International Journal of Control, Automation and Systems 18(3) (2020) 643-649
http://dx.doi.org/10.1007/s12555-018-0938-4
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Gaussian Sum FIR Filtering for 2D Target Tracking

Jung Min Pak

Abstract: The constant velocity (CV) motion model has been typically used in 2-dimensional (2D) target tracking problems, but it has an uncertain process noise covariance problem. Unlike the Kalman filter (KF), the least square finite impulse response filter (LSFF) does not require noise covariance information and can overcome the uncertain process noise covariance problem. However, the LSFF has a cumbersome problem that is to select a suitable value of design parameter called the horizon size. This paper proposes a Gaussian sum FIR filter (GSFF), where the Gaussian sum method is used to deal with the horizon size in LSFFs. The GSFF overcomes the uncertain process noise covariance problem and can be alternative to existing filters in 2D target tracking. Superior performance of GSFF is demonstrated by comparison with the Gaussian sum KF (GSKF) that is an existing filter to solve the uncertain process noise covariance problem.

Keywords: Finite impulse response (FIR) filter, Gaussian sum, Gaussian sum FIR filter (GSFF), target tracking.

1. INTRODUCTION

Over the past few decades, state estimators, such as the Kalman (KF) and particle (PF) filters, have been developed for target tracking problems and successfully applied to surveillance/navigation systems [1–8]. Measurement systems are represented by nonlinear equations in many target tracking problems, hence nonlinear estimators, such as the extended Kalman filter (EKF) [5, 6], unscented Kalman filter [7], and PF [3, 4, 8], have been intensively studied. However, linear estimators, e.g., KF, have been often used in combination with measurement conversion or for image based target tracking applications [9–11].

The Kalman and particle filters are Bayesian filters that recursively perform time update and measurement update processes to iteratively refine a rough initial state estimate. However, the recursive filter structure also accumulates modeling and/or computational errors, which may lead to the filter divergence. Thus, estimators having finite memory structure [12–17] have been developed to overcome this divergence. In this regard, finite impulse response (FIR) filters have been intensively studied [13, 14, 16, 18–27] and generally provide superior performance, such as robustness against model parameter uncertainties and fast tracking for abrupt target motion changes.

Various FIR filters have been developed and applied to tracking applications, such as human/robot localization [20,26,27], frequency tracking [23], visual object tracking [17,24], and target tracking [18,25]. The constant velocity (CV) motion model is commonly employed for tracking applications. However, the CV model has a drawback that the design parameter, process noise covariance, is very uncertain and inappropriate selection of process noise covariance may worsen tracking performance [25, 28]. On the other hand, FIR filters are robust against process noise covariance uncertainty from the CV model compared with KF [17, 23–25]. Thus, FIR filter may be a good alternative to KF for tracking applications using the CV model.

Although the FIR filter has better robustness than KF, it is difficult to handle. The FIR filter uses a finite set of measurements (in the discrete-time case), called the horizon or memory size. The horizon size is an uncertain design parameter for FIR filtering and several methods have been proposed to select the most appropriate parameter. The method to find optimal horizon size proposed in [29] is appropriate for time-invariant systems, and the method to adaptively adjust horizon size proposed in [21] can cope with time-varying systems, but requires significantly more computation time compared with KF.

This paper proposes a Gaussian sum FIR filter (GSFF) for tracking applications. We focus on two-dimensional (2D) target tracking where a target moves in a 2D space and employ the 2D CV model. We use the Gaussian sum filtering method to solve the horizon size problem. In the proposed GSFF algorithm, several FIR filters using different horizon sizes are operated in parallel. At each time step, outputs of the FIR filters are merged together using

Manuscript received December 26, 2018; revised April 4, 2019; accepted July 30, 2019. Recommended by Associate Editor Young Soo Suh under the direction of Editor Chan Gook Park. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2017R1C1B5017021).

Jung Min Pak is with the Department of Electrical Engineering, Wonkwang University, 460 Iksandae-ro, Iksan, Jeonbuk 54538, Korea (e-mail: destin11@wku.ac.kr).

the Gaussian sum method. The resulting GSFF has robustness against uncertainties in process noise covariance when using the CV model in 2D target tracking problem. We compare the proposed GSFF with the Gaussian sum KF (GSKF) that is an existing filter to solve the uncertain process noise covariance problem. Superior performance of the GSFF is demonstrated by simulations in comparisons with the GSKF.

