Estimation-Based Ocean Flow Field Reconstruction Using Profiling Floats

Huazhen Fang, Raymond A. de Callafon and Jorge Cortés

Abstract

This note considers ocean flow field monitoring using profiling floats and investigates a foundational estimation problem underlying flow field reconstruction, which is known as simultaneous input and state estimation. We take a Bayesian perspective to develop the needed estimation approaches. With this perspective, we first build Bayesian estimation principles for input and state estimation for both the cases of filtering and smoothing. Then, we formulate maximum a posteriori estimation problems and solve them using the classical Gauss-Newton method, leading to a set of algorithms. The proposed algorithms represent a new development of the Bayesian estimation theory to address joint input and state estimation and generalize a number of relevant methods in the literature. We illustrate the effectiveness of our approach in addressing an oceanographic flow field estimation problem based on profiling floats that measure position intermittently and acceleration continuously.

Keywords

Ocean sampling, input and state estimation, Bayesian estimation, nonlinear estimation, filtering and smoothing

I. Introduction

This chapter presents a study of ocean flow field reconstruction using profiling floats. Buoyancy-controlled, semi-autonomous profiling floats provide an economic and flexible means to monitor ocean flows. However, a key challenge in enabling this capability lies in how to extract the velocities of the ocean flow at sampled locations from the recorded motion data of floats. This leads to a problem that can be cast as the joint estimation of a nonlinear dynamic system's inputs and states. This chapter develops solutions from the perspective of Bayesian estimation and evaluates their application to ocean flow field reconstruction through simulations. The proposed results may also be useful in many other scientific and engineering problems involving input and state estimation.

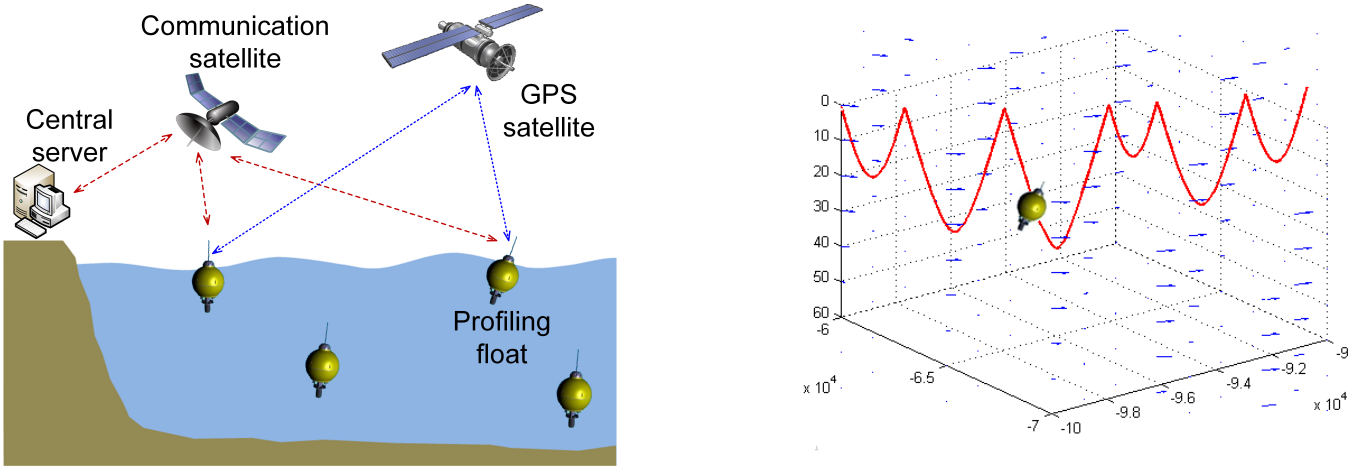
A. Background

The oceans, which cover over two-thirds of the Earth's surface, have been an essential part of human life as food sources and transportation routes for thousands of years. Today, the world is increasingly looking to them in seeking solutions to various grand challenges such as natural resource and energy shortage and climate change (Costanza, 1999). Associated with this trend is a growing commitment from the research community to understand the oceans with greater spatial and temporal coverage and finer accuracy. A fundamental research problem is to monitor ocean flows that result from continuous, directed movement of ocean water and significantly impact the marine environment, maritime transport, pollution spread, and global climate.

Recent years have witnessed an exponential interest in deploying autonomous underwater vehicles (AUVs) for flow monitoring, which are capable of operating in dangerous underwater environments without the need for human

H. Fang is with the Department of Mechanical Engineering, University of Kansas, Lawrence, KS 66045, USA e-mail: fang@ku.edu.

R. A. de Callafon and J. Cortés are with Department of Mechanical and Aerospace Engineering, University of California, San Diego, CA 92093, USA e-mail: {callafon, cortes}@ucsd.edu.



(a) (b)

Fig. 1: (a) The scenario for flow-field reconstruction based on submersible profiling floats; (b) the traveling profile (submerging/surfacing) of a profiling float.

presence. Crucial for the development of advanced AUVs is an integration of control, communication and computation technologies, which has benefited significantly from the recent sweeping advances in these fields. Technological sophistication, however, comes initially with high economic costs, making AUVs expensive and not widely available to oceanographic research. This has provided impetu for using small, inexpensive AUVs, known as profiling floats. A profiling float only has buoyancy control to change its vertical position and laterally drifts along the flow. Despite underactuation, it can travel across the ocean at different depths for long durations to observe temperature, salinity and currents. About 3,900 such instruments, through the internationally collaborative Argo program, have been deployed in global oceans to collect data for climate and oceanographic research (Riser et al., 2016).

Our work here considers a profiling-float-based ocean observing system (Colgan, 2006; Ouimet and Cortes, 2014), which is schematically illustrated in Fig. 1a. This system consists of a few profiling floats. They can be released at different locations in a region of interest in the ocean. Then, each one travels in cycling movement patterns of submerging/surfacing (see Fig. 1b), with lateral motion driven by ocean currents and vertical motion regulated by buoyancy. While underwater, each float stores a time record of its current depth, acceleration and other oceanographically-relevant quantities. When coming up to the surface, it determines its geographical location and then transmits all the data *via* a communication satellite for analysis and computation. The acceleration is continually measured by an on-board accelerometer, and the position by a satellite-based Global Positioning System (GPS). It should be noted that a float's position is intermittently available—it can be measured only when the float is on surface, because GPS signals are seriously attenuated while underwater.

The objective here is to reconstruct the flow field and monitor the movement of the profiling floats. Hence, it is necessary to simultaneously estimate the flow velocities, which act as external inputs applied to a float, and the float's underwater positions and velocities, which make up the float's states, using the float's position and acceleration measurements. This leads us to investigate the problem of simultaneous input and state estimation (SISE) for a nonlinear system based on the system's output measurement data. It is noteworthy that problems of a similar kind also arise in other fields, with examples including fault detection, disturbance rejection, vehicle tire-road friction estimation, and weather forecasting (Schubert et al., 2012; Imine et al., 2006; Kitanidist, 1987), where a system's input and state variables are both unknown and need to be estimated. This observation, together with the problem of

flow field reconstruction specifically considered here, motivates us to develop general-purpose solutions.

B. Literature Survey

Since the 1960s, the systems and control community has explored state estimation with unknown inputs. The earliest relevant work, to our knowledge, is (Friedland, 1969), which studies state estimation with a constant but unknown disturbance input. The proposed solution is to augment the state vector to include the disturbance and then apply the Kalman filter (KF). In (Kitanidis, 1987), minimum variance unbiased estimation (MVUE) is exploited to estimate states in the presence of completely unknown inputs. Based on (Kitanidis, 1987), a series of MVUE-based methods are proposed in (Darouach and Zasadzinski, 1997; Darouach et al., 2003; Cheng et al., 2009), along with a detailed analysis of the existence and stability of the estimators. For these studies, the focus is on enabling state estimation despite unknown inputs, leaving aside input estimation. However, the unknown inputs are also important for many real-world systems and, if successfully estimated, can be useful for system analysis and control synthesis. As such, SISE presents even more appeal and has attracted significant attention from researchers.

The literature on SISE includes two main subjects, simultaneous input and state filtering (SISF) and smoothing (SISS). As the names suggest, SISF seeks to estimate the input and state at the current time instant using the history of output measurements, and SISS seeks to estimate the input and state at a past time instant using measurements inclusive of those lagging behind that time instant. The former thus allows for real-time estimation, and the latter requires off-line computation leading to better estimation accuracy in spite of an increase in estimator complexity.

For SISF, an early contribution is (Mendel, 1977), which adapts the KF to estimate a linear system's states along with covariance-known white noise disturbances. More recent works usually consider completely unknown inputs and develop SISF techniques by modifying some existing state estimation approaches. Among them, we highlight those based on the KF (Hsieh, 2000, 2010, 2011), moving horizon estimation (MHE) (Pina and Botto, 2006), H_∞-filtering (You et al., 2008), sliding mode observers (Floquet et al., 2007), and MVUE (Gillijns and De Moor, 2007a,b; Fang et al., 2008, 2011; Fang and de Callafon, 2012; Yong et al., 2016; Shi et al., 2016). Though reaching a certain level of maturity for linear systems, SISF can become rather complicated when nonlinear systems are considered. In (Corless and Tu, 1998; Ha and Trinh, 2004), SISF methods are developed for a special class of nonlinear deterministic systems, which consist of a nominally linear part and a nonlinear part. The work (Hsieh, 2013) decouples the unknown inputs from the nonlinear system and then extends linear SISF methods to handle the estimation. In our prior work (Fang et al., 2013; Fang and de Callafon, 2011; Fang et al., 2017), we have shown that Bayesian estimation is a viable approach to cope with SISF for generic-form nonlinear stochastic systems, from low to high dimensions.

When it comes to SISS, and despite its significance, only few studies have been reported. Extrapolating the Bayesian approach in (Fang and de Callafon, 2011; Fang et al., 2013), we proposed SISS algorithms for nonlinear systems in (Fang and de Callafon, 2013; Fang et al., 2015) and also specialized them to linear systems. A linear smoothing algorithm is also developed in (Yong et al., 2014) as an extension of MVUE-based SISF in (Yong et al., 2016). An exhaustive search shows no more results in the literature other than these studies. It is worth pointing out that ocean flow field estimation is a problem well suited for SISS, because a typical ocean flow or circulation changes on time scales ranging from a few days to a season, thus allowing off-line but more accurate estimation.

C. Overview of this Chapter

This chapter summarizes our previous work (Fang and de Callafon, 2011; Fang et al., 2013; Fang and de Callafon, 2013; Fang et al., 2015) to present a systematic introduction of Bayesian-estimation-based SISF and SISS methods for nonlinear systems. The core idea of Bayesian estimation is updating the probabilistic belief of an unknown variable

using measurement data and then extracting the best estimate in a probabilistic sense. This methodology has been proven useful for constructing various state and parameter estimation techniques over the past decades. Our study further advances this actively researched area to investigate SISE. We start by developing Bayesian SISF and SISS paradigms to conceptually look at and understand the SISE problems from the perspective of statistical estimation. This view opens up the possibility of developing Bayesian SISF and SISS estimators. Building upon the Bayesian paradigm, we formulate maximum a posteriori probability (MAP) estimation problems and solve them using the Gauss-Newton method, which is a numerical optimization approach capable of overcoming the effects of a system's intrinsic nonlinearity on estimation accuracy. For each case of SISF and SISS, we develop separate investigations for systems with and without direct input-to-output feedthrough, leading to a set of estimation algorithms custom-built according to the system structure. We conduct a simulation study to apply the proposed algorithms in the reconstruction of an ocean flow field using observation data collected by a group of buoyancy-controlled profiling floats. The estimation performance validates the effectiveness of the proposed algorithms.

II. SISF for Systems with Direct Feedthrough

This section studies the problem of SISF for dynamic systems with direct feedthrough. It starts with developing a Bayesian estimation principle and then derives an SISF algorithm using the idea of Bayesian MAP estimation.

A. The Bayesian Paradigm

Consider a nonlinear system with direct input-to-output feedthrough:

$$\begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{w}_k, \\ \boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{v}_k, \end{cases} \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector, $\boldsymbol{u} \in \mathbb{R}^m$ the input vector, $\boldsymbol{y} \in \mathbb{R}^p$ the measurement vector, and $\boldsymbol{w} \in \mathbb{R}^n$ and $\boldsymbol{v} \in \mathbb{R}^m$ mutually independent zero-mean white Gaussian noise sequences, with covariances $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$, respectively. The mappings $\boldsymbol{f} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$ and $\boldsymbol{h} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^p$ are the state transition and measurement functions, respectively, which are assumed to be C^1. We also assume $\nabla_{\boldsymbol{u}} \boldsymbol{h}$ has full rank. For the above system, our objective is to estimate $\boldsymbol{u}_k$ and $\boldsymbol{x}_k$ based on the measurement set $\boldsymbol{Y}_{1:k} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_k\}$. To build a Bayesian estimator, one is interested in finding out the probability density functions (pdf's) of $\boldsymbol{u}_k$ and $\boldsymbol{x}_k$ conditioned on $\boldsymbol{Y}_{1:k}$, i.e., $p(\boldsymbol{u}_k, \boldsymbol{x}_k | \boldsymbol{Y}_{1:k})$. As it is also desirable to achieve sequential estimation, the problem becomes how to enable the passing to $p(\boldsymbol{u}_k, \boldsymbol{x}_k | \boldsymbol{Y}_{1:k})$ from $p(\boldsymbol{u}_{k-1}, \boldsymbol{x}_{k-1} | \boldsymbol{Y}_{1:k-1})$. Akin to Bayesian state estimation Candy (2009), this can be accomplished in a two-step procedure of prediction and update.

