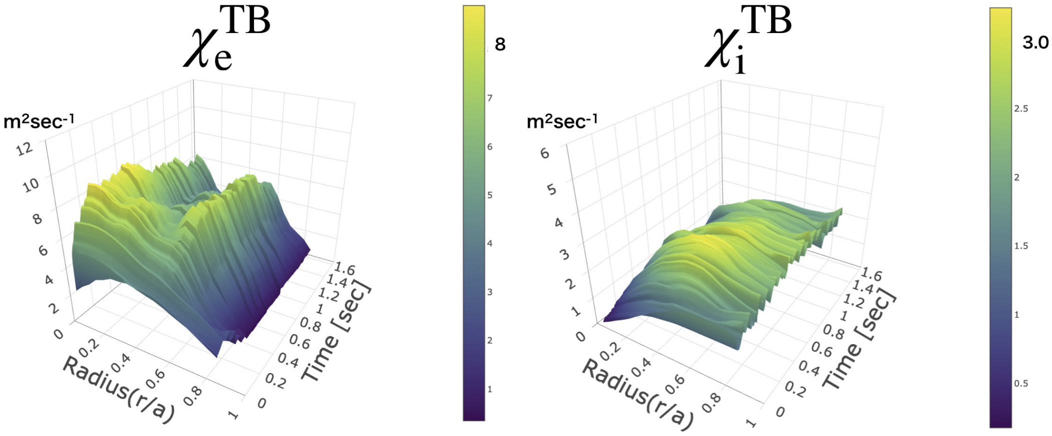


Figure 3.6: $\chi_{\rm e}^{\rm TB}$ and $\chi_{\rm i}^{\rm TB}$ calculated by using the $c_{\rm e}$ and $c_{\rm i}$ in Fig. 3.5.

is the largest in the central region.

The filtered estimate of c_s can include key information to predict the plasma behavior with high accuracy and to develop higher-performance transport models. However, the filter can not necessarily optimize the state vector uniquely. If a state vector can not be optimized uniquely like k_s and c_s in this assimilation, we must assimilate other observed data or must impose appropriate physical restrictions on state variables.

This data assimilation calculation takes about five hours for one simulation run (two seconds in actual plasma), using 2000 cores of Plasma Simulator at NIFS (Fujitsu PRIMEHPC FX100). Including the pre-calculation of the NBI heating terms, it takes more than ten hours. However, most of them are spent by the simulation code (TASK3D). The calculation time can be shortened drastically by making the computational grid rougher and the time step longer, and using simplex simulation models. In Section 3.2, we will discuss data assimilations when the number of spatial grid points

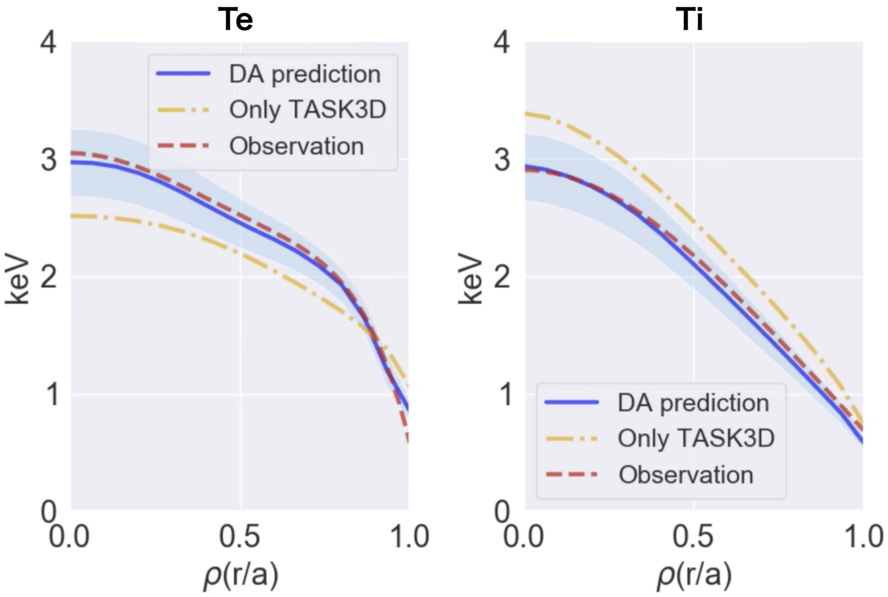


Figure 3.7: Radial profiles of T_e and T_i at 960 msec in case 1.

included in the state vector is reduced and fast computation models are employed.

Estimation of model parameters by the EnKS

We execute the smoothing procedure by the EnKS to the filtered distribution (ensemble) discussed in the previous section. Figure 3.8 shows the temporal change of c_s at $\rho = 0.25$. As shown in Fig. 3.8, the estimates of c_e and c_i by the EnKF are corrected and smoothed by the EnKS. In particular, the peak position of c_i moves downward from 1.1 s in the filtered estimate to 0.9 s in the smoothed estimate. This is the result of the EnKS correcting the filtered estimates to be reasonable both temporally and spatially using future data. Figure. 3.9 shows the TASK3D simulation results of T_e and T_i using the smoothed estimates of c_s and k_s (curves labeled "TASK3D*"). We can confirm that the TASK3D simulation using the smoothed estimates can reproduce experimental temperature data with high accuracy.

The smoothed estimate of c_s includes the information which should be taken into account in the turbulent transport models to predict the temperature of plasma more accurately. If we can reproduce the estimates of c_s for various plasmas using a parametric or nonparametric model, we obtain a more valid turbulent transport model with relevant physical interpretation. ASTI is expected to be a powerful tool to analyze fusion plasmas, which should consider many variables simultaneously and consistently.

In the next section, we will discuss the application of ASTI to the particle and heat transport in NBI-heated plasmas. There, we employ a system model where the prediction step can be calculated in a reasonable computation time. We investigate the performance and validity of ASTI through the data assimilations to some time-series experimental data.

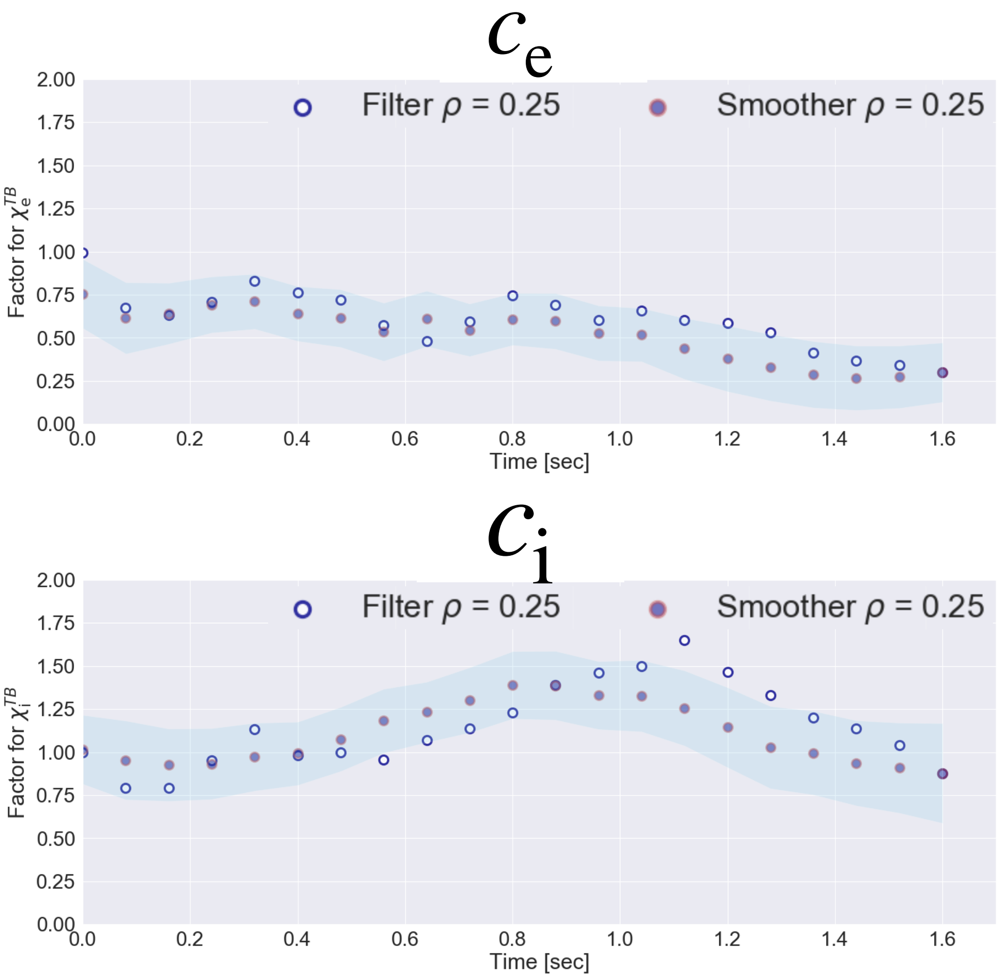
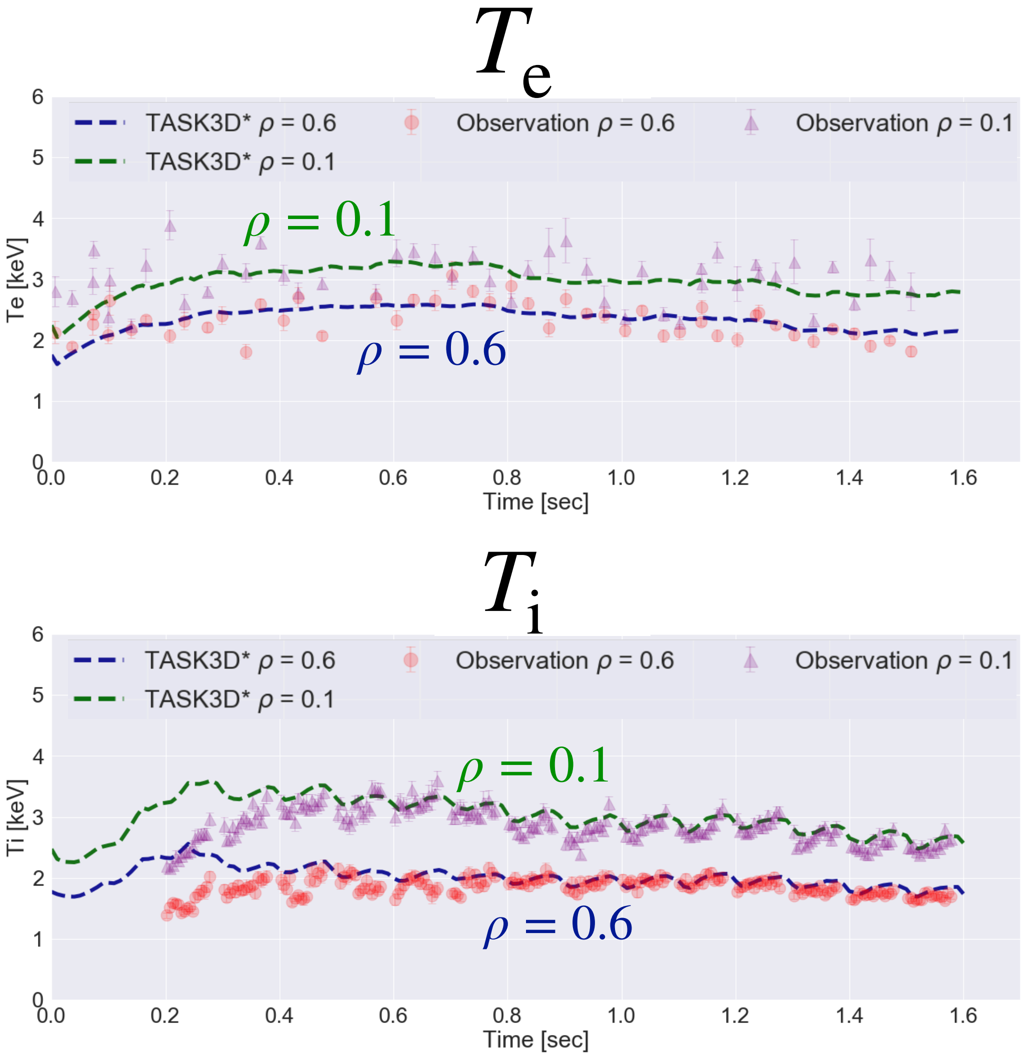


Figure 3.8: The temporal change of smoothed estimates of c_e and c_i at $\rho = 0.25$. The highlighted areas around the smoothed estimates represent the standard deviations of the smoothed ensemble.



T_e

T_i

Figure 3.9: TASK3D simulation results of T_e and T_i using the smoothed estimates of c_s and k_s for $\rho = 0.1$ and $\rho = 0.6$.

3.2 Data assimilation to particle and heat transport simulation in LHD

In this section, we apply ASTI to 12 time-series data sets of NBI-heated hydrogen plasmas in LHD (shot number: 111366, 116126, 116195, 117100, 119801, 119802, 119803, 119804, 119979, 119789, 120035, 120051), and investigate the capability of the prediction and the estimation [66]. These discharges are the typical stable NBI-heated plasmas without break down of the NBI heating and sufficient time series data. The range of NBI power in these discharges is 13-24 MW (4-16 MW tangential injection and 8-11 MW perpendicular injection). The range of electron temperature at the plasma center is 1-4 keV, that of ion temperature is 2-7 keV, and that of density is 1.0-2.0×10^{19} m^{-3}. In common with all the discharges, the major radius of the magnetic axis at the vacuum is 3.6 m and the magnetic field strength at the plasma center is 2.85 T.

3.2.1 State-space model

We accelerate the prediction calculation utilizing the reduced NBI heating model, FIT3D-RC, described in Section 2.4. The uncertainties increased by employing the reduced model are suppressed by data assimilation optimizing the parameters in the model. Conventional simulation codes of NBI heating take a long calculation time from dozens of minutes [47] to dozens of hours [54] to evaluate the NBI heating power for a second in an actual plasma, depending on the accuracy the simulation needs. When these codes are coupled with an integrated code to solve for the transport and the heating simultaneously, even more calculation time is required. The combination of simplified models and data assimilation enables fast and accurate prediction and analysis. Such speeding up of simulation is also one of the advantages of employing data assimilation. Furthermore, unlike the assimilation to optimize the factor for pre-computed power deposition (k_s) in Section 3.1, the transport simulation can be optimized from the inside of the NBI heating model.

state vector

In this study, we consider the uncertainties of plasma density, temperature, parameters in the transport model, parameters in the reduced NBI heating model (FIT3D-RC), and parameters in the particle source model. The state variables are listed in Table 3.2. The variables n, T_e, and T_i are the density, electron temperature and ion temperature, respectively. We assume that the densities of electron and ion are the same, that is, $n_\mathrm{e} = n_\mathrm{i} = n$, $D_\mathrm{e} = D_\mathrm{i} = D$, and $V_\mathrm{e} = V_\mathrm{i} = V$. The variables d, v, c_e, and c_i are the numerical factors to optimize the transport coefficients using data assimilation, and the variables $\xi_{180\mathrm{keV}}$, $\xi_{40\mathrm{keV}}$, ξ_c, and ξ_sd are the numerical factors to optimize the parameters in the FIT3D-RC model. The following parameters with * are used in the prediction calculation by TASK3D instead of the parameters without * :

$$D^{\mathrm{TB}*} = dD^{\mathrm{TB}}, \quad V^* = V + v, \quad \chi_\mathrm{e}^{\mathrm{TB}*} = c_\mathrm{e}\chi_\mathrm{e}^{\mathrm{TB}}, \quad \chi_\mathrm{i}^{\mathrm{TB}*} = c_\mathrm{i}\chi_\mathrm{i}^{\mathrm{TB}}, \tag{3.9}$$

$$f^*_{180\mathrm{keV}} = \xi_{180\mathrm{keV}} f_{180\mathrm{keV}}, \quad f^*_{40\mathrm{keV}} = \xi_{40\mathrm{keV}} f_{40\mathrm{keV}}, \quad v^*_\mathrm{c} = \xi_\mathrm{c} v_\mathrm{c}, \quad \tau^*_\mathrm{sd} = \xi_\mathrm{sd}\tau_\mathrm{sd}. \tag{3.10}$$

Table 3.2: State variables, their dimension in the state vector (M_i), and their rates of the standard deviation of initial distribution (σ_I) and system noise (σ_Q).

No.	State variable (target of optimization)		M_i	σ_I	σ_Q
1	n	Density	7	10%	10%
2	T_e	Electron temperature	7	10%	10%
3	T_i	Ion temperature	7	10%	10%
4	d	Particle turbulent diffusivity	7	20%	20%
5	v	Particle convection velocity	6	1.0 [m/s]	1.0 [m/s]
6	c_e	Electron thermal turbulent diffusivity	7	20%	20%
7	c_i	Ion thermal turbulent diffusivity	7	20%	20%
8	$\xi_{180\mathrm{keV}}$	Beam ion birth profile (180 keV)	7	10%	5%
9	$\xi_{40\mathrm{keV}}$	Beam ion birth profile (40 keV)	7	10%	5%
10	ξ_c	Critical velocity	7	10%	5%
11	ξ_sd	Slow down time of beam ion	7	10%	5%
12	n_n	Neutral density at plasma edge	1	10%	10%
13	T_n	Neutral temperature at plasma edge	1	10%	10%

These variables are allowed to have spatial variation. The dimension (M_i) in Table 3.2 refers to the number of points in the radial profile. The values of parameters without * are determined by the employed models in normal TASK3D simulations (e.g., Eqs. (2.50)-(2.52)). The variables $f_{180\mathrm{keV}}$ and $f_{40\mathrm{keV}}$ are the heat deposition of 180 keV beam and 40 keV beam evaluated by the ion-birth part of the FIT3D-RC model (Eqs. (2.61) and (2.62)). The variables v_c and τ_sd are the critical velocity and slowing-down time, respectively, in the slowing-down part of the FIT3D-RC model (Section 2.4.1). The variables n_n and T_n are the parameters in the particle source model (AURORA module). Through these state variables, we quantify the uncertainties contained in the transport coefficients and source terms as probability distributions like Fig. 3.10. The time evolution of the variables 1-3 ($n, T_\mathrm{e}, T_\mathrm{i}$) is affected by the variables 4-13 through the employed model, like the parameters with * in Eqs. (3.9) and (3.10), while the variables 4-13 do not change (constant) in the prediction calculation by TASK3D. The values of variables 4-13 for each ensemble member are changed only by filtering or adding the system noise.

The state vector has a structure as follows:

$$\mathbf{x} = (\mathbf{x}_1^\mathrm{T}, \mathbf{x}_2^\mathrm{T}, \cdots, \mathbf{x}_S^\mathrm{T})^\mathrm{T}, \tag{3.11}$$

$$\mathbf{x}_i = (x_i^1, x_i^2, \cdots, x_i^{M_i})^\mathrm{T}, \tag{3.12}$$

where S denotes the number of state variables, i.e., $S = 13$, and M_i denotes the dimension which the state variable x_i has in the state vector. In this section, the radial profile in the state vector is defined on 7 grid points (ρ'_j=0, 1/6, 2/6,$\cdots$,1) to reduce calculation costs of the filter and the smoother, while that is defined on 60 grid points in TASK3D. In addition, it also allows us to perform the data assimilation processes with a smaller number of ensemble members. The radial profiles in the state vector are interpolated using the basis functions $\beta_{\rho'_j}(\rho)$ and converted to the profiles on TASK3D

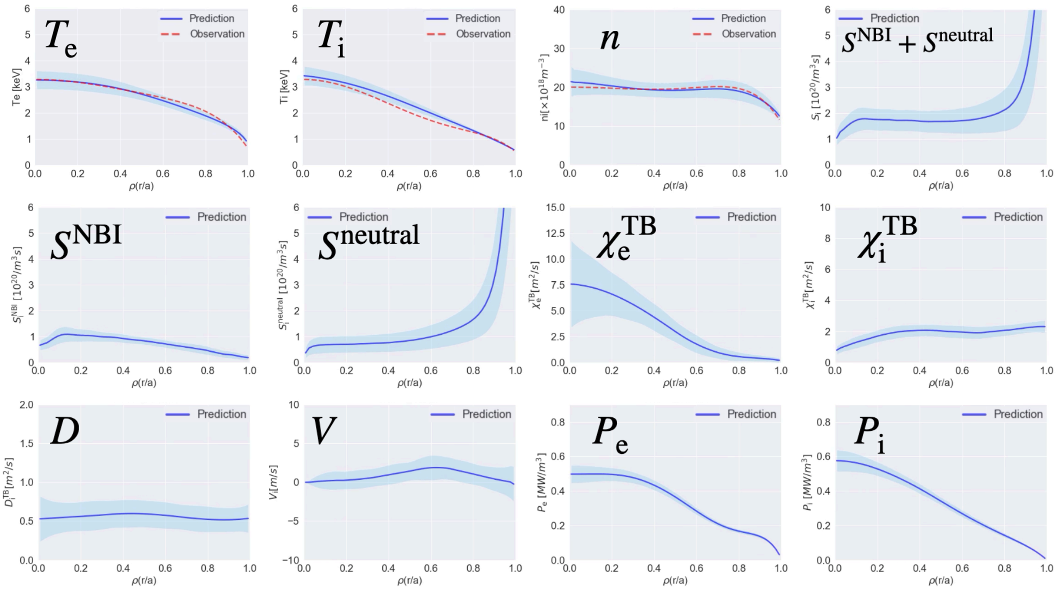


Figure 3.10: An example of the probability distributions of density, temperature, transport coefficients, and source terms. The highlighted areas around prediction profiles represent the standard deviations.

grid as

$$\phi_{\rm TASK3D}(\rho) = \sum_{j=1}^{7} \beta_{\rho'_j}(\rho)\phi_{\rm SV}(\rho'_j), \tag{3.13}$$

where $\phi_{\rm TASK3D}$ and $\phi_{\rm SV}$ are the radial profiles defined on the grid in TASK3D and the state vector, respectively. The basis functions $\beta_{\rho'_j}(\rho)$ are chosen as non-uniform quadratic B-spline basis functions [67] as shown in Fig. 3.11. The B-spline basis functions in Fig. 3.11(a) are chosen such that the boundary values ($\rho = 0$ and 1) are determined only by the values at $\rho' = 0$ and 1. These functions are used to interpolate the profiles of model parameters: d, v, $c_{\rm e}$, $c_{\rm i}$, $\xi_{\rm 180keV}$, $\xi_{\rm 40keV}$, $\xi_{\rm c}$, and $\xi_{\rm sd}$. The basis functions in Fig. 3.11(b) are furthermore chosen such that all basis functions have zero derivative at $\rho = 0$ to satisfy the Neumann boundary condition in TASK3D, $\partial\phi_{\rm TASK3D}/\partial\rho|_{\rho=0} = 0$. These functions are used to interpolate the profiles of variables n, $T_{\rm e}$, and $T_{\rm i}$. It has been confirmed that the typical radial profiles on the 60 grid points in TASK3D can be sufficiently reproduced from the profile on the 7 grid points. The value of v at ρ=0 is fixed at 0 [m/s] because there is no particle flux at ρ=0. Therefore, the dimension of v in the state vector is 6.

