A Novel Estimation Technique Based on Curve Fitting and Sigma-Point Kalman Filtering with Time-delayed Measurements

Liang Sun[1] and Daniel Pack[2]

***Abstract*— In this paper, we present a novel mobile target tracking filter that incorporates delayed measurements captured by a video camera onboard a flying platform, such as Unmanned Aerial Vehicles (UAVs). A curve-fit-based technique is developed to characterize the system state information in the past. The coefficients are obtained and updated using an Extended Kalman Filter (EKF) based technique, which requires a much smaller amount of storage memory compared to a buffer based method. A filtering technique is then developed based on the EKF-fitted curve and the Sigma Point Kalman Filter (SPKF) technique. The simulations results show that the proposed tracking filter outperforms two other existing tracking filters, out-of-order SPKF and augmented-state SPKF.**

I. Introduction

Real-time tracking of a fast moving ground target from an unmanned aerial vehicle (UAV) using only a video sensor is a challenging task due to 3D to 2D mapping and relatively large time delays it takes to process images. The relatively large time required by image processing algorithms calls for a new estimation approach that can update the target states using time-delayed and out-of-sequence measurements. The techniques of utilizing time-delayed measurements to correct current state estimates can be classified into two categories: (1) approaches to backward-predict the state estimate when the measurement was taken and update the current state estimates using associated cross-correlation; and (2) approaches to forward-predict delayed measurements to the current time and update estimated states accordingly. We adopted the first approach due to its superior performance over the second one.

Literature search shows researchers have been working to incorporate time-delayed measurements in estimation techniques using linear system dynamic models [1], [2], [3], [4], [5], [6], [7], [8], [9]. When dealing with nonlinear models, buffer-based methods are typically used, applicable when dealing with predictable time-delayed measurements [10], [11]. For measurements with uncertain time delays, such as measurements obtained by processing the output of a video camera sensor, buffer-based approaches require a large storage, making them impractical.

In this paper, we developed a novel, non-linear system based estimator to handle time-delayed measurements. We compare its performance with two other estimation techniques, Out-Of-Order Sigma-Point Kalman Filter (O3SPKF) and an augmented state estimator. The O3SPKF estimation technique was first proposed in [10]. The approach advocates to predict a previous state from the current estimated state when the delayed measurement was obtained. The technique's merit was demonstrated with an assumption that the state propagation function is invertible, allowing the backward propagation possible. As an alternative method to the O3SPKF technique, an augmented-state-based Sigma-Point Kalman Filter (ASBSPKF) technique was proposed in [11]. The technique uses a SPKF and a time delay sensor fusion technique to update the state estimation.

The reminder of the paper is structured as follows. Section II introduces three, one new and two existing, techniques to handle delayed measurements for state estimation. Simulation results using the three different techniques are presented in Section III, and Section IV concludes the paper.

II. Estimation Techniques with Time-delayed Measurements

A. Problem Formulation

Before presenting the three estimation techniques, we define some key notations. Let t_m be the time when a measurement is taken and t be the present time. Further, let $x(t)$ be the true state at time t and $\hat{x}(t)$ be an estimate of the state corresponding to time t. The problem of our interest is finding an optimal method to use a delayed measurement, $y(t_m)$, to correct the state estimate at the current time, $\hat{x}(t)$.

B. Two existing estimation techniques

The complete Out-Of-Order Sigma-Point Kalman Filter (O3SPKF) algorithm is presented in Table I. A detailed description can be found in [10]. The algorithm used in the Augmented State Based SPKF (ASBSPKF) technique is shown in Table II.

C. Curve-Fit-based SPKF (CFBSPKF)

In this paper, the target position state is expressed as

$$p_i(t) = a_{i0}\phi_0(t) + a_{i1}\phi_1(t) + \cdots + a_{im}\phi_m(t),\ i = n, e.$$

The notations n and e represent the target positions in the north and east directions with respect to the inertial frame onboard the flying robot, respectively. Defining

$$\Theta_i \triangleq \begin{pmatrix} a_{i0} & a_{i1} & \cdots & a_{im} \end{pmatrix}^T \in \mathbb{R}^{(m+1)\times 1}$$

[1]Liang Sun is with Department of Mechanical and Aerospace Engineering, New Mexico State University, Las Cruces, NM, USA, 88003. lsun@nmsu.edu

[2]Daniel Pack is with College of Engineering and Computer Science, University of Tennessee at Chattanooga, Chattanooga, TN 37403. Daniel-Pack@utc.edu

TABLE I: O3SPKF

Hybrid nonlinear state-space model:

$$x(t) = A(t_0)\, x(t - t_0) + w_{t_0}(t),$$
$$y(t) = h(x(t), u(t)) + v(t),$$

where $w_{t_0}(t)$ and $v(t)$ are independent, zero-mean Gaussian noise processes of covariance matrices $\Sigma_{w_{t_0}}$ and Σ_v, respectively.
Definition: Let $p = 2 \times \dim(x(t))$.
Initialization: At time zero, set $t = 0$ and

$$\hat{x}^+(0) \triangleq \mathbb{E}[x(0)],$$
$$\Sigma^+_{\tilde{x}(0)} \triangleq \mathbb{E}\left[\left(x(0) - \hat{x}^+(0)\right)\left(x(0) - \hat{x}^+(0)\right)^T\right].$$

Computation: For each sample occurring out-of-order, (i.e., $t_m < t$) compute:
Initialize time pointers: $\Delta t = t - t_m$.
Delayed state and covariance:

$$\hat{x}^-(t_m) = A(\Delta t)\, \hat{x}^+(t),$$
$$\Sigma^-_{\tilde{x}(t_m)} = A(\Delta t)^{-1}\left(\Sigma^+_{\tilde{x}(t)} + \Sigma_{w_{t_0}}\right)\left(A(\Delta t)^{-1}\right)^T.$$

