



LUND UNIVERSITY

On Innovation-Based Triggering for Event-Based Nonlinear State Estimation Using the Particle Filter

Ruuskanen, Johan; Cervin, Anton

Published in:
2020 19th European Control Conference (ECC)

2020

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

[Link to publication](#)

Citation for published version (APA):
Ruuskanen, J., & Cervin, A. (2020). On Innovation-Based Triggering for Event-Based Nonlinear State Estimation Using the Particle Filter. In *2020 19th European Control Conference (ECC)*

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

On Innovation-Based Triggering for Event-Based Nonlinear State Estimation Using the Particle Filter

Johan Ruuskanen¹ and Anton Cervin¹

Abstract—Event-based sampling has been proposed as a general technique for lowering the average communication rate, energy consumption and computational burden in remote state estimation. However, the design of the event trigger is critical for good performance. In this paper, we study the combination of innovation-based triggering and state estimation of nonlinear dynamical systems using the particle filter. It is found that innovation-based triggering is easily incorporated into the particle filter framework, and that it vastly outperforms the classical send-on-delta scheme for certain types of nonlinear systems. We further show how the particle filter can be used to jointly precompute the future state estimates and trigger probabilities, thus eliminating the need for periodic observer-to-sensor communication, at the cost of increased computational burden at the observer. For wireless, battery-powered sensors, this enables the radio to be turned off between sampling events, which is key to saving energy.

I. INTRODUCTION

State estimation is a ubiquitous problem in tracking, control, supervision, and forecasting applications. For a regularly sampled, linear and Gaussian (LG) system, the Kalman filter [1] provides an optimal, closed-form solution. For nonlinear and non-Gaussian systems, the particle filter [2] offers a general—albeit computationally expensive—Monte Carlo-based numerical solution.

When considering remote state estimation in resource-constrained environments, it is important to minimize the communication overhead to save bandwidth and energy. Such scenarios are common in Internet of Things (IoT) applications, where sensors often are cheap, wireless and battery powered [3]. One promising solution is to employ *event-based sampling* [4], in which a new sample is sent only when it carries enough new information.

What constitutes sufficiently new information to trigger a sample is in general an open problem, and thus event-based sampling can be performed in a multitude of ways. Broadly, the methods proposed in the literature can be split into *open-loop* and *closed-loop* triggering [5]. Open-loop triggering includes triggers where no new information transfer between the sensor and observer is needed in between sampling events. Such triggers include the simple yet popular send-on-delta (SOD) method [6] as well as variance-based triggering [7]. Closed-loop triggering on the other hand tries to improve the sampling decision by incorporating some observer information at the sensor. Such triggers include the well-known innovation-based trigger (IBT) [8], [9] and triggers based on the Kullback–Leibler divergence [10], [11].

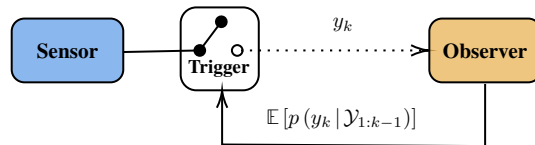


Fig. 1: Remote event-based state estimation with innovation-based triggering. The event trigger is dependent on the estimation of the measurement, which is transferred back from the observer to the sensor.

A clear downside with the closed-loop triggers is the reliance on the estimated state, which needs to be available at the sensor. This is usually solved by either letting the observer send information continuously to the sensor or equipping the sensor with a local filter, see Fig. 1 for an illustration. Both these standard approaches have their downsides if a low-cost, low-powered sensor is desired. Sending information online requires communication between the observer and the sensor at all sampling instances, which strains the energy and bandwidth, and a local filter requires hardware and energy to operate. The effects of the local filter solution might be considered negligible when a computationally cheap state estimator is used, but for nonlinear state estimation where the particle filter is employed this is not the case.

The combination of event-based sampling and particle filtering has recently been explored in several papers. In [12] and [13], the authors derive bootstrap particle filters under the SOD scheme. In [14], IBT is instead used, but the choice of trigger is not analyzed further. These filters suffer from particle degeneracy at event instances; in our previous work [15] we tackled this issue by using an auxiliary particle filter. Li et al. [11] introduced a more advanced event trigger suited for skewed or multi-modal measurement densities. Based the Kullback–Leibler divergence, a two-step scheme is proposed, where a measurement-based test function is first approximated by its particle representation and then further approximated by a spline. The spline parameters are to be broadcast to the sensor nodes at each time step, which reconstructs the function and determines whether an event should be triggered. Although the approach is promising for systems with skewed or multi-modal measurement densities, the sensor sampling criterion is costly to evaluate and for the general case, not guaranteed to outperform simpler schemes such as the IBT which is shown by the authors.

In this paper we focus on the benefits of innovation-based triggering for remote nonlinear state estimation, compared to the simpler send-on-delta scheme. Special attention is given to the closed-loop trigger problem, which is a major

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

hindrance in the usability of schemes such as IBT. In particular, we aim at providing insight with the following contributions:

- We show that there exist simple nonlinear systems for which the SOD strategy performs *arbitrarily* bad in terms of communication reduction (Sec. III-A).
- We demonstrate how IBT can be achieved for a general, nonlinear system (Sec. III-B). Although IBT has been proposed for these settings before [11], [14], we here show how it can be derived from a particle-based approximation of the predictive likelihood density.
- We provide an energy-efficient and simple solution to the closed-loop triggering problem (Sec. IV). The particle filter can be used to jointly estimate the state and trigger probabilities, which allows for the states and trigger rule to be precomputed over a well-chosen time horizon.
- Finally, we perform a simulation study (Sec. V) to empirically demonstrate our findings, and to empirically quantify the effect of particle degeneracy on the estimation quality.

