

On event based state estimation

Citation for published version (APA):

Sijs, J., & Lazar, M. (2009). On event based state estimation. In R. Majumdar, & P. Tabuada (Eds.), *Hybrid Systems : Computation and Control* (pp. 336-350). (Lecture Notes in Computer Science; Vol. 5469). Springer. https://doi.org/10.1007/978-3-642-00602-9_24

DOI:

[10.1007/978-3-642-00602-9_24](https://doi.org/10.1007/978-3-642-00602-9_24)

Document status and date:

Published: 01/01/2009

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

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.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

On Event Based State Estimation

Joris Sijs¹ and Mircea Lazar²

¹ TNO Science and Industry
2600 AD Delft, The Netherlands
joris.sijs@tno.nl

² Eindhoven University of Technology
5600 MB Eindhoven, The Netherlands
m.lazar@tue.nl

Abstract. To reduce the amount of data transfer in networked control systems and wireless sensor networks, measurements are usually taken only when an event occurs, rather than at each synchronous sampling instant. However, this complicates estimation and control problems considerably. The goal of this paper is to develop a state estimation algorithm that can successfully cope with event based measurements. Firstly, we propose a general methodology for defining event based sampling. Secondly, we develop a state estimator with a hybrid update, i.e. when an event occurs the estimated state is updated using measurements; otherwise the update makes use of the knowledge that the monitored variable is within a bounded set that defines the event. A sum of Gaussians approach is employed to obtain a computationally tractable algorithm.

1 Introduction

Different methods for state estimation have been introduced during the last decades. Each method is specialized in the type of process, the type of noise or the type of system architecture. In this paper we focus on the design of a state estimator that can efficiently cope with event based sampling. By event sampling we mean that measurements are generated only when an a priori defined event occurs in the data monitored by sensors. Such an estimator is very much needed in networked control systems and wireless sensor networks (WSNs) [1]. Especially in WSNs, where the limiting resource is energy, data transfer and processing power must be minimized. Existing estimators that could be used in this framework are discussed in Section 4. For related research on event based control, the interested reader is referred to the recent works [2, 3, 4, 5, 6].

The contribution of this paper is twofold. Firstly, using standard probability notions we set up a general mathematical description of event sampling depending on time and previous measurements. We assume that the estimator does not have information about when new measurements are available, which usually results in an unbounded error-covariance matrix. To prevent this from happening, we develop an estimation algorithm with hybrid update, which is the second main contribution. The developed event based estimator is updated both when an event occurs, with a received measurement sample, as well as at sampling instants synchronous in time, without receiving a measurement sample. In the latter case the update makes use of the knowledge that the monitored

variable, i.e. the measurement, is within a bounded set that defines the event. In order to meet low processing power specifications, the proposed state estimator is based on the Gaussian sum filter [7, 8], which is known to be computationally tractable.

2 Background Notions and Notation

\mathbb{R} defines the set of real numbers whereas the set \mathbb{R}_+ defines the non-negative real numbers. The set \mathbb{Z} defines the integer numbers and \mathbb{Z}_+ defines the set of non-negative integer numbers. The notation $\underline{0}$ is used to denote either the null-vector or the null-matrix. Its size will become clear from the context.

Suppose a vector $x(t) \in \mathbb{R}^n$ depends on time $t \in \mathbb{R}$ and is sampled using some sampling method. Two different sampling methods are discussed. The first one is time sampling in which samples are generated whenever time t equals some predefined value. This is either synchronous in time or asynchronous. In the synchronous case the time between two samples is constant and defined as $t_s \in \mathbb{R}_+$. If the time t at sampling instant $k_a \in \mathbb{Z}_+$ is defined as t_{k_a} , with $t_{0_a} := 0$, we define:

$$x_{k_a} := x(t_{k_a}) \quad \text{and} \quad x_{0_a:k_a} := (x(t_{0_a}), x(t_{1_a}), \dots, x(t_{k_a})).$$

The second sampling method is event sampling, in which samples are taken only when an event occurs. If t at event instant $k_e \in \mathbb{Z}_+$ is defined as t_{k_e} , with $t_{0_e} := 0$, we define:

$$x_{k_e} := x(t_{k_e}) \quad \text{and} \quad x_{0_e:k_e} := (x(t_{0_e}), x(t_{1_e}), \dots, x(t_{k_e})).$$

A transition-matrix $A_{t_2-t_1} \in \mathbb{R}^{a \times b}$ relates the vector $u(t_1) \in \mathbb{R}^b$ to a vector $x(t_2) \in \mathbb{R}^a$ as follows: $x(t_2) = A_{t_2-t_1} u(t_1)$.

The transpose, inverse and determinant of a matrix $A \in \mathbb{R}^{n \times n}$ are denoted as A^\top , A^{-1} and $|A|$ respectively. The i^{th} and maximum eigenvalue of a square matrix A are denoted as $\lambda_i(A)$ and $\lambda_{max}(A)$ respectively. Given that $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times n}$ are positive definite, denoted with $A \succ 0$ and $B \succ 0$, then $A \succ B$ denotes $A - B \succ 0$. $A \succeq 0$ denotes A is positive semi-definite.

The probability density function (PDF), as defined in [9] section B2, of the vector $x \in \mathbb{R}^n$ is denoted with $p(x)$ and the conditional PDF of x given $u \in \mathbb{R}^q$ is denoted as $p(x|u)$. The expectation and covariance of x are denoted as $E[x]$ and $cov(x)$ respectively. The conditional expectation of x given u is denoted as $E[x|u]$. The definitions of $E[x]$, $E[x|u]$ and $cov(x)$ can be found in [9] sections B4 and B7.

The Gaussian function (shortly noted as Gaussian) of vectors $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^n$ and matrix $P \in \mathbb{R}^{n \times n}$ is defined as $G(x, u, P) : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$, i.e.:

$$G(x, u, P) = \frac{1}{\sqrt{(2\pi)^n |P|}} e^{-0.5(x-u)^\top P^{-1}(x-u)}. \tag{1}$$

If $p(x) = G(x, u, P)$, then by definition it holds that $E[x] = u$ and $cov(x) = P$.