To calculate the time evolution using TASK3D, the radial profiles need to be smooth after adding the noise to them. Thus, The spatial correlation of the noise (covariance component of the covariance matrix) is required to generate the smooth noise samples from the Gaussian distribution. In this case, we can employ a diagonal matrix for the covariance matrix of the noise because the B-spline interpolation guarantees the smoothness of the radial profiles and allows the noise to be spatially independent. This dimensionality reduction also saves the computational cost of Cholesky decompo-

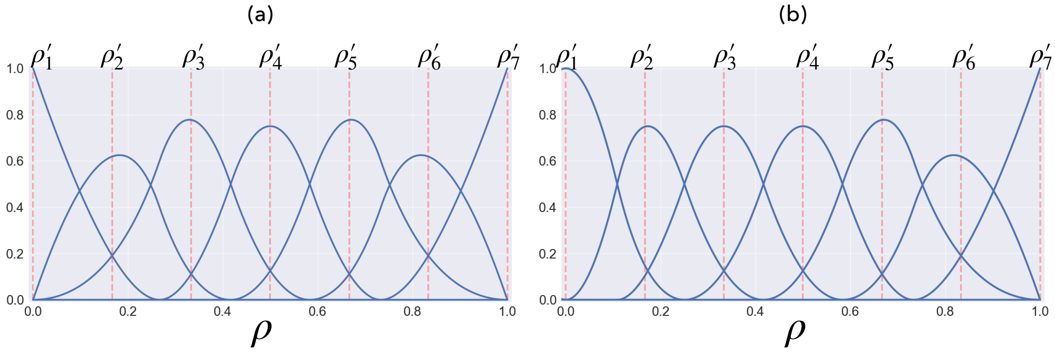


Figure 3.11: The non-uniform quadratic B-spline basis functions on the ρ grid.

sition of the covariance matrix required to generate noise samples from the Gaussian distribution.

Observation model

The experimental time-series data of radial profiles of density and electron and ion temperatures are assimilated into the integrated transport simulation of NBI-heated plasma in LHD. To make it easy to assimilate the observed data, the time-series data are fitted by a sixth-order polynomial (only even degrees) in the radial direction and linearly interpolated in the time direction. Since n, T_{e}, and T_{i} are represented by three 7-dimensional vectors $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ in Eq. (3.11), respectively, the observation matrix H_t is given by

$$H_t = (I_{21\times 21} \; O_{21\times 58}) \tag{3.14}$$

to extract observable variables from $\mathbf{x}$. Here, $I_{21\times 21}$ is a 21 × 21 identity matrix and $O_{21\times 58}$ is a 21 × 58 zero matrix.

3.2.2 Initial distribution and noise

The initial ensemble mean is set to the observation data at the start of simulation for density and temperatures, and to the conventional values for other state variables, that is, $d = c_{\mathrm{e}} = c_{\mathrm{i}} = \xi_{180\mathrm{keV}} = \xi_{40\mathrm{keV}} = \xi_{\mathrm{c}} = \xi_{\mathrm{sd}} = 1$, $v = 0$ [m/s], $n_{\mathrm{n}} = 10^{16}$ [m^{-3}], and $T_{\mathrm{n}} = 10$ [eV]. The covariance matrices $V_{0|0}$, Q_t, and R_t are assumed to be diagonal matrices as mentioned above. The standard deviations of initial ensemble and system noise are set to be proportional to the ensemble mean, and their rates are listed in Table 3.2. Since the initial mean of v is assumed to be zero, the standard deviations of the noises for v are set to fixed values of 1 [m/s]. The standard deviations of the noises for the transport coefficients are assumed to be larger than those for other model parameters because the transport coefficients have greater uncertainties and more unreliable than the other model parameters. This assumption distinguishes the roles of the model parameters and realizes stable estimation. The standard deviation of observation noise is assumed to be proportional to the difference between the prediction and the observation data as in Section 3.1. We employ the rate of 0.8.

Table 3.3: RMSE and RMSPE between the observed values and the predicted values of electron temperature, ion temperature, and density by TASK3D and ASTI.

		RMSE		RMSPE	
	ρ	TASK3D	ASTI	TASK3D	ASTI
Electron temperature	0.1	0.310	0.0869	9.99%	2.86%
[keV]	0.6	0.321	0.0770	12.25%	3.06%
Ion temperature	0.1	0.860	0.310	20.85%	7.97%
[keV]	0.6	0.605	0.196	31.29%	9.31%
Density	0.1	0.322	0.0534	26.49%	4.56%
[10^{19}m^{-3}]	0.6	0.410	0.0755	34.53%	6.26%

3.2.3 Prediction by the EnKF

We apply ASTI to 12 time-series data sets of NBI-heated hydrogen plasmas in LHD. All assimilations are performed with 500 ensemble members. Figures 3.12 and 3.13 show the prediction and the filtered estimates of electron and ion temperatures by ASTI (EnKF) for 5 shots out of the 12 shots (the columns labeled “ASTI”). These assimilations are performed for the assimilation cycle τ_{DA}=40 ms. For comparison, the simulation results by TASK3D (without data assimilation) are shown in the columns labeled “TASK3D”. Figure 3.14 is similar graphs about the density. We can see that ASTI’s predictions agree well for observation data at all of the electron temperature, ion temperature, and density, while TASK3D’s predictions have large discrepancies with respect to the observed values, especially in the ion temperature and density. The state variables have been greatly improved in the first few times of filter, especially in Fig. 3.13 (shot: 117100, 119801, and 120035). Although the modification of the electron temperature at each filtering step (the differences between the prediction and the filtered estimate in Fig. 3.12) seems small, the model parameters for electron heat transport are optimized (e.g., c_{e} in Fig. 3.17, which will be mentioned in Section 3.2.4). As a result, the ASTI’s prediction of the electron temperature can reproduce the experimental time series data with high accuracy, even for the discharges with relatively large errors in the TASK3D simulation (e.g., shot: 117100, 119801, and 119802 in Fig. 3.12).

Figure 3.15 shows the comparisons between the observed value and the prediction by TASK3D (left column) and by ASTI (right column). The points in Fig. 3.15 represent the values at ρ=0.1 and 0.6 for all 262 timings in the 12 shots. Table 3.3 shows the root mean square error (RMSE) and root mean square percentage error (RMSPE) between the observed value and the prediction of density, electron temperature and ion temperature at ρ=0.1 and 0.6. From Fig. 3.15 and Table 3.3, it is confirmed that the prediction errors of the density, electron temperature, and ion temperature are significantly reduced by the EnKF. The RMSE at ρ=0.1 are reduced by 72% for electron temperature, by 64% for the ion temperature, and by 83% for the density. The RMSPEs of the ion temperature predicted by ASTI are larger than those of the electron temperature and density. This is because the temporal variation and measurement error of ion temperature are larger than those of electron temperature and density in the 12 data sets.

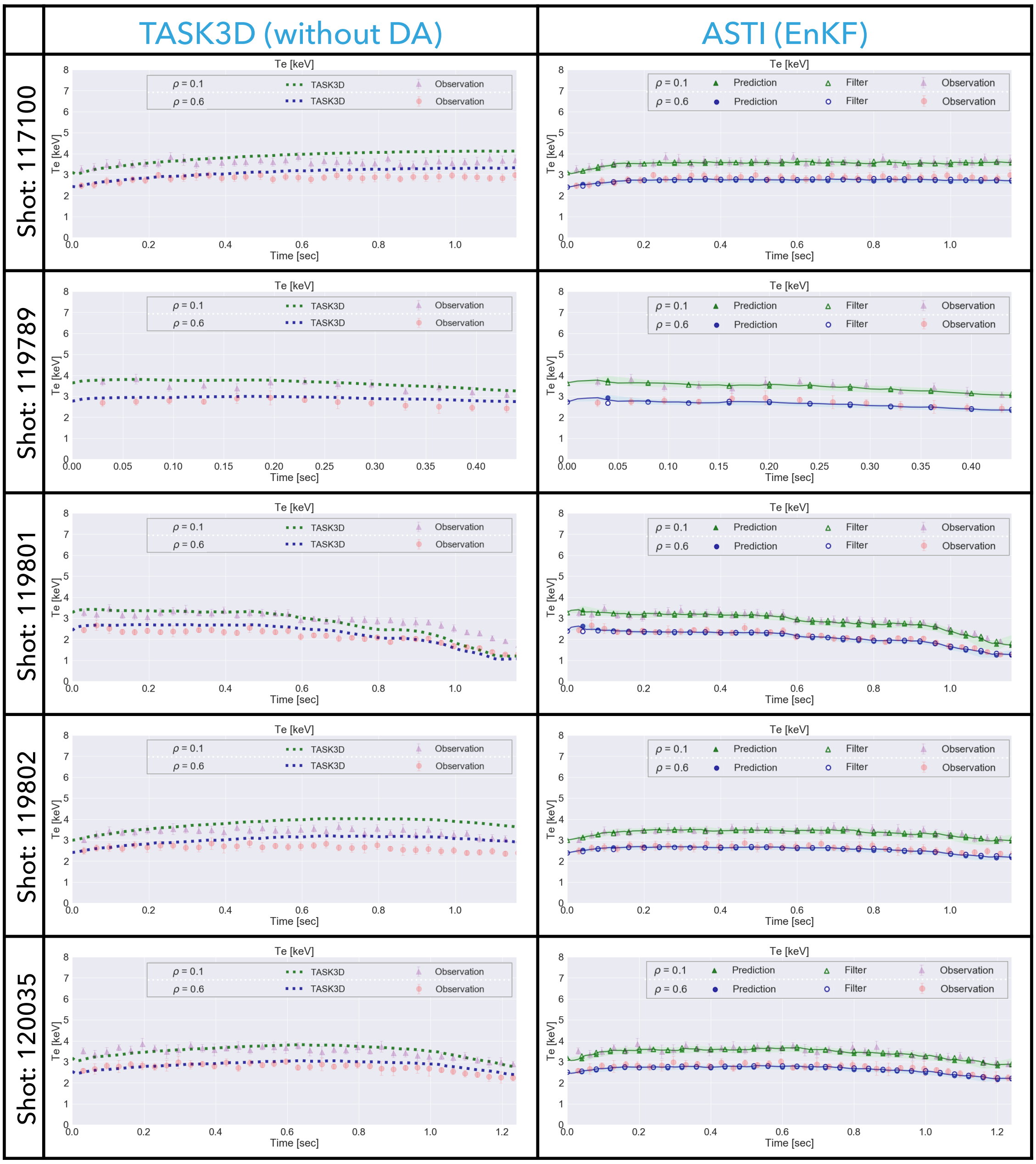


Figure 3.12: Simulation results of time evolution of electron temperature profile by TASK3D (column labeled "TASK3D") at ρ=0.1 and 0.6, and the prediction and filtered estimate by ASTI (column labeled "ASTI") for 5 shots. The highlighted areas around predictions represent the standard deviations of predicted ensemble.

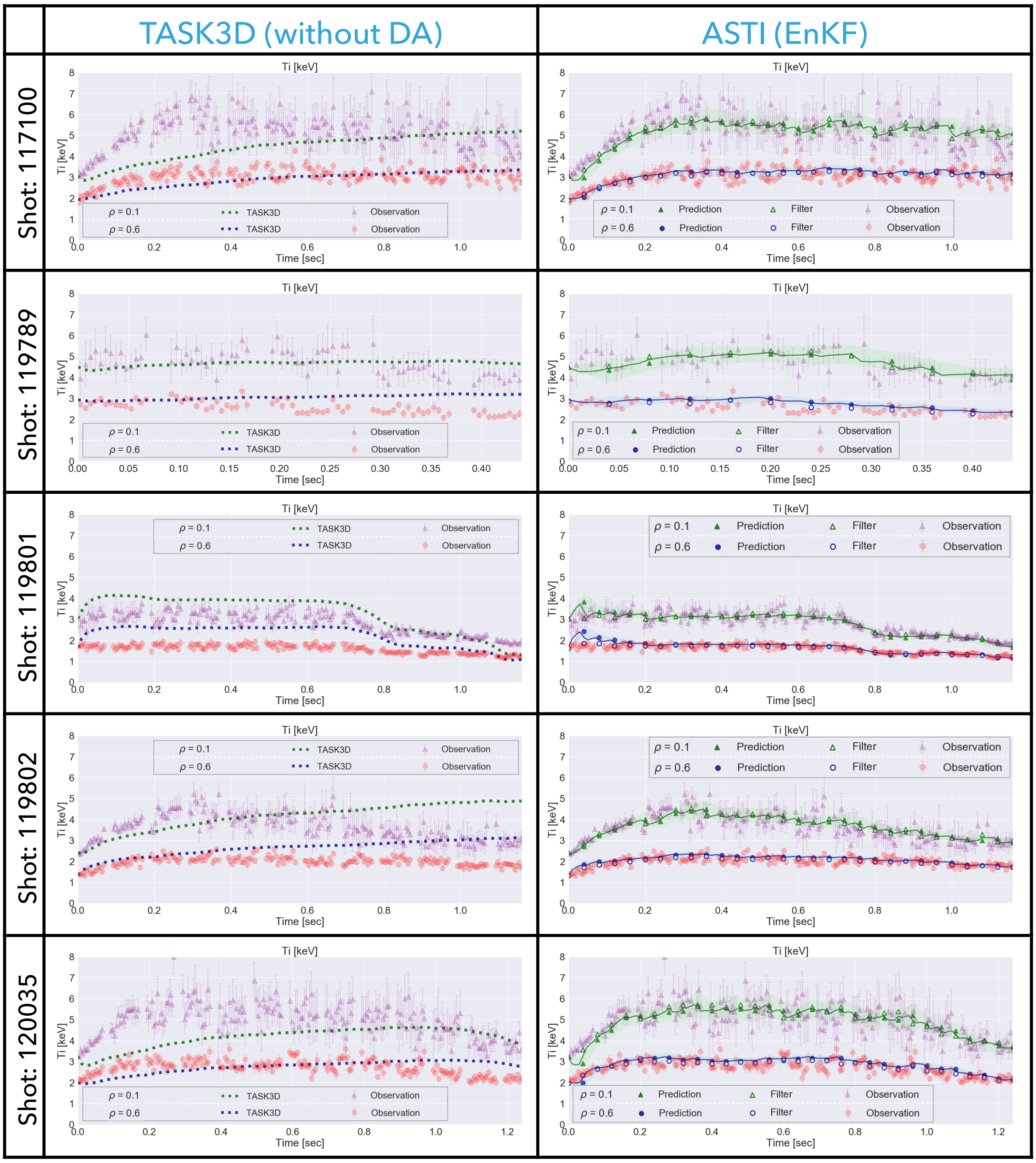


Figure 3.13: As Fig. 3.12, but ion temperature.

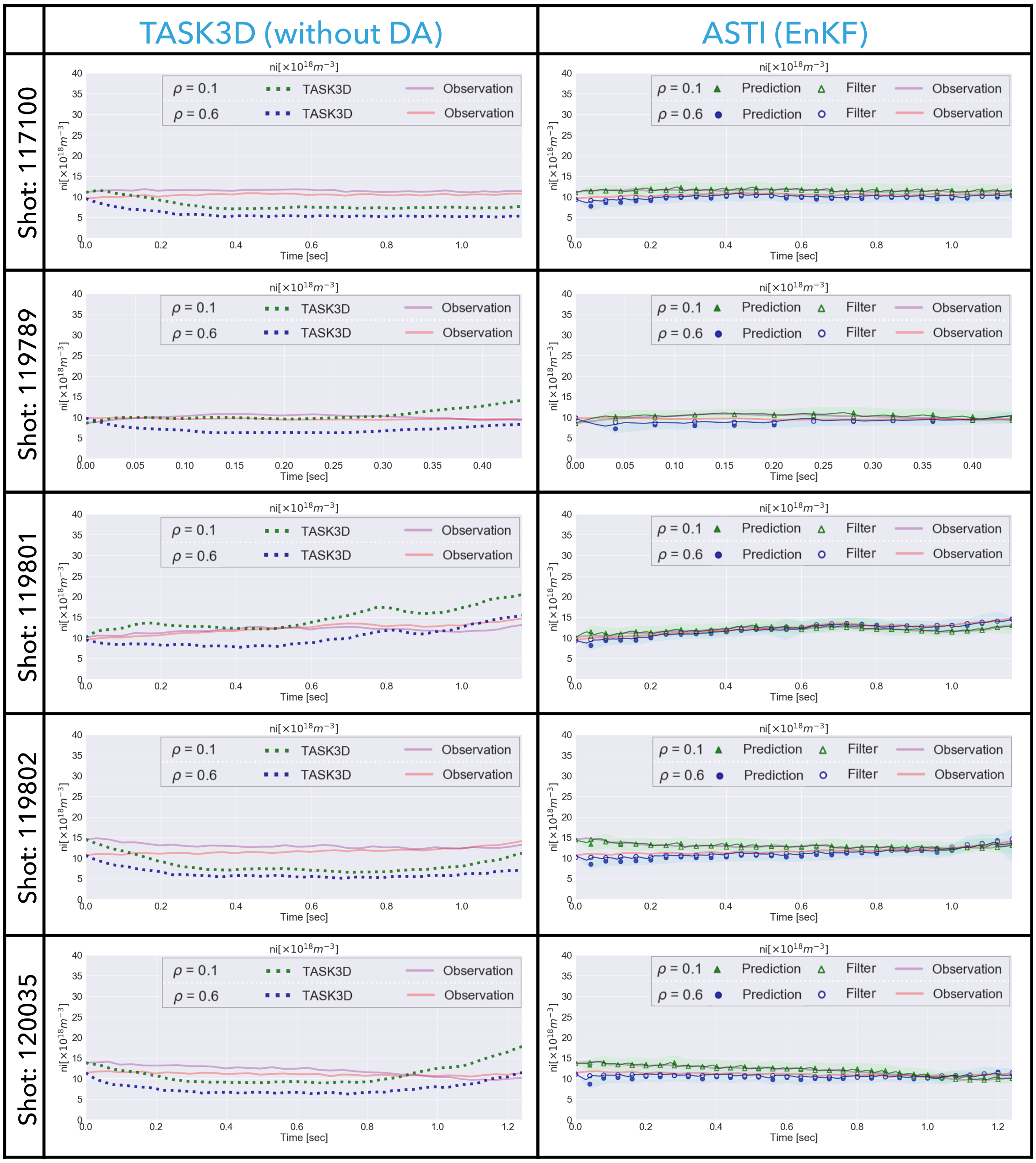


Figure 3.14: As Fig. 3.12, but density.

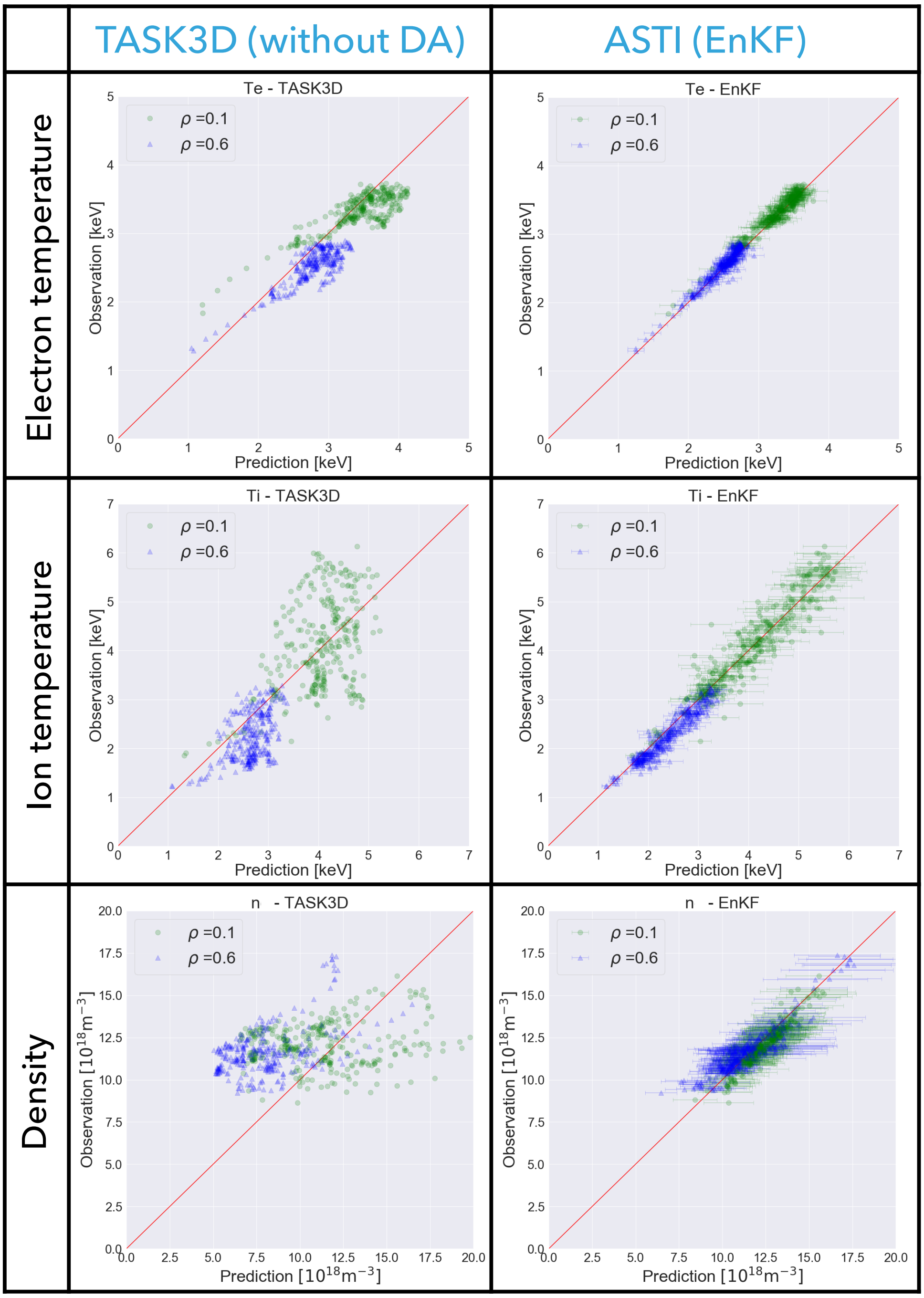


Figure 3.15: Comparisons between the observed and predicted values by TASK3D (column labeled "TASK3D") and ASTI (column labeled "ASTI") at ρ=0.1 and 0.6 for all the 12 shots. The error bars at the scatter plots in the column labeled "ASTI" are the standard deviations of predicted ensemble.