II. BACKGROUND

A. Preliminaries

The following definitions will be used throughout the paper. Assume $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$, let $k_a : k_b = [k_a, k_b] := \{\forall k \in \mathbb{N}, \text{ s.t. } k_a \leq k \leq k_b\}$, let $\{a^i\}_{i \in 1:b}$ be the set of values a^i in the interval $i \in [1, b]$, and let $p(x) \propto q(x)$ denote that two functions p, q are proportional, i.e., equal up to a constant factor. Further, let the density $p(x_k | x_{k-1})$ be referred to as the *transition density*, $p(y_k | x_k)$ the *likelihood density* and $p(y_k | x_{k-1})$ or $p(y_k | y_{k-1})$ as the *predictive likelihood density*. We assume a state-space model

$$\begin{aligned} x_{k+1} &= f_k(x_k, w_k) \\ y_k &= h_k(x_k, v_k) \end{aligned} \Leftrightarrow \begin{aligned} x_k &\sim p(x_k | x_{k-1}) \\ y_k &\sim p(y_k | x_k) \end{aligned} \quad (1)$$

where w_k and v_k are independent white noise processes with known distributions. Further, $p(x_k | x_{k-1})$ and $p(y_k | x_k)$ are considered known.

Our goal is to estimate the state given the measurements, i.e., finding $p(x_k | y_{1:k})$, $\forall k \in [1:T]$. Using Bayes' theorem, the recursive filtering equation is given by

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

where

$$p(x_k | y_{1:k-1}) = \int p(x_k | x_{k-1}) p(x_{k-1} | y_{1:k-1}) dx_{k-1}$$

Assuming an LG system, $p(x_k | y_{1:k})$ has a closed-form expression given by the Kalman filter. For a nonlinear and/or non-Gaussian system, solving the recursive filtering equation is in general intractable.

B. Event-based state estimation

In event-based sampling, a measurement is considered known when it triggers some predefined trigger rule. Let H_k be the set of all possible values of y_k at time instance k that

do not result in a trigger. Further, define $\gamma_k = 1$ as instances where $y_k \notin H_k$ and $\gamma_k = 0$ when $y_k \in H_k$.

Given that the trigger rule is known, instances where $\gamma_k = 0$ also carries information in the form of H_k . This can be captured by defining a hybrid variable $\mathcal{Y}_k | (\gamma_k = 1) = \{y_k\}$ and $\mathcal{Y}_k | (\gamma_k = 0) = H_k$. The likelihood of \mathcal{Y}_k can be expressed as

$$p(\mathcal{Y}_k | x_k) \propto \int p(\mathcal{Y}_k | y_k) p(y_k | x_k) dy_k$$

Following [16], $p(\mathcal{Y}_k | y_k)$ can be rewritten as

$$p(\mathcal{Y}_k | y_k) = \begin{cases} \delta_{y_k}(x) & \gamma_k = 1 \\ \mathcal{U}(y_k \in H_k) & \gamma_k = 0 \end{cases}$$

where $\mathcal{U}(a \in A)$ is the uniform distribution over the set A . This implies that

$$p(\mathcal{Y}_k | x_k) \propto \begin{cases} p(y_k | x_k) & \gamma_k = 1 \\ \int_{y_k \in H_k} p(y_k | x_k) dy_k & \gamma_k = 0 \end{cases} \quad (3)$$

and the recursive filtering equation for event-based state estimation becomes

$$p(x_k | \mathcal{Y}_{1:k}) = \frac{p(\mathcal{Y}_k | x_k) p(x_k | \mathcal{Y}_{1:k-1})}{p(\mathcal{Y}_k | \mathcal{Y}_{1:k-1})} \quad (4)$$

For SOD and IBT, H_k can be obtained as

$$\begin{aligned} H_k^{\text{SOD}} &:= \{y_k \in \mathbb{R}^m, \text{ s.t. } \|F_k(y_k^- - y_k)\|_\infty \leq \Delta\} \\ H_k^{\text{IBT}} &:= \{y_k \in \mathbb{R}^m, \text{ s.t. } \|F_k(\hat{y}_k - y_k)\|_\infty \leq \Delta\} \end{aligned}$$

where y_k^- is the previous triggered measurement, $\hat{y}_k = \mathbb{E}(p(y_k | \mathcal{Y}_{1:k-1}))$ and F_k is some weight matrix for normalization normally chosen as the solution to $F_k^T F_k = [C(\hat{y}_k - y_k | \mathcal{Y}_{1:k-1})]^{-1}$ [17].

C. The particle filter

One popular choice for dealing with the intractable recursive filtering equation (2) is the particle filter [2], which approximates $p(x_k | y_{1:k})$ as an empirical distribution of a set of particles X_k^i with corresponding weights W_k^i for $i \in [1:N]$, i.e., $p(x_k | y_k) \approx \sum_{i=1}^N W_k^i \delta_{X_k^i}(x)$. Given $\{W_{k-1}^i, X_{k-1}^i\}_{i \in 1:N}$, for which $p(x_{k-1} | y_{1:k-1}) \approx \sum_{i=1}^N W_{k-1}^i \delta_{X_{k-1}^i}$ is known, a particle-based approximation $p(x_k | y_{1:k}) \approx \sum_{i=1}^N W_k^i \delta_{X_k^i}$ can be obtained as follows:

- 1) Resample the particles by drawing new indices with replacements $j = \mathcal{C}(i = W_{k-1}^i)$ to combat degeneracy and to create $p(x_{k-1} | y_{k-1}) \approx \sum_{j=1}^N \delta_{X_{k-1}^j}$.
- 2) Draw X_k^i for $i \in 1:N$ from some proposal density $q(x_k | X_{k-1}^j, y_k)$.
- 3) Calculate new weights $W_k^j \propto \frac{p(y_k | X_k^j) p(X_k^j | X_{k-1}^j)}{q(X_k^j | X_{k-1}^j, y_k)}$.

By initially drawing $X_0^i \sim p(x_0)$, it can under some minor assumptions on the state-space model and $q(\cdot)$ be shown that $\lim_{N \rightarrow \infty} \hat{p}_N(x_{1:k} | y_{1:k}) = p(x_{1:k} | y_{1:k})$ [18].

The event-based system can be seen as a simple hybrid system, where the likelihood changes according to Eq. (3). Applying the particle filter to event-based state estimation is

thus quite straightforward when considering SOD triggering and the bootstrap filter [19], where $q(x_k | x_{k-1}, y_k) = p(x_k | x_{k-1})$. If $\gamma_k = 0$ and $m = 1$, the likelihood can be obtained in closed form as the CDF of $p(y_k | x_k)$. Otherwise, if $m > 1$, the integral in (3) will have to be numerically evaluated for the general case.

