Ensemble Kalman Filter for Multisensor Fusion with Multistep Delayed Measurements

Sirichai Pornsarayouth and Masaki Yamakita
Department of Mechanical and Control Engineering
Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku
Tokyo, Japan 152-8550
yamakita@ctrl.titech.ac.jp

Abstract—For a target tracking problem, such as tracking of a mobile robot or an unmanned vehicle, multiple sensors are required to achieve accurate estimated position of the target. Practically, measurements from sensors arrive out of sequence, e.g., delayed data due to the processing of images. We call these measurements Out of Sequence Measurements (OOSMs). Many researches propose solutions to OOSMs using an Extended Kalman filter (EKF) or particle filter (PF) as a basic algorithm. Our previous research proposes an algorithm that applies Ensemble Kalman filter (EnKF) to handle the OOSM problem. We store ensembles of the state particles during the filtering process and make use of the information about those ensembles later. By calculating a cross covariance between ensembles from different points of time, we can directly update the current estimated state with delayed measurements. Moreover, by using EnKF, we can simply apply the method to systems with strong nonlinear models without finding any Jacobian or backward transition matrix. However, our previous algorithm only preforms well for one-step lag measurements. In order to handle multistep lag measurement, in this paper, we propose an algorithm with an additional backward updating step. We illustrate the results of simulations comparing with Rauch-Tung-Striebel (RTS) smoothing filter and the conventional algorithms proposed by [1] and [2] which apply EKF and particle filter techniques, respectively. The proposed algorithm shows commendable results compared to others.

TABLE OF CONTENTS

1 978-1-4244-7351-9/11/$26.00 ©2011 IEEE.
2 IEEEAC Paper #1492, Version 7, Updated 11/01/2011.

1. INTRODUCTION

In target tracking and sensor fusion schemes, delay of measurements always occurs; for example, processed data from an image processor or an angle reporting by a bearing sensor, etc. exhibit such delay characteristics. These measurements are considered to be Out of Sequence Measurements (OOSMs) which are prior measurements reported together with their time-stamp. For example, at time t, we have an estimated state which we believe is the most accurate estimation, but at the same time one sensor reports a prior measurement of time τ which was never reported before $t > \tau$. We can use this late measurement for updating the current estimated state to get more accurate estimation. According to [1], OOSMs whose $t - \tau$ is less than one sampling interval, are considered to be one-step lag OOSMs and to be n-step lag OOSMs, otherwise. Our goal is to find a data-association method which can effectively update the present state with these OOSMs.

General solutions to OOSM problems have been broadly proposed. An exact solution to the one-step lag OOSM problem using Extended Kalman filter (EKF) techniques was proposed by [3] . Also, the one-step solution to n-step lag problem is available in [1]. Unfortunately, the effectiveness of EKF is limited to mild nonlinear systems[3] whose noise is Gaussian and whose Jacobian matrices can be determined. For strong nonlinear system[4] whose noise is non-Gaussian, particle filter is more suitable, see [4] and [2]. However, the algorithm in [4] is not relevant because while the algorithms do filter OOSMs, they do not update the state at the present time. To a more specific case when two measurements come in sequence at fixed rate but are delayed, we can fuse delayed and non-delayed measurements together using EKF, see [5].

In our previous unpublished work where the probability of all random variables are assumed to be Gaussian, we applied the Ensemble Kalman filter (EnKF) to this problem. We calculate a cross covariance of the state between different points of time using a history of the estimated particles. Then, we can use the cross covariance to update the current estimated state with OOSMs. Moreover, the EnKF can handle strong non-

[3] Mild nonlinear system is the system which contains nonlinear function and those function is differentiable and continuous.
[4] Strong nonlinear system is the system which contains non-differentiable or discontinuous nonlinear function.

linear characteristics such as step, saturation, etc. As in [6] and [4], we applied the Cramér-Rao theorem to determine the lower bound of the covariance of an estimator. The algorithm works well on one-step lag case but does not perform very well compared to the algorithms in [1] when it is extended to the case of an arbitrary (multistep) lag. Despite of that, in this paper which is an extension of our previous work, we show that for one-step lag OOSMs there is a condition where our previous algorithm has lower estimating covariance than the covariance calculated by other methods. Furthermore, we introduce an additional backward updating step to enhance the effectiveness of the proposed filter in multistep lag case. We compare the simulation results of the modified algorithm to those of the EKF techniques in [1], the particle filter techniques in [2], and a Rauch-Tung-Striebel (RTS) smoothing filter [7].

The rest of the paper is organized as follows: section 2 presents a formulation of the OOSM problem; section 3 presents a solution to this problem which using our previous EnKF techniques; section 4 presents an effective condition for one-step lag case of this previous algorithm, and the modification that improves the algorithm in multistep lag case; section 5 introduces a constant velocity target observed by two bearing sensors which we will use as the evaluating system later, the posterior Cramér-Rao lower bound (PCRLB) algorithm for OOSMs is given in this section as well; section 6 presents the evaluation of the EnKF, EKF, particle filter, and RTS smoothing techniques for OOSMs via simulations and discuss the results, advantages, and disadvantages of those algorithms; we conclude the effectiveness of each algorithm in section 7; and the appendix summarizes brief details of EKF, particle filter techniques for OOSMs and RTS smoothing filter.

2. Problem Formulation

Consider a time-varying discrete-time system whose state propagates according to the model

$$x(k) = f_{k,k-1}(x(k-1)) + v(k, k-1), \tag{1}$$

where $f_{k,k-1}$ is a transition function that evolves the state x from time t_{k-1} to time t_k; $v(k, k-1)$ is a Gaussian zero-mean process noise which have been accumulated between t_{k-1} and t_k with the covariance $Q(k, k-1)$,

$$Q(k_2, k_1) \triangleq Q(|t_{k_2} - t_{k_1}|). \tag{2}$$

Also, the measurement model of the system is

$$z^n(k) = h^n_k(x(k)) + w^n(k), \tag{3}$$

where $z^n(k)$ is the measurement of the state x from the n^{th} sensor about t_k; h^n_k is the measurement function; $w^n(k)$ is the Gaussian zero-mean measurement noise with the covariance $R(k)$ of the n^{th} sensor at t_k. We assume that $v(k, k-1)$ and $w(k)$ are sampled from the independent and identically distributed (iid) white Gaussian noise.

At time t_k, we have the estimation of the state

$$\hat{x}(k|k) \triangleq E[x(k)|Z_k], \tag{4}$$

$$P(k|k) \triangleq COV[x(k)|Z_k], \tag{5}$$

where Z_k represents all measurements up to time t_k; for example, $\{z^2(1), z^1(2), z^1(3), \ldots, z^n(k)\}$. Suppose that at this time the earlier measurement of time

$$\tau \triangleq t_l, \quad t_l \leq t_k$$

arrives. The subscript $\bullet_l$ denotes the time step l invoked by an OOSM. We want a new estimate $\hat{x}(k|l^+)$ which includes the information about the measurement $z^n(l)$. Define

$$\hat{x}(k|l^+) \triangleq E[x(k)|Z^+_l], \tag{6}$$

$$P(k|l^+) \triangleq COV[x(k)|Z^+_l], \tag{7}$$

where

$$Z^+_l \triangleq \{Z_k, z^n(l)\}. \tag{8}$$

For simplicity, if a single measurement arrives at a time, let us neglect the superscript $\bullet^n$, and assume that we know which sensor is the source of the measurement.

3. The Previous Works

This section presents the EnKF techniques for OOSMs that was proposed in our previous research.

Solutions to the Problem

Using Bayes filter, we can update $\hat{x}(k|k)$ with an OOSM $z(l)$. The updating equations are

$$\hat{x}(k|l^+) = \hat{x}(k|k) + P_{\text{xz}}(k, l|k)P_{\text{zz}}(l|k)^{-1} \cdot (z(l) - \hat{z}(l|k)), \tag{9}$$

$$P(k|l^+) = P(k|k) - P_{\text{xz}}(k, l|k)P_{\text{zz}}(l|k)^{-1} \cdot P_{\text{xz}}(k, l|k)^{\text{T}}, \tag{10}$$

where

$$P_{\text{xz}}(k, l|k) \triangleq E[x(k) - \hat{x}(k|k)][z(l) - \hat{z}(l|k)]^{\text{T}}, \tag{11}$$

$$P_{\text{zz}}(l|k) \triangleq E[z(l) - \hat{z}(l|k)][z(l) - \hat{z}(l|k)]^{\text{T}}, \tag{12}$$

$$\hat{z}(l|k) \triangleq E[z(l)|Z_k]. \tag{13}$$

Define $f^{-1}_{k,l}$ as an inverse function of $f_{k,l}$, then

$$x(l) = f^{-1}_{k,l}(x(k) - v(k, l)) \triangleq f_{l,k}(x(k) - v(k, l)). \tag{14}$$

From (9) and (10), one may use $f_{l,k}$ to estimate the state at t_l,

$$\hat{x}(l|k) = f_{l,k}(\hat{x}(k|k) - \hat{v}(k, l|k)), \tag{15}$$

and use techniques in prior research, e.g., [1], [2], [4] to find

$$\hat{v}(k, l|k) \triangleq E[v(k, l)|Z_k], \tag{16}$$

$P_{\text{xz}}(k, l|k)$, $P_{\text{zz}}(k, l|k)$, and $\hat{z}(l|k)$ by EKF or particle filter techniques. If we apply an EKF, we need $f^{-1}_{k,l}$ and also its Jacobian. If we choose a particle filter, we need prior knowledge about the probability density function $p(z(l)|x(k))$, etc. Sometimes these functions and distributions cannot be easily found. Therefore, in our previous research, we proposed an algorithm with an approximation instead and apply it together with EnKF.

A Solution via EnKF

Let us introduce $\mathcal{X}(k_2|k_1)$ which is an ensemble of particles corresponding to the estimated state $\hat{x}(k_2|k_1)$,

$$\mathcal{X}(k_2|k_1) \triangleq \{x_1(k_2|k_1), x_2(k_2|k_1), \ldots, x_p(k_2|k_1)\}\,, \quad (17)$$

where $x_i(k_2|k_1)$ is the i^{th} particle of the estimated state $\hat{x}(k_2|k_1)$ and p is the number of particles in one ensemble. Distribution of the particles of each ensemble represents the probability of the corresponding estimate. We calculate the sample mean and covariance of an ensemble by

$$\hat{x}(k_2|k_1) \approx \frac{1}{p}\sum_{i=1}^{p} x_i(k_2|k_1), \quad (18)$$

$$P(k_2|k_1) \approx \frac{1}{p-1}\sum_{i=1}^{p}[x_i(k_2|k_1) - \hat{x}(k_2|k_1)] \cdot [x_i(k_2|k_1) - \hat{x}(k_2|k_1)]^{\mathrm{T}}. \quad (19)$$

Accordingly, we define $\mathcal{Z}(k_2|k_1)$, $\mathcal{V}(k_2,k_1)$, and $\mathcal{W}(k_1)$ as

$$\mathcal{Z}(k_2|k_1) \triangleq \{z_1(k_2|k_1), z_2(k_2|k_1), \ldots, z_p(k_2|k_1)\}\,, \quad (20)$$

$$\mathcal{V}(k_2,k_1) \triangleq \{v_1(k_2,k_1), v_2(k_2,k_1), \ldots, v_p(k_2,k_1)\}\,, \quad (21)$$

$$\mathcal{W}(k_1) \triangleq \{w_1(k_1), w_2(k_1), \ldots, w_p(k_1)\}\,, \quad (22)$$

which represent $\hat{z}(k_2|k_1)$, $v(k_2,k_1)$, and $w(k_1)$ respectively; where $z_i(k_2|k_1)$ is the i^{th} particle of the estimated measurement $\hat{z}(k_2|k_1)$; the expected value and covariance of $\mathcal{Z}(k_2|k_1)$ is approximated by the same way as (18) and (19); $v_i(k_2,k_1)$ is the i^{th} random particle of $v(k_2,k_1)$ drawn from a zero mean multivariate normal distribution with the covariance $Q(k_2,k_1)$; and $w_i(k_2,k_1)$ is the i^{th} random particle of $w(k_1)$ drawn from a zero mean multivariate normal distribution with the covariance $R(k_1)$.

According to (9) and (10), we have to determine an ensemble $\mathcal{Z}(l|k)$. Since an explicit form of $f_{l,k}$ is sometimes so difficult to find and the exact estimate of the i^{th} particle of $\mathcal{V}(k,l)$ is unknown, then we cannot use (15) and (3) for estimating $z_i(l|k)$ from $x_i(k|k)$. Therefore, We alternatively approximate $\mathcal{Z}(l|l)$ by an interpolation technique and substitute $\mathcal{Z}(l|l)$ for $\mathcal{Z}(l|k)$. It follows that (9) and (10) becomes

$$\hat{x}(k|l^+) = \hat{x}(k|k) + P_{\mathrm{xz}}(k,l|k,l)P_{\mathrm{zz}}(l|l)^{-1} \cdot (z(l) - \hat{z}(l|l))\,, \quad (23)$$

$$P(k|l^+) = P(k|k) - P_{\mathrm{xz}}(k,l|k,l)P_{\mathrm{zz}}(l|l)^{-1} \cdot P_{\mathrm{xz}}(k,l|k,l)^{\mathrm{T}}, \quad (24)$$

where

$$P_{\mathrm{xz}}(k,l|k,l) \triangleq E[x(k) - \hat{x}(k|k)][z(l) - \hat{z}(l|l)]^{\mathrm{T}}. \quad (25)$$

Suppose that $\mathcal{X}(a|a)$ and $\mathcal{X}(b|b)$ are two ensembles which were updated at t_a and t_b, respectively, where $t_a \le t_l \le t_b$. Firstly, we choose an appropriate interpolation technique (e.g., bilinear, polynomial, or spline) and interpolate $\mathcal{X}(l|l)$ from $\mathcal{X}(a|a)$ and $\mathcal{X}(b|b)$. For instance, if we choose a bilinear interpolation technique,

$$x_i(l|l) \approx x_i(a|a) + \frac{t_l - t_a}{t_b - t_a}(x_i(b|b) - x_i(a|a)), \quad \{i = 1, \ldots, p\}. \quad (26)$$

Then, for each particle of $\mathcal{X}(l|l)$, calculate

$$z_i(l|l) = h_l(x_i(l|l)) + w_i(l),\ \{i = 1, \ldots, p\}, \quad (27)$$

Now that we get $\mathcal{Z}(l|l)$, we can approximate $P_{\mathrm{xz}}(k,l|k,l)$ and $P_{\mathrm{zz}}(l|l)$ as follows:

$$\hat{z}(l|l) \approx \frac{1}{p}\sum_{i=1}^{p} z_i(l|l), \quad (28)$$

$$P_{\mathrm{zz}}(l|l) \approx \frac{1}{p-1}\sum_{i=1}^{p}[z_i(l|l) - \hat{z}(l|l)] \cdot [z_i(l|l) - \hat{z}(l|l)]^{\mathrm{T}}, \quad (29)$$

$$\hat{x}(k|k) \approx \frac{1}{p}\sum_{i=1}^{p} x_i(k|k), \quad (30)$$

$$P_{\mathrm{xz}}(k,l|k,l) \approx \frac{1}{p-1}\sum_{i=1}^{p}[x_i(k|k) - \hat{x}(k|k)] \cdot [z_i(l|l) - \hat{z}(l|l)]^{\mathrm{T}}, \quad (31)$$

Then we can apply (23),

$$x_i(k|l^+) = x_i(k|k) + P_{xz}(k,l|k,l)P_{zz}(l|l)^{-1} \cdot (z(l) - z_i(l|l)), \quad \{i = 1, \ldots, p\}, \quad (32)$$

and get $\mathcal{X}(k|l^+)$. Next, we determine $\hat{x}(k|l^+)$ and $P(k|l^+)$ by (18) and (19), respectively. If more than one measurement arrive at the same time, substitute $\hat{x}(k|l^+)$ for $\hat{x}(k|k)$ in (30)-(32) and apply the algorithm again. Note that the prior measurement $z(l)$ usually arrives before a measurement about t_k. Hence, if an estimate of the state about t_k does not exist yet, predict $\mathcal{X}(k|k-1)$ from $\mathcal{X}(k-1|k-1)$ as

$$x_i(k|k-1) = f_{k,k-1}(x_i(k-1|k-1)) + v_i(k,k-1), \quad \{i = 1, \ldots, p\}, \quad (33)$$

where $v_i(k,k-1)$ is the i^{th} particle of $\mathcal{V}(k,k-1)$ which is sampled from the iid white Gaussian noise whose covariance is equal to $Q(k,k-1)$. Then we can substitute $\mathcal{X}(k|k-1)$ for $\mathcal{X}(k|k)$ in (30)-(32).

Since the method requires a history of ensembles, we have to keep track of the ensemble of particles at all sampling time which greatly increases the storage requirement. However the delay is practically limited, so the memory usage can be bounded for one system.

Multisensor Fusion

In many applications, when short processing time is required, it is more practical to have a fusing module associates two estimated states which were simultaneously processed. Let $x_i^n(k|k)$ be the i^{th} particle of an ensemble $\mathcal{X}^n(k|k)$ of the most correct estimate $\hat{x}^n(k|k)$ from the n^{th} of all m sensors. Suppose that $x_i^1(k|k)$ and $x_i^2(k|k)$ are calculated from different sensors[5]. Hereafter, we explain the fusing technique for the EnKF. For each $\{i = 1, \ldots, p\}$,

$$x_i^{12}(k|k) = x_i^1(k|k) + P_{\mathrm{x}^1\delta}(k|k)P_{\delta\delta}(k|k)^{-1} \cdot (x_i^2(k|k) - x_i^1(k|k)), \quad (34)$$

where $x_i^{12}(k|k)$ is an i^{th} particle of an ensemble $\mathcal{X}^{12}(k|k)$ which represents the fused estimate $\hat{x}^{12}(k|k)$; and

$$e_i^1(k|k) \triangleq x_i^1(k|k) - \hat{x}^1(k|k), \quad (35)$$

$$e_i^2(k|k) \triangleq x_i^2(k|k) - \hat{x}^2(k|k), \quad (36)$$

$$P_{\mathrm{x}^1\delta}(k|k) = \frac{1}{p-1}\sum_{i=1}^{p}[e_i^1(k|k)][e_i^1(k|k) - e_i^2(k|k)]^{\mathrm{T}}, \quad (37)$$

$$P_{\delta\delta}(k|k) = \frac{1}{p-1}\sum_{i=1}^{p}[e_i^1(k|k) - e_i^2(k|k)] \cdot [e_i^1(k|k) - e_i^2(k|k)]^{\mathrm{T}}. \quad (38)$$

By pairing m measurements, the number of fusing levels reduces from m to $\lceil \log_2 m \rceil + 1$, where $\lceil \bullet \rceil$ represents a ceiling function[6].

4. An Extended Note and Improvements

In this section, we give an extended note on our previous algorithm and propose an additional step to improve the estimator in multistep lag case.

The Difference Between Using $\hat{z}(l|k)$ and Using $\hat{z}(l|l)$

In our previous work, we use $\hat{z}(l|l)$ in place of $\hat{z}(l|k)$. Both (9) and (23) are the optimal updating equations, but the covariance and cross covariance being used in (23) are different from which being used in (9). To show the difference between two cases, we give an example by investigating the covariance of the state updated by $\hat{z}(l|k)$ and the state updated by $\hat{z}(l|l)$ where $l = k-1$ (one-step lag case). For simplicity, we consider a linear time invariant system when an OOSM about $t_l = t_{k-1}$ arrives at t_k.

$$x(k) = Fx(k-1) + v(k), \quad (39)$$

$$z(k) = Hx(k) + w(k). \quad (40)$$

where F is the state transition matrix; H is the output matrix; $v(k)$ is a Gaussian random process noise with the covariance Q; and $w(k)$ is a Gaussian random measurement noise with the covariance R. Let

$$K(k|k-1) \triangleq P_{\mathrm{xz}}(k|k-1)P_{\mathrm{zz}}(k|k-1)^{-1}. \quad (41)$$

If we choose $\hat{z}(l|l)$, the corresponding covariances in (23) are

$$P_{\mathrm{xz}}(k,l|k,l) = FP(k-1|k-1)H^{\mathrm{T}} - K(k|k-1)H \cdot FP(k-1|k-1)H^{\mathrm{T}}, \quad (42)$$

$$P_{\mathrm{zz}}(l|l) = HP(k-1|k-1)H^{\mathrm{T}} + R. \quad (43)$$

If we choose $\hat{z}(l|k)$, the corresponding covariances in (9) are

$$P_{\mathrm{xz}}(k,l|k) = P_{\mathrm{xz}}(k,l|k,l) \quad (44)$$

$$P_{\mathrm{zz}}(l|k) = P_{\mathrm{zz}}(l|l) - HF^{-1}K(k|k-1)H \cdot \left[FP(k-1|k-1) - QF^{-1}\right]H^{\mathrm{T}}. \quad (45)$$

From (10) and (24), $G \triangleq P_{\mathrm{xz}}P_{\mathrm{zz}}^{-1}P_{\mathrm{xz}}^{\mathrm{T}}$ tells us how much the updating process improves the covariance of the estimate. Because of (44), only the matrix P_{zz}^{-1} indicates the effectiveness of an estimation. Equation (45) leads to a condition that using $\hat{z}(l|k)$ gives better estimation than using $\hat{z}(l|l)$ if

$$FP(k-1|k-1)F^{\mathrm{T}} - Q \succ 0. \quad (46)$$

where $\succ 0$ denotes positive definiteness of a matrix.

However, (46) is only a condition for the linear system whose measurement is a one-step lag OOSM about $t_l = t_{k-1}$. Besides, we found that using $\hat{z}(l|l)$ exhibits inferior tracking performance in multistep lag case. To address multistep lag OOSMs, one may obtain $\mathcal{X}(l|k)$ by applying Rauch-Tung-Striebel (RTS) smoothing filter, see [7] and a brief introduction in the appendix. However, RTS smoothing filter must be done recursively at any backward step from $t_k, t_{k-1}, \ldots, t_l$ which consumes the computational cost as much as redo the filtering process from t_l to t_k again. Thus we consider RTS smoothing filter as a reference algorithm and propose an additional one-step-back updating process to improve our previous algorithm for multistep lag OOSMs.

Improvements by One-step-back Update

Since the update by estimating $\hat{z}(l|l)$ is not satisfactory for multistep lag OOSMs, we try to find a method that can determine $\hat{z}(l|k)$ faster than the RTS smoothing filter. Subsequently, we propose an additional step by updating $\mathcal{X}(l|l)$ with $\mathcal{X}(k|k)$ after an interpolation in (26). Suppose that we have obtained $\mathcal{X}(l|l)$ by (26). Firstly we predict $\mathcal{X}(k|l)$ from $\mathcal{X}(l|l)$, then use the difference between $\mathcal{X}(k|k)$ and $\mathcal{X}(k|l)$ to update $\mathcal{X}(l|l)$, and obtain $\mathcal{X}(l|k)$. The update equation is

$$x_i(l|k) = x_i(l|l) + P_{\mathrm{xd}}(l,k)P_{\mathrm{dd}}(k)^{-1}(x_i(k|l) - x_i(k|k)), \quad \{i = 1, \ldots, p\}. \quad (47)$$

where

$$e_i(l|l) \triangleq x_i(l|l) - \hat{x}(l|l), \quad (48)$$

$$d_i(k) \triangleq (x_i(k|k) - x_i(k|l)) - (\hat{x}(k|k) - \hat{x}(k|l)), \quad (49)$$

$$P_{\mathrm{xd}}(l|k) = \frac{1}{p-1}\sum_{i=1}^{p}[e_i(l|l)][d_i(k)]^{\mathrm{T}}, \quad (50)$$

$$P_{\mathrm{dd}}(k) = \frac{1}{p-1}\sum_{i=1}^{p}[d_i(k)][d_i(k)]^{\mathrm{T}}. \quad (51)$$

Then, we proceed to substitute $x_i(l|k)$ for $x_i(l|l)$ in (27).

[5]Two estimated states may be calculated simultaneously by different computer or different core of processors. All filtering module share the same data pool that is continuously updated by multiple fusion modules.
[6]Note that the calculation cost does not decrease, it requires less time because multiple filters can be processed simultaneously.

5. EVALUATION

Target System

We study a two-dimension constant velocity system with time-varying step size. The mathematical model is

$$x(k) = F(\Delta t)x(k-1) + v(k,k-1), \tag{52}$$
$$v(k,k-1) \sim \mathcal{N}(0, Q(\Delta t)), \tag{53}$$

where $x = [x^{\mathrm{x}}\ \dot{x}^{\mathrm{x}}\ x^{\mathrm{y}}\ \dot{x}^{\mathrm{y}}]^{\mathrm{T}}$ is a vector of position in $\mathbf{x}$ axis, velocity in $\mathbf{x}$ axis, position in $\mathbf{y}$ axis, and velocity in $\mathbf{y}$ axis, respectively.

$$Q(\Delta t) = q \times \begin{bmatrix} \Delta t^3/3 & \Delta t^2/2 & 0 & 0 \\ \Delta t^2/2 & \Delta t & 0 & 0 \\ 0 & 0 & \Delta t^3/3 & \Delta t^2/2 \\ 0 & 0 & \Delta t^2/2 & \Delta t \end{bmatrix}, \tag{54}$$

where $\Delta t \triangleq |t_k - t_{k-1}|$ and q is a constant parameter.

We simulate two bearing sensors which measure a target from different positions. Each sensor has a distinct scanning period. These sensors report an angle where the scanning line intercepts the target at the end of scanning alongside with the intercepting time, which means the measurement from a bearing sensor itself is a one-step lag OOSM. The measurement model of a bearing sensor is

$$z(k) = \operatorname{atan2}(x^{\mathrm{y}}(k), x^{\mathrm{x}}(k)) + w(k), \tag{55}$$

$$\operatorname{atan2}(y,x) \triangleq \begin{cases} \arctan(y/x); & x > 0 \\ \pi + \arctan(y/x); & y \geq 0, x < 0 \\ -\pi + \arctan(y/x); & y < 0, x < 0 \\ \pi/2; & y > 0, x = 0 \\ -\pi/2; & y < 0, x = 0 \\ \text{undefined}; & y = 0, x = 0, \end{cases} \tag{56}$$

where $w(k)$ is a Gaussian zero-mean measurement noise with the covariance $R(k)$. This model is undefined at $x = y = 0$, discontinuous on $x < 0, y = 0$ and differentiable elsewhere. However, on real systems, $x = y = 0$ never occurs and the discontinuity of the function can be avoided by the condition that the target never travels from the angle $-\pi$ to π in an instant. Therefore, practically, the model is mildly nonlinear; EKF, particle filter, and Cramér-Rao bounds can be easily implemented. So we can investigate the efficiency of EnKF for OOSMs compared with the efficiency those algorithms.

Posterior Cramér-Rao Bound : PCRLB

By following the derivation in [6], suppose that we have a measurement z of a random variable with the distribution $f(z;\theta)$. A covariance of the estimation of any unbiased estimator is bounded by a certain value, that is

$$\mathrm{COV}(\hat{\theta}(z)) \geq J(\theta)^{-1}, \tag{57}$$

where $J(\theta)$ is the Fisher information defined by

$$J(\theta) = E\left[(\nabla_\theta \log f(z;\theta))(\nabla_\theta \log f(z;\theta))^{\mathrm{T}} | \theta\right]. \tag{58}$$

∇_θ is the partial differentiation with respect to θ. In the application of discrete time bearing sensor filtering, the parameter θ which we want to estimate is a trajectory of the state from time t_0 to t_k – $X(0{:}k)$, and the measurement is the series of over all measurement between time t_0 and t_k – $Z(0{:}k)$. So the Cramér-Rao inequality becomes

$$COV(X(0{:}k)) \geq \begin{pmatrix} E[(\nabla_{x(0:k)} \log f(Z(0{:}k), X(0{:}k)))] \\ \cdot[(\nabla_{x(0:k)} \log f(Z(0{:}k), X(0{:}k)))]^{\mathrm{T}} \end{pmatrix}^{-1}. \tag{59}$$

For simplicity, let $\log f(Z(0{:}k), X(0{:}k)) = g$. To determine $J(x(k))$ recursively, we write

$$J(X(0{:}k)) = \begin{bmatrix} \nabla_{x(0:k-1)x(0:k-1)} g & \nabla_{x(k)x(0:k-1)} g \\ \nabla_{x(0:k-1)x(k)} g^{\mathrm{T}} & \nabla_{x(k)x(k)} g \end{bmatrix}, \tag{60}$$
$$\triangleq \begin{bmatrix} A & B \\ B^{\mathrm{T}} & C \end{bmatrix}, \tag{61}$$

where $\nabla_{ab} g \triangleq \nabla_a g \nabla_b^{\mathrm{T}} g$. By applying the Schur's complement, we get

$$J(x(k)) = C - B^{\mathrm{T}} A B. \tag{62}$$

When we apply Markov properties, a recursive equation can be calculated as (63) on the next page. From (63), by using the Schur's complement, we can find $J(x(k+1))$ as

$$J(x(k+1)) = D_{22} - D_{12}^{\mathrm{T}} \left(D_{11} + J(x(k))\right)^{-1} D_{12}. \tag{64}$$

Using this recursive form, we can calculate the minimum variance of any estimate $\hat{x}(k)$ given by an actual state $x(k)$.

For a constant velocity target with bearing sensors in section 5 whose noise is assumed to be Gaussian, at time t_k, D_{11}, D_{12}, and D_{22} are determined,

$$D_{11} = [F(\Delta t)]^{\mathrm{T}} [Q(\Delta t)]^{-1} [F(\Delta t)], \tag{65}$$
$$D_{12} = -[F(\Delta t)]^{\mathrm{T}} [Q(\Delta t)]^{-1}, \tag{66}$$
$$D_{22} = [Q(\Delta t)]^{-1} + \frac{1}{R(k)} \begin{pmatrix} E[\nabla_{x(k)} h_k(x(k))] \\ \cdot[\nabla_{x(k)} h_k(x(k))]^{\mathrm{T}} \end{pmatrix}. \tag{67}$$

The Jacobian matrix of the arctan function is

$$\nabla_{x_k} h_k(x(k)) = \frac{\begin{bmatrix} -\Delta x^{\mathrm{y}} & 0 & \Delta x^{\mathrm{x}} & 0 \end{bmatrix}^{\mathrm{T}}}{\left((\Delta x^{\mathrm{x}})^2 + (\Delta x^{\mathrm{y}})^2\right)}, \tag{68}$$

where Δx^{x} and Δx^{y} is the distance from a sensor to the target along $\mathbf{x}$ and $\mathbf{y}$ directions, respectively.

For OOSM, we consider the late measurement $z(l)$ as a measurement of the present time t_k with an equivalent measurement function as

$$h_k^*(x(k)) \triangleq h_l(f_{l,k}(x(k))), \tag{69}$$

and the covariance of an equivalent measurement noise is

$$R^*(k) = R(k) - [\nabla_{x(k)} h_k^*(x(k))][Q(k,l)][\nabla_{x(k)} h_k^*(x(k))]^{\mathrm{T}} \tag{70}$$

Substituting $h_k^*(x(k))$ and $R^*(k)$ for $h_k(x(k))$ and $R(k)$ in (67), respectively, then we can calculate PCRLB for OOSM filtering.

$$
\begin{aligned}
&E\left[\nabla_{x(0:k+1)}\log f(Z(0{:}k+1),X(0{:}k+1))\right]\left[\nabla_{x(0:k+1)}\log f(Z(0{:}k+1),X(0{:}k+1))\right]^{\mathrm{T}}\\
&\quad= E\left[\nabla_{x(0:k+1)x(0:k+1)}\log f(Z(0{:}k+1),X(0{:}k+1))\right]\\
&\quad= E\left[\nabla_{x(0:k+1)x(0:k+1)}\left\{\log f(Z(0{:}k),X(0{:}k))+\log p(x(k+1)|x(k))+\log p(z(k+1)|x(k+1))\right\}\right]\\
&\quad=\begin{bmatrix} A & B & 0\\ B^{\mathrm{T}} & C & 0\\ 0 & 0 & 0\end{bmatrix}+\begin{bmatrix} 0 & 0 & 0\\ 0 & E\left[\nabla_{x(k)x(k)}p(x(k+1)|x(k))\right] & E\left[\nabla_{x(k)x(k+1)}p(x(k+1)|x(k))\right]\\ 0 & E\left[\nabla_{x(k+1)x(k)}p(x(k+1)|x(k))\right] & E\left[\nabla_{x(k+1)x(k+1)}\left\{p(x(k+1)|x(k))+p(z(k+1)|x(k+1))\right\}\right]\end{bmatrix}\\
&\quad\triangleq\begin{bmatrix} A & B & 0\\ B^{\mathrm{T}} & C & 0\\ 0 & 0 & 0\end{bmatrix}+\begin{bmatrix} 0 & 0 & 0\\ 0 & D_{11} & D_{12}\\ 0 & D_{12}^{\mathrm{T}} & D_{22}\end{bmatrix}=\left[\begin{array}{cc|c} A & B & 0\\ B^{\mathrm{T}} & C+D_{11} & D_{12}\\ \hline 0 & D_{12}^{\mathrm{T}} & D_{22}\end{array}\right].
\end{aligned}
\tag{63}
$$

6. Results and Discussions

We simulate the system described in section 5 with five different types of filters: 1) tracking the target with the measurements from the first sensor ('Fil-1') or from the second sensor ('Fil-2') only; 2) tracking the target with the measurements from both sensors using an algorithm summarized by [1] which uses the EKF as a basic filter ('OOSM-EKF'); 3) tracking the target by the particle filter techniques proposed by [2] ('OOSM-PF') where the Monte-Carlo simulations run with 10000 particles; 4) tracking the target by the RTS smoothing filter ('OOSM-RTS'); and 5) tracking the target by our previous algorithm ('OOSM-EnKF1') as well as the algorithm proposed in this paper ('OOSM-EnKF2'), running with 250 particles. Brief introductions to 'OOSM-EKF', 'OOSM-PF', and 'OOSM-RTS' are provided in the appendix.

In the simulation, two sensors are located at $(-200, 0)$ and $(200, 0)$, each sensor has a distinct scanning period, 2.6 and 2.8 seconds of $360°$ scanning for the first and second sensors, respectively. Resolution of each sensor is set to $0.1°$. The target starts moving with the initial state $[-150\ 6\ 150\ 1]^{\mathrm{T}}$, each filter starts with an initial estimated value $[-150\ 6\ 160\ 1]^{\mathrm{T}}$. The simulations are done in two cases: one-step lag OOSM and multistep lag OOSM.

For the one-step lag case where the measurement from each sensor is fed directly to the filters, 'OOSM-EnKF1' and 'OOSM-EnKF2' are the same algorithm marked as 'OOSM-EnKF'. Figure 1 shows tracking trajectory of the filters sourced by two sensors. We can see that all filters can track the target accurately except the 'Fil-1' and 'Fil-2' lines. Figure 2, Figure 3, Figure 4, and Figure 5 present the standard deviation (SD) and the absolute error of the tracking trajectory in $\mathbf{x}$ and $\mathbf{y}$ directions, respectively. In Figure 2 and Figure 4, the solid line ('OOSM-PCRLB') represents the lower bound of the SD of any unbiased estimator determined by PCRLB provided in section 5. The SD of the estimate from each algorithm is close to one another.

For the multistep lag case, We simulate two experiments. Firstly, the delay of the measurement from the first sensor is introduced by 5s and is introduced by 10s for the second experiment. By observing the 'Fil-1' line in Figure 6, we can

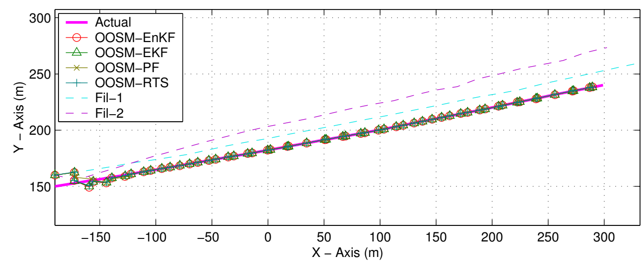


Figure 1. Tracking trajectory: one-Step lag.

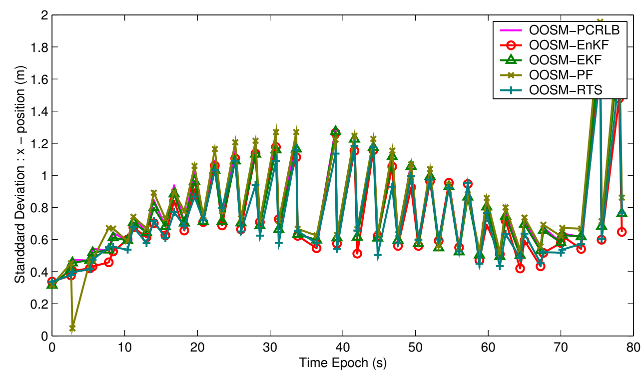


Figure 2. Standard deviation of the estimated position in $\mathbf{x}$ axis: one-Step lag.

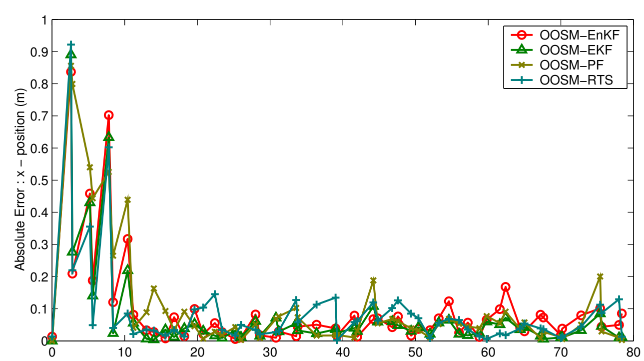


Figure 3. Absolute error of the estimated position in $\mathbf{x}$ axis: one-Step lag.

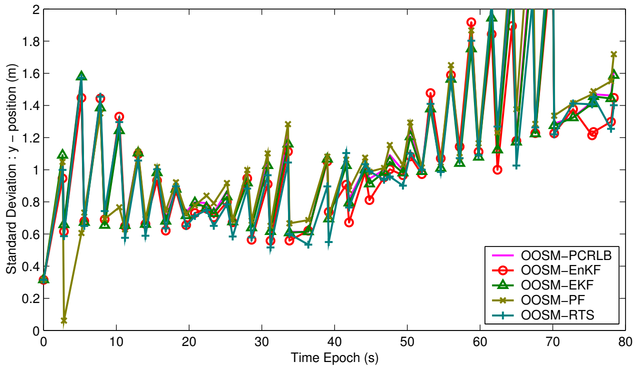


Figure 4. Standard deviation of the estimated position in $\mathbf{y}$ axis: one-Step lag.

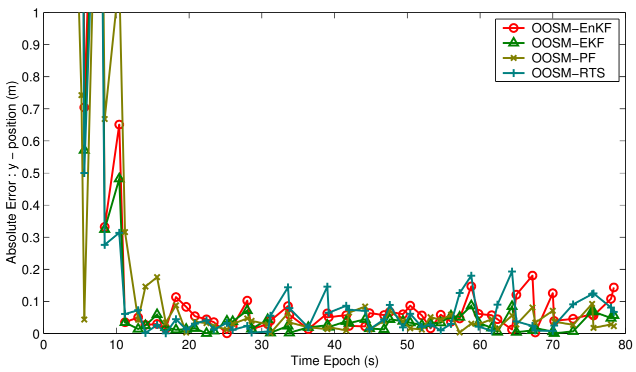


Figure 5. Absolute error of the estimated position in $\mathbf{y}$ axis: one-Step lag.

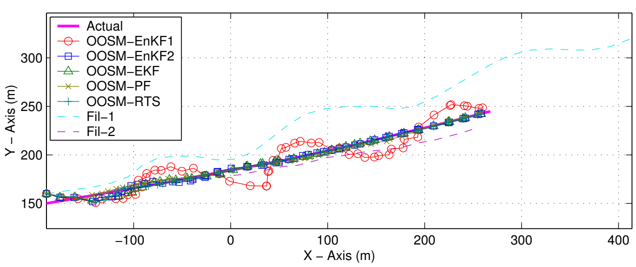


Figure 6. Tracking trajectory: multistep lag (5s).

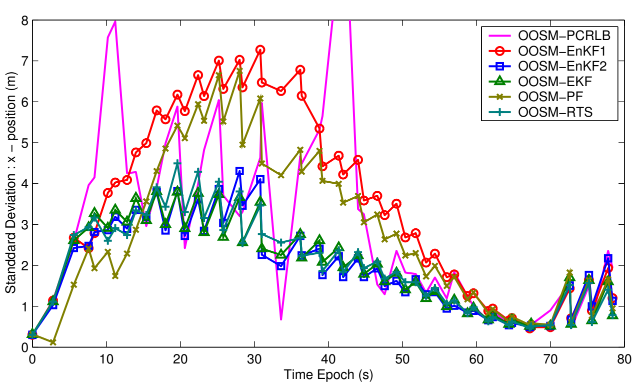


Figure 7. Standard deviation of the estimated position in $\mathbf{x}$ axis: multistep lag (5s).

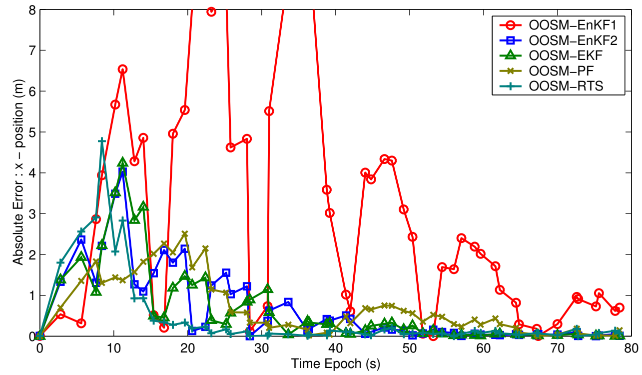


Figure 8. Absolute error of the estimated position in $\mathbf{x}$ axis: multistep lag (5s).

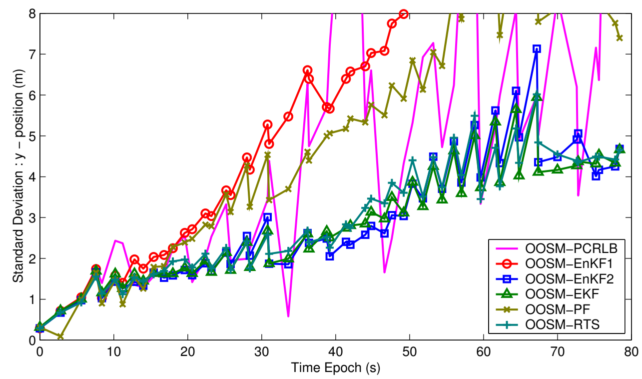


Figure 9. Standard deviation of the estimated position in $\mathbf{y}$ axis: multistep lag (5s).

see that the additional delay of the first sensor is the cause of false tracking in filtering processes. The SD and absolute error of each estimate in Figure 7, Figure 8, Figure 9, and Figure 10 show that the ‘OOSM-EnKF2’, ‘OOSM-EKF’, and ‘OOSM-RTS’ techniques exhibit very good tracking performance. While the ‘OOSM-PF’ algorithm also gives good tracking performance but the SD is still high despite requiring as many as 10000 particles to keep the filter stable (unstable with 5000 particles). We inspected that, in the ‘OOSM-PF’ algorithm, $v_i(k,l)$ generated by (88) is independent of the actual $v(k,l)$. As a result, (89) increases the sample covariance of $\hat{x}(l|k)$ while other algorithms reduce this covariance during the same process. For multistep lag case, the results of ‘OOSM-EnKF1’ is the worst among all algorithms presented in this paper. We also inspected that the lower bounds determined by ‘OOSM-PCRLB’ are much worse than the SD of ‘OOSM-EKF’ because ‘OOSM-EKF’ estimates the process noise accumulated between t_l and t_k, and use it in (15) while the ‘OOSM-PCRLB’ technique presented in section 5 consider it as zero. If the delay of the measurement greatly increases, e.g., 10s as the results in Figure 11, Figure 12, Figure 13, Figure 14, and Figure 15, the ‘OOSM-PF’ algorithm becomes unstable despite running with 10000 particles. The simulated results of ‘OOSM-EnKF1’ suffer from the effect of inaccurate state estimation about the measured time (high delay). While the ‘OOSM-EKF’ and ‘OOSM-EnKF2’ algo-

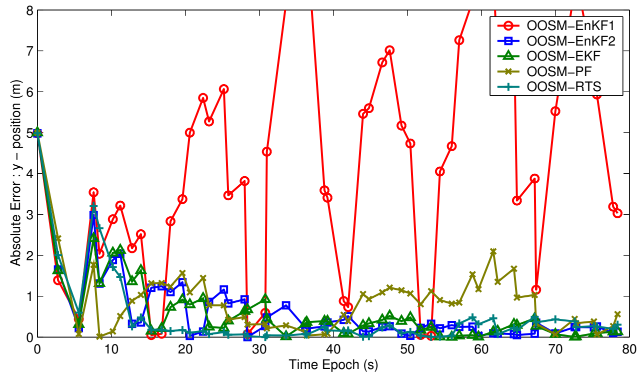


Figure 10. Absolute error of the estimated position in $\mathbf{y}$ axis: multistep lag (5s).

rithms show better results; however, the tracking performance of both algorithms are still not as accurate as the performance of 'OOSM-RTS'; and 'OOSM-EKF' is superior to 'OOSM-EnKF2' for extremely high-delay measurements.

7. Conclusions

From the condition we give in section 4 and the results of one-step lag case, we conclude that the algorithm of our previous research ('OOSM-EnKF1') is a good choice for one-step lag OOSM filtering, especially for strong nonlinear systems and complex systems in which the inverse of $f_{k,k-1}$ can hardly or cannot be determined. For multistep lag case, if the system is linear or mildly nonlinear, the 'OOSM-EKF' algorithm is better in the tracking performance and consumes less computational cost. But if the system is strongly nonlinear so that EKF technique is not applicable, either 'OOSM-PF' or 'OOSM-EnKF' which represents the state probability by a group of estimated particles can be applied. However, we recommend the 'OOSM-EnKF2' algorithm because it indeed reduces complexity of the calculation, consumes less computational cost, and is more stable than 'OOSM-PF'. Similar to 'OOSM-EnKF1', the 'OOSM-EnKF2' algorithm requires only the state transition function which makes the algorithm also applicable to the system whose the inverse function of $f_{k,k-1}$ cannot be found.

Appendix

A solution via EKF

For simplicity, we illustrate the solution only for the linear system. For the nonlinear case, we can simply approximate the state transition matrix by the Jacobian matrix of the propagating function of the system. Thus, by following [1], let us consider the system

$$x(k) = F(k,k-1)x(k-1) + v(k,k-1), \tag{71}$$

$$z(k) = H(k)x(k) + w(k), \tag{72}$$

where $F(k,k-1)$ is the state transition matrix that innovates the state from t_{k-1} to t_k and $H(k)$ is the measurement matrix at time t_k. For a one-step lag case where $t_{k-1} \leq \tau \leq t_k$, we

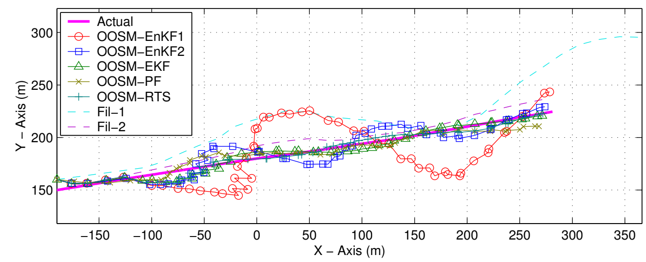


Figure 11. Tracking trajectory: Multistep lag (10s).

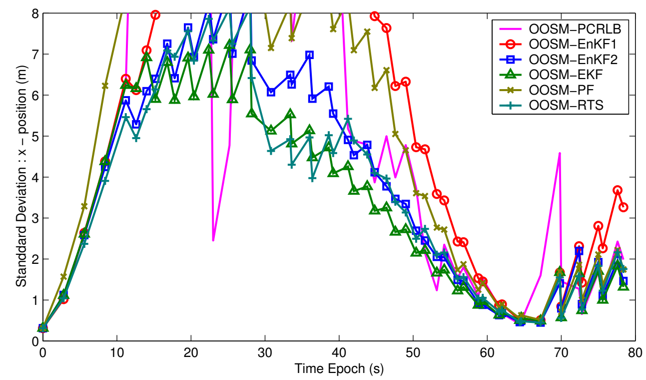


Figure 12. Standard deviation of the estimated position in $\mathbf{x}$ axis: multistep lag (10s).

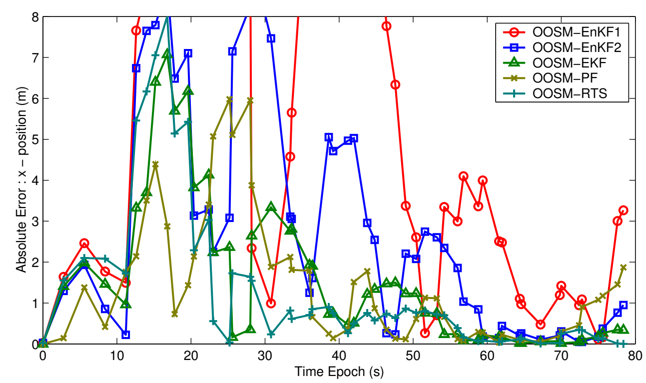


Figure 13. Absolute error of the estimated position in $\mathbf{x}$ axis: multistep lag (10s).

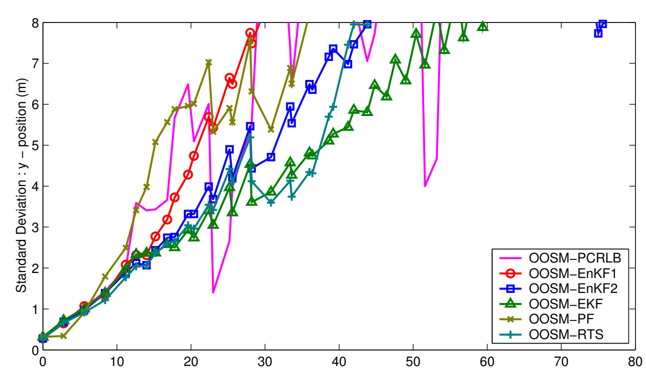


Figure 14. Standard deviation of the estimated position in $\mathbf{y}$ axis: multistep lag (10s).

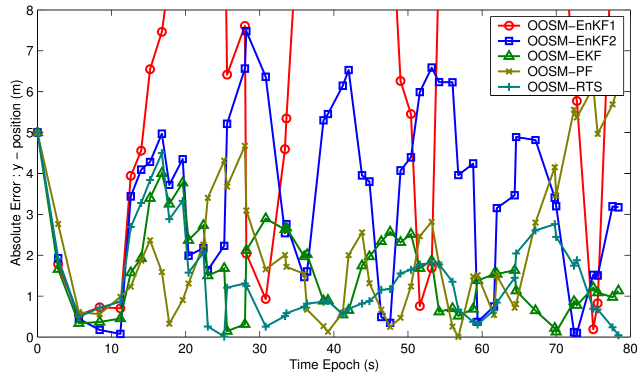


Figure 15. Absolute error of the estimated position in $\mathbf{y}$ axis: multistep lag (10s).

rewrite a relationship between $x(k)$ and $x(l)$ as

$$x(l) = F(l,k)\left[x(k) - v(k,l)\right], \tag{73}$$

where $F(l,k) = F(k,l)^{-1}$ represents a backward transition matrix that estimates the state back from t_k to τ. From (9) and (10), we can achieve the optimal estimation. Using (72) and (73), we determine $\hat{x}(l|k)$ and $\hat{z}(l|k)$ by

$$\hat{x}(l|k) = F(l,k)[\hat{x}(k|k) - \hat{v}(k,l|k)], \tag{74}$$
$$\hat{z}(l|k) = H(l)\hat{x}(l|k). \tag{75}$$

During an update from $\hat{x}(k|k-1)$ to $\hat{x}(k|k)$ with the measurement $z(k)$, we can also estimate the process noise which have been accumulated between t_l and t_k by

$$\hat{v}(k,l|k) = Q(k,l)H(k)^{\mathrm{T}}P_{\mathrm{zz}}(k|k-1)^{-1}(z(k) - \hat{z}(k|k-1)). \tag{76}$$

Using (76), we can compute (74). To complete (9) and (10), we have to find $P_{\mathrm{xz}}(k,l|k)$ and $P_{\mathrm{zz}}(l|k)$ which is the cross-covariance between $\hat{x}(k|k)$ and $\hat{z}(l|k)$, and the covariance of $\hat{z}(l|k)$, respectively. They can be determined by

$$P_{\mathrm{xz}}(k,l|k) = \left(P\left(k|k\right) - P_{\mathrm{xv}}\left(k,l|k\right)\right)F(l,k)^{\mathrm{T}}H(l)^{\mathrm{T}}, \tag{77}$$
$$P_{\mathrm{zz}}(l|k) = H(l)P(l|k)H(l)^{\mathrm{T}} + R(l), \tag{78}$$

where the covariance of the backward estimated state is

$$\begin{aligned} P(l|k) = F(l,k)(P\left(k|k\right) + P_{\mathrm{vv}}\left(l|k\right) \\ - P_{\mathrm{xv}}\left(k,l|k\right) - P_{\mathrm{xv}}\left(k,l|k\right)^{\mathrm{T}})F(l,k)^{\mathrm{T}}. \end{aligned} \tag{79}$$

$P_{\mathrm{vv}}(l|k)$ which is the covariance of $\hat{v}(k,l|k)$; and $P_{\mathrm{xv}}(l|k)$ which is the cross-covariance of $\hat{x}(k|k)$ and $\hat{v}(k,l|k)$ are derived as follows:

$$\begin{aligned} P_{\mathrm{vv}}(l|k) = Q(k,l) - Q(k,l)H(k)^{\mathrm{T}} \\ \cdot P_{\mathrm{zz}}(k|k-1)^{-1}H(k)Q(k,l)^{\mathrm{T}}, \end{aligned} \tag{80}$$
$$\begin{aligned} P_{\mathrm{xv}}(k,l|k) = Q(k,l) - P(k|k-1)H(k)^{\mathrm{T}} \\ \cdot P_{\mathrm{zz}}(k|k-1)^{-1}H(k)Q(k,l)^{\mathrm{T}}. \end{aligned} \tag{81}$$

For the n-step lag case, $t_{k-n} \leq \tau \leq t_{k-n+1}$, according to [1], they find an equivalent measurement $z^*(k)$ and its covariance $R^*(k)$ that update $\hat{x}(k|k-n)$ into $\hat{x}(k|k)$ and consider it as a one-step propagation from t_{k-n} to t_k with a measurement $z^*(k)$. With an equivalent measurement, a conventional Kalman filter would update the estimate by

$$\begin{aligned} \hat{x}(k|k) = \hat{x}(k|k-n) + P(k|k)H^*(k)^{\mathrm{T}}R^*(k)^{-1} \\ \cdot(z^*(k) - H^*(k)\hat{x}(k|k-n)), \end{aligned} \tag{82}$$
$$P(k|k)^{-1} = P(k|k-n)^{-1} + H^*(k)^{\mathrm{T}}R^*(k)^{-1}H^*(k). \tag{83}$$

Since $z^*(k)$ is a virtual measurement, one can simply choose $H^*(k) = I$, we get

$$R^*(k)^{-1} = P(k|k)^{-1} - P(k|k-n)^{-1}. \tag{84}$$

By letting $W^*(k) = P(k|k)H^*(k)^{\mathrm{T}}R^*(k)^{-1}$, from (82), we can calculate an equivalent measurement as

$$z^*(k) = \hat{x}(k|k-n) + W^*(k)^{-1}\left(\hat{x}(k|k) - \hat{x}(k|k-n)\right), \tag{85}$$

and its covariance

$$P^*_{\mathrm{zz}}(k|k-n) = P^*(k|k-n) + R^*(k). \tag{86}$$

Now, we can apply the one-step lag algorithm with the equivalent measurement to the n-step lag OOSMs by: substituting $z^*(k)$ and $\hat{z}(k|k-n)$ for $z(k)$ and $\hat{z}(k|k-1)$ in (76), respectively; substituting $P^*_{\mathrm{zz}}(k|k-n)$ for $P^*_{\mathrm{zz}}(k|k-1)$ in (76), (80), and (81). Note that $\hat{v}(k,n|k)$ is assumed to be orthogonal on the process noise in the interval $T_n = (t_{k-n}, t_k]$.