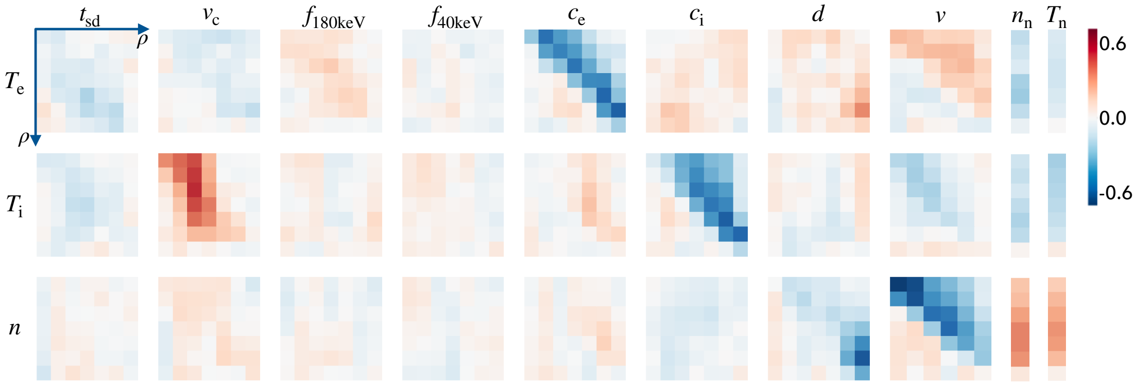


Figure 3.16: Part of ensemble correlation matrix in the data assimilation for shot: 117100 (4.19 s).

It is considered that the introduction of the NBI heating model, FIT3D-RC, contributes to the good agreement between ASTI's predictions and observations. The introduction of FIT3D-RC allows for finer optimization of the simulation model by the data assimilation compared to the data assimilation discussed in Section 3.1, where the NBI heating power profiles are pre-computed by a high cost model (GNET-TD). The time evolution curves predicted by ASTI shown in Figs. 3.12, 3.13, and 3.14 appear to be smoother than the curves shown in Fig. 3.3. As TASK3D simulation can be optimized in more detail than the previous study, the prediction capability of ASTI increases, and the differences between prediction and filtered estimate become smaller.

Figure 3.16 shows a part of the ensemble correlation matrix in the data assimilation for shot number: 117100. It represents the correlations between observed variables (three rows) and unobserved variables (ten columns). The relationship between the state variables in the simulation model appears as a correlation. The correlation enables the EnKF and EnKS to optimize the entire state vector using limited observation information. In common with all the data sets, the correlations between the observed variables and the transport model parameters ($c_\mathrm{e}, c_\mathrm{i}, d$ and v) are particularly strong, and those with the other model parameters are ±0.2-0.5. In the T_e-c_e, T_i-c_i, and n-v parts of Fig. 3.16, we can see strong correlations in the upper triangular region. This means that the outer transport coefficients have a strong influence on the inner temperature or density. The n-d correlation is strong in the lower right region because the density gradient is larger in the outer radial points.

3.2.4 Estimation of model parameters by the EnKS

The upper panels in Fig. 3.17 show the smoothed estimates of the parameters in transport models at all assimilation timings for the 12 shots and the lower panels show the kernel density estimation of the upper panel. We can see that the smoothed estimates have a common structure in each radial profile of v, c_e, and c_i. The smoothed estimates of c_e has a structure that enhances the turbulent diffusivity around the center ($\rho = 0$) and the edge ($\rho = 0.9$) and reduces it around $\rho = 0.6$. The estimates of c_i has a structure similar to c_e, but reduces the diffusivity (smaller than one) around $\rho = 0$. The

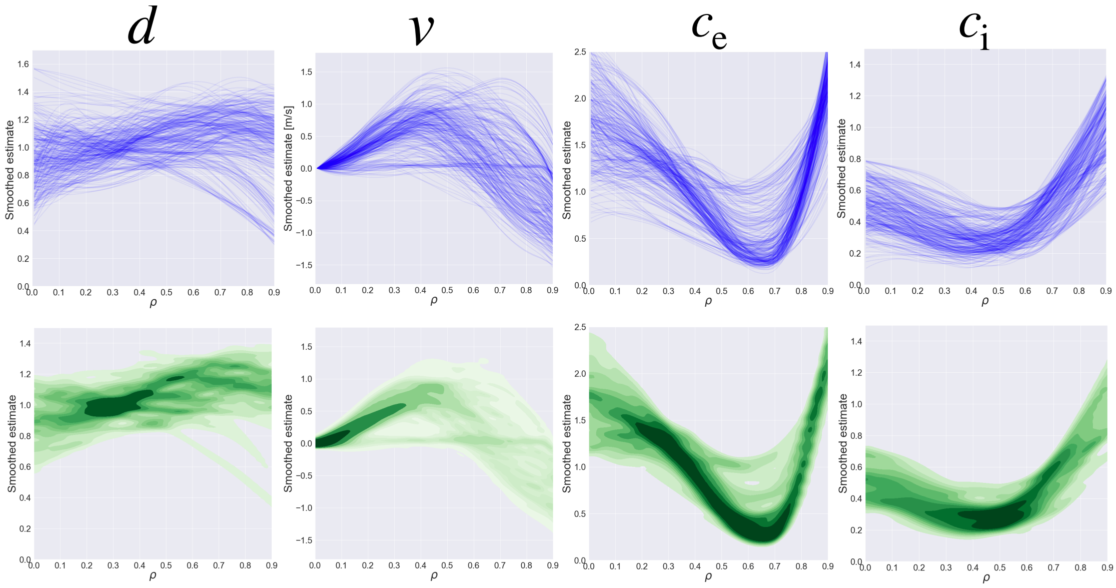


Figure 3.17: Smoothed estimates of the parameters in transport coefficient models at all assimilation timings for the 12 shots. The upper panels plot all the estimates as curves, and the lower panels are the color maps of the kernel density estimation corresponding to the upper figure.

radial profile of v has a structure that peaks near $\rho = 0.5$ and becomes negative near the edge. The variation of the estimated v is large near the edge as shown in the lower panel. On the other hand, no clear characteristic structure is found in the radial profile of d. This large variation of d profiles and v values near the plasma edge would be due to no model assumption for the particle turbulent diffusivity, D^{TB}, and the additional convection velocity, v, on the plasma profiles. These variables do not change by themselves during discharge, and thus they change only by data assimilation. Therefore, it is considered that the uncertainties included in d and v are larger than those in c_{e} and c_{i}. The smoothed estimates of model parameter indicate the elements missing in the employed simulation model and required to predict or reproduce the time evolution of the system. The estimates also give the key to improve the employed simulation model for higher performance.

To examine the validity of the estimated model parameters (variables 4-13 in Table 3.2), we perform the TASK3D simulations using the time series of estimated parameters and verify if the experimental time series data can be reproduced. Figure 3.18 shows the simulation results of electron and ion temperatures by TASK3D for 5 discharges out of the 12 discharges. The simulation results using the smoothed estimates can reproduce the experimental time-series data with high accuracy. Figure 3.19 shows the comparisons between the smoothed estimates and the simulation results using the smoothed estimates of model parameters. Table 3.4 lists the RMSE and RMSPE between smoothed estimates and the simulation results of density, electron temperature, and ion temperature at ρ=0.1 and 0.6. It is confirmed that the estimated model parameters can reproduce the experimental time-series data with low error, and these results

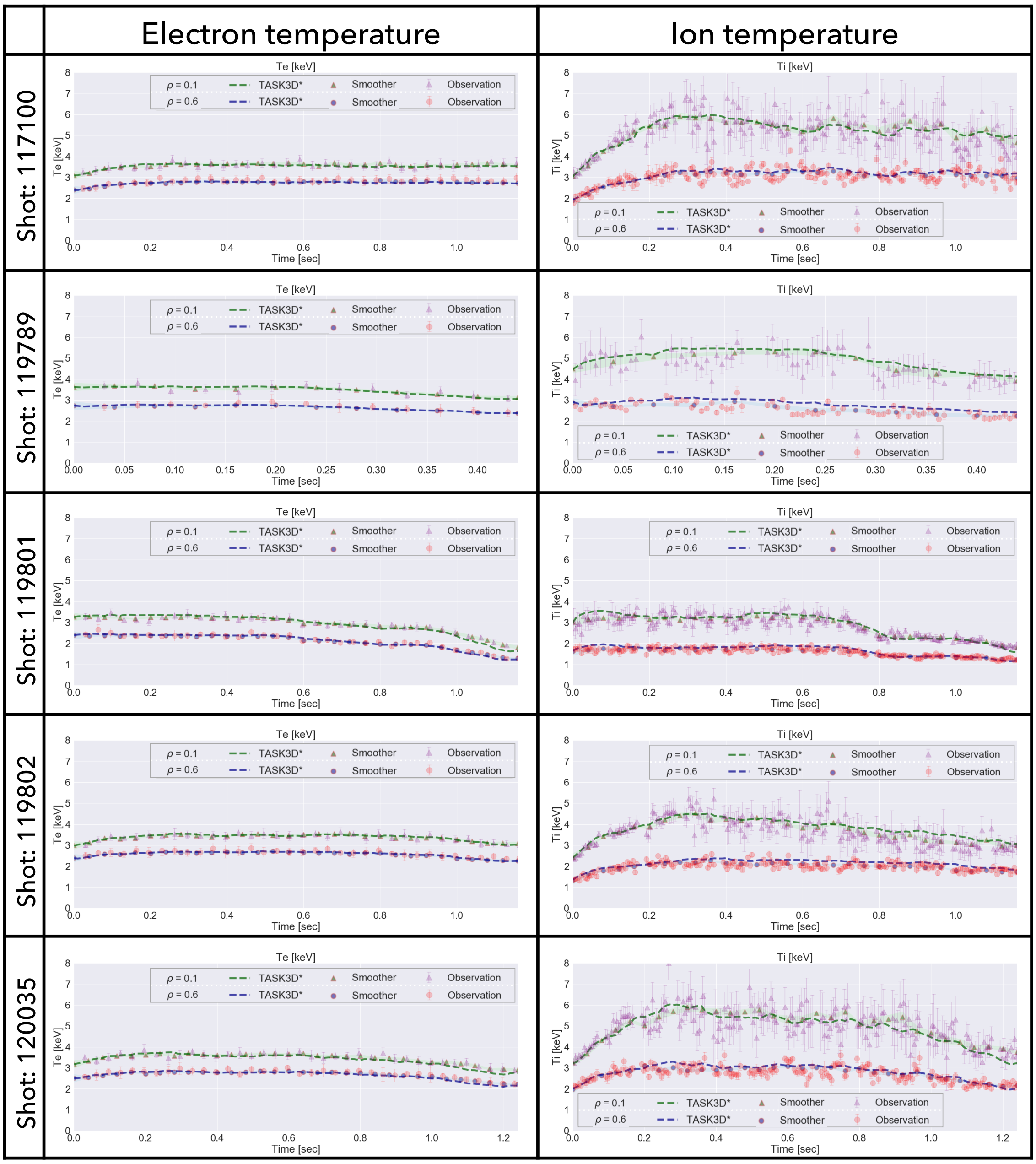


Figure 3.18: Simulation results of electron and ion temperatures by TASK3D using the smoothed estimates of the model parameters.

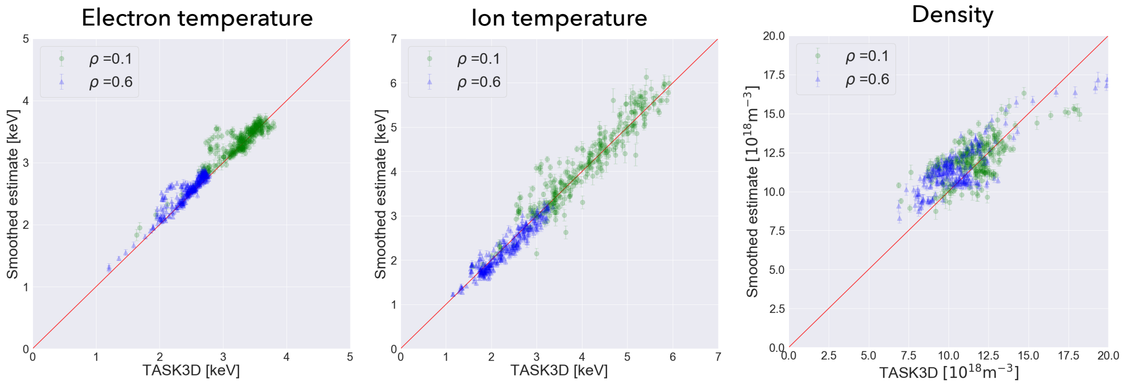


Figure 3.19: Comparisons between the smoothed estimates and the TASK3D simulation using the smoothed estimates of the model parameters at ρ=0.1 and 0.6 for all the 12 shots. The error bars are the standard deviations of smoothed ensemble.

Table 3.4: RMSE and RMSPE between the smoothed estimates and the TASK3D simulation using the smoothed estimates of the model parameters.

	ρ	RMSE	RMSPE
Electron temperature	0.1	0.123	3.17%
[keV]	0.6	0.0958	3.19%
Ion temperature	0.1	0.323	5.29%
[keV]	0.6	0.165	4.52%
Density	0.1	0.109	11.32%
[10^{19}m^{-3}]	0.6	0.137	12.75%

reinforce the validity of the estimation by ASTI (EnKS). The RMSPEs of density are larger than those of electron and ion temperatures. This is also thought to be due to the constant model for the particle turbulent diffusivity, which does not change even if the density and temperature profiles change. It is considered that the accuracy can be further enhanced by improving the assumed model.

We can see that ASTI is able to estimate reasonable model parameters that satisfy the dynamic constraints by the system model and reproduce the experimental data. In order to verify the effectiveness of the estimated model parameters for other discharges or to build a model that can predict other discharges, it is necessary to regress the estimation results of many discharges or to assimilate observation data of many discharges simultaneously to extract properties common to the discharges. These are issues that we would like to address in the future.

3.2.5 Prediction performance

To investigate the prediction performance, we vary the data assimilation cycle τ_{DA} from 40 ms to 560 ms for the 12 discharges. Figure 3.20 shows the τ_{DA} dependence of the prediction error (RMSE between the prediction and the smoothed estimate) for density, electron temperature, and ion temperature. Prediction performance depends

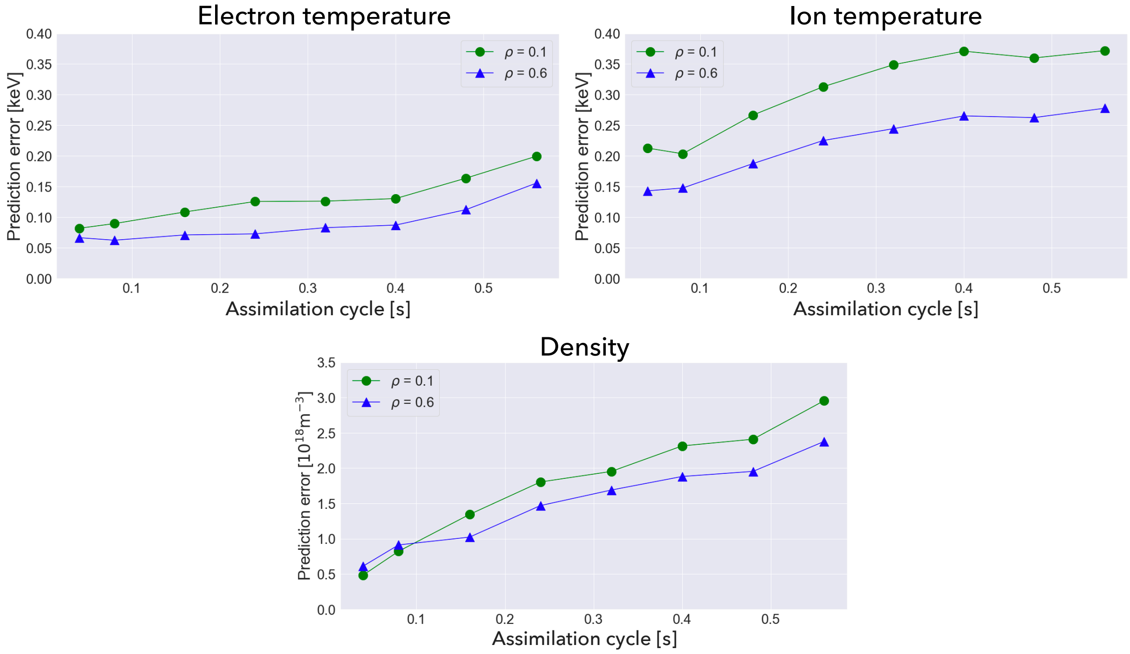


Figure 3.20: Prediction errors (RMSE between the prediction and the smoothed estimates) at ρ=0.1 and 0.6 when changing the data assimilation cycle.

on the time scale in which the model parameters should change by data assimilation. For shorter assimilation cycle than the time scale, the EnKF can follow the temporal change of the model parameters and predict the time evolution of the system with high accuracy. On the other hand, for longer assimilation cycle than the time scale, the EnKF can not follow the temporal change of the model parameters and the prediction error becomes large. Figure 3.20 shows that the prediction error of electron temperature stays low (around 0.1 keV at ρ=0.1) up to 0.4 s, but rises beyond it. The EnKF can follow the change of the parameters associated with electron heat transport if the data assimilation cycle is shorter than 0.4 s. It indicates that the time scale of the parameter changes required to predict and reproduce the time evolution of electron temperature is around 0.4 s. The prediction error of ion temperature stays low up to 0.1 s, rises from there, and saturates around 0.4 s, where the error is 0.35 keV (ρ=0.1). Thus, we can infer that the time scale of the parameters associated with ion heat transport is around 0.1 s. These time scales correspond to the temporal variation of electron and ion temperatures in the 12 shots of NBI-heated plasma. In fact, the electron temperature changes slowly, while the ion temperature changes significantly on the order of 0.1 seconds at the start of heating as shown in Fig. 3.18.

The prediction error of density monotonically increases with the assimilation cycle. It is partly because the constant model is employed for the turbulent particle diffusivity. It indicates that the state variables associated with particle transport need to change in a faster time scale than the assimilation cycle and the uncertainty contained in the particle transport models is larger than that in the heat transport models.

These prediction calculations takes four minutes for a second in actual plasma including the calculation of the NBI heating (the assimilation cycle is 40 ms and the time step of TASK3D's transport simulation is 1 ms). For these calculations, we have used 500 cores of IFERC-CSC (Intel Xeon Gold 6148) and assigned one core to one ensemble member. The calculation time has been significantly reduced compared to the calculation in Section 3.1 (more than ten hours including the calculation for NBI heating), using the reduced NBI heating model, FIT3D-RC, and saving the dimension of the state vector by B-spline interpolation. However, further speeding up is required to use ASTI for the real-time prediction of the fusion plasma behavior. We plan to introduce reduced models like the NBI heating model and parallel computing to TASK3D. In addition, as described in Section 3.1, we can also shorten the calculation time by making the computational grid coarser and the time step longer according to the actual application.

3.3 Summary

We have applied ASTI to the experimental time-series data of NBI-heated plasmas in LHD. In Section 3.1, we assimilated the observation data into only heat transport simulation and estimated mainly the thermal turbulent diffusivities. In Section 3.2, we assimilated the observation data into particle and heat transport simulation and estimated the parameters of turbulent transport and NBI heating. There, the reduced NBI heating model has been employed to speed up the prediction calculation by TASK3D, and the uncertainty increased by the reduction is compensated by data assimilation adjusting the model parameters. In both cases, the predicted radial profiles of density, electron temperature, and ion temperature by ASTI (EnKF) have agreed well with the observed profiles, and the errors between the prediction and observation are significantly reduced from the TASK3D simulation (without data assimilation).

In addition, we have obtained the radial profiles and temporal changes of the model parameters that can reproduce the experimental time-series data using the EnKS. For all the discharges, it has been confirmed that the TASK3D simulations using the smoothed estimates of the model parameters reproduce the experimental time-series data with high accuracy. Furthermore, the estimates of the transport model parameters have characteristic structures and suggest missing elements in the transport models. Such information is useful for improving the transport models. Through these assimilatios, we have demonstrated the effectiveness and validity of ASTI for accurate prediction and analysis of fusion plasmas.

The prediction performance of ASTI has been examined by changing the data assimilation cycle in Section 3.2.5. We have seen that the prediction performance strongly depends on the time scale in which the model parameters in the state vector should change to approximate the behavior of the real system. This time scale can be extended by improving the employed simulation model. We plan to develop a method to build high-performance transport models using data assimilation.

The calculation of ASTI took about four minutes for a second in actual plasma using FIT3D-RC. High-speed and accurate integrated transport simulation is realized by the data assimilation and the reduced model compared to conventional simulations. Data assimilation can also be used for the purposes of transport analyses such as model comparison, quantification of model deviations, estimation of unobservable variables, correction of observed values, and interpolation of observation data. Therefore, it is possible to employ ASTI as an analysis system for the transport analyses such as estimation of turbulent transport model, verification of isotope effect, spatiotemporal interpolation of time series data of radial profile. It is expected that data assimilation plays important roles also in the field of fusion research.

Chapter 4

Data Assimilation for Control

The ultimate goal of this study is to construct a system that continuously monitors and controls the state of fusion plasma for operation of future fusion reactors. However, fusion plasmas, which are confined in a magnetic field and should be maintained under extremely high temperature and density conditions, are typical complex systems [4] and their behavior is difficult to predict and control. Predicting the behavior of complex systems with simulation models generally involves large uncertainties, because it is inherently difficult to model all the components of a complex system and their interactions with sufficient accuracy [68–70]. Therefore, model-based control of complex systems is a challenging task, particularly when the system model is nonlinear and involves many uncertain elements. In the case of fusion plasmas, we must consider conditions outside the plasma (e.g., wall condition [10]) as well as inside the plasma, further increasing the uncertainty of the simulation [11].

To achieve model-based control of complex systems, we require a method to estimate optimal control inputs that can produce the target state while suppressing the model uncertainties. This control is classified as adaptive model predictive control and a challenging and open problem when the system is nonlinear and contains many uncertain elements [71].