The element-wise Dirac-function of a vector $x \in \mathbb{R}^n$, denoted as $\delta(x) : \mathbb{R}^n \rightarrow \{0, 1\}$, satisfies:

$$\delta(x) = \begin{cases} 0 & \text{if } x \not\equiv \underline{0}, \\ 1 & \text{if } x \equiv \underline{0}, \end{cases} \quad \text{and} \quad \int_{-\infty}^{\infty} \delta(x) dx = 1. \tag{2}$$

For a vector $x \in \mathbb{R}^n$ and a bounded Borel set [10] $Y \subset \mathbb{R}^n$, the set PDF is defined as $\Lambda_Y(x) : \mathbb{R}^n \rightarrow \{0, v\}$ with $v \in \mathbb{R}$ defined as the Lebesque measure [11] of the set Y , i.e.:

$$\Lambda_Y(x) = \begin{cases} 0 & \text{if } x \notin Y, \\ v^{-1} & \text{if } x \in Y. \end{cases} \tag{3}$$

3 Event Sampling

Many different methods for sampling a vector $y(t) \in \mathbb{R}^q$ can be found in literature. The one mostly used is time sampling in which the k^{th} sampling instant is defined at time $t_{k_a} := t_{k_a-1} + \tau_{k_a-1}$ for some $\tau_{k_a-1} \in \mathbb{R}_+$. Recall that if $y(t)$ is sampled at t_a it is denoted as y_{k_a} . This method is formalized by defining the observation vector $z_{k_a-1} := (y_{k_a-1}^\top, t_{k_a-1})^\top \in \mathbb{R}^{q+1}$ at sampling instant k_a-1 . Let us define the set $H_{k_a}(z_{k_a-1}) \subset \mathbb{R}$ containing all the values that t can take between t_{k_a-1} and $t_{k_a-1} + \tau_{k_a-1}$, i.e.:

$$H_{k_a}(z_{k_a-1}) := \{t \in \mathbb{R} \mid t_{k_a-1} \leq t < t_{k_a-1} + \tau_{k_a-1}\}. \tag{4}$$

Then time sampling defines that the next sampling instant, i.e. k_a , takes place whenever present time t exceeds the set $H_{k_a}(z_{k_a-1})$. Therefore z_{k_a} is defined as:

$$z_{k_a} := (y_{k_a}^\top, t_{k_a})^\top \quad \text{if } t \notin H_{k_a}(z_{k_a-1}). \tag{5}$$

In the case of synchronous time sampling $\tau_{k_a} = t_s, \forall k_a \in \mathbb{Z}_+$, which is graphically depicted in Figure 1(a). Notice that with time sampling, the present time t specifies when samples of $y(t)$ are taken, but time t itself is independent of $y(t)$. As a result $y(t)$ in between the two samples can have any value within \mathbb{R}^q . Recently, asynchronous sampling methods have emerged, such as, for example ‘‘Send-on-Delta’’ [12, 13] and ‘‘Integral sampling’’ [14]. Opposed to time sampling, these sampling methods are not controlled by time t , but by $y(t)$ itself.

Next, we present a general definition of event based sampling. In this case a sampling instant is specified by an event of $y(t)$ instead of t . As such, one has to constantly check whether the measurement $y(t)$ satisfies certain conditions, which depend on time t and

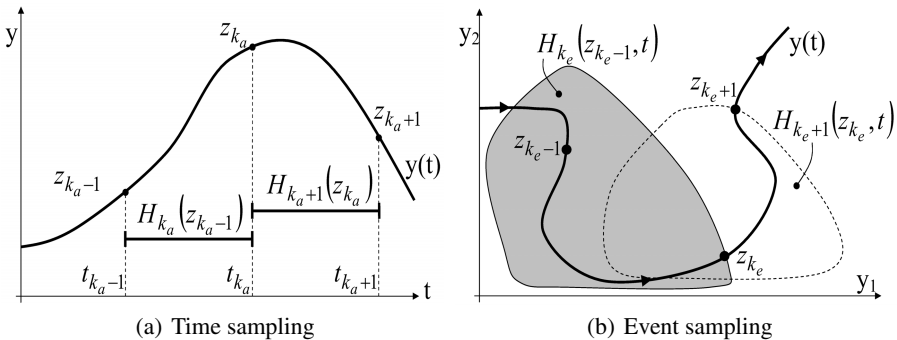


Fig. 1. The two different methods for sampling a signal $y(t)$

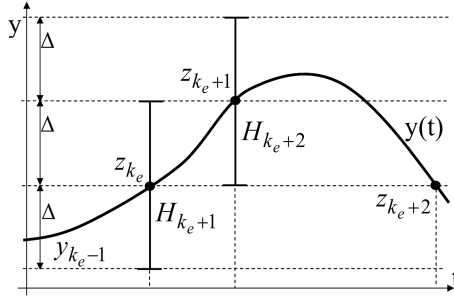


Fig. 2. Event sampling: Send-on-Delta

previous samples of the measurement. This method recovers the above mentioned asynchronous methods, for a particular choice of ingredients. Let us define the observation vector at sampling instant $k_e - 1$ as $z_{k_e-1} := (y_{k_e-1}^\top, t_{k_e-1})^\top \in \mathbb{R}^{q+1}$. With that we define the following bounded Borel set in *time-measurement-space*, i.e. $H_{k_e}(z_{k_e-1}, t) \subset \mathbb{R}^{q+1}$, which depends on both z_{k_e-1} and t . In line with time sampling the next event instant, i.e. k_e , takes place whenever $y(t)$ leaves the set $H_{k_e}(z_{k_e-1}, t)$ as shown in Figure 1(b) for $q = 2$. Therefore z_{k_e} is defined as:

$$z_{k_e} := (y_{k_e}^\top, t_{k_e})^\top \quad \text{if} \quad y(t) \notin H_{k_e}(z_{k_e-1}, t). \tag{6}$$

The exact description of the set $H_{k_e}(z_{k_e-1}, t)$ depends on the actual sampling method. As an example $H_{k_e}(z_{k_e-1}, t)$ is derived for the method ‘‘Send-on-Delta’’, with $y(t) \in \mathbb{R}$. In this case the event instant k_e occurs whenever $|y(t) - y_{k_e-1}|$ exceeds a predefined level Δ , see Figure 2, which results in $H_{k_e}(z_{k_e-1}, t) = \{y \in \mathbb{R} \mid -\Delta < y - y_{k_e-1} < \Delta\}$.

In event sampling, a well designed $H_{k_e}(z_{k_e-1}, t)$ should contain the set of all possible values that $y(t)$ can take in between the event instants $k_e - 1$ and k_e . Meaning that if $t_{k_e-1} \leq t < t_{k_e}$, then $y(t) \in H_{k_e}(z_{k_e-1}, t)$. A sufficient condition is that $y_{k_e-1} \in H_{k_e}(z_{k_e-1}, t)$, which for ‘‘Send-on-Delta’’ results in $y(t) \in [y_{k_e-1} - \Delta, y_{k_e-1} + \Delta]$ for all $t_{k_e-1} \leq t < t_{k_e}$.

Besides the event sampling methods discussed above, it is worth to also point out the related works [2, 4, 3], which focus on event based control systems rather than event based state estimators. Therein event sampling methods are proposed using additional information from the state of the system, which is assumed to be available.