Delayed estimated output:

$$\begin{aligned}
\chi^-(t_m) &= \left\{\hat{x}^-(t_m), \hat{x}^-(t_m) + \gamma\sqrt{\Sigma^-_{\tilde{x}(t_m)}},\right. \\
&\quad \left.\hat{x}^-(t_m) - \gamma\sqrt{\Sigma^-_{\tilde{x}(t_m)}}\right\}, \\
\mathcal{Y}_i(t_m) &= h\left(\chi_i^-(t_m), u(t_m)\right), \\
\hat{y}(t_m) &= \sum_{i=0}^{p} \alpha_i^{(m)} \mathcal{Y}_i(t_m).
\end{aligned}$$

Update of the state and covariance at time t:

$$\begin{aligned}
\chi^+(t) &= \left\{\hat{x}^+(t), \hat{x}^+(t) + \gamma\sqrt{\Sigma^+_{\tilde{x}(t)}},\right. \\
&\quad \left.\hat{x}^+(t) - \gamma\sqrt{\Sigma^+_{\tilde{x}(t)}}\right\}, \\
\Sigma^-_{\tilde{y}(t_m)} &= \sum_{i=0}^{p} \alpha_i^{(c)} \left(\mathcal{Y}_i(t_m) - \hat{y}(t_m)\right) \\
&\quad \cdot \left(\mathcal{Y}_i(t_m) - \hat{y}(t_m)\right)^T + \Sigma_v, \\
\Sigma^-_{\tilde{x}(t)\tilde{y}(t_m)} &= \sum_{i=0}^{p} \alpha_i^{(c)} \left(\chi_i^+(t) - \hat{x}^+(t)\right) \\
&\quad \cdot \left(\mathcal{Y}_i(t_m) - \hat{y}(t_m)\right)^T, \\
L(t, t_m) &= \Sigma^-_{\tilde{x}(t)\tilde{y}(t_m)}\left(\Sigma^-_{\tilde{y}(t_m)}\right)^{-1}, \\
\hat{x}^+(t) &= \hat{x}^-(t) + L(t, t_m)\left(y(t_m) - \hat{y}(t_m)\right), \\
\Sigma^+_{\tilde{x}(t)} &= \Sigma^-_{\tilde{x}(t)} - L(t, t_m)\, \Sigma^-_{\tilde{y}(t_m)} L^T(t, t_m).
\end{aligned}$$

$$\Phi(t) = \begin{pmatrix} \phi_0(t) & \phi_1(t) & \cdots & \phi_m(t) \end{pmatrix}^T \in \mathbb{R}^{(m+1)\times 1}$$

we have

$$p_i(t) = \Phi^T(t)\, \Theta_i.$$

Similarly, the target velocity can be expresses as

$$\dot{p}_i(t) = \dot{\Phi}^T(t)\, \Theta_i.$$

Given the historical target states information, such as position and velocity, coefficient vectors Θ_i can be calculated accordingly. For real-time operation, we develop an Extended Kalman Filter to update Θ_i, which is essential since it does not require storing any historical state information. The Kalman filter recursively updates Θ_i as time progresses. Table III shows the algorithm for the new estimator. When delayed measurements become available, a fitted curve is generated and used to calculate the previous state estimates when the measurement was taken. As the time elapses, the target's trajectory may become increasingly long and the curve patterns complicated, making it necessary for the corresponding polynomial to have higher order terms. Since the latency of delayed measurements typically fall within some time period, we used a periodic mechanism to reset the Kalman filter to manage the order of a polynomial function.

Managing the order of each polynomial function necessitates dealing with multiple polynomials to accommodate a delayed measurement. That is, when we need to travel back in time spanned by more than a single polynomial, we need a method to retrieve coefficients of the previous polynomials. We currently use an algorithm that stores the polynomial coefficients of the previous period's fitted curve, which is summarized in Algorithm 1. This leads to the natural question of finding the correct time to reset the Kalman filter, starting a new polynomial function. The times should be adaptive based on the agility, speed, and predicted behavior of the target. In this paper, we assume that the covariance matrix at t_m, $\Sigma^-_{\tilde{x}(t_m)}$, can be determined by the covariance matrix at the current time, $\Sigma^-_{\tilde{x}(t)}$, and follow a proportional relationship, such as $\Sigma^-_{\tilde{x}(t_m)} = k\Sigma^-_{\tilde{x}(t)}$, where k is a positive constant gain and is tuned by users to obtain a satisfactory estimation error.

We then developed a new estimator which combines the fitted curve and the SKPF technique using a similar mechanism as O3SPKF and ASBSPKF. Table IV summarized the technique, called curve-fit-based SPKF (CFBSPKF). Similar to O3SPKF, when a delayed measurement, $y(t_m)$, is available, the state at the corresponding time, $\hat{x}(t_m)$, and corresponding covariance matrix, $\Sigma^-_{\tilde{x}(t_m)}$, are obtained using the information stored in the fitted curve and covariance matrix at the current time.

D. Computational cost

The additional computational cost of the proposed CFB-SPKF compared with O3SPKF depends only on the number of coefficients used by polynomials and not the reset period. That is, the complexity of the motion determines the computational cost, not the time to generate a polynomial. The additional computational cost introduced by the curve fitting Kalman filter is the same as a regular Kalman filter with the order of polynomial determining the size of the estimated states.

