

Article

On-Board Event-Based State Estimation for Trajectory Approaching and Tracking of a Vehicle

Miguel Martínez-Rey *, Felipe Espinosa, Alfredo Gardel and Carlos Santos

Department of Electronics, University of Alcalá. Polytechnic School, Campus Universitario, Alcalá de Henares 28871, Spain; E-Mails: felipe.espinosa@uah.es (F.E.); alfredo@depeca.uah.es (A.G.); carlos.santos@depeca.uah.es (C.S.)

* Author to whom correspondence should be addressed; E-Mail: miguel.martinez@depeca.uah.es; Tel.: +34-918-856-906.

Academic Editor: Albert M. K. Cheng

Received: 1 April 2015 / Accepted: 17 June 2015 / Published: 19 June 2015

Abstract: For the problem of pose estimation of an autonomous vehicle using networked external sensors, the processing capacity and battery consumption of these sensors, as well as the communication channel load should be optimized. Here, we report an event-based state estimator (EBSE) consisting of an unscented Kalman filter that uses a triggering mechanism based on the estimation error covariance matrix to request measurements from the external sensors. This EBSE generates the events of the estimator module on-board the vehicle and, thus, allows the sensors to remain in stand-by mode until an event is generated. The proposed algorithm requests a measurement every time the estimation distance root mean squared error (DRMS) value, obtained from the estimator's covariance matrix, exceeds a threshold value. This triggering threshold can be adapted to the vehicle's working conditions rendering the estimator even more efficient. An example of the use of the proposed EBSE is given, where the autonomous vehicle must approach and follow a reference trajectory. By making the threshold a function of the distance to the reference location, the estimator can halve the use of the sensors with a negligible deterioration in the performance of the approaching maneuver.

Keywords: event-based state estimation; indoor localization; non-linear filtering, trajectory tracking

1. Introduction

Recent years have witnessed a growing interest in the use of event-based communications for cyber-physical systems. One example is autonomous vehicle guidance [1] in intelligent spaces [2]. In these environments, sensors often have to cope with scarce resources, such as communication bandwidth and processing capacity. In addition, the sensors are often battery-powered, so it is necessary to optimize all sensor functions in order to extend battery life.

Several strategies have recently been developed to reduce the number of measurements used by the estimator and transmitted through the network channel. Most of them rely on the send-on-delta (SoD) method, also known as Lebesgue sampling [3,4]. According to this method, a sample is sent to the estimator whenever the measured value exceeds certain limits with respect to the previous sample sent.

Many variants of the same principle have been proposed. In [5], the authors study and compare some SoD extensions that include integrating the difference between the current sensor value and the last sample transmitted (send-on-area) or integrating this difference squared (send-on-energy). In [6], a predictor for the expected next sample is used based on the previous samples. In addition, in [7], a delta variable is set in the presence of disturbances.

In all of the above-mentioned triggering schemes, the sensors get to decide when a sample should be sent to the estimator. This implies that the sensor must be continuously running and monitoring the measured variable in order to detect the event. Meanwhile, the estimator at the other side of the communication channel waits for the measurements and uses them when they arrive. Thus, this kind of estimator is called an event-based state estimator (EBSE).

These are distributed estimator systems, because the event is generated by the sensor module rather than by the estimator. Along the same lines, some authors have explored the implementation of distributed Kalman filters, whereby each sensor node runs its own filter with the information that it is capable of sensing. The nodes communicate with each other to achieve a common estimation and error covariance matrix [8–11].

Some authors have proposed estimators that can refine their estimation even in the absence of information from the sensors. Each triggering criterion defines a region where the sensed signal must lie when there are no updates from the sensors. In [12], a Kalman filter is applied assuming a uniform probability for the aforementioned region. However, the Kalman filter assumes a Gaussian density function for the measurement noise, and hence, a sum of Gaussians to approximate the probability of the region is proposed in [13]. The authors of [14] provide a method to obtain the optimal gain for an estimator with point- and set-valued measurements. This concept is extended in [15], where the minimum mean-squared error (MMSE) estimator is obtained for multiple sensors. The maximum likelihood estimator is also developed in [16].

If event generation is performed independently from the sensed signal, the sensors can be maintained in a standby state. Scheduling of the sample times would thus be performed by a centralized estimator that requests measurements from the sensors when they are necessary.

In controlled systems, there exist several works that perform the sampling in relation to the stability of the system based on a Lyapunov function [17–19]. However, the authors of these papers assume perfect measurements, and hence, estimator uncertainties are neglected.

The performance of an estimator is often evaluated by its estimation error covariance, so it is logical to consider using it to generate sensing events. In [20], the covariance matrix is used to determine an optimal schedule in a heterogeneous sensor network. In [21], a sensor uses the Kullback–Leibler divergence of the estimation error to decide whether a measurement should be sent. In the field of robot localization, in [22], the magnitude of the estimation error covariance is used by the estimator in to choose between using inertial measurements or the GPS signal.

Use of a triggering criterion based on the estimation error covariance for requesting a measurement is also analyzed in [23,24]. In these papers, the authors apply this scheme to linear systems, and for stationary problems, they typically observe convergence to periodic sampling. This convergence is proven for the special case of an unstable scalar system under some conditions.

The contribution of the present paper resides in the combination of a variance-based EBSE with an unscented Kalman filter (UKF) applied to the localization of an autonomous vehicle using external sensors, yielding a system that has the capacity to adapt to the maneuvers performed by the vehicle. As this is a non-linear system, the use of variance-triggered measurements does not make the EBSE converge to periodic solutions (unlike in [24]), and hence, sampling times must be computed online by the estimator module. Since the estimator module located in the vehicle decides the sampling times, the sensors can be maintained in standby, saving energy, bandwidth and processing power.

In a preliminary work [25], the triggering condition was obtained by evaluating the error variance of each state independently. The error variance of each state was maintained below a bounding value by requesting measurements every time the uncertainty of a state approached the bounding value. Here, we propose a new triggering condition that can be used with vehicle guidance control algorithms. The two state variables that represent the position in the Cartesian coordinate system are combined to obtain a single triggering condition, which is related to the estimation distance error.

The advantage of working with a distance error is that it provides a more meaningful and easier to tune threshold value that can be set according to the circumstances. In this case, the proposed EBSE is used in combination with a guidance control algorithm. The triggering threshold tightens as the vehicle approaches the trajectory to be tracked. As a result, many measurements can be prevented, because they are not necessary to fulfil the guidance task.

The remaining part of the paper is structured as follows: Section 2 presents a description of the system under study and the mathematical background used by the estimator and explains how the estimation error covariance is propagated. The contribution of the paper is located in Section 3, which introduces the concept of covariance-triggered measurements and the proposed adaptive distance error threshold. In Section 4, the proposed EBSE is tested by running a simulation, and the results obtained are discussed. Finally, some conclusions are drawn in Section 5.

2. Problem Description

This paper deals with the localization and guidance of an autonomous vehicle using a state-space model and an external sensorial system based on cameras. Figure 1 depicts the main elements of the system. In the center of the figure, there is the autonomous vehicle that executes an estimation algorithm, as well as a guidance control to follow a pre-configured reference path. Above it, camera

sensors that detect the position are connected to the vehicle via a wireless network. The technology used for the external sensors is not relevant, since the proposed method would work with any other kind of localization sensors, such as lasers, ultrasound or infra-red local positioning systems.

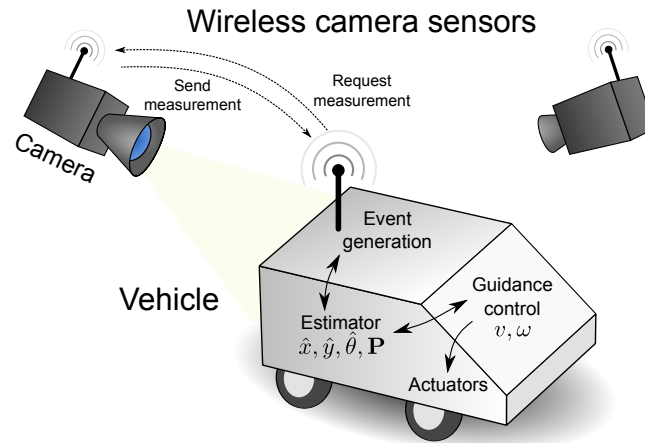


Figure 1. General description of the system showing the most important elements: the vehicle (controller and estimator) and sensors linked by a wireless network.

The sensors are only active when a measurement is requested. When a request is received, the corresponding camera activates and takes its measurement at the desired time. Then, the camera sends the measurement back to the vehicle, where it is processed to refine its pose estimation.

On-boarded sensors, such as wheel encoders or inertial measurement units, could be also used to refine the estimation, as in [25]. However, this paper does not consider any of these sensors in order to focus on the event generation of the remote sensors.

2.1. Mathematical Background

The system states are the coordinates x and y of the vehicle and its orientation angle θ . The continuous-time kinematic equations of the vehicle are as follows:

$$\begin{aligned}\dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= \omega\end{aligned}\quad (1)$$

The symbols v and ω are the system inputs and represent linear and angular velocity, respectively. The input vector $\mathbf{u}_c = [v_c \ \omega_c]^T$ is the combination of the speed commands provided by a guidance algorithm. We consider that there might be uncertainties on the actual speeds due to model inaccuracies and input noises. These uncertainties are modeled as a zero-mean Gaussian random processes added to the input commands. The actual speeds of the system are then expressed as:

$$\begin{aligned}v &= v_c + \Delta v \\ \omega &= \omega_c + \Delta \omega\end{aligned}\quad (2)$$

The above-mentioned random noise has a covariance matrix $\Sigma_{\mathbf{u}}$:

$$E \left[\begin{bmatrix} \Delta v & \Delta \omega \end{bmatrix}^T \begin{bmatrix} \Delta v & \Delta \omega \end{bmatrix} \right] = \Sigma_{\mathbf{u}} \quad (3)$$

The system described in Equation (1) can be summarized in vector form:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \quad (4)$$

where $\mathbf{x} = [x \ y \ \theta]^T \in \mathcal{X} \subset \mathbb{R}^3$ is the state vector, $\mathbf{u} = [v \ \omega]^T \in \mathcal{U} \subset \mathbb{R}^2$ is the input vector and \mathcal{X} and \mathcal{U} are the sets of all possible state points and inputs.

In practice, the system is controlled by a digital system that executes its algorithms periodically with a sample time T . To do so, the continuous-time system Equation (1) must be discretized, for which we propose the second order Runge–Kutta method [26].

$$\begin{aligned} x(t+T) &= x(t) + Tv(t) \cos\left(\theta(t) + \frac{T}{2}\omega(t)\right) \\ y(t+T) &= y(t) + Tv(t) \sin\left(\theta(t) + \frac{T}{2}\omega(t)\right) \\ \theta(t+T) &= \theta(t) + T\omega(t) \end{aligned} \quad (5)$$

The discrete system above approximates the continuous-time system Equation (1) by turning the derivatives into difference equations. The system inputs are computed by the control module at discrete times $T, 2T, 3T, \text{etc.}$, and remain constant within the period T .

Equation (4) becomes:

$$\mathbf{x}(t+T) = f_d(\mathbf{x}(t), \mathbf{u}(t)) \quad (6)$$

The measured states are the x and y coordinates plus some measuring noise. They are obtained by a sensorial system at asynchronous discrete instants t_k . Although these instants are assumed to be a multiple of the sample time T , they are not necessarily periodic and are scheduled by the estimator's event generator. The symbol t_k represents the time of the k -th measurement:

$$t_k = n_k T, \quad n_k \in \mathbb{N}, \quad n_k < n_{k+1} \quad \forall k \quad (7)$$

The output equation of the system is assumed to be a linear equation:

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{w}_k \quad (8)$$

where:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (9)$$

The k suffix applied to a variable denotes the index of an asynchronous event. \mathbf{x}_k and \mathbf{y}_k are short forms for $\mathbf{x}(t_k)$ and $\mathbf{y}(t_k)$. \mathbf{w}_k is an uncorrelated random discrete noise vector with covariance matrix:

$$E[\mathbf{w}_k \mathbf{w}_k^T] = \mathbf{R}_k \quad (10)$$

A Kalman filter can compute the estimated state vector $\hat{\mathbf{x}}$, as well as the estimation error covariance matrix \mathbf{P} .

$$\hat{\mathbf{x}} = E[\mathbf{x}] \quad (11)$$

$$\mathbf{P} = E[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T] = \begin{bmatrix} \sigma_x^2 & \sigma_{x,y} & \sigma_{x,\theta} \\ \sigma_{y,x} & \sigma_y^2 & \sigma_{y,\theta} \\ \sigma_{\theta,x} & \sigma_{\theta,y} & \sigma_\theta^2 \end{bmatrix} \quad (12)$$

Let e_x and e_y be the estimation errors for both x and y coordinates. These are correlated random variables with zero mean. Let $P_{i,j}$ be the i -th row and j -th column entry of \mathbf{P} . Then,

$$\begin{aligned} P_{1,1} &= E[(x - \hat{x})^2] = E[e_x^2] \\ P_{2,2} &= E[(y - \hat{y})^2] = E[e_y^2] \\ P_{1,2} &= P_{2,1} = E[(x - \hat{x})(y - \hat{y})] = E[e_x e_y] \end{aligned} \quad (13)$$

Unfortunately, the Kalman filter is only optimal for linear systems affected by white Gaussian noise processes. The Gaussian probability density functions of the noise propagated through non-linear systems render the density function of the estimated state non-Gaussian, and iterating this non-Gaussian distribution over time becomes an intractable problem.

For a non-linear system, there are some algorithms that extend the idea of the Kalman filter, such as the extended Kalman filter (EKF) and the unscented Kalman filter (UKF) [27]. In both cases, the computed mean of the estimation $\hat{\mathbf{x}}$ and the estimation error covariance matrix \mathbf{P} are not exact, but approximations.

Since Equation (1) is a non-linear set of equations, in our case, the estimation of the state vector is performed with the UKF. Although its computational cost is heavier, it provides a better approximation of \mathbf{P} than the EKF [27].

There are two different stages for Kalman filter estimators. The prediction stage takes into account the plant model equations and the known inputs to advance the estimated point over time. The correction stage updates the estimation with the information obtained from a sensor measurement.

2.2. Prediction Stage

In this stage, $\hat{\mathbf{x}}$ must be propagated through the set of non-linear Equations (5). This is achieved by the unscented transformation [28]. The prediction stage is executed periodically for every T step. Because this stage does not require any information from the real world, It can be executed in real time or not. After receiving a measurement, the prediction for an arbitrary time span can be calculated in advance.

For each time step, a set of sigma-points are generated around the current estimated state point $\hat{\mathbf{x}}$ and are spread out according to \mathbf{P} . The sigma points are state points that sample the probability density function of the state. In order to also take into account the input uncertainties, the state vector and the error covariance matrix are augmented with the mean and covariance of the noise.

$$\hat{\mathbf{x}}_a = \begin{bmatrix} \hat{\mathbf{x}}^T & 0 & 0 \end{bmatrix}^T, \quad \mathbf{P}_a = \begin{bmatrix} \mathbf{P} & 0 \\ 0 & \Sigma_u \end{bmatrix} \quad (14)$$

A total of $2N$ sigma-points $\hat{\mathbf{x}}_a^{(i)}$ are calculated with the formulas:

$$\begin{aligned} \hat{\mathbf{x}}_a^{(i)}(t) &= \hat{\mathbf{x}}_a(t) + \left(\sqrt{N\mathbf{P}_a(t)} \right)_i^T \quad i = 1, 2, \dots, N \\ \hat{\mathbf{x}}_a^{(N+i)}(t) &= \hat{\mathbf{x}}_a(t) - \left(\sqrt{N\mathbf{P}_a(t)} \right)_i^T \quad i = 1, 2, \dots, N \end{aligned} \quad (15)$$

where N is the number of states of the augmented state vector; so, in this case, $N = 5$ (three states plus two input noises), and $\left(\sqrt{N\mathbf{P}_a} \right)_i$ is the i -th row of the Cholesky decomposition of matrix $N\mathbf{P}_a$.

The next estimated point is obtained as the mean of the sigma-points transformed by the discrete system function f_d :

$$\hat{\mathbf{x}}^{(i)}(t+T) = f_d \left(\hat{\mathbf{x}}_{a,1:3}^{(i)}(t), \mathbf{u}_c(t) + \hat{\mathbf{x}}_{a,4:5}^{(i)}(t) \right) \quad (16)$$

$$\hat{\mathbf{x}}(t+T) = \frac{1}{2N} \sum_{i=1}^{2N} \hat{\mathbf{x}}_a^{(i)}(t+T) \quad (17)$$

and the covariance matrix as the cross-covariance of the transformed sigma-points:

$$\mathbf{P} = \frac{1}{2N} \sum_{i=1}^{2N} (\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}}) (\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}})^T \quad (18)$$

2.3. Correction Stage

When the output vector is a linear combination of the states, as described in Equation (8), there is no need to apply the unscented transformation again for the correction stage. The measurement update is computed with the asynchronous Kalman filter (AKF) Equations [29]:

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{L}_k (\mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}_k^-) \quad (19)$$

The symbol $\hat{\mathbf{x}}_k^-$ corresponds to the *a priori* estimation at time t_k (before the correction is performed) and $\hat{\mathbf{x}}_k^+$ to the *a posteriori* estimation (after the correction).

The Kalman filter algorithm is geared to minimize the *a posteriori* estimation error covariance by finding the optimal value of \mathbf{L}_k for each measurement update. It is calculated with the formula:

$$\mathbf{L}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R}_k)^{-1} \quad (20)$$

The resulting *a posteriori* covariance matrix is:

$$\mathbf{P}_k^+ = (\mathbf{I}_N - \mathbf{L}_k \mathbf{H}) \mathbf{P}_k^- \quad (21)$$

where \mathbf{I}_N is the N -th dimensional identity matrix. \mathbf{P}_k^- and \mathbf{P}_k^+ are the *a priori* and *a posteriori* error covariance matrices, respectively (*i.e.*, the values of \mathbf{P} before Equation (18) and after Equation (21), the measurement update). As can be deduced from Equations (20) and (21), the correction stage helps to reduce the trace of \mathbf{P} , and the magnitude of the reduction depends on the precision of the measurement, which is given by the noise covariance matrix \mathbf{R}_k .

3. Covariance-Triggered Measurements

The main requirement for an estimator algorithm is that it should have the capacity to provide an estimation with a small degree of uncertainty. When working with Kalman filters, this means that \mathbf{P} must be bounded.

One idea for obtaining a bounded \mathbf{P} would be to apply a measurement correction whenever the estimation error covariance matrix approaches an imposed threshold condition, which leads to an EBSE [23]. It has been observed that on linear systems with stationary noise, event generation converges to periodic sampling. In [24], this convergence is proven for scalar systems subject to specific conditions.

Moreover, if a periodic sampling is chosen, the problem becomes finding the appropriate sampling time that leads to the desired uncertainty level. To do so, a Riccati equation can be used to determine the steady-state value of \mathbf{P} , and the detectability test of the system is a condition that ensures the existence of a positive-definite solution for that equation [28].

However, in the case of non-linear systems with variable noise parameters, a covariance-triggered EBSE does not converge to a periodic solution. The application of such an EBSE serves two different purposes. On the one hand, it maintains the estimation error covariance bounded. On the other hand, a sample is taken only when it is needed, so the use of sensors, network communications and processing resources is more efficient. In contrast to SoD methods, it is the estimator module (inside the vehicle), rather than the sensor that decides when to take a measurement (event generation).

The sampling intervals depend on the growth and initial value of the covariance matrix. The growth of \mathbf{P} is independent from the sensed signals, so the next sampling instant can be calculated in advance by the estimator module. However, after applying the correction, the value of the estimation is influenced by the measurements, which, in turn, determines the dynamics of the prediction stage. This is why only the following time event, and not the subsequent ones, can be obtained in advance.

The proposed EBSE algorithm is outlined in the flowchart shown in Figure 2.

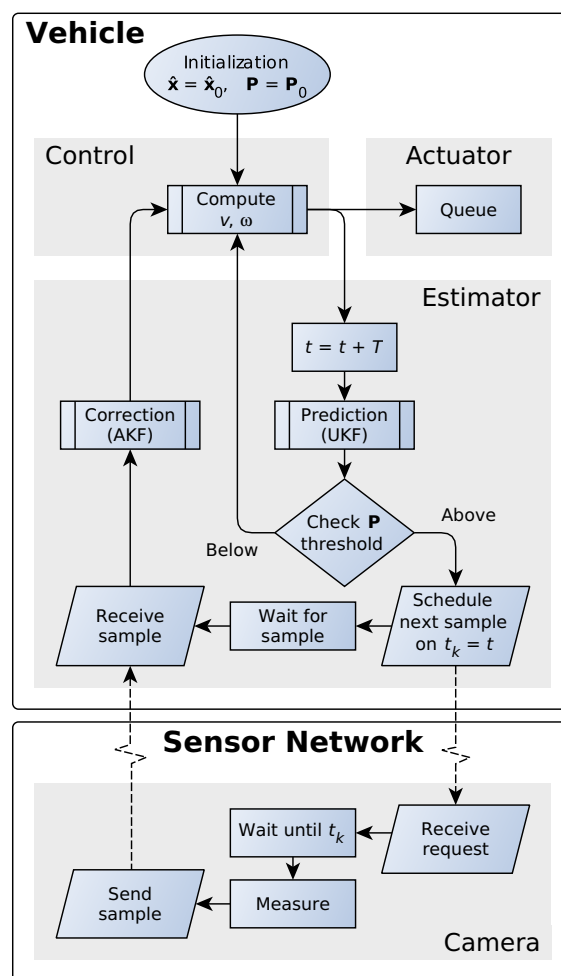


Figure 2. Flowchart of the event-based state estimator (EBSE), executed by the vehicle and the sensors.

With the estimated pose and the reference trajectory, the control module calculates the speed commands for the actuators (motors) with a guidance algorithm, such as [30,31] or similar. Using these commands, the estimator module can predict the location of the vehicle after T seconds and its error covariance. If the estimation error covariance remains below the threshold value, it is possible to calculate in advance the following speed commands, as well as the pose (after $2T$, $3T$, and so on). The speed commands are stored in a queue, so that they can be applied to the motion actuators at the right time.

This prediction process can be repeated until the estimation error covariance exceeds the threshold. When this happens, the estimated location is no longer sufficiently precise, and a measurement is required to reduce \mathbf{P} . Thus, a measurement event is triggered for the time instant t_k , where t_k is the time instant when \mathbf{P} will infringe on the triggering condition.

The estimator module then sends a request for a measurement at time t_k to the corresponding sensor through the network. The sensor remains in a low energy state until t_k and only switches on to take the measurement and send it to the vehicle, then switches off afterwards.

3.1. Distance Error and Orientation Error Thresholds

To design a covariance-triggered EBSE as described above, a condition for \mathbf{P} must be chosen. In [25], each diagonal value was compared independently to a threshold value. A measurement was triggered every time any of the thresholds were exceeded. This condition ensures that the estimated error of each state remains at safe levels.

This paper presents a more intuitive and practical triggering condition. Instead of considering the errors of each state (coordinate) independently, it is more meaningful to have some knowledge about the location error as a distance to the real location. Dealing with the distance error is difficult, because it is a non-linear function of two correlated random variables. This is why it is easier to work with the squared error. Let e_d^2 be the squared distance error, defined as:

$$e_d^2 = e_x^2 + e_y^2 \quad (22)$$

The mean value of e_d^2 can be easily obtained from Equation (13).

$$E[e_d^2] = E[e_x^2 + e_y^2] = E[e_x^2] + E[e_y^2] = P_{1,1} + P_{2,2} \quad (23)$$

The square root of this value is known as the distance root mean squared error (DRMS), and it is a commonly-used indicator of localization precision [32].

$$\text{DRMS} = \sqrt{E[e_d^2]} = \sqrt{P_{1,1} + P_{2,2}} \quad (24)$$

The probability of finding the real location within a ball centered on the estimated location and with a radius of DRMS for a Gaussian distribution is about 65%. The same ball, but with twice the radius (known as 2DRMS) raises the probability to 95%.

The estimation error of the orientation angle θ should also be taken into account. Accurate estimation of θ is critical for computation of the guidance control algorithm. Therefore, its own triggering condition

is included to ensure that orientation uncertainty is always sufficiently small. The variance of the orientation angle error is given by the third element of the diagonal of \mathbf{P} , as shown in Equation (12).

Although the orientation angle is not measured by the sensors, an observability test [33] on the non-linear system Equations (1) and (8) can determine that the system is locally observable if the linear speed v is non-zero [34]. This means that every time the vehicle is moving, a measurement of position carries some information about the orientation and therefore can be used to reduce its estimation error variance.

By combining the distance and orientation error as mentioned above, the triggering condition for the sensors is the following: request a sample from a sensor iff:

$$(P_{1,1} + P_{2,2} > D_{\text{thr}}^2) \vee (P_{3,3} > \tilde{\theta}_{\text{thr}}^2) \quad (25)$$

D_{thr} and $\tilde{\theta}_{\text{thr}}$ are threshold values that must be adjusted by the designer according to his or her needs. Thus, the expected DRMS would be lower than D_{thr} , and more than 95% of the time, the distance error will be below $2D_{\text{thr}}$. Similarly, the orientation estimation error is expected to be around $\tilde{\theta}_{\text{thr}}$, with a 95% chance of being lower than $2\tilde{\theta}_{\text{thr}}$.

The threshold values may not be constant and may vary along the route in order to adapt to the changing circumstances. In the following section, an adaptive threshold is proposed that takes into account the distance of the vehicle to the reference point.

3.2. Adaptive Distance Error Threshold

In the guidance problem, the vehicle must follow a reference track, but usually starts somewhere away from the initial position of this reference trajectory. The solution of the guiding problem can be divided into two different stages. First, the vehicle needs to approach the area of the reference trajectory and then follow it. The time during which the control algorithm is approaching the vehicle towards the trajectory is referred to as the approaching time.

When this task is completed, the vehicle is near the reference point and simply moves along the reference path. This stage is referred to as the tracking time.

During approaching time, the control algorithm does not need a very accurate estimated location. If the vehicle is far from the reference position, it is sufficient to have a rough idea of where it is, because the speed commands computed by the guidance algorithm would not differ greatly among the uncertainty region of the vehicle.

While approaching the trajectory, the radius of the uncertainty area (defined by the distance error threshold D_{appr}) should be small compared with the distance of the vehicle to the reference point. This is easy to achieve if they maintain a fixed ratio of K_D . In other words, during the approaching maneuver, D_{thr} can be set as:

$$D_{\text{thr}} \approx \hat{L}K_D = D_{\text{appr}} \quad (26)$$

where \hat{L} is the distance from the estimated position to the reference point.

To understand the meaning of constant K_D , let α be the angle between the estimated location and the limit of the uncertainty area as seen from the reference location (see Figure 3). Then,

$$\alpha = \arctan \left(\frac{D_{\text{appr}}}{\hat{L}} \right) \quad (27)$$

Substituting Equation (26) into Equation (27) and solving for K_D yields:

$$K_D = \tan \alpha \quad (28)$$

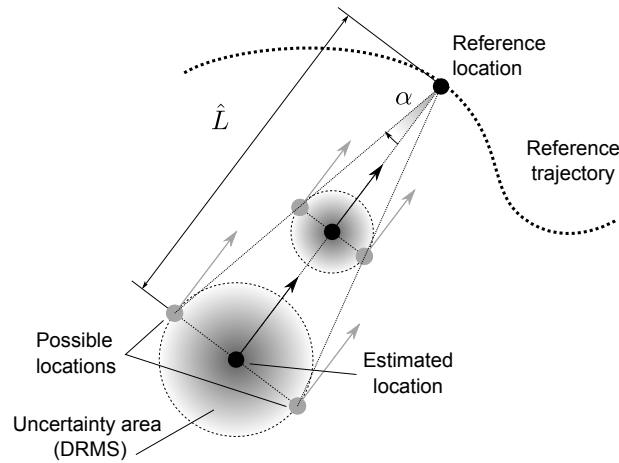


Figure 3. Diagram explaining the relation Equation (26) between the distance to the reference location and the size of the uncertainty region.

The problem with a linear relation between \hat{L} and D_{appr} is that the vehicle will move closer and closer to the reference trajectory, so \hat{L} will tend to zero, and thus, D_{thr} will also tend to zero. An excessively low threshold leads to periodic sampling at the sensor's fastest sampling rate. The threshold should have a minimum value D_{trk} . This parameter must be tuned according to the acceptable error during tracking time. The following smoothing function for the distance error threshold is proposed:

$$D_{\text{thr}} = \sqrt{D_{\text{trk}}^2 + D_{\text{appr}}^2} = \sqrt{D_{\text{trk}}^2 + (\hat{L}K_D)^2} \quad (29)$$

This is a smoothing function that is close to D_{appr} when the reference point is distant (while approaching) and close to D_{trk} when the vehicle is near the reference point (while tracking). A graphical representation of the function is plotted in Figure 4.

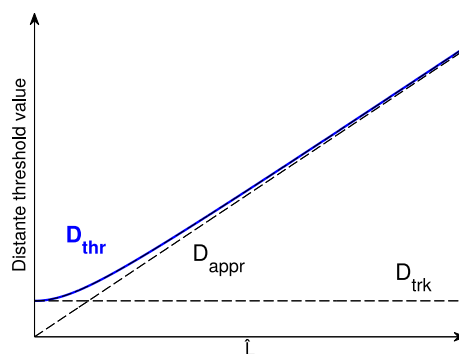


Figure 4. Adaptive distance threshold function D_{thr} as a function of the distance to the reference location.

Although the distance error is not critical during approaching time, orientation is still important. It is always critical to know where the vehicle is heading, even when it is far away from the reference point.

Otherwise, any attempt by the control module to approach the desired point is not guaranteed to actually take it closer. This means that the threshold value for orientation error should remain a fixed value.

3.3. Limitations

A few assumptions are required to guarantee the correct operation of the proposed EBSE.

The first assumption is a perfect communication channel between the estimator and the sensors. However, it is possible to allow some delay in the transmission of the measurement packets, because the AKF can take care of them, even if they arrive out of sequence [29]. In other words, when the packet is received, the AKF can be applied to correct the estimation, even though it corresponds to a past time instant. Nonetheless, the covariance matrix would increase over the threshold until the packet with the measurement arrives.

The second assumption is that the only disturbances that affect the states are those modeled by input noise and its covariance matrix Σ_u (see Equation (3)). If, for example, the wheels slip on the ground, this will produce an estimation error, and the estimator will not react to it. As the estimator receives new measurements, the estimation will ultimately converge to the actual state, but the sampling times will not be adjusted to ensure a rapid correction of the estimation.

This is also a requirement for having an accurate \mathbf{P} computed by the filter that truly reflects the estimation error covariance, which is the cornerstone of the proposed EBSE. As explained above, the UKF only computes an approximation of this covariance matrix, and the performance of the EBSE is tied to the accuracy that the filter can achieve on this approximation. Anyway, the UKF is well known for providing a good approximation of \mathbf{P} , but nevertheless, in every practical scenario where the UKF does not work well, this EBSE will become impractical.

Finally, the third assumption is that the growth of \mathbf{P} during the minimum acquisition time of the sensors is less than the reduction that such measurements can perform on \mathbf{P} . If this assumption is not met, then even measuring at the fastest rate would not effectively reduce \mathbf{P} over time. As a consequence, it cannot be guaranteed that \mathbf{P} will remain below the desired limits. In this case, the estimator will request measurements from the sensors constantly, the sensors will provide samples at their fastest rate and the estimator will behave in a periodic fashion (where the time period is the minimum sampling time of the sensor). This problem arises when the threshold values are set too low, and therefore, the desired uncertainty cannot be met.

In order to determine the highest limits in DRMS and σ_θ^2 (the variance of the orientation estimation error, *i.e.*, $P_{3,3}$) that the system could be subjected to, it is possible to examine the worst-case scenarios where estimation uncertainty grows faster and the measurements have the highest possible noise covariance.

Different worst-case scenarios must be found for each of the two triggering conditions (DRMS and angle). Let:

$$\mathbf{F}_w = \frac{\partial f_d(\mathbf{x}_w, \mathbf{u}_w)}{\partial \mathbf{x}_w}, \quad \mathbf{G}_w = \frac{\partial f_d(\mathbf{x}_w, \mathbf{u}_w)}{\partial \mathbf{u}_w} \quad (30)$$

be the linearization matrices of the system for the worst-case scenario state point and inputs.

Assuming that the minimum acquisition time of the sensor is $T_s = MT$, where $M \in \mathbb{N}$, we define the equivalent matrices for a discrete system:

$$\mathbf{F}_{\text{eq}} = \mathbf{F}_w^M \quad (31)$$

$$\mathbf{Q}_{\text{eq}} = \sum_{i=0}^{M-1} \mathbf{F}_w^i \mathbf{G}_w \Sigma_u \mathbf{G}_w^T (\mathbf{F}_w^i)^T \quad (32)$$

If the system stayed in these worst-case scenario conditions for a long time, it would be equivalent to a linear system, and therefore, matrix \mathbf{P} would converge to a steady-state value that can be computed by solving a discrete-time algebraic Riccati equation (DARE):

$$\mathbf{P}_w = \mathbf{F}_{\text{eq}} \mathbf{P}_w \mathbf{F}_{\text{eq}}^T + \mathbf{F}_{\text{eq}} \mathbf{P}_w \mathbf{H}^T (\mathbf{H} \mathbf{P}_w \mathbf{H}^T + \mathbf{R}_w)^{-1} \mathbf{H} \mathbf{P}_w \mathbf{F}_{\text{eq}}^T + \mathbf{Q}_{\text{eq}} \quad (33)$$

where \mathbf{R}_w is the worst measuring noise covariance matrix that the sensor can provide. The solution \mathbf{P}_w of Equation (33) defines the maximum uncertainty that the EBSE could reach.

For each of the two triggering conditions (DRMS and angle), finding \mathbf{x}_w and \mathbf{u}_w for each case is an optimization problem: these are the points that lead to a \mathbf{P}_w , which has the maximum DRMS or σ_θ^2 , respectively.

Let D_w and $\tilde{\theta}_w$ be the values of DRMS and σ_θ of the worst-case scenarios mentioned above.

$$D_w^2 = \max_{\substack{\mathbf{x}_w \in \mathcal{X} \\ \mathbf{u}_w \in \mathcal{U}}} \mathbf{P}_w(1, 1) + \mathbf{P}_w(2, 2) \quad (34)$$

$$\tilde{\theta}_w^2 = \max_{\substack{\mathbf{x}_w \in \mathcal{X} \\ \mathbf{u}_w \in \mathcal{U}}} \mathbf{P}_w(3, 3) \quad (35)$$

For the case under study, \mathbf{u}_w for the DRMS condition is related to the maximum linear speed of the vehicle, and conversely, \mathbf{u}_w for the σ_θ^2 depends on the minimum linear speed. As stated above in this section, the orientation angle is an observable state as long as $v \neq 0$, so if the vehicle is not moving, θ cannot be estimated at all. Otherwise, the orientation and position uncertainties are correlated, and therefore, applying a position measurement update must reduce the orientation error variance. Additionally, the slower the speed, the less information can be drawn from a measurement, but provided that the vehicle moves with a guaranteed minimum speed, a solution to Equation (35) can be found.

If the triggering threshold values of Equation (25) are set to equal to or greater than D_w and $\tilde{\theta}_w$, then it can be guaranteed that the EBSE will maintain uncertainty below those bounds. This is true because the uncertainties will never grow faster than they do in the system represented by Equation (33), and the measurements will always have a better than or equal noise covariance matrix \mathbf{R}_w , but even under these worst-case conditions, the estimator is able to keep the uncertainties bounded.

These threshold values might be too high for some applications, and in general, the EBSE can perform better. However, in this case, the EBSE would work in a best-effort fashion where the uncertainty goal may not be achieved. Nevertheless, this uncertainty will be lower than or equal to D_w and $\tilde{\theta}_w$.

4. Illustrative Example

As a proof of concept, the proposed EBSE technique was tested in a simulation and compared to a periodic sampling estimator. To do so, the camera sensors were modeled and simulated, as well.

4.1. Simulation Setup

The tests were run with a discrete time step of $T = 10$ ms, as was the controller. The selected reference trajectory was a figure-eight shape described by:

$$\begin{cases} x_r(t) = 5 + 4.5 \sin\left(\frac{2\pi}{100}t + \frac{\pi}{2}\right) & [\text{m}] \\ y_r(t) = 5 + 3.5 \sin\left(\frac{4\pi}{100}t\right) & [\text{m}] \end{cases} \quad (36)$$

The covariance matrix of the noise added to the inputs, explained in Equation (3), was:

$$\Sigma_u = \begin{bmatrix} 10^{-4} & 0 \\ 0 & 10^{-2} \end{bmatrix}. \quad (37)$$

The vehicle implemented the Lyapunov-based guidance control (LGC) described in [30] for approaching and tracking the trajectory. This controller, based on the 2D non-holonomic unicycle system Equation (1), is intended for guiding a mobile robot when approaching and following a pre-programmed path.

The initial state vector of the vehicle was:

$$\mathbf{x}_0 = \begin{bmatrix} x_0 \\ y_0 \\ \theta_0 \end{bmatrix} = \begin{bmatrix} 7 \\ 5 \\ 0 \end{bmatrix} \quad (38)$$

where the coordinate values are measured in meters and the orientation angle is in radians.

To ensure a short transient time for the estimator, the starting position of the robot is assumed to be known, with a small degree of uncertainty. This makes it possible to focus on the behavior of the estimator once it has converged to a value close to the real state. The initial estimation vector was identical to the real state vector, and the initial state estimation error covariance matrix was:

$$\mathbf{P}_0 = \begin{bmatrix} 0.1^2 & 0 & 0 \\ 0 & 0.1^2 & 0 \\ 0 & 0 & \left(\frac{\pi}{6}\right)^2 \end{bmatrix} \quad (39)$$

4.2. Camera Sensors

The sensors used to measure the location (x and y) of the vehicle were two cameras covering the working scenario. These cameras were simulated using the pin-hole geometric model [35] to imitate an inexpensive camera, such as the Unibrain Fire-i.

They were located at a height of three meters, pointing towards the ground at a 30° angle. The image resolution was 640×480 pixels, and the focal length was 4.3 mm. The minimum time between consecutive measurements taken by the cameras was 80 ms. The maximum rate of the sensor was therefore 12.5 frames per second (FPS). These cameras are able to deliver up to 30 FPS (33.3 ms of acquisition time), but the processing time of each frame must also be taken into account.

Figure 5 shows the reference trajectory, the location of the cameras and each camera's field of view (FOV). Each camera covered one side of the figure-eight shape. Both cameras' FOV overlapped in the

center, but for the sake of simplicity, only one of them was used at a time. The corresponding area for each camera is delimited by the red dashed line.

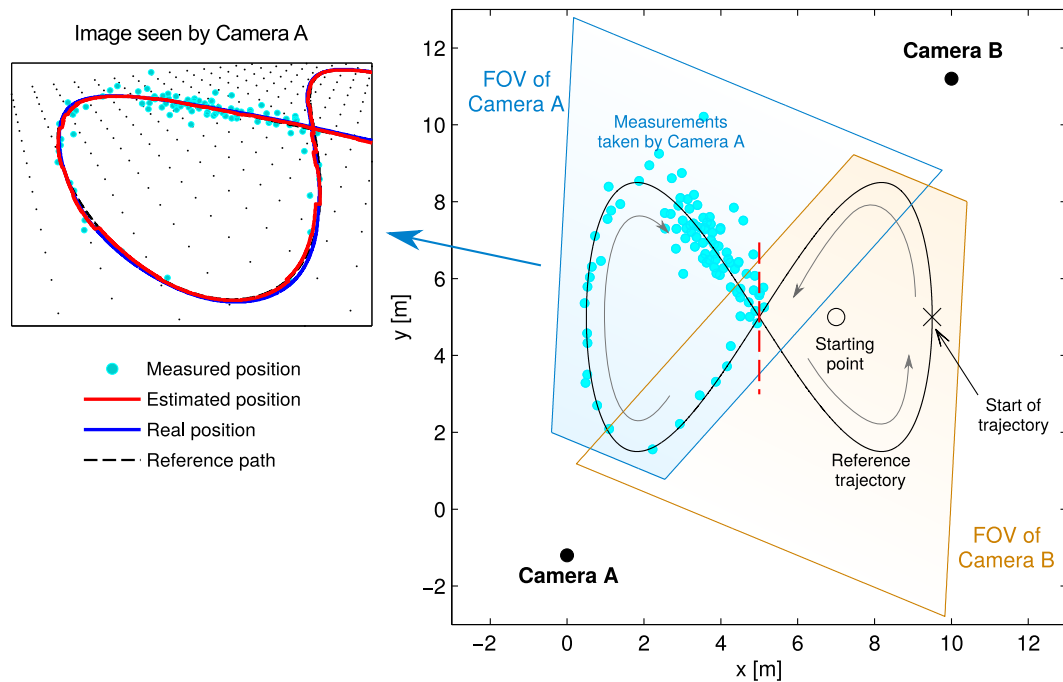


Figure 5. Diagram showing the reference trajectory and the camera locations. Each camera's field of view (FOV) is given.

The non-linear transformation of the camera model could be performed by the UKF in the correction stage. However, in order to keep the estimator module independent from the kind of sensor technology used, it is assumed that the sensors are capable of delivering a position vector, such as Equation (8) and a noise covariance matrix \mathbf{R}_k . Otherwise, the process of calibrating the cameras, or maybe substituting them with some other sensing technology, such as laser or ultrasound, would involve reconfiguration of the estimator.

The position of the vehicle is assumed to be determined by an image recognition algorithm that identifies the vehicle in a pair of coordinates (U_k, V_k) in the picture taken by the corresponding camera (e.g., [36]). To simulate the errors and deviations that the algorithm might make, zero-mean Gaussian random numbers ΔU_k and ΔV_k were added to each of the exact coordinates.

$$\begin{aligned} U'_k &= U_k + \Delta U_k \\ V'_k &= V_k + \Delta V_k \end{aligned} \quad (40)$$

In the above equation, U_k and V_k represent the exact point of the vehicle in the image and U'_k and V'_k represent the pixel coordinates that the simulated vision algorithm provides. Thus, the pixel in the image is related to a point on the floor ($z = 0$ plane) by the geometric equations of the camera model. The transformation of the point (U'_k, V'_k) results in vector \mathbf{y}_k , which contains the noise \mathbf{w}_k , as described in Equation (8).

The noises added to the two axes have a standard deviation of 12 pixels, and they are independent of each other. This yields the covariance matrix:

$$E \left[\begin{bmatrix} \Delta U_k & \Delta V_k \end{bmatrix}^T \begin{bmatrix} \Delta U_k & \Delta V_k \end{bmatrix} \right] = \Sigma_i \quad (41)$$

$$\Sigma_i = \begin{bmatrix} 12^2 & 0 \\ 0 & 12^2 \end{bmatrix} \quad (42)$$

However, because the transformation of coordinates from the image to the world is a non-linear function, \mathbf{w}_k has a covariance matrix \mathbf{R}_k that is not diagonal and depends on the perspective of the point from the camera. This is small for points closer to the camera and becomes larger as the point moves farther away. \mathbf{R}_k is calculated from Σ_i using the unscented transformation.

The bottom plot in Figure 5 shows how the measurements were simulated. It represents the scene as seen by one of the cameras. The black dots are intended to give an idea of the perspective of the ground. The distance between them is 50 cm.

Each cyan dot represents a measurement. They can be related one by one to the points in the top diagram to understand the effect of the perspective transformation performed by the camera.

This error magnitude is similar for every measurement, as seen in the picture. However, the measurements taken when the vehicle is close to the camera (in the lower half) are fairly accurate, whereas the ones corresponding to the upper half are not, in terms of distance in the real world. Consequently, a higher number of measurements are taken when the vehicle is moving at a distance from the camera, because the EBSE needs more of them to estimate its position with the same level of uncertainty.

4.3. Results

The results are presented as the comparison of three different cases. Figure 6 shows the trajectory followed by each strategy, and also the reference trajectory.

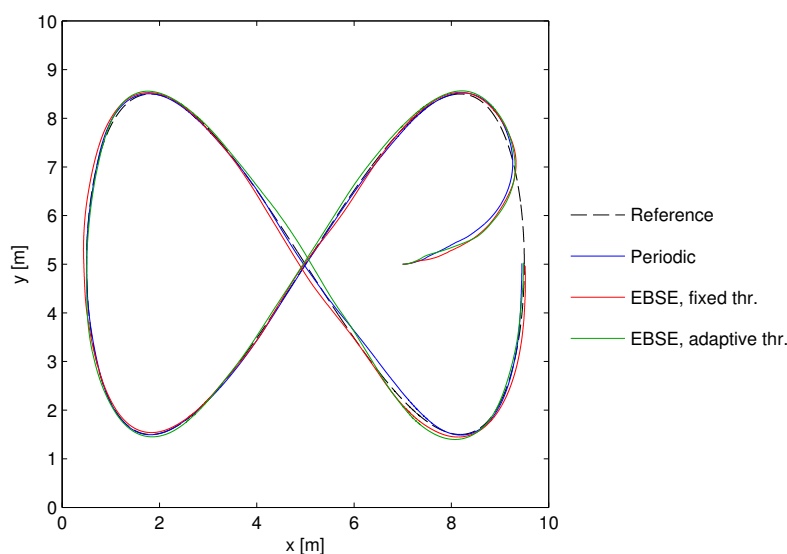


Figure 6. Comparison of the trajectory followed by the three sampling schemes, as well as the reference trajectory. The position of the camera sensors and their field-of-view (FOV) are also shown.

The three cases are:

1. (Blue) The UKF with periodic sampling. This represents the best possible estimation for the frame rate offered by the cameras.
2. (Red) The EBSE with a fixed distance threshold of $D_{\text{thr}} = 75$ mm.
3. (Green) The EBSE with the adaptive threshold described in Equation (29), where $D_{\text{trk}} = 75$ mm and $K_D = 1/6$.

For the two last cases, the angle threshold was set to $\tilde{\theta}_{\text{thr}} = \pi/10$ rad.

The periodic sampling strategy showed the best performance, because it used all of the information that the sensors could provide. Nevertheless, the other two methods also performed well, while only using a small fraction of the total number of measurements. The trajectories were within an error margin that would be acceptable for practical applications.

In order to compare the estimation error, the DRMS of each error is plotted in Figure 7, calculated with Equation (24). The top plot shows the number of measurements taken per second.

During tracking time (from $t = 8$ s onwards), the two EBSE showed the same behavior and maintained their DRMS below 75 mm.

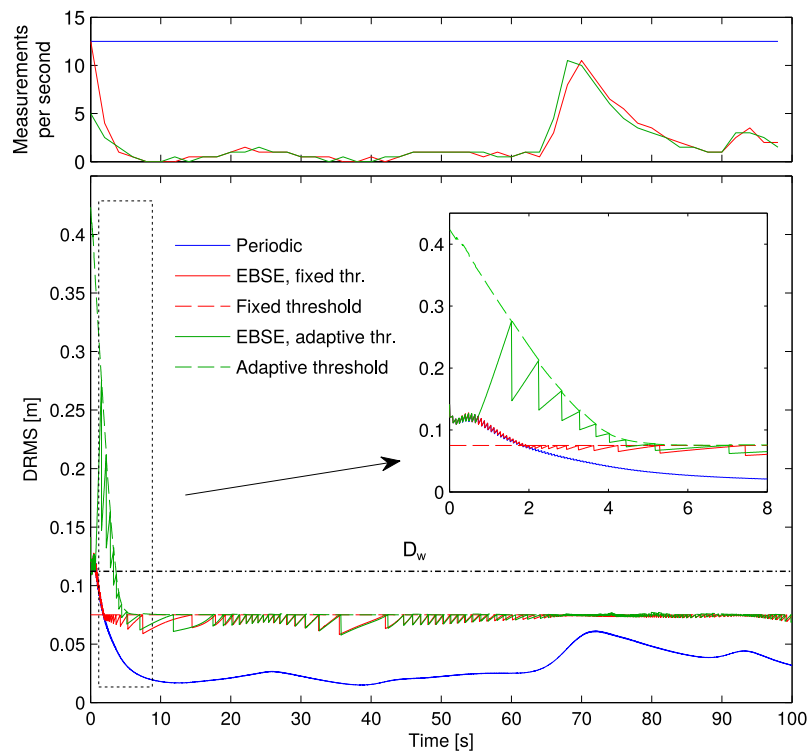


Figure 7. Measurement rate and evolution of the distance root mean squared error (DRMS) over time for the three estimation methods. The threshold values are also shown. The approaching time (until the eighth second of the simulation) is zoomed in the plot in the top left corner.

The periodic sampling estimator had a smaller DRMS, but it increased up to 61 mm in $t = 72$ s. When the vehicle was moving at a distance from the cameras (by the second half of the simulation time), their measurements contained more noise and, hence, provided less information to the estimator. As a result, this yielded greater uncertainty for the estimation.

This level of uncertainty was not far from the imposed threshold of the EBSE, and if it were considered tolerable, then it would be more efficient to reduce the use of the sensors whenever the estimation error was good enough.

The solution of Equation (34) for this case is $D_w = 113$ mm for a maximum speed $v < 0.7$ m/s, and the solution of Equation (35) is $\tilde{\theta}_w = 0.082$ rad for a minimum speed $v > 0.01$ m/s. Since $\tilde{\theta}_{thr} > \tilde{\theta}_w$, the expected angle uncertainty can be guaranteed. In contrast, D_{trk} is set below D_w , and thus, the desired DRMS might not be achieved all of the time; but as the plot shows, in this case, it was achieved.

Within the DRMS plot, there is also a zoomed plot of the first eight seconds of the simulation. In the case of the fixed threshold EBSE, the estimator started with an initial uncertainty (P_0) larger than the threshold, so it required as many measurements as it could obtain to reduce it quickly. The behavior was therefore identical to the periodic estimator until the DRMS dropped below the threshold. Then, a reduction in the use of the sensors began to take place.

However, the adaptive threshold EBSE only used a few measurements during approaching time. It obtained some at the beginning, triggered by the orientation threshold $\tilde{\theta}_{thr}$ in order to accurately determine the vehicle's orientation. Subsequently, very few measurements were required to locate the vehicle. The DRMS was very high compared with the other two methods, but it was still good enough to guide the vehicle towards the reference trajectory.

The approaching maneuver performed by the vehicle using the three different estimators can be compared in the trajectories shown in Figure 6 and also in the plot in Figure 8. This latter plot represents the distance of the vehicle to the corresponding point of the reference trajectory over time. The dashed line represents the minimum DRMS threshold of the EBSE estimators. Provided that the estimation error is somewhere around this value, the tracking performance is also limited by it.

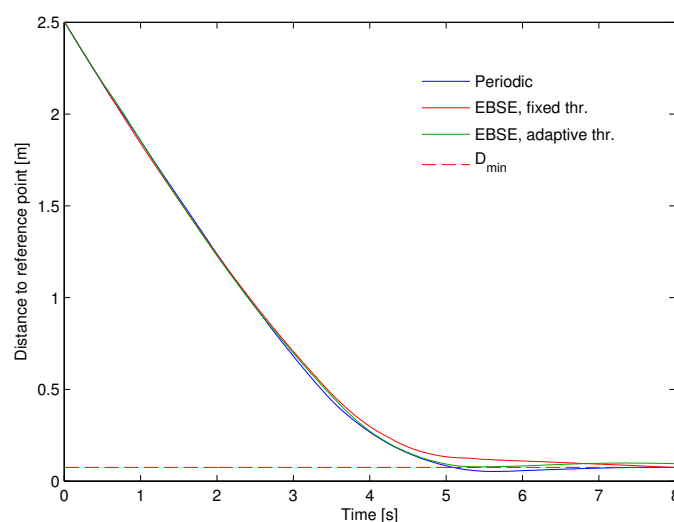


Figure 8. Distance to the point of the corresponding reference trajectory over time for the first eight seconds of the simulation (approaching time).

The plot shows a very similar behavior for the three alternatives. In other words, the adaptive threshold EBSE's reduced use of the sensors did not imply a noticeable deterioration of the guidance during the approaching time.

Table 1 compares the performance of the three estimators during tracking time. The numbers shown are the average of 20 different simulations of each case. The estimation distance error column is the root mean square of the total distance error committed by the estimator for all of the tracking time. The position error column is the root mean squared distance between the real location of the vehicle and the corresponding point of the reference trajectory. The mean number of measurements taken is also shown.

Table 1. Numerical results during tracking time (average results of 20 simulations).

	Number of Measurements	Estimation Distance RMS Error (mm)	Position RMS Error (mm)
Periodic sampling	1150	31.2	41.0
EBSE, fixed threshold	173.1	67.4	76.7
EBSE, adaptive threshold	170.8	68.9	78.5

As mentioned above, the effect of the adaptive threshold is hardly noticeable during tracking. However, Table 2 shows the results for the approaching time. The adaptive threshold EBSE halves the number of measurements compared with the fixed threshold EBSE, while the position error is very similar for all three cases.

Table 2. Numerical results during approaching time (average results of 20 simulations).

	Number of Measurements	Estimation Distance RMS Error (mm)	Position RMS Error (mm)
Periodic sampling	100	54.8	1028.5
EBSE, fixed threshold	35.6	76.4	1031.8
EBSE, adaptive threshold	17.6	102.4	1041.8

5. Conclusions

This paper presents a combination of adaptive variance-based EBSE with a UKF that complements the guidance control of an autonomous vehicle whose position is detected by external sensors. Its use reduces the number of measurements taken without generating a deterioration in vehicle performance while performing approaching and tracking maneuvers. In addition, the desired DRMS of the estimation (which is one of the system parameters) is achieved. The results of the simulation tests confirm these conclusions and show that the number of measurements can be reduced to a small fraction of the total taken when using periodic sampling.

To implement the proposed algorithm, the remote sensor modules require limited intelligence: simply the capacity to respond to a vehicle request. In turn, they can be maintained in a standby state for most of the time.

The algorithm is tuned by adjusting three parameters: D_{trk} , K_D and $\tilde{\theta}_{\text{thr}}$, which are directly related to the desired estimation performance. Where the parameters are too restrictive and the desired performance

cannot be met, the system would demand measurements from the sensors at their fastest rate. In this worst-case scenario, the EBSE would then simply become a periodic UKF.

Acknowledgments

This work has been supported by the Spanish Ministry of Economy and Competitiveness through the ALCOR project (ref. DPI2013-47347-C2-1-R) and by the University of Alcalá FPI 2012 grant program (Formación del Personal Investigador).

Author Contributions

Miguel Martínez-Rey and Felipe Espinosa conceived of the main proposal of the paper and developed the theoretical basis for the adaptive EBSE. Carlos Santos gave technical advice about guidance control algorithms and trajectory tracking for mobile robots, and Alfredo Gardel helped with the modeling of the camera sensors. The simulation experiment was designed by Alfredo Gardel and Carlos Santos, and Miguel Martínez-Rey implemented it. Miguel Martínez-Rey redacted the manuscript, while Felipe Espinosa and Alfredo Gardel proofread it.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Guinaldo, M.; Fábregas, E.; Farias, G.; Dormido-Canto, S.; Chaos, D.; Sánchez, J.; Dormido, S. A Mobile Robots Experimental Environment with Event-Based Wireless Communication. *Sensors* **2013**, *13*, 9396–9413.
2. Rampinelli, M.; Covre, V.B.; de Queiroz, F.M.; Vassallo, R.F.; Bastos-Filho, T.F.; Mazo, M. An Intelligent Space for Mobile Robot Localization Using a Multi-Camera System. *Sensors* **2014**, *14*, 15039–15064.
3. Åström, K.J.; Bernhardsson, B.M. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In Proceedings of the 41st IEEE Conference on Decision and Control (CDC), Las Vegas, NV, USA, 10–13 December 2002; Volume 2, pp. 2011–2016.
4. Miskowicz, M. Send-on-Delta Concept: An Event-Based Data Reporting Strategy. *Sensors* **2006**, *6*, 49–63.
5. Miskowicz, M. Event-based sampling strategies in networked control systems. In Proceedings of the 10th IEEE Workshop on Factory Communication Systems (WFCS), Toulouse, France, 5–7 May 2014; pp. 1–10.
6. Suh, Y.S. Send-on-Delta Sensor Data Transmission With A Linear Predictor. *Sensors* **2007**, *7*, 537–547.
7. Sijs, J.; Kester, L.; Noack, B. A study on event triggering criteria for estimation. In Proceedings of the 17th International Conference on Information Fusion (FUSION), Salamanca, Spain, 7–10 July 2014; pp. 1–8.

8. Sijs, J.; Lazar, M. A distributed Kalman filter with global covariance. In Proceedings of the American Control Conference (ACC), San Francisco, CA, USA, 29 June–1 July 2011; pp. 4840–4845.
9. Sijs, J.; Papp, Z. Towards self-organizing Kalman filters. In Proceedings of the 15th International Conference on Information Fusion (FUSION), Singapore, 9–12 July 2012; pp. 1012–1019.
10. Trimpe, S.; D’Andrea, R. An Experimental Demonstration of a Distributed and Event-Based State Estimation Algorithm. In Proceedings of the 18th IFAC World Congress, Milano, Italy, 28 August–2 September 2011; pp. 8811–8818.
11. Trimpe, S. Event-Based State Estimation with Switching Static-Gain Observers. In Proceedings of the 3rd IFAC Workshop on Distributed Estimation and Control of Networked Systems, Santa Barbara, CA, USA, 14–15 September 2012; Volume 3, pp. 91–96.
12. Suh, Y.S.; Nguyen, V.H.; Ro, Y.S. Modified Kalman filter for networked monitoring systems employing a send-on-delta method. *Automatica* **2007**, *43*, 332–338.
13. Sijs, J.; Lazar, M. Event Based State Estimation with Time Synchronous Updates. *IEEE Trans. Autom. Control.* **2012**, *57*, 2650–2655.
14. Sijs, J.; Noack, B.; Hanebeck, U.D. Event-based state estimation with negative information. In Proceedings of the 16th International Conference on Information Fusion (FUSION), Istanbul, Turkey, 9–12 July 2013; pp. 2192–2199.
15. Shi, D.; Chen, T.; Shi, L. An event-triggered approach to state estimation with multiple point- and set-valued measurements. *Automatica* **2014**, *50*, 1641–1648.
16. Shi, D.; Chen, T.; Shi, L. Event-triggered maximum likelihood state estimation. *Automatica* **2014**, *50*, 247–254.
17. Durand, S.; Marchand, N.; Guerrero Castellanos, J.F. Simple Lyapunov Sampling for Event-Driven Control. In Proceedings of the 18th IFAC World Congress, Milan, Italy, 28 August–2 September 2011; pp. 8724–8730.
18. Marchand, N.; Durand, S.; Castellanos, J.F.G. A General Formula for Event-Based Stabilization of Nonlinear Systems. *IEEE Trans. Autom. Control* **2013**, *58*, 1332–1337.
19. Tabuada, P. Event-Triggered Real-Time Scheduling of Stabilizing Control Tasks. *IEEE Trans. Autom. Control* **2007**, *52*, 1680–1685.
20. Sandberg, H.; Rabi, M.; Skoglund, M.; Johansson, K.H. Estimation over heterogeneous sensor networks. In Proceedings of the IEEE 47th Conference on Decision and Control (CDC), Cancun, Mexico, 9–11 December 2008; pp. 4898–4903.
21. Marck, J.W.; Sijs, J. Relevant Sampling Applied to Event-Based State-Estimation. In Proceedings of the 4th International Conference on Sensor Technologies and Applications (SENSORCOMM), Venice, Italy, 18–25 July 2010; pp. 618–624.
22. Marín, L.; Vallés, M.; Soriano, A.; Valera, A.; Albertos, P. Multi Sensor Fusion Framework for Indoor-Outdoor Localization of Limited Resource Mobile Robots. *Sensors* **2013**, *13*, 14133–14160.

23. Trimpe, S.; D'Andrea, R. Reduced communication state estimation for control of an unstable networked control system. In Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), Orlando, FL, USA, 12–15 December 2011; pp. 2361–2368.
24. Trimpe, S.; D'Andrea, R. Event-Based State Estimation with Variance-Based Triggering. *IEEE Trans. Autom. Control* **2014**, *59*, 3266–3281.
25. Martínez, M.; Espinosa, F.; Gardel, A.; Santos, C.; García, J. Pose Estimation of a Mobile Robot Based on Network Sensors Adaptive Sampling. In *ROBOT2013: First Iberian Robotics Conference*; Armada, M.A., Sanfeliu, A., Ferre, M., Eds.; Springer: Madrid, Spain, 2014; Volume 253, pp. 569–583.
26. De Luca, A.; Oriolo, G.; Vendittelli, M. Control of Wheeled Mobile Robots: An Experimental Overview. In *Ramsete*; Nicosia, S., Siciliano, B., Bicchi, A., Valigi, P., Eds.; Springer: Berlin, Germany, 2001; Volume 270, pp. 181–226.
27. Julier, S.J.; Uhlmann, J.K. Unscented filtering and nonlinear estimation. *IEEE Proc.* **2004**, *92*, 401–422.
28. Simon, D. *Optimal State Estimation: Kalman, H_∞ , and Nonlinear Approaches*; Wiley: Hoboken, NJ, USA, 2006.
29. Mallick, M.; Coraluppi, S.; Carthel, C. Advances in asynchronous and decentralized estimation. In Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, 10–17 March 2001; Volume 4, pp. 4/1873–4/1888.
30. Amoozgar, M.H.; Zhang, Y.M. Trajectory tracking of Wheeled Mobile Robots: A kinematical approach. In Proceedings of the IEEE/ASME International Conference on Mechatronics and Embedded Systems and Applications (MESA), Suzhou, China, 8–10 July 2012; pp. 275–280.
31. Kelouwani, S.; Ouellette, C.; Cohen, P. Stable and Adaptive Control for Wheeled Mobile Platform. *Intell. Control Autom.* **2013**, *4*, 391–405.
32. Kaplan, E.D.; Hegarty, C.J. *Understanding GPS: Principles and Applications*, 2nd ed.; Artech House: Norwood, MA, USA, 2005.
33. Hermann, R.; Krener, A.J. Nonlinear controllability and observability. *IEEE Trans. Autom. Control* **1977**, *22*, 728–740.
34. Bayat, M.; Hassani, V.; Aguiar, A.P. Nonlinear Kalman based filtering for pose estimation of a robotic vehicle from discrete asynchronous range measurements. In Proceedings of the 8th Portuguese Conference on Automatic Control (CONTROLO), Vila Real, Portugal, 21–23 July 2008; pp. 1–6.
35. Hartley, R.I.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2004.
36. Pflugfelder, R.; Bischof, H. Localization and Trajectory Reconstruction in Surveillance Cameras with Nonoverlapping Views. *IEEE Trans. Pattern Anal. Machine Intell.* **2010**, *32*, 709–721.