In this section, to address this control problem, we propose a data assimilation framework, named as "data assimilation and control system" (DACS) [72]. The DACS framework includes system model updates and optimal control-input estimation. In general, data assimilation is a technology that integrates observations and simulation models by optimizing state variables from limited observed information. Thus, the existing data assimilation framework does not include control processes. The DACS framework is an extension of the sequential Bayesian filter (Section 2.1) to realize an adaptive model predictive control. It can also be one approach to achieve a digital twin [73–76]. In addition, since the DACS framework does not require additional prediction steps, the framework can even be applied to a large system in which iterative model prediction is prohibitive due to computational burden. This framework can be applied to fusion plasma control and a wide range of other control problems in which system models can be prepared even in a simplified manner.

4.1 Data assimilation and control system

The DACS framework is an extension of the sequential Bayesian filter to include control processes as shown in Fig.4.1. The procedure of the sequential Bayesian filter comprises two steps on the state distribution, i.e., prediction and filtering. The DACS framework differs from the sequential Bayesian filter mainly in the following two points.

(a) Real system behavior is controlled to approach a given target-state time series.

(b) The time in numerical space is always ahead of that in real space because control systems must constantly predict future system states and estimate control inputs.

The DACS framework involves control estimation by assimilating target-state information and model optimization by assimilating observed information. The second constraint above requires the assimilation of observed information into the latest predicted distribution. Furthermore, simulation models for complex systems generally require non-negligible prediction time compared to the time scale of control, and iterative prediction is computationally prohibitive. Therefore, it is desirable to construct control algorithms with as few predictive calculations as possible. The DACS framework allows us to construct a control algorithm without overlapping prediction intervals, i.e., with only one forward computation. In addition, a simplified system model can be employed on condition that the model is updated sequentially via observation information. We believe that the DACS framework is also applicable to large-scale complex systems, e.g., traffic control, virus spread prevention measures, and river level control.

4.1.1 State-space model

Consider a situation where control input is adjusted at every time interval Δt_z, and system state is observed at every time interval Δt_y. In this situation, we control the system behavior to approach a given target-state time series. For simplicity, we assume that $\Delta t_y = n\Delta t_z$ $(n \in \mathbf{N})$ and introduce the time notation,

$$t_{i,j} = t_{0,0} + i\Delta t_y + j\Delta t_z, \tag{4.1}$$

where $t_{0,0}$ is the initial time, and i and j are integers. Here, $\Delta t_y = n\Delta t_z$; thus, $t_{i,n} = t_{i+1,0}$. When time is written in a subscript, it is denoted as $A_{(i,j)}$, which indicates variable A at time $t_{i,j}$. Variables related to the observations are denoted by omitting j, because $j = 0$. For example, let $\mathbf{y}_i$ be the observation vector at time $t_{i,0}$. In addition, we use the multiple time notation $A_{(0,0):(i,j)} \equiv \{A_{(0,0)}, A_{(0,1)}, A_{(0,2)}, \cdots, A_{(1,0)}, \cdots, A_{(i,j)}\}$.

The state vector at time $t_{i,j}$, $\mathbf{x}_{(i,j)}$, is defined as follows:

$$\mathbf{x}_{(i,j)} = \begin{pmatrix} \tilde{\mathbf{x}}_{(i,j)} \\ \mathbf{u}_{(i,j)} \end{pmatrix}, \tag{4.2}$$

where $\tilde{\mathbf{x}}_{(i,j)}$ is the part of the state vector containing the system state and model parameters, which is used as the state vector in typical data assimilation. Vector $\mathbf{u}_{(i,j)}$ is the control input that determines the time evolution of the system from time $t_{i,j-1}$ to

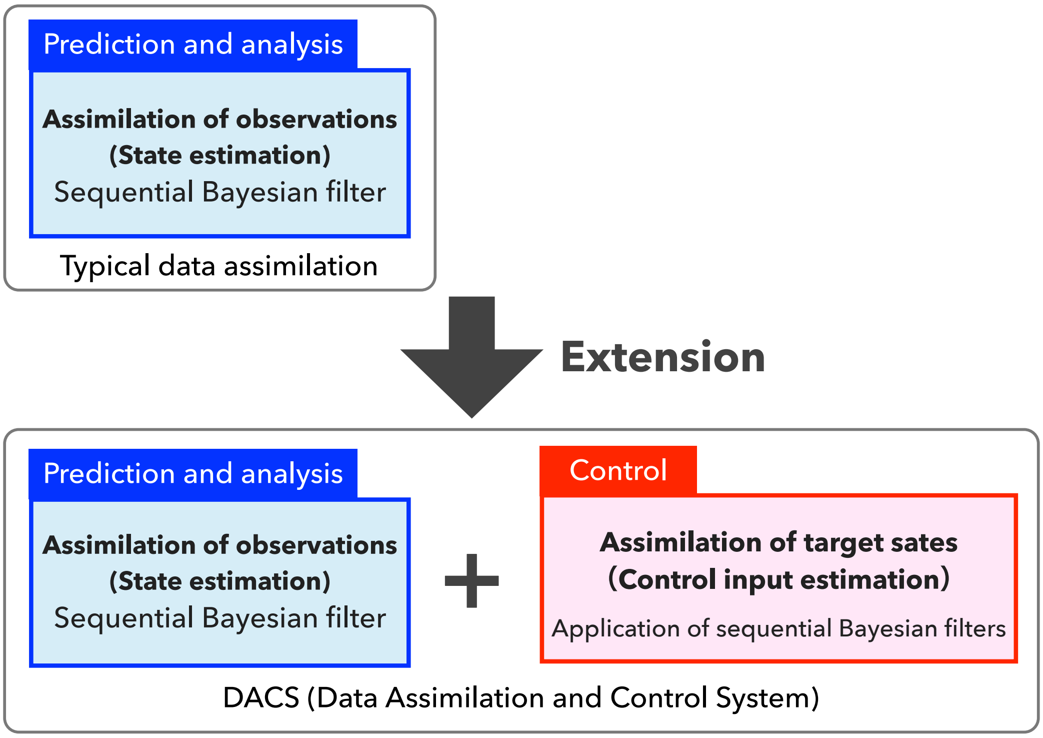

Prediction and analysis
Assimilation of observations
(State estimation)
Sequential Bayesian filter
Typical data assimilation
Extension
Prediction and analysis
Assimilation of observations
(State estimation)
Sequential Bayesian filter
+
Control
Assimilation of target sates
(Control input estimation)
Application of sequential Bayesian filters
DACS (Data Assimilation and Control System)

Figure 4.1: Concept of data assimilation and control system.

$t_{i,j} = t_{i,j-1} + \Delta t_z$. The target state of system $\mathbf{z}$ is prepared as time-series data in Δt_z increments, and the target-state time series can be either predetermined throughout all time steps or set adaptively at each time step depending on the operation. Here, we define the vector $\mathbf{u}^*_{(i,j)}$ as the control input estimated to produce the target state $\mathbf{z}_{(i,j)}$.

The DACS framework is based on the following state-space model:

$$\begin{aligned} \mathbf{x}_{(i,j+1)} &= f_{(i,j+1)}(\mathbf{x}_{(i,j)},\ \mathbf{v}_{(i,j+1)}), & (4.3)\\ \mathbf{z}_{(i,j)} &= h^{\mathbf{z}}_{(i,j)}(\mathbf{x}_{(i,j)}, \mathbf{w}^{\mathbf{z}}_{(i,j)}), & (4.4)\\ \mathbf{u}^*_{(i,j)} &= H^{\mathbf{u}}\mathbf{x}_{(i,j)} + \mathbf{w}^{\mathbf{u}}_{(i,j)}, & (4.5)\\ \mathbf{y}_i &= h^{\mathbf{y}}_i(\mathbf{x}_{(i,0)}, \mathbf{w}^{\mathbf{y}}_i). & (4.6) \end{aligned}$$

Eq. (4.3) is the system model which describes the time evolution of the system, $\mathbf{x}_{(i,j)} \rightarrow \mathbf{x}_{(i,j+1)}$, considering the effect of system noise $\mathbf{v}_{(i,j+1)}$. We assume that the value of $\mathbf{u}_t$ is constant during the prediction interval (Δt_z), i.e.,

$$\begin{aligned} \mathbf{u}_{(i,j+1)} &= \mathbf{u}_{(i,j)} + \mathbf{v}^{\mathbf{u}}_{(i,j+1)}, & (4.7)\\ \tilde{\mathbf{x}}_{(i,j+1)} &= f_{(i,j+1)}(\tilde{\mathbf{x}}_{(i,j)},\ \mathbf{u}_{(i,j+1)},\ \tilde{\mathbf{v}}_{(i,j+1)}). & (4.8) \end{aligned}$$

Here, the system noise for control input $\mathbf{v}^{\mathbf{u}}_{(i,j+1)}$ is added to the distribution $p(\mathbf{u}_{(i,j)})$ before the time evolution calculation. Eq. (4.4) represents the relationship between the state vector $\mathbf{x}_{(i,j)}$ and target-state vector $\mathbf{z}_{(i,j)}$ considering the target-state noise $\mathbf{w}^{\mathbf{z}}_{(i,j)}$.

In the same manner, Eqs. (4.5) and (4.6) represent the relationships between $\mathbf{x}_{(i,j)}$ and control input $\mathbf{u}^*_{(i,j)}$ with control-input noise $\mathbf{w}^{\mathbf{u}}_{(i,j)}$, and that between $\mathbf{x}_{(i,j)}$ and observation vector $\mathbf{y}_i$ with observation noise $\mathbf{w}^{\mathbf{y}}_i$, respectively. Note that Eqs. (4.4), (4.5) and (4.6) are referred to as "target-state model", "control-input model", and "observation model", respectively. Here, from Eq. (4.2), the control-input model is written using a matrix $H^{\mathbf{u}}$. In this section, we also assume that the ensemble approximation of the state distribution is employed.

4.1.2 Steps for control input estimation and system model update

The characteristic feature of the DACS framework is to control the time evolution of the system state by allowing the target state to act on the predicted distribution. In addition, the system model is optimized based on observed information to reduce prediction uncertainty. The DACS framework involves four main steps on the state distribution: prediction, z-filter, u-filter, and y-filter steps. The change in the probability distribution of state vector for each step can be summarized as follows.

- Prediction

$$p(\mathbf{x}_{(i,j)}|\mathbf{y}_{0:k}, \mathbf{u}^*_{(0,1):(i,j)}) \to p(\mathbf{x}_{(i,j+1)}|\mathbf{y}_{0:k}, \mathbf{u}^*_{(0,1):(i,j)}). \tag{4.9}$$

- z-filter

$$p(\mathbf{x}_{(i,j+1)}|\mathbf{y}_{0:k}, \mathbf{u}^*_{(0,1):(i,j)}) \to \mathbf{u}^*_{(i,j+1)} = E(\mathbf{u}_{(i,j+1)}|\mathbf{y}_{0:k}, \mathbf{u}^*_{(0,1):(i,j)}, \mathbf{z}_{(i,j+1)}). \tag{4.10}$$

- u-filter

$$p(\mathbf{x}_{(i,j+1)}|\mathbf{y}_{0:k}, \mathbf{u}^*_{(0,1):(i,j)}) \to p(\mathbf{x}_{(i,j+1)}|\mathbf{y}_{0:k}, \mathbf{u}^*_{(0,1):(i,j+1)}). \tag{4.11}$$

- y-filter

$$p(\mathbf{x}_{(i,j)}|\mathbf{y}_{0:k}, \mathbf{u}^*_{(0,1):(i,j)}) \to p(\mathbf{x}_{(i,j)}|\mathbf{y}_{0:k+1}, \mathbf{u}^*_{(0,1):(i,j)}). \tag{4.12}$$

Here, in Eq. (4.12), the time relation $t_{i,j} = t_{k+1} + h$ holds, using the prediction horizon $h(\geq \Delta t_z)$ as described below.

In the prediction step, Eq. (4.9), the state distribution Δt_z ahead is predicted based on the system model ($t_{i,j} \to t_{i,j+1}$). Given the distribution $p(\mathbf{x}_{(i,j)}|\mathbf{y}_{0:k}, \mathbf{u}^*_{(0,1):(i,j)})$, the predicted distribution $p(\mathbf{x}_{(i,j+1)}|\mathbf{y}_{0:k}, \mathbf{u}^*_{(0,1):(i,j)})$ is calculated, where $t_{k,0}$ is the latest observation time ($t_{k,0} < t_{i,j}$). When employing an ensemble approximation of the state distribution, prediction step is performed by computing the time evolution of the ensemble members by the system model [29], as shown in Fig 4.2(a). Here, the predicted distribution contains uncertainties in the system state before the time evolution, the parameters of the employed models, and the control inputs.

The z-filter, Eq. (4.10), estimates the control input $\mathbf{u}^*_{(i,j+1)}$ from the predicted distribution and the target state $\mathbf{z}_{(\mathbf{i},\mathbf{j}+\mathbf{1})}$. The distribution $p(\mathbf{u}_{(i,j+1)}|\mathbf{y}_{0:k}, \mathbf{u}^*_{(0,1):(i,j)}, \mathbf{z}_{(i,j+1)})$ is obtained from the distribution $p(\mathbf{x}_{(i,j+1)}|\mathbf{y}_{0:k}, \mathbf{u}^*_{(0,1):(i,j)}, \mathbf{z}_{(i,j+1)})$ via the following

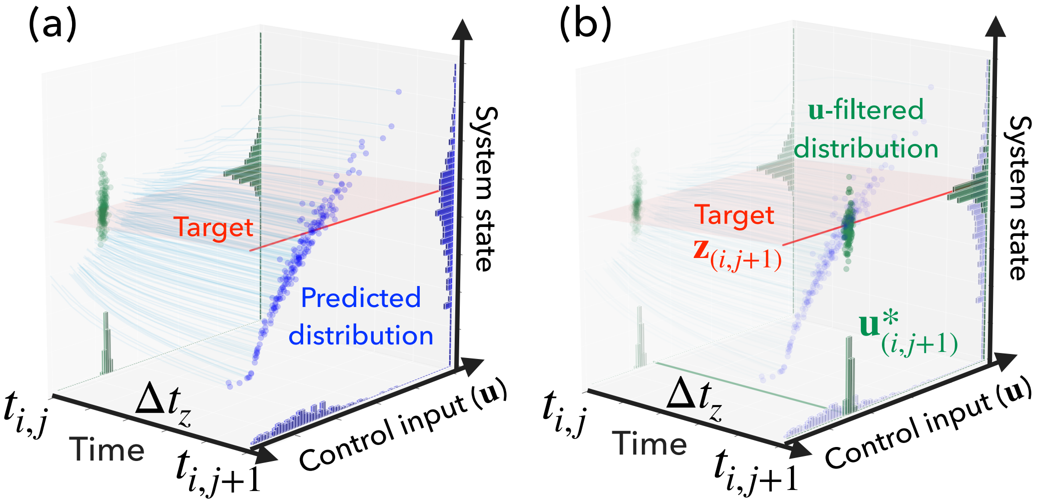


Figure 4.2: Changes in ensemble members at the prediction (a), z-filter and u-filter (b).

marginalization:

$$\begin{aligned}
&p(\mathbf{u}_{(i,j+1)}|\mathbf{y}_{0:k},\mathbf{u}^*_{(0,1):(i,j)},\mathbf{z}_{(i,j+1)}) \\
&= \int p(\mathbf{u}_{(i,j+1)},\tilde{\mathbf{x}}_{(i,j+1)}|\mathbf{y}_{0:k},\mathbf{u}^*_{(0,1):(i,j)},\mathbf{z}_{(i,j+1)})\mathrm{d}\tilde{\mathbf{x}}_{(i,j+1)}, \\
&= \int p(\mathbf{x}_{(i,j+1)}|\mathbf{y}_{0:k},\mathbf{u}^*_{(0,1):(i,j)},\mathbf{z}_{(i,j+1)})\mathrm{d}\tilde{\mathbf{x}}_{(i,j+1)}. \qquad (4.13)
\end{aligned}$$

When using an ensemble approximation of the state distribution, this marginalization can be performed by simply removing $\tilde{\mathbf{x}}$ part from the ensemble of the distribution $p(\mathbf{x}_{(i,j+1)}|\mathbf{y}_{0:k},\ \mathbf{u}^*_{(0,1):(i,j)},\mathbf{z}_{(i,j+1)})$. The distribution $p(\mathbf{x}_{(i,j+1)}|\mathbf{y}_{0:k},\mathbf{u}^*_{(0,1):(i,j)},\mathbf{z}_{(i,j+1)})$ can be calculated by assimilating the target state $\mathbf{z}_{(i,j+1)}$ to the predicted distribution $p(\mathbf{x}_{(i,j+1)}|\mathbf{y}_{0:k},\ \mathbf{u}^*_{(0,1):(i,j)})$ using the target-state model, Eq. (4.4). This procedure is a Bayesian filter that can be implemented using the EnKF or the PF. Here, we define the optimal control input $\mathbf{u}^*_{(i,j+1)}$ as the expected value (ensemble mean) as follows:

$$\mathbf{u}^*_{(i,j+1)} = \int \mathbf{u}_{(i,j+1)}p(\mathbf{u}_{(i,j+1)}|\mathbf{y}_{0:k},\mathbf{u}^*_{(0,1):(i,j)},\mathbf{z}_{(i,j+1)})\mathrm{d}\mathbf{u}_{(i,j+1)}. \qquad (4.14)$$

This control-input estimation is robust against the uncertainties in the system state at time $t_{i,j}$ and the model parameters [77,78].

The u-filter, Eq. (4.11), estimates the predicted distribution controlled by $\mathbf{u}^*_{(i,j+1)}$, $p(\mathbf{x}_{(i,j+1)}|\mathbf{y}_{0:k},\mathbf{u}^*_{(0,1):(i,j+1)})$, from the predicted distribution. The u-filter is executed by assimilating the estimated input $\mathbf{u}^*_{(i,j+1)}$ to the predicted distribution $p(\mathbf{x}_{(i,j+1)}|\mathbf{y}_{0:k},\ \mathbf{u}^*_{(0,1):(i,j)})$ using a Bayesian filter with the target-state model, Eq. (4.5), as shown in Fig. 4.2(b). The u-filtered distribution corresponds to the expected state distribution when $\mathbf{u}^*_{(i,j+1)}$ is input, in other words, the predicted distribution without the uncertainty of the control input. The remaining uncertainties work to carry the observed

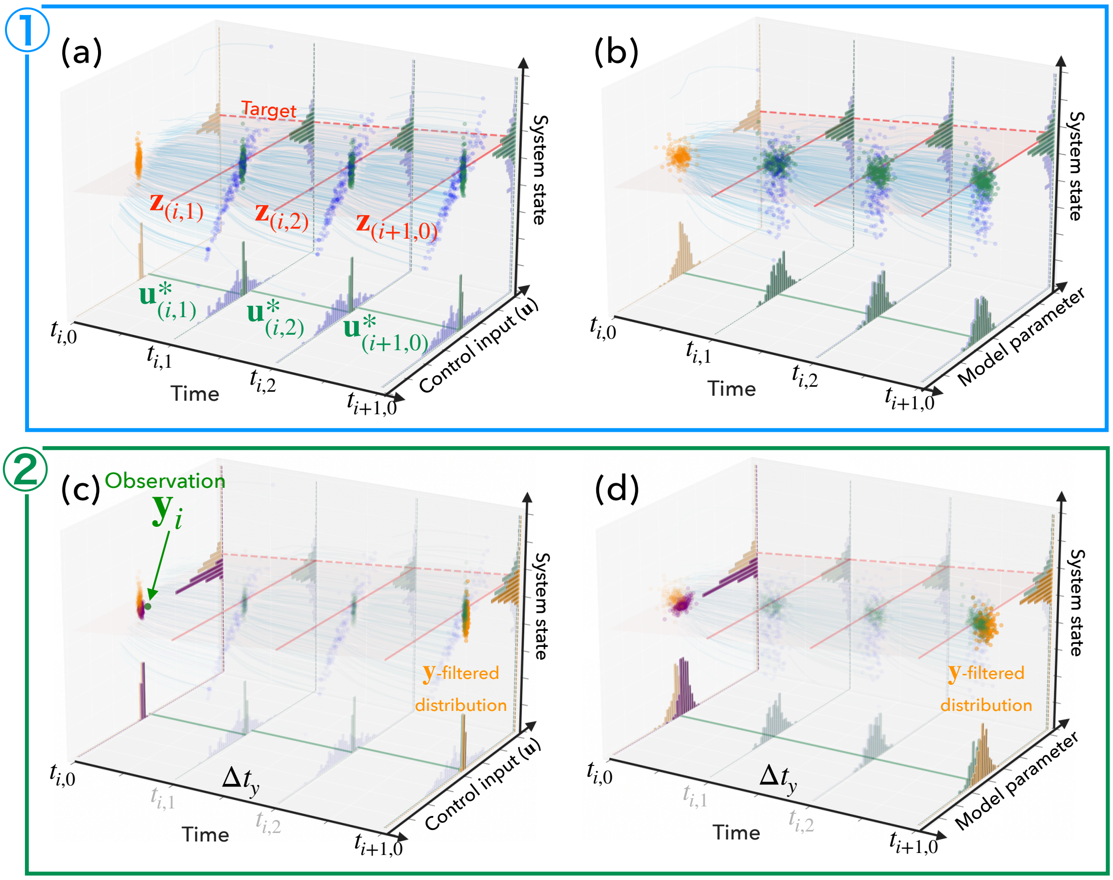


Figure 4.3: Changes in ensemble members in ① prediction and control-input estimation (focuses on control inputs (a); focuses on model parameters (b)), and ② y-filter (focuses on control inputs (c); focuses on model parameters (d)) for $n = 3$ case.

information into the future state distribution at the y-filter step. For the estimation of the predicted distribution controlled by $\mathbf{u}^*_{(i,j+1)}$, it is more accurate to recalculate the prediction step from $t_{i,j}$ using $\mathbf{u}^*_{(i,j+1)}$ than to perform this filtering. Note that the u-filter eliminates this second prediction calculation, thereby reducing the computational cost of control algorithms. Repeated execution of the prediction, z-filter, and u-filter steps evolves the state distribution over time while estimating the control inputs that produce the target state as shown in Fig 4.3①.