III. Simulation Results

A. Curve Fitting

To show the effectiveness of the Kalman filter used to find polynomials that model state trajectories of a dynamic system, we conducted a curve fitting simulation with a figure-8 shape trajectory. Select $\phi_i(t) = t^i$, $m = 10$, CurveFitPeriod $= 2$ sec, $T_s = 0.2$ sec, Delayed time $\Delta t =$

TABLE II: Augmented State Based SPKF

Augmented state vector: $x_a(t) = \left(x^T(t), x_{lag}^T\right)^T$, where $x_{lag} = x(t_m), t_m < t$.
Augmented nonlinear state-space model:

$$\dot{x}_a(t) = \begin{pmatrix} f(x(t), u(t)) + w(t) \\ \mathbf{0} \end{pmatrix},$$
$$y_{lag} = h\left(x_{lag}, u_{lag}\right) + v_{lag},$$

where $w(t)$ and v_{lag} are independent, zero-mean Gaussian noise processes of covariance matrices Σ_w and $\Sigma_{v_{lag}}$, respectively.
Initialization: At time zero, set $t = 0$,

$$\hat{x}^+(0) = \mathbb{E}[x(0)], \ \Sigma^+_{\tilde{x}(0)} = \mathbb{E}\left[\left(x(0) - \hat{x}^+(0)\right)\left(x(0) - \hat{x}^+(0)\right)^T\right].$$

If $t < t_m$, set $\hat{x}_{lag} = \mathbf{0}, \Sigma^+_{\tilde{x}_{lag}} = \mathbf{0}$.
If $t \geq t_m$, set $\hat{x}_{lag} = \mathbb{E}[x(t_m)], \Sigma^+_{\tilde{x}_{lag}} = \mathbb{E}\left[\left(x(t_m) - \hat{x}^+(t_m)\right)\left(x(t_m) - \hat{x}^+(t_m)\right)^T\right]$.
Definition: Let $p = 2 \times \dim(x(t))$ and $\Sigma^+_{\tilde{x}_a(t)} = \text{diag}\left\{\Sigma^+_{\tilde{x}(t)}, \Sigma^+_{\tilde{x}_{lag}}\right\}$.
Computation: , for each sample occurring out-of-order, (i.e., $t_m < t$) compute:
Delayed estimated output:

$$\chi^-(t_m) = \left\{\hat{x}^-(t_m), \hat{x}^-(t_m) + \gamma\sqrt{\Sigma^-_{\tilde{x}(t_m)}}, \hat{x}^-(t_m) - \gamma\sqrt{\Sigma^-_{\tilde{x}(t_m)}}\right\},$$
$$\mathcal{Y}_i(t_m) = h\left(\chi_i^-(t_m), u(t_m)\right),$$
$$\hat{y}(t_m) = \sum_{i=0}^{p} \alpha_i^{(m)} \mathcal{Y}_i(t_m).$$

Update of the state and covariance at time t:

$$\chi^+(t) = \left\{\hat{x}^+(t), \hat{x}^+(t) + \gamma\sqrt{\Sigma^+_{\tilde{x}(t)}}, \hat{x}^+(t) - \gamma\sqrt{\Sigma^+_{\tilde{x}(t)}}\right\},$$
$$\Sigma^-_{\tilde{y}(t_m)} = \sum_{i=0}^{p} \alpha_i^{(c)} \left(\mathcal{Y}_i(t_m) - \hat{y}(t_m)\right)\left(\mathcal{Y}_i(t_m) - \hat{y}(t_m)\right)^T + \Sigma_v,$$
$$\Sigma^-_{\tilde{x}(t)\tilde{y}(t_m)} = \sum_{i=0}^{p} \alpha_i^{(c)} \left(\chi_i^+(t) - \hat{x}^+(t)\right)\left(\mathcal{Y}_i(t_m) - \hat{y}(t_m)\right)^T,$$
$$\Sigma^-_{\tilde{x}(t_m)\tilde{y}(t_m)} = \sum_{i=0}^{p} \alpha_i^{(c)} \left(\chi_i^+(t_m) - \hat{x}^+(t_m)\right)\left(\mathcal{Y}_i(t_m) - \hat{y}(t_m)\right)^T,$$
$$L_a(t, t_m) = \begin{pmatrix} \Sigma^-_{\tilde{x}(t)\tilde{y}(t_m)}\left(\Sigma^-_{\tilde{y}(t_m)}\right)^{-1} \\ \Sigma^-_{\tilde{x}(t_m)\tilde{y}(t_m)}\left(\Sigma^-_{\tilde{y}(t_m)}\right)^{-1} \end{pmatrix},$$
$$\hat{x}_a^+(t) = \hat{x}_a^-(t) + L_a(t, t_m)\left(y(t_m) - \hat{y}(t_m)\right),$$
$$\Sigma^+_{\tilde{x}_a(t)} = \Sigma^-_{\tilde{x}_a(t)} - L_a(t, t_m)\Sigma^-_{\tilde{y}(t_m)} L_a^T(t, t_m).$$

$t - t_m = 0.4$ sec,

$$\hat{x}(t_m)^- = g_x\left(\hat{p}_i(t_m), \hat{\dot{p}}_i(t_m)\right) = \begin{pmatrix} \hat{p}_i(t) \\ \hat{\dot{p}}_i(t) \end{pmatrix}$$

$$\Sigma^-_{\tilde{x}(t_m)} = k\Sigma^-_{\tilde{x}(t)},$$

where k is tuned and selected as $k = 0.1$. The simulation results of three continuous segments for curving fitting are shown in Figure 1. The red dotted line shows the figure eight which we want our system to track. Scattered x's are measurements used to find coefficients of polynomials. The blue lines are the resulting ones generated using the parameters estimated by the Kalman filter. It can be seen that the fitted curve essentially captures the real motion of the target with noisy measurements.