4 Problem Formulation: State Estimation Based on Event Sampling

Assume a perturbed, dynamical system with state-vector $x(t) \in \mathbb{R}^n$, process-noise $w(t) \in \mathbb{R}^m$, measurement-vector $y(t) \in \mathbb{R}^q$ and measurement-noise $v(t) \in \mathbb{R}^q$. This process is

described by a state-space model with $A_\tau \in \mathbb{R}^{n \times n}$, $B_\tau \in \mathbb{R}^{n \times m}$ and $C \in \mathbb{R}^{q \times n}$. An event sampling method is used to sample $y(t)$. The model of this process becomes:

$$x(t + \tau) = A_\tau x(t) + B_\tau w(t), \tag{7a}$$

$$y(t) = Cx(t) + v(t), \tag{7b}$$

$$z_{k_e} = (y_{k_e}^\top, t_{k_e})^\top \quad \text{if } y(t) \notin H_{k_e}(z_{k_e-1}, t), \tag{7c}$$

$$\text{with } p(w(t)) := G(w(t), 0, Q) \quad \text{and} \quad p(v(t)) := G(v(t), 0, V). \tag{7d}$$

The state vector $x(t)$ of this system is to be estimated from the observation vectors $z_{0:k_e}$. Notice that the estimated states are usually required at all synchronous time samples k_a , with $t_s = t_{k_a} - t_{k_a-1}$, e.g., as input to a discrete monitoring system (or a discrete controller) that runs synchronously in time. For clarity system (7a) is considered autonomous, i.e. there is no control input. However, the estimation algorithm presented in this paper can be extended to controlled systems.

The goal is to construct an event-based state-estimator (EBSE) that provides an estimate of $x(t)$ not only at the event instants t_{k_e} , at which measurement data is received, but also at the sampling instants t_{k_a} , without receiving any measurement data. Therefore, we define a new set of sampling instants t_n as the combination of sampling instants due to event sampling, i.e. k_e , and time sampling, i.e. k_a :

$$\{t_{0:n-1}\} := \{t_{0:k_a-1}\} \cup \{t_{0:k_e-1}\} \quad \text{and} \quad t_n := \begin{cases} t_{k_a} & \text{if } t_{k_a} < t_{k_e}, \\ t_{k_e} & \text{if } t_{k_a} \geq t_{k_e}. \end{cases} \tag{8a}$$

$$\text{and } t_0 < t_1 < \dots < t_n, \quad x_n := x(t_n), \quad y_n := y(t_n). \tag{8b}$$

The estimator calculates the PDF of the state-vector x_n given all the observations until t_n . This results in a hybrid state-estimator, for at time t_n an event can either occur or not, which further implies that measurement data is received or not, respectively. In both cases the estimated state must be updated (not predicted) with all information until t_n . Therefore, depending on t_n a different PDF must be calculated, i.e.:

$$\text{if } t_n = t_{k_a} \Rightarrow p(x_n | z_{0:k_e-1}) \quad \text{with } t_{k_e-1} < t_{k_a} < t_{k_e}, \tag{9a}$$

$$\text{if } t_n = t_{k_e} \Rightarrow p(x_n | z_{0:k_e}). \tag{9b}$$

The performance of the state-estimator is related to the expectation and error-covariance matrix of its calculated PDF. Therefore, from (9) we define:

$$x_{n|n} := \begin{cases} E[x_n | z_{0:k_e-1}] & \text{if } t_n = t_{k_a} \\ E[x_n | z_{0:k_e}] & \text{if } t_n = t_{k_e} \end{cases} \quad \text{and} \quad P_{n|n} := \text{cov}(x_n - x_{n|n}). \tag{10}$$

The PDFs of (9) are described as the Gaussian $G(x_n, x_{n|n}, P_{n|n})$. Together with $x_{n|n}$, the square root of each eigenvalue of $P_{n|n}$, i.e. $\sqrt{\lambda_i(P_{n|n})}$ (or $\sqrt{\lambda(P_{n|n})}$ if there is only one eigenvalue), indicate the bound which surrounds 63% of the possible values for x_n . This is graphically depicted in Figure 3(a) for the 1D case and Figure 3 for the 2D case, in a top view.

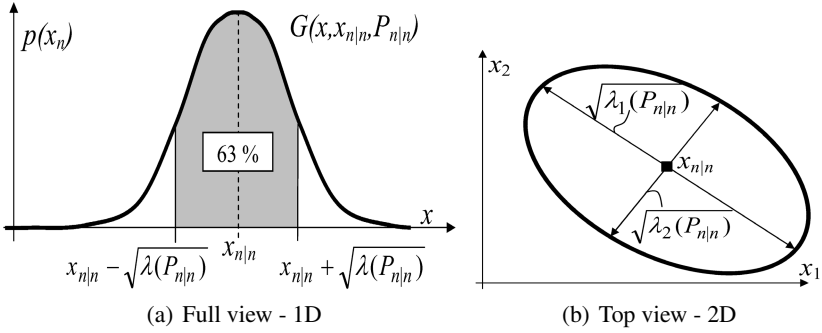


Fig. 3. Two examples of a Gaussian function

As such, the problem of interest in this paper is to construct a state-estimator suitable for the general event sampling method introduced in Section 3 and which is computationally tractable. Also, it is desirable that $P_{n|n}$ has bounded eigenvalues for all n .

Existing state estimators can be divided into two categories. The first one contains estimators based on time sampling: the (a)synchronous Kalman filter [15, 16] (linear process, Gaussian PDF), the Particle filter [17] and the Gaussian sum filter [7, 8] (non-linear process, non-Gaussian PDF). These estimators cannot be directly employed in event based sampling as if no new observation vector z_{k_e} is received, then $t_n - t_{k_e} \rightarrow \infty$ and $\lambda_i(P_{n|k_e-1}) \rightarrow \infty$. The second category contains estimators based on event sampling. In fact, to the best of our knowledge, only the method proposed in [18] fits this category. However, this EBSE is only applicable in the case of ‘‘Send-on-Delta’’ event sampling and it requires that any PDF is approximated as a single Gaussian function. Moreover, the asymptotic property of $P_{n|n}$ is not investigated in [18].

In the next section we propose a novel event-based state-estimator, suitable for any event sampling method based on the general set-up introduced in Section 3.