Novelty and contribution of this paper can be summarized as follows. The GSFF, which is a new FIR filter that solves the horizon size problem by Gaussian sum method, was firstly proposed. The proposed GSFF overcomes the uncertain process noise problem in 2D target tracking. Estimation accuracy of the GSFF is superior to the existing filters, KF and GSKF. In addition, the GSFF is computationally efficient compared with the GSKF.

The remainder of this paper is organized as follows: Section 2 briefly explains the 2D target tracking scheme and formulates the problem to be solved. Section 3 provides the proposed GSFF details and Section 4 presents simulation results. Section 5 summarizes and concludes the paper.

2. 2D TARGET TRACKING SCHEME AND PROBLEM FORMULATION

We assume that a target moves in 2D space in diverse directions at different speeds, where target direction and speed change arbitrarily and unpredictably. The CV motion model is often used to represent target motion for this case, assuming target velocity is constant over a short time interval, i.e., the target moves straight at constant speed between neighbouring points. We denote target 2D positions and velocities at time step k by (x_k, y_k) and $(\dot{x}_k, \dot{y}_k)$, respectively, and the position at next time step $k+1$ can be expressed as

$$x_{k+1} = x_k + \dot{x}_k T, \quad (1)$$

$$y_{k+1} = y_k + \dot{y}_k T, \quad (2)$$

where T is the time interval between time steps k and $k+1$. The 2D positions and velocities can be combined by defining the state vector, $\mathbf{x}_k = [x_k\ y_k\ \dot{x}_k\ \dot{y}_k]^T$. Thus, the CV motion model in state-space form is

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{G}\mathbf{w}_k, \quad (3)$$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} T^2/2 & 0 \\ 0 & T^2/2 \\ T & 0 \\ 0 & T \end{bmatrix}, \quad (4)$$

where $\mathbf{w}_k$ is the process noise vector. We assume that $\mathbf{w}_k$ is zero-mean white Gaussian noise with the covariance matrix, $\mathbf{Q}$. In the CV model, $\mathbf{Q}$ is a key parameter related to the amount of change of target motion. Increasing $\mathbf{Q}$ means increasing target velocity (course and speed) change.

We assume that 2D measurements (i.e., x- and y-coordinates of the target) are available and define the measurement vector at time k as $\mathbf{y}_k = [x_k\ y_k]$. Thus, the measurement model can be expressed as

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k, \quad (5)$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad (6)$$

where $\mathbf{v}_k$ is the measurement noise vector with the covariance matrix $\mathbf{R}$.

Measurements are corrupted by noise, hence the measured 2D target positions are not accurate. The KF is typically used to filter the noise and obtain more accurate 2D positions, recursively performing prediction and correction steps. The prediction step produces an *a priori* estimated state, $\mathbf{x}_k^-$, using the motion model and the correction step produces an *a posteriori* estimated state, $\mathbf{x}_k^+$, using the measurements and measurement model. KF for 2D target tracking can be expressed as

$$\mathbf{x}_k^- = \mathbf{A}\mathbf{x}_{k-1}^+, \quad (7)$$

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}^+\mathbf{A}^T + \mathbf{Q}, \quad (8)$$

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{C}^T(\mathbf{C}\mathbf{P}_k^-\mathbf{C}^T + \mathbf{R})^{-1}, \quad (9)$$

$$\mathbf{x}_k^+ = \mathbf{x}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{C}\mathbf{x}_k^-), \quad (10)$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k\mathbf{C})\mathbf{P}_k^-, \quad (11)$$

where $\mathbf{P}_k$ is estimation error covariance matrix, $\mathbf{K}_k$ is Kalman gain, and superscripts $-$ and $+$ mean *a priori* and *a posteriori*, respectively.

Equation (8) shows that KF requires $\mathbf{Q}$, which is inherently uncertain information, because no prior information regarding target movement is available. This $\mathbf{Q}_k$ uncertainty causes mismatch between model and real target motion, which degrades KF estimation accuracy.