III. ON SEND-ON-DELTA AND INNOVATION-BASED TRIGGERING FOR NONLINEAR SYSTEMS

Innovation-based triggering has repeatedly been shown to outperform send-on-delta [11][20][21]. In fact, when considering nonlinear systems, there exist simple types of systems where the SOD strategy in theory performs arbitrarily bad.

A. Send-on-delta shortcomings

Since SOD triggers directly on the deviation between the current measurement and the previous triggering measurement, systems with frequently occurring and predictable large deviations will cause many unnecessary events compared to if the prediction was taken into account. This is especially true for systems where the measurement function has a periodic behaviour. If the system is also deterministic, the SOD trigger time actually becomes bounded above by the period.

Proposition 1: Assume a time-continuous, deterministic system with output $y(t) \in \mathbb{R}^m$ and denote A as the set of all possible values of $y(t)$. Let A be bounded, such that $\max_{u, v \in A} \|u - v\| = D < \infty$. Further, let $y(t)$ be periodic with period t_p , such that $y(t) = y(t + t_p) \forall t$. If the system is then sampled under SOD, the longest achievable *finite* trigger time is bounded by t_p .

Proof: Using SOD over a bounded measurement yields a clear distinction in the choice of Δ . Either $\Delta > D$, and the event can never occur, yielding an infinite trigger time, or $\Delta < D$ and we will at some point trigger, giving a finite trigger time. Because of the periodicity, all possible values of $y(t)$ will occur within one period, thus if $\Delta < D$ then $D \geq \|y(t) - y(t + t_n)\| > \Delta$ is true for some $t_n < t_p$. ■

The proposition above is only strictly valid for deterministic systems, but it gives some intuition about which types of systems the SOD method should be avoided if a low communication rate is desired. Using this insight, the following simple yet pathological nonlinear system can be constructed:

$$\begin{aligned} x_{k+1} &= f(x_k, w_k) \\ y_k &= A \cos\left(\frac{2\pi k}{T_p} + x_k\right) + v_k \quad v_k \sim \mathcal{N}(0, R) \end{aligned} \quad (5)$$

This describes a sinusoidal with a state-dependent phase shift. Assuming that we sample the system using SOD and that $R \ll A$, increasing $\Delta \leq 2A$ will not reduce the amount of triggering below roughly two per period T_p . Further, setting $\Delta > 2A$ ensures that we will almost never trigger. By reducing T_p , the SOD method is forced to trigger more often.

The innovation-based triggering does not have the same difficulties with these types of systems, as long as the

measurement is predictable. This difference is illustrated in Fig. 2.

B. Innovation-based triggering using the particle filter

To use the innovation-based trigger, the mean (and sometimes also the covariance) of the predictive likelihood $p(y_k | \mathcal{Y}_{1:k-1})$ needs to be estimated. The predictive likelihood can be expressed as follows:

$$\begin{aligned} p(y_k | \mathcal{Y}_{1:k-1}) &= \int p(y_k, x_{1:k} | \mathcal{Y}_{1:k-1}) dx_{1:k} \\ &\propto \int p(y_k | x_k, x_{1:k-1}) p(x_k | x_{1:k-1}, \mathcal{Y}_{1:k-1}) \\ &\quad p(x_{1:k-1} | \mathcal{Y}_{1:k-1}) dx_{1:k} \\ &\propto \int p(y_k | x_k) p(x_k | x_{k-1}) p(x_{1:k-1} | \mathcal{Y}_{1:k-1}) dx_{1:k} \end{aligned} \quad (6)$$

Since the state density $p(x_{1:k-1} | \mathcal{Y}_{1:k-1})$ is in general not known, finding a closed expression of the first moment of $p(y_k | \mathcal{Y}_{1:k-1})$ is tricky. Instead, we can utilize our particle approximation at time step $k-1$ to generate an empirical approximation of the predictive likelihood density. Given

$$p(x_{1:k-1} | \mathcal{Y}_{1:k-1}) \approx \sum_{i=1}^N W_{1:k-1}^i \delta_{X_{1:k-1}^i}(x_{1:k-1})$$

the integral (6) reduces to

$$\begin{aligned} &\int p(y_k | x_k) p(x_k | x_{k-1}) p(x_{1:k-1} | \mathcal{Y}_{1:k-1}) dx_{1:k} \\ &\approx \int p(y_k | x_k) \sum_{i=1}^N W_{k-1}^i p(x_k | X_{k-1}^i) dx_k \end{aligned}$$

From here, numerical integration can be performed by sampling from $\sum_{i=1}^N W_{k-1}^i p(x_k | X_{k-1}^i)$. The easiest way to achieve this is to perform a particle resampling/propagation step with $q(x_k | x_{k-1}, y_k) = p(x_k | x_{k-1})$, which is obtained at no extra computational cost if the bootstrap filter is used. This yields the following approximation to the predictive likelihood:

$$\begin{aligned} p(y_k | \mathcal{Y}_{1:k-1}) &\approx \int p(y_k | x_k) \frac{1}{N} \sum_{i=1}^N \delta_{X_k^i} dx_k \\ &= \frac{1}{N} \sum_{i=1}^N p(y_k | X_k^i), \quad X_k^i \sim p(x_k | X_{k-1}^i) \end{aligned} \quad (7)$$

This is a mixture model of equally weighted likelihoods over the particles. The mean and covariance of the predictive likelihood can thus be estimated as [22]

$$\mathbb{E}(p(y_k | \mathcal{Y}_{1:k-1})) \approx \frac{1}{N} \sum_{i=1}^N \mu_k^i = \hat{\mu}_k \quad (8)$$

$$\text{where } \mu_k^i = \mathbb{E}(p(y_k | X_k^i))$$

$$\begin{aligned} \mathbb{C}(p(y_k | \mathcal{Y}_{1:k-1})) &\approx \frac{1}{N} \sum_{i=1}^N \left[(\mu_k^i - \hat{\mu}_k) (\mu_k^i - \hat{\mu}_k)^T \right. \\ &\quad \left. + \mathbb{C}(p(y_k | X_k^i)) \right] \end{aligned} \quad (9)$$

