



LUND UNIVERSITY

Event-Based State Estimation Using the Auxiliary Particle Filter

Ruuskanen, Johan; Cervin, Anton

Published in:
2019 European Control Conference (ECC)

DOI:
[10.23919/ECC.2019.8796091](https://doi.org/10.23919/ECC.2019.8796091)

2019

Document Version:
Peer reviewed version (aka post-print)

[Link to publication](#)

Citation for published version (APA):
Ruuskanen, J., & Cervin, A. (2019). Event-Based State Estimation Using the Auxiliary Particle Filter. In *2019 European Control Conference (ECC)* IEEE Press. <https://doi.org/10.23919/ECC.2019.8796091>

Total number of authors:
2

General rights

Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Event-Based State Estimation Using the Auxiliary Particle Filter

Johan Ruuskanen¹ and Anton Cervin¹

Abstract—Event-based sampling provides a way of lowering the resource utilization in sensing and communication applications. By sending a sample only when some triggering condition is fulfilled, we can ensure that the transmitted samples actually carry innovation. However, in an event-based system, the state estimation problem becomes complicated, as the information of not receiving a measurement must be taken into consideration.

Recent research has examined the feasibility of using particle filters for solving the event-based state estimation problem. To the best of our knowledge, only the simple bootstrap particle filter has so far been considered in this setting. We argue that, as this filter does not fully utilize the current measurement, it is not well suited for state estimation in event-based systems.

We propose an extension to the auxiliary particle filter for systems with event-based measurements, in which certain existing techniques for finding an approximation of the fully adapted filter can easily be utilized. In a simulation study, we demonstrate that at new measurement events, the benefits of using the auxiliary particle filter increases when fewer measurements are being sent.

I. INTRODUCTION

Event-based sampling, communication, and control has gained increased attention in recent years [1]. One motivation is to reduce the communication bandwidth and to prolong the lifetime of battery-powered sensors. In this kind of setting, an event triggering mechanism such as send-on-delta (SOD) [2] is designed and implemented in the sensor node. In other cases, event-based sampling is inherent in the measurement device. Examples of this include wheel encoders, digital positioning sensors, and A/D converters with a resolution down to 1 bit.

With event-based sampling, the state estimation problem becomes non-trivial even for the case of linear process dynamics with Gaussian additive process noise (LG). The key difficulty is that even the *absence* of measurements contains some information, which in general can not be described using a Gaussian kernel. One notable exception is the usage of *stochastic* triggers proposed in [3], [4], for which the closed-form, optimal filters for LG systems can be found. For *deterministic* triggering methods, such as the common SOD method, this is no longer the case. Here, non-LG filtering techniques must be employed. Popular nonlinear extensions to the Kalman filter [5] such as the extended Kalman filter and the unscented Kalman filter [6] are not suitable since they assume a Gaussian noise model. A better candidate for event-based state estimation is the particle filter, which makes no particular assumptions on the noise [7].

State estimation under event-based sampling carries some similarity to estimation with missing values [8] or quantized measurements [9]. In both cases the particle filter has

been considered [10], [11], [12]. The key difference to the missing value concept is the fact that absent measurements carry no innovation, while the nature of event-triggering gives information about the measurement when no triggering has occurred. Compared to quantized measurements, the quantization level gives an area of possible values for the true underlying measurement, which the event-triggering also provides. However, in the quantized case this area is fixed, while in the event-based case the area is free to change at or between new events. Further, a shift to a new measurement in a quantized system will also be quantized, while in event-based settings new measurements are not distorted.

A handful of earlier works on event-based state estimation using particle filters exist. In [13] the use of event-based sampling in nonlinear filtering is described, and as an example a particle filter is demonstrated. In [14], Sid and Chitraganti derive an event-based extension to the particle filter where the likelihood is computed using numerical integration. In [15], Davar and Mohammadi derive an event-based particle filter tailored for an LG system with SOD triggering. In the work by Liu et al. [16], the particle filter is used as a central event-based estimator, where the sensor triggering conditions are governed by local, less cumbersome estimators. These works all utilize the bootstrap particle filter (BPF) [17]. The BPF is an easy-to-implement version of the standard particle filter, in which the particles are simply propagated according to the process model. This has a major drawback that becomes particularly amplified in conjunction with event-based sampling—it does not utilize the latest measurement in the propagation/resampling step. This can lead to particle depletion at the critical instant of processing a new measurement.

In this work we propose to use the auxiliary particle filter (APF) [18] to improve the estimator performance at new measurement events. Assuming a known triggering condition, we show that, by using an approximation of the likelihood based on a Gaussian mixture model, an adequate proposal distribution and approximate predictive likelihood can be found. The likelihood approximation was previously proposed by Sijs and Lazar [19], who used it for event-based state estimation in conjunction with a Gaussian state approximation in each step to form their event-based state estimator (EBSE). By contrast, using a particle filter, we are able to track the non-Gaussian state distribution at all times.

The rest of the paper is organized as follows. In Sec. II, event-based state estimation and its implications are presented. In Sec. III, the auxiliary particle filter is reviewed. In Sec. IV, a motivation is first provided on why the APF extension is more important in event-based settings, and then

¹ Department of Automatic Control LTH, Lund University, Sweden. Email: {johan.ruuskanen, anton.cervin}@control.lth.se

a framework on how to extend current methods for designing of APF filters to event-based system is presented. In Sec. V, the benefits of utilizing the APF over the BPF in an event-based setting is demonstrated on an example system. Finally, in Sec. VI, the results and findings are discussed.