The newly observed data $\mathbf{y}_{k+1}$ is reflected in the latest u-filtered distribution by the y-filter, Eq. (4.12). The y-filter estimates the distribution $p(\mathbf{x}_{(i,j)}|\mathbf{y}_{0:k+1}, \mathbf{u}^*_{(0,1):(i,j)})$, where $t_{i,j}$ is the time of the latest u-filtered distribution. Since it is necessary to predetermine control ahead of the observation time, the latest u-filtered distribution represents the future state beyond the given observation time ($t_{i,j} > t_{k+1,0}$). We define this time difference as the prediction horizon h ($\geq \Delta t_z$), i.e.,

$$h = t_{i,j} - t_{k+1}. \tag{4.15}$$

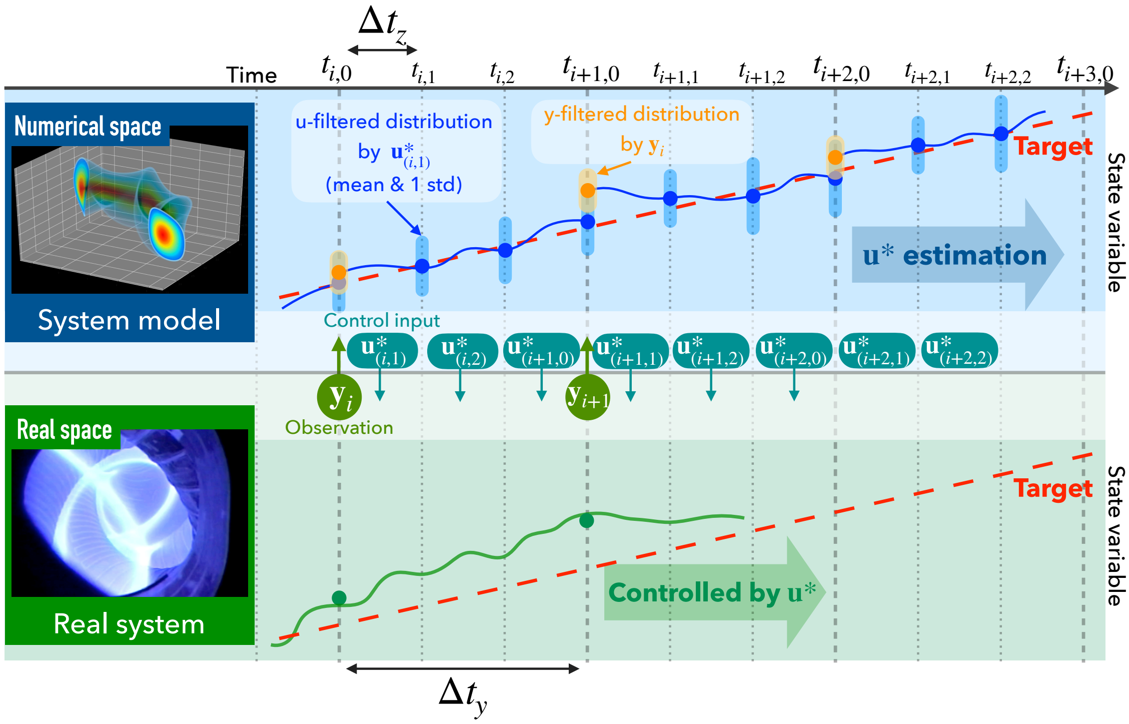


Figure 4.4: Time relationships in control algorithm. An application example of fusion plasma (LHD) control is illustrated. The time relationship to control a state variable (e.g., electron temperature) is shown separately in numerical space (state space) and real space for the $\Delta t_y = 3\Delta t_z$ and $h = \Delta t_y$ case. In this case, when the plasma in real space is controlled by $\mathbf{u}^*_{(i+1,1):(i+2,0)}$, the control inputs $\mathbf{u}^*_{(i+2,1):(i+3,0)}$ are estimated in the numerical space.

At time t in the real space, the control input has been estimated previously and determined up to at least $t+h$; thus, the observed information at t is reflected to the control estimation from $t+h$. The y-filter is executed by the assimilation of the observation data to the joint distribution of state vectors at two time points, i.e., the observation time t_{k+1} and the time of the latest u-filtered distribution $t_{i,j}$, as shown in Fig 4.3②. By concatenating the u-filtered ensemble at $t_{i,j}$ and the stored ensemble at t_{k+1}, the ensemble approximating the joint distribution $p(\mathbf{x}_{(i,j)}, \mathbf{x}_{(k+1,0)}|\mathbf{y}_{0:k}, \mathbf{u}^*_{(0,1):(i,j)})$ can be obtained. The filtered distribution $p(\mathbf{x}_{(i,j)}, \mathbf{x}_{(k+1,0)}|\mathbf{y}_{0:k+1}, \mathbf{u}^*_{(0,1):(i,j)})$ can be calculated by assimilating $\mathbf{y}_{k+1}$ to $p(\mathbf{x}_{(i,j)}, \mathbf{x}_{(k+1,0)}|\mathbf{y}_{0:k}, \mathbf{u}^*_{(0,1):(i,j)})$, and we obtain the ensemble of the distribution $p(\mathbf{x}_{(i,j)}|\mathbf{y}_{0:k+1}, \mathbf{u}^*_{(0,1):(i,j)})$ by marginalization of $\mathbf{x}_{(k+1,0)}$. Note that the y-filter also eliminates the computational cost of recalculating the system time evolution from the observation time to the latest prediction time. Using the u-filter and y-filter, we can construct a control algorithm without overlapping prediction intervals. Therefore, it can be applied even to systems where prediction is computationally expensive.

4.1.3 Construction of control algorithm

We can construct a control algorithm by setting the parameters $n(\geq 1)$ and $h(\geq \Delta t_z)$. To determine these parameters, we should consider the processing of the observed information, the communication time, and the computational costs of the prediction and filtering. In this section, we discuss a control algorithm constructed for $n = 3$ $(\Delta t_y = 3\Delta t_z)$ and $h = \Delta t_y$, as a specific example. Figure 4.4 shows the time relationship in this case. In the control algorithm, each step functions as follows.

- Prediction

$$p(\mathbf{x}_{(i,j)}|\mathbf{y}_{0:i-1}, \mathbf{u}^*_{(0,1):(i,j)}) \to p(\mathbf{x}_{(i,j+1)}|\mathbf{y}_{0:i-1}, \mathbf{u}^*_{(0,1):(i,j)}). \tag{4.16}$$

- z-filter

$$p(\mathbf{x}_{(i,j+1)}|\mathbf{y}_{0:i-1}, \mathbf{u}^*_{(0,1):(i,j)}) \to \mathbf{u}^*_{(i,j+1)} = E(\mathbf{u}_{(i,j+1)}|\mathbf{y}_{0:i-1}, \mathbf{u}^*_{(0,1):(i,j)}, \mathbf{z}_{(i,j+1)}). \tag{4.17}$$

- u-filter

$$p(\mathbf{x}_{(i,j+1)}|\mathbf{y}_{0:i-1}, \mathbf{u}^*_{(0,1):(i,j)}) \to p(\mathbf{x}_{(i,j+1)}|\mathbf{y}_{0:i-1}, \mathbf{u}^*_{(0,1):(i,j+1)}). \tag{4.18}$$

- y-filter

$$p(\mathbf{x}_{(i+1,0)}|\mathbf{y}_{0:i-1}, \mathbf{u}^*_{(0,1):(i+1,0)}) \to p(\mathbf{x}_{(i+1,0)}|\mathbf{y}_{0:i}, \mathbf{u}^*_{(0,1):(i+1,0)}). \tag{4.19}$$

The flow of state distributions absorbing the observed and target-state information is illustrated in Fig. 4.5. Given $p(\mathbf{x}_{(i,0)}|\mathbf{y}_{0:i-1}, \mathbf{u}^*_{(0,1):(i,0)})$ (♡ in Fig. 4.5), the u-filtered distribution Δt_y ahead, $p(\mathbf{x}_{(i+1,0)}|\mathbf{y}_{0:i-1}, \mathbf{u}^*_{(0,1):(i+1,0)})$ (♠), can be calculated by repeating the prediction, z-filter, and u-filter steps, while estimating the control inputs $\mathbf{u}^*_{(i,1):(i+1,0)}$. When observation $\mathbf{y}_i$ is obtained, the u-filtered distribution is modified by the y-filter to $p(\mathbf{x}_{(i+1,0)}|\mathbf{y}_{0:i}, \mathbf{u}^*_{(0,1):(i+1,0)})$ (♣). From this distribution, the predictive calculation is performed again to $t_{i+2,0}$, and control can proceed. Projecting the 3D flow in Fig. 4.5 onto the **xy**-plane, the 2D flow (gray solid line) appears. This 2D flow is the same as that shown in Fig. 2.1 (procedure of the sequential Bayesian filter), except that the times of observation are shifted. It follows from this that the DACS framework is a data assimilation framework where the prediction step of the sequential Bayesian filter is extended to the three steps, i.e., prediction, z-filter, and u-filter. Under the assumption that the observations are available from $t_{0,0}$, the procedure of the control algorithm can be summarized as follows.

Generate the initial ensemble members and store them as the initial y-filtered ensemble. For $i = 0, \cdots, i_{\max}$:

1. Estimate control inputs $\mathbf{u}^*_{(i,1):(i+1,0)}$ and u-filtered ensemble at $t_{i+1,0}$ (Fig 4.3①): For j=0,1,2:

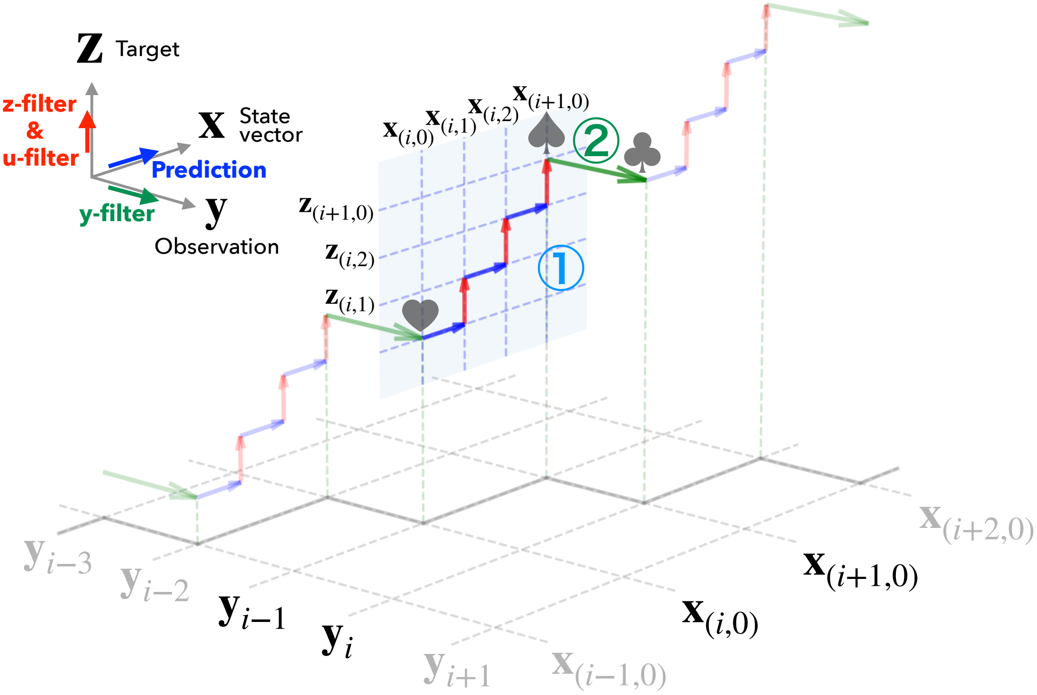


Figure 4.5: Assimilation flow of observation data and target state for a control algorithm ($\Delta t_y = 3\Delta t_z$ and $h = \Delta t_y$).

(a) Calculate time evolution of ensemble members from $t_{i,j}$ to $t_{i,j+1}$ ($t_{i,3} = t_{i+1,0}$).

(b) Execute z-filter using target $\mathbf{z}_{(i,j+1)}$ and ensemble obtained at 1-(a) and estimate control input $\mathbf{u}^*_{(i,j+1)}$.

(c) Execute u-filter using $\mathbf{u}^*_{(i,j+1)}$ and ensemble obtained at 1-(a).

2. Observe the system state at $t_{i,0}$ and obtain $\mathbf{y}_i$.

3. Update latest u-filtered ensemble (Fig 4.3②):

   (a) Concatenate y-filtered ensemble at $t_{i,0}$ stored at previous y-filter and u-filtered ensemble at $t_{i+1,0}$.

   (b) Execute y-filter using $\mathbf{y}_i$ and the concatenated ensemble.

   (c) Extract the part of $t_{i+1,0}$ from the filtered concatenated ensemble as the y-filtered ensemble and store it for the next y-filter.

4.2 ASTI as a control system

We implement the control algorithm discussed in Section 4.1.3 in ASTI. In this thesis, we implement all the filters using the EnKF. When the state variables are linked to each other via strong nonlinearities, and the state distribution deviates significantly from a Gaussian distribution, we should employ other filters (e.g., the PF) for implementation. In the case of EnKF, the state space model (Eqs. (4.3)-(4.6)) is expressed as follows:

$$\begin{aligned}
\mathbf{x}_{(i,j+1)} &= f_{(i,j+1)}(\mathbf{x}_{(i,j)},\ \mathbf{v}_{(i,j+1)}), && (4.20)\\
\mathbf{z}_{(i,j)} &= H^{\mathbf{z}}\mathbf{x}_{(i,j)} + \mathbf{w}^{\mathbf{z}}_{(i,j)}, && (4.21)\\
\mathbf{u}^{*}_{(i,j)} &= H^{\mathbf{u}}\mathbf{x}_{(i,j)} + \mathbf{w}^{\mathbf{u}}_{(i,j)}, && (4.22)\\
\mathbf{y}_{i} &= H^{\mathbf{y}}\mathbf{x}_{(i,0)} + \mathbf{w}^{\mathbf{y}}_{i}. && (4.23)
\end{aligned}$$

Here, the matrices $H^{\mathbf{z}}$ and $H^{\mathbf{y}}$ are the linear operators to project the state vector in each corresponding space. It is assumed that system noise $\mathbf{v}_{(i,j+1)}$ follows a Gaussian distribution with zero mean and covariance matrix $Q_{(i,j+1)}$, i.e., $\mathbf{v}_{(i,j+1)} \sim N(\mathbf{0}, Q_{(i,j+1)})$. Similarly, noises $\mathbf{w}^{\mathbf{z}}_{(i,j)}$, $\mathbf{w}^{\mathbf{u}}_{(i,j)}$, and $\mathbf{w}^{\mathbf{y}}_{i}$ are assumed to follow the probability distributions $N(\mathbf{0}, R^{\mathbf{z}}_{(i,j)})$, $N(\mathbf{0}, R^{\mathbf{u}}_{(i,j)})$, and $N(\mathbf{0}, R^{\mathbf{y}}_{i})$, respectively. These covariance matrices, $Q_{(i,j+1)}$, $R^{\mathbf{z}}_{(i,j)}$, $R^{\mathbf{u}}_{(i,j)}$, and $R^{\mathbf{y}}_{i}$ are key hyper parameters that determine overall control performance of the constructed algorithm. In addition, we can determine the priority of each variable in the target state and the importance of the observed variables through the target and observation noise. In this thesis, we use diagonal matrices for these covariance matrices, i.e., the covariance component of the noise is not considered. Spatial smoothness of the state variables is guaranteed by the B-spline interpolation discussed in Section 3.2.

The covariance matrix $Q_{(i,j+1)}$ controls the uncertainty of the state distribution. The system noise to $\mathbf{u}$ determines the magnitude of the state change and the focused region of $\mathbf{u}$ to look for the optimal control input. Note that a larger variance of Q for $\mathbf{u}$ is required to realize global estimation for control problems with jumps in optimal control input values. On the other hand, when the target state changes slowly or when the control input changes at a finite speed, such as the steering wheel of a car, a relatively small variance is sufficient for the system noise. The system noise to $\mathbf{u}$ is added before each prediction step is performed, and the system noise to $\tilde{\mathbf{x}}$ is only added after the y-filter is executed to prevent shrinkage of the distribution. Here, $Q_{(i,j+1)}$ should be determined considering the sophistication of the employed system model, the time series of the scheduled target state, and the control constraints due to the corresponding devices.

Covariance matrix $R^{\mathbf{z}}_{(i,j)}$ affects the performance of the z-filter and determines how close the system state gets to the target state in the control estimation. A large variance of $R^{\mathbf{z}}_{(i,j)}$ weakens the z-filter's force to attract the predicted distribution to the target state. As a result, the u-filtered distribution changes gradually (with a slower rate of change). Here, we determine the diagonal components of $R^{\mathbf{z}}_{(i,j)}$ at each z-filtering step as Eq. (4.24) using the proportionality coefficient r_z:

$$\left(R^{\mathbf{z}}_{(i,j)}\right)_{ll} = r_z^2 \left(H^{\mathbf{z}} V_{(i,j)} (H^{\mathbf{z}})^{\mathrm{T}}\right)_{ll}, \tag{4.24}$$

where $V_{(i,j)}$ is the covariance matrix of the ensemble that approximates the predicted distribution at $t_{i,j}$, $(\ )_{ll}$ denotes the l-th diagonal component, and superscript $^{\mathrm{T}}$ denotes matrix transposition.

Uncertainties in the control input can be considered through the covariance matrix $R^{\mathbf{u}}_{(i,j)}$. The rate of change of the control input $\mathbf{u}$ can be adjusted independently of the z filter, but the resulting effect is almost the same as the effect of the target state noise. Note that this uncertainty does not need to be considered in many cases, and the variance of $R^{\mathbf{u}}_{(i,j)}$ can be set sufficiently small. In the case of EnKFs, the variance can even be set to 0 if the dimension of $\mathbf{u}$ is sufficiently smaller than the number of ensemble members. When using the PF, care should be taken with this matrix to avoid degeneracy of the ensemble.

The covariance matrix $R^{\mathbf{y}}_i$ affects the performance of the y-filter and determines the impact of the observations on the state distribution, as with that in the sequential Bayesian filter. Here, the standard deviation of the observation noise is assumed to be proportional to the difference between the observation data and the mean of the state distribution as Section 3.1 and 3.2,

$$\left(R^{\mathbf{y}}_i\right)_{ll} = r_y^2 \left(\mathbf{y}_i - H^{\mathbf{y}} \hat{\mathbf{x}}_{(i,0)}\right)^2_l , \tag{4.25}$$

where r_y is the proportional coefficient, $\hat{\mathbf{x}}_{(i,0)}$ is the mean of the ensemble approximating $p(\mathbf{x}_{(i,0)}|\mathbf{y}_{0:i-1}, \mathbf{u}^*_{(0,1):(i,0)})$, and $(\ )_l$ represents the l-th element of the vector. Under this assumption, the variance of the y-filtered ensemble is maintained at an adequate magnitude. For a large difference between the predicted and observed responses, this assumption works to increase the variance of the observation noise, which results in more flexible state variable optimization. In contrast, for a small difference, this assumption works to reduce variance of the observation noise and prevents the variance of the state distribution from becoming excessively large. This assumption is also effective in terms of constructing a stable control system that is robust against observed values with large measurement errors.

In the next chapter, we discuss numerical experiments in controlling virtual (numerically created) fusion plasma. Though the numerical experiments, we demonstrate the control performance of ASTI and investigate the dependence of the control performance on the choice of hyper parameters and the discrepancies between the system model and the real system.

Chapter 5

Numerical Experiments to Control Virtual LHD Plasma

To demonstrate a practical application of the DACS framework, we consider the control of temperature and density in fusion plasma in LHD. In this section, we demonstrate the effectiveness of the DACS framework through numerical experiments to control a virtual LHD plasma (hydrogen plasma). We also investigate the dependence of control accuracy on noise parameters, which are key hyper parameters of the DACS framework, and on the discrepancy between the assumed system model and actual behavior.

5.1 Settings of numerical experiments

Again, we employ the integrated transport simulation code TASK3D as the system model of ASTI. We consider a problem of controlling the electron density and temperature at the plasma center when the radial profiles of electron density, electron temperature, and ion temperature are observed every Δt_y=0.6 s. We assume that Δt_z=0.2 s and $h = \Delta t_y$=0.6 s in consideration of the computational costs of the prediction and filters, communication time, and processing time of the observed information. The virtual plasma is generated in numerical space using TASK3D and controlled by ASTI, which implements the control algorithm discussed in Section 4.1.3 ($n = 3$ and $h = \Delta t_y$).

5.1.1 State-space model

Table 5.1 defines the state variables, target-state variables, and observation variables. The state vector $\mathbf{x}$ comprises $\tilde{\mathbf{x}}$ and $\mathbf{u}$, as in Eq. (4.2). In addition, $\tilde{\mathbf{x}}$ comprises the plasma state (n, T_e, T_i), and model parameters (d, c_e, c_i, ξ_T). In Table 5.1, the values with % as the unit represent the rate to determine the standard deviation in proportion to the mean of the state distribution. The variables with * are the numerical factors to optimize the corresponding model parameters, i.e., their value of 1 corresponds to a simple simulation with no modification by data assimilation. The following parameters with $'$ are used in the prediction calculation by TASK3D instead of the parameters

Table 5.1: State variables, target variables, and observation variables for numerical experiments to control virtual fusion plasma. Their dimension in the vectors (M_i), and their standard deviation of the initial state distribution (σ_I) and system noise (σ_Q) are also shown.

Variable			M_i	σ_I	σ_Q
$\tilde{\mathbf{x}}$	n	Density	11	3%	3%
	T_e	Electron temperature	11	3%	3%
	T_i	Ion temperature	11	3%	3%
	d	Turbulent particle diffusivity*	11	0.2	0.1
	c_e	Turbulent electron thermal diffusivity*	11	0.2	0.1
	c_i	Turbulent ion thermal diffusivity*	11	0.2	0.1
	ξ_T	Temperature of neutral particles at plasma edge*	1	0.1	0.05
$\mathbf{u}$	ξ_n	Density of neutral particles at plasma edge*	1	0	0.4
	P_ECH	ECH input power	1	0	0.7 MW
$\mathbf{z}$	$n_{\rho=0}$	Density at plasma center	1		
	$T_{\mathrm{e},\rho=0}$	Electron temperature at plasma center	1		
$\mathbf{y}$	n	Density	11		
	T_e	Electron temperature	11		
	T_i	Ion temperature	11		

without $'$:

$$D^{\mathrm{TB}\prime} = dD^{\mathrm{TB}},\ \chi_\mathrm{e}^{\mathrm{TB}\prime} = c_\mathrm{e}\chi_\mathrm{e}^{\mathrm{TB}},\ \chi_\mathrm{i}^{\mathrm{TB}\prime} = c_\mathrm{i}\chi_\mathrm{i}^{\mathrm{TB}},\ T_\mathrm{n}' = \xi_T T_\mathrm{n},\ n_\mathrm{n}' = \xi_n n_\mathrm{n}.$$

The radial profiles of the state variables are defined on 11 grid points ($\rho = 0, 0.1, 0.2, \cdots, 1$) in the state vector, and the radial profile in TASK3D is defined on 60 grid points. The radial profiles of the state variables are transformed to the profiles on the TASK3D's grid using the B-spline interpolation introduced in Section 3.2.