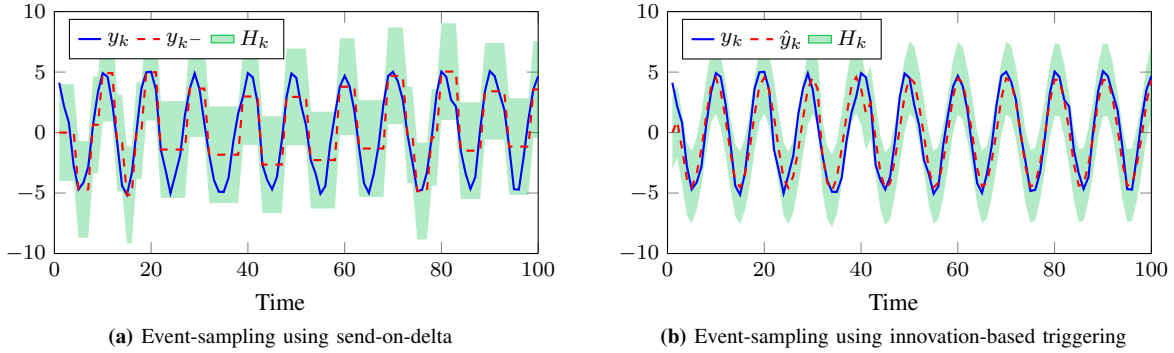


Fig. 2: Comparison between event-triggering instances on the pathological system (5) with $f(x_k, w_k) = 0.99x_k + w_k$, where $w_k \sim \mathcal{N}(0, 0.2)$ and $A = 5$, $T = 10$, $R = 0.1$. The blue line shows the measurement, the dashed red line the comparison value and the green area displays the set H_k . If the measurement passes outside H_k , a new measurement event is triggered.

IV. PRECOMPUTING THE STATE ESTIMATE AND TRIGGER

Using closed-loop triggering implies that the trigger mechanism has access to information dependent on the estimated state, which in the literature has often been achieved by either transferring information from the observer to the sensor at every time step or by running a local filter at the sensor. Both approaches have clear downsides in real settings. However, it is entirely possible to circumvent these problems if the observer is allowed to perform some extra computations.

Note that, for a given time horizon n where $\gamma_{k:k+n} = 0$, the entire state estimation problem for $p(x_{k:k+n} | \mathcal{Y}_{1:k+n})$ is independent of the sensor measurements. Thus, for a given n , the entire posterior $p(x_{k:k+n} | \mathcal{Y}_{1:k+n}, \gamma_{k:k+n} = 0)$ can be obtained at time k . When the event then eventually triggers at $t + 1$, the already obtained $p(x_{k:k+t} | \mathcal{Y}_{k:k+t})$ is the optimal solution to the state estimation problem in $[k, k + t]$. This is formalized in the theorem below.

Theorem 1: Given that no triggering occurs at time instances $[k, k + n]$, then the recursive filtering equation for calculating $p(x_{k:k+n} | \mathcal{Y}_{1:k+n}, \gamma_{k:k+n} = 0)$ can be obtained at time k .

Proof: Quite trivially, no triggering implies that $p(\mathcal{Y}_{k+i} | x_{k+i}) = p(y_{k+i} \in H_{k+i} | x_{k+i}) \forall i \in [k, k + n]$. Further, the sets $H_{k:k+n}$ are constructed using only information at the observer. By using the recursive filtering equation, we get that

$$p(x_k | \mathcal{Y}_{1:k}) \propto p(y_k \in H_k | x_k) \cdot \int p(x_k | x_{k-1}) p(x_{k-1} | \mathcal{Y}_{1:k-1}) dx_{k-1}$$

By recursion, $p(x_{k+i} | \mathcal{Y}_{1:k+i})$, $i \in [1, n]$ can be obtained in the same manner using only information local to the observer. ■

Corollary 1.1: Given that $\gamma_k = 1$ and assuming that $\gamma_{k+1:k+n} = 0$, the posterior $p(x_{k:k+n} | \mathcal{Y}_{1:k+n})$ can be estimated at time k . Thus, the trigger rule for $k + 1 : k + n$ can be calculated and sent to the sensor at time k , which removes both the need for a local filter at the sensor and the need to keep a communication channel open from the observer to the sensor.

This simple solution comes at a cost, as it is not known when the sensor will actually trigger. Choosing n becomes a

trade off between wasted computations at the observer if we trigger before n , and quality if all n steps are utilized and the sensor is forced to trigger prematurely. In systems where computationally cheap estimators are used, n can simply be set to some arbitrarily and sufficiently large value such that we very seldom hit it. However, if we use a computationally heavy algorithm such as the particle filter, setting n such that the amount of wasted computations becomes manageable can be difficult. Further, the probability of triggering at time $k + i$ might very well change depending on $p(x_k | y_k)$, making it even harder to find a static n that will be sufficient at every k .

When using the particle filter, n can be dynamically chosen based on the estimated trigger probability. First consider the probability of triggering in the next time step, $p(\gamma_k = 1 | x_{k-1})$. Using Bayes' theorem and the marginal distribution, this probability can be expressed as

$$p(\gamma_k = 1 | x_{k-1}) = p(y_k \notin H_k | x_{k-1}) \propto \int p(y_k | x_{k-1}) \mathbb{1}(y_k \notin H_k) dy_k$$

By utilizing the particle approximation of the predictive likelihood (7), this probability can be estimated as

$$p(y_k \notin H_k | x_{k-1}) \approx \frac{1}{N} \sum_{i=1}^N \int_{y \notin H_k} p(y_k | X_k^i) dy_k \quad (10)$$

where $X_k^i \sim p(x_k | X_{k-1}^i)$

For one-dimensional measurements where the primitive to the likelihood density is known, the trigger probability estimate (10) can be calculated via the CDF. For higher dimensions, numerical methods would in the general case have to be used.