The step of prediction is used to determine the conditional pdf $p(\boldsymbol{x}_k | \boldsymbol{Y}_{1:k-1})$. By the Chapman-Kolmogorov equation (Honerkamp, 1993), we have

$$p(\boldsymbol{x}_k | \boldsymbol{Y}_{1:k-1}) = \iint p(\boldsymbol{x}_k | \boldsymbol{u}_{k-1}, \boldsymbol{x}_{k-1}, \boldsymbol{Y}_{1:k-1}) \cdot p(\boldsymbol{u}_{k-1}, \boldsymbol{x}_{k-1} | \boldsymbol{Y}_{1:k-1}) \mathrm{d}\boldsymbol{u}_{k-1} \mathrm{d}\boldsymbol{x}_{k-1}.$$

We note that $p(\boldsymbol{x}_k | \boldsymbol{u}_{k-1}, \boldsymbol{x}_{k-1}, \boldsymbol{Y}_{1:k-1}) = p(\boldsymbol{x}_k | \boldsymbol{u}_{k-1}, \boldsymbol{x}_{k-1})$, since $\boldsymbol{x}_k$ depends on only $\boldsymbol{u}_{k-1}$ and $\boldsymbol{x}_{k-1}$ because of the Markovian state propagation as shown in (1). Hence, it follows that

$$p(\boldsymbol{x}_k | \boldsymbol{Y}_{1:k-1}) = \iint p(\boldsymbol{x}_k | \boldsymbol{u}_{k-1}, \boldsymbol{x}_{k-1}) \cdot p(\boldsymbol{u}_{k-1}, \boldsymbol{x}_{k-1} | \boldsymbol{Y}_{1:k-1}) \mathrm{d}\boldsymbol{u}_{k-1} \mathrm{d}\boldsymbol{x}_{k-1}. \tag{2}$$

When the measurement $\boldsymbol{y}_k$ arrives, it can be used to update $p(\boldsymbol{x}_k | \boldsymbol{Y}_{1:k-1})$ along with the conditional pdf of $\boldsymbol{u}_k$ (because it is the first measurement conveying information about $\boldsymbol{u}_k$) *via* determining $p(\boldsymbol{u}_k, \boldsymbol{x}_k | \boldsymbol{Y}_{1:k})$. To proceed, we make the following assumption:

(A1) $\{\boldsymbol{u}_k\}$ is a white process, independent of $\boldsymbol{x}_0$, $\{\boldsymbol{w}_k\}$ and $\{\boldsymbol{v}_k\}$.

Here, 'white' means that $\boldsymbol{u}_k$ and $\boldsymbol{u}_l$ are independent random vectors for $k \neq l$. Such a whiteness assumption is inspired by (Robinson, 1957), which has been a foundation for many seismic data processing algorithms. The intuitions underlying it are: 1) $\boldsymbol{u}_k$, completely unknown to us, may assume all possible values; 2) from the knowledge of $\boldsymbol{u}_k$ we cannot predict $\boldsymbol{u}_l$ for $k \neq l$. A similar treatment of $\{\boldsymbol{u}_k\}$ as a stochastic process is proposed in (Friedland, 1969), which yet assumes the wide-sense description of $\boldsymbol{u}_k$ as known. By (A1), $\boldsymbol{u}_k$ is independent of $\boldsymbol{x}_k$ and $\boldsymbol{Y}_{1:k-1}$ (Gut, 2005, Theorem 10.4, pp. 71).

Using the Bayes' rule repeatedly, we obtain

$$p(\boldsymbol{u}_k, \boldsymbol{x}_k|\boldsymbol{Y}_{1:k}) = \frac{p(\boldsymbol{y}_k|\boldsymbol{u}_k, \boldsymbol{x}_k, \boldsymbol{Y}_{1:k-1}) \cdot p(\boldsymbol{u}_k, \boldsymbol{x}_k|\boldsymbol{Y}_{1:k-1})}{p(\boldsymbol{y}_k|\boldsymbol{Y}_{1:k-1})}.$$

Note that $p(\boldsymbol{y}_k|\boldsymbol{u}_k, \boldsymbol{x}_k, \boldsymbol{Y}_{1:k-1}) = p(\boldsymbol{y}_k|\boldsymbol{u}_k, \boldsymbol{x}_k)$ due to the fact that $\boldsymbol{y}_k$ entirely depends on $\boldsymbol{u}_k$ and $\boldsymbol{x}_k$, and that $p(\boldsymbol{u}_k, \boldsymbol{x}_k|\boldsymbol{Y}_{1:k-1}) = p(\boldsymbol{x}_k|\boldsymbol{Y}_{1:k-1}) \cdot p(\boldsymbol{u}_k)$ as a result of $\boldsymbol{u}_k$'s independence from $\boldsymbol{x}_k$ and $\boldsymbol{Y}_{1:k-1}$. Consequently,

$$p(\boldsymbol{u}_k, \boldsymbol{x}_k|\boldsymbol{Y}_{1:k}) = \frac{p(\boldsymbol{y}_k|\boldsymbol{u}_k, \boldsymbol{x}_k) \cdot p(\boldsymbol{x}_k|\boldsymbol{Y}_{1:k-1}) \cdot p(\boldsymbol{u}_k)}{p(\boldsymbol{y}_k|\boldsymbol{Y}_{1:k-1})}$$

One can see that $p(\boldsymbol{u}_k)/p(\boldsymbol{y}_k|\boldsymbol{Y}_{1:k-1})$ plays the role of a proportionality coefficient. This implies

$$p(\boldsymbol{u}_k, \boldsymbol{x}_k|\boldsymbol{Y}_{1:k}) \propto p(\boldsymbol{y}_k|\boldsymbol{u}_k, \boldsymbol{x}_k) \cdot p(\boldsymbol{x}_k|\boldsymbol{Y}_{1:k-1}). \tag{3}$$

Here, (2) and (3) form the Bayesian SISF paradigm for systems with direct feedthrough. Sequentially updating them not only provides a conceptual Bayesian solution to the considered SISF problem, but also yields a statistical framework within which different SISE methods can be developed. Our next step is to derive an SISF algorithm by formulating and solving an MAP estimation problem based on the proposed Bayesian paradigm.

B. SISF Algorithm Development

Let us begin with some Gaussian distribution assumptions for concerned pdf's. Specifically, we assume

(A2) $p(\boldsymbol{u}_k, \boldsymbol{x}_k|\boldsymbol{Y}_{1:k}) \sim \mathcal{N}\left(\begin{bmatrix}\hat{\boldsymbol{u}}_{k|k}\\ \hat{\boldsymbol{x}}_{k|k}\end{bmatrix}, \begin{bmatrix}\boldsymbol{P}^{\boldsymbol{u}}_{k|k} & \boldsymbol{P}^{\boldsymbol{ux}}_{k|k}\\ (\boldsymbol{P}^{\boldsymbol{ux}}_{k|k})^\top & \boldsymbol{P}^{\boldsymbol{x}}_{k|k}\end{bmatrix}\right)$,

(A3) $p(\boldsymbol{y}_k|\boldsymbol{u}_k, \boldsymbol{x}_k) \sim \mathcal{N}\left(\boldsymbol{h}(\boldsymbol{u}_k, \boldsymbol{x}_k), \boldsymbol{R}_k\right)$,

(A4) $p(\boldsymbol{x}_k|\boldsymbol{Y}_{1:k-1}) \sim \mathcal{N}\left(\hat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{P}^{\boldsymbol{x}}_{k|k-1}\right)$,

where $\hat{\boldsymbol{u}}_{k|k}$ is the estimate of $\boldsymbol{u}_k$ given $\boldsymbol{Y}_k$ with associated covariance $\boldsymbol{P}^{\boldsymbol{u}}_{k|k}$, $\hat{\boldsymbol{x}}_{k|k-1}$ and $\hat{\boldsymbol{x}}_{k|k}$ are the estimates of $\boldsymbol{x}_k$ given $\boldsymbol{Y}_{1:k-1}$ and $\boldsymbol{Y}_{1:k}$ with covariances $\boldsymbol{P}^{\boldsymbol{x}}_{k|k-1}$ and $\boldsymbol{P}^{\boldsymbol{x}}_{k|k}$, respectively. Ideally, if knowledge of $p(\boldsymbol{u}_k, \boldsymbol{x}_k|\boldsymbol{Y}_{1:k})$ is available for each k, $\hat{\boldsymbol{u}}_{k|k}$ and $\hat{\boldsymbol{x}}_{k|k}$ can be readily obtained by MAP or some other way. However, contrary to this ideal, determining $p(\boldsymbol{u}_k, \boldsymbol{x}_k|\boldsymbol{Y}_{1:k})$ accurately is known as an intractable issue for nonlinear systems. In order to overcome this problem, (A2)-(A4) are made to approximately describe the pdf's by replacing each with a Gaussian distribution with the same mean and covariance. Assumptions on Gaussian distributions analogous to (A2)-(A4) are commonly held in nonlinear estimation algorithms, e.g., (Anderson and Moore, 1979; Bell and Cathey, 1993; Spinello and Stilwell, 2010).

Based on the idea of MAP estimation, we intend to enable state prediction by considering

$$\hat{\boldsymbol{x}}_{k|k-1} = \arg\max_{\boldsymbol{x}_k} p(\boldsymbol{x}_k|\boldsymbol{Y}_{1:k-1}), \tag{4}$$

which maximizes the probabilistic presence of $\boldsymbol{x}_k$ given $\boldsymbol{Y}_{1:k-1}$. To solve this problem, we first look at the first-order Taylor series expansion of $\boldsymbol{f}(\boldsymbol{u}_k, \boldsymbol{x}_k)$ around $(\hat{\boldsymbol{u}}_{k|k}, \hat{\boldsymbol{x}}_{k|k})$:

$$\boldsymbol{f}\left(\boldsymbol{u}_k, \boldsymbol{x}_k\right) \approx \boldsymbol{f}\left(\hat{\boldsymbol{u}}_{k|k}, \hat{\boldsymbol{x}}_{k|k}\right) + \nabla \boldsymbol{f}\left(\hat{\boldsymbol{u}}_{k|k}, \hat{\boldsymbol{x}}_{k|k}\right) \begin{bmatrix}\boldsymbol{u}_k - \hat{\boldsymbol{u}}_{k|k}\\ \boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k}\end{bmatrix}, \tag{5}$$

where $\nabla \boldsymbol{f} = \begin{bmatrix} \nabla_{\boldsymbol{u}} \boldsymbol{f} & \nabla_{\boldsymbol{x}} \boldsymbol{f} \end{bmatrix}$. Then by (A2), (2) and (5), the approximate solution to (4) is given by

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{f}\left(\hat{\boldsymbol{u}}_{k-1|k-1}, \hat{\boldsymbol{x}}_{k-1|k-1}\right), \tag{6}$$

with the associated prediction error covariance $\boldsymbol{P}^{\boldsymbol{x}}_{k|k-1}$ given by

$$\boldsymbol{P}_k \approx \nabla \boldsymbol{f}\left(\hat{\boldsymbol{u}}_{k-1|k-1}, \hat{\boldsymbol{x}}_{k-1|k-1}\right) \begin{bmatrix} \boldsymbol{P}^{\boldsymbol{u}}_{k-1|k-1} & \boldsymbol{P}^{\boldsymbol{ux}}_{k-1|k-1} \\ (\boldsymbol{P}^{\boldsymbol{ux}}_{k-1|k-1})^\top & \boldsymbol{P}^{\boldsymbol{x}}_{k-1|k-1} \end{bmatrix} \nabla \boldsymbol{f}^\top \left(\hat{\boldsymbol{u}}_{k-1|k-1}, \hat{\boldsymbol{x}}_{k-1|k-1}\right) + \boldsymbol{Q}_{k-1}. \tag{7}$$

Then, (6) and (7) constitute the prediction formulae together, computing the state prediction and prediction error covariance, respectively.