II. EVENT-BASED STATE ESTIMATION

In this paper we consider the state estimation problem for the discrete-time state-space model

$$x_k = f(x_{k-1}, v_k), \quad y_k = h(x_k, w_k), \quad (1)$$

where $k \in \mathbb{N}$ is the time index, $x_k \in \mathbb{R}^n$ is the unknown state vector, and $y_k \in \mathbb{R}^m$ is the known measurement vector, while $v_k \in \mathbb{R}^v$ and $w_k \in \mathbb{R}^w$ are mutually uncorrelated noise processes following some known distributions. The transition function is defined as $f : \mathbb{R}^n \times \mathbb{R}^v \rightarrow \mathbb{R}^n$ and the measurement function as $h : \mathbb{R}^n \times \mathbb{R}^w \rightarrow \mathbb{R}^m$. Let $p(A | B)$ denote the probability density function (PDF) of A conditioned on B . Further assume that the PDFs of the state and measurement vectors can be found as $x_k \sim p(x_k | x_{k-1})$ and $y_k \sim p(y_k | x_k)$, denoting the transition and likelihood densities, respectively. Finally, denote an event at time k as $\gamma_k = 1$ and the absence of an event as $\gamma_k = 0$.

In the state estimation problem, we are tasked with finding the sequence of hidden state vectors $x_{1:T}$ given the observed measurements $y_{1:T}$. From a Bayesian point of view, this can be formulated as $p(x_{1:T} | y_{1:T})$, i.e., finding the distribution of the hidden state vectors conditioned on the observed measurements. In many settings, it is most relevant to consider the causal *filtering* problem, in which the measurements $y_{k+1:T}$ are not available at time k . Assuming causality, the state-space density can be rewritten as the following recursion using Bayes' law:

$$p(x_{1:k} | y_{1:k}) = p(x_{1:k-1} | y_{1:k-1}) \frac{p(y_k | x_k) p(x_k | x_{k-1})}{p(y_k | y_{1:k-1})}. \quad (2)$$

In an event-based setting, new measurements are not guaranteed to be available at each time instance k . From the viewpoint of the observer, the set of all possible values can be formalized as

$$\mathcal{Y}_k = \begin{cases} \{y_k\}, & \gamma_k = 1, \\ \mathcal{H}_k, & \gamma_k = 0. \end{cases} \quad (3)$$

If a new measurement is obtained, then the set of possible values will simply be the sent value y_k . Otherwise, the possible values will belong to some set \mathcal{H}_k that depends on the event generation. The likelihood over \mathcal{Y}_k can be expressed as

$$p(\mathcal{Y}_k | x_k) = \int p(\mathcal{Y}_k | \bar{y}_k) p(\bar{y}_k | x_k) d\bar{y}_k. \quad (4)$$

For a deterministic triggering rule, a new event is guaranteed to be generated once a measurement is outside some area Ξ . If no event is generated, then $P(y_k \in \Xi | \gamma_k = 0) = 1$. Following the reasoning from [19], the likelihood over \mathcal{Y}_k can thus be expressed as

$$p(\mathcal{Y} | y_k, \gamma_k) = \begin{cases} \delta_{y_k}(x) & \gamma_k = 1, \\ \mathcal{U}(\bar{y}_k \in \Xi) & \gamma_k = 0, \end{cases} \quad (5)$$

where $\delta_{y_k}(x)$ is the Dirac delta evaluated at y_k , and $\mathcal{U}(\bar{y} \in \Xi)$ the uniform distribution over the area Ξ .

III. THE AUXILIARY PARTICLE FILTER

Particle filters are a class of sequential Monte Carlo methods commonly used in state estimation, which aim at approximating the posterior $p(x_{1:k} | y_{1:k})$ with a set of discrete points referred to as *particles*. In this paper we only give a quick walk-through of the most important concepts for the APF. For a more thorough and rigorous derivation spanning the basic sequential Monte Carlo methods up until the auxiliary particle filter, we refer to the compendium by Doucet and Johansen [7, Ch. 3-4].

Define a particle i as the Dirac delta with its entire support located at X^i , i.e., $\delta_{X^i}(x)$. Using a set of N particles, an approximation of an arbitrary density $p(x)$ can be found as $p(x) \approx \sum_{i=1}^N \delta_{X^i}(x)$, where $X^i \sim p(x)$. In general, generating samples from $p(x)$ can be hard, if not impossible. Instead, particles can be drawn from a *proposal density* $X^i \sim q(x)$ and weighted as $W^i \propto p(X^i) / q(X^i)$ to generate the approximation $p(x) \approx \sum_{i=1}^N W^i \delta_{X^i}$. For a state estimation problem, this can be formulated as drawing i sequences $\{X_{1:T}^i\}_{i=1:N}$ with the corresponding weights $\{W_{1:T}^i\}_{i=1:N}$ to form

$$p(x_{1:k} | y_{1:k}) \approx \sum_{i=1}^N W_k^i \delta_{X_{1:k}^i}(x_{1:k}), \quad \forall k \in 1:T. \quad (6)$$

To draw and weight the particles, the recursion (2) can be utilized at each time step k as

$$\begin{aligned} X_k^i &\sim q(x_k | X_{k-1}^i, y_k), \\ W_k^i &\propto W_{k-1}^i \frac{p(y_k | X_k^i) p(X_k^i | X_{k-1}^i)}{q(X_k^i | X_{k-1}^i, y_k)}. \end{aligned} \quad (7)$$

Starting from some initial distribution $X_0^i \sim p(x_0)$, the particles and weights obtained from this recursive formula can be shown to generate an approximate state-space density that converges as $N \rightarrow \infty$. To make the algorithm feasible for a finite number of particles, however, a couple of extensions can be made.

First, finding a good proposal density is of importance. In order to minimize the variance of the weights, it should optimally be chosen as $q(x_k | x_{k-1}, y_k) \propto p(y_k | x_k) p(x_k | x_{k-1})$. Finding this optimal proposal is often not possible; instead it must be approximated in some manner. Another simple solution is setting the proposal to the transition density, $q(x_k | x_{k-1}, y_k) = p(x_k | x_{k-1})$, which results in the bootstrap particle filter.