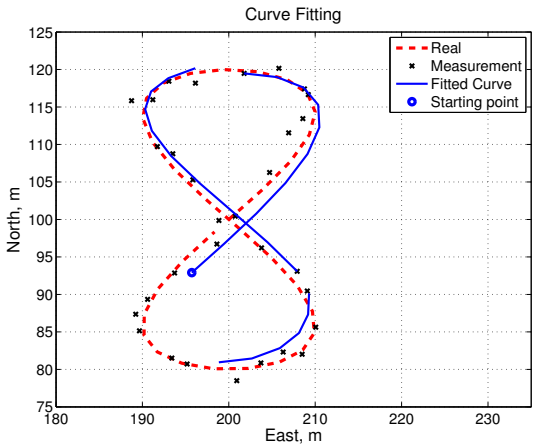


Fig. 1: Periodic curve fitting results for a figure-8 shape trajectory

TABLE III: Curve Fitting Using Kalman Filter

System Dynamics

$$\Theta_i[k+1] = \Theta_i[k] + w_{\Theta_i}[k],$$
$$\begin{pmatrix} p_i(t) \\ \dot{p}_i(t) \end{pmatrix} = \begin{pmatrix} \Phi_i^T(t) \\ \dot{\Phi}_i^T(t) \end{pmatrix} \Theta_i[k] + \begin{pmatrix} v_{p_i}(t) \\ v_{\dot{p}_i}(t) \end{pmatrix},$$

where $w_{\Theta_i}[k]$, $v_{p_i}(t)$, and $v_{\dot{p}_i}(t)$ are independent, zero-mean Gaussian noise processes of covariance matrices $\Sigma_{w_{\Theta_i}}$, $\Sigma_{v_{p_i}}$, and $\Sigma_{v_{\dot{p}_i}}$,respectively.

Definition:

$$\Sigma_v \triangleq \text{diag}\left\{\Sigma_{v_{p_i}}, \Sigma_{v_{\dot{p}_i}}\right\}.$$

$$\begin{aligned} \hat{\Theta}_i^+[k-1] &\triangleq \mathbb{E}[\Theta[k-1]], \\ \Sigma^+_{\tilde{\Theta}_i[k-1]} &\triangleq \mathbb{E}\left[\left(\Theta[k-1] - \hat{\Theta}^+[k-1]\right)\right. \\ &\quad \left.\left(\Theta[k-1] - \hat{\Theta}^+[k-1]\right)^T\right]. \end{aligned}$$

Initialization:

$$\hat{\Theta}_i^+[0] = \mathbf{1}^{(m+1)\times 1},$$
$$\Sigma^+_{\tilde{\Theta}[0]} = \mathbf{I}^{(m+1)\times(m+1)}.$$

Computation: For each sample occurring time t, compute:
Prediction:

$$\hat{\Theta}_i^-[k] = \hat{\Theta}_i^+[k-1],$$
$$\Sigma^-_{\tilde{\Theta}_i[k]} = \Sigma^+_{\tilde{\Theta}_i[k-1]} + \Sigma_{w_{\Theta_i}}.$$

Update:

$$C_i \triangleq \begin{pmatrix} \Phi_i^T(t) \\ \dot{\Phi}_i^T(t) \end{pmatrix},$$
$$L_i = \Sigma^-_{\tilde{\Theta}_i[k]} C_i^T \left(C_i \Sigma^-_{\tilde{\Theta}_i[k]} C_i^T + \Sigma_v\right)^{-1},$$
$$\hat{\Theta}_i^+[k] = \hat{\Theta}_i^-[k] + L_i\left(\begin{pmatrix} \hat{p}_i(t) \\ \hat{\dot{p}}_i(t) \end{pmatrix} - C_i \hat{\Theta}_i^-[k]\right),$$
$$\Sigma^+_{\tilde{\Theta}_i[k]} = (\mathbf{I} - L_i C_i)\,\Sigma^-_{\tilde{\Theta}_i[k]}.$$

Algorithm 1 Calculate Delayed State and Covariance Using Fitted Curve

Initialize: $\Theta_i^{old} = \mathbf{0}$, current elapse time $t = 0$.
CurveFitPeriod = the period time to reset the Kalman filter for curve fitting.
At each sample time T_s:
QuatientCurrent = interger part of t/CurveFitPeriod
ModulusCurrent = modulus of t/CurveFitPeriod
if ModulusCurrent$< T_s$ **then**
    $\Theta_i^{old} = \Theta_i$
end if
When the delayed measurement is available at t_m:
QuatientDelay = interger part of t_m/CurveFitPeriod
ModulusDelay = modulus of t_m/CurveFitPeriod
if QuatientDelay<QuatientCurrent **then**
    $\begin{pmatrix} \hat{p}_i(t_m) \\ \hat{\dot{p}}_i(t_m) \end{pmatrix} = \begin{pmatrix} \Phi_i^T(t_m) \\ \dot{\Phi}_i^T(t_m) \end{pmatrix} \Theta_i^{old}[k]$
else
    $\begin{pmatrix} \hat{p}_i(t_m) \\ \hat{\dot{p}}_i(t_m) \end{pmatrix} = \begin{pmatrix} \Phi_i^T(t_m) \\ \dot{\Phi}_i^T(t_m) \end{pmatrix} \Theta_i[k]$
end if
$\hat{x}(t_m)^- = g_x\left(\hat{p}_i(t_m), \hat{\dot{p}}_i(t_m)\right)$
$\Sigma^-_{\tilde{x}(t_m)} = g_{cov}\left(\Sigma^-_{\tilde{x}(t)}\right)$
return $\hat{x}(t_m)^-$ and $\Sigma^-_{\tilde{x}(t_m)}$

B. Comparison study using CFBSPKF, O3SPKF and ASB-SPKF

1) Different target motion patterns with the same time delay : To evaluate the system performance, we conducted target estimation simulations using the three techniques presented in Tables I, II and IV. We performed two different simulations with varying target behaviors. Namely, the target moved in a spiral shaped, or a random path. Figures 2 and 3 show the performance results using the three techniques when the targets moved in a spiral and random shapes, respectively. The average and standard deviation of the tracking errors are summarized in Table V. It can be seen that the CFBSPKF is superior to the other two techniques in all three simulations.