By Corollary 1.1, it is then possible to estimate $p(\gamma_i = 1 | x_k)$, $i \in [k, k + n]$ at time k . Due to the binary nature of event sampling, the probability of triggering for the first time at n can then be calculated as

$$p_T(n | x_k) = p(\gamma_{k+n} = 1 | x_k) \prod_{i=1}^{n-1} [1 - p(\gamma_{k+i} = 1 | x_k)]$$

The cumulative sum of $p_T(n)$ will then give the probability of triggering at or before the time horizon n . It is then

possible to choose an desired quantile value α and decide n via the generalized inverse distribution function

$$n_\alpha = \inf \left\{ n \in \mathbb{Z}_+ : \sum_{i=1}^n p_T(i | x_k) \geq \alpha \right\} \quad (11)$$

This ensures that the precomputed values will be sufficiently many in a fraction α of the instances.

The amount of wasted computations can be further characterized. Let the stochastic variable t be the number of time steps until triggering. Given the maximum horizon n_α , the probability mass function can be obtained as

$$p_w(t=i) = \begin{cases} p_T(i) & \text{if } i < n_\alpha \\ 1 - \sum_{j=1}^{n_\alpha-1} p_T(j) & \text{if } i = n_\alpha \\ 0 & \text{if } i > n_\alpha \end{cases}$$

The unnecessarily computed steps for each horizon is then given by the stochastic variable $n_\alpha^- = n_\alpha - t$. Since n_α is a constant, different statistics such as the mean number of wasted steps can be easily calculated through $p_w(t)$.

Iterative algorithm and its real-time properties

The fact that both $p(\gamma_{k+i} = 1 | x_k)$ and \hat{y}_{k+i} can be estimated using only the approximation $\hat{p}(x_{k+i} | \mathcal{Y}_{1:k+i})$ makes it possible to choose an α , and then iteratively perform the precomputations to calculate the estimates, as shown in Algorithm 1.

In a real scenario, the *receive* command would have to include some fine-tuned waiting time, as no received measurement within that interval at the observer would imply that $\gamma_k = 0$. Further, it is standard to assume that $p(x_0)$ is not known. Instead an approximation will have to be used and the particle filter will need some *burn-in* time for the posterior estimation to converge. This could be managed by setting $\gamma_{1:T_b} = 1$ for some burn-in time horizon T_b .

The precomputation scheme introduces some new problems concerning real-time guarantees. Since no upper bound on n_α exists, the computation time of generating the posterior approximations is potentially unbounded. One quick fix is to introduce an upper bound on n_α as $\max n_\alpha = \lfloor D/C \rfloor$, where D is the sample deadline and C the computation time of one iteration. However, there is no guarantee that $\max n_\alpha$ gives an satisfactory horizon in terms of estimation-vs-communication trade off. Instead if $\max n_\alpha$ is too small due to computational constraints but $C < D$ still holds, it is possible to stream \hat{y}_{k+i} to the sensor as its computation completes. In this case, the choice of n_α would have to consider the cost of running the sensor radio for longer than a single sample period.

V. SIMULATION STUDY

The simulation study consists of two parts. First, the validity of the claims in this paper are checked: in Sec. V-A an empirical comparison between the innovation-based triggering and send-on-delta is performed, while in Sec. V-B the accuracy of how well n_α corresponds to the true triggering quantile is examined. Second, in Sec. V-C the effect of

```

Draw  $\{X_0^i\}_{i \in 1..N}$  from the prior  $p(x_0)$ ;
for  $k = 1:T$  do
  At Observer:
  if receive:  $y_k$  from sensor then
    | set  $\gamma_k = 1$ 
  else
    | set  $\gamma_k = 0$ 
  end
  if  $\gamma_k = 1$  or  $k = 1$  then
    |  $\hat{p}(x_k | \mathcal{Y}_{1:k})$  using the particle filter;
    | Set  $n_\alpha = 1, p_N = 1, p_T = 0$ ;
    | while  $p_T < \alpha$  do
      |  $\hat{p}(x_{k+n_\alpha} | \mathcal{Y}_{1:k+n_\alpha})$  using the particle filter;
      |  $\hat{y}_{k+n_\alpha}$  using Eq. (8);
      |  $p_T = p_N \cdot \hat{p}(\gamma_{k+n_\alpha} = 1)$  using Eq. (10);
      |  $p_N := p_N \cdot (1 - \hat{p}(\gamma_{k+n_\alpha} = 1))$ ;
      |  $n_\alpha := n_\alpha + 1$ ;
    | end
    | send:  $\hat{y}_{k:k+n_\alpha}$  to sensor
  end
  At Sensor:
  if  $y_k \notin H_k$  or  $k > i$  or  $k = 1$  then
    | send:  $y_k$  to observer
    | receive:  $\hat{y}_{k:k+n_\alpha}$  from observer
    | set  $i = k + n_\alpha$ ;
  end
end

```

Algorithm 1: Event-based particle filter, with the iterative n -step state estimation precomputation.

particle degeneracy on the estimated trigger probabilities is examined.

The particle filtering was performed using the proposal density $q(x_k | x_{k-1}, y_k) = p(x_k | x_{k-1})$, also known as the bootstrap particle filter [19]. This has its limitations when it comes to event-based systems as discussed by the authors in [15]; however, for the scope of this paper the filtering is simply run with an increased particle count to combat the imposed degeneracy at sample events.

In the study, the following two systems will be considered.