The least square FIR filter (LSFF) [25] was proposed to overcome this problem for target tracking, since it does not require noise covariance information, and is robust against $\mathbf{Q}$ uncertainty in the CV model. LSFF performs batch processing, and produces estimated state, $\hat{\mathbf{x}}_k$, using recent finite measurements on the time horizon $[m, n]$, where $m = k - N$ and $n = k - 1$ are the horizon initial and final time steps, respectively; and N is the horizon size, i.e., the number of measurements on the horizon. Thus, the LSFF for 2D target tracking can be expressed as

$$\hat{\mathbf{x}}_k = \mathbf{H}_N\mathbf{Y}_n, \quad (12)$$

$$\mathbf{H}_N \triangleq (\widetilde{\mathbf{C}}_N^T\widetilde{\mathbf{C}}_N)^{-1}\widetilde{\mathbf{C}}_N^T, \quad (13)$$

$$\widetilde{\mathbf{C}}_N \triangleq \begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \vdots \\ \mathbf{C}\mathbf{A}^{N-1} \end{bmatrix}\mathbf{A}^{-N}, \quad (14)$$

$$\mathbf{Y}_n \triangleq [\mathbf{y}_m^T\ \mathbf{y}_{m+1}^T\ \cdots\ \mathbf{y}_n^T]^T, \tag{15}$$

where $\mathbf{H}_N$ is filter gain and subscript N means that the gain depends on the horizon size. $\mathbf{H}_N$ is time-invariant, and is computed only once in the filtering process, which relieves computational burden.

Although LSFF is robust against $\mathbf{Q}$ uncertainty, it has a cumbersome problem. LSFF uses recent finite measurements on the time horizon $[k-N, k-1]$ to produce $\hat{\mathbf{x}}_k$. Thus, N has significant effects on estimation performance, and optimal N is not constant but changes according to various model parameters, such as process and measurement noise magnitudes, sampling interval, etc. If the motion and measurement models are accurate, N can be chosen by simulation to minimize estimation errors. However, 2D target tracking has high $\mathbf{Q}$ uncertainty and hence selecting an appropriate LSFF horizon size is difficult. Thus, the problem is to find a method to deal with the LSFF horizon size. By solving this problem, we obtain a new state estimator offering reliable performance for 2D target tracking without requiring suitable $\mathbf{Q}$ and N values.

3. GAUSSIAN SUM FIR FILTER

This section proposes GSFF to solve the problem formulated in Section 2. GSFF is based on Gaussian sum approximation [30, 31], where the estimated state probability density function (pdf) is approximated by a weighted sum of Gaussian density functions [32]. Let the estimated state pdf be a conditional density, $p(\mathbf{x}_k|\mathbf{Y}_k)$ and $\mathbf{Y}_k \triangleq \{\mathbf{y}_i,\ i=1,\cdots,k\}$ be the sequence of measurements up to time k. Then, the Gaussian sum approximation can be expressed as

$$p(\mathbf{x}_k|\mathbf{Y}_k) \approx \sum_{i=1}^{M} w_k^i \mathcal{N}(\mathbf{x}_k^i; \mathbf{x}_{k|k}^i, \mathbf{P}_{k|k}^i), \tag{16}$$

where w_k^i are weights, $\mathcal{N}(\mathbf{x}_k^i; \mathbf{x}_{k|k}^i, \mathbf{P}_{k|k}^i)$ are Gaussian densities with mean $\mathbf{x}_{k|k}^i$ and variance $\mathbf{P}_{k|k}^i$, and M is the number of Gaussian densities [32].

The GSFF runs several independent LSFFs in parallel, and merges the outputs using the Gaussian approximation. The independent LSFFs use different horizon sizes, and the merged output contains various characteristics of the horizon sizes. By determining the range of horizon size as $[N_{\min}, N_{\max}]$ and the number of LSFFs N_F, the horizon sizes for the LSFFs are determined as

$$N_i = N_{\min} + i \times \rho,\ \ i = 1, 2, \cdots, N_F, \tag{17}$$

where $\rho \triangleq (N_{\max} - N_{\min})/N_F$.

Design parameters $N_{\min}$, $N_{\max}$, and N_F affect GSFF performance. However, GSFF avoids the difficult and cumbersome task of selecting FIR filter horizon size. Thus, rather than discussing $N_{\min}$, $N_{\max}$, and N_F selection, in detial, we demonstrate that GSFF with roughly selected parameters outperforms both KF and conventional FIR filter.