2) Different time delays with the same target motion: We also conducted simulations in estimating states for a target when it moved in a spiral shape path with varying delayed measurements of its state. The Figures 4, 2, and 5, show the performance results using the three techniques when the targets are moving in a spiral shape using measurements with time delays of 0.2 sec, 0.4 sec, and 0.8 sec, respectively. The average and standard deviation of the tracking errors are summarized in Table VI. Again, it can be seen that the CFBSPKF is superior to the other two techniques in all three simulations.

IV. Conclusion and Future Work

In this paper, we developed a curve fit-based state estimator to handle out-of-order and time-delayed sensor measurements. The simulations showed its effectiveness over two existing estimation techniques that deal with delayed measurement data.

References

[1] T. Larsen, N. Andersen, O. Ravn, and N. Poulsen, "Incorporation of time delayed measurements in a discrete-time kalman filter," in *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, vol. 4, Dec 1998, pp. 3972–3977.

[2] S. Björklund, "A survey and comparison of time-delay estimation methods in linear systems," Ph.D. dissertation, Department of Electrical Engineering, Linköping University, Dec 2003.

[3] S. Julier and J. Uhlmann, "Fusion of time delayed measurements with uncertain time delays," in *American Control Conference*, vol. 6, June 2005, pp. 4028–4033.

[4] M. Choi, J. Choi, J. Park, and W. K. Chung, "State estimation with delayed measurements considering uncertainty of time delay," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 3987–3992.

[5] M. Choi, J. Choi, and W. Chung, "State estimation with delayed measurements incorporating time-delay uncertainty," *Control Theory Applications, IET*, vol. 6, no. 15, pp. 2351–2361, Oct 2012.

[6] A. Aguiar and J. Hespanha, "State estimation for continuous-time systems with perspective outputs from discrete noisy time-delayed measurements," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 3, Dec 2004, pp. 3126–3131.

[7] ——, "State estimation of continuous-time systems with implicit outputs from discrete noisy time-delayed measurements," in *Decision and Control, 2006 45th IEEE Conference on*, Dec 2006, pp. 4781–4786.

TABLE IV: Curve-Fit-Based SPKF

Nonlinear state-space model:

$$\dot{x}(t) = f(x(t), u(t)) + w(t),$$
$$y(t_m) = h(x(t_m), u(t_m)) + v(t_m),$$

where $w(t)$ and $v(t_m)$ are independent, zero-mean Gaussian noise processes of covariance matrices Σ_w and Σ_v, respectively.
Definition: Let $p = 2 \times \dim(x(t))$.

$$\hat{x}^+(t) = \mathbb{E}[x(t)],$$
$$\Sigma^+_{\tilde{x}(t)} = \Sigma^+_{\tilde{x}(t)}.$$

Computation: For each sample occurring out-of-order, (i.e., $t_m < t$) compute:
Delayed state estimate and covariance: $\hat{x}^-(t_m)$ and $\Sigma^-_{\tilde{x}(t_m)}$ are obtained by Algorithm 1.
Delayed estimated output:

$$\begin{aligned}
\chi^-(t_m) &= \left\{\hat{x}^-(t_m), \hat{x}^-(t_m) + \gamma\sqrt{\Sigma^-_{\tilde{x}(t_m)}}, \hat{x}^-(t_m) - \gamma\sqrt{\Sigma^-_{\tilde{x}(t_m)}}\right\},\\
\mathcal{Y}_i(t_m) &= h\left(\chi^-_i(t_m), u(t_m)\right),\\
\hat{y}(t_m) &= \sum_{i=0}^{p} \alpha_i^{(m)} \mathcal{Y}_i(t_m).
\end{aligned}$$

Update of the state and covariance at time t***:***

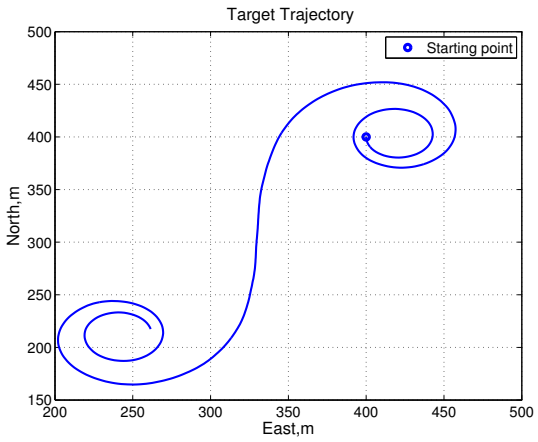
$$\begin{aligned}
\chi^+(t) &= \left\{\hat{x}^+(t), \hat{x}^+(t) + \gamma\sqrt{\Sigma^+_{\tilde{x}(t)}}, \hat{x}^+(t) - \gamma\sqrt{\Sigma^+_{\tilde{x}(t)}}\right\},\\
\Sigma^-_{\tilde{y}(t_m)} &= \sum_{i=0}^{p} \alpha_i^{(c)} \left(\mathcal{Y}_i(t_m) - \hat{y}(t_m)\right)\left(\mathcal{Y}_i(t_m) - \hat{y}(t_m)\right)^T + \Sigma_v,\\
\Sigma^-_{\tilde{x}(t)\tilde{y}(t_m)} &= \sum_{i=0}^{p} \alpha_i^{(c)} \left(\chi^+_i(t) - \hat{x}^+(t)\right)\left(\mathcal{Y}_i(t_m) - \hat{y}(t_m)\right)^T,\\
L(t, t_m) &= \Sigma^-_{\tilde{x}(t)\tilde{y}(t_m)} \left(\Sigma^-_{\tilde{y}(t_m)}\right)^{-1},\\
\hat{x}^+(t) &= \hat{x}^-(t) + L(t, t_m)\left(y(t_m) - \hat{y}(t_m)\right),\\
\Sigma^+_{\tilde{x}(t)} &= \Sigma^-_{\tilde{x}(t)} - L(t, t_m)\,\Sigma^-_{\tilde{y}(t_m)} L^T(t, t_m).
\end{aligned}$$