Secondly, resampling of the particles is commonly introduced. In each step, the propagation will disperse the particles evermore over the state space, resulting in a majority of the total weight ending up in a small subset of the particles. By resampling the particles this weight degeneracy can be remedied. Let a^i be indexes drawn from the categorical distribution $\mathcal{C}(\{W_k^i\}_{i=1:N})$. The resampled particles with their corresponding weights then become

$$\{X_{1:k}^i, W_{1:k}^i\}_{i=1:N} \approx \{X_{1:k}^{a^i}, 1/N\}_{i=1:N}. \quad (8)$$

Performing resampling in each step if often not necessary. Instead it can be triggered when the *effective sample size* $N_e \approx 1 / \sum_i^N (W_k^i)^2$ falls below some limit. The recursive algorithm (7) together with the resampling step constitutes the general particle filter algorithm.

As with the propagation density, the resampling could also be conditioned on the current measurement y_k . This is the idea behind the auxiliary particle filter, where the *predictive likelihood* $p(y_k | x_{k-1})$ is used evaluate the likelihood of observing y_k for a particle at time $k-1$ and used in the resampling step, i.e., $V_{k-1}^i \propto W_{k-1}^i p(y_k | X_{k-1}^i)$. The indexes a^i are then drawn from $\mathcal{C}(\{V_{k-1}^i\}_{i=1:N})$ instead, and the resampled particles propagated and weighted as

$$\begin{aligned} X_k^i &\sim q(x_k | X_{k-1}^{a^i}, y_k), \\ W_k^i &\propto W_{k-1}^{a^i} \frac{p(y_k | X_k^i) p(X_k^i | X_{k-1}^{a^i})}{p(y_k | X_{k-1}^{a^i}) q(X_k^i | X_{k-1}^{a^i}, y_k)}. \end{aligned} \quad (9)$$

Finding the predictive likelihood is in general difficult. Instead, an approximation can be used. However, if the true predictive likelihood is obtainable and used in conjunction with the optimal proposal, the APF is referred to as *fully adapted* [20].

IV. APPROXIMATING THE FULLY-ADAPTED FILTER FOR EVENT-BASED SYSTEMS

Earlier work has extended the bootstrap particle filter for event-based systems with deterministic kernels. However, the lack of inclusion of the current measurement makes the BPF worse than the ordinary particle filter. This should also be true for event-based systems. Moreover, we argue that inclusion of the current measurement is even more important in event-based settings. In these systems, one can typically have long periods of no measurements between rare instances of new events. The approximated state estimate density right before an event at time k would be broad, depending on the broad spectrum of the particles X_{k-1}^i that could generate a high likelihood on \mathcal{Y}_{k-1} . When a new event happens, it will due to the nature of event-triggering most probably be at a location where $p(\mathcal{Y}_{k-1} | x_{k-1})$ is low. Thus, by not utilizing the information from the event, X_{k-1}^i propagated in the BPF manner will most likely end up with a low $p(y_k | X_k^i)$, resulting in a bad approximation of the target distribution.

The same could be argued for the inclusion of the APF extension. As a new event indicates that the measurement y_k is outside the kernel interval defining the broad likelihood $p(\mathcal{Y}_{k-1} | x_{k-1})$, only a fraction of the particles with a high likelihood on \mathcal{Y}_{k-1} would actually have a large predictive likelihood on y_k . In other words, the states of the true underlying y_{k-1} will most likely be located at the “edges” of the likelihood density. Particles not located in this vicinity would have practically zero likelihood of generating the observed event y_k , and thus give a poor performance on the approximation of the state estimation density, even though the chosen proposal is sound. The larger the kernel interval is, the more particles should have a low predictive likelihood. By using the APF approach, this effect should be mitigated

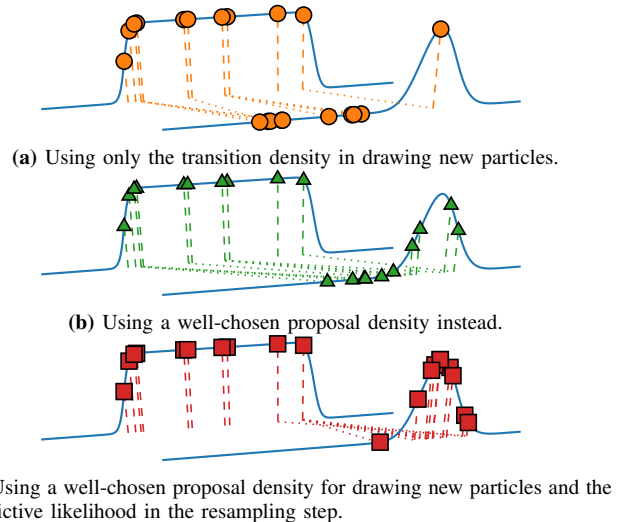


Fig. 1: Illustration of the effect of using a well-chosen proposal density and a resampling strategy conditioned on the predictive likelihood in a one dimensional event-based setting. The blue lines represent a snapshot of the underlying posterior density $p(x_{1:k} | \mathcal{Y}_{1:k})$, the line with a broad bump represents the likelihood with no triggered event $\gamma_k = 0$ while the line with the narrow bump represents the likelihood with a triggered event $\gamma_k = 1$. The markers represent the particles.

by choosing particles in the resampling step that would have a high $p(y_k | x_{k-1})$.

Thus far, a motivation as to why it could be of importance to use an APF over a BPF in event-based systems has been provided. See Fig. 1 for an illustration. We now show that constructing an APF for an event-based system is entirely possible, with a proposal and approximate predictive likelihood approximating the fully-adapted filter. First, we state two lemmas that will be needed.