Let us now consider updating $p(\boldsymbol{x}_k|\boldsymbol{Y}_{1:k-1})$ using $\boldsymbol{y}_k$ and define the following MAP estimator:

$$\begin{bmatrix} \hat{\boldsymbol{u}}_{k|k} \\ \hat{\boldsymbol{x}}_{k|k} \end{bmatrix} = \arg\max_{\boldsymbol{u}_k, \boldsymbol{x}_k} p(\boldsymbol{u}_k, \boldsymbol{x}_k|\boldsymbol{Y}_{1:k}). \tag{8}$$

We further define the MAP cost function as $L(\boldsymbol{u}_k, \boldsymbol{x}_k) = p(\boldsymbol{u}_k, \boldsymbol{x}_k|\boldsymbol{Y}_{1:k})$. According to (3) and (A3)-(A4), one has

$$L(\boldsymbol{u}_k, \boldsymbol{x}_k) = \lambda \cdot \exp\left[-\boldsymbol{\alpha}_k^\top \boldsymbol{R}_k^{-1} \boldsymbol{\alpha}_k - \boldsymbol{\beta}_k^\top (\boldsymbol{P}^{\boldsymbol{x}}_{k|k-1})^{-1} \boldsymbol{\beta}_k\right],$$

where λ combines all the constants, $\boldsymbol{\alpha}_k = \boldsymbol{y}_k - \boldsymbol{h}(\boldsymbol{u}_k, \boldsymbol{x}_k)$ and $\boldsymbol{\beta}_k = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1}$. It is easier to deal with the logarithmic cost function $\ell(\boldsymbol{u}_k, \boldsymbol{x}_k) = -\ln L(\boldsymbol{u}_k, \boldsymbol{x}_k)$:

$$\ell(\boldsymbol{u}_k, \boldsymbol{x}_k) = \delta + \boldsymbol{r}^\top(\boldsymbol{u}_k, \boldsymbol{x}_k) \cdot \boldsymbol{r}(\boldsymbol{u}_k, \boldsymbol{x}_k), \tag{9}$$

where $\delta = -\ln \lambda$ and

$$\boldsymbol{r}(\boldsymbol{u}_k, \boldsymbol{x}_k) = \begin{bmatrix} \boldsymbol{R}_k^{-\frac{1}{2}} \boldsymbol{\alpha}_k \\ \left(\boldsymbol{P}^{\boldsymbol{x}}_{k|k-1}\right)^{-\frac{1}{2}} \boldsymbol{\beta}_k \end{bmatrix}.$$

Thus, (8) can be equivalently written as

$$\begin{bmatrix} \hat{\boldsymbol{u}}_{k|k} \\ \hat{\boldsymbol{x}}_{k|k} \end{bmatrix} = \arg\min_{\boldsymbol{u}_k, \boldsymbol{x}_k} \ell(\boldsymbol{u}_k, \boldsymbol{x}_k). \tag{10}$$

The MAP optimization in (10) usually defies the development of a closed-form solution when considered for a nonlinear system. However, as a nonlinear least-squares problem, it can be numerically addressed using the Gauss-Newton method (Björck, 1996). The classical Gauss-Newton method can iteratively compute the sequences of approximations $\hat{\boldsymbol{u}}_k^{(i)}$ and $\hat{\boldsymbol{x}}_k^{(i)}$, where (i) denotes the iteration step. Specifically,

$$\hat{\boldsymbol{\xi}}_k^{(i+1)} = \hat{\boldsymbol{\xi}}_k^{(i)} - \left[\nabla_{\boldsymbol{\xi}}^\top \boldsymbol{r}\left(\hat{\boldsymbol{\xi}}_k^{(i)}\right) \cdot \nabla_{\boldsymbol{\xi}} \boldsymbol{r}\left(\hat{\boldsymbol{\xi}}_k^{(i)}\right)\right]^{-1} \cdot \nabla_{\boldsymbol{\xi}}^\top \boldsymbol{r}\left(\hat{\boldsymbol{\xi}}_k^{(i)}\right) \cdot \boldsymbol{r}\left(\hat{\boldsymbol{\xi}}_k^{(i)}\right), \tag{11}$$

where $\boldsymbol{\xi}_k = \begin{bmatrix} \boldsymbol{u}_k^\top & \boldsymbol{x}_k^\top \end{bmatrix}^\top$, and $\nabla_{\boldsymbol{\xi}} \boldsymbol{r} = \begin{bmatrix} \nabla_{\boldsymbol{u}} \boldsymbol{r} & \nabla_{\boldsymbol{x}} \boldsymbol{r} \end{bmatrix}$. One can let the initial guess be $\hat{\boldsymbol{\xi}}_k^{(0)} = [\boldsymbol{0}^\top \ \hat{\boldsymbol{x}}_{k|k-1}^\top]^\top$ for convenience, though it can be set to arbitrary values. The iteration continues until the iteration step (i) reaches the preselected maximum $i_{\max}$ or the difference between two consecutive iterations is less than a preselected small value. Then, $\hat{\boldsymbol{\xi}}_k^{(i)}$ obtained in the final iteration will be exported and assigned to $\hat{\boldsymbol{u}}_{k|k}$ and $\hat{\boldsymbol{x}}_{k|k}$, respectively. The iteration process in (11) refines the input and state estimates continually by re-evaluating the joint estimator around the latest estimated input and state operating point. Despite demanding more computational power, the iterative refinement enhances not only the estimation performance but also the robustness to nonlinearities. In its practical use, one can try to strike a balance between computational complexity and estimation performance by selecting a proper stopping condition.

initialize: $k=0$, $\hat{\boldsymbol{\xi}}_{0|0}=\mathrm{E}(\boldsymbol{\xi}_0)$, $\boldsymbol{P}^{\boldsymbol{\xi}}_{0|0}=p_0\boldsymbol{I}$, where p_0 is typically a large positive value
repeat
$k \leftarrow k+1$
Prediction:
predict the state *via* (6)
compute prediction error covariance *via* (7)
Update:
initialize: $i=0$, $\hat{\boldsymbol{\xi}}_k^{(0)}=[\boldsymbol{0}^\top \ \hat{\boldsymbol{x}}_{k|k-1}^\top]^\top$
while $i<i_{\max}$ **do**
perform Gauss-Newton based joint input and state filtering *via* (11)
$i \leftarrow i+1$
end while
export $\hat{\boldsymbol{\xi}}_k^{(i_{\max})}$ and assign to $\hat{\boldsymbol{u}}_{k|k}$ and $\hat{\boldsymbol{x}}_{k|k}$
compute joint filtering error covariance *via* (12)-(14)
until no more measurements arrive

TABLE I: The `SISF-wDF` algorithm (SISF for systems with direct feedthrough).

The estimation error covariance is equal to the inverse of the Fisher information matrix, as is known for MAP estimators under Gaussian distributions (Mutambara, 1998). Then we have

$$\begin{bmatrix} \boldsymbol{P}^{\boldsymbol{u}}_{k|k} & \boldsymbol{P}^{\boldsymbol{ux}}_{k|k} \\ \left(\boldsymbol{P}^{\boldsymbol{ux}}_{k|k}\right)^\top & \boldsymbol{P}^{\boldsymbol{x}}_{k|k} \end{bmatrix} = \boldsymbol{\mathcal{F}}^{-1}\left(\hat{\boldsymbol{u}}_{k|k}, \hat{\boldsymbol{x}}_{k|k}\right), \tag{12}$$

where $\boldsymbol{\mathcal{F}}$ is the Fisher information matrix defined as

$$\boldsymbol{\mathcal{F}} = \begin{bmatrix} \boldsymbol{\mathcal{F}}^{\boldsymbol{u}} & \boldsymbol{\mathcal{F}}^{\boldsymbol{ux}} \\ (\boldsymbol{\mathcal{F}}^{\boldsymbol{ux}})^\top & \boldsymbol{\mathcal{F}}^{\boldsymbol{x}} \end{bmatrix} = \mathrm{E}\left(\begin{bmatrix} \nabla_{\boldsymbol{u}}^\top \ell \\ \nabla_{\boldsymbol{x}}^\top \ell \end{bmatrix} \begin{bmatrix} \nabla_{\boldsymbol{u}} \ell & \nabla_{\boldsymbol{x}} \ell \end{bmatrix} \right). \tag{13}$$

The explicit formulae for the involved gradients are as follows:

$$\begin{aligned} \nabla_{\boldsymbol{u}} \boldsymbol{r} &= \begin{bmatrix} -\boldsymbol{R}^{-\frac{1}{2}} \nabla_{\boldsymbol{u}} \boldsymbol{h} \\ \boldsymbol{0} \end{bmatrix}, \quad \nabla_{\boldsymbol{x}} \boldsymbol{r} = \begin{bmatrix} -\boldsymbol{R}^{-\frac{1}{2}} \nabla_{\boldsymbol{x}} \boldsymbol{h} \\ \left(\boldsymbol{P}^{\boldsymbol{x}}_{k|k-1}\right)^{-\frac{1}{2}} \end{bmatrix}, \\ \nabla_{\boldsymbol{u}} \ell &= \boldsymbol{r}^\top \nabla_{\boldsymbol{u}} \boldsymbol{r} = \boldsymbol{\alpha}^\top \boldsymbol{R}^{-1} \nabla_{\boldsymbol{u}} \boldsymbol{h}, \\ \nabla_{\boldsymbol{x}} \ell &= \boldsymbol{r}^\top \nabla_{\boldsymbol{x}} \boldsymbol{r} = \boldsymbol{\alpha}^\top \boldsymbol{R}^{-1} \nabla_{\boldsymbol{x}} \boldsymbol{h} + \boldsymbol{\beta}^\top \left(\boldsymbol{P}^{\boldsymbol{x}}_{k|k-1}\right)^{-1}. \end{aligned}$$

Hence, $\boldsymbol{\mathcal{F}}$ is given by

$$\boldsymbol{\mathcal{F}} = \begin{bmatrix} \nabla_{\boldsymbol{u}}^\top \boldsymbol{h} \boldsymbol{R}^{-1} \nabla_{\boldsymbol{u}} \boldsymbol{h} & \nabla_{\boldsymbol{u}}^\top \boldsymbol{h} \boldsymbol{R}^{-1} \nabla_{\boldsymbol{x}} \boldsymbol{h} \\ \nabla_{\boldsymbol{x}}^\top \boldsymbol{h} \boldsymbol{R}^{-1} \nabla_{\boldsymbol{u}} \boldsymbol{h} & \nabla_{\boldsymbol{x}}^\top \boldsymbol{h} \boldsymbol{R}^{-1} \nabla_{\boldsymbol{x}} \boldsymbol{h} + \left(\boldsymbol{P}^{\boldsymbol{x}}_{k|k-1}\right)^{-1} \end{bmatrix}. \tag{14}$$

Putting together the above results yields a nonlinear SISF algorithm named `SISF-wDF`, which is formally described in Table I. This algorithm is based on a novel Bayesian perspective to addressing the SISE problems, while the literature usually considers the problem from the viewpoint of filter design and optimal gain selection. It should be pointed out that the `SISF-wDF` algorithm can be applied to nonlinear systems of general form, instead of being restricted to systems of some required special forms.

Remark 1. *(Improvements to the Gauss-Newton method).* While the basic Gauss-Newton iteration shown in (11) solves linear problems within only a single iteration and has fast local convergence on mildly nonlinear problems, it may suffer from divergence for some nonlinear problems. To improve the convergence performance, a damping coefficient $\alpha^{(i)} > 0$ can be added:

$$\hat{\boldsymbol{\xi}}_k^{(i+1)} = \hat{\boldsymbol{\xi}}_k^{(i)} - \alpha^{(i)} \left[\nabla_{\boldsymbol{\xi}}^\top \boldsymbol{r}\left(\hat{\boldsymbol{\xi}}_k^{(i)}\right) \cdot \nabla_{\boldsymbol{\xi}} \boldsymbol{r}\left(\hat{\boldsymbol{\xi}}_k^{(i)}\right) \right]^{-1} \cdot \nabla_{\boldsymbol{\xi}}^\top \boldsymbol{r}\left(\hat{\boldsymbol{\xi}}_k^{(i)}\right) \cdot \boldsymbol{r}\left(\hat{\boldsymbol{\xi}}_k^{(i)}\right). \tag{15}$$

One can show that the damped Gauss-Newton iteration keeps moving to the critical point in a descent direction for sufficiently small $\alpha^{(i)} > 0$, thus guaranteeing its local convergence. Yet, $\alpha^{(i)}$ must be selected with caution to ensure the viability of the damped Gauss-Newton, and a few methods have been proposed, e.g., the Armijo-Goldstein step length principle. A further improvement is to introduce a stabilizing term:

$$\hat{\boldsymbol{\xi}}_k^{(i+1)} = \hat{\boldsymbol{\xi}}_k^{(i)} - \alpha^{(i)} \left[\nabla_{\boldsymbol{\xi}}^{\top} \boldsymbol{r} \left(\hat{\boldsymbol{\xi}}_k^{(i)} \right) \cdot \nabla_{\boldsymbol{\xi}} \boldsymbol{r} \left(\hat{\boldsymbol{\xi}}_k^{(i)} \right) + \delta^{(i)} \boldsymbol{D}^{(i)} \right]^{-1} \cdot \nabla_{\boldsymbol{\xi}}^{\top} \boldsymbol{r} \left(\hat{\boldsymbol{\xi}}_k^{(i)} \right) \cdot \boldsymbol{r} \left(\hat{\boldsymbol{\xi}}_k^{(i)} \right),$$

whereby the rank deficiency problem of $\left(\nabla_{\boldsymbol{\xi}}^{\top} \boldsymbol{r} \nabla_{\boldsymbol{\xi}} \boldsymbol{r} \right)$ that may appear in (11) and (15) can be avoided, given that $\delta^{(i)} > 0$ and $\boldsymbol{D}^{(i)}$ is a specified symmetric positive definite matrix. This is known as the trust region method or Levenberg-Marquardt method. For more details about Gauss-Newton-type methods, the reader is referred to (Björck, 1996). •

Remark 2. *(Generality of the Bayesian SISF algorithm).* The work (Fang et al., 2013) proves that the `SISF-wDF` algorithm, if applied to a linear system with direct feedthrough, will given the same input and state estimation as in (Gillijns and De Moor, 2007b). The method in (Gillijns and De Moor, 2007b) is based on MVUE, and development of the `SISF-wDF` algorithm provides its statistical interpretation. That is, it is statistically optimal if the following assumptions hold: (1) $\boldsymbol{x}_0 \sim \mathcal{N}(\hat{\boldsymbol{x}}_{0|0}, \boldsymbol{P}_{0|0}^{\boldsymbol{x}})$; (2) $\{\boldsymbol{w}_k\}$ and $\{\boldsymbol{v}_k\}$ are zero-mean white Gaussian; (3) $\boldsymbol{x}_0$, $\{\boldsymbol{w}_k\}$ and $\{\boldsymbol{v}_k\}$ are independent of each other; (4) $\{\boldsymbol{u}_k\}$ is white Gaussian and independent of $\boldsymbol{x}_0$, $\{\boldsymbol{w}_k\}$ and $\{\boldsymbol{v}_k\}$. Compared to the assumptions made for the classical KF, (4) is the only additional one, which ensures that the state propagation and output measurement sequences are Gaussian distributed. It is noteworthy that, even though the derivation of the method in (Gillijns and De Moor, 2007b) proceeds without imposing the assumption (A1), its statistical optimality still implicitly relies on this assumption. We note that the observations here partially reflects the fact that Bayesian estimation can offer a general framework to solve SISE problems. •