TABLE V: Summary of target estimation errors using 0.4 sec delayed measurements

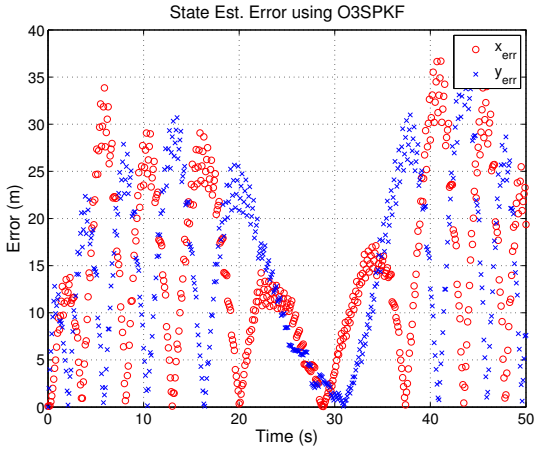
Target Path	Estimator	avg_x (m)	std_x (m)	avg_y (m)	std_y (m)
Spiral	O3SPKF	14.42	9.16	14.68	9.27
	ASBSPKF	11.17	7.92	10.65	7.47
	CFBSPKF	7.98	5.42	8.38	5.14
Random	O3SPKF	10.35	6.46	12.61	9.59
	ASBSPKF	7.96	5.20	7.91	7.79
	CFBSPKF	6.49	4.42	7.18	6.30
avg=average, std= standard deviation.					

TABLE VI: Summary of estimation errors for a target in a spiral path with varying time delays

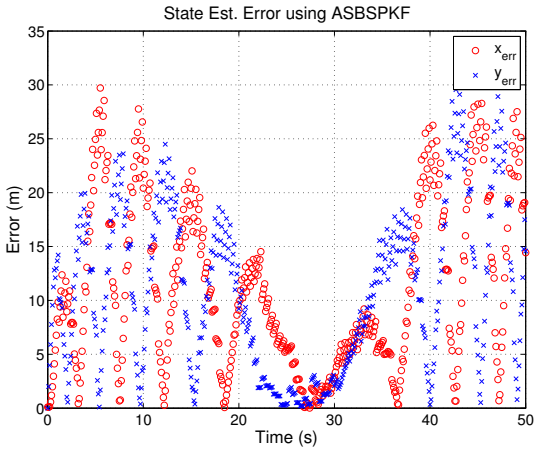
Time delay	Estimator	avg_x (m)	std_x (m)	avg_y (m)	std_y (m)
0.2 sec	O3SPKF	10.17	6.51	10.06	6.57
	ASBSPKF	6.14	4.58	6.47	4.64
	CFBSPKF	4.59	2.94	4.83	3.06
0.4 sec	O3SPKF	14.42	9.16	14.68	9.27
	ASBSPKF	11.17	7.92	10.65	7.47
	CFBSPKF	7.98	5.42	8.38	5.14
0.8 sec	O3SPKF	20.83	12.79	21.85	13.11
	ASBSPKF	19.38	12.94	18.20	12.71
	CFBSPKF	14.07	9.99	15.27	9.66
avg=average, std= standard deviation.					



(a) Target trajectory of a spiral shape.



(b) O3SPKF.



(c) ASBSPKF.

(d) CFBSPKF.

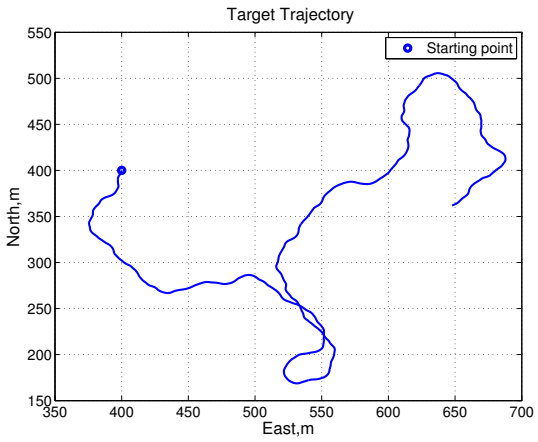
Fig. 2: State estimation errors of a target following (a) a spiral shape trajectory using (b) O3SPKF, (c) ASBSPKF, and (d) CFBSPKF with 0.4 sec delayed measurements.

[8] M. Mauthner, W. Elmenreich, A. Kirchner, and D. Boesel, "Out-of-sequence measurement treatment in sensor fusion applications: Buffering versus advanced algorithms," in *Workshop Fahrerassistenzsysteme*, Löwenstein/Hößlinsülz, Canada, 2006, pp. 20–30.