5 An Event-Based State Estimator

The EBSE estimates x_n given the received observation vectors until time t_n . Notice that due to the definition of event sampling we can extract information of all the measurement vectors $y_{0:n}$, i.e. also at the instants $t_n = t_{k_a}$, when the estimator does not receive y_{k_a} . For with $t_i \in \{t_{0:n}\}$ and $t_{j_e} \in \{t_{0_e:k_e}\}$ it follows that:

$$\begin{cases} y_i \in H_{j_e}(z_{j_e-1}, t_i) & \text{if } t_{j_e-1} \leq t_i < t_{j_e}, \\ y_i = y_{j_e} & \text{if } t_i = t_{j_e}. \end{cases} \tag{11}$$

Therefore, from the observation vectors $z_{0_e:k_e}$ and (11) the PDF of the hybrid state-estimation of (9), with the bounded, Borel set $Y_i \subset \mathbb{R}^q$, results in:

$$p(x_n | y_0 \in Y_0, y_1 \in Y_1, \dots, y_n \in Y_n) \quad \text{with} \tag{12a}$$

$$Y_i := \begin{cases} H_{j_e}(z_{j_e-1}, t_i) & \text{if } t_{j_e-1} < t_i < t_{j_e}, \\ \{y_{j_e}\} & \text{if } t_i = t_{j_e}. \end{cases} \tag{12b}$$

For brevity (12a) is denoted as $p(x_n|y_{0:n} \in Y_{0:n})$ and with Bayes-rule [19] yields:

$$p(x_n|y_{0:n} \in Y_{0:n}) := \frac{p(x_n|y_{0:n-1} \in Y_{0:n-1}) p(y_n \in Y_n|x_n)}{p(y_n \in Y_n|y_{0:n-1} \in Y_{0:n-1})}. \tag{13}$$

To have an EBSE with low processing demand, multivariate probability theory [20] is used to make (13) recursive:

$$p(a|b) := \int_{-\infty}^{\infty} p(a|c)p(c|b)dc \quad \Rightarrow \tag{14a}$$

$$p(x_n|y_{0:n-1} \in Y_{0:n-1}) = \int_{-\infty}^{\infty} p(x_n|x_{n-1})p(x_{n-1}|y_{0:n-1} \in Y_{0:n-1})dx_{n-1}, \tag{14b}$$

$$p(y_n \in Y_n|y_{0:n-1} \in Y_{0:n-1}) = \int_{-\infty}^{\infty} p(x_n|y_{0:n-1} \in Y_{0:n-1})p(y_n \in Y_n|x_n)dx_n. \tag{14c}$$

The calculation of $p(x_n|y_{0:n} \in Y_{0:n})$ is done in three steps:

1. Assimilate $p(y_n \in Y_n|x_n)$ for both $t_n = t_{k_e}$ and $t_n = t_{k_a}$;
2. Calculate $p(x_n|y_{0:n} \in Y_{0:n})$ as a summation of N Gaussians;
3. Approximate $p(x_n|y_{0:n} \in Y_{0:n})$ as a single Gaussian function.

The last step ensures that $p(x_n|y_{0:n} \in Y_{0:n})$ is described by a finite set of Gaussians, which is crucial for attaining computational tractability. Notice that (13) gives a unified description of the hybrid state-estimator.

5.1 Step 1: Measurement Assimilation

This section gives a unified formula of the PDF $p(y_n \in Y_n|x_n)$ valid for both $t_n = t_{k_e}$ and $t_n = t_{k_a}$. From multivariate probability theory [20] and (7b) we have:

$$p(y_n \in Y_n|x_n) := \int_{-\infty}^{\infty} p(y_n|x_n)p(y_n \in Y_n)dy_n \quad \text{and} \quad p(y_n|x_n) = G(y_n, Cx_n, V). \tag{15}$$

The PDF $p(y_n \in Y_n)$ is modeled as a uniform distribution for all $y_n \in Y_n$. Therefore, depending on the type of instant, i.e. event or not, we have:

$$p(y_n \in Y_n) := \begin{cases} \Lambda_{H_{k_e}}(y_n) & \text{if } t_{k_e-1} < t_n < t_{k_e}, \\ \delta(y_n - y_{k_e}) & \text{if } t_n = t_{k_e}. \end{cases} \tag{16}$$

Substitution of (16) into (15) gives that $p(y_n \in Y_n) = G(y_{k_e}, Cx_n, V)$ if $t_n = t_{k_e}$. However, if $t_n = t_{k_a}$ then $p(y_n \in Y_n|x_n)$ equals $\Lambda_{H_{k_e}}(y_n)$, which is not necessarily Gaussian. Moreover, it depends on the set H_{k_e} and therefore on the actual event sampling method that is employed. In order to have a unified expression of $p(y_n \in Y_n|x_n)$ for both types of t_n , independent of the event sampling method, $\Lambda_{H_{k_e}}(y_n)$ can be approximated as a summation of N Gaussians, i.e.

$$\Lambda_{H_{k_e}}(y_n) \approx \sum_{i=1}^N \alpha_n^i G(y_n, y_n^i, V_n^i) \quad \text{and} \quad \sum_{i=1}^N \alpha_n^i := 1. \tag{17}$$

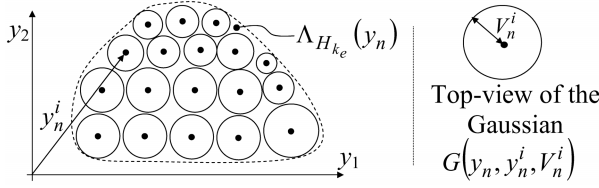


Fig. 4. Approximation of $\Lambda_{H_{k_e}}(y_n)$ as a sum of Gaussian functions

This is graphically depicted in Figure 4 for $y_n \in \mathbb{R}^2$. The interested reader is referred to [7] for more details.

Substituting (17) into (16) yields the following $p(y_n \in Y_n | x_n)$ if $t_n = t_{k_a}$:

$$p(y_n \in Y_n | x_n) \approx \sum_{i=1}^N \alpha_n^i \int_{-\infty}^{\infty} G(y_n, Cx_n, V) G(y_n, y_n^i, V_n^i) dy_n. \tag{18}$$

Proposition 1. [15, 17] *Let there exist two Gaussians of random vectors $x \in \mathbb{R}^n$ and $m \in \mathbb{R}^q$, with $\Gamma \in \mathbb{R}^{q \times n}$: $G(m, \Gamma x, M)$ and $G(x, u, U)$. Then they satisfy:*

$$\int_{-\infty}^{\infty} G(x, u, U) G(m, \Gamma x, M) dx = G(\Gamma u, m, \Gamma U \Gamma^T + M), \tag{19}$$

$$G(x, u, U) G(m, \Gamma x, M) = G(x, d, D) G(m, \Gamma u, \Gamma U \Gamma^T + M), \tag{20}$$

with $D := (U^{-1} + \Gamma^T M^{-1} \Gamma)^{-1}$ and $d := D U^{-1} u + D \Gamma^T M^{-1} m$.

Applying Proposition 1 ((19) to be precise) and $G(x, y, Z) = G(y, x, Z)$ on (18) yields:

$$p(y_n \in Y_n | x_n) \approx \sum_{i=1}^N \alpha_n^i G(y_n^i, Cx_n, V + V_n^i), \quad \text{if } t_n = t_{k_a}. \tag{21}$$

In conclusion we can state that the unified expression of the PDF $p(y_n \in Y_n | x_n)$, at both $t_n = t_{k_e}$ and $t_n = t_{k_a}$, for any event sampling method results in:

$$p(y_n \in Y_n | x_n) \approx \sum_{i=1}^N \alpha_n^i G(y_n^i, Cx_n, R_n^i) \quad \text{with } R_n^i := V + V_n^i. \tag{22}$$

If $t_n = t_{k_e}$ the variables of (22) are: $N = 1$, $\alpha_n^1 = 1$, $y_n^1 = y_{k_e}$ and $V_n^1 = \underline{0}$. If $t_n = t_{k_a}$ the variables depend on $\Lambda_{H_{k_e}}(y_n)$ and its approximation. As an example these variables are calculated for the method ‘‘Send-on-Delta’’ with $y \in \mathbb{R}$.

Example 1. In ‘‘Send-on-Delta’’, for certain N , the approximation of $\Lambda_{H_{k_e}}(y_n)$, as presented in (17), is obtained with $i \in \{1, 2, \dots, N\}$ and:

$$y_n^i = y_{k_e-1} - \left(\frac{N - 2(i - 1) - 1}{2N} \right) 2\Delta, \tag{23}$$

$$\alpha_n^i = \frac{1}{N}, \quad V_n^i = \left(\frac{2\Delta}{N} \right)^2 \left(0.25 - 0.05e^{-\frac{4(N-1)}{15}} - 0.08e^{-\frac{4(N-1)}{180}} \right), \quad \forall i.$$

With the result of (22), $p(x_n | y_{0:n} \in Y_{0:n})$ can also be expressed as a sum of N Gaussians.

5.2 Step 2: State Estimation

First the PDF $p(x_n|y_{0:n-1} \in Y_{0:n-1})$ of (14b) is calculated. From the EBSE we have $p(x_{n-1}|y_{0:n-1} \in Y_{0:n-1}) := G(x_{n-1}, x_{n-1}|_{n-1}, P_{n-1, n-1})$ and from (7a) with $\tau_n := t_n - t_{n-1}$ we have $p(x_n|x_{n-1}) := G(x_n, A\tau_n x_{n-1}, B\tau_n QB\tau_n^\top)$. Therefore using (19) in (14b) yields:

$$\begin{aligned} p(x_n|y_{0:n-1} \in Y_{0:n-1}) &= G(x_n, x_n|_{n-1}, P_{n, n-1}) \quad \text{with} \\ x_n|_{n-1} &:= A\tau_n x_{n-1}|_{n-1} \quad \text{and} \quad P_{n|n-1} := A\tau_n P_{n-1}|_{n-1} A\tau_n^\top + B\tau_n QB\tau_n^\top. \end{aligned} \quad (24)$$

Next $p(x_n|y_{0:n} \in Y_{0:n})$, defined in (13), is calculated after multiplying (22) and (24):

$$p(x_n|y_{n-1} \in Y_{0:n-1})p(y_n \in Y_n|x_n) \approx \sum_{i=1}^N \alpha_n^i G(x_n, x_n|_{n-1}, P_{n|n-1}) G(y_n^i, Cx_n, R_n^i). \quad (25)$$

Equation (25) is explicitly solved by applying Proposition 1:

$$p(x_n|y_{0:n-1} \in Y_{0:n-1})p(y_n \in Y_n|x_n) \approx \sum_{i=1}^N \alpha_n^i \beta_n^i G(x_n, x_n^i, P_n^i) \quad \text{with} \quad (26a)$$

$$x_n^i := P_n^i \left(P_{n|n-1}^{-1} x_n|_{n-1} + C^\top (R_n^i)^{-1} y_n^i \right), \quad P_n^i := \left(P_{n|n-1}^{-1} + C^\top (R_n^i)^{-1} C \right)^{-1} \quad (26b)$$

$$\text{and} \quad \beta_n^i := G(y_n^i, Cx_n|_{n-1}, CP_{n|n-1}C^\top + R_n^i).$$

The expression of $p(x_n|y_{0:n} \in Y_{0:n})$ as a sum of N Gaussians is the result of the following substitutions: (26) into (13), (26) into (14c) to obtain $p(y_n \in Y_n | y_{0:n-1} \in Y_{0:n-1})$ and the latter into (13) again. This yields

$$p(x_n|y_{0:n} \in Y_{0:n}) \approx \sum_{i=1}^N \frac{\alpha_n^i \beta_n^i}{\sum_{i=1}^N \alpha_n^i \beta_n^i} G(x_n, x_n^i, P_n^i). \quad (27)$$

The third step is to approximate (27) as a single Gaussian, as this facilitates a computationally tractable algorithm. For if $p(x_{n-1}|y_{0:n-1} \in Y_{0:n-1})$ is described using M_{n-1} Gaussians and $p(y_n \in Y_n|x_n)$ is described using N Gaussians, the estimate of x_n in (27) is described with $M_n = M_{n-1}N$ Gaussians. Meaning that M_n increases after each sample instant and with it also the processing demand of the EBSE increases.

5.3 Step 3: State Approximation

$p(x_n|y_{0:n} \in Y_{0:n})$ of (27) is approximated as a single Gaussian with an equal expectation and covariance matrix, i.e.:

$$p(x_n|y_{0:n} \in Y_{0:n}) \approx G(x_n, x_n|_n, P_n|_n) \quad \text{with} \quad (28a)$$

$$x_n|_n := \sum_{i=1}^N \frac{\alpha_n^i \beta_n^i x_n^i}{\sum_{i=1}^N \alpha_n^i \beta_n^i}, \quad P_n|_n := \sum_{i=1}^N \frac{\alpha_n^i \beta_n^i}{\sum_{i=1}^N \alpha_n^i \beta_n^i} \left(P_n^i + (x_n|_n - x_n^i)(x_n|_n - x_n^i)^\top \right). \quad (28b)$$

The expectation and covariance of (27), equal to $x_n|_n$ and $P_n|_n$ of (28), can be derived from the corresponding definitions. Notice that because the designed EBSE is based on the equations of the Kalman filter, the condition of computational tractability is met.

5.4 On Asymptotic Analysis of the Error-Covariance Matrix

In this section we present some preliminary results on the asymptotic analysis of the error-covariance matrix of the developed EBSE, i.e. $\lim_{n \rightarrow \infty} P_{n|n}$ which for convenience is denoted as P_∞ . The main result of this section is obtained under the standing assumption that $\Lambda_{H_{k_e}}(y_n)$ is approximated using a single Gaussian. Note that the result then also applies to the estimator presented in [18], as a particular case. Recall that H_{k_e} is assumed to be a bounded set. Therefore, it is reasonable to further assume that $\Lambda_{H_{k_e}}(y_n)$ can be approximated using the formula (17), for $N = 1$, and that there exists a constant matrix R such that $V + V_n^1 \preceq R$ for all n .