Lemma 1: Given the joint distribution of two correlated Gaussian distributed variables, $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$,

$$p(x, y) = \mathcal{N} \left(\begin{pmatrix} x \\ y \end{pmatrix} \middle| \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right), \quad (10)$$

the conditional distribution with respect to $y = \alpha$ becomes

$$\begin{aligned} p(x | y = \alpha) &= \mathcal{N}(x | \mu, \Sigma), \\ \mu &= \mu_1 + \Sigma_{12} (\Sigma_{22})^{-1} (y - \mu_2), \\ \Sigma &= \Sigma_{11} - \Sigma_{12} (\Sigma_{22})^{-1} \Sigma_{12}^T. \end{aligned} \quad (11)$$

Proof: See [21, p. 116 - 117]. \blacksquare

Lemma 2: The product of two independent Gaussian densities becomes a new Gaussian density weighted with a Gaussian function:

$$\begin{aligned} \mathcal{N}(x | \mu_1, \Sigma_1) \mathcal{N}(x | \mu_2, \Sigma_2) &= \mathcal{N}(\mu_1 | \mu_2, \Sigma_1 + \Sigma_2) \mathcal{N}(x | \mu_3, \Sigma_3), \\ \mu_3 &= (\Sigma_1^{-1} + \Sigma_2^{-1}) (\Sigma_1^{-1} \mu_1 + \Sigma_2^{-1} \mu_2), \\ \Sigma_3 &= (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}. \end{aligned} \quad (12)$$

Proof: See [22]. With a few algebraic manipulations the expression can be written as two normal densities. \blacksquare

Consider the likelihood for the event-based system with deterministic sampling. When an event is triggered ($\gamma_k = 1$), the likelihood reduces to the normal state-space model likelihood density by (5) applied to (4). In the case of no

event ($\gamma_k = 0$), the likelihood reduces to the integral

$$\begin{aligned} p(\mathcal{Y}_k | x_k) &= \int p(\bar{y}_k | x_k) \mathcal{U}(\bar{y}_k \in \Xi) d\bar{y}_k \\ &= \frac{1}{Z} \int_{\Xi} p(\bar{y}_k | x_k) d\bar{y}_k, \end{aligned} \quad (13)$$

where Z is the normalizing constant. For a one-dimensional $h(x_k, w_k)$, where the likelihood density has a known primitive and the bounds of Ξ are known, the integral is trivial to evaluate. In [15], this is used for the case when the likelihood is Gaussian, but it is applicable to a much broader class of densities. If the density instead has an unknown primitive and/or is multivariate, the ideas from *constrained Bayesian state estimation* can instead be utilized, as Liu *et al.* [16] have shown. Thus the weights of a particle filter (7) or APF (9) in an event-based setting with deterministic triggering can always be found for some $q(x_k | x_{k-1}, \mathcal{Y}_k)$ and $\hat{p}(\mathcal{Y}_k | x_{k-1})$. To find these densities, the two cases $\gamma_k = 1$ and $\gamma_k = 0$ will be considered separately.

$\gamma_k = 1$: Here, $p(\mathcal{Y}_k | x_k)$ reduces to the original likelihood density $p(y_k | x_k)$ for the state-space model (1). The problem boils down to finding a proposal and approximate predictive likelihood for an ordinary system, for which many results exist [20]. One solution is to first create a joint Gaussian density $p(x_k, y_k | x_{k-1})$ from the state-space densities. Since y_k given x_k is independent from x_{k-1} , the joint density is proportional to the optimal proposal and can be found using Bayes as

$$\begin{aligned} q(x_k | x_{k-1}, y_k) &\propto p(y_k | x_k) p(x_k | x_{k-1}) \\ &= p(x_k, y_k | x_{k-1}). \end{aligned} \quad (14)$$

Using Bayes again, we can find the following expression:

$$\begin{aligned} q(x_k | x_{k-1}, y_k) &\propto p(x_k, y_k | x_{k-1}) \\ &= p(x_k | x_{k-1}, y_k) p(y_k | x_{k-1}) \\ &\propto p(x_k | x_{k-1}, y_k). \end{aligned} \quad (15)$$

Since y_k and x_{k-1} are given, the predictive likelihood $p(y_k | x_{k-1})$ simply becomes a scaling factor that can be removed by proportionality. If the system is LG, then the joint Gaussian (14) can be found analytically. If the system is additive Gaussian but not linear, the resulting joint Gaussian can be approximated by, e.g., linearization [23] or exact moment matching [24]. Otherwise, Gaussian approximations of the state-space densities can first be found as

$$\begin{aligned} p(y_k | x_k) &\approx \mathcal{N}[y_k | \mathbb{E}(p(y_k | x_k)), \\ &\quad \alpha_1 \mathbb{V}(p(y_k | x_k))], \end{aligned} \quad (16)$$

$$\begin{aligned} p(x_k | x_{k-1}) &\approx \mathcal{N}[x_k | \mathbb{E}(p(x_k | x_{k-1})), \\ &\quad \alpha_2 \mathbb{V}(p(x_k | x_{k-1}))], \end{aligned} \quad (17)$$

where α_1, α_2 are scaling factors to keep the support of the approximations wider than the original densities. The joint Gaussian density can then be found either directly if linear, or through approximations. From the joint Gaussian, the proposal can be calculated using Lemma 1, and the approximate predictive likelihood can be found as the marginalization of the joint probability over x_k :

$$\begin{aligned} q(x_k | x_{k-1}, y_k) &= \mathcal{N}(x_k | \mu_3, \Sigma_3), \\ \hat{p}(y_k | x_{k-1}) &= \mathcal{N}(y_k | \mu_2, \Sigma_{22}). \end{aligned} \quad (18)$$