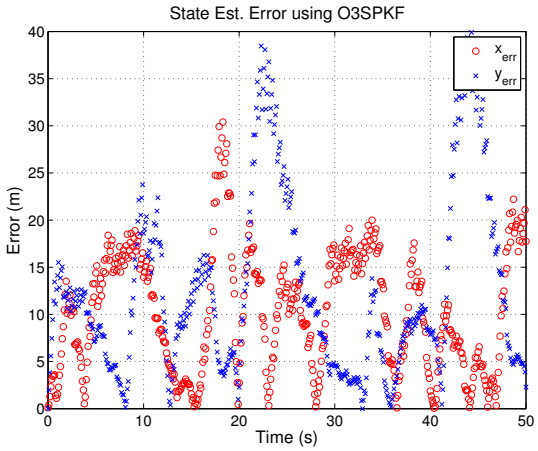
[9] W. Wang, H. Zhang, and L. Xie, "Kalman filtering for continuous-time systems with time-varying delay," *IET control theory & applications*, vol. 4, no. 4, pp. 590–600, 2010.

[10] G. L. Plett, D. Zarzhitsky, and D. J. Pack, "Out-of-order sigma-point kalman filtering for target localization using cooperating unmanned aerial vehicles," in *Advances in Cooperative Control and Optimization*, ser. Lecture Notes in Control and Information Sciences, P. Pardalos, R. Murphey, D. Grundel, and M. Hirsch, Eds. Springer Berlin Heidelberg, 2007, vol. 369, pp. 21–43.

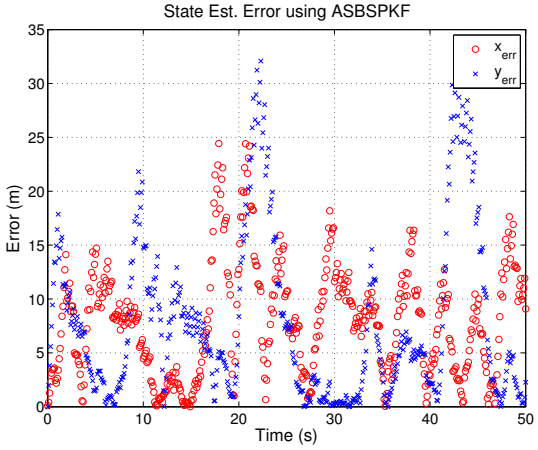
[11] R. Van Der Merwe, E. A. Wan, S. Julier, *et al.*, "Sigma-point kalman filters for nonlinear estimation and sensor-fusion: Applications to integrated navigation," in *Proceedings of the AIAA Guidance, Navigation & Control Conference*, 2004, pp. 16–19.



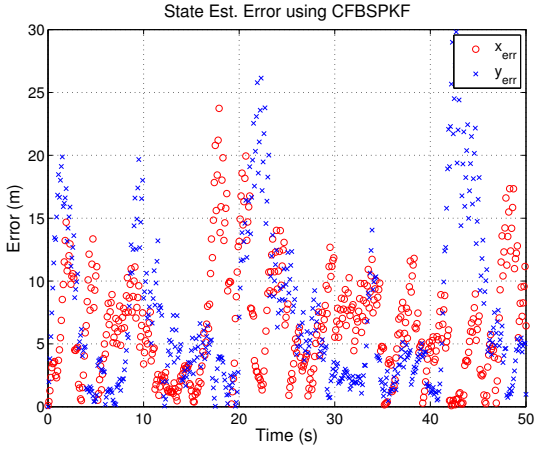
(a) Target trajectory of a spiral shape.



(b) O3SPKF.

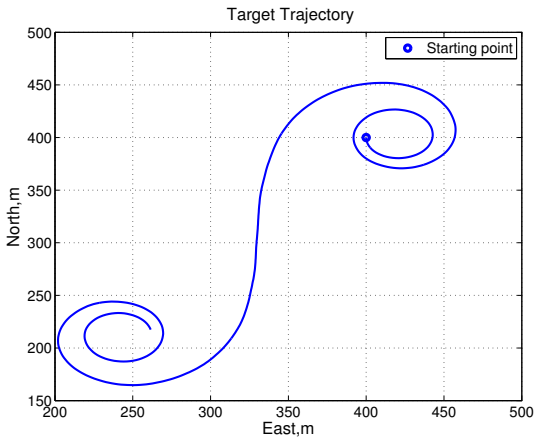


(c) ASBSPKF.

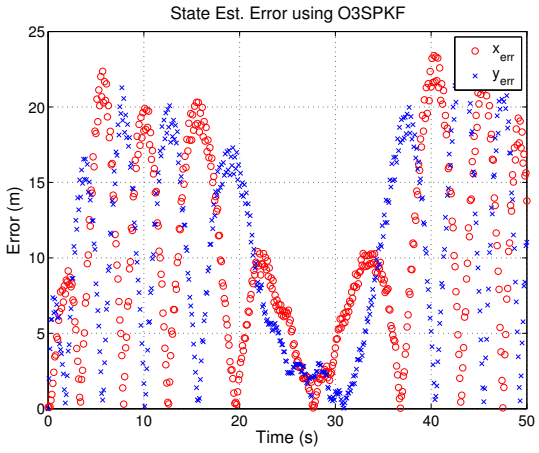


(d) CFBSPKF.