Note that if the classical Kalman filter (KF) [15] is used to perform a state-update only at the synchronous time instant $t_n = t_{k_a}$ (with a measurement covariance matrix equal to R), then such an analysis is already available. In [21, 22] it is proven that if the eigenvalues of A_{t_s} are within the unit circle and (A_{t_s}, C) is observable, then the error-covariance matrix of the synchronous KF, denoted with $P^{(s)}$, converges to P_K , with P_K defined as the solution of:

$$P_K = \left(\left(A_{t_s} P_K A_{t_s}^\top + B_{t_s} Q B_{t_s}^\top \right)^{-1} + C^\top R^{-1} C \right)^{-1}. \quad (29)$$

In case that the classical asynchronous Kalman filter (AKF) [16] is used, then the estimation would occur only at the instants that a measurement is received, i.e. $t_n = t_{k_e}$. As it is not known when a new measurement is available, the time between two samples keeps on growing, as well as the eigenvalues of the AKF's error-covariance matrix, denoted with $\lambda_i(P^{(a)})$. Moreover, in [23] (see also [24]) it is proven that $P^{(a)}$ will diverge if no new measurements are received.

To circumvent this problem, instead of a standard AKF, we consider an artificial AKF (denoted by CKF for brevity) obtained as the combination of a synchronous KF and a standard AKF. By this we mean that the CKF performs a state-update at all time instants t_n with a measurement covariance matrix equal to R . Therefore its error-covariance matrix, denoted with $P_{n|n}^{(c)}$, is updated according to:

$$P_{n|n}^{(c)} := \left(\left(A_{t_n} P_{n-1|n-1}^{(c)} A_{t_n}^\top + B_{t_n} Q B_{t_n}^\top \right)^{-1} + C^\top R^{-1} C \right)^{-1}. \quad (30)$$

Notice that because the CKF is updated at more time instants than the KF, it makes sense that its error-covariance matrix is "smaller" than the one of the KF, i.e. $P^{(c)} \preceq P^{(s)}$ holds at the synchronous time instants $t_n = t_{k_a}$. However, this does not state anything about $P^{(c)}$ at the event instants. As also at these sample instants the CKF performs an update rather than just a prediction, the following assumption is needed. Let $P_\infty^{(c)}$ denote $\lim_{n \rightarrow \infty} P_{n|n}^{(c)}$.

Assumption 1. *There exists $\Delta_\lambda \in \mathbb{R}_+$ such that $\lambda_{\max}(P_\infty^{(c)}) < \lambda_{\max}(P_K) + \Delta_\lambda$.*

Next we will employ Assumption 1 to obtain an upper bound on the error-covariance matrix of the developed EBSE. The following technical Lemma will be of use.

Lemma 1. *Let any square matrices $V_1 \succeq V_2$ and $W_1 \succeq W_2$ with $V_1 \succ 0$ and $W_1 \succ 0$ be given. Suppose that the matrices U_1 and U_2 are defined as $U_1 := (V_1^{-1} + C^\top W_1^{-1} C)^{-1}$ and $U_2 := (V_2^{-1} + C^\top W_2^{-1} C)^{-1}$, for any C of suitable size. Then it holds that $U_1 \preceq U_2$.*

Proof. As shown in [25], it holds that $V_1^{-1} \succeq V_2^{-1}$ and $C^\top W_1^{-1} C \succeq C^\top W_2^{-1} C$. Hence, it follows that $V_1^{-1} + C^\top W_1^{-1} C \succeq V_2^{-1} + C^\top W_2^{-1} C$, which yields $U_1^{-1} \succeq U_2^{-1}$. Thus, $U_1 \preceq U_2$, which concludes the proof. \square

Theorem 1. *Suppose that the EBSE, as presented in Section 5, approximates $\Lambda_{H_{k_e}}(y_n)$ according to (17) with $N = 1$. Then $\lambda_{\max}(P_\infty) \leq \lambda_{\max}(P_\infty^{(e)})$.*

The proof of the above theorem, which makes use of Lemma 1, is given in the Appendix. Obviously, under Assumption 1 the above result further implies that the error-covariance matrix of the developed EBSE is bounded. Under certain reasonable assumptions, including the standard ones (i.e. the eigenvalues of the A_{t_s} -matrix are within the unit-circle and (A_{t_s}, C) is an observable pair), it is possible to derive an explicit expression of Δ_λ , which validates Assumption 1. However, this is beyond the scope of this manuscript.

6 Illustrative Example

In this section we illustrate the effectiveness of the developed EBSE in terms of state-estimation error, sampling efficiency and computational tractability. The case study is a 1D object-tracking system. The states $x(t)$ of the object are position and speed while the measurement vector $y(t)$ is position. The process-noise $w(t)$ represents the object’s acceleration. Then given a maximum acceleration of $0.5[m/s^2]$ its corresponding Q , according to [26], equals 0.02. Therefore the model as presented in (7) yields $A = \begin{pmatrix} 1 & \tau \\ 0 & 1 \end{pmatrix}$, $B = \begin{pmatrix} \frac{\tau^2}{2} & \tau \end{pmatrix}^\top$, $C = (1 \ 0)$ and $D = 0$, which is in fact a discrete-time double integrator. The acceleration, i.e. process noise $w(t)$, in time is shown in Figure 5 together with the object’s position and speed, i.e. the elements of the real state-vector $x(t)$. The sampling time is $t_s = 0.1$ and the measurement-noise covariance is $V = 0.1 \cdot 10^{-3}$.

Three different estimators are tested. The first two estimators are the EBSE and the asynchronous Kalman filter (AKF) of [16]. For simplicity, in both estimators we used the “Send-on-Delta” method with $\Delta = 0.1[m]$. For the EBSE we approximated $\Lambda_{H_{k_e}}(y_n)$ using (23) with $N = 5$. The AKF estimates the states only at the event instants t_{k_e} . The states at t_{k_e} are calculated by applying the prediction-step of (14b). The third estimator

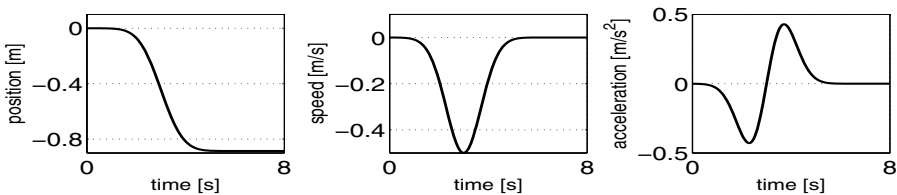


Fig. 5. The position, speed and acceleration of the object

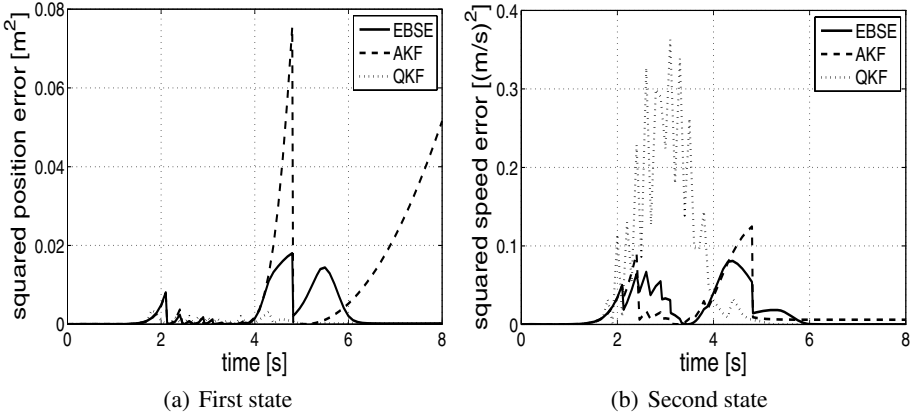


Fig. 6. The squared estimation error of the two states

is based on the quantized Kalman filter (QKF) introduced in [26] that uses synchronous time sampling of y_{k_a} . The QKF can deal with quantized data, which also results in less data transfer, and therefore can be considered as an alternative to EBSE. In the QKF \bar{y}_{k_a} is the quantized version of y_{k_a} with quantization level 0.1, which corresponds to the “Send-on-Delta” method. Hence, a comparison can be made.

In Figure 6(a) and Figure 6(b) the squared state estimation-error of the three estimators is plotted. They show that the QKF estimates the position of the object with the least error. However, its error in speed is worse compared to the EBSE. Further, the plot of the AKF clearly shows that prediction of the state gives a significant growth in estimation-error when the time between the event sampling-instants increases ($t > 4$).

Beside estimation error, sampling efficiency η is also important due to the increased interest in WSNs. For these systems communication is expensive and one aims to have the least data transfer. We define $\eta \in \mathbb{R}_+$ as

$$\eta := \frac{(x_i - x_{i|i})^\top (x_i - x_{i|i})}{(x_i - x_{i|i-1})^\top (x_i - x_{i|i-1})},$$

which is a measure of the change in the estimation-error after the measurement update with either z_{k_e} or \bar{y}_{k_a} was done. Notice that if $\eta < 1$ the estimation error decreased after an update, if $\eta > 1$ the error increased and if $\eta = 1$ the error remained the same. For the EBSE $i = k_e$ with $i - 1$ equal to $k_e - 1$ or $k_a - 1$. For the AKF $i = k_e$ with $i - 1 = k_e - 1$. For the QKF $i = k_a$ and $i - 1 = k_a - 1$. Figure 7 shows that for the EBSE $\eta < 1$ at all time instants. The AKF has one instant, $t = 3.4$, at which $\eta > 1$. In case of the QKF the error sometimes decreases but it can also increase considerably after an update. Also notice that η of the QKF converges to 1. Meaning that for $t > 5.6$ the estimation error does not change after an update and new samples are mostly used to bound $\lambda_i(P_{k_a|k_a})$. The EBSE has the same property, although for this method the last sample was received at $t = 4.9$.

The last comparison criterion is the total amount of processing time that was required by each of the three estimators. From the equations of the EBSE one can see that for

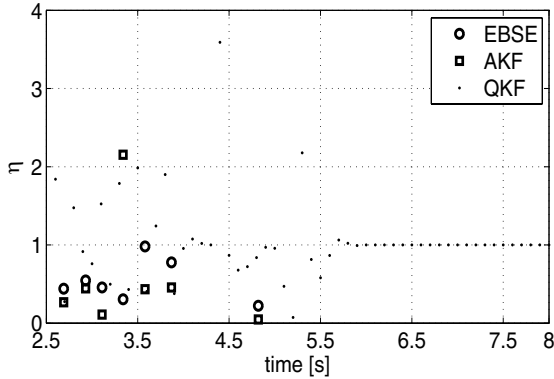


Fig. 7. The factor of increase in estimation error after z_{k_e} or \bar{y}_{k_a}

every Gaussian (recall that there are N Gaussians employed to obtain an approximation of $\Lambda_{H_{k_e}}(y_n)$) a state-update is calculated similar to a synchronous Kalman filter. Therefore, a rule of thumb is that the EBSE will require N times the amount of processing time of the Kalman filter [15]. Because the QKF is in fact such a Kalman filter, with a special measurement-estimation, the EBSE of this application example will roughly cost about 5 times more processing time then the QKF. After running all three algorithms in Matlab on an Intel®Pentium®processor of 1.86 GHz with 504 MB of RAM we have obtained the following performances. The AKF estimated x_{k_e} and predicted x_{k_a} in a total time of 0.016 seconds while the QKF estimated x_{k_a} and its total processing time equaled 0.022 seconds. For the EBSE, both x_{k_e} and x_{k_a} were estimated and it took 0.094 seconds, which is less than $0.11 = 5 \times 0.022$. This means that although the EBSE results in the most processing time, it is still computationally comparable to the AKF and QKF. On the overall, it can be concluded that the EBSE provides an estimation-error similar to the one attained by the QKF, but with significantly less data transmission. The application case study also indicate that the number of Gaussians becomes a tuning factor that can be used to achieve a desired tradeoff between numerical complexity (which further translates into energy consumption) and estimation error. As such, the proposed EBSE it is most suited for usage in networks in general and WSNs in particular.

7 Conclusions

In this paper a general event-based state-estimator was presented. The distinguishing feature of the proposed EBSE is that *estimation* of the states is performed at two different type of time instants, i.e. at event instants, when measurement data is used for update, and at synchronous time sampling, when no measurement is received, but an update is performed based on the knowledge that the monitored variable lies within a set used to define the event. As a result, under certain assumptions, it was established that the error-covariance matrix of the EBSE is bounded, even in the situation when no

new measurement is received anymore. Its effectiveness for usage in WSNs has been demonstrated on an application example.

As a final remark we want to indicate that future work, besides a more general proof of asymptotic stability, is focused on determining specific types of WSNs applications where the developed EBSE would be most suitable.

Acknowledgements. Research partially supported by the Veni grant “Flexible Lyapunov Functions for Real-time Control”, grant number 10230, awarded by STW (Dutch Science Foundation) and NWO (The Netherlands Organization for Scientific Research).

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless Sensor Networks: a survey. *Computer Networks* 38, 393–422 (2002)
2. Tabuada, P.: Event-Triggered Real-Time Scheduling for Stabilizing Control Tasks. *IEEE Transactions on Automatic Control* 52, 1680–1685 (2007)
3. Velasco, M., Marti, P., Lozoya, C.: On the Timing of Discrete Events in Event-Driven Control Systems. In: Egerstedt, M., Mishra, B. (eds.) *HSCC 2008*. LNCS, vol. 4981, pp. 670–673. Springer, Heidelberg (2008)
4. Wang, X., Lemmon, M.: State based self-triggered feedback control systems with \mathcal{L}_2 stability. In: 17th IFAC World Congress, Seoul, South Korea (2008) (accepted in *IEEE Transactions on Automatic Control*)
5. Heemels, W.P.M.H., Sandee, J.H., van den Bosch, P.P.J.: Analysis of event-driven controllers for linear systems. *International Journal of Control* 81(4) (2008)
6. Henningsson, T., Johannesson, E., Cervin, A.: Sporadic event-based control of first-order linear stochastic systems. *Automatica* 44(11), 2890–2895 (2008)
7. Sorenson, H.W., Alspach, D.L.: Recursive Bayesian estimation using Gaussian sums. *Automatica* 7, 465–479 (1971)
8. Kotecha, J.H., Djurić, P.M.: Gaussian sum particle filtering. *IEEE Transaction Signal Processing* 51(10), 2602–2612 (2003)
9. Johnson, N.L., Kotz, S., Kemp, A.W.: *Univariate discrete distributions*. John Wiley and Sons, Chichester (1992)
10. Aggoun, L., Elliot, R.: *Measure Theory and Filtering*. Cambridge University Press, Cambridge (2004)
11. Lebesgue, H.L.: *Integrale, longueur, aire*. PhD thesis, University of Nancy (1902)
12. Åström, K.J., Bernhardsson, B.M.: Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In: 41st IEEE Conf. on Dec. and Contr., Las Vegas, USA (2002)
13. Miskowicz, M.: Send-on-delta concept: an event-based data-reporting strategy. *Sensors* 6, 49–63 (2006)
14. Miskowicz, M.: Asymptotic Effectiveness of the Event-Based Sampling according to the Integral Criterion. *Sensors* 7, 16–37 (2007)
15. Kalman, R.E.: A new approach to linear filtering and prediction problems. *Transaction of the ASME Journal of Basic Engineering* 82(D), 35–42 (1960)
16. Mallick, M., Coraluppi, S., Carthel, C.: *Advances in Asynchronous and Decentralized Estimation*. In: *Proceeding of the 2001 Aerospace Conference*, Big Sky, MT, USA (2001)
17. Ristic, B., Arulampalam, S., Gordon, N.: Beyond the Kalman filter: Particle filter for tracking applications. Artech House, Boston (2004)
18. Nguyen, V.H., Suh, Y.S.: Improving estimation performance in Networked Control Systems applying the Send-on-delta transmission method. *Sensors* 7, 2128–2138 (2007)

19. Mardia, K.V., Kent, J.T., Bibby, J.M.: *Multivariate analysis*. Academic Press, London (1979)
20. Montgomery, D.C., Runger, G.C.: *Applied Statistics and Probability for Engineers*. John Wiley and Sons, Chichester (2007)
21. Hautus, M.L.J.: Controllability and observability conditions of linear autonomous systems. In: *Indagationes Mathematicae*, vol. 32, pp. 448–455 (1972)
22. Balakrishnan, A.V.: *Kalman Filtering Theory*. Optimization Software, Inc., New York (1987)
23. Sinopoli, B., Schenato, L., Franceschetti, M., Poolla, K., Jordan, M., Sastry, S.: Kalman Filter with Intermittent Observations. *IEEE Transactions on Automatic Control* 49, 1453–1464 (2004)
24. Mo, Y., Sinopoli, B.: A characterization of the critical value for Kalman filtering with intermittent observations. In: *47th IEEE Conference on Decision and Control*, Cancun, Mexico, pp. 2692–2697 (2008)
25. Bernstein, D.S.: *Matrix Mathematics*. Princeton University Press, Princeton (2005)
26. Curry, R.E.: *Estimation and control with quantized measurements*. MIT Press, Boston (1970)

A Proof of Theorem 1

Under the hypothesis, for the proposed EBSE, $P_{n|n}$ of (28), with $\tau_n := t_n - t_{n-1}$ and $R_n := V + V_n^1$, becomes:

$$P_{n|n} = \left(\left(A_{\tau_n} P_{n-1|n-1} A_{\tau_n}^\top + B_{\tau_n} Q B_{\tau_n}^\top \right)^{-1} + C^\top R_n^{-1} C \right)^{-1}. \quad (31)$$

The upper bound on $\lambda_{\max}(P_\infty)$ is proven by induction, considering the asymptotic behavior of a CKF that runs in parallel with the EBSE, as follows. The EBSE calculates $P_{n|n}$ as (31) and the CKF calculates $P_{n|n}^{(c)}$ as (30). Note that this implies that $R_n \preceq R$ for all n . Let the EBSE and the CKF start with the same initial covariance matrix P_0 .

The first step of induction is to prove that $P_{1|1} \preceq P_{1|1}^{(c)}$. From (31) and (30) we have that

$$P_{1|1} = \left(\left(A_{\tau_1} P_0 A_{\tau_1}^\top + B_{\tau_1} Q B_{\tau_1}^\top \right)^{-1} + C^\top R_1^{-1} C \right)^{-1},$$

$$P_{1|1}^{(c)} = \left(\left(A_{\tau_1} P_0 A_{\tau_1}^\top + B_{\tau_1} Q B_{\tau_1}^\top \right)^{-1} + C^\top R^{-1} C \right)^{-1}.$$

Suppose we define $V_1 := A_{\tau_1} P_0 A_{\tau_1}^\top + B_{\tau_1} Q B_{\tau_1}^\top$, $V_2 := A_{\tau_1} P_0 A_{\tau_1}^\top + B_{\tau_1} Q B_{\tau_1}^\top$, $W_1 := R_1$ and $W_2 := R$, then $W_1 \preceq W_2$ and $V_1 = V_2$. Therefore applying Lemma 1, with $U_1 := P_{1|1}$ and $U_2 := P_{1|1}^{(c)}$, yields $P_{1|1} \preceq P_{1|1}^{(c)}$.

The second and last step of induction is to show that if $P_{n-1|n-1} \preceq P_{n-1|n-1}^{(c)}$, then $P_{n|n} \preceq P_{n|n}^{(c)}$. Let $V_1 := A_{\tau_n} P_{n-1|n-1} A_{\tau_n}^\top + B_{\tau_n} Q B_{\tau_n}^\top$, $V_2 := A_{\tau_n} P_{n-1|n-1}^{(c)} A_{\tau_n}^\top + B_{\tau_n} Q B_{\tau_n}^\top$, $W_1 := R_n$ and $W_2 := R$. Notice that this gives $W_1 \preceq W_2$ and starting from $P_{n-1|n-1} \preceq P_{n-1|n-1}^{(c)}$ it follows that $V_1 \preceq V_2$ (see, e.g. [25]). Hence, applying Lemma 1, with $U_1 := P_{n|n}$ and $U_2 := P_{n|n}^{(c)}$ yields $P_{n|n} \preceq P_{n|n}^{(c)}$. This proves that $P_\infty \preceq P_\infty^{(c)}$, which yields (see e.g., [25]) $\lambda_{\max}(P_\infty) \preceq \lambda_{\max}(P_\infty^{(c)})$. \square