$\gamma_k = 0$: Following Sijs and Lazar [19], the uniform distribution in the likelihood (5) can be approximated by a mixture of M Gaussians with the means located at some discretization of the kernel interval, $\hat{y}_k^j \in \Xi \forall j \in [1, M]$, and with some covariance matrices, V_k^j . Using this approximation, the likelihood can be written as

$$\begin{aligned} p(\mathcal{Y}_k | x_k) &= \int \mathcal{U}(\bar{y}_k | \bar{y}_k \in \Xi) p(\bar{y}_k | x_k) d\bar{y}_k \\ &\approx \frac{1}{M} \sum_j \int \mathcal{N}(\bar{y}_k | \hat{y}_k^j, V_k^j) p(\bar{y}_k | x_k) d\bar{y}_k. \end{aligned} \quad (19)$$

For this integral to be tractable, the original likelihood $p(y_k | x_k)$ cannot be allowed to be arbitrarily defined. Instead, for an arbitrary system we could approximate the original likelihood with a Gaussian to allow the integral to be evaluated. Use the same notation as in (16):

$$\hat{p}(y_k | x_k) = \mathcal{N}(y_k | \mathbb{E}(p(y_k | x_k)), \alpha_1 \mathbb{V}(p(y_k | x_k))). \quad (20)$$

Using this approximation, the likelihood (19) can be rewritten using Lemma 2 as

$$\begin{aligned} \hat{p}(\mathcal{Y}_k | x_k) &= \frac{1}{M} \sum_j \int \mathcal{N}(\bar{y}_k | \hat{y}_k^j, V_k^j) \hat{p}(\bar{y}_k | x_k) d\bar{y}_k \\ &= \frac{1}{M} \sum_j \mathcal{N}(\hat{y}_k^j | \mathbb{E}(p(y_k | x_k)), \\ &\quad V_k^j + \alpha_1 \mathbb{V}(p(y_k | x_k))) \\ &= \frac{1}{M} \sum_j \hat{p}(\hat{y}_k^j | x_k). \end{aligned} \quad (21)$$

Since the Gaussian weights are independent of \bar{y}_k , the integral over the resulting density becomes $\int \mathcal{N}(\bar{y}_k | \mu_3, \Sigma_3) d\bar{y}_k = 1$. The following optimal proposal can then be stated:

$$\begin{aligned} q(x_k | x_{k-1}, \mathcal{Y}_k) &\propto p(\mathcal{Y}_k | x_k) p(x_k | x_{k-1}) \\ &\approx \frac{1}{M} \sum_{j=1}^M \hat{p}(\hat{y}_k^j | x_k) p(x_k | x_{k-1}). \end{aligned} \quad (22)$$

Using the same logic as for $\gamma_k = 1$, the joint Gaussian density can be found either directly or by first approximating $p(x_k | x_{k-1})$ by a Gaussian as in (17):

$$\begin{aligned} q(x_k | x_{k-1}, \mathcal{Y}_k) &\propto \frac{1}{M} \sum_{j=1}^M p(\hat{y}_k^j, x_k | x_{k-1}) \\ &\propto \sum_{j=1}^M p(x_k | x_{k-1}, \hat{y}_k^j) p(\hat{y}_k^j | x_{k-1}). \end{aligned} \quad (23)$$

The proposal thus becomes a mixture of Gaussians $p(x_k | x_{k-1}, \hat{y}_k^j)$ found using Lemma 1, where each Gaussian is weighted using the predictive likelihood $p(\hat{y}_k^j | x_{k-1})$, found by marginalizing the joint distribution over x_k . Similarly, the predictive likelihood of \mathcal{Y}_k can be found as

$$\begin{aligned} p(\mathcal{Y}_k | x_{k-1}) &\approx \int \frac{1}{M} \sum_{j=1}^M \hat{p}(\hat{y}_k^j | x_k) p(x_k | x_{k-1}) dx_k \\ &\propto \sum_{j=1}^M p(\hat{y}_k^j | x_{k-1}). \end{aligned} \quad (24)$$

V. SIMULATION STUDY

In order to evaluate the performance of the suggested proposal and approximated predictive likelihood, a simulation study is performed on an LG system sampled with a deterministic event kernel. Both the BPF and the APF are compared to the EBSE filter introduced by Sijs and Lazar [19]. The system considered is the same simple, one-dimensional target tracking problem as Davar and Mohammadi used for demonstrating their event-based BPF [15]:

$$\begin{aligned} x_k &= \begin{bmatrix} 0.8 & 1 \\ 0 & 0.95 \end{bmatrix} x_{k-1} + v_k & v_k &\sim \mathcal{N}\left(0, \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}\right), \\ y_k &= [0.7 \quad 0.6] x_k + w_k & w_k &\sim \mathcal{N}(0, 0.01). \end{aligned} \quad (25)$$

As event-kernel the simple SOD method for a one dimensional measurement is considered:

$$\{z_k, \gamma_k\} = \begin{cases} \{y_k, 1\} & \text{if } |y_k - z_{k-1}| > \delta, \\ \{z_{k-1}, 0\} & \text{otherwise.} \end{cases} \quad (26)$$

Each simulation was run for 1000 time steps over different settings of the two parameters N —*number of particles* and δ —*event kernel width*. For each parameter pair, the simulation was run 1000 times, with a new system simulated for each run, to create an accurate Monte-Carlo estimation of performance. For the particle filters, adaptive resampling with systematic resampling was used in the resampling step where the limit for the effective sample size was set to $N_{\text{lim}} = N/2$. In the special case of a LG system, the joint density (14) can for $\gamma_k = 1$ be found exactly as

$$\begin{aligned} p(x_k, y_k | x_{k-1}) &= \\ \mathcal{N}\left(\begin{pmatrix} x_k \\ y_k \end{pmatrix} \middle| \begin{bmatrix} Ax_{k-1} \\ CAx_{k-1} \end{bmatrix}, \begin{bmatrix} Q & QC^T \\ CQ & CQC^T + R \end{bmatrix}\right). \end{aligned} \quad (27)$$