A Solution via Particle Filter

The derivation here is based on the algorithm proposed by [2] with two differences 1) if $\hat{x}(k|k)$ is not available, we use the distribution of $\hat{x}(k|k-1)$ as a proposal distribution instead; 2) we estimate the state at t_k not t_l. The algorithm is as follows:

1. Initialize the algorithm by selecting the number of particles (N) and the threshold of resampling (N_{thres}), and generate N particle $\{x^i(1)\}_{i=1}^N$ from its probability density $p_0(x(1))$ which is Gaussian in our interest. The initial weight of each particle $\{w^i(1) = 1/N\}_{i=1}^N$.

2. For each step k, generate N samples of the process noise $\{v^i(k,k-1)\}_{i=1}^N \sim \mathcal{N}(0, Q(k,k-1))$ and predict the particle from latest updated particle to the present time,

$$\{x^i(k) = f_{k,k-1}(x^i(k-1)) + v^i(k,k-1)\}_{i=1}^N. \tag{87}$$

3. For each step at t_k, generate the estimated process noise,

$$\{v^i(k,l)\}_{i=1}^N \sim \mathcal{N}(0, Q(k,l)). \tag{88}$$

4. Estimate the particles of the state back to t_l,

$$\{x^i(l) = f_{l,k}(x^i(k) - v^i(k,l))\}_{i=1}^N. \tag{89}$$

5. For the measurement $z(l)$, calculate the likelihood $\{p(z(l)|x^i(l))\}_{i=1}^N$ which is also Gaussian in our case.

6. Update the weight $\{w^i(k)\}_{i=1}^N$,

$$\left\{ w^i(k|l^+) = \frac{w^i(k)p(z(l)|x^i(l))}{\sum_{i=1}^N w^i(k)p(z(l)|x^i(l))} \right\}_{i=1}^N. \tag{90}$$

7. The updated state and its covariance at t_k are given as

$$\hat{x}(k|l^+) = \sum_{i=1}^{N} w^i(k|l^+) x^i(k),\ e^i(k) \triangleq x^i(k) - \hat{x}(k|l), \quad (91)$$
$$P(k|l^+) = \sum_{i=1}^{N} w^i(k|l^+)[e^i(k)][e^i(k)]^{\mathrm{T}}. \quad (92)$$

8. Compute, $N_{\text{eff}} = 1/\sum_{i=1}^{N} w^i(k|l^+)^2$. If $N_{\text{eff}} < N_{\text{thres}}$, perform resampling.

9. Store $\{w^i(k|l^+)\}_{i=1}^N$ and $\{x^i(k)\}_{i=1}^N$ for the next update at t_{k+1}, then assign $k := k+1$ and Go to step 2.

Rauch-Tung-Striebel Smoothing Filter

Supposed that at the step $k+1 \in \{1,\ldots,T\}$, where T is the present step and $T \geq k+1$, we have a smooth estimation $\hat{x}^s(k+1|T)$ and the covariance $P^s(k+1|T)$ we can use the orthogonal principle like the conventional Kalman filter to estimate the smooth estimation $\hat{x}^s(k|T)$ from $\hat{x}(k|k)$. Using (1), we can determine $\hat{x}(k+1|k)$, $P(k+1|k)$, and $P(k,k+1|k)$ from $\hat{x}(k|k)$ and $P(k|k)$, where $P(k,k+1|k)$ is the cross covariance of the estimation of the state between t_k and t_{k+1}. The smoothing algorithm is

$$\begin{aligned}\hat{x}^s(k|T) &= \hat{x}(k|k) + P(k,k+1|k)P(k+1|k)^{-1} \\ &\quad \cdot(\hat{x}^s(k+1|T) - \hat{x}(k+1|k)), \qquad (93)\\ P^s(k|T) &= P(k|k) + P(k,k+1|k)P(k+1|k)^{-1} \\ &\quad \cdot(P^s(k+1|T) - P(k+1|k)) \\ &\quad \cdot P(k+1|k)^{-1}P(k,k+1|k). \qquad (94)\end{aligned}$$

Starting from the step T we can recursively determine the smooth estimation $\hat{x}^s(k|T); k \in \{1,\ldots,T\}$. In OOSM cases, we use the RTS smoothing algorithm to determine $\hat{x}^s(l|k)$ from the latest state which may be $\hat{x}(k)$ if $z(k)$ has arrived and been processed or may be $\hat{x}(k-1)$ if $z(k)$ has not arrived yet. For more details, see [7] and [8].

REFERENCES

[1] Y. Bar-Shalom, H. Chen, and M. Mallick, "One-step solution for the multistep out-of-sequence-measurement problem in tracking," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 40, no. 1, pp. 27–37, January 2004.

[2] M. Mallick, T. Kirubarajan, and S. Arulampalam, "Out-of-sequence measurement processing for tracking ground target using particle filters," in *Aerospace Conference Proceedings, 2002. IEEE*, 2002, pp. 4–1809–4–1818.

[3] Y. Bar-Shalom, "Update with out-of-sequence measurements in tracking: exact solution," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 38, no. 3, pp. 769–777, July 2002.

[4] M. Orton and A. Marrs, "Particle filters for tracking with out-of-sequence measurements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 2, pp. 693–702, April 2005.

[5] S. Pornsarayouth and M. Wongsaisuwan, "Sensor fusion of delay and non-delay signal using Kalman Filter with moving covariance," in *ROBIO '09: Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetics*. Washington, DC, USA: IEEE Computer Society, February 2009, pp. 2045–2049.

[6] P. Tichavsky, C. H. Muravchik, and A. Nehorai, "Posterior Cramer-Rao bounds for discrete-time nonlinear filtering," *IEEE Transactions on Signal Processing*, vol. 46, no. 5, pp. 1386–1396, May 1998.

[7] H. E. Rauch, C. T. Striebel, and F. Tung, "Maximum Likelihood Estimates of Linear Dynamic Systems," *Journal of the American Institute of Aeronautics and Astronautics*, vol. 3, no. 8, pp. 1445–1450, August 1965.

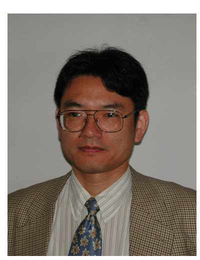
[8] S. SÄrkkÄ, "Unscented Rauch–Tung–Striebel Smoother," *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 845–849, April 2008.

[9] S. Ishihara and M. Yamakita, "Constrained state estimation for nonlinear systems with non-Gaussian noise," in *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. IEEE, December 2009, pp. 1279–1284.

[10] J. B. Gao and C. J. Harris, "Some remarks on Kalman filters for the multisensor fusion," *Information Fusion*, vol. 3, no. 3, pp. 191–201, September 2002.

BIOGRAPHY

Sirichai Pornsarayouth received his B.S. and M.S. degrees in Electrical Engineering from Chulalongkorn University, Thailand in 2005 and 2008, respectively. He is currently a Ph.D. student in the Department of Mechanical and Control Engineering at Tokyo Institute of Technology, Japan. His research interests include multisensor fusion, localization, estimation theory, bicycle robot, quadruped walking robot, and embedded control system.



Masaki Yamakita received the B.S., M.S. and Ph.D. degrees from Tokyo Institute of Technology, in 1984, 1986 and 1989, respectively. From 1989, he was a Research Associate in Department of Control Engineering of the Institute and, from 1993, he was a Lecturer at Toyohashi University of Technology. He is currently an Associate Professor in the Department of Mechanical and Control Engineering at Tokyo Institute of Technology, Japan. His research interests include robotics, learning control, and robust and nonlinear control. He is a Member of the IEEE, SICE and RSJ.