System 1: The pathological example system (5), with

$$\begin{aligned} x_{k+1} &= 0.99x_k + w_k, & w_k &\sim \mathcal{N}(0, 0.2) \\ y_k &= 5 \cos\left(\frac{2\pi k}{10} + x_k\right) + v_k, & v_k &\sim \mathcal{N}(0, 0.1) \end{aligned}$$

System 2: The classical, multi-modal nonlinear system often used for benchmarking particle filters [11], [14], [18], [19]:

$$\begin{aligned} x_{k+1} &= \frac{x_k}{2} + \frac{25x_k}{1+x_k^2} + 8 \cos(1.2k) + w_k, & w_k &\sim \mathcal{N}(0, 1) \\ y_k &= \frac{x_k^2}{20} + v_k, & v_k &\sim \mathcal{N}(0, 0.1) \end{aligned}$$

A. Comparison of SOD and IBT

The comparisons are made in the form of mean estimation-vs-communication trade-off diagrams. The target system was simulated over a time period of $T = 10000$ with $N = 10000$

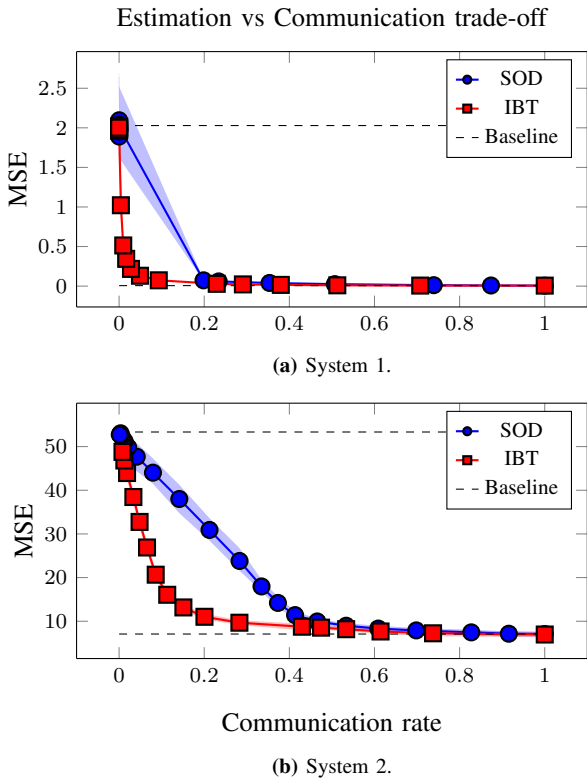


Fig. 3: Communication rate versus mean squared error for send-on-delta and innovation-based triggering. Each dot corresponds to a specific Δ . The opaque areas show the interval between the upper and lower 5th percentiles of the corresponding estimation error. The baselines are the estimation errors obtained by sampling at every time instance or none.

particles. The *mean squared error* (MSE) of the state estimation and the fraction of time instances where $\gamma_k = 1$ are logged in the simulation. This was performed over a set of Δ values and repeated 50 times for each value in order to capture estimation variability.

The results are shown in Fig. 3. Specific Δ values are not shown, as their corresponding performance is unique to both the trigger rule and the system. From the figure it can clearly be seen that IBT outperforms SOD for both systems, which agrees with the results from [11]. For the pathological system in Fig. 3a it can as expected be seen that the error and communication rate suddenly jump to the open-loop baseline when $\Delta > 2A$.

B. Accuracy of the precomputation horizon

The target system was simulated over a time period of $T = 10000$ and the estimation performed using Algorithm 1 with $N = 10000$. Each n_α and the corresponding actual triggering instance n_t were recorded. From the set of such pairs $\{n_\alpha, n_t\}$, the quantile can be estimated from the fraction of instances where $n_t < n_\alpha$. The estimation was run over a set of quantile values, and repeated 50 times for each value to capture the estimation variability.

The results are reported in Fig. 4, and it is seen that the accuracy of the estimated triggering quantile seems well in line with the desired value. Because of the discrete nature

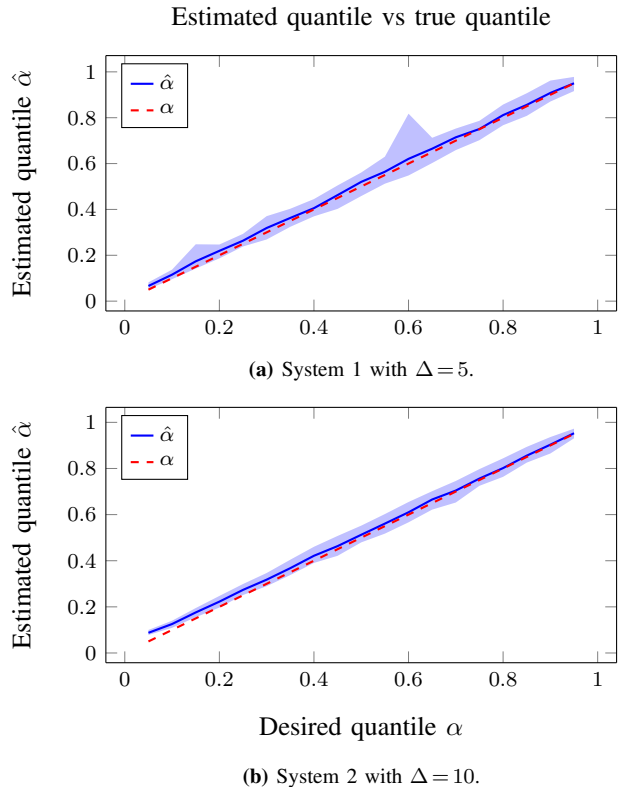


Fig. 4: The estimated trigger probability quantile from the simulation compared to the desired quantile when generating the precomputation horizon. The opaque areas shows the interval between the upper and lower 5th percentiles of the estimated quantile.

of n_α and how it is defined, we will actually trigger at a slightly larger quantile value than the desired one, which is reflected in the figure.

C. Effects of particle degeneracy

The precomputation scheme relies on the state approximation at $\gamma_k = 1$ to generate $\hat{p}_T(n | x_k)$. As we show in [15], at the sampling event instances we are likely to have a poor particle representation, and it is thus of high interest to evaluate the effect of particle degeneracy on the estimation of the trigger probability.

To emulate degeneracy at events, the target system is simulated over the short time frame of $T = 100$ where $\gamma_0 = 1$ is assumed. Because the system is simulated, $p(x_0)$ is known and an empirical distribution easily generated from it. Degeneracy can then be introduced by subsampling the empirical distribution $\{X_0^i\}_{i=1..N}$ such that only $M < N$ unique particles exist. From the degenerate representation of $p(x_0)$ the probability $p_T(n | x_0)$ can be estimated.