Algorithm 1: Filtering using GSFF

Data: $N_i (i = 1, 2, \cdots, N_F)$
Result: $\hat{\mathbf{x}}_k$
begin
  - Compute gain matrices $\mathbf{H}_{N_i}$ for horizon sizes $N_i (i = 1, 2, \cdots, N_F)$ using the following equations:
  $\widetilde{\mathbf{C}}_{N_i} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA}, \\ \vdots \\ \mathbf{CA}^{N_i - 1} \end{bmatrix} \mathbf{A}^{-N_i}$
  $\mathbf{H}_{N_i} = (\widetilde{\mathbf{C}}_{N_i}^T \widetilde{\mathbf{C}}_{N_i})^{-1} \widetilde{\mathbf{C}}_{N_i}^T$,
  for $k = N_1 + 1, N_1 + 2, \cdots$ **do**
    for $i = 1, 2, \cdots, N_F$ **do**
      if $k == N^i + 1$ **then**
        - Generate N_i weight variables and set them to $1/N_i$.
      end if
      if $k > N^i$ **then**
        - Construct the augmented measurement matrix $\mathbf{Y}_{N_i,k}$:
        $\mathbf{Y}_{N_i,k} = [\mathbf{y}_{k-N_i}^T\ \mathbf{y}_{k-N_i+1}^T\ \cdots\ \mathbf{y}_{k-1}^T]^T$
        - Obtain the state estimate of the i-th LSFF, $\hat{\mathbf{x}}_k^i$:
        $\hat{\mathbf{x}}_k = \mathbf{H}_{N_i} \mathbf{Y}_{N_i,k}$
        - Compute the likelihood using the following equations:
        $\hat{\mathbf{y}}_k^i = \mathbf{C}\hat{\mathbf{x}}_k^i,\ p(\mathbf{y}_k|i) = \frac{1}{2\pi|\mathbf{R}|^{1/2}} \exp[(\mathbf{y}_k - \hat{\mathbf{y}}_k^i)^{\mathrm{T}} \mathbf{R}^{-1} (\mathbf{y}_k - \hat{\mathbf{y}}_k^i)]$,
      end if
    end for
    for $i = 1, 2, \cdots, N_F$ **do**
      - Update the weight of the i-th LSFF:
      $w_k^i = \frac{p(\mathbf{y}_k|i) w_{k-1}^i}{\sum_{j=1} N_F\, p(\mathbf{y}_k|j) w_{k-1}^i}$
    end for
    - Obtain the state estimate of the GSFF at time k by Gaussian sum:
    $\mathbf{x}_k = \sum_{i=1}^{N_F} w_k^i \hat{\mathbf{x}}_k^i$
  end for
end

Once N_F is determined, the weights for Gaussian sum approximation shown in (16) are initialized as

$$w_0^i = \frac{1}{N_F},\ \ i = 1, 2, \ldots, N_F, \tag{18}$$

and weights at time k are computed using the update rule [32]

$$w_k^i = \frac{p(\mathbf{y}_k|i) w_{k-1}^i}{\sum_{j=1} N_F\, p(\mathbf{y}_k|j) w_{k-1}^i}, \tag{19}$$

where $p(\mathbf{y}_k|i)$ is the likelihood of measurement $\mathbf{y}_k$ given $\hat{\mathbf{x}}_k^i$, which is obtained from the i-th LSFF. Assuming Gaussian measurement noise, the likelihood $p(\mathbf{y}_k|i)$ is computed as follows:

$$p(\mathbf{y}_k|i) = \frac{1}{2\pi|\mathbf{R}|^{1/2}}\exp[(\mathbf{y}_\mathrm{k}-\hat{\mathbf{y}}_\mathrm{k}^\mathrm{i})^\mathrm{T}\mathbf{R}^{-1}(\mathbf{y}_\mathrm{k}-\hat{\mathbf{y}}_\mathrm{k}^\mathrm{i})], \quad (20)$$

$$\hat{\mathbf{y}}_k^i = \mathbf{C}\hat{\mathbf{x}}_k^i, \quad (21)$$

where $|\mathbf{R}|$ is the determinant of $\mathbf{R}$.

The LSFFs produce $\hat{\mathbf{x}}_k^i$ $(i = 1, 2, \cdots, N_F)$ at each time step. Thus, the GSFF state estimate at time k is computed as

$$\mathbf{x}_k = \sum_{i=1}^{N_F} w_k^i \hat{\mathbf{x}}_k^i. \quad (22)$$

The overall process of filtering using the GSFF is summarized in Algorithm 1.

4. SIMULATION

This section presents simulation results to demonstrate the proposed GSFF performance. We simulate 2D target tracking for a single target and use the CV model described by (3). The simulation generates target trajectories (i.e., target position sequences) using (3).

We also use the CV model for the filters. Although KF requires $\mathbf{Q}$, FIR filters (i.e., GSFF and LSFF) do not. Since target motion is unpredictable, $\mathbf{Q}$ is actually unknown, and it is difficult to select suitable $\mathbf{Q}$ for KF. Thus, we test several $\mathbf{Q}$ hypotheses in the simulation. We define $\mathbf{Q}$ as $\mathbf{Q} = q\mathbf{I}_2$, where $\mathbf{I}_2$ is a 2×2 identity matrix, and set $q = 1$ when generating target trajectories. We test three q values (0.1, 1, and 10) for the KF.