The control input $\mathbf{u}$ comprises the numerical factor for neutral density at the plasma edge ξ_n and the electron cyclotron heating (ECH) input power P_ECH. Here, ξ_n determines the particle source term and primarily affects the density profile, and P_ECH determines the heat source term and primarily affects the electron temperature profile. Although the equivalent of ξ_n in actual control is the intensity of the gas puff, for simplicity, we consider the density of neutral particles at the plasma edge in the numerical experiments. The ECH input power P_ECH takes discrete values from 0.5 to 5 MW in 0.5 MW increments due to equipment constraints.

In this example application, we control the density and electron temperature at the plasma center to increase along the predetermined target-state time series. The target state $\mathbf{z} = (n_{\rho=0}, T_{\mathrm{e},\rho=0})$ starts at (1×10^{19} m^{-3}, 2 keV), begins to increase from 2.4 s, and remains at (2×10^{19} m^{-3}, 5 keV) from 4.8 s, as shown in Figs. 5.1.

In the numerical experiments for the virtual plasma control, P_ECH and ξ_n affect the density and temperature profiles through the source terms P_e and S, respectively. The particle source S is primarily determined by the ionization of neutral particles evaluated by the AURORA module of TASK3D. AURORA calculates the component of S coming from the ionization of neutral particles using the plasma profiles and the density and

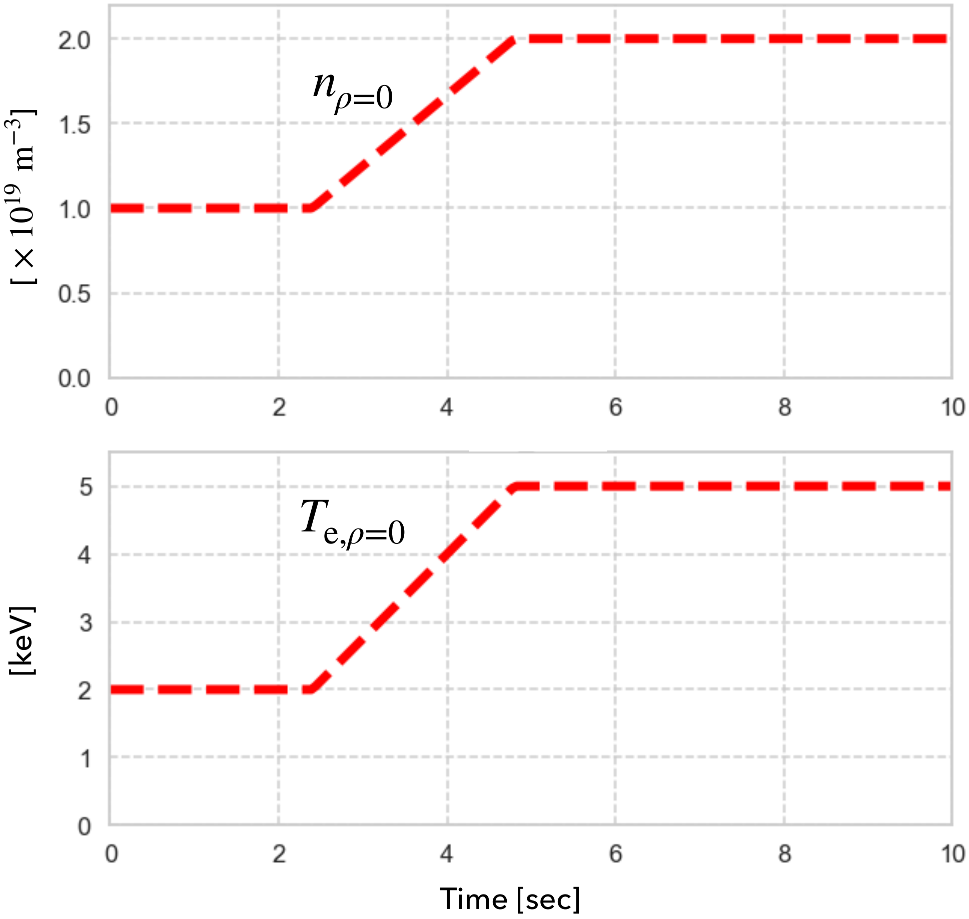


Figure 5.1: Target state of the numerical experiments

temperature of neutral particles at the plasma edge (n_{n} and T_{n}). The heating power P_s comprises the externally applied ECH, the power exchange between species, and the loss term by interaction with neutrals. Note that the ECH only contributes to the heating term of the electron. In this experiment, the following simple ECH model is employed:

$$P_{\mathrm{e}}^{\mathrm{ECH}}(\rho) = A\exp\left(-\frac{1}{2}\frac{(\mu_{\mathrm{ECH}}-\rho)^2}{\sigma_{\mathrm{ECH}}^2}\right), \tag{5.1}$$

where $\mu_{\mathrm{ECH}} = 0.1$ and $\sigma_{\mathrm{ECH}} = 0.05$, which are reasonable values from detailed ray-tracing calculations. The coefficient A is determined from ECH total input power given by

$$P_{\mathrm{ECH}} = \int_0^1 P_{\mathrm{e}}^{\mathrm{ECH}}(\rho)\mathcal{V}'(\rho)\mathrm{d}\rho. \tag{5.2}$$

The standard deviation of the system noise is fixed at the values shown in Table 5.1. We assume that uncertainties of the system model are mainly in the transport parameters. The standard deviations of the control-input noise are set to sufficiently small values, i.e., 0.05 MW for P_{ECH} and 0.02 for ξ_n, as described in Section 4.2. The parameters, r_z and r_y, which determine the target-state noise and observation noise, respectively, are described with the control results in Section 5.2

5.1.2 Virtual LHD plasma

To validate the adaptive capacity of DACS, we create a difference between the system model and the actual system. For this purpose, we assume that the transport models in the virtual plasma are different from that employed in the system model in ASTI. In this numerical experiment, we add $1/n$ dependent terms to the turbulent particle diffusivity D^{TB} and the turbulent electron thermal diffusivity $\chi_{\mathrm{e}}^{\mathrm{TB}}$ in the transport model of the virtual plasma. For D^{TB} in Eq. (2.43), the constant model,

$$D^{\mathrm{TB}} = d_{\mathrm{const}} \times d, \tag{5.3}$$

is employed in ASTI, whereas in the virtual plasma, the model with a $1/n$ dependent term,

$$D^{\mathrm{TB}} = 2d_{\mathrm{Alc}} \left((1-\mu) + \mu \frac{1}{n} \right), \tag{5.4}$$

is employed, where $d_{\mathrm{const}} = 1.4$ and $d_{\mathrm{Alc}} = 0.4$. Here, parameter μ is the parameter that can vary the difference between the system model and real system continuously. When the value of μ is 1, the properties of the constant model disappear, and the model becomes perfectly proportional to $1/n$. For the turbulent electron thermal diffusivity in Eq. (2.44), the gyro-Bohm model [42, Sections 4.15 and 4.16],

$$\chi_{\mathrm{e}}^{\mathrm{TB}} = C_{\mathrm{gB}} \frac{T_{\mathrm{e}}}{eB} \frac{\rho_{\mathrm{i}}}{a} \times c_{\mathrm{e}}, \tag{5.5}$$

is employed in ASTI's system model, whereas in the virtual plasma, the model with a $1/n$ dependent term,

$$\chi_{\mathrm{e}}^{\mathrm{TB}} = \frac{1}{2} \left(C_{\mathrm{gB}} \frac{T_{\mathrm{e}}}{eB} \frac{\rho_{\mathrm{i}}}{a} + C_{\mathrm{Alc}} \frac{1}{n} \right), \tag{5.6}$$

is employed, where $C_{\mathrm{gB}} = 1.5$ and $C_{\mathrm{Alc}} = 1.61$. Here, B, ρ_{i}, and a are the magnetic field strength, ion Larmor radius, and plasma minor radius, respectively. The values of d_{const}, d_{Alc}, C_{gB}, and C_{Alc} are set to reasonable values from previous studies [21,79]. Parameter d in Eq. (5.3) and parameter c_{e} in Eq. (5.5), which are introduced for data assimilation and are optimized by the y-filter to enhance the prediction capability of the plasma response. The time step of TASK3D as the virtual plasma is set to 10^{-3} seconds, while that of TASK3D as the system model in ASTI is set to 10^{-2} seconds.

5.2 Control results

In this section, we discuss the control results by ASTI and the dependence of control accuracy on the noise parameters in the DACS framework and on the discrepancy between the system model and actual system behavior.

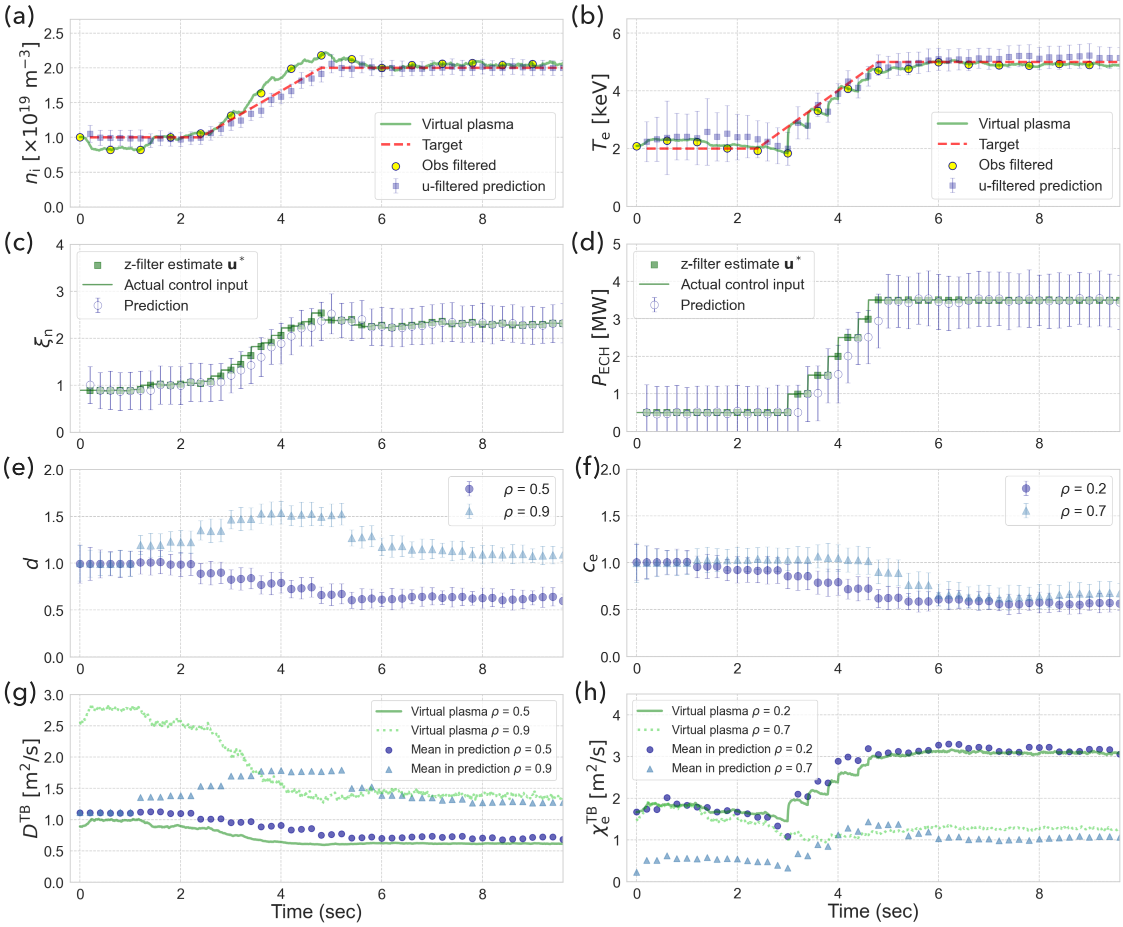


Figure 5.2: Results of a numerical experiment to control virtual plasma. (a) and (b), Control result of density and electron temperature at plasma center, respectively. Error bars represent one standard deviation of the state distribution. (c) and (d), Estimation results for control inputs ξ_n and P_{ECH}, respectively. (e) and (f), Time variation of state variables for transport model. The distributions (mean and one standard deviation) used in prediction step are shown. (g) and (h), Time variation of diffusivities D^{TB} and $\chi_{\mathrm{e}}^{\mathrm{TB}}$ calculated in prediction step (scattered points) and those in the virtual plasma (solid and dashed lines).

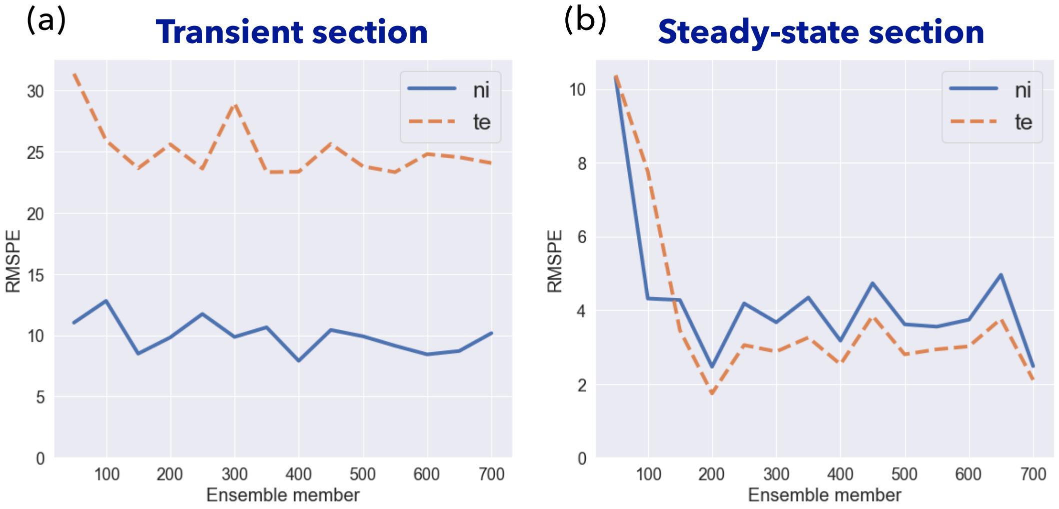


Figure 5.3: RMSPEs between the observation and target state (a) in the transient section (2.4-4.8 s) and (b) in the steady-state section (4.8 s-).

5.2.1 Control results of virtual plasma

Figure 5.2 shows the results for the numerical experiment where r_z=0.6 and r_y=0.8 for the virtual plasma with μ=0.5. Figures 5.2(a) and (b) show that the plasma state follows the target state with high accuracy in the steady-state section (4.8 s-) while there is a slight difference in the transient section (2.4-4.8 s). These differences between the observation and target state arise from differences between the employed system model and the real system (virtual plasma). The y-filter optimizes the model parameters in $\tilde{\mathbf{x}}$ such that the difference between the predicted response (i.e., the u-filtered distribution) and the actual response (i.e., the observation) becomes small. Figures 5.2(e) and (f) show the model parameters d and c_{e} adjusted to bridge the gap between the system model and real system. The model parameters are primarily optimized in the transient section, where the difference between the predicted and actual responses is expected to increase, and the diffusivities D^{TB} and $\chi_{\mathrm{e}}^{\mathrm{TB}}$ calculated in ASTI and the virtual plasma are close from approximately 4 s (Figs. 5.2(g) and (h)).

The number of ensemble members was set to 420 for the above control experiment. Figure 5.3 shows the root mean square percentage error (RMSPE) between the observation and target state for varying the number of ensemble members. We confirmed that the control performance saturates at approximately 200 members form Fig. 5.3(b) (the change in the RMSPE in the steady-state section). The valid number of ensemble members cannot be determined from the change in the transient section because of the large error in the control itself.

Figure 5.4 shows how the ensemble approximating the state distribution at time t_i = 4.2 is affected at each step. In the algorithm constructed in Section 4.1.3, the ensemble at an observation time undergoes three assimilations from the predicted distribution

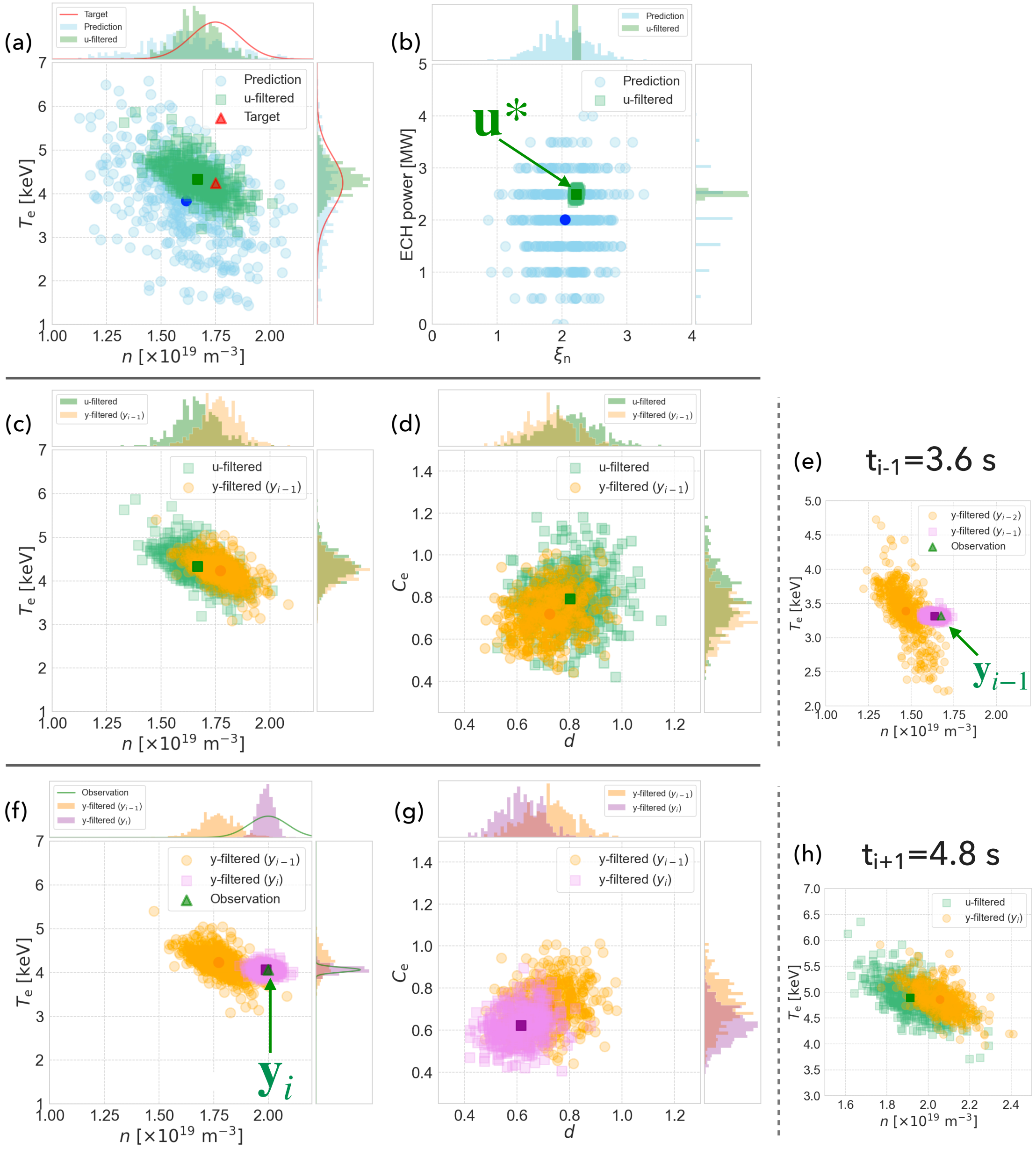


Figure 5.4: Changes in the ensemble members approximating the state distribution at time $t_i = 4.2$ in the control experiment.

$p(\mathbf{x}_{(i,0)} \mid \mathbf{y}_{0:i-2}, \mathbf{u}^*_{(0,1):(i-1,2)})$:

$$\begin{aligned} p &\quad (\mathbf{x}_{(i,0)} \mid \mathbf{y}_{0:i-2}, \mathbf{u}^*_{(0,1):(i-1,2)}) \\ \to &\quad p(\mathbf{x}_{(i,0)} \mid \mathbf{y}_{0:i-2}, \mathbf{u}^*_{(0,1):(i,0)}) \qquad (5.7) \\ \to &\quad p(\mathbf{x}_{(i,0)} \mid \mathbf{y}_{0:i-1}, \mathbf{u}^*_{(0,1):(i,0)}) \qquad (5.8) \\ \to &\quad p(\mathbf{x}_{(i,0)} \mid \mathbf{y}_{0:i}, \mathbf{u}^*_{(0,1):(i,0)}) \qquad (5.9) \end{aligned}$$

Figures 5.4(a) and (b) show the first assimilation, Eq. 5.7, where the predicted distribution (”Prediction”) absorbs the optimal control input and becomes the u-filtered distribution (”u-filtered”) by the u-filter. Figure 5.4(a) is a scatter plot for the electron temperature and density at the plasma center, and (b) is that for the control input variables. We can see that the u-filtered distribution gets close to the target (”Target”) by the z-filter and u-filter. In addition, the variation of the ensemble members approximating the u-filtered distribution is reduced to a practically acceptable level because we employ a small variance of the control-input noise.

Figures 5.4(c) and (d) show the second assimilation, where the u-filtered distribution (”u-filtered”) is optimized to the y-filtered distribution (”y-filtered ($\mathbf{y}_{i-1}$)”) by the observation at $t_{i-1} = 3.6$ s ($\mathbf{y}_{i-1}$). Figure 5.4(c) is a scatter plot for the electron temperature and density, and (d) is that for the transport model parameters (c_e at $\rho = 0.2$ and d at $\rho = 0.5$). In addition, Fig. 5.4(e) shows the changes in the ensemble members at $t_{i-1} = 3.6$ s by the same y-filter. The y-filter based on $\mathbf{y}_{i-1}$ optimizes the u-filtered ensemble at $t_i = 4.2$ s through the y-filtered ensemble at $t_{i-1} = 3.6$ s (e).

The y-filtered ensemble at $t_i = 4.2$ s is also used to optimize the u-filtered ensemble at $t_{i+1} = 4.8$ s by the y-filter based on the observation at $\mathbf{y}_i$. Figures 5.4(f) and (g) show the third assimilation; (f) shows the ensemble for the electron temperature and density, and (g) shows the transport model parameters. The y-filter reflects the information of $\mathbf{y}_i$ to the u-filtered ensemble at $t_{i+1} = 4.8$ s through the the y-filtered ensemble at $t_i = 4.2$ s. Figure 5.4(h) shows the changes in the ensemble members at t_{i+1} by the y-filter.