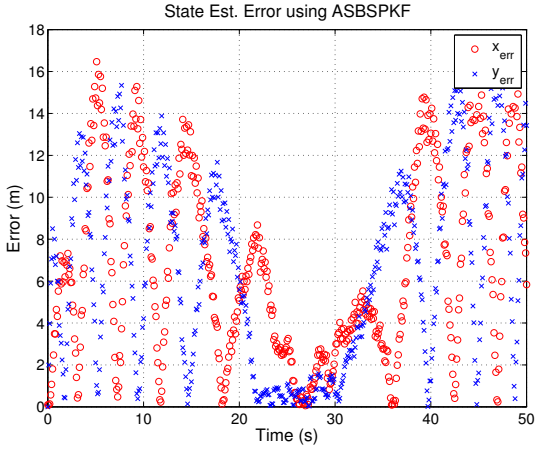
Fig. 3: State estimation errors of a target following (a) a random trajectory using (b) O3SPKF, (c) ASBSPKF, and (d) CFBSPKF with 0.4 sec delayed measurements.



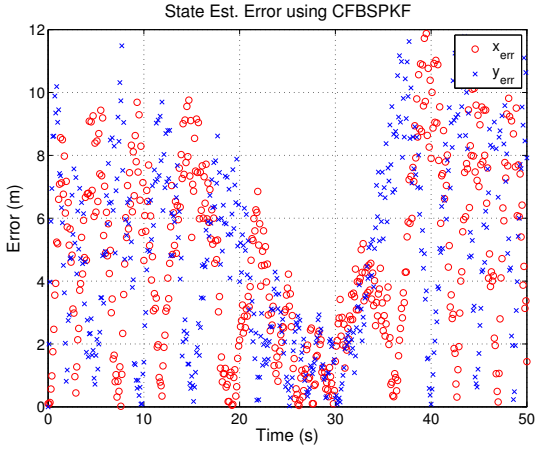
(a) Target trajectory of a spiral shape.



(b) O3SPKF.

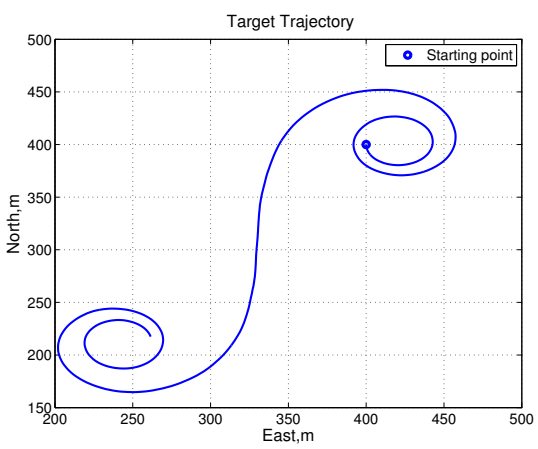


(c) ASBSPKF.

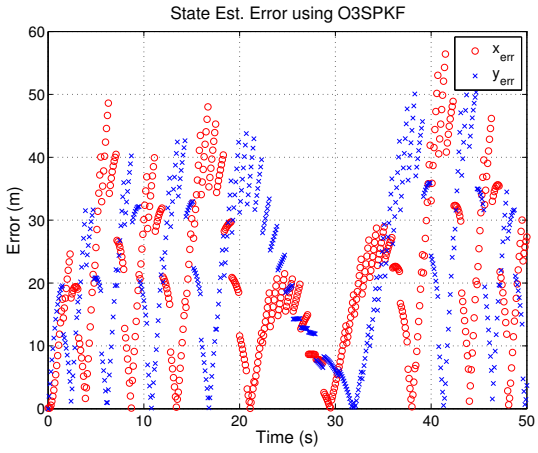


(d) CFBSPKF.

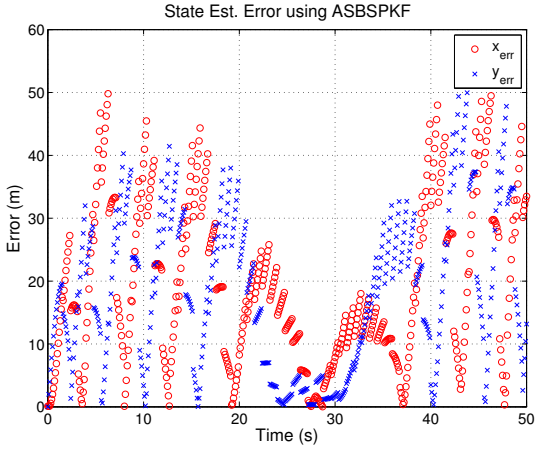
Fig. 4: State estimation errors of a target following (a) a spiral shape trajectory using (b) O3SPKF, (c) ASBSPKF, and (d) CFBSPKF with 0.2 sec delayed measurements.



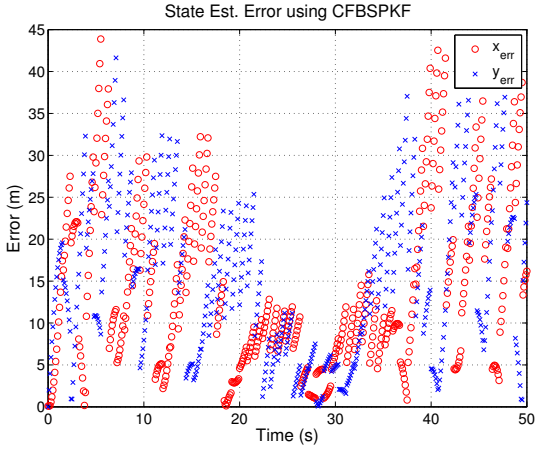
(a) Target trajectory of a spiral shape.



(b) O3SPKF.



(c) ASBSPKF.



(d) CFBSPKF.

Fig. 5: State estimation errors of a target following (a) a spiral shape trajectory using (b) O3SPKF, (c) ASBSPKF, and (d) CFBSPKF with 0.8 sec delayed measurements.