The measurement comprises the 2D target position, i.e., x- and y-coordinates. If the measurement vector at time k is $\mathbf{y}_k = [x_k\ y_k]$, the measurement equation is described by (5) and (6), and we set $\mathbf{R} = \mathbf{I}_2$.

Horizon size is an important design parameter of FIR filters, and can significantly affect estimation performance. If the horizon size is unsuitable, estimation accuracy of the FIR filter can be significantly worsened. Although LSFF requires appropriate horizon size, GSFF relieves this requirement. Since the suitable horizon size is unknown, we tested three horizon sizes (10, 20, and 30) for the single LSFF. In this case, the GSFF is the Gaussian mixture of three LSFFs using the three horizon sizes. We compared single LSFF and GSFF performances. Next, we demonstrated that GSFF is more accurate than GSKF by simulation. GSKF overcomes q uncertainty in the CV motion model. GSKF is obtained by replacing LSFFs in GSFF algorithm with KFs.

First, we compare single KF and GSKF performance. Single KF uses constant q values, 0.1, 1, and 10, and GSKF is the Gaussian mixture of these three KFs. Lastly, we compare the GSFF and the GSKF performance.

In the simulation, the target state was generated using the CV model with $q = 1$ for the whole simulation time, $1 \le k \le 600$. We set sampling interval, $T = 0.1$ s, and the simulation time was 60 s. Filter performance was evaluated by root mean square position error (RMSPE) and root time averaged mean square error (RTAMSE). We ran 100 Monte Carlo (MC) simulations and RMSPE and RTAMS were calculated as

$$\mathbf{RMSPE}_k = \sqrt{\frac{1}{100}\sum_{i=1}^{100}(x_k^i - \hat{x}_k^i)^2 + (y_k^i - \hat{y}_k^i)^2}, \quad (23)$$

$$\mathbf{RTAMS}_k = \sqrt{\frac{1}{100}\sum_{k=1}^{100}\sum_{i=1}^{M}(x_k^i - \hat{x}_k^i)^2 + (y_k^i - \hat{y}_k^i)^2}. \quad (24)$$

If RMSPE of a filter is larger than that of the measured output, there is no need to use the filter. Hence, we used RMSPE of the measured output to evaluate filter effectiveness.

Fig. 1(a) compares GSKF and standard KFs using $q = 10$, 1, and 0.1. All KFs initially exhibit significantly large RMSPE that decrease over time, converging to relatively constant value. RMSPE for both $q_k = 1$ and $q_k = 10$ are larger than those of the measured output for early stages but become smaller than those after a few seconds, although $q_k = 1$ exhibits slower convergence than $q_k = 10$. However, after convergence (approximately 15 s after simulation starts), KF with $q_k = 1$ results in the smallest RMSPE because this process noise covariance corresponds to the real one (i.e., the q_k used to generate the target trajectories). Thus, the KF with $q_k = 1$ is the ideal KF, which is difficult to obtain for real experiments. GSKF is a Gaussian mixture of three KFs using $q = 10$, 1, and 0.1. Because the KF with $q = 10$ exhibits the smallest RMSPE in the initial stage, the weight for this KF increases. Weight for the KF with $q = 10$ converges to 1 and weights for other KFs converge to 0. After the convergence is completed, GSKF output equals to the KF with $q = 10$. Hence, GSKF RMSPE becomes the same as that of the KF with $q = 10$. About 12 seconds after the simulation start, the KF with $q_k = 1$ exhibits the smallest RMSPE. However, GSKF RMSPE is still the same as the KF with $q_k = 10$. This is because weights do not change after completion of convergence.

Fig. 1(b) compares GSFF and conventional FIR filters, i.e., single LSFF with $N = 10$, 20, and 30. The large RMSPEs for early stage KFs do not occur for the FIR filters, and three LSFFs exhibit smaller RMSPE compared with measured output. In this simulation, $N = 30$ provided the best performance in terms of RMSPE. Although $N = 30$ is the best horizon size among the three considered, it is not the optimal horizon size, N_opt. N_opt can be estimated by Monte Carlo simulation as the horizon size producing minimum RMSPE. However, in practise the best

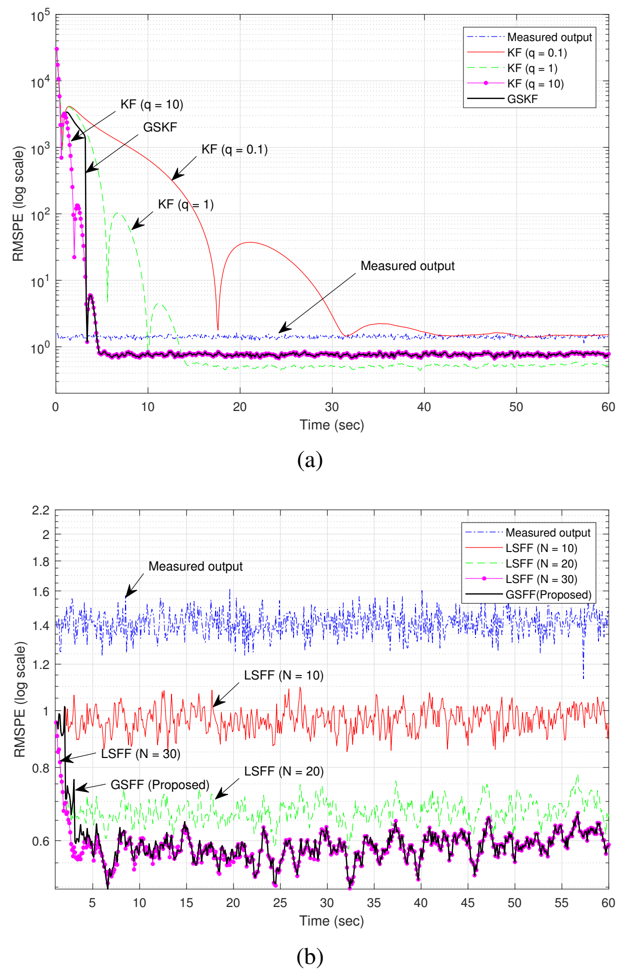


Fig. 1. Root mean square position error (RMSPE) of (a) Kalman filters and (b) finite impulse response (FIR) filters.

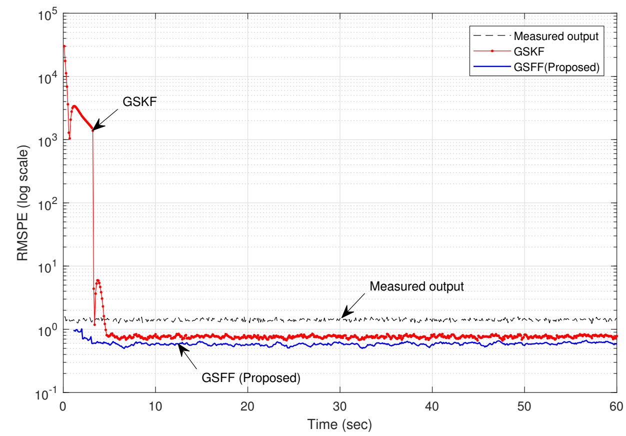


Fig. 2. Root mean square position error (RMSPE) for inconsistent target mobility comparing the proposed algorithm with (a) Kalman filters and (b) least square finite impulse response filters (LSFFs).

Table 1. Root time averaged mean square error for consistent target mobility.

KF	857.0 ($q_k = 0.1$)	646.3 ($q_k = 1$)	190.3 ($q_k = 10$)
LSFF	0.9676 ($N = 10$)	0.6775 ($N = 20$)	0.5898 ($N = 30$)
GSKF	471.4		
GSFF	0.5971		
Measurement	1.188		

Table 2. Total operation time (TOT).

Filter	GSKF	GSFF
TOT (s)	0.1134	0.0820

FIR filter horizon size is unknown unless elaborate and extensive pre-simulations are performed. On the other hand, GSFF exhibits similar RMSPEs to the best LSFF ($N = 30$). Thus, GSFF provides performance similar to the best horizon size without requiring pre-simulation. We see that GSFF RMSPE converges to that of the best LSFF. This is because the weight of the best LSFF converges to 1.

Fig. 2 compares the GSKF and the proposed GSFF. As clearly seen in this figure, GSFF RMSPE is smaller than that of GSKF. Thus, GSFF can offer more accurate tracking than GSKF. Especially, GSFF is much better than GSKF in the early stage.

Table 1 shows RTAMS of all filters used in the simulations. KF RTAMSs are significantly large because of large RMSPEs for early stage. GSKF RTAMS is larger than that of best the KF with $q_k = 10$. This is because the GSKF produced larger RMSPE than that of the best KF until the weight for this KF becomes 1. LSFF RTAMSs are significantly smaller than those of KFs. GSFF RTAMS is close to the best LSFF with $N = 30$. In terms of RTAMS, GSFF is superior to GSKF.

Table 2 compares GSFF and GSKF in terms of total operation time (TOT). GSFF TOT is smaller than that for GSKF. GSKF and GSFF operate three KFs and three LSFFs, respectively. While KF gain is time-varying, LSFF gain is time-invariant. Hence, GSFF can reduce computation time by computing the gain beforehand. As a result, the GSFF exhibited smaller TOT compared with the GSKF.

5. CONCLUSIONS

This paper proposed a Gaussian sum FIR filter (GSFF) for 2D target tracking. Simulations confirmed that GSFF exhibited comparable performance to ideal LSFF without requiring any information on horizon size. GSFF exhibited better performance compared with KF and GSKF without requiring any information on process noise covariance. Thus, GSFF provides an alternative solution for 2D target tracking problems using the CV model. However, GSFF is only applicable for linear systems, which limits its application. Therefore, future work will investigate extending GSFF general nonlinear systems.

REFERENCES

[1] W.-S. R. S.-K. H and J. B. Park, "Tdoa/fdoa based target tracking with imperfect position and velocity data of distributed moving sensors," *International Journal of Control, Automation, and Systems*, vol. 15, no. 3, pp. 1155-1166, Jun. 2017.

[2] X. L. P. Wu and Y. Bo, "Iterated square root unscented Kalman filter for maneuvering target tracking using tdoa measurements," *International Journal of Control, Automation, and Systems*, vol. 11, no. 4, pp. 761-767, Aug. 2013.

[3] B. Li and J. Zhao, "Auxiliary particle bernoulli filter for target tracking," *International Journal of Control, Automation, and Systems*, vol. 15, no. 3, pp. 1249-1258, Jun. 2017.

[4] B. Li, "Multiple-model rao-blackwellized particle probability hypothesis density filter for multitarget tracking," *International Journal of Control, Automation, and Systems*, vol. 13, no. 2, pp. 426-433, Apr. 2015.

[5] P. . Gutman and M. Velger, "Tracking targets using adaptive Kalman filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 5, pp. 691-699, Sep. 1990.

[6] T. R. Kronhamn, "Bearings-only target motion analysis based on a multihypothesis kalman filter and adaptive ownship motion control," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 145, no. 4, pp. 247-252, Aug. 1998.

[7] W. F. Leven and A. D. Lanterman, "Unscented kalman filters for multiple target tracking with symmetric measurement equations," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 370-375, Feb. 2009.

[8] M. R. Morelande and S. Challa, "Manoeuvring target tracking in clutter using particle filters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 1, pp. 252-270, Jan. 2005.

[9] A. N. Bishop, P. N. Pathirana, and A. V. Savkin, "Radar target tracking via robust linear filtering," *IEEE Signal Processing Letters*, vol. 14, no. 12, pp. 1028-1031, Dec. 2007.

[10] A. N. Bishop, A. V. Savkin, and P. N. Pathirana, "Vision-based target tracking and surveillance with robust set-valued state estimation," *IEEE Signal Processing Letters*, vol. 17, no. 3, pp. 289-292, March 2010.

[11] Z. Zhao, T. X. R. Li, and V. P. Jilkov, "Best linear unbiased filtering with nonlinear measurements for target tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 4, pp. 1324-1336, Oct. 2004.

[12] A. Jazwinski, "Limited memory optimal filtering," *IEEE Transactions on Automatic Control*, vol. 13, no. 5, pp. 558-563, October 1968.

[13] W. H. Kwon, P. S. Kim, and P. Park, "A receding horizon kalman FIR filter for discrete time-invariant systems," *IEEE Trans. Autom. Control*, vol. 44, no. 9, pp. 1787-1791, Sep. 1999.

[14] W. H. Kwon, K. S. Lee, and O. K. Kwon, "Optimal FIR filters for time-varying state-space models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 26, no. 6, pp. 1011-1021, Nov. 1990.

[15] W. H. Kwon and S. Han, *Receding Horizon Control: Model Predictive Control for State Models*, Springer-Verlag, London, 2005.

[16] Y. S. Shmaliy, "Linear optimal FIR estimation of discrete time-invariant state-space models," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3086-3096, Jun. 2010.

[17] I. H. Choi, J. M. Pak, C. K. Ahn, Y. H. Mo, M. T. Lim, and M. K. Song, "New preceding vehicle tracking algorithm based on optimal unbiased finite memory filter," *Measurement*, vol. 73, pp. 262-274, Sep. 2015.