By using Lemma 2 and marginalization over x_k , the proposal density and predictive likelihood can be found as

$$\begin{aligned} p(y_k | x_{k-1}) &= \mathcal{N}(y_k | CAx_{k-1}, S), \\ q(x_k | x_{k-1}, y_k) &= \mathcal{N}(x_k | \mu_q, \Sigma_q), \\ \mu_q^j &= Ax_{k-1} + QC^T S^{-1} (y_k - CAx_{k-1}), \\ \Sigma_q &= Q - QC^T S^{-1} CQ, \\ S &= CQC^T + R. \end{aligned} \quad (28)$$

For $\gamma_k = 0$, the proposal and approximate likelihood can be found in similar manner as

$$\hat{p}(\mathcal{Y}_k | x_{k-1}) \propto \sum_{j=1}^M \hat{p}(\hat{y}_k^j | x_{k-1}), \quad (29)$$

$$q(x_k | x_{k-1}, \mathcal{Y}_k) \propto \sum_{j=1}^M \hat{p}(\hat{y}_k^j | x_{k-1}) \mathcal{N}(x_k | \mu_q^j, \Sigma_q),$$

where

$$\begin{aligned} \hat{p}(\hat{y}_k^j | x_{k-1}) &= \mathcal{N}(\bar{y}_k^j | CAx_{k-1}, S) \\ \mu_q^j &= Ax_{k-1} + QC^T S^{-1} (\hat{y}_k^j - CAx_{k-1}) \\ \Sigma_q &= Q - QC^T S^{-1} CQ, \\ S &= CQC^T + R + V_k^j. \end{aligned} \quad (30)$$

The reader can recognize the formulas for μ_q and Σ_q from the Kalman filter.

At each event, the uniform distribution will need to be approximated by a Gaussian mixture with M densities, located at \hat{y}_k^j with covariance V_k^j . For the choice of the SOD kernel, the length of Ξ is static at 2δ . M and V_k^j can thus be chosen as static numbers. The points \hat{y}_k^j are chosen to be equidistant over the kernel interval, with the first and last points situated at the edges of the uniform distribution to ensure that the approximation does not lose accuracy at these vital places.

The number of discretization points was chosen to $M = 5$, with a manually tuned $V_k^j = 2\delta/5 \cdot 0.5, \forall j$. The choice of V_k^j is based on the length of Ξ , divided by the number of points scaled with a constant. The same Gaussian mixture approximation is used for both the APF and the EBSE filter. Since we are dealing with an LG system, a comparison is also made with a standard Kalman filter to form the baseline optimal estimator for $\delta = 0$.

The simulation study was implemented in Julia v.0.6.3 and can be found on Github¹.

Results

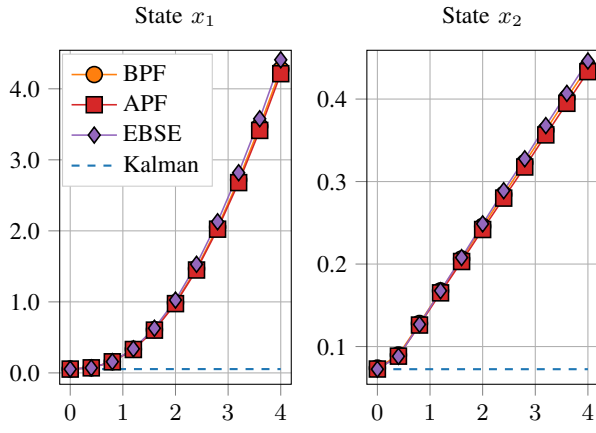
The results of the simulation study are displayed in Fig. 2 and 3, where the *mean squared error* (MSE) between the estimated and true states for the different estimators and parameter settings are shown. In Fig. 2, the MSE as a function over δ with $N = 500$ is depicted, while in Fig. 3, the MSE as a function over N with $\delta = 4.0$ is shown instead.

In Fig. 2a and 3a, the MSE was evaluated over all time steps. As can be seen, the choice of estimator only seems to have an impact when considering a low number of particles N . Otherwise the estimators have an equal performance. This is most likely due the nature of event triggering, which trades performance for an reduction in sent measurements. These “acceptable errors” defined by δ then overshadow any noticeable gains in using either of the considered estimators.

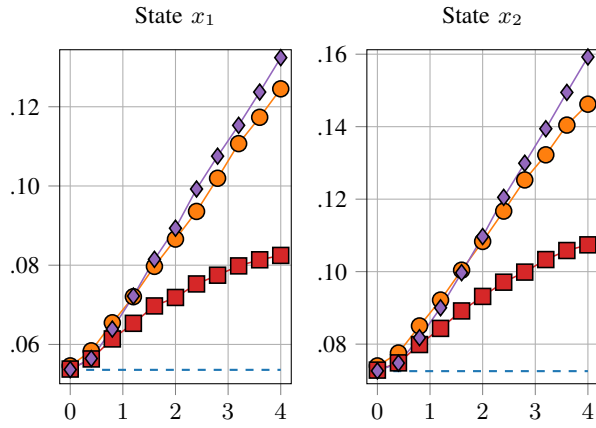
If we instead only consider the MSE at times k where $\gamma_k = 1$, thus examining the ability of the algorithms to adapt to new events, we get the results shown in Fig. 2b and 3b. Note that Fig. 3b is plotted in log scale. Here it is clear that the APF outperforms both the BPF and the EBSE filter in terms of estimation error. The error of the APF seems to evolve more slowly with increasing δ than the two other filters, and it manages with a fewer particles than the BPF. This is in accordance with our earlier motivation, as the APF manages to better capture the true underlying state right before an event by resampling conditioned on the new measurement. In comparison, the EBSE utilizes the Gaussian mixture approximation to create a single Gaussian approximation of the true state. However, right before an event, this single Gaussian approximation no longer gives an accurate representation of the true underlying state, as it will be close to the borders of the likelihood set.