Event-sampling under IBT is however itself dependent on $\{X_0^i\}_{i=1..N}$, as it is used to calculate \hat{y}_k . Thus care needs to be taken when comparing the true $p_T(n | x_0)$ with the particle filter estimation $\hat{p}_{PF}(n | x_0)$. They both need to be conditioned on *the same* $\{X_0^i\}_{i=1..N}$ to perform a fair comparison.

Further, obtaining a closed-form solution to the trigger probability $p_T(n | x_k)$ is in general intractable. Instead, the

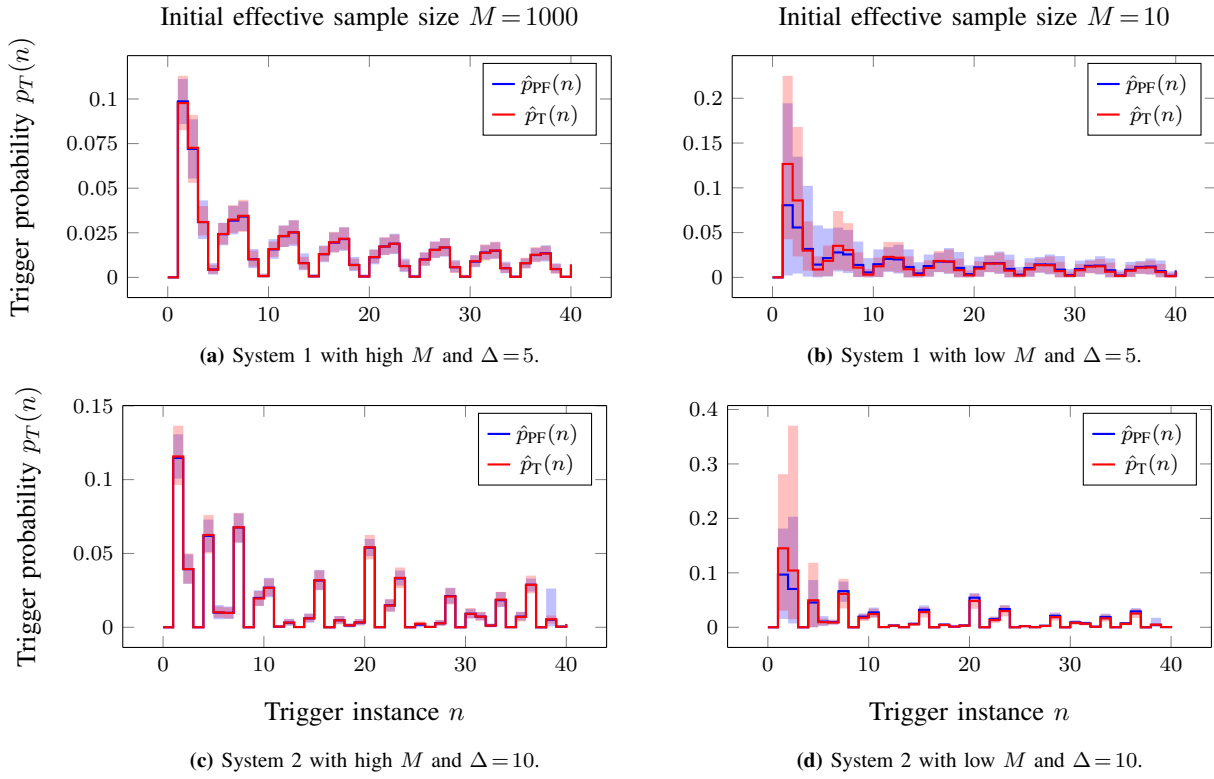


Fig. 5: Effect of initial particle degeneracy on trigger probability estimate. The opaque areas shows the interval between the upper and lower 5th percentiles of the corresponding estimated trigger probability.

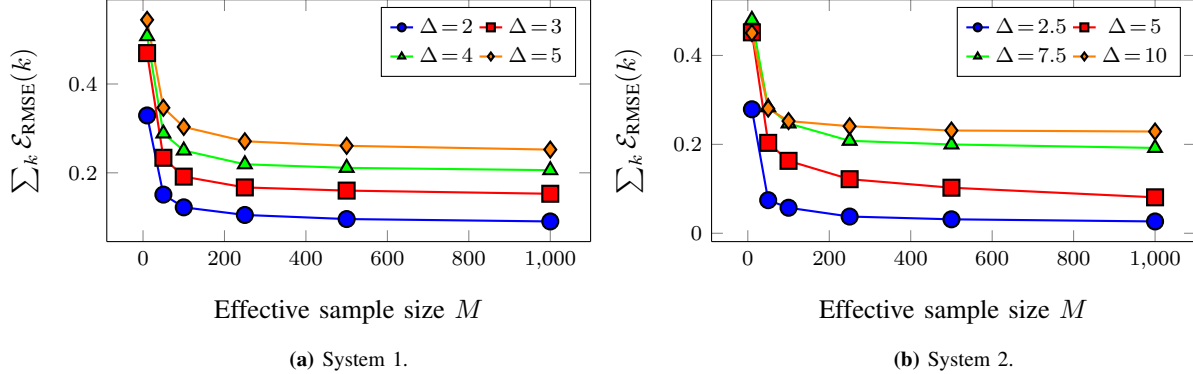


Fig. 6: Total RMSE between particle filter and Monte Carlo estimate of $p_T(n)$ over initial effective sample size and different values of Δ .