[18] C. J. Lee, J. M. Pak, C. K. Ahn, K. M. Min, P. Shi, and M. T. Lim, "Multi-target FIR tracking algorithm for markov jump linear systems based on true-target decision-making," *Neurocomputing*, vol. 168, pp. 298-307, Jan. 2015.

[19] J. M. Pak, C. K. Ahn, M. T. Lim, and M. K. Song, "Horizon group shift FIR filter: alternative nonlinear filter using finite recent measurements," *Measurement*, vol. 57, pp. 33-45, Nov. 2014.

[20] J. M. Pak, C. K. Ahn, Y. S. Shmaliy, and M. T. Lim, "Improving reliability of particle filter-based localization in wireless sensor networks via hybrid particle/FIR filtering," *IEEE Trans. on Industrial Informatics*, vol. 11, no. 5, pp. 1089-1098, Oct. 2015.

[21] J. M. Pak, C. K. Ahn, Y. S. Shmaliy, P. Shi, and M. T. Lim, "Switching extensible FIR filter bank for adaptive horizon state estimation with application," *IEEE Trans. on Control Systems Technology*, vol. 24, no. 3, pp. 1052-1058, May 2016.

[22] J. M. Pak, C. K. Ahn, C. J. Lee, P. Shi, M. T. Lim, and M. K. Song, "Fuzzy horizon group shift FIR filtering for nonlinear systems with takagi-sugeno model," *Neurocomputing*, vol. 174, no. Part B, pp. 1013-1020, Jan. 2016.

[23] J. M. Pak, C. K. Ahn, P. Shi, and M. T. Lim, "Self-recovering extended kalman filtering algorithm based on model-based diagnosis and resetting using an FIR filter," *Neurocomputing*, vol. 173, no. Part 3, pp. 645-658, Jan. 2016.

[24] J. M. Pak, C. K. Ahn, Y. H. Mo, M. T. Lim, and M. K. Song, "Maximum likelihood FIR filter for visual object tracking," *Neurocomputing*, vol. 216, pp. 543-553, Dec. 2016.

[25] J. M. Pak, P. S. Kim, S. H. You, S. S. Lee, and M. K. Song, "Extended least square unbiased FIR filter for target tracking using the constant velocity motion model," *International Journal of Control, Automation and Systems*, vol. 15, no. 2, pp. 947-951, Apr. 2017.

[26] J. M. Pak, C. K. Ahn, P. Shi, Y. S. Shmaliy, and M. T. Lim, "Distributed hybrid particle/FIR filtering for mitigating NLOS effects in TOA-based localization using wireless sensor networks," *IEEE Trans. on Industrial Electronics*, vol. 64, no. 6, pp. 5182-5191, Jun. 2017.

[27] J. M. Pak, C. K. Ahn, Y. S. Shmaliy, P. Shi, and M. T. Lim, "Accurate and reliable human localization using composite particle/FIR filtering," *IEEE Trans. on Human-Machine Systems*, vol. 47, no. 3, pp. 332-342, Jun 2017.

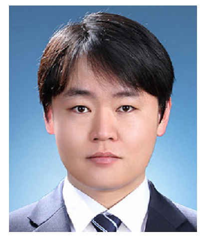
[28] B. F. L. Scala and R. R. Bitmead, "Design of an extended kalman filter frequency tracker," *IEEE Transactions on Signal Processing*, vol. 44, no. 3, pp. 739-742, March 1996.

[29] F. Ramirez-Echeverria, A. Sarr, and Y. S. Shmaliy, "Optimal memory of discrete-time FIR filters in state-space," *IEEE Trans. on Signal Processing*, vol. 62, no. 3, pp. 557-561, Feb. 2014.

[30] H. W. Sorenson and D. L. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica*, vol. 7, no. 4, pp. 465-497, Jul. 1971.

[31] D. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Trans. on Automatic Control*, vol. 17, no. 4, pp. 439-448, Aug. 1972.

[32] S. A. B. Ristic and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Arctech House, Norwood, MA, 2004.



Jung Min Pak received his B.S., M.S., and Ph.D. degrees in electrical engineering from Korea University, Seoul, Korea, in 2006, 2008, and 2015, respectively. He was a Research Professor of the Brain Korea 21 Plus Humanware IT Program at Korea University from 2015 to 2016. He is currently a Professor in the Department of Electrical Engineering, Wonkwang University, Iksan, Korea. His research interests include state estimation, localization, and visual object tracking.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.