III. SISF for Systems without Direct Feedthrough

In this section, we extend the results in Section II and consider a nonlinear system described by equations of the following form:

$$\begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{w}_k, \\ \quad \boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{v}_k, \end{cases} \tag{16}$$

where no direct input-to-output feedthrough exists. In this situation, the input estimation has to be delayed by one time step, because the first measurement containing information about $\boldsymbol{u}_{k-1}$ is $\boldsymbol{y}_k$. Therefore, it is $p(\boldsymbol{u}_{k-1}, \boldsymbol{x}_k | \boldsymbol{Y}_{1:k})$ that is of interest here and should be sequentially updated. We impose the same assumption as (A1) to $\boldsymbol{u}_k$ for the system in (16), i.e., $\{\boldsymbol{u}_k\}$ is a white process independent of $\boldsymbol{x}_0$, $\{\boldsymbol{w}_k\}$ and $\{\boldsymbol{v}_k\}$. Using the Bayes' rule, we can construct a Bayesian SISF paradigm for this case. Omitting the intermediate steps,

$$p(\boldsymbol{u}_{k-1}, \boldsymbol{x}_k | \boldsymbol{Y}_{1:k}) \propto p(\boldsymbol{y}_k | \boldsymbol{x}_k) \int p(\boldsymbol{x}_k | \boldsymbol{u}_{k-1}, \boldsymbol{x}_{k-1}) \cdot p(\boldsymbol{x}_{k-1} | \boldsymbol{Y}_{1:k-1}) \mathrm{d}\boldsymbol{x}_{k-1}. \tag{17}$$

We also introduce the following assumptions in order to formulate a tractable MAP-based SISF problem:

$$\begin{aligned} p(\boldsymbol{u}_{k-1}, \boldsymbol{x}_k | \boldsymbol{Y}_k) &\sim \mathcal{N} \left(\begin{bmatrix} \hat{\boldsymbol{u}}_{k-1|k} \\ \hat{\boldsymbol{x}}_{k|k} \end{bmatrix}, \begin{bmatrix} \boldsymbol{P}_{k-1|k}^{\boldsymbol{u}} & \boldsymbol{P}_{k-1,k|k}^{\boldsymbol{ux}} \\ (\boldsymbol{P}_{k-1,k|k}^{\boldsymbol{ux}})^{\top} & \boldsymbol{P}_{k|k}^{\boldsymbol{x}} \end{bmatrix} \right), \\ p(\boldsymbol{y}_k | \boldsymbol{x}_k) &\sim \mathcal{N} \left(\boldsymbol{h}(\boldsymbol{x}_k), \boldsymbol{R}_k \right). \end{aligned}$$

From (17), a MAP cost function can be defined as done previously in (9), which is

$$\ell(\boldsymbol{u}_{k-1}, \boldsymbol{x}_k) = \delta + \boldsymbol{r}^{\top}(\boldsymbol{u}_{k-1}, \boldsymbol{x}_k) \cdot \boldsymbol{r}(\boldsymbol{u}_{k-1}, \boldsymbol{x}_k). \tag{18}$$

Here, δ is a constant and

$$\boldsymbol{r}(\boldsymbol{u}_{k-1}, \boldsymbol{x}_k) = \begin{bmatrix} \boldsymbol{R}_k^{-\frac{1}{2}} \boldsymbol{\rho}_k \\ \boldsymbol{\Pi}^{-\frac{1}{2}}(\boldsymbol{u}_{k-1}) \boldsymbol{\zeta}_k \end{bmatrix},$$

where $\boldsymbol{\rho}_k = \boldsymbol{y}_k - \boldsymbol{h}(\boldsymbol{x}_k)$, $\boldsymbol{\zeta}_k = \boldsymbol{x}_k - \boldsymbol{f}(\boldsymbol{u}_{k-1}, \hat{\boldsymbol{x}}_{k-1|k-1})$, and $\boldsymbol{\Pi}(\boldsymbol{u}_{k-1}) = \nabla_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{u}_{k-1}, \hat{\boldsymbol{x}}_{k-1|k-1}) \cdot \boldsymbol{P}^{\boldsymbol{x}}_{k-1|k-1} \cdot \nabla^{\top}_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{u}_{k-1}, \hat{\boldsymbol{x}}_{k-1|k-1}) + \boldsymbol{Q}_{k-1}$. For this problem, a numerical solution can also be developed using the Gauss-Newton method through the following iterative procedure:

$$\hat{\boldsymbol{\sigma}}_k^{(i+1)} = \hat{\boldsymbol{\sigma}}_k^{(i)} - \left[\nabla^{\top}_{\boldsymbol{\sigma}} \boldsymbol{r} \left(\hat{\boldsymbol{\sigma}}_k^{(i)} \right) \cdot \nabla_{\boldsymbol{\sigma}} \boldsymbol{r} \left(\hat{\boldsymbol{\sigma}}_k^{(i)} \right) \right]^{-1} \cdot \nabla^{\top}_{\boldsymbol{\sigma}} \boldsymbol{r} \left(\hat{\boldsymbol{\sigma}}_k^{(i)} \right) \cdot \boldsymbol{r} \left(\hat{\boldsymbol{\sigma}}_k^{(i)} \right), \tag{19}$$

where $\boldsymbol{\sigma}_k = \begin{bmatrix} \boldsymbol{u}_{k-1}^{\top} & \boldsymbol{x}_k^{\top} \end{bmatrix}^{\top}$ and $\nabla_{\boldsymbol{\sigma}} \boldsymbol{r} = \begin{bmatrix} \nabla_{\boldsymbol{u}} \boldsymbol{r} & \nabla_{\boldsymbol{x}} \boldsymbol{r} \end{bmatrix}$. Here, one can set the initial condition as $\hat{\boldsymbol{\sigma}}_k^{(0)} = \boldsymbol{0}$, run the procedure iteratively, and finally assign the obtained $\hat{\boldsymbol{\sigma}}_k^{(i_{\max})}$ to $\hat{\boldsymbol{u}}_{k-1|k}$ and $\hat{\boldsymbol{x}}_{k|k}$.

Proceeding further, the associated estimation error covariance matrix can be computed by evaluating the Fisher information matrix at $\hat{\boldsymbol{u}}_{k-1|k}$ and $\hat{\boldsymbol{x}}_{k|k}$, i.e.,

$$\begin{bmatrix} \boldsymbol{P}^{\boldsymbol{u}}_{k-1|k} & \boldsymbol{P}^{\boldsymbol{ux}}_{k-1,k|k} \\ \left(\boldsymbol{P}^{\boldsymbol{ux}}_{k-1,k|k} \right)^{\top} & \boldsymbol{P}^{\boldsymbol{x}}_{k|k} \end{bmatrix} = \boldsymbol{\mathcal{F}}^{-1}(\hat{\boldsymbol{u}}_{k-1|k}, \hat{\boldsymbol{x}}_{k|k}), \tag{20}$$

where the definition of $\boldsymbol{\mathcal{F}}$ is identical to (13). Fully determining each block of $\boldsymbol{\mathcal{F}}$ entails the computation as below.

The l-th column of the gradient matrix of $\boldsymbol{r}$ with respect to (w.r.t.) $\boldsymbol{u}$ is given by,

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}_l} = \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{\Pi}^{-\frac{1}{2}} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}_l} - \frac{1}{2} \boldsymbol{\Pi}^{\frac{1}{2}} \boldsymbol{\Pi}^{-1} \frac{\partial \boldsymbol{\Pi}}{\partial \boldsymbol{u}_l} \boldsymbol{\Pi}^{-1} \boldsymbol{\zeta} \end{bmatrix}.$$

The following relation is used here:

$$\frac{\partial \boldsymbol{X}^{\frac{1}{2}}}{\partial \tau} = -\frac{1}{2} \boldsymbol{X}^{\frac{1}{2}} \boldsymbol{X}^{-1} \frac{\partial \boldsymbol{X}}{\partial \tau} \boldsymbol{X}^{-1},$$

where $\boldsymbol{X}$ is a symmetric positive definite matrix dependent on τ (Spinello and Stilwell, 2010). The l-th column of the gradient matrix of $\boldsymbol{r}$ w.r.t. x is

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}_l} = \begin{bmatrix} -\boldsymbol{R}^{-\frac{1}{2}} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_l} \\ \boldsymbol{\Pi}^{-\frac{1}{2}} \boldsymbol{e}_l \end{bmatrix}, \quad \nabla_{\boldsymbol{x}} \boldsymbol{r} = \begin{bmatrix} -\boldsymbol{R}^{-\frac{1}{2}} \nabla_{\boldsymbol{x}} \boldsymbol{h} \\ \boldsymbol{\Pi}^{-\frac{1}{2}} \end{bmatrix},$$

where $\boldsymbol{e}_l$ is the standard basis vector with a 1 in the l-th element and 0's elsewhere. The lj-th entries of $\nabla^{\top}_{\boldsymbol{u}} \boldsymbol{r} \nabla_{\boldsymbol{u}} \boldsymbol{r}$, $\nabla^{\top}_{\boldsymbol{u}} \boldsymbol{r} \nabla_{\boldsymbol{x}} \boldsymbol{r}$ and $\nabla^{\top}_{\boldsymbol{x}} \boldsymbol{r} \nabla_{\boldsymbol{x}} \boldsymbol{r}$ are expressed as, respectively,

$$\begin{aligned}
\frac{\partial \boldsymbol{r}^{\top}}{\partial \boldsymbol{u}_l} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}_j} &= \frac{\partial \boldsymbol{f}^{\top}}{\partial \boldsymbol{u}_l} \Pi^{-1} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}_j} + \frac{1}{2} \boldsymbol{\zeta}^{\top} \boldsymbol{\Pi}^{-1} \cdot \left(\frac{\partial \boldsymbol{\Pi}}{\partial \boldsymbol{u}_l} \boldsymbol{\Pi}^{-1} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}_j} + \frac{\partial \boldsymbol{\Pi}}{\partial \boldsymbol{u}_j} \boldsymbol{\Pi}^{-1} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}_l} \right) + \frac{1}{4} \boldsymbol{\zeta}^{\top} \boldsymbol{\Pi}^{-1} \frac{\partial \boldsymbol{\Pi}}{\partial \boldsymbol{u}_l} \boldsymbol{\Pi}^{-1} \frac{\partial \boldsymbol{\Pi}}{\partial \boldsymbol{u}_j} \Pi^{-1} \boldsymbol{\zeta}, \\
\frac{\partial \boldsymbol{r}^{\top}}{\partial \boldsymbol{u}_l} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}_j} &= -\frac{\partial \boldsymbol{f}^{\top}}{\partial \boldsymbol{u}_l} \boldsymbol{\Pi}^{-1} \boldsymbol{e}_j - \frac{1}{2} \boldsymbol{\zeta}^{\top} \boldsymbol{\Pi}^{-1} \frac{\partial \boldsymbol{\Pi}}{\partial \boldsymbol{u}_l} \boldsymbol{\Pi}^{-1} \boldsymbol{e}_j, \\
\frac{\partial \boldsymbol{r}^{\top}}{\partial \boldsymbol{x}_l} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}_j} &= \frac{\partial \boldsymbol{h}^{\top}}{\partial \boldsymbol{x}_l} \boldsymbol{R}^{-1} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_j} + \boldsymbol{e}_l^{\top} \boldsymbol{\Pi}^{-1} \boldsymbol{e}_j, \\
\nabla^{\top}_{\boldsymbol{x}} \boldsymbol{r} \nabla_{\boldsymbol{x}} \boldsymbol{r} &= \nabla^{\top}_{\boldsymbol{x}} \boldsymbol{h} \boldsymbol{R}^{-1} \nabla_{\boldsymbol{x}} \boldsymbol{h} + \boldsymbol{\Pi}^{-1}.
\end{aligned}$$

Then, we have

$$\begin{aligned}
\frac{\partial \ell}{\partial \boldsymbol{u}_l} &= \boldsymbol{r}^{\top} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}_l} = -\boldsymbol{\zeta}^{\top} \boldsymbol{\Pi}^{-1} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}_l} - \frac{1}{2} \boldsymbol{\zeta}^{\top} \boldsymbol{\Pi}^{-1} \frac{\partial \boldsymbol{\Pi}}{\partial \boldsymbol{u}_l} \boldsymbol{\Pi}^{-1} \boldsymbol{\zeta}, \\
\frac{\partial \ell}{\partial \boldsymbol{x}_l} &= \boldsymbol{r}^{\top} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}_l} = -\boldsymbol{\rho}^{\top} \boldsymbol{R}^{-1} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_l} + \boldsymbol{\zeta}^{\top} \boldsymbol{\Pi}^{-1} \boldsymbol{e}_l, \\
\nabla_{\boldsymbol{x}} \ell &= -\boldsymbol{\rho}^{\top} \boldsymbol{R}^{-1} \nabla_{\boldsymbol{x}} \boldsymbol{h} + \boldsymbol{\zeta}^{\top} \boldsymbol{\Pi}^{-1}.
\end{aligned}$$

initialize: $k=0$, $\hat{\boldsymbol{\sigma}}_0=\mathrm{E}(\boldsymbol{\sigma}_0)$, $\boldsymbol{P}_0^{\sigma}=p_0\boldsymbol{I}$, where p_0 is a large positive value
repeat
  $k\leftarrow k+1$
  initialize: $i=0$, $\hat{\boldsymbol{\sigma}}_k^{(0)}=\boldsymbol{0}$
  while $i<i_{\max}$ **do**
    perform Gauss-Newton based joint input and state filtering *via* (19)
    $i\leftarrow i+1$
  end while
  export $\hat{\boldsymbol{\sigma}}_k^{(i_{\max})}$ and assign to $\hat{\boldsymbol{u}}_{k-1|k}$ and $\hat{\boldsymbol{x}}_{k|k}$
  compute joint filtering error covariance *via* (20)
until no more measurements arrive

TABLE II: The `SISF-w/oDF` algorithm (SISF for systems without direct feedthrough).