Monte Carlo method can be used to estimate it by simulating the target systems multiple times with the same $\{X_0^i\}_{i \in 1..N}$ and recording the first triggering instance in each iteration. Let $\mathcal{T}_{\{X_0^i\}}$ be the set of such triggering instances and I_{MC} the total number of Monte Carlo iterations. Then

$$\hat{p}_T(n|\{X_0^i\}) = \frac{\sum \mathbb{1}(k \in \mathcal{T}_{\{X_0^i\}})}{I_{MC}}$$

By drawing I_{X_0} different sets of $\{X_0^i\}_{i \in 1..N}$, we can compare the root mean squared error (RMSE) between the two

estimates for each n as

$$\mathcal{E}_{\text{RMSE}}(n) = \sqrt{\frac{1}{I_{X_0}} \sum_{S=1}^{I_{X_0}} (\hat{p}_{\text{PF}}(n|\{X_0^i\}_S) - \hat{p}_T(n|\{X_0^i\}_S))^2} \quad (12)$$

In Fig. 5 the estimated trigger probability for 40 time steps is displayed for both estimation methods, for both systems, and for a high and a low initial effective sample size. As can be seen, for a good posterior approximation the estimation quality is high, and both the mean and the bounds match very well for the two estimators. The same can not be said when the approximation is poor. However, one interesting

observation is that the effect of the poor posterior approximation seems to disappear for higher values of n . This is most likely due to the fact that the estimated trigger probability and the actual triggering instance are both dependent on the measurement predictions, and as time progresses and no triggering has occurred the effective sample size increases, resulting in a better estimate.

Further, in Fig. 6 the effect of the initial effective sample size on the total RMSE between the estimated trigger probability from the particle filter and the Monte Carlo method, as shown in (12), is displayed for multiple values of Δ . Here a clear trend can be seen for all values of Δ , as M decreases the uncertainty of the estimation increases. A larger Δ value further increases this error, but this could be attributed to the fact that the amount of non-near-zero trigger probabilities increases when Δ increases.

VI. CONCLUSION

In this paper, we have shown how the innovation-based trigger can be implemented using the particle filter for nonlinear systems, and how there exist certain types of simple nonlinear systems that are unsuitable for event-based sampling using the send-on-delta method. Although in this case the innovation-based trigger has no trouble tackling the example systems, one can think of more advanced scenarios where the IBT could perform badly, e.g., heavily skewed or multi-modal measurement densities, as discussed in [11]. We have further shown how it is possible to precompute the state estimates and trigger condition in an optimal sense by assuming that no event will happen over some time horizon, and accurately estimating the time horizon corresponding to a certain trigger probability. Further, we have investigated how the estimation is affected by the imposed particle degeneracy at event-triggering instances.

The precomputation scheme allows one to send trigger conditions at event instances, circumventing the need for a local filter or the need to continuously send the conditions to the sensor, at the expense of added computations at the observer. By setting the time horizon based on a certain cumulative trigger probability enables a well-chosen trade off between wasted computations and quality. Although the trigger probability estimation is accurate in the simulation study, it seems to be somewhat sensitive to the estimation quality at event instances which in turn are prone to particle degeneracy.

Finally, as mentioned in the paper the evaluation of the event-based likelihood (3) and the trigger probability (10) can be done in a closed form for most systems where $m = 1$. Otherwise, the affected integrals will for the general case be intractable. This greatly encumbers the practicality of our approach (or event-based particle filtering in general) for certain applications, as it introduces the need for numerical integration for each particle.

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 ELLIIT Excellence Center at Lund University.

REFERENCES

- [1] R. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME—Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.
- [2] 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.
- [3] E. I. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith, and R. Rednic, "Edge mining the Internet of Things," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3816–3825, Oct 2013.
- [4] 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.
- [5] A. Mohammadi and K. N. Plataniotis, "Event-based estimation with information-based triggering and adaptive update," *IEEE Transactions on Signal Processing*, vol. 65, no. 18, pp. 4924–4939, Sep. 2017.
- [6] M. Miskowicz, "Send-on-delta concept: An event-based data reporting strategy," *Sensors*, vol. 6, pp. 49–63, 2006.
- [7] S. Trimpe and R. D'Andrea, "Event-based state estimation with variance-based triggering," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3266–3281, Dec 2014.
- [8] S. Trimpe, "Event-based state estimation with switching static-gain observers," *IFAC Proceedings Volumes*, vol. 45, no. 26, pp. 91–96, 2012, 3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems.
- [9] J. Wu, Q. Jia, K. H. Johansson, and L. Shi, "Event-based sensor data scheduling: Trade-off between communication rate and estimation quality," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 1041–1046, April 2013.
- [10] J. W. Marck and J. Sijs, "Relevant sampling applied to event-based state-estimation," in *2010 Fourth International Conference on Sensor Technologies and Applications*, July 2010, pp. 618–624.
- [11] W. Li, Z. Wang, Q. Liu, and L. Guo, "An information aware event-triggered scheme for particle filter based remote state estimation," *Automatica*, vol. 103, pp. 151–158, May 2019.
- [12] M. A. Sid and S. Chitraganti, "Nonlinear event-based state estimation using particle filtering approach," in *8th International Conference on Modeling, Identification and Control*, Nov. 2016, pp. 874–879.
- [13] S. Davar and A. Mohammadi, "Event-based particle filtering with point and set-valued measurements," in *25th European Signal Processing Conference*, Aug. 2017, pp. 211–215.
- [14] S. Chitraganti and M. A. H. Darwish, "Nonlinear event-based state estimation using sequential Monte Carlo approach," in *56th IEEE Conference on Decision and Control*, Dec 2017, pp. 2170–2175.
- [15] J. Ruuskanen and A. Cervin, "Event-based state estimation using the auxiliary particle filter," in *18th European Control Conference*, June 2019, pp. 1854–1860.
- [16] J. Sijs and M. Lazar, "Event based state estimation with time synchronous updates," *IEEE Transaction on Automatic Control*, vol. 57, pp. 2650–2655, Oct. 2012.
- [17] 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.
- [18] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, July 2010.
- [19] 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, Apr. 1993, pp. 107–113.
- [20] S. Trimpe and M. C. Campi, "On the choice of the event trigger in event-based estimation," in *2015 International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, June 2015, pp. 1–8.
- [21] W. Chen, D. Shi, J. Wang, and L. Shi, "Event-triggered state estimation: Experimental performance assessment and comparative study," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 5, pp. 1865–1872, Sep. 2017.
- [22] S. Frühwirth-Schnatter, *Finite Mixture and Markov Switching Models*. Springer New York, 2006. [Online]. Available: <https://doi.org/10.1007/978-0-387-35768-3>