CAN, GPS and IMU Asynchronous Sensor Fusion for Heavy-Duty Vehicles

Vicent Girbés-Juan[1], Leopoldo Armesto, Daniel Hernández-Ferrándiz, Juan F. Dols and Antonio Sala

Abstract—**In heavy-duty vehicles multiple signals are available to estimate the vehicle's kinematics, such as inertial, GPS and CAN linear and angular speed readings. These signals have different noise variance, bandwidth and sampling rate. In this paper we present a non-linear sensor fusion algorithm allowing asynchronous sampling and non-causal smoothing, and we apply it to study the accuracy improvements when incorporating CAN measurements to standard GPS+IMU kinematic estimation, as well as the robustness against missing data. Our results show that these asynchronous CAN+GPS+IMU sensor fusion is advantageous in low-speed manoeuvres and GPS-denial environments. Accuracy and robustness to missing data is of course improved with non-causal filtering. The proposed algorithm is based on Extended Kalman Filter and Smoother with exponential discretization of continuous-time stochastic differential equations, in order to process measurements at arbitrary time instants.**

Sensor fusion; Asynchronous sampled-data; Extended Kalman filter; RTS smoother; Multi-rate; Dynamic systems; Heavy-duty vehicles; CAN bus; SAE J1939

I. INTRODUCTION

This paper deals with data sensor fusion techniques of systems with data coming from industrial communications, such as Controller Area Network (CAN) bus, combined with measurements from several sensors such as GPS and IMU, as shown in Figure 1. The goal is to provide an accurate state estimate combining traditional fusion of IMU and GPS data with CAN data. Indeed, we can take into account vehicle's non-holonomicity considering that sources of information will come from wheels speeds to improve position drift-reset effect typically obtained with GPS-IMU sensor fusion estimation at low speeds [1].

Many works have benefited from vehicle's internal network, either in cars or heavy-duty vehicles. For instance, an integrated self-diagnosis system was proposed in [2]. A similar approach was done for monitoring and diagnostic of automobile smart and integrated control systems [3], [4]. The authors of [5] participated in the DARPA Grand Challenge and to control the vehicle had to access J1939 bus to measure internal vehicle variables. In [6] a modular controller for the IVECO ISG hybrid electric vehicle (HEV) was developed based on the SAE J1939 CAN bus, whilst some authors implemented the SAE 1939 protocol to access the information of the distributed control system of electric city buses [7], [8].

Controller Area Network (CAN) in heavy-duty vehicles use J1939 protocol, as defined by the Society of Automotive Engineers (SAE) [9], which is used in vehicle networks for trucks and buses, agriculture and forestry machinery (ISO 11783), truck-trailer connections, Diesel power-train applications, military vehicles (MiLCAN), fleet management systems, recreational vehicles, marine navigation systems (NMEA2000), etc.

A well-known method to fuse data measurements from different sensors is the Kalman filter (KF). In tracking and ego-motion estimation applications [10], [11], [12], this method updates the estimation of a state, usually containing position and orientation of an object, based on sensor measurements. Typically used variables in the state of these filters in vehicles are position, linear and angular velocities and accelerations [13], but there are situations where biases of sensors are also estimated to get a better overall estimation [14], [15], [16].

In tracking systems, fusion of IMU (Inertial Measurement Unit) and GPS (Global Positioning System) is a conventional solution for general purposes and more particularly for the estimation of land vehicles [17], [18], [19] or UAVs (unmanned areal vehicles) [20]. IMUs, typically provide the ability to estimate fast accelerations or angular rates, which are suitable to track movements with fast dynamics. Fusion of GPS provides long-term stability, which in the end is used to compensate offsets of IMUs measurements. In these applications, fusion techniques with multiple sample rates or even irregular (non-periodic) sampling instants becomes a relevant tool. The integration of the continuous-time variance equation gives rise to the sampled-data Kalman filter and Extended versions [11], [12], [21], [19] for causal applications, or the (Kalman/RTS) smoother [22], [23] for non-causal ones; incorporating model's Jacobians would give rise to the extended versions for nonlinear cases. The advantage of non-causal smoothing techniques is that they can be used for data analysis, i.e. for vehicle modelling and identification; of course, in real-time applications for tracking and control, causal filters would be needed.

The main contribution of the paper is the use of a non-conventional sampling data fusion using an asynchronous smoother to fuse data gathered from different sources, which provides signal reconstruction at any arbitrary instant as the underlying algorithm is continuous-time. Our results, on an urban bus vehicle, show that the proposed experimental data gathering and asynchronous smoothing algorithm allow better performance at low-speeds than synchronous filtering

[1]V. Girbés-Juan is with Departament d'Enginyeria Electrònica, Universitat de València, Av. Universitat s/n, 46100, Burjassot (Spain) vicent.girbes@uv.es

[2]D. Hernández and A. Sala are with Instituto de Automática e Informática Industrial (AI2), Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia (Spain) daherfe2@etsid.upv.es/asala@isa.upv.es

[3]L. Armesto and J.F. Dols are with Instituto de Diseño y Fabricación, Universitat Politècnica de València, Camino de Vera s/n, 46022, Valencia (Spain) larmesto@idf.upv.es/jdols@mcm.upv.es