The noise setting where the control-input noise is sufficiently small makes the correlation in the u-filtered distribution between $\mathbf{u}$ and $\mathbf{x}$ quite small (about 0). Therefore, the distribution of $\mathbf{u}$ is little affected by the y-filter. However, when a larger control-input noise is required for some reason (e.g., when we want to employ the PF), the distribution $p(\mathbf{u}_{(i+1,0)}|\mathbf{y}_{0:i-1}, \mathbf{u}^*_{(0,1):(i+1,0)})$ can be affected by the y-filter based on $\mathbf{y}_i$, and the mean of $\mathbf{u}_{(i+1,0)}$ can be shifted from $\mathbf{u}^*_{(i+1,0)}$. One possible countermeasure for the problem is to generate ensemble members for $\mathbf{u}$ part from a new Gaussian each time before the prediction step to eliminate the correlation between $\mathbf{u}$ and $\tilde{\mathbf{x}}$.

5.2.2 Dependence of control performance on hyper parameters

The control algorithm constructed in the DACS framework involves the following hyper parameters: number of ensemble members, initial state distribution, system noise, target noise, control-input noise, and observation noise. Here, we investigate the dependence of control accuracy on the hyper parameters, focusing on target noise and observation noise. We introduced parameters r_z in Eq. (4.24) and r_y in Eq. (4.25) in Section 4.2 to determine the target-state noise in the z-filter and the observation noise

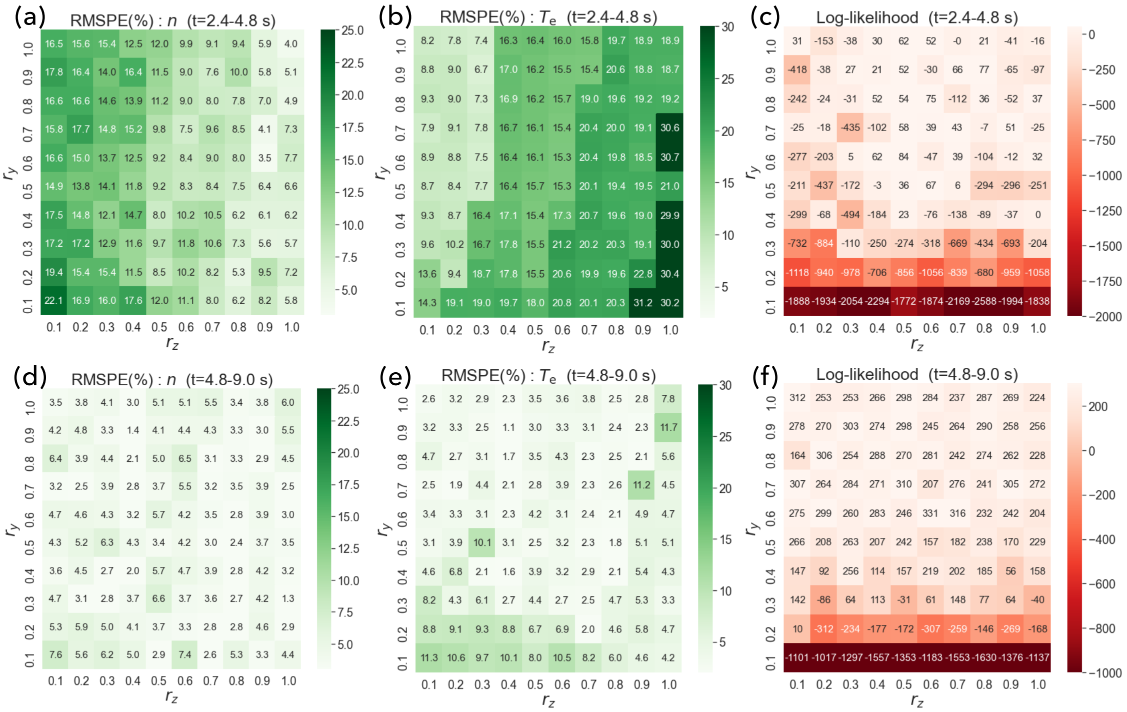


Figure 5.5: Results of numerical experiments with various r_z and r_y values: RMSPEs between the observation and target state for density (a) and electron temperature (b) in the transient section (2.4-4.8 s). (c) Log-likelihoods of observation for u-filtered distribution in the transient section. (d)-(f) as (a)-(c), respectively, but in the steady-state section (4.8-9.0 s).

in the y-filter, respectively. Parameter r_z can be used to adjust how close the predicted distribution gets to the target state at the control-input estimate (z-filter). For a large r_z value, the z-filter's force to attract the predicted distribution to the target state is weakened and the change in the u-filtered distribution becomes small. In contrast, parameter r_y controls the impact of the observation in the y-filter. A larger r_y value results in a smaller effect of the observed information to the state distribution.

Figure 5.5 shows the root mean square percentage error (RMSPE) between the observation and target state, as well as the log-likelihood of the observation for the corresponding u-filtered distribution in the transient section (2.4-4.8 s) and steady-state section (4.8-9.0 s). The RMSPE value represents pure control accuracy, and the log-likelihood represents the difference between the predicted and actual responses.

First, we focus on the r_z-dependence of the RMSPE and the log-likelihood. In the transient section, the r_z-dependence of the RMSPEs is strong, whereas in the steady-state section, the RMSPEs are low (approximately 3%) over a wide parameter range. For n in the transient section (Fig. 5.5(a)), the RMSPE is lower for large r_z, while it is higher for T_e (Fig. 5.5(b)). The log-likelihood is relatively low in the small r_z region in Fig 5.5(c). We can see a similar tendency in the RMSPE of n (Fig. 5.5(a)). These results indicate that the prediction of n is more strongly affected by model imperfection (uncertainties) than that of T_e, and optimization of the particle transport model in ASTI do not maintain pace with changes in virtual plasma. This can be also observed by comparing the Figs. 5.2(a) and (b). The difference between the u-filtered prediction and filter estimate using the corresponding observation ("Obs filtered") in Fig. 5.2(a) (n) is greater than that in Fig. 5.2(b) (T_e). Thus, the RMSPE of n decreases when the large r_z slows down the state change rate. For T_e, where the prediction uncertainty is relatively small, a greater r_z value simply increases the RMSPE as the state change rate slows relative to the target-state time series. Of course, the RMSPE of n also increases due to this effect when r_z further increases, although this effect is more pronounced for T_e because the target state of T_e has a wider change range, and the control input P_{ECH} takes discrete values. It is considered that the change in the RMSPE of T_e relative to r_z occurs within the prediction by the system model, while the change in the RMSPE of n occurs outside the prediction. In situations where the system model is unsophisticated or its ability to approximate the real system is unknown, it is safe to select a large r_z value ($\gtrsim 0.5$).

On the other hand, a small r_y value reduces the log-likelihood significantly, as shown in Figs. 5.5(c) and (f). This occurs because the variance of the state distribution becomes excessively small for a small r_y value, and the state distribution cannot maintain sufficient uncertainty for sequential adaptation and robust control estimation. Note that it is also safe to select a reasonably large r_y value ($\gtrsim 0.5$).

5.2.3 Limitation of adaptation

Note that the control systems based on the DACS framework have certain limitations in terms of the actual systems that can be approximated even with a system model optimized via observation. In this section, we investigate the dependence of ASTI's control performance on the discrepancy between the employed system model and actual system behavior. Figure 5.6 shows the results of numerical experiments conducted with various μ introduced in Section 5.1.2 and r_z values. Here, the parameter r_y is fixed at 0.8. Figure 5.6 shows the root mean square percentage error (RMSPE) between the observation and target state, as well as the log-likelihood of the observation for the corresponding u-filtered distribution in the transient section (2.4-4.8 s) and steady-state section (4.8-9.0 s). The RMSPE values for density and electron temperature increase significantly near $\mu = 1$ regardless of the transient section (Figs. 5.6(a) and (d)) or steady-state section (Figs. 5.6(b) and (e)). A similar tendency is observed in terms of the log-likelihood (Figs. 5.6(c) and 5.6(f)), i.e., the prediction accuracy drops near μ=1, where the system model behavior deviates significantly from the virtual plasma behavior. In the vicinity of $\mu = 1$, the real system changes on a time scale faster than the observation cycle Δt_y. Therefore, even a system model whose parameters are optimized sequentially based on observation information cannot approximate the real system adequately, and the data assimilation system fails to estimate the control inputs.

Next, we focus on the r_z-dependence of the RMSPE and the log-likelihood. The r_z-dependence found in Fig. 5.6 is the same as the dependence mentioned in the previous section. In the transient section, the r_z-dependence of the RMSPEs is strong, whereas in the steady-state section, the RMSPEs are low (approximately 5%) over a wide r_z

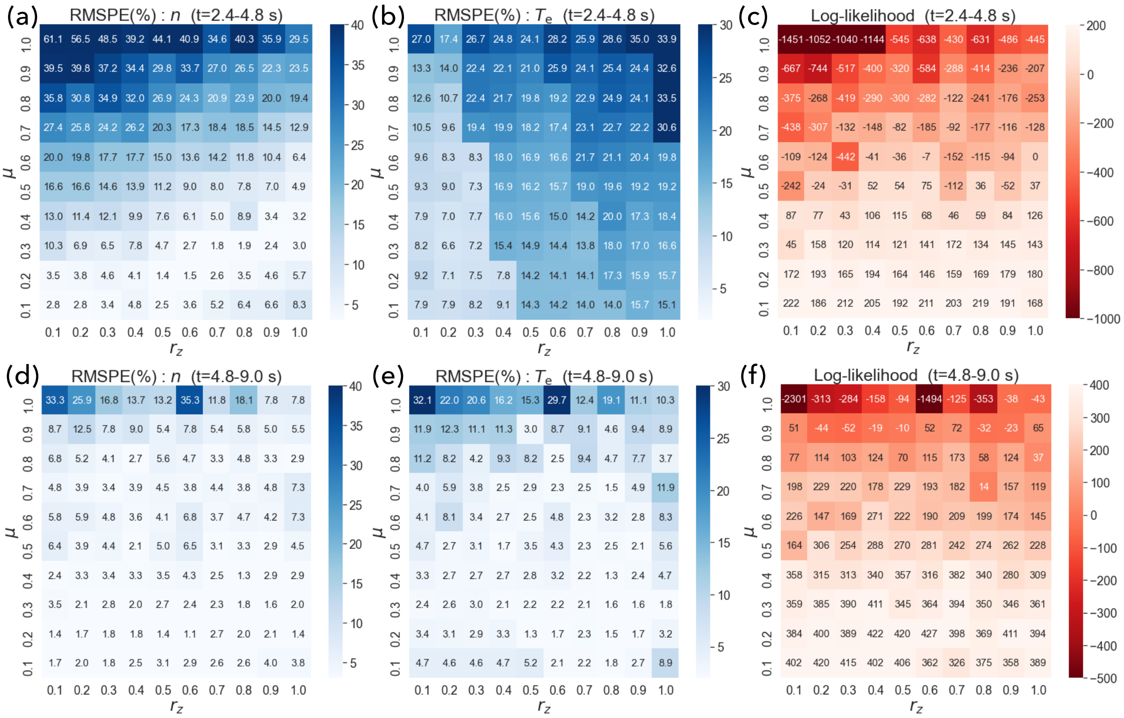


Figure 5.6: Results of numerical experiments with various r_z and μ values: RMSPEs between the observation and target state for density (a) and electron temperature (b) in the transient section (2.4-4.8 s). (c) Log-likelihoods of observation for u-filtered distribution in the transient section. (d)-(f) as (a)-(c), respectively, but in the steady-state section (4.8-9.0 s).

range. For n in the transient section (Fig. 5.6(a)), the RMSPE is lower for large r_z, while it is higher for T_e (Fig. 5.6(b)). The log-likelihood is relatively low in the small r_z region in Fig 5.6(c).

The range of systems that can be stably controlled is primarily determined by the assumed system model, the observation cycle Δt_y, and the prediction horizon h. Thus, the range of target systems can be expanded by employing more expressive models in the system model or by constructing a control algorithm for smaller Δt_y and h. In addition, the target noise parameter r_z can reduce the difference between the predicted and actual responses. As shown in Fig. 5.6(a) (or 5.6(c)), for large r_z, a system with a large μ value can be controlled with high accuracy by reducing the rate of state change. We can construct a more robust control system by developing methods to adjust the hyper parameters for the given situation, e.g., the difference between the predicted and actual responses.

5.3 Summary

We have established a data assimilation framework to integrate system model updates and estimation of control inputs. The DACS framework comprises a prediction step and three filtering steps and provides adaptive model predictive control algorithms, i.e., effective control methods even when the system model involves large uncertainties. Since the DACS framework does not require additional prediction steps for control processes, we can construct a control algorithm without overlapping the prediction intervals. Thus, we can apply this framework to a large system in which iterative model prediction is prohibitive due to computational burden. In other words we can employ simulation models with non-negligible computational costs for the control time scale as the system model.

In this chapter, we presented an example application of the DACS framework with a focus on fusion plasma control. In the numerical experiments to control the virtual LHD plasma using ASTI based on the DACS framework, we have seen that the behavior of the virtual plasma followed the time series of target state with high accuracy. On the other hand, we have observed limitations of adaptation, which lead to a situation where the gap between the system model and the real system cannot be completely bridged. Through this application example, we have demonstrated the effectiveness and the performance characteristics of the DACS framework. This application is only one example of the DACS framework's applicability to a much wider range of model-based control problems.

Chapter 6

Conclusions and Future Perspectives

The purpose of this study is to develop a numerical system that can analyze and control the behavior of fusion plasma with high accuracy applying data assimilation. Through this thesis, we have presented the concepts and details of the data assimilation system for fusion plasmas, ASTI, and demonstrated the effectiveness and validity of ASTI. ASTI can enhance the predictive and analytical capabilities of the system model (integrated simulation code) by connecting numerical space and real space.

ASTI as an analysis system

In Chapter 2 and 3, we have discussed the analytical applications of ASTI based on the sequential Bayesian filter. The EnKF and EnKS were employed as the data assimilation methods. First, we have assimilated the density, electron temperature, and ion temperature data of an NBI-heated plasma in LHD and estimated the thermal turbulent diffusivities. We have confirmed that ASTI can estimate reasonable model parameters that satisfy the dynamic constraints by the system model and reproduce the experimental data. Data assimilation has been successfully introduced into integrated transport simulation of fusion plasmas.

Next, we have applied ASTI to the 12 experimental time-series data sets of NBI-heated plasmas in LHD. In this case, we solved particle and heat transport by TASK3D and considered the uncertainties included in the density, temperature, transport models, NBI heating model, and particle source model. In addition, we employed the reduced NBI heating model to speed up the prediction calculation expecting that the uncertainty increased by the reduction is compensated by data assimilation adjusting the model parameters. The predicted radial profiles of density, electron temperature, and ion temperature by ASTI (EnKF) have agreed well with the observed profiles for all the 12 discharges, and the errors between the prediction and observation have been significantly reduced from the TASK3D simulation (without data assimilation). Using the EnKS, we have obtained the radial profiles and temporal changes of the model parameters that can reproduce the experimental time-series data. It has been confirmed that the TASK3D simulations using the smoothed estimates of the model parameters reproduce the experimental time-series data with high accuracy. These results indicate the effectiveness and validity of ASTI for accurate prediction and analysis of particle and heat transport in fusion plasmas. Furthermore, it was found that the calculation by ASTI can be performed in about four minutes for a second in actual plasma, while the previous case takes more than 10 hours. A high-speed and accurate analysis system has been realized by using the data assimilation.

The estimates of the transport model parameters have characteristic structures and suggest missing elements in the transport model. Such information is useful for improving the transport model. In order to build a valid model for many other discharges, it is necessary to extract properties common to a large number of discharges and combine them into a model. These are issues that we would like to address in the future. We are now developing a transport model built by regressing the estimated model parameters for a large number of discharges. At the same time, we are also developing a method for multi-discharge data assimilation, which directly extracts common information from a large number of discharges and creates a high-performance transport model.

ASTI as a control system

In Chapter 4 and 5, we have discussed the control application of ASTI. We have proposed a data assimilation framework, referred to as DACS, to integrate system model updates and estimation of control inputs and investigated the control performance. This framework is the foundation for ASTI as a control system. The DACS framework provides effective control methods even when the system model involves large uncertainties. With this framework, we can construct a control algorithm without overlapping the prediction intervals. Thus, we can employ simulation models with non-negligible computational costs for the control time scale as the system model.

To demonstrate the effectiveness of DACS and reveal the characteristics of the control performance related to the hyper parameters and the discrepancies between the system model and the real system, we have performed the numerical experiments to control the virtual LHD plasma. As a result of the numerical experiments, we have confirmed that the behavior of the virtual plasma followed the time series of target state with high accuracy. Moreover, it has been found that the control is more stable when the target-state noise and observation noise are sufficiently large. When the gap between the system model and the real system is large, it is important to increase the flexibility of the system model by taking larger variances for the target-state and observation noise. Through this application example, we have demonstrated the effectiveness and the performance characteristics of the DACS framework.

Note that this application is only one example of the DACS framework's applicability to a much wider range of model-based control problems. The DACS framework can handle various types of sate variables, observed data, controlled variables, and control constraints in an integrated manner. We believe that the DACS framework provides a foundation to build flexible and powerful control systems for a wide variety of complex control problems including fusion plasma control

Particularly important issue to be addressed in the future is the development of methods to mitigate loss of control accuracy due to limitations relative to the adaptability of the system model. For system models with large uncertainties, it is necessary to prepare highly expressive models or adjust the observation cycle in consideration of the intensity of changes in the target state, amount of observation information and computational cost. In addition, we should introduce state variables that express the state stability and safety for an actual control situation. Fusion plasma, in particular, has many unstable states and may disappear if the control is improper. Data-driven researches are being conducted to sustain or enhance fusion plasma performance [80–83], and cooperation with such approaches is important for actual control.

Data assimilation connects real space and numerical space and enables us to construct a comprehensive system in which the simulation code and the actual fusion plasma are integrated. ASTI is both a powerful analysis system and a powerful control system and can be flexibly applied to a variety of analysis and control problems. We believe that ASTI (or data assimilation) will play an important role in both analysis and control in the fusion field.

Acknowledgement

The author was able to complete this thesis with the help of many wonderful people. He is especially grateful to his supervisor, Professor Sadayoshi Murakami, who gave him continuous guidance, many opportunities to gain valuable experience, and very enjoyable six years of research. He is also very grateful to Professor Masayuki Yokoyama and Professor Genta Ueno for their valuable advice, kind help, and fresh perspectives. He is indebted to Honorary Professor Atsushi Fukuyama and Professor Mitsuru Honda for their kind support in the laboratory.

The author would like to thank the past and present members of the laboratory, especially Mr. Maeta, Dr. Yamamoto, Mr. Shinogi, Mr. Kotera, Mr. Ota, Mr. Okada, Mr. Suzuki, Mr. Tahara, Mr. Baigetsu, Mr. Anzai, and Mr. Matsuzawa. They made his life in the laboratory very enjoyable, and he learned a lot through discussions with them. Moreover, he also wishes to thank his friends, Mr. Sakami and Mr. Shimura, for meaningful and enjoyable discussions.

This work was supported by the NIFS Collaborative Research Program, ISM Co-operative Research Program, the Grant-in-Aid for scientific reserach from JSPS, and QST Research Collaboration for Fusion DEMO. This work was conducted using the JFRS-1 supercomputer system at IFERC-CSC and the Plasma Simulator (FUJITSU FX100 and NEC SX-Aurora TSUBASA) of NIFS.

Finally, he would like to express his gratitude to his parents and grandparents for their continuous and warm support all his life.

Publication List

Section 2.3

Application of the Ensemble Kalman Smoother to Turbulent Transport Analysis in LHD Plasma,
Yuya Morishita, Sadayoshi Murakami, Masayuki Yokoyama, Genta Ueno,
Plasma and Fusion Research **16**, 2403016 (2021).
doi:10.1585/pfr.16.2403016.

Section 2.4

Development of Rapid Simulation Code for NBI Heating Analysis in LHD,
Yuya Morishita, Sadayoshi Murakami, Masayuki Yokoyama, Ryosuke Seki, Hideo Nuga, Masaki Osakabe, Genta Ueno,
Journal of Fusion Energy **41**, 1-8 (2022).
doi:10.1007/s10894-021-00313-5.

Section 3.1

Data assimilation system based on integrated transport simulation of Large Helical Device plasma,
Yuya Morishita, Sadayoshi Murakami, Masayuki Yokoyama, Genta Ueno,
Nuclear Fusion **60**, 056001 (2020).
doi:10.1088/1741-4326/ab7596.

Section 3.2

ASTI: Data assimilation system for particle and heat transport in toroidal plasmas,
Yuya Morishita, Sadayoshi Murakami, Masayuki Yokoyama, Genta Ueno,
Computer Physics Communications **274**, 108287 (2022).
doi:10.1016/j.cpc.2022.108287.

Bibliography

[1] Y. Takeiri, T. Morisaki, M. Osakabe, M. Yokoyama, S. Sakakibara, H. Takahashi, Y. Nakamura, T. Oishi, G. Motojima, S. Murakami, et al., Extension of the operational regime of the LHD towards a deuterium experiment, Nuclear Fusion 57 (10) (2017) 102023. doi:https://doi.org/10.1088/1741-4326/aa7fc2.

[2] M. Osakabe, H. Takahashi, H. Yamada, K. Tanaka, T. Kobayashi, K. Ida, S. Ohdachi, J. Varela, K. Ogawa, M. Kobayashi, et al., Recent results from deuterium experiments on the large helical device and their contribution to fusion reactor development, Nuclear Fusion 62 (4) (2022) 042019. doi:https://doi.org/10.1088/1741-4326/ac3cda.

[3] URL https://www-lhd.nifs.ac.jp.

[4] A. Fasoli, S. Brunner, W. Cooper, J. Graves, P. Ricci, O. Sauter, L. Villard, Computational challenges in magnetic-confinement fusion physics, Nature Physics 12 (5) (2016) 411–423. doi:10.1038/nphys3744.

[5] F. Imbeaux, S. Pinches, J. Lister, Y. Buravand, T. Casper, B. Duval, B. Guillerminet, M. Hosokawa, W. Houlberg, P. Huynh, et al., Design and first applications of the iter integrated modelling & analysis suite, Nuclear Fusion 55 (12) (2015) 123006. doi:https://doi.org/10.1088/0029-5515/55/12/123006.