To compute the Fisher information matrix, $\mathrm{E}\left(\nabla_{\boldsymbol{u}}^{\top}\ell\nabla_{\boldsymbol{u}}\ell\right)$, $\mathrm{E}\left(\nabla_{\boldsymbol{u}}^{\top}\ell\nabla_{\boldsymbol{x}}\ell\right)$ and $\mathrm{E}\left(\nabla_{\boldsymbol{x}}^{\top}\ell\nabla_{\boldsymbol{x}}\ell\right)$ are needed. Their lj-th entries are

$$\begin{aligned}
\mathrm{E}\left(\frac{\partial\ell^{\top}}{\partial\boldsymbol{u}_l}\frac{\partial\ell}{\partial\boldsymbol{u}_j}\right)&=\frac{\partial\boldsymbol{f}^{\top}}{\partial\boldsymbol{u}_l}\boldsymbol{\Pi}^{-1}\frac{\partial\boldsymbol{f}}{\partial\boldsymbol{u}_j}+\frac{1}{4}\mathrm{tr}\left(\frac{\partial\boldsymbol{\Pi}}{\partial\boldsymbol{u}_l}\Pi^{-1}\frac{\partial\boldsymbol{\Pi}}{\partial\boldsymbol{u}_j}\boldsymbol{\Pi}^{-1}\right),\\
\mathrm{E}\left(\frac{\partial\ell^{\top}}{\partial\boldsymbol{u}_l}\frac{\partial\ell}{\partial\boldsymbol{x}_j}\right)&=-\frac{\partial\boldsymbol{f}^{\top}}{\partial\boldsymbol{u}_l}\boldsymbol{\Pi}^{-1}\boldsymbol{e}_j,\\
\mathrm{E}\left(\frac{\partial\ell^{\top}}{\partial\boldsymbol{x}_l}\frac{\partial\ell}{\partial\boldsymbol{x}_j}\right)&=\frac{\partial\boldsymbol{h}^{\top}}{\partial\boldsymbol{x}_l}\boldsymbol{\Pi}^{-1}\frac{\partial\boldsymbol{h}}{\partial\boldsymbol{x}_j}+\boldsymbol{e}_l^{\top}\boldsymbol{\Pi}^{-1}\boldsymbol{e}_j,\\
\mathrm{E}\left(\nabla_{\boldsymbol{x}}^{\top}\ell\nabla_{\boldsymbol{x}}\ell\right)&=\nabla_{\boldsymbol{x}}^{\top}\boldsymbol{h}\boldsymbol{R}^{-1}\nabla_{\boldsymbol{x}}\ell+\boldsymbol{\Pi}^{-1}.
\end{aligned}$$

With the above derivation, we have fully developed the SISF algorithm for the system in (16), which is named `SISF-w/oDF` and summarized in Table II. Note that, if applied to a linear system, the `SISF-w/oDF` algorithm will reduce to (Gillijns and De Moor, 2007a), which offers a MVUE-based SISF algorithm for linear systems without direct feedthrough. Or in other words, if the conditions (1)-(4) proposed in Remark 2 are also valid for the linear version of (16), the algorithm in (Gillijns and De Moor, 2007a) can be directly derived using the Bayesian paradigm along with MAP estimation. In addition, (Gillijns and De Moor, 2007a) gives the same state update as (Kitanidis, 1987; Darouach and Zasadzinski, 1997) and the same input update as (Hsieh, 2000). This suggests that these methods can be regarded as special cases of the `SISF-w/oDF` algorithm.

IV. SISS for Systems with and without Direct Feedthrough

Our results above are concerned with the filtering problem, where $\boldsymbol{u}_k$ (or $\boldsymbol{u}_{k-1}$) and $\boldsymbol{x}_k$ are estimated based on the measurement $\boldsymbol{Y}_{1:k}$. Another interesting problem is to make the estimation when all the measurements are available. Suppose that the total time is N. Then, the question is to make an estimate of $\boldsymbol{u}_k$ (or $\boldsymbol{u}_{k-1}$) and $\boldsymbol{x}_k$ using $\boldsymbol{Y}_{1:N}$. This is a problem of smoothing or SISS as termed before, which is the theme of this section. Following a similar structure as Sections II and III, we will first investigate Bayesian SISS for systems with direct feedthrough and then move forward to those without it. Here, we would like to point out that developing effective and efficient enough SISS algorithms for nonlinear systems without direct feedthrough is rather difficult, because of the asynchronous coupling between $\boldsymbol{u}_{k-1}$ and $\boldsymbol{x}_k$ and the nonlinearity involved. We thus focus on only linear systems in this case.

A. SISS for Nonlinear Systems with Direct Feedthrough

Consider the nonlinear system in (1). Section II discusses SISF for (1) from the viewpoint of determining $p(\boldsymbol{u}_k,\boldsymbol{x}_k|\boldsymbol{Y}_{1:k})$, or more specifically, the passing from $p(\boldsymbol{u}_{k-1},\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{1:k-1})$ to $p(\boldsymbol{u}_k,\boldsymbol{x}_k|\boldsymbol{Y}_{1:k})$. Here, we shift

our attention to $p(\boldsymbol{u}_k, \boldsymbol{x}_k|\boldsymbol{Y}_{1:N})$ in order to achieve fixed-interval SISS. Because the reader has been familiar with the notation, we use $\boldsymbol{\xi}_k = \left[\boldsymbol{u}_k^\top \ \boldsymbol{x}_k^\top\right]^\top$ to improve notational simplicity. Applying the Bayes' rule, we can build the Bayesian SISS paradigm for (1), which unveils the backward recursion of $p(\boldsymbol{\xi}_k|\boldsymbol{Y}_{1:N})$ from $p\left(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:N}\right)$:

$$p(\boldsymbol{\xi}_k|\boldsymbol{Y}_{1:N}) = p(\boldsymbol{\xi}_k|\boldsymbol{Y}_{1:k}) \int \frac{p(\boldsymbol{x}_{k+1}|\boldsymbol{\xi}_k) \cdot p\left(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:N}\right)}{p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k})} \mathrm{d}\boldsymbol{x}_{k+1}. \tag{21}$$

Here, we demonstrate the derivation. First,

$$p(\boldsymbol{\xi}_k|\boldsymbol{Y}_{1:N}) = \int p(\boldsymbol{\xi}_k, \boldsymbol{\xi}_{k+1}|\boldsymbol{Y}_{1:N}) \mathrm{d}\boldsymbol{\xi}_{k+1} = \int p\left(\boldsymbol{\xi}_k|\boldsymbol{\xi}_{k+1}, \boldsymbol{Y}_{1:N}\right) \cdot p\left(\boldsymbol{\xi}_{k+1}|\boldsymbol{Y}_{1:N}\right) \mathrm{d}\boldsymbol{\xi}_{k+1}.$$

Note that $\boldsymbol{\xi}_k$ is conditionally independent of $\boldsymbol{Y}_{k+1:N}$ and $\boldsymbol{u}_{k+1}$ given $\boldsymbol{x}_{k+1}$, due to the Markovian state propagation and (A1). Hence, $p\left(\boldsymbol{\xi}_k|\boldsymbol{\xi}_{k+1}, \boldsymbol{Y}_{1:N}\right) = p\left(\boldsymbol{\xi}_k|\boldsymbol{x}_{k+1}, \boldsymbol{Y}_{1:k}\right)$. Then, we have

$$\begin{aligned} p(\boldsymbol{\xi}_k|\boldsymbol{Y}_{1:N}) &= \int p\left(\boldsymbol{\xi}_k|\boldsymbol{x}_{k+1}, \boldsymbol{Y}_{1:k}\right) \cdot p\left(\boldsymbol{\xi}_{k+1}|\boldsymbol{Y}_{1:N}\right) \mathrm{d}\boldsymbol{\xi}_{k+1} \\ &= \int p\left(\boldsymbol{\xi}_k|\boldsymbol{x}_{k+1}, \boldsymbol{Y}_{1:k}\right) \cdot p\left(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:N}\right) \mathrm{d}\boldsymbol{x}_{k+1}. \end{aligned} \tag{22}$$

Meanwhile, we have

$$p\left(\boldsymbol{\xi}_k|\boldsymbol{x}_{k+1}, \boldsymbol{Y}_{1:k}\right) = \frac{p(\boldsymbol{\xi}_k, \boldsymbol{x}_{k+1}, \boldsymbol{Y}_{1:k})}{p(\boldsymbol{x}_{k+1}, \boldsymbol{Y}_{1:k})} = \frac{p(\boldsymbol{x}_{k+1}|\boldsymbol{\xi}_k, \boldsymbol{Y}_{1:k}) \cdot p(\boldsymbol{\xi}_k|\boldsymbol{Y}_{1:k})}{p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k})} = \frac{p(\boldsymbol{x}_{k+1}|\boldsymbol{\xi}_k) \cdot p(\boldsymbol{\xi}_k|\boldsymbol{Y}_{1:k})}{p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k})}. \tag{23}$$

Inserting (23) into (22), we can obtain (21).

The Bayesian paradigm in (21) is an input and state smoother in a statistical sense, which illustrates the backward update of $\boldsymbol{\xi}_k$ given $\boldsymbol{Y}_{1:N}$. However, a direct analytical evaluation of the pdf's is known to be quite difficult, if not impossible, for nonlinear systems. Hence, we will formulate an MAP estimation problem based on (21) and seek a numerical solution. To proceed further, we make Gaussian distribution assumptions for the following pdf's:

(A5) $p(\boldsymbol{x}_{k+1}|\boldsymbol{\xi}_k) \sim \mathcal{N}(\boldsymbol{f}(\boldsymbol{\xi}_k), \boldsymbol{Q}_k)$;
(A6) $p(\boldsymbol{\xi}_k|\boldsymbol{Y}_{1:k}) \sim \mathcal{N}\left(\hat{\boldsymbol{\xi}}_{k|k}, \boldsymbol{P}^{\boldsymbol{\xi}}_{k|k}\right)$;
(A7) $p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:N}) \sim \mathcal{N}\left(\hat{\boldsymbol{x}}_{k+1|N}, \boldsymbol{P}^{\boldsymbol{x}}_{k+1|N}\right)$;
(A8) $p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k}) \sim \mathcal{N}\left(\hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{P}^{\boldsymbol{x}}_{k+1|k}\right)$.

Here, $\hat{\boldsymbol{\xi}}_{k|k}$ is the filtered estimate of $\boldsymbol{\xi}_k$ given $\boldsymbol{Y}_{1:k}$, $\boldsymbol{P}^{\boldsymbol{\xi}}_{k|k}$ is the filtering error covariance, $\hat{\boldsymbol{x}}_{k+1|N}$ is the smoothed estimate of $\boldsymbol{x}_{k+1}$ given $\boldsymbol{Y}_{1:N}$, and $\boldsymbol{P}^{\boldsymbol{x}}_{k+1|N}$ is the smoothing error covariance. These assumptions are made to bridge the gap from the Bayesian paradigm in (21) to an executable smoother.

We now consider developing an MAP-based smoother to estimate $\boldsymbol{u}_k$ and $\boldsymbol{x}_k$ *via* maximizing $p(\boldsymbol{\xi}_k|\boldsymbol{Y}_{1:N})$. The smoother then can be expressed as:

$$\hat{\boldsymbol{\xi}}_{k|N} = \arg\max_{\boldsymbol{\xi}_k} p(\boldsymbol{\xi}_k|\boldsymbol{Y}_{1:N}). \tag{24}$$

The above maximization of $p(\boldsymbol{\xi}_k|\boldsymbol{Y}_{1:N})$ can be transformed into the following problem of minimizing a cost function with assistance of the assumptions (A5)-(A8), which is given as:

$$\hat{\boldsymbol{\xi}}_{k|N} = \arg\min_{\boldsymbol{\xi}_k} \ell(\boldsymbol{\xi}_k), \tag{25}$$

where

$$\ell(\boldsymbol{\xi}_k) := \left(\boldsymbol{\xi}_k - \hat{\boldsymbol{\xi}}_{k|k}\right)^\top \left(\boldsymbol{P}^{\boldsymbol{\xi}}_{k|k}\right)^{-1} \left(\boldsymbol{\xi}_k - \hat{\boldsymbol{\xi}}_{k|k}\right) + \left(\boldsymbol{f}(\boldsymbol{\xi}_k) - \boldsymbol{\delta}_k\right)^\top \boldsymbol{\Delta}_k^{-1} \left(\boldsymbol{f}(\boldsymbol{\xi}_k) - \boldsymbol{\delta}_k\right), \tag{26a}$$

$$\boldsymbol{\Delta}_k = \left[\left(\boldsymbol{P}^{\boldsymbol{x}}_{k+1|N}\right)^{-1} - \left(\boldsymbol{P}^{\boldsymbol{x}}_{k+1|k}\right)^{-1}\right]^{-1} + \boldsymbol{Q}_k, \tag{26b}$$

$$\boldsymbol{\delta}_k = \left[\left(\boldsymbol{P}^{\boldsymbol{x}}_{k+1|N}\right)^{-1} - \left(\boldsymbol{P}^{\boldsymbol{x}}_{k+1|k}\right)^{-1}\right]^{-1} \cdot \left[\left(\boldsymbol{P}^{\boldsymbol{x}}_{k+1|N}\right)^{-1} \hat{\boldsymbol{x}}_{k+1|N} - \left(\boldsymbol{P}^{\boldsymbol{x}}_{k+1|k}\right)^{-1} \hat{\boldsymbol{x}}_{k+1|k}\right]. \tag{26c}$$