Finally, in Fig. 2c, the average number of triggered events $\gamma_k = 1$ per time series for Fig. 2a and 2b is shown, and in Fig. 3c, the average number of numerical errors per time

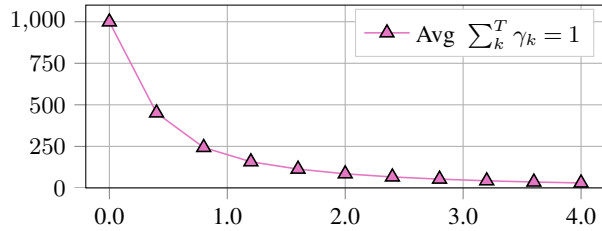
¹https://github.com/JohanRuuskanen/event-based_apf



(a) MSE considering all measurements z_k plotted over δ .

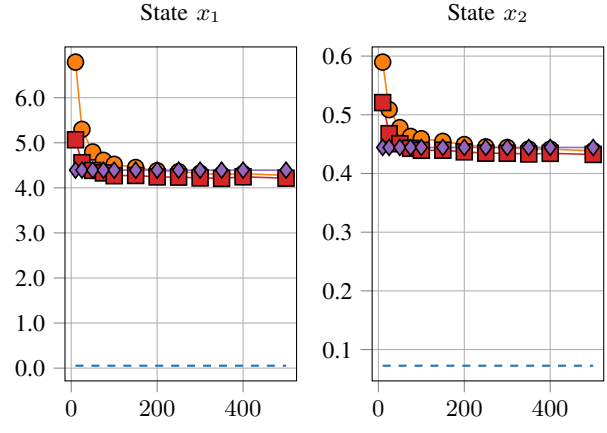


(b) MSE considering only measurements at triggered events $\gamma_k = 1$ plotted over δ .

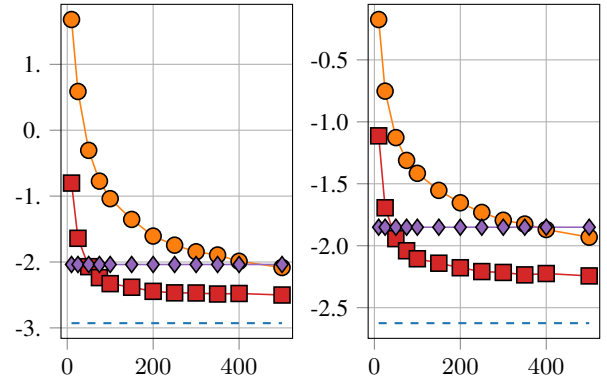


(c) Average number of triggered events per time series plotted over δ .

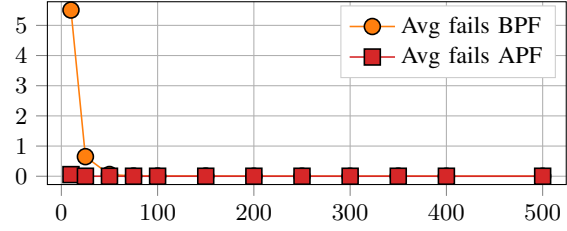
Fig. 2: Result of the simulation, plotted over the SOD event kernel width δ for a particle count of $N = 500$.



(a) MSE considering all measurements z_k plotted over N .



(b) Log-MSE considering only measurements at triggered events $\gamma_k = 1$ plotted over N .



(c) Average number of numerical errors per time series, plotted over N .

Fig. 3: Result of the simulation, plotted over the particle count N for a SOD kernel width of $\delta = 4$.

series for Fig. 3a and 3b is instead displayed. A numerical error happens when the likelihoods for all particles are too low, such that the normalization of $\{W_k^i\}_{i=1:N}$ becomes unfeasible as $\sum_{i=1}^N W_k^i \approx 0$. As can be seen, the number of numerical errors becomes fairly large at low N for the BPF while they are virtually non-existent for the APF.

VI. CONCLUSION & DISCUSSION

In this paper we have presented a general framework on how to extend the auxiliary particle filter to event-based systems with deterministic kernels via a Gaussian mixture approximation of the uniform distribution in the no-event likelihood. Existing methods for finding the proposal and approximate predictive likelihood that approximates the fully-adapted filter via a joint Gaussian naturally fall into this

framework. The results presented in the simulation study give strength to the motivation given in Sec. IV, as conditioning the proposal and resampling on the current measurement have a large impact on estimation accuracy when considering the immediate effect of a new event.

The strength of particle filters lies in their ability to tackle non-LG problems, so a natural question that arises is whether this approach can be used to solve event-based nonlinear state estimation problems. In the simulation study we only considered the LG case, mainly to demonstrate the benefits of the APF and the increasing degeneracy for the BPF in an event-based setting. But when deriving the event-based APF extension, no assumptions on linearity and Gaussianity are made—only that the transition and likelihood densities can

be approximated adequately as Gaussians by tuning α_1, α_2 . Extensions to certain well-behaved non-LG system should therefore be straightforward using already existing methods for finding proposals in these settings. However, more research is needed to validate this claim. Initial examination has shown that both the BPF and our event-based APF break down when considering typical nonlinear particle filter test systems sampled with common deterministic triggering rules. A probable root cause of this is the fact that common kernels are designed for linear systems. An interesting direction for future work could thus be to identify trigger kernels that can incorporate nonlinearities in a suitable manner.