[6] M. Honda, S. Ide, T. Takizuka, N. Hayashi, M. Yoshida, M. Yagi, T. Fujita, Development of the transport-code framework for self-consistent predictions of rotation and the radial electric field, Nuclear Fusion 53 (7) (2013) 073050. doi:https://doi.org/10.1088/0029-5515/53/7/073050.

[7] J. M. Park, J. Ferron, C. T. Holcomb, R. J. Buttery, W. M. Solomon, D. Batchelor, W. Elwasif, D. Green, K. Kim, O. Meneghini, et al., Integrated modeling of high βn steady state scenario on diii-d, Physics of Plasmas 25 (1) (2018) 012506. doi:https://doi.org/10.1063/1.5013021.

[8] F. Felici, J. Citrin, A. Teplukhina, J. Redondo, C. Bourdelle, F. Imbeaux, O. Sauter, J. Contributors, E. M. Team, et al., Real-time-capable prediction of temperature and density profiles in a tokamak using raptor and a first-principle-based transport model, Nuclear Fusion 58 (9) (2018) 096006. doi:https://doi.org/10.1088/1741-4326/aac8f0.

[9] J. Artaud, F. Imbeaux, J. Garcia, G. Giruzzi, T. Aniel, V. Basiuk, A. Bécoulet, C. Bourdelle, Y. Buravand, J. Decker, et al., Metis: a fast integrated tokamak modelling tool for scenario design, Nuclear Fusion 58 (10) (2018) 105001.

[10] J. Winter, Wall conditioning in fusion devices and its influence on plasma performance, Plasma Physics and Controlled Fusion 38 (9) (1996) 1503.

[11] S. Smolentsev, G. A. Spagnuolo, A. Serikov, J. J. Rasmussen, A. H. Nielsen, V. Naulin, J. Marian, M. Coleman, L. Malerba, On the role of integrated computer modelling in fusion technology, Fusion Engineering and Design 157 (2020) 111671. doi:https://doi.org/10.1016/j.fusengdes.2020.111671.

[12] A. Gettelman, A. J. Geer, R. M. Forbes, G. R. Carmichael, G. Feingold, D. J. Posselt, G. L. Stephens, S. C. van den Heever, A. C. Varble, P. Zuidema, The future of earth system prediction: Advances in model-data fusion, Science Advances 8 (14) (2022) eabn3488. doi:10.1126/sciadv.abn3488.

[13] P. M. Gregg, Y. Zhan, F. Amelung, D. Geist, P. Mothes, S. Koric, Z. Yunjun, Forecasting mechanical failure and the 26 june 2018 eruption of sierra negra volcano, galápagos, ecuador, Science Advances 8 (22) (2022) eabm4261. doi:10.1126/sciadv.abm4261.

[14] X. Hou, S. Gao, Q. Li, Y. Kang, N. Chen, K. Chen, J. Rao, J. S. Ellenberg, J. A. Patz, Intracounty modeling of covid-19 infection with human mobility: Assessing spatial heterogeneity with business traffic, age, and race, Proceedings of the National Academy of Sciences 118 (24) (2021) e2020524118. doi:10.1073/pnas.2020524118.

[15] T. Miyoshi, M. Kunii, J. Ruiz, G.-Y. Lien, S. Satoh, T. Ushio, K. Bessho, H. Seko, H. Tomita, Y. Ishikawa, “big data assimilation” revolutionizing severe weather prediction, Bulletin of the American Meteorological Society 97 (8) (2016) 1347–1354. doi:https://doi.org/10.1175/BAMS-D-15-00144.1.

[16] A. Carrassi, M. Bocquet, L. Bertino, G. Evensen, Data assimilation in the geosciences: An overview of methods, issues, and perspectives, Wiley Interdisciplinary Reviews: Climate Change 9 (5) (2018) e535. doi:https://doi.org/10.1002/wcc.535.

[17] M. Hoshiba, S. Aoki, Numerical shake prediction for earthquake early warning: Data assimilation, real-time shake mapping, and simulation of wave propagation, Bulletin of the Seismological Society of America 105 (3) (2015) 1324–1338.

[18] S. Nakano, G. Ueno, Y. Ebihara, M.-C. Fok, S. Ohtani, P. Brandt, D. Mitchell, K. Keika, T. Higuchi, A method for estimating the ring current structure and the electric potential distribution using energetic neutral atom data assimilation, Journal of Geophysical Research: Space Physics 113 (A5). doi:https://doi.org/10.1029/2006JA011853.

[19] S. Murakami, H. Yamaguchi, A. Sakai, A. Wakasa, A. Fukuyama, K. Nagaoka, H. Takahashi, H. Nakano, M. Osakabe, K. Ida, et al., Integrated transport simulations of high ion temperature plasmas of LHD, Plasma Physics and Controlled Fusion 57 (5) (2015) 054009. doi:https://doi.org/10.1088/0741-3335/57/5/054009.

[20] M. Yokoyama, R. Seki, C. Suzuki, M. Sato, M. Emoto, S. Murakami, M. Osakabe, T. I. Tsujimura, Y. Yoshimura, T. Ido, et al., Extended capability of the integrated transport analysis suite, task3d-a, for LHD experiment, Nuclear Fusion 57 (12) (2017) 126016. doi:https://doi.org/10.1088/1741-4326/aa800a.

[21] A. Sakai, S. Murakami, H. Yamaguchi, A. Wakasa, A. Fukuyama, K. Nagaoka, H. Takahashi, H. Nakano, M. Osakabe, et al., Integrated particle transport simulation of nbi plasmas in LHD, Plasma and Fusion Research 10 (2015) 3403048–3403048. doi:https://doi.org/10.1585/pfr.10.3403048.

[22] M. Yokoyama, A. Wakasa, R. Seki, M. Sato, S. Murakami, C. Suzuki, Y. Nakamura, A. Fukuyama, L. E. Group, et al., Development of integrated transport code, task3d, and its applications to LHD experiment, Plasma and Fusion Research 7 (2012) 2403011–2403011. doi:10.1585/pfr.7.2403011.

[23] S. Nakano, S. Miura, P. Victor, A. Torisaka, T. Miyashita, Data assimilation using particle filter for real-time identification of organ properties, The Journal of Engineering 2019 (14) (2019) 517–521. doi:https://doi.org/10.1049/joe.2018.9410.

[24] H. Kato, A. Yoshizawa, G. Ueno, S. Obayashi, A data assimilation methodology for reconstructing turbulent flows around aircraft, Journal of Computational Physics 283 (2015) 559–581. doi:https://doi.org/10.1016/j.jcp.2014.12.013.

[25] R. E. Kalman, A New Approach to Linear Filtering and Prediction Problems, Journal of Basic Engineering 82 (1) (1960) 35–45. doi:10.1115/1.3662552.

[26] Z. Chen, et al., Bayesian filtering: From kalman filters to particle filters, and beyond, Statistics 182 (1) (2003) 1–69. doi:10.1080/02331880309257.

[27] G. Kitagawa, Monte carlo filter and smoother for non-gaussian nonlinear state space models, Journal of computational and graphical statistics 5 (1) (1996) 1–25. doi:https://doi.org/10.1080/10618600.1996.10474692.

[28] O. Talagrand, P. Courtier, Variational assimilation of meteorological observations with the adjoint vorticity equation. i: Theory, Quarterly Journal of the Royal Meteorological Society 113 (478) (1987) 1311–1328.

[29] G. Evensen, The ensemble kalman filter: Theoretical formulation and practical implementation, Ocean dynamics 53 (4) (2003) 343–367. doi:https://doi.org/10.1007/s10236-003-0036-9.

[30] G. Ueno, T. Higuchi, T. Kagimoto, N. Hirose, Application of the ensemble kalman filter and smoother to a coupled atmosphere-ocean model, Sola 3 (2007) 5–8. doi:https://doi.org/10.2151/sola.2007-002.

[31] K. Law, A. Stuart, K. Zygalakis, Data assimilation, Cham, Switzerland: Springer 214 (2015) 52.

[32] Y. Morishita, S. Murakami, M. Yokoyama, G. Ueno, Application of the ensemble kalman smoother to turbulent transport analysis in LHD plasma, Plasma and Fusion Research 16 (2021) 2403016–2403016. doi:https://doi.org/10.1585/pfr.16.2403016.

[33] A. Fukuyama, S. Murakami, M. Honda, Y. Izumi, M. Yagi, N. Nakajima, Y. Nakamura, T. Ozeki, Advanced transport modeling of toroidal plasmas with transport barriers, proceeding of 20th Fusion Energy Conference (Villamoura, Portugal), IAEA-CSP-250CD/TH/P2-3 (2004).

[34] M. Honda, A. Fukuyama, Comparison of turbulent transport models of l- and h-mode plasmas, Nuclear Fusion 46 (5) (2006) 580. doi:10.1088/0029-5515/46/5/009.

[35] F. L. Hinton, R. D. Hazeltine, Theory of plasma transport in toroidal confinement systems, Rev. Mod. Phys. 48 (1976) 239–308. doi:10.1103/RevModPhys.48.239.

[36] S. Hirshman, P. Merkel, et al., Three-dimensional free boundary calculations using a spectral green's function method, Computer Physics Communications 43 (1) (1986) 143–155. doi:https://doi.org/10.1016/0010-4655(86)90058-5.

[37] A. Wakasa, S. Murakami, A. Fukuyama, C. Beidler, H. Maaßberg, M. Yokoyama, M. Sato, Development of the neoclassical transport module for the integrated simulation code in helical plasmas, Contributions to Plasma Physics 50 (6-7) (2010) 582–585. doi:https://doi.org/10.1002/ctpp.200900049.

[38] A. Wakasa, S. Murakami, C. BEIDLER, S.-i. OIKAWA, M. ITAGAKI, Monte carlo simulations study of neoclassical transport in inward shifted LHD configurations, J. Plasma Fusion Res. Ser 4 (2001) 408–2001.

[39] C. D. Beidler, W. D. D'haeseleer, A general solution of the ripple-averaged kinetic equation (gsrake), Plasma Physics and Controlled Fusion 37 (4) (1995) 463. doi:10.1088/0741-3335/37/4/007.

[40] M. Yokoyama, A. Wakasa, S. Murakami, K. Watanabe, S. Satake, S. Nishimura, H. Sugama, N. Nakajima, H. Funaba, Y. Nakamura, Role of neoclassical transport and radial electric field in LHD plasmas, Fusion Science and Technology 58 (1) (2010) 269–276. doi:https://doi.org/10.13182/FST10-A10813.

[41] A. Wakasa, A. Fukuyama, S. Murakami, N. Takeda, M. Yokoyama, R. Seki, M. Osakabe, H. Takahashi, I. Yamada, K. Nagaoka, K. Ida, M. Yoshinuma, K. Tanaka, et al., Transport simulation in LHD plasmas using gyro-bohm transport models including the effect of temperature gradient, proc. 39th EPS Conf. and 16th Int. Conf. Plasma Physics (Stockholm) P2.028 (2012).

[42] J. Wesson, Tokamaks (4th edition), Oxford University Press, Oxford, 2011.

[43] M. Hughes, D. Post, A monte carlo algorithm for calculating neutral gas transport in cylindrical plasmas, Journal of Computational Physics 28 (1) (1978) 43–55. doi:https://doi.org/10.1016/0021-9991(78)90045-1.

[44] K. Tani, M. Azumi, H. Kishimoto, S. Tamura, Effect of toroidal field ripple on fast ion behavior in a tokamak, Journal of the Physical Society of Japan 50 (5) (1981) 1726–1737. doi:https://doi.org/10.1143/JPSJ.50.1726.

[45] R. J. Goldston, D. McCune, H. Towner, S. Davis, R. Hawryluk, G. Schmidt, New techniques for calculating heat and particle source rates due to neutral beam injection in axisymmetric tokamaks, Journal of computational physics 43 (1) (1981) 61–78. doi:https://doi.org/10.1016/0021-9991(81)90111-X.

[46] M. Weiland, R. Bilato, R. Dux, B. Geiger, A. Lebschy, F. Felici, R. Fischer, D. Rittich, M. Van Zeeland, E. M. Team, et al., Rabbit: Real-time simulation of the nbi fast-ion distribution, Nuclear Fusion 58 (8) (2018) 082032. doi:https://doi.org/10.1088/1741-4326/aabf0f.

[47] S. Murakami, N. Nakajima, M. Okamoto, Finite beta effects on the icrf and nbi heating in the large helical device, Trans. Fusion Technol. 27 (1995) 256.

[48] S. Murakami, A. Fukuyama, T. Akutsu, N. Nakajima, V. Chan, M. Choi, S. Chiu, L. Lao, V. Kasilov, T. Mutoh, et al., A global simulation study of icrf heating in the LHD, Nuclear fusion 46 (7) (2006) S425. doi:https://doi.org/10.1088/0029-5515/46/7/S05.

[49] H. Lee, K. Ida, M. Osakabe, M. Yokoyama, C. Suzuki, K. Nagaoka, R. Seki, M. Yoshinuma, N. Tamura, L. E. Group, et al., Dynamic transport study of heat and momentum transport in a plasma with improved ion confinement in the large helical device, Plasma Physics and Controlled Fusion 55 (1) (2012) 014011. doi:https://doi.org/10.1088/0741-3335/55/1/014011.

[50] M. Honda, S. Satake, Y. Suzuki, G. Matsunaga, K. Shinohara, M. Yoshida, A. Matsuyama, S. Ide, H. Urano, Experimental analyses and predictive simulations of toroidal rotation driven by the neoclassical toroidal viscosity in rippled tokamaks, Nuclear Fusion 54 (11) (2014) 114005. doi:https://doi.org/10.1088/0029-5515/54/11/114005.

[51] Y. Morishita, S. Murakami, K. Likin, D. T. Anderson, Simulation study of neutral beam injection heating in the hsx plasma, Plasma and Fusion Research 14 (2019) 3403152–3403152. doi:https://doi.org/10.1585/pfr.14.3403152.

[52] H. YAMAGUCHI, S. MURAKAMI, Effect of impurity ions on nbi heating in LHD plasmas, Plasma and Fusion Research 9 (2014) 3403127–3403127. doi:10.1585/pfr.9.3403127.

[53] H. YAMAGUCHI, S. MURAKAMI, Nbi beam ion distributions in the presence of magnetic islands in helical plasmas, Plasma and Fusion Research 11 (2016) 2403094–2403094. doi:10.1585/pfr.11.2403094.

[54] H. Yamaguchi, S. Murakami, Simulation study of nbi heating in the time-evolving and multi-ion-species plasmas of LHD, Nuclear Fusion 56 (2) (2015) 026003. doi:https://doi.org/10.1088/0029-5515/56/2/026003.

[55] R. Fischer, A. Dinklage, E. Pasch, Bayesian modelling of fusion diagnostics, Plasma Physics and Controlled Fusion 45 (7) (2003) 1095. doi:10.1088/0741-3335/45/7/304.

[56] E. Maljaars, F. Felici, T. Blanken, C. Galperti, O. Sauter, M. De Baar, F. Carpanese, T. Goodman, D. Kim, S. Kim, et al., Profile control simulations and experiments on tcv: a controller test environment and results using a model-based predictive controller, Nuclear Fusion 57 (12) (2017) 126063. doi:https://doi.org/10.1088/1741-4326/aa8c48.

[57] M. Messmer, F. Felici, O. Sauter, A. Teplukhina, J. Loenen, M. Reich, R. Fischer, D. Rittich, R. Jaspers, et al., Optimal mse polarisation angle and q-profile estimation using kalman filters and the plasma simulator raptor, Plasma Physics and Controlled Fusion 61 (3) (2019) 035011. doi:https://orcid.org/0000-0001-7585-376X.

[58] Y. Morishita, S. Murakami, M. Yokoyama, G. Ueno, Data assimilation system based on integrated transport simulation of large helical device plasma, Nuclear Fusion 60 (5) (2020) 056001. doi:https://doi.org/10.1088/1741-4326/ab7596.

[59] R. Seki, K. Ogawa, M. Isobe, M. Yokoyama, S. Murakami, H. Nuga, S. Kamio, Y. Fujiwara, M. Osakabe, L. E. Group, et al., Evaluation of neutron emission rate with fit3d-dd code in large helical device, Plasma and Fusion Research 14 (2019) 3402126–3402126. doi:https://doi.org/10.1585/pfr.14.3402126.

[60] M. Yokoyama, C. Suzuki, R. Seki, M. Osakabe, M. Yoshinuma, M. Sato, A. Wakasa, S. Murakami, A. Fukuyama, Y. Suzuki, et al., Development of integrated transport analysis suite for LHD plasmas towards transport model validation and increased predictability, Plasma and Fusion Research 8 (2013) 2403016–2403016. doi:https://doi.org/10.1585/pfr.8.2403016.

[61] Y. Morishita, S. Murakami, M. Yokoyama, R. Seki, H. Nuga, M. Osakabe, G. Ueno, Development of rapid simulation code for nbi heating analysis in LHD, Journal of Fusion Energy 41 (1) (2022) 1–8.

[62] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, New York, 2007.

[63] H. Nuga, R. Seki, K. Ogawa, S. Kamio, Y. Fujiwara, M. Osakabe, M. Isobe, T. Nishitani, M. Yokoyama, L. E. Group, et al., Analysis of energetic particle confinement in LHD using neutron measurement and simulation codes, Plasma and Fusion Research 14 (2019) 3402075–3402075. doi:https://doi.org/10.1585/pfr.14.3402075.

[64] M. F. Møller, A scaled conjugate gradient algorithm for fast supervised learning, Neural networks 6 (4) (1993) 525–533. doi:https://doi.org/10.1016/S0893-6080(05)80056-5.

[65] J. D. Gaffey, Energetic ion distribution resulting from neutral beam injection in tokamaks, Journal of Plasma Physics 16 (2) (1976) 149–169. doi:https://doi.org/10.1017/S0022377800020134.

[66] Y. Morishita, S. Murakami, M. Yokoyama, G. Ueno, ASTI: Data assimilation system for particle and heat transport in toroidal plasmas, Computer Physics Communications 274 (2022) 108287. doi:https://doi.org/10.1016/j.cpc.2022.108287.

[67] C. De Boor, A practical guide to splines, springer-verlag New York, 1978.

[68] L. T. Murray, A. M. Fiore, D. T. Shindell, V. Naik, L. W. Horowitz, Large uncertainties in global hydroxyl projections tied to fate of reactive nitrogen and carbon, Proceedings of the National Academy of Sciences 118 (43) (2021) e2115204118. doi:10.1073/pnas.2115204118.

[69] M. Marsili, I. Mastromatteo, Y. Roudi, On sampling and modeling complex systems, Journal of Statistical Mechanics: Theory and Experiment 2013 (09) (2013) P09003. doi:10.1088/1742-5468/2013/09/P09003.

[70] J. C. Fyfe, V. V. Kharin, B. D. Santer, J. N. S. Cole, N. P. Gillett, Significant impact of forcing uncertainty in a large ensemble of climate model simulations, Proceedings of the National Academy of Sciences 118 (23) (2021) e2016549118. doi:10.1073/pnas.2016549118.

[71] D. Q. Mayne, Model predictive control: Recent developments and future promise, Automatica 50 (12) (2014) 2967–2986. doi:http://dx.doi.org/10.1016/j.automatica.2014.10.128.

[72] Y. Morishita, S. Murakami, M. Yokoyama, G. Ueno, submitted to Journal of Computational Science.

[73] A. Rasheed, O. San, T. Kvamsdal, Digital twin: Values, challenges and enablers from a modeling perspective, Ieee Access 8 (2020) 21980–22012. doi:https://doi.org/10.1109/ACCESS.2020.2970143.

[74] M. Liu, S. Fang, H. Dong, C. Xu, Review of digital twin about concepts, technologies, and industrial applications, Journal of Manufacturing Systems 58 (2021) 346–361. doi:https://doi.org/10.1016/j.jmsy.2020.06.017.

[75] J. Masison, J. Beezley, Y. Mei, H. Ribeiro, A. C. Knapp, L. S. Vieira, B. Adhikari, Y. Scindia, M. Grauer, B. Helba, W. Schroeder, B. Mehrad, R. Laubenbacher, A modular computational framework for medical digital twins, Proceedings of the National Academy of Sciences 118 (20) (2021) e2024287118. doi:10.1073/pnas.2024287118.

[76] M. G. Kapteyn, J. V. Pretorius, K. E. Willcox, A probabilistic graphical model foundation for enabling predictive digital twins at scale, Nature Computational Science 1 (5) (2021) 337–347. doi:https://doi.org/10.1038/s43588-021-00069-0.

[77] N. Kantas, J. Maciejowski, A. Lecchini-Visintini, Sequential monte carlo for model predictive control, in: Nonlinear model predictive control, Springer, 2009, pp. 263–273.

[78] C. Andrieu, A. Doucet, S. S. Singh, V. B. Tadic, Particle methods for change detection, system identification, and control, Proceedings of the IEEE 92 (3) (2004) 423–438. doi:10.1109/JPROC.2003.823142.

[79] A. Wakasa, A. Fukuyama, S. Murakami, M. Miki, M. Yokoyama, M. Sato, S. Toda, H. Funaba, K. Tanaka, K. Ida, H. Yamada, M. Honda, N. Nakajima, Integrated transport simulation of LHD plasmas using task3d, proc. 23rd IAEA Fusion Energy Conf. (Daejon) P4.029 (2010).

[80] J. Kates-Harbeck, A. Svyatkovskiy, W. Tang, Predicting disruptive instabilities in controlled fusion plasmas through deep learning, Nature 568 (7753) (2019) 526–531. doi:10.1038/s41586-019-1116-4.

[81] J. Degrave, F. Felici, J. Buchli, M. Neunert, B. Tracey, F. Carpanese, T. Ewalds, R. Hafner, A. Abdolmaleki, D. de Las Casas, et al., Magnetic control of tokamak plasmas through deep reinforcement learning, Nature 602 (7897) (2022) 414–419. doi:10.1038/s41586-021-04301-9.

[82] F. Matos, V. Menkovski, F. Felici, A. Pau, F. Jenko, T. Team, E. M. Team, et al., Classification of tokamak plasma confinement states with convolutional recurrent neural networks, Nuclear Fusion 60 (3) (2020) 036022. doi:10.1088/1741-4326/ab6c7a.

[83] T. Yokoyama, H. Yamada, S. Masuzaki, J. Miyazawa, K. Mukai, B. J. Peterson, N. Tamura, R. Sakamoto, G. Motojima, K. Ida, et al., Prediction of radiative collapse in large helical device using feature extraction by exhaustive search, Journal of Fusion Energy 39 (6) (2020) 500–511.