For a detailed derivation of the above, an interested reader is referred to (Fang et al., 2015). Furthermore, the sum of the weighted 2-norms in (26) can be rewritten as

$$\ell(\boldsymbol{\xi}_k) = \boldsymbol{r}^\top(\boldsymbol{\xi}_k) \cdot \boldsymbol{r}(\boldsymbol{\xi}_k), \tag{27}$$

where

$$\boldsymbol{r}(\boldsymbol{\xi}_k) = \begin{bmatrix} \left(\boldsymbol{P}^{\boldsymbol{\xi}}_{k|k}\right)^{-\frac{1}{2}} \left(\boldsymbol{\xi}_k - \hat{\boldsymbol{\xi}}_{k|k}\right) \\ \boldsymbol{\Delta}_k^{-\frac{1}{2}} \left(\boldsymbol{f}(\boldsymbol{\xi}_k) - \boldsymbol{\delta}_k\right) \end{bmatrix}.$$

This would allow the use of the classical Gauss-Newton method, which performs an iterative searching process that linearizes around the current arrival point, determines the best search direction and then moves forward to the next point. Specifically, we have

$$\hat{\boldsymbol{\xi}}^{(i+1)}_{k|N} = \hat{\boldsymbol{\xi}}^{(i)}_{k|N} - \left[\nabla^\top_{\boldsymbol{\xi}} \boldsymbol{r}\left(\hat{\boldsymbol{\xi}}^{(i)}_{k|N}\right) \cdot \nabla_{\boldsymbol{\xi}} \boldsymbol{r}\left(\hat{\boldsymbol{\xi}}^{(i)}_{k|N}\right)\right]^{-1} \cdot \nabla^\top_{\boldsymbol{\xi}} \boldsymbol{r}\left(\hat{\boldsymbol{\xi}}^{(i)}_{k|N}\right) \cdot \boldsymbol{r}\left(\hat{\boldsymbol{\xi}}^{(i)}_{k|N}\right), \tag{28}$$

where (i) denotes the iteration number, and

$$\nabla_{\boldsymbol{\xi}} \boldsymbol{r}(\boldsymbol{\xi}_k) = \begin{bmatrix} \left(\boldsymbol{P}^{\boldsymbol{\xi}}_{k|k}\right)^{-\frac{1}{2}} \\ \boldsymbol{\Delta}_k^{-\frac{1}{2}} \nabla_{\boldsymbol{\xi}} \boldsymbol{f}(\boldsymbol{\xi}_k) \end{bmatrix}.$$

One can let $\hat{\boldsymbol{\xi}}_{k|N} = \hat{\boldsymbol{\xi}}^{(i_{\max})}_{k|N}$, where $i_{\max}$ is the maximum number of iterations. The Fisher information matrix $\boldsymbol{\mathcal{F}}$ for (36) is approximately given by

$$\boldsymbol{\mathcal{F}}(\boldsymbol{\xi}_k) = \left(\boldsymbol{P}^{\boldsymbol{\xi}}_{k|k}\right)^{-1} + \nabla^\top_{\boldsymbol{\xi}} \boldsymbol{f}(\boldsymbol{\xi}_k) \boldsymbol{\Delta}_k^{-1} \nabla_{\boldsymbol{\xi}} \boldsymbol{f}(\boldsymbol{\xi}_k).$$

Evaluating $\boldsymbol{\mathcal{F}}$ at $\hat{\boldsymbol{\xi}}_{k|N}$ and inverting it will lead to the error covariance, i.e.,

$$\boldsymbol{P}^{\boldsymbol{\xi}}_{k|N} = \boldsymbol{\mathcal{F}}^{-1}(\hat{\boldsymbol{\xi}}_{k|N}) = \left[\left(\boldsymbol{P}^{\boldsymbol{\xi}}_{k|k}\right)^{-1} + \nabla^\top_{\boldsymbol{\xi}} \boldsymbol{f}(\hat{\boldsymbol{\xi}}_{k|N}) \boldsymbol{\Delta}_k^{-1} \nabla_{\boldsymbol{\xi}} \boldsymbol{f}(\hat{\boldsymbol{\xi}}_{k|N})\right]^{-1}. \tag{29}$$

We say that (28)-(29) are the backward smoothing equations for input and state estimation. The corresponding forward filtering equations are given by the `SISF-wDF` algorithm. The above nonlinear SISS algorithm can be readily specialized to the linear case. Consider a linear system of the following form

$$\begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{F}_k \boldsymbol{x}_k + \boldsymbol{G}_k \boldsymbol{u}_k + \boldsymbol{w}_k, \\ \boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{J}_k \boldsymbol{u}_k + \boldsymbol{v}_k. \end{cases} \tag{30}$$

Then, the smoother in (28)-(29) can find the best input and state estimates in a single iteration, which leads to a linear SISS algorithm. It can be proven that the algorithm can be expressed as

$$\hat{\boldsymbol{\xi}}_{k|N} = \hat{\boldsymbol{\xi}}_{k|k} + \boldsymbol{K}_k(\hat{\boldsymbol{x}}_{k+1|N} - \bar{\boldsymbol{F}}_k \hat{\boldsymbol{\xi}}_{k|k}), \tag{31a}$$

$$\boldsymbol{K}_k = \boldsymbol{P}^{\boldsymbol{\xi}}_{k|k} \bar{\boldsymbol{F}}^\top_k \left(\boldsymbol{P}^{\boldsymbol{x}}_{k+1|k}\right)^{-1}, \tag{31b}$$

where $\bar{\boldsymbol{F}}_k = [\boldsymbol{G}_k \ \boldsymbol{F}_k]$. The associated smoothing error covariance is given by

$$\boldsymbol{P}^{\boldsymbol{\xi}}_{k|N} = \boldsymbol{P}^{\boldsymbol{\xi}}_{k|k} + \boldsymbol{K}_k(\boldsymbol{P}^{\boldsymbol{x}}_{k+1|N} - \boldsymbol{P}^{\boldsymbol{x}}_{k+1|k}) \boldsymbol{K}^\top_k. \tag{32}$$

To sum up, we call the obtained forward-backward smoothing algorithm as `SISS-wDF` and summarize it in Table III.

Forward filtering:
for $k = 1$ to N **do**
  compute filtered input and state estimate *via* the `SISF-wDF` algorithm
end for

Backward smoothing:
for $k = N-1$ to 1 **do**
  initialize: $i = 0$, $\hat{\boldsymbol{\xi}}_{k|N}^{(0)} = \hat{\boldsymbol{\xi}}_{k|k}$
  while $i < i_{\max}$ **do**
    perform Gauss-Newton-based joint input and state smoothing *via* (28) (or (31a)-(31b) for linear systems)
    $i \leftarrow i + 1$
  end while
  export $\hat{\boldsymbol{u}}_{k|N}$ and $\hat{\boldsymbol{x}}_{k|N}$ from $\hat{\boldsymbol{\xi}}_{k|N}^{(i_{\max})}$
  compute joint smoothing error covariance *via* (29) (or (32) for linear systems)
end for

TABLE III: The `SISS-wDF` algorithm: SISS for systems with direct feedthrough.

B. SISS for Systems without Direct Feedthrough

Now, let us consider Bayesian SISS for systems without direct feedthrough, which are shown in (16). In this case, we use the notation $\boldsymbol{\sigma}_k = \left[\boldsymbol{u}_{k-1}^\top \ \boldsymbol{x}_k^\top\right]^\top$. Given the assumption (A1), the Bayesian smoothing paradigm for input and state estimation is given by (Fang et al., 2015):

$$p(\boldsymbol{\sigma}_k|\boldsymbol{Y}_{1:N}) = p(\boldsymbol{\sigma}_k|\boldsymbol{Y}_{1:k}) \iint \frac{p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k,\boldsymbol{u}_k)\cdot p(\boldsymbol{\sigma}_{k+1}|\boldsymbol{Y}_{1:N})}{\int p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k,\boldsymbol{u}_k)\cdot p(\boldsymbol{x}_k|\boldsymbol{Y}_{1:k})\mathrm{d}\boldsymbol{x}_k}\mathrm{d}\boldsymbol{u}_k\mathrm{d}\boldsymbol{x}_{k+1}. \tag{33}$$

Our research shows that it is challenging to derive a viable SISS method based on (33) for a nonlinear system of general form as shown in (16). This is because $\boldsymbol{u}_{k-1}$ and $\boldsymbol{x}_k$ are asynchronous in time with a gap of one time step between them. The asynchronous coupling complicates the nonlinear relationship between $\boldsymbol{u}_{k-1}$, $\boldsymbol{x}_k$ and the output measurements, which is only exacerbated further in the backward smoothing scenario. Hence, we instead constrain our scope to SISS for linear systems. Consider

$$\begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{F}_k\boldsymbol{x}_k + \boldsymbol{G}_k\boldsymbol{u}_k + \boldsymbol{w}_k, \\ \quad \boldsymbol{y}_k = \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{v}_k. \end{cases} \tag{34}$$

For (34), the MAP smoother $\boldsymbol{\xi}_{k|N} = \arg\max_{\boldsymbol{\xi}_k} p(\boldsymbol{\xi}_k|\boldsymbol{Y}_{1:N})$ can be converted into the following problem:

$$\hat{\boldsymbol{\sigma}}_{k|N} = \arg\min_{\boldsymbol{\sigma}_k} \ell(\boldsymbol{\sigma}_k), \tag{35}$$

where

$$\ell(\boldsymbol{\sigma}_k) := \left(\boldsymbol{\sigma}_k - \hat{\boldsymbol{\sigma}}_{k|k}\right)^\top \left(\boldsymbol{P}_{k|k}^{\boldsymbol{\sigma}}\right)^{-1} \left(\boldsymbol{\sigma}_k - \hat{\boldsymbol{\sigma}}_{k|k}\right) + \left(\boldsymbol{F}_k\boldsymbol{N}\boldsymbol{\sigma}_k - \boldsymbol{\delta}_k\right)^\top \boldsymbol{\Delta}_k^{-1}\left(\boldsymbol{F}_k\boldsymbol{N}\boldsymbol{\sigma}_k - \boldsymbol{\delta}_k\right), \tag{36a}$$

$$\boldsymbol{M}_k = \begin{bmatrix} -\boldsymbol{G}_k & \boldsymbol{I} \end{bmatrix}, \quad \boldsymbol{N} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \tag{36b}$$

$$\boldsymbol{\Delta}_k = \left[\left(\boldsymbol{M}_k\boldsymbol{P}_{k+1|N}^{\boldsymbol{\sigma}}\boldsymbol{M}_k^\top\right)^{-1} - \left(\boldsymbol{P}_{k+1|k}^{\boldsymbol{x}*}\right)^{-1}\right]^{-1} + \boldsymbol{Q}_k \tag{36c}$$

$$\boldsymbol{\delta}_k = \boldsymbol{F}_k\hat{\boldsymbol{x}}_{k|k} - \boldsymbol{P}_{k+1|k}^{\boldsymbol{x}*}\left(\boldsymbol{M}_k\boldsymbol{P}_{k+1|N}^{\boldsymbol{\sigma}}\boldsymbol{M}_k^\top - \boldsymbol{P}_{k+1|k}^{\boldsymbol{x}*}\right)^{-1}\left(\boldsymbol{M}_k\hat{\boldsymbol{\sigma}}_{k+1|N} - \boldsymbol{F}_k\hat{\boldsymbol{x}}_{k|k}\right), \tag{36d}$$

$$\boldsymbol{P}_{k+1|k}^{\boldsymbol{x}*} = \boldsymbol{F}_k\boldsymbol{P}_{k|k}^{\boldsymbol{x}}\boldsymbol{F}_k^\top + \boldsymbol{Q}_k. \tag{36e}$$

The solution to (35) can then be derived as follows:

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k}^{\boldsymbol{\sigma}}\boldsymbol{N}^\top\boldsymbol{F}_k^\top\left(\boldsymbol{P}_{k+1|k}^{\boldsymbol{x}*}\right)^{-1}, \tag{37a}$$

Forward filtering:
for $k = 1$ to N **do**
compute filtered input and state estimate *via* the `SISF-w/oDF` algorithm or the algorithm in (Gillijns and De Moor, 2007a)
end for

Backward smoothing:
for $k = N - 1$ to 1 **do**
perform joint input and state smoothing *via* (37a)-(37b)
export $\hat{\boldsymbol{u}}_{k-1|N}$ and $\hat{\boldsymbol{x}}_{k|N}$ from $\hat{\boldsymbol{\sigma}}_{k|N}$
compute joint smoothing error covariance *via* (37c)
end for

TABLE IV: The `SISS-w/oDF` algorithm: SISS for linear systems without direct feedthrough.

$$\hat{\boldsymbol{\sigma}}_{k|N} = \hat{\boldsymbol{\sigma}}_{k|k} + \boldsymbol{K}_k \left(\boldsymbol{M}_k \hat{\boldsymbol{\sigma}}_{k+1|N} - \boldsymbol{F}_k \hat{\boldsymbol{x}}_{k|k} \right), \tag{37b}$$

$$\boldsymbol{P}^{\boldsymbol{\sigma}}_{k|N} = \boldsymbol{P}^{\boldsymbol{\sigma}}_{k|k} + \boldsymbol{K}_k \left[\boldsymbol{M}_k \boldsymbol{P}^{\boldsymbol{\sigma}}_{k+1|N} \boldsymbol{M}_k^\top - \left(\boldsymbol{P}^{\boldsymbol{x}*}_{k+1|k} \right) \right]^{-1} \boldsymbol{K}_k^\top. \tag{37c}$$

In above, we formulate and solve an MAP-optimization-based SISS problem for linear systems without direct feedthrough. The backward smoothing solution is offered in (37a)-(37c), and its associated forward filtering can be accomplished by running the `SISF-w/oDF` algorithm, which only requires a single iteration. Please note that application of the `SISF-w/oDF` algorithm to linear systems leads to the joint input and state estimator proposed in (Gillijns and De Moor, 2007a), as mentioned in Section III. Finally, we outline the obtained SISS algorithm, named `SISS-w/oDF`, in Table. IV.

V. Application Example

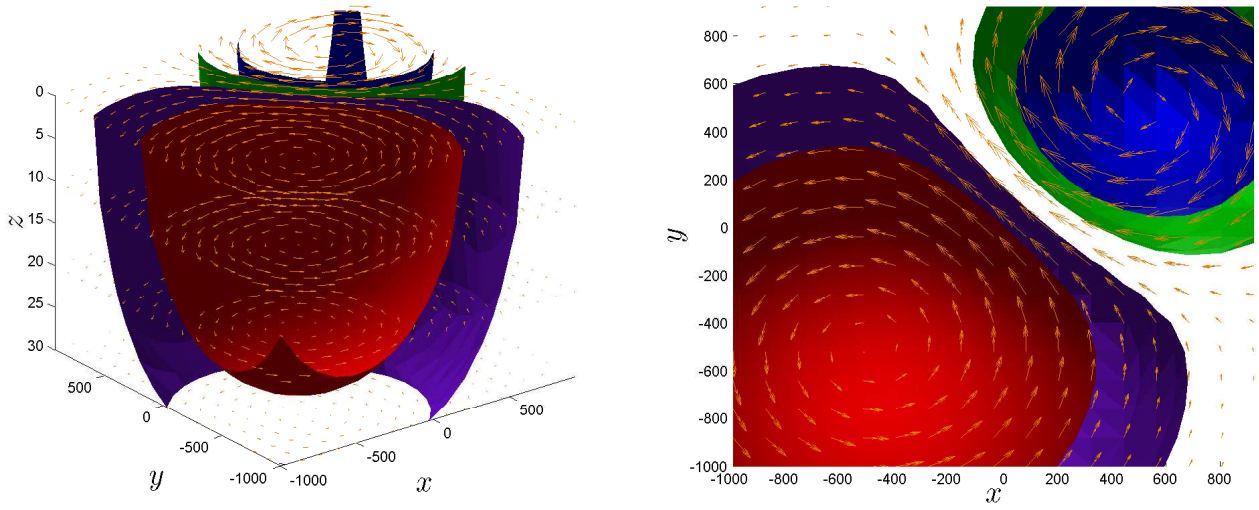
Ocean flow field reconstruction has been a research subject of intense interest for its vital role in helping oceanographers understand the oceans. Flows are known to be crucial for fishing, shipping, navigation, weather forecasting, environmental monitoring and climate change. To study these flows, consider a swarm of inexpensive buoyancy-controlled profiling floats acting as an ocean observing system (Colgan, 2006; Ouimet and Cortes, 2014; Han et al., 2010), deployed to traverse a region of the ocean. The floats are capable of arbitrary vertical migration while traveling along the flows. During the travel, each float measures and stores a time record of its depth, acceleration, position and some oceanographic quantities such as temperature and salinity. The data record is sent to a central server for processing when the float comes up to the surface. Here, we only consider a three-dimensional flow domain (see Fig. 2), the space of which is occupied mainly by two adjacent eddies of opposite direction of rotation.

A. Float Dynamics

For simplicity, we only examine a float's motion along the x-direction. The same results can apply to y-direction due to the independence of perpendicular components of motion. For a float, the flow velocity $v(d_x, z)$ at its x-displacement d_x is time-stationary and dependent only on its depth z. The dynamics of a float is described in (Booth, 1981):

$$m\ddot{d}_x = c \cdot \operatorname{sign}\left(v(d_x, z) - \dot{d}_x \right) \cdot \left(v(d_x, z) - \dot{d}_x \right)^2, \tag{38}$$

where m is the constant rigid mass and c the drag parameter. The right hand side of the above equation represents the drag force that quantifies the resistance exercised on the profiling float in the flow field.



(a) (b)

Fig. 2: (a) The three-dimensional flow field; (b) top view of the flow field.

From (38), we define two state variables $x_1 := d_x$ and $x_2 := \dot{d}_x$. Further, $v(d_x, z)$ can be viewed as the unknown external input into the float, naturally implying the definition of $u := v(d_x, z)$. Then (38) can be rewritten as,

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= \frac{c}{m} \cdot \operatorname{sign}(u - x_2) \cdot (u - x_2)^2 . \end{aligned} \tag{39}$$

Its discrete-time representation, obtained by hypothetically holding the input constant over half open intervals $[kT, (k+1)T)$, doing forward finite difference and measuring output measurements at kT, is given by

$$\begin{aligned} x_{1,k+1} &= x_{1,k} + T \cdot x_{2,k}, \\ x_{2,k+1} &= x_{2,k} + T \cdot \frac{c}{m} \cdot \operatorname{sign}(u_k - x_{2,k}) \cdot (u_k - x_{2,k})^2 , \end{aligned} \tag{40}$$

where $u_k := u(kT)$ and $x_{i,k} := x_i(kT)$ for $i = 1, 2$. The above equation can be expressed as

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, u_k), \tag{41}$$

where $\boldsymbol{f}$ can be easily determined from the equations.

The motion of the float is characterized by an irregularly cycling submerging/surfacing pattern—it submerges and moves underwater for a certain duration, then resurfaces, and repeats the process over time. No matter whether it is underwater or on the surface, the depth $z_k := z(kT)$ and acceleration $\ddot{d}_{x,k} := \ddot{d}_x(kT)$ are measurable; however, the position $d_{x,k} := d_x(kT)$ can only be measured when it is at surface. Thus, irregularly sampled measurements arise as a result, with the fast one $\tau_k := \ddot{d}_{x,k}$ and slow one $\eta_k := d_{x,k}$ given by, respectively,

$$\begin{aligned} \tau_k &= \frac{c}{m} \cdot \operatorname{sign}(u_k - x_{2,k}) \cdot (u_k - x_{2,k})^2 , \\ \eta_k &= x_{1,k}. \end{aligned} \tag{42}$$

For simplicity of notation, we rewrite (42) as

$$\begin{aligned} \tau_k &= \varphi(u_k, x_k), \\ \eta_k &= \phi(x_k). \end{aligned} \tag{43}$$

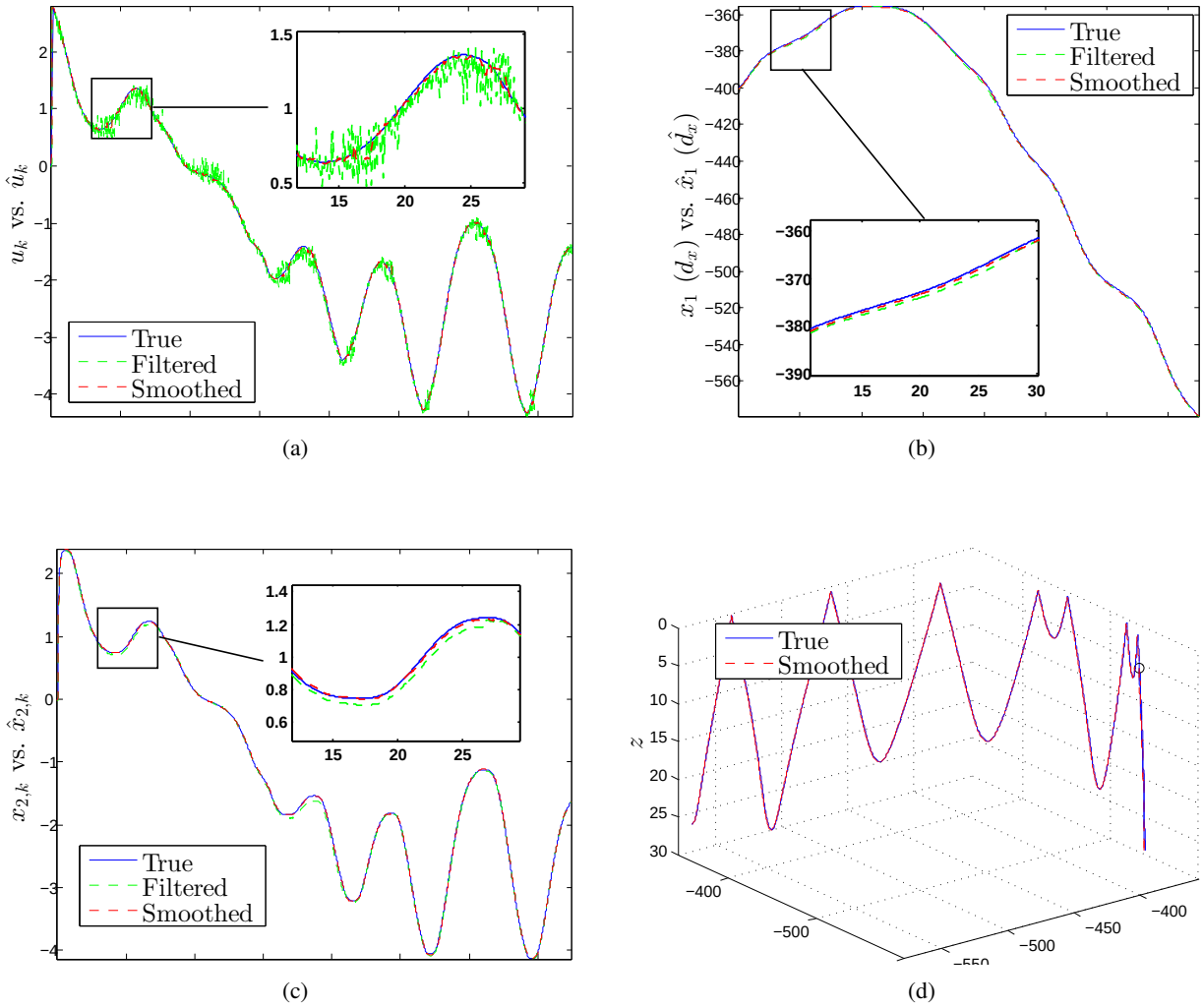


Fig. 3: Estimation results for the profiling float released at $(-400, -600)$ m: (a) x_1 — displacement along x-direction; (b) x_2 — velocity along x-direction; (c) u — flow velocity; (d) trajectory of the float (the circle denotes the location where the float is released).

Combining (40) and (42), we obtain the state space model to capture the dynamics of the float:

$$\Sigma : \begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, u_k) + \boldsymbol{w}_k, \\ \boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, u_k) + \boldsymbol{v}_k, \end{cases} \tag{44}$$

Here, when the float is underwater, $\boldsymbol{y}_k = \tau_k$ and $\boldsymbol{h} = \varphi$; when at surface, $\boldsymbol{y}_k = \begin{bmatrix} \tau_k^\top & \eta_k^\top \end{bmatrix}^\top$ and $\boldsymbol{h} = \begin{bmatrix} \varphi^\top & \phi^\top \end{bmatrix}^\top$. In addition, $\boldsymbol{w}$ and $\boldsymbol{v}$ are added to account for noise in real world. They are assumed to be white Gaussian and independent of each other. The proposed `SISF-wDF` and `SISS-wDF` algorithms are applicable to the system Σ in (44) to acquire the information estimates of not only the velocities of the flow field (unknown input variables) but also the trajectory and velocity profile of the float (state variables).

	Filtering	Smoothing
Err_u	340.02	90.61
Err_{x_1}	33931.97	19123.04
Err_{x_2}	83.08	26.89