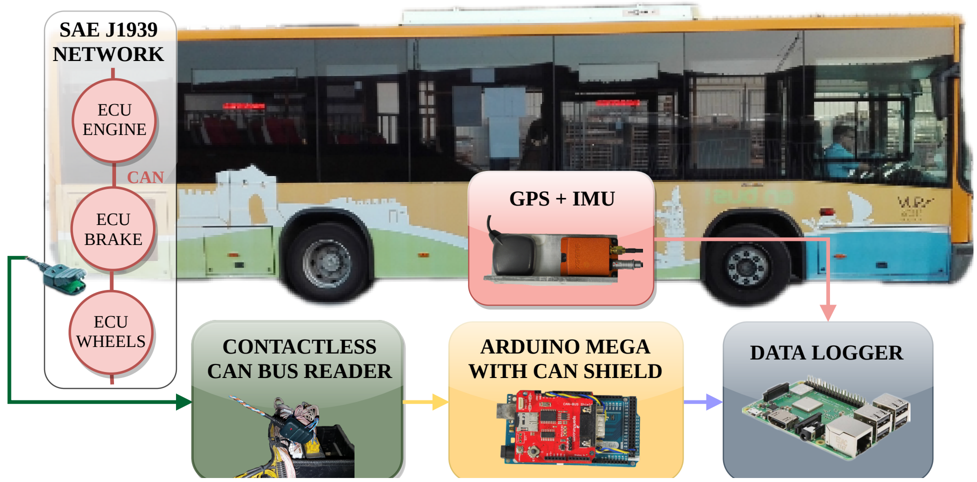


Fig. 1. Schematic representation of sensors and embedded devices used on the vehicle and their connections. The internal data of the sensors of the vehicle is accessed via CAN Network (SAE J1939 Protocol) using a contactless reader and processed by a micro-controller that sends it to an embedded PC. The external wireless sensors transmit data to the main computer for data logging.

approaches in terms of accuracy and robustness even when data samples are missing. We also provide insights on the convenience for fusing data from CAN data such as wheel's velocity and IMU and GPS data for a robust estimation even with significant losses of GPS data. This has been shown most relevant when using on-line (causal) estimation methods such as EKF. In a our previous work [24], the basic data-acquisition setup was presented, as well as the results using a regular-sampling causal KF for subsequent identification of some dynamic parameters.

The paper is organised as follows. Section II introduces some preliminaries about sampled-data systems and sensor fusion The problem statement is explained in Section III. The proposed algorithm for multi-rate asynchronous data fusion is explained in Section IV, whereas the setup used for the experimentation is defined in Section V. Section VI shows the main results obtained from the driving tests carried out to evaluate the proposed algorithm. Finally, the paper ends with a discussion and a summary of the main contributions of the paper in Section VII.

II. Preliminaries

A. Sampled-data systems and Sensor fusion and Smoothing

One of the most common techniques for state estimation of non-linear discrete-time dynamic systems is the Extended Kalman filter (EKF) [25], [18] or recently Unscented Kalman filter (UKF) [26], [27]. Kalman filter gives a robust, optimal, recursive state estimation to fuse redundant information from different sensors. This can be improved when using non-causal smoothing techniques such as Extended Rauch-Tung-Striebel (ERTS) smoother [22], [23], which uses an EKF and backward smoothing to produce quasi-optimal state trajectory estimation in optimal control settings.

Many real systems sensors provide data at different sampling periods due to the nature of their sensing technology or due to limitations of data transmission channels [28], [19], [29]. There are situations in which we can assume that there is a periodicity between all sampling rates and, as a consequence, we can work with periodic sampled-data systems. However, there are other situations in which measurements are asynchronous and, thus, state estimation needs to be performed based on available measurements [30]. The possibly irregular separation between sampling instants will be denoted as δt.

Let $\dot{x} = f(x, \epsilon_w)$ be a non-linear stochastic dynamic system and let $y = h(x) + \epsilon_v$ be the output equation, being $x \in \mathbb{R}^{n_x}$ the system state and $y \in \mathbb{R}^{n_y}$ the measurement vector; ϵ_w is a Gaussian process continuous-time noise with power spectral density Q_c (constant) and $\epsilon_v \sim N(0, R)$ is a measurement noise at sampling instants (sensor dynamics is assumed much faster than that of the measured signals). The non-linear functions $f(x, \epsilon_w) : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ and $h(x) : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$, describe the system dynamic and output measurement non-linear equations. The linearized equations for a given linearization point are

$$\dot{x} \approx A_c x + B_c \epsilon_w \tag{1}$$

$$y \approx H x + \epsilon_v \tag{2}$$

where A_c, B_c and H are the Jacobians for the state x, process noise w and output measurement y, respectively, which can be computed as:

$$A_c := \frac{\partial f(x, \epsilon_w)}{\partial x} \tag{3}$$

$$B_c := \frac{\partial f(x, \epsilon_w)}{\partial \epsilon_w} \tag{4}$$

$$H := \frac{\partial h(x)}{\partial x} \tag{5}$$

From the above continuous-time linear stochastic process, we can obtain a time-varying discretization as follows. For a given state x_t, noise ϵ_w and inter-sample time δt, the approximate discretized state equation (for the state expectation) can be computed as in [31]:

$$x_{t+\delta t} = \Psi_1, \text{ with } \Psi = \begin{bmatrix} \Psi_1 \\ \Psi_2 \end{bmatrix} = e^{\begin{bmatrix} A_c & I \\ 0 & 0 \end{bmatrix} \delta t} \cdot \begin{bmatrix} x_t \\ f(x_t, \epsilon_w) - A_c x_t \end{bmatrix} \tag{6}$$

and the discretization of the evolution of the state covariance can be done with [32]

$$A := e^{A_c \delta t} \tag{7}$$

$$Q := A_t \Phi_{12} \tag{8}$$

with $\Phi = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ 0 & \Phi_{22} \end{bmatrix} = e^{\begin{bmatrix} -A_c & B_c Q_c B_c^T \\ 0 & A_c^T \end{bmatrix} \delta t}$ and δt the sampling period.