A valid critique of our choice of approximating the uniform distribution with M Gaussians for deriving the proposal and approximate likelihood, is that each particle must be evaluated over these M points at each $\gamma_k = 0$, potentially increasing the computational burden from $\mathcal{O}(N)$ to $\mathcal{O}(NM)$ at these time instances. In our simulation study we chose M and V_k^j by hand, but an interesting question for future work would be to examine how M can be minimized by clever positioning of the discretization points and choice of V_k^j and still retain a good estimation accuracy. Another possibility to reduce the computational overhead is to consider heuristic solutions where for example the bootstrap filter could be used to propagate particles between new events, and the APF used when an event is triggered.

It is worth noting that the comparison with the BPF filter is not entirely fair, as the computational burden of the APF is greater. However, by timing the execution time for the APF and matching it to the computation time for the BPF by increasing its N and to the EBSE filter by increasing its M , we still retained an advantage for the APF, albeit smaller than in Fig. 2 and 3. To draw any definitive conclusions about such a comparison, time would need to be spent on optimizing the choice of M, V_k^j and the implementation.

Finally, particle filters are in general sensitive to increasing dimensions. For event-based systems, this *curse of dimensionality* should be even worse as the particles at $\gamma_k = 0$ will span a larger area in each dimension compared to $\gamma_k = 1$. Although the APF mitigates this problem somewhat, it is still dependent on there existing at least one particle close to the true underlying state that could be used to form the posterior approximation. It could thus be worth considering particle filtering methods that does not resample, but transform the particle set before propagation, e.g., the feedback particle filter [25], to increase robustness as the dimensions increase.

ACKNOWLEDGMENTS

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The authors are members of the LCCC Linnaeus Center and the ELLIIT Excellence Center at Lund University.

REFERENCES

- [1] Q. Liu, Z. Wang, X. He, and D. Zhou, "A survey of event-based strategies on control and estimation," *Systems Science & Control Engineering: An Open Access Journal*, vol. 2, pp. 90–97, 2014.
- [2] M. Miskowicz, "Send-on-delta concept: An event-based data reporting strategy," *Sensors*, vol. 6, pp. 49–63, 2006.
- [3] D. Han, Y. Mo, J. Wu, S. Weerakkody, B. Sinopoli, and L. Shi, "Stochastic event-triggered sensor schedule for remote state estimation," *IEEE Transactions on Automatic Control*, vol. 60, pp. 2661–2675, 2015.
- [4] D. Shi, L. Shi, and T. Chen, *Event-based State Estimation: A Stochastic Perspective*, ser. Studies in Systems, Decision and Control. Springer International Publishing, 2016, vol. 41.
- [5] R. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the AMSE-Journal of Basic Engineering*, no. Series D, pp. 35–45, 1960.
- [6] E. Wan and R. van der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, Lake Louise, Alberta, Canada, Oct. 2000, pp. 153–158.
- [7] A. Doucet and A. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," in *The Oxford Handbook of Nonlinear Filtering*, D. Crisan and B. Rozovsky, Eds. Oxford University Press, 2008.
- [8] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, Sep. 2004.
- [9] R. E. Curry, *Estimation and Control with Quantized Measurements*. The MIT Press, 1970.
- [10] R. Gopaluni, T. B. Schön, and A. G. Wills, "Particle filter approach to nonlinear system identification under missing observations with a real application," *IFAC Proceedings Volumes*, vol. 42, no. 10, pp. 810–815, 2009.
- [11] R. Karlsson and F. Gustafsson, "Particle filtering for quantized sensor information," in *2005 13th European Signal Processing Conference*, Sep. 2005, pp. 1–4.
- [12] M. Morin, "Optimal control under quantised measurements," Master's thesis TFRT-6031, Department of Automatic Control, Lund University, June 2017.
- [13] M. Cea and G. Goodwin, "Event based sampling in non-linear filtering," *Control Engineering Practice*, vol. 20, pp. 963–971, 2012.
- [14] M. A. Sid and S. Chitraganti, "Nonlinear event-based state estimation using particle filtering approach," in *8th International Conference on Modeling, Identification and Control*, Algiers, Algeria, Nov. 2016, pp. 874–879.
- [15] S. Davar and A. Mohammadi, "Event-based particle filtering with point and set-valued measurements," in *25th European Signal Processing Conference*, Kos, Greece, Aug. 2017, pp. 211–215.
- [16] X. Liu, L. Li, Z. Li, X. Chen, T. Fernando, H. Iu, and G. He, "Event-trigger particle filter for smart grids with limited communication bandwidth infrastructure," *IEEE Transactions on Smart Grid*, vol. 9, pp. 6918 – 6928, Nov. 2018.
- [17] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *IEE Proceedings F - Radar and Signal Processing*, vol. 140, 1993, pp. 107 – 113.
- [18] M. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 590–599, 1999.
- [19] J. Sijts and M. Lazar, "Event based state estimation with time synchronous updates," *IEEE Transaction on Automatic Control*, vol. 57, pp. 2650–2655, Oct. 2012.
- [20] N. Whiteley and A. Johansen, "Recent developments in auxiliary particle filtering," D. Barber, A. T. Cemgil, and S. Chiappa, Eds. Cambridge University Press, 2011, ch. 3, p. 5281.
- [21] M. Eaton, *Multivariate Statistics: a Vector Space Approach*. John Wiley and Sons, 1983.
- [22] P. Bromiley, "Products and convolutions of Gaussian probability density functions," Division of Informatics, Imaging and Data Sciences, School of Health Sciences, University of Manchester, Tech. Rep., 2003, last updated 22 / 6 / 2018.
- [23] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, Jul 2000.
- [24] S. Saha, P. K. Mandal, Y. Boers, H. Driessen, and A. Bagchi, "Gaussian proposal density using moment matching in SMC methods," *Statistics and Computing*, vol. 19, no. 2, pp. 203–208, Jun 2009.
- [25] T. Yang, P. G. Mehta, and S. P. Meyn, "Feedback particle filter," *IEEE Transactions on Automatic Control*, vol. 58, no. 10, pp. 2465–2480, Oct 2013.