TABLE V: A quantified comparison between filtering and smoothing errors

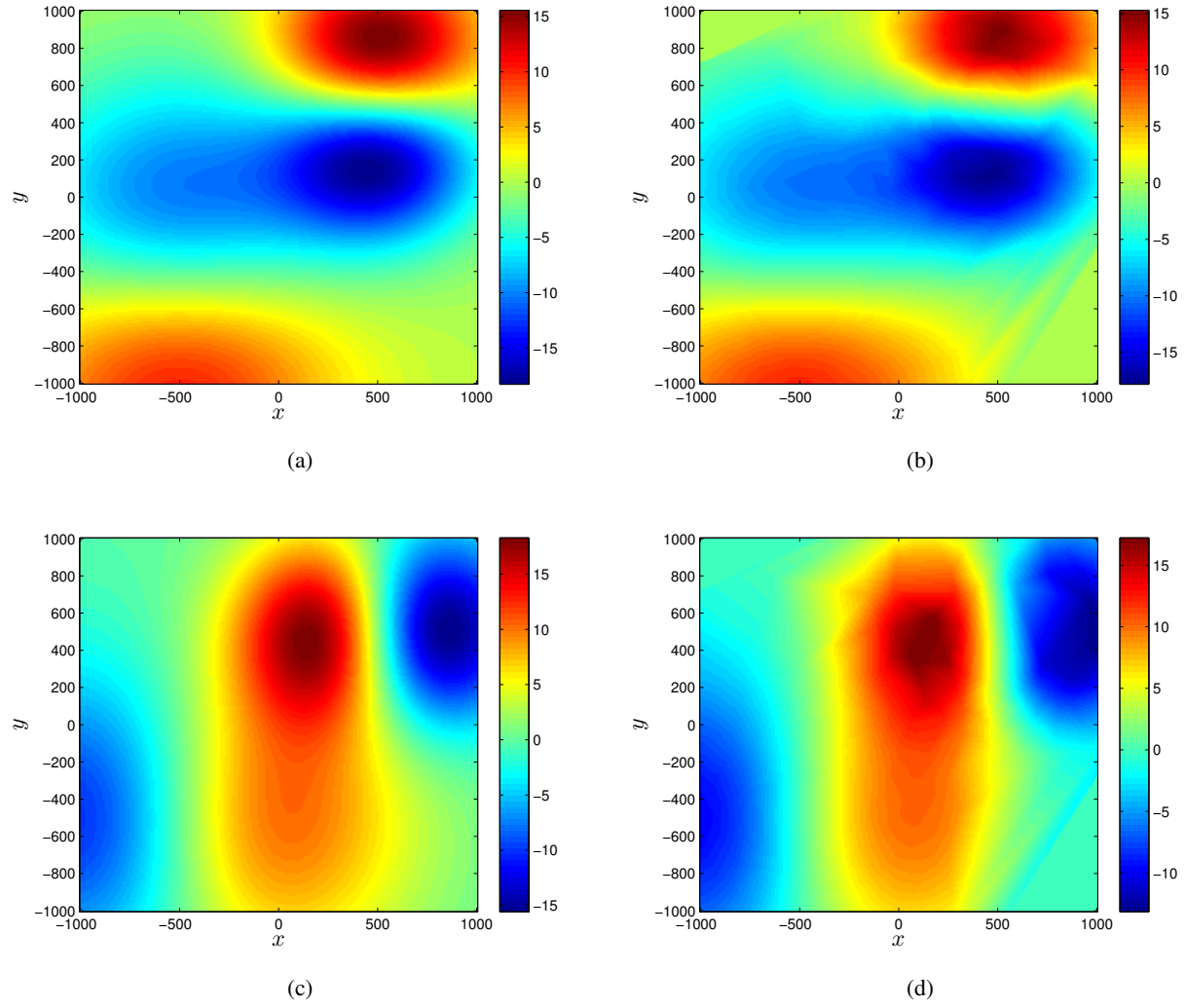


Fig. 4: (a) True surface flow velocity along x-direction; (b) smoothed surface flow velocity along x-direction; (c) True surface flow velocity along y-direction; (b) smoothed surface flow velocity along y-direction.

B. Numerical Simulation

The flow field considered has dimensions of $(-1000,1000)$ m $\times$ $(-1000,1000)$ m $\times$ $(0, 30)$ m, and the eddies are centered at $(500, 500)$ m and $(-500, -500)$ m, respectively, as shown in Fig. 2. Compared to the typical size of these flows, its scale is intentionally narrowed to reduce computational burden (this does not restrict the applicability of the proposed algorithms to larger flow fields). Let 20 profiling floats be deployed evenly along the line segment from $(-800, -1000)$ m to $(1000, 800)$ m. The mass of a profiling float is 1.5Kg, the drag coefficient c is 2 $\mathrm{N}\cdot\mathrm{s}^2/\mathrm{m}^2$, and the sampling period T is 0.05 s. The total traversing duration is 52 min.

The SISF−wDF and SISS−wDF algorithms are used together to build smoothed estimates of the inputs and states in the state-space model of a float. Let us examine the float released at $(-400, -600)$ m and consider its motion in the x-direction. Fig. 3a-3b demonstrate the filtered and smoothed estimation of the flow velocity, x-displacement, and x-velocity, respectively. We observe that the smoothed estimates are overall closer to the truth than the filtered ones. This is particularly evident for input estimation in Fig. 3a—the filtered input estimates are quite noisy, but the smoothing reduces the errors significantly, thus improving the reconstruction accuracy. A quantitative comparison is further presented in Table V. The metric is accumulative estimation error, defined by

$$\mathrm{Err}_s = \frac{1}{N}\sum_{k=1}^{N}(\hat{s}_k - s_k)^2,$$

where s_k for $k = 0, 1, \cdots, N$ is a discrete-time signal and $\hat{s}_k$ is its estimate. Table V shows that the smoothing errors are considerably smaller than the filtering errors, illustrating the enhancement of accuracy achieved by smoothing. Fig. 3d shows a good match between the smoothed trajectory and the true one.

Further, the estimated inputs of all profiling floats, which are the smoothed flow velocity data at different locations, are put together and used to reconstruct the flow field *via* the tessellation-based linear interpolation. The first and second rows in Fig. 4 illustrate the true and reconstructed flow velocity fields along x- and y-directions, respectively. One can see that the estimated velocities at the two eddies and the transition area are quit close to the actual case. Despite some minor differences observed, the overall reconstruction accuracy is quite satisfactory. This indicates that the proposed algorithms are able to provide reliable input and state estimates.

VI. Conclusion

The world has increasingly realized the scientific and commercial importance of oceans for our society. Propelled by this trend, research of ocean monitoring based on advanced mechatronic systems has seen remarkable progress. This chapter considers the critical problem of ocean flow field estimation based on inexpensive profiling floats. The success of such an ocean observing system largely depends on the availability and effectiveness of algorithms capable of extracting necessary information from the motion data of the floats. Our analysis shows that the fundamental problem is to achieve joint input and state estimation of a dynamic system, which motivates our focus on the SISE problem from the perspective of Bayesian estimation, considering both the scenarios of filtering (SISF) and smoothing (SISS). We propose Bayesian paradigms, revealing how to update the probability density functions of the unknown input and state variables conditioned on the measurement data. Based on the paradigms and leveraging some Gaussian distribution assumptions, we formulate MAP estimation problems and exploit the Gauss-Newton method to build numerical solutions, leading to a catalog of SISF and SISS algorithms. We validate the effectiveness of our approach by applying two of the algorithms to simulations for flow field estimation using a swarm of profiling floats.

References

Anderson, B. D. O. and Moore, J. B. (1979). *Optimal Filtering*. Prentice-Hall, Englewood Cliffs.

Bell, B. and Cathey, F. (1993). The iterated Kalman filter update as a Gauss-Newton method. *IEEE Transactions on Automatic Control*, 38(2):294 –297.

Björck, Å. (1996). *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia.

Booth, D. A. (1981). On the use of drogues for measuring subsurface ocean currents. *Ocean Dynamics*, 34:284–294.

Candy, J. V. (2009). *Bayesian Signal Processing: Classical, Modern and Particle Filtering Methods*. Wiley-Interscience, New York, NY, USA.

Cheng, Y., Ye, H., Wang, Y., and Zhou, D. (2009). Unbiased minimum-variance state estimation for linear systems with unknown input. *Automatica*, 45(2):485–491.

Colgan, C. (2006). Underwater laser shows. *Explorations, Scripps Institution of Oceanography*, 12(4):20–27.

Corless, M. and Tu, J. (1998). State and input estimation for a class of uncertain systems. *Automatica*, 34(6):757–764.

Costanza, R. (1999). The ecological, economic, and social importance of the oceans. *Ecological Economics*, 31:199–213.

Darouach, M. and Zasadzinski, M. (1997). Unbiased minimum variance estimation for systems with unknown exogenous inputs. *Automatica*, 33(4):717 – 719.

Darouach, M., Zasadzinski, M., and Boutayeb, M. (2003). Extension of minimum variance estimation for systems with unknown inputs. *Automatica*, 39(5):867 – 876.

Fang, H. and de Callafon, R. (2011). Nonlinear simultaneous input and state estimation with application to flow field estimation. In *Proc. of IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 6013–6018.

Fang, H. and de Callafon, R. A. (2012). On the asymptotic stability of minimum-variance unbiased input and state estimation. *Automatica*, 48(12):3183 – 3186.

Fang, H. and de Callafon, R. A. (2013). Simultaneous input and state smoothing and its application to oceanographic flow field reconstruction. In *Proceedings of American Control Conference*, pages 4705–4710.

Fang, H., de Callafon, R. A., and Cortés, J. (2013). Simultaneous input and state estimation for nonlinear systems with applications to flow field estimation. *Automatica*, 49(9):2805 – 2812.

Fang, H., de Callafon, R. A., and Franks, P. J. S. (2015). Smoothed estimation of unknown inputs and states in dynamic systems with application to oceanic flow field reconstruction. *International Journal of Adaptive Control and Signal Processing*, pages 1224–1242.

Fang, H., Shi, Y., and Yi, J. (2008). A new algorithm for simultaneous input and state estimation. In *Proceedings of American Control Conference*, pages 2421–2426.

Fang, H., Shi, Y., and Yi, J. (2011). On stable simultaneous input and state estimation for discrete-time linear systems. *International Journal of Adaptive Control and Signal Processing*, 25(8):671–686.

Fang, H., Srivas, T., de Callafon, R. A., and Haile, M. A. (2017). Ensemble-based simultaneous input and state estimation for nonlinear dynamic systems with application to wildfire data assimilation. *Control Engineering Practice*, 63:104–115.

Floquet, T., Edwards, C., and Spurgeon, S. K. (2007). On sliding mode observers for systems with unknown inputs. *International Journal of Adaptive Control & Signal Processing*, 21(8-9):638–656.

Friedland, B. (1969). Treatment of bias in recursive filtering. *IEEE Transactions on Automatic Control*, 14(4):359–367.

Gillijns, S. and De Moor, B. (2007a). Unbiased minimum-variance input and state estimation for linear discrete-time systems. *Automatica*, 43(1):111–116.

Gillijns, S. and De Moor, B. (2007b). Unbiased minimum-variance input and state estimation for linear discrete-time systems with direct feedthrough. *Automatica*, 43(5):934–937.

Gut, A. (2005). *Probability: A Graduate Course*. Springer, New York.

Ha, Q. P. and Trinh, H. (2004). State and input simultaneous estimation for a class of nonlinear systems. *Automatica*, 40(10):1779–1785.

Han, Y., De Callafon, R. A., Cortés, J., and Jaffe, J. (2010). Dynamic modeling and pneumatic switching control of a submersible drogue. In *International Conference on Informatics in Control, Automation and Robotics*, volume 2, pages 89–97, Funchal, Madeira, Portugal.

Honerkamp, J. (1993). *Stochastic Dynamical Systems: Concepts, Numerical Methods, Data Analysis*. Wiley, New York, NY, USA.

Hsieh, C.-S. (2000). Robust two-stage Kalman filters for systems with unknown inputs. *IEEE Transactions on*

Automatic Control, 45(12):2374–2378.

Hsieh, C.-S. (2010). On the optimality of two-stage Kalman filtering for systems with unknown inputs. *Asian Journal of Control*, 12(4):510–523.

Hsieh, C.-S. (2011). Optimal filtering for systems with unknown inputs via the descriptor Kalman filtering method. *Automatica*, 47(10):2313–2318.

Hsieh, C. S. (2013). A unified framework for state estimation of nonlinear stochastic systems with unknown inputs. In *Proceedings of the 9th Asian Control Conference*, pages 1–6.

Imine, H., Delanne, Y., and M'Sirdi, N. K. (2006). Road profile input estimation in vehicle dynamics simulation. *Vehicle System Dynamics*, 44(4):285–303.

Kitanidis, P. K. (1987). Unbiased minimum-variance linear state estimation. *Automatica*, 23(6):775–778.

Kitanidist, P. K. (1987). Unbiased Minimum-variance Linear State Estimation. *Automatica*, 23(6):775–778.

Mendel, J. (1977). White-noise estimators for seismic data processing in oil exploration. *IEEE Transactions on Automatic Control*, 22(5):694–706.

Mutambara, A. G. O. (1998). *Decentralized Estimation and Control for Multisensor Systems*. CRC Press, Inc., Boca Raton, FL, USA.

Ouimet, M. and Cortes, J. (2014). Robust, distributed estimation of internal wave parameters via inter-drogue measurements. *IEEE Transactions on Control Systems Technology*, 22(3):980–994.

Pina, L. and Botto, M. A. (2006). Simultaneous state and input estimation of hybrid systems with unknown inputs. *Automatica*, 42(5):755–762.

Riser, S. C., Freeland, H. J., Roemmich, D., Wijffels, S., Troisi, A., Belboch, M., Gilbert, D., Xu, J., Pouliquen, S., Thresher, A., Le Traon, P.-Y., Maze, G., Klein, B., Ravichandran, M., Grant, F., Poulain, P.-M., Suga, T., Lim, B., Sterl, A., Sutton, P., Mork, K.-A., Vlez-Belch, P. J., Ansorge, I., King, B., Turton, J., Baringer, M., and Jayne, S. R. (2016). Fifteen years of ocean observations with the global Argo array. *Nature Climate Change*, 6(2):145–153.

Robinson, E. (1957). Predictive decomposition of seismic traces. *Geophysics*, 22(4):767–778.

Schubert, U., Kruger, U., Wozny, G., and Arellano-Garcia, H. (2012). Input reconstruction for statistical-based fault detection and isolation. *AIChE Journal*, 58(5):1513–1523.

Shi, D., Chen, T., and Darouach, M. (2016). Event-based state estimation of linear dynamic systems with unknown exogenous inputs. *Automatica*, 69:275 – 288.

Spinello, D. and Stilwell, D. (2010). Nonlinear estimation with state-dependent Gaussian observation noise. *IEEE Transactions on Automatic Control*, 55(6):1358–1366.

Yong, S. Z., Zhu, M., and Frazzoli, E. (2014). Simultaneous input and state smoothing for linear discrete-time stochastic systems with unknown inputs. In *Proceedings of IEEE Conference on Decision and Control*, pages 4204–4209.

Yong, S. Z., Zhu, M., and Frazzoli, E. (2016). A unified filter for simultaneous input and state estimation of linear discrete-time stochastic systems. *Automatica*, 63:321 – 329.

You, F.-Q., Wang, F.-L., and Guan, S.-P. (2008). Hybrid estimation of state and input for linear discrete time-varying systems: A game theory approach. *Acta Automatica Sinica*, 34(6):665–669.