Also, it seems relevant to use a proper discretization of nonlinear stochastic processes using Van Loan method [32] to reduce discretization errors typically introduced by Euler discretization if the sampling period is relatively large compared to signal dynamics. This might be particularly relevant when using Gaussian filters such as EKF and ERTS to keep linearization errors low and to avoid more accurate, but computationally demanding, approaches such as Runge-Kutta. In this way, if non-linearities are smooth so keeping Gaussian assumptions is reasonable, a suitable combination of the above equation (6) propagating the mean, and expressions (8) propagating the variance, can be used to build a causal extended Kalman filter or, for off-line data, an extended Rauch-Tung-Striebel smoother (ERTS), and Model-Predictive Approaches [33], [34], [23], [35] setup for irregularly-sampled systems, later discussed in Algorithm 1. This assumption essentially holds if the update time instants are reasonably close enough compared to the system dynamics.

III. Problem statement

Based on the previous preliminaries, this paper will provide an integrated solution for sensor fusion that combines causal (EKF filtering) and non-causal (ERTS smoothing) for nonlinear sampled data systems with irregular sampling period.

Indeed, due to technological limitations of some sensors, but mostly due to CAN bus priority-based messaging policy, data is not generally available at fixed single sampling period. Thus, it is more realistic to assume that data might be read asynchronously and, in most cases, it is not produced at the desired frequency for a good system identification, control design or system identification. Therefore, we need to considering as part of the filtering/smoothing process of signals the irregular sampling nature of many signals.

In addition to this, we would like to study the advantages of combining CAN, GPS and IMU data in urban scenarios with buses with a non-invasive logging system as the one proposed in Figure 1. Having access to such information in low-speed movements might be crucial to accurately estimate the actual motion of a bus. We intend to analyse performance on incorrect state initialisation and robustness under GPS data missing issues as a consequence of poor signal reception or GPS-denied areas (such as tunnels); consequences of GPS errors coming from multi-path signal trajectories typically obtained in urban scenarios. Combining GPS with IMU and speed data obtain from CAN will be useful to determine which information is more relevant and accurate for a candidate application.

IV. Asyncrhonous Extended Rauch-Tung Striebel Smoother

Let us introduce an asynchronous Extended Rauch-Tung-Striebel smoother (AERTS) algorithm for sampled data systems, which combines a continuous-discrete linearized Extended Kalman Filter (EKF) and a discrete linearized Extended Rauch-Tung-Striebel smoother (ERTS).

In the first stage, the AERTS algorithm performs a continuous in time prediction and a discrete measurement update, based on a linearized EKF. Due to technological limitations of CAN-based measurements, data is assumed to be measured asynchronously, which implies that the linearized EKF needs to integrate state predictions over the time until a new measurement data is available. When this occurs state updates are carried out only with available measurements at that time instant. In the following this is referred as Asynchronous EKF.

In a second stage of the algorithm, the estimated state coming from the Asynchronous EKF is smoothed to provide better estimations, given that ERTS is a non-causal filter that runs backward in time to provide estimations with data information from future time instants.

The AERTS algorithm is described, in pseudo-code, as Algorithm 1 on next page. In order to execute the referred algorithm, we need:

- A dataset with the following information: $\mathcal{D} := \{(y_{t1}, s_1, t_1), (y_{t2}, s_2, t_2), \cdot\}$, being y_{ti} the sensor measurement value, s_i the sensor number and t_i the timestamp when the measurement was acquired. Simultaneous measurements will be represented by two different triplets with coincident time, with no loss of generality.
- Model non-linearities f and h plus process and measurement noise parameters Q_c and R.
- An initial *a priori* state estimate $\mathcal{X} := \{x_{ini}, P_{ini}, t_1\}$ with the initial mean, variance and timestamp. Under no initial information P_{ini} is usually chosen a large diagonal matrix; in such a case, when $P_{ini} \to \infty$ the value of x_{ini} is irrelevant, usually chosen to be zero.

In the algorithm, contrarily to Q_c, matrix R is not discretised from a continuous-time stochastic differential equation as we will assume that the dynamics of the sensor is very fast, so the *correlation time* of the additive sensor noise will be negligible, smaller than any of the inter-sample intervals in the dataset $\mathcal{D}$.

As measurements are not always available, in the following, h_t^s will denote the expected value of the measurement given by sensor s; likewise, given any matrix M, M^s will denote the s-th row, and M^{ss} will refer to the s-th diagonal element. Thus, the measurement noise covariance of sensor s will be denoted as R^{ss}.

The detail of the algorithm steps are as follows. First, lines 3 to 14 implement the Asynchronous extended Kalman filter. In particular, lines 5-8 perform the asynchronous prediction step, where we compute mean using equations (3)-(6) and forward state covariance prediction using (7). It is worth mentioning that state covariance update is implemented using lines 10-12, which requires a proper selection of the available measurements.

Algorithm 1 AERTS

1: **function** $\mathcal{Z}$=AERTS($\mathcal{D}$,$\mathcal{X}$)
Inputs: $\mathcal{D}$: a set of samples $\{(y_{t_1}, s_1, t_1), (y_{t_2}, s_2, t_2) \dots\}$ and $\mathcal{X}$: initial estimate with $\{x_{ini}, P_{ini}, t_1\}$.
[Initialisation]
2: $\hat{x}_t \leftarrow x_{ini}$, $P_t \leftarrow P_{ini}$, $\bar{t} \leftarrow t_0$, $\mathcal{T} \leftarrow \{\}$
[Asynchronous EKF (forward in time)]
3: **for each** triplet $\{y_t, s, t\}$ in $\mathcal{D}$ **do**
[Sampling time]
4: $\delta t \leftarrow t - \bar{t}$ and $\bar{t} \leftarrow t$
[State and covariance prediction]
5: $A_c \leftarrow \left.\frac{\partial f(x,\epsilon_w)}{\partial x}\right|_{\substack{x=\hat{x}_t \\ \epsilon_w=0}}$, $B_c \leftarrow \left.\frac{\partial f(x,\epsilon_w)}{\partial \epsilon_w}\right|_{\substack{x=\hat{x}_t \\ \epsilon_w=0}}$
6: $\hat{x}_t \leftarrow \Psi_1$, with $\Psi = \begin{bmatrix} \Psi_1 \\ \Psi_2 \end{bmatrix} = e^{\begin{bmatrix} A_c & I \\ 0 & 0 \end{bmatrix}\delta t} \cdot \begin{bmatrix} \hat{x}_t \\ f(\hat{x}_t,0)-A_c\hat{x}_t \end{bmatrix}$
7: $A_t \leftarrow \Phi_{22}$, $Q_t \leftarrow \Phi_{22}^T\Phi_{12}$, with $\Phi := \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ 0 & \Phi_{22} \end{bmatrix} = e^{\begin{bmatrix} -A_c^T & B_cQ_cB_c^T \\ 0 & A_c \end{bmatrix}\delta t}$
8: $P_t \leftarrow A_tP_tA_t^T + Q_t$
[State and covariance update]
9: $H_t^s \leftarrow \left.\frac{\partial h^s(x)}{\partial x}\right|_{x=\hat{x}_t}$
10: $K_t \leftarrow P_t(H_t^s)^T(H_t^sP_t(H_t^s)^T + R^{ss})^{-1}$
11: $\hat{x}_t \leftarrow \hat{x}_t + K_t(y_t - h_t^s)$
12: $P_t \leftarrow (I - K_tH_t)P_t$
13: Prepend $\{A_t, Q_t, \hat{x}_t, P_t, t\}$ to $\mathcal{T}$
14: **end for**
[RTS Smoother (backward in time)]
15: $\hat{z}_t \leftarrow \hat{x}_t$, $\mathcal{Z} \leftarrow \{z_t\}$
16: **for each** tuple $\{A_t, Q_t, \hat{x}_t, P_t, t\}$ in $\mathcal{T}$ **do**
17: $L_t \leftarrow P_tA_t^T(A_tP_tA_t^T + Q_t)^{-1}$
18: $\hat{z}_t \leftarrow \hat{x}_t + L_t(\hat{z}_t - f(\hat{x}_t, 0))$
19: Prepend z_t to $\mathcal{Z}$
20: **end for**
21: **end function**

As a result of the prediction and update steps, we store estimated state mean and covariance, as well as state linearization Jacobian and system noise covariance, required by the ERTS smoother (implemented on lines 16-20).

It is important to remark that the proposed algorithm is designed for data coming asynchronously, but it allows to estimate the state for any arbitrary time instant. This can be carried out by running Algorithm 1 combining real measurements time instants and arbitrary time instants, without executing lines 15-20 if there is no real measurement at that time. This has the advantage of reduced linearization errors due to smaller sampling times at a higher computational cost. For instance, we can provide state estimates at regularly spaced time instants at any desired sampling frequency, while real asynchronous measurements will be interleaved within such regular samplings. Even if the proposed algorithm uses samples at given timestamps, it can provide estimations at any higher sampling rate, allowing us reconstruct the smoothed signal/state at any arbitrary time instant, regardless of its coincidence (or lack thereof) with any measurement.

A. AERTS vehicle data

In this section, we explain data used in AERTS in the context of heavy-duty vehicles, as well as vehicle's kinematic and measurement models. In particular, the vehicle's geographic coordinates have been measured from a GPS, whilst its orientation, angular velocity and linear acceleration have been sensed using an IMU. In addition, wheels' velocities and linear acceleration are obtained from CAN bus and a second IMU has been attached to the steering wheel to measure its angular position. Note that IMUs can be affected by bias due to calibration errors and other external effects, such as magnetic field perturbations, so an offset and its derivative has to be included in the estimation, because there is also a drift produced by the integration of gyroscope and accelerometer's biases.

It is interesting to remark that, in our application, the IMU is the sensor with the fastest sampling rate at a regular frequency of 100 Hz. However, GPS provides new data at approximately 1 Hz, whilst CAN data are measured at irregular time instants as new messages come depending on the ECUs manufacturer publishing frequencies and message priorities, according to the SAE J1939 standard.

In the SAE J1939 protocol each message has its own range of transmission rates, but the priority of each group may change and affect that rate. Depending on the level of priority transmission rates vary from 10 ms, for high-priority nodes, to 1 second, for low-priority nodes. Messages with higher priority will gain bus access within shortest time even when the bus load is high due to the number of lower priority messages. The J1939 message format consists mainly in two different fields, the identification (ID) field and the data field. The ID field controls the message priority and includes the Parameter Group Number (PGN) field, which identifies the message type. In particular, the messages used are those related to wheels' velocity and linear acceleration, obtained from PGNs 61443, 65215 and 61449 (see [9] for details). The angular position of the steering wheel could have been obtained from a specific J1939 message (PGN 61449), but it was not implemented by the ECU manufacturer.

B. AERTS vehicle model

However, it was estimated from measurements of front wheels velocity published in the CAN network, assuming an Ackermann steering geometry [36].

The Ackermann mechanical configuration implies that the vehicle's front wheels turn at different speeds in order to trace out circles of different radii. In such a configuration, there is an equivalent tricycle configuration with only one front wheel, whose orientation ϕ_w, known as the Ackermann angle, is the average angle of the front wheels and can be computed indirectly from their velocity using the following formula

$$\phi_w = 0.5 \arcsin \frac{4L(v_R - v_L)}{W(v_R + v_L)} \tag{9}$$

where L is the distance between front and rear axes; W is the separation between left and right wheels; and the speeds of right and left wheels are defined by v_R and v_L, respectively. The reader is referred to [36] for details.

To compute the steering wheel orientation ρ^{st}, the ratio between the steering wheel angle and the Ackermann angle (front wheels average orientation) must be applied so that $\rho^{st} = f_s \phi^w$, where $f_s = 17$ is the steer factor for the bus used in the experimentation, which is obtained from manufacturer specifications.

Finally, vehicle's angular velocity can also be obtained from CAN messages, using the linear velocity v and wheel's orientation ϕ_w computed from equation (9):

$$\omega^{CAN} = \frac{v \tan \phi_w}{L} \tag{10}$$

In the sequel, super-index GPS corresponds to data coming from GPS sensor, super-index IMU refers to IMU sensor, and super-index CAN is related to all available measurements from the contactless CAN J1939 reader (see Figure 1 for details). As an example, v^{CAN} will denote linear velocity coming from CAN bus, whereas $\{p_x^{GPS}, p_y^{GPS}\}$ will refer to Cartesian position read from the GPS coordinates.

In order to estimate vehicle kinematics in a planar surface, the state vector is defined as $x = [p_x \ p_y \ \theta \ \omega \ v \ a \ o_\omega^{IMU} o_a^{IMU} \ o_\omega^{CAN}]^T$, which includes vehicle's position p_x and p_y; heading or orientation θ; angular velocity ω; linear velocity v and linear acceleration a, angular velocity IMU offset o_ω^{IMU}, linear acceleration IMU offset o_a^{IMU} and angular velocity CAN offset o_ω^{CAN}. Offsets have been considered to compensate systematic and non-systematic errors such as IMU misalignment measurements and wheel's speed measurements from CAN data (such as incorrect tyre pressure, wheels misalignment, etc.). The output vector includes data from GPS, IMU and CAN as follows: $y = [p_x^{GPS} \ p_y^{GPS} \ \theta^{IMU} \ \omega^{IMU} \ a^{IMU} \ \omega^{CAN} \ v^{CAN} \ a^{CAN}]^T$. On the other hand, process noise vector is assumed to be $\epsilon_w = [\epsilon_{p_x} \ \epsilon_{p_y} \ \epsilon_\omega \ \epsilon_a \ \epsilon_{\omega_o}^{IMU} \ \epsilon_{a_o}^{IMU} \ \epsilon_{\omega_o}^{CAN}]^T$; measurement noise vector is $\epsilon_v = [\epsilon_{p_x}^{GPS} \ \epsilon_{p_y}^{GPS} \ \epsilon_\theta^{IMU} \ \epsilon_\omega^{IMU} \ \epsilon_a^{IMU} \ \epsilon_\omega^{CAN} \ \epsilon_v^{CAN} \ \epsilon_a^{CAN}]^T$; and system dynamics and output measurement equation are described as:

$$\dot{x} = f(x, \epsilon_w) := A_c(x)x + B_c \epsilon_w \tag{11}$$

$$y = h(x, \epsilon_v) := Hx + \epsilon_v \tag{12}$$

where A_c, B_c and H are the Jacobians for the state x, process noise w and output measurement y, respectively, computed as in Equations (3), (4) and (5)

$$A_c = \begin{bmatrix} 0 & 0 & 0 & 0 & \cos(\theta_t) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sin(\theta_t) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, B_c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

V. Experimentation Setup

A. Materials

The urban bus shown in Figure 1 was used for the experimentation. It is a MAN 14250 HOCL-NL with the following kinematic specifications: distance between axes $L = 5875$ mm and wheels track width $W = 2550$ mm. Figure 1 also shows sensors and other electronic devices used for the data acquisition system.

To track the position of the bus and other kinematic variables, such as orientation, angular velocity and linear acceleration, an Xsens MTi-G-710 GNSS inertial measurement unit with GPS was mounted on vehicle's centre of rotation. The IMU incorporates the following components: 3 axes magnetometer (full range ± 8 Gauss, RMS noise 0.5 mG), 3 axes gyroscope (full range $\pm 450^o$/s, bias error 0.2^o/s), 3 axes accelerometer (full range ± 200 m/s^2, bias error 0.05 m/s^2) and barometer (full range $30 - 110$ kPa, RMS noise 3.6 Pa). The dynamic accuracy of the orientation is 0.3^o (pitch/roll) and 0.8^o (yaw). Regarding the GPS, the horizontal accuracy is 1 m (Cartesian coordinates x/y) and the vertical accuracy is 2 m (z coordinate).

A contactless CAN connector was used to safely read data from vehicle CAN bus (see Figure 1). Using this device, data reading is non-invasive as it occurs without electrical connection and without damaging CAN wires. It works in "listen" mode only, i.e. it does not change original J1939 messages and does not send any signals to CAN bus.

In order to collect all SAE J1939 messages in vehicle's CAN network and send them to the data logger through a serial protocol, a micro-controller Arduino Mega 2560 with a CAN-BUS shield was used as a sniffer (see Figure 1). Finally, a Raspberry PI 3 Model B with an embedded Linux was used as the main computer for data logging.

B. Driving tests

Two driving tests have been carried out in order to collect data, whose short description is included next:

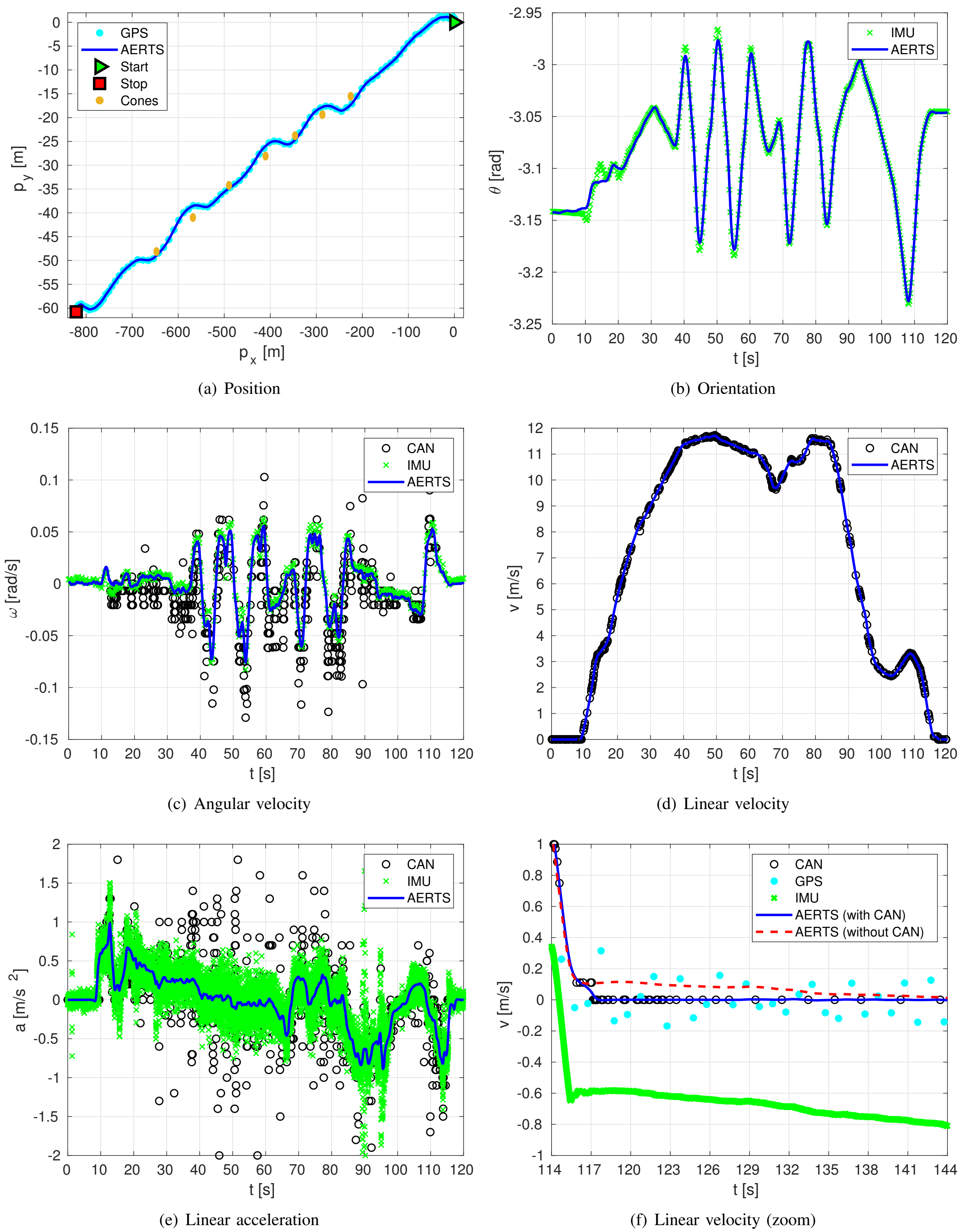
- **Slalom**: Several cones were placed in a straight line, separated 80 m distance. The test started with the bus completely stopped. The driver sped up until the vehicle was at the specified linear velocity of 40 km/h (≈ 11 m/s). While keeping a constant speed, the cones were alternatively avoided in a slalom movement. Finally, the driver braked until the bus was completely stopped.
- **Urban driving**: This driving test was conducted in an urban scenario, where different manoeuvres were combined (straight driving, corners, roundabouts, etc.) and the bus driver faced everyday situations. The overall duration of data recording was around 400 s.

VI. Results

To estimate vehicle's kinematics, considering the measurement uncertainties, covariance matrices are initialised as follows in all driving tests:

$$Q_c = \mathrm{diag}(\{0.25, 0.25, 0.01, 0.0025, 10^{-10}, 10^{-10}, 10^{-10}\}),$$
$$R = \mathrm{diag}(\{0.25, 0.25, 0.01, 0.04, 0.0025, 0.25, 0.0001, 1\}).$$

*1) **Slalom driving**:* Some linear and angular kinematic variables from the slalom driving test have been reconstructed at high-rate using the proposed AERTS smoother (Figure 2).



(a) Position

(b) Orientation

(c) Angular velocity

(d) Linear velocity

(e) Linear acceleration

(f) Linear velocity (zoom)

Fig. 2. (a-e): Estimation at high-rate of vehicle's linear and angular kinematic variables from the slalom driving test; (f): Detail of the last 30 seconds of vehicle's linear velocity after braking to analyse the effect of making the data fusion without including information from CAN bus (dashed red line) and using the CAN messages of velocity and acceleration (blue line).

The data fusion algorithm combines signals coming asynchronously at different sampling rates from IMU (orientation, angular velocity, linear acceleration), GPS (position) and CAN messages (linear velocity, linear acceleration and wheels velocity).

a) Linear kinematics: As observed in Figure 2(a), the position estimation does not have any bias, as expected, because GPS measures do not have it either. AERTS allows to have position estimates at 100 Hz sampling rate even if the original GPS rate is 1Hz.

Figures 2(d) and 2(e) show linear velocity and acceleration, respectively. It is well known that odometry based on encoders or tachographs is not reliable, as its accuracy depends on many factors, such as wheels radius (which depends on tires pressure, as they can be more or less inflated), wheels slippage (which vary in different surfaces), among other things. But, although wheels velocity and linear acceleration coming from the CAN network might have an offset at certain speed (due to wheels misalignment, different wheels pressure, or discrepancies between measures of tachometers or encoders), when the vehicle is stopped that offset is zero. So, estimation errors of vehicle's linear velocity can be reduced by using CAN bus data, even when it comes very sparse and asynchronously.

In fact, the only actual measurement of linear velocity comes from vehicle's internal odometry, since the other two possible sources would be integration of linear acceleration coming from IMU (green crosses) and derivation of travelled distance computed from GPS position (cyan dots). However, the former can have a high bias after a while, whilst the latter is very noisy due to inaccurate and jumpy readings from satellites.

b) Braking: As shown in Figure 2(f), when the vehicle brakes the proposed algorithm estimates that it is not completely stopped when it actually is, if CAN data is not used (dashed red line). It takes about 27 s for the proposed algorithm to estimate that the vehicle is not moving. However, the AERTS estimation using CAN data (blue line) is much more accurate, as it returns zero velocity thanks to the wheels' velocity reading from CAN bus messages. So, among other advantages, using the information from CAN bus can be very useful to identify properly when the vehicle is not moving at all and, therefore, it can help in a dynamic identification and process modelling, very useful for crash analysis [37], driving assistance [38] and self-driving vehicles [39], [40].

In fact, GPS was outputting non-constant positions originating a velocity offset, as can be seen in Figure 2(f). The same happens to the IMU, that returns certain value of acceleration producing non-zero linear velocity.

c) Angular kinematics: In Figures 2(b) and 2(c), it can be observed that the orientation measured by the IMU in slalom driving test is pretty accurate, but in other situations it could be biased due to, for instance, magnetic disturbances. In fact, if we take a look at Figure 2(c), the angular velocity from the IMU has a constant bias. Furthermore, the CAN measurement also has a negative offset, which is even bigger in magnitude. Nevertheless, thanks to the data fusion of these measures together with GPS position, the estimation is improved and the biases compensated.

d) Instrument offset identification: Based on the results obtained from the experimentation, we identified the average bias of the data coming from the CAN network (angular velocity offset $o_\omega^{CAN} = -0.007$ rad/s) and signals from the IMU sensor (gyroscope offset $o_\omega^{IMU} = 0.003$ rad/s and accelerometer offset $o_a^{IMU} = -0.015$ m/s^2). Even though these biases might seem low values, they are integrated to obtain orientation and linear velocity signals, so the error is accumulated and, in a few seconds, the drift can be very high unless such offsets are compensated with sensor fusion.

*2) **Urban driving:*** Once the AERTS data fusion algorithm is validated, it has been applied in a long driving test in an urban scenario to show that it is working properly (see Figure 3). The position on map is depicted in Figure 3(a), whilst Figure 3(b) represents some kinematic variables (linear velocity, linear acceleration and angular velocity) for this experiment. It can be observed that the AERTS algorithm is able to estimate correctly linear and angular velocities, being the acceleration noise reduced considerably and all sensor biases compensated.

The urban driving test has been also used to analyse the robustness and the accuracy of the proposed sensor fusion setting. Two different studies have been performed: effect of incorrect state space initialisation and robustness (it can be seen as a kidnapped robot problem [41] after a long-term of data-missing in GPS-denied environments). Some aspects considered for such studies are described below.

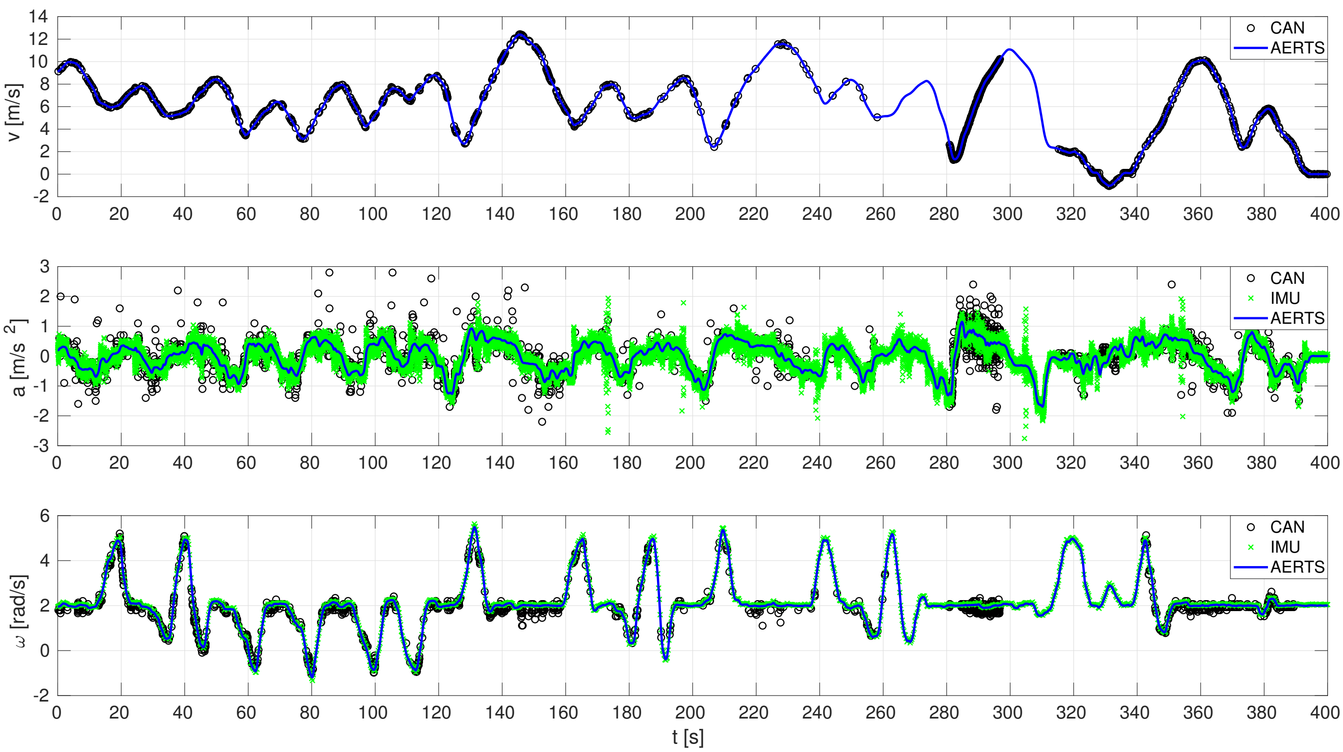
a) Initial performance of AEKF vs. AERTS: Let us evaluate the performance in the first instants of the experiment with large initial variance (imprecise initial information). Without any *a priori* knowledge the first velocity estimations of the Asynchronous EKF without CAN are unreliable, as they are roughly a numerical differentiation during the first handful of samples; on the other hand CAN measurements provide an accurate AEKF estimation due to the direct speed measurements, as shown in Figure 4(a). The knowledge of future samples mitigates this initialization problem in the AERTS case, as observed in Figure 4(b).

b) Study of robustness under missing data: In the second study, to quantify the estimation robustness against GPS data missing, we analyze the error in the estimation of Cartesian position e_p with randomly missing certain percentage of GPS data (10%, 30%, 50% and 70%). EKF and AERTS data fusion algorithms are compared, in both cases with and without in-vehicle CAN bus information. The estimation at very high frequency (1 kHz) using the AERTS algorithm with all the available information (GPS, IMU and CAN data) is used as ground truth, i.e., later error figures will be computed as the difference wit the estimation with 100% available data.

Figure 5 shows the box-and-whisker plot of the position estimation error e_p with random GPS data missing. The red and green boxes correspond to estimations using EKF, whilst the magenta and blue whiskers diagrams represent estimations with AERTS sensor fusion algorithm. Cases where CAN data are not used are represented by red and magenta diagrams, whereas the green and blue boxes use the same amount of GPS data, but include CAN measurements.

The results clearly show that using CAN yields a much more robust estimate under missing GPS data conditions. In fact, both algorithms improve the estimation around 10% (or even more, for the case of AERTS with 10% of GPS data missing) just by plugging information from CAN bus into the algorithm. Figure 5 also illustrates the clear advantage of the non-causal smoothing in missing-data situations.

(a) Position on map



(b) Kinematic variables

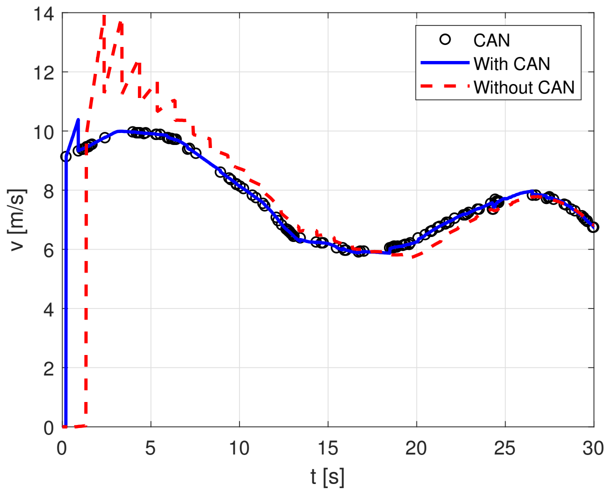
Fig. 3. Driving test in urban scenario: (a) trace of vehicles' position estimated with AERTS algorithm (blue line) with start point (green triangle) and stop point (red square); (b) vehicle's kinematic variables: linear velocity (top), linear acceleration (middle), and angular velocity (bottom).

VII. Conclusion

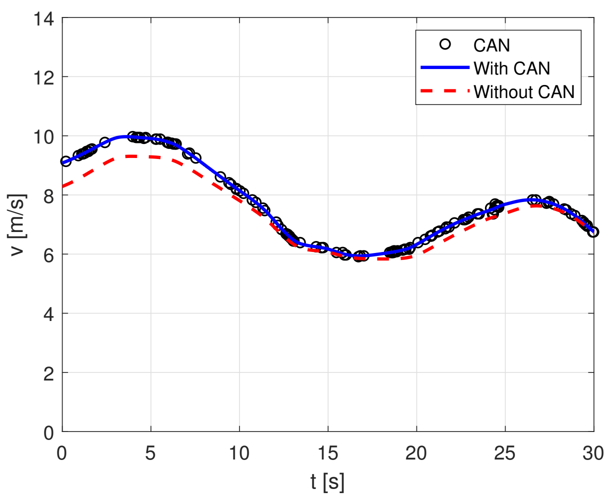
This paper has presented an asynchronous smoothing algorithm combining data from IMU, GPS and CAN data bus at arbitrary sampling rates. Based on the experimental results on an urban bus, it can be concluded that, as CAN messages provide accurate measurements of the linear velocity of the bus, fusion of the three sources is shown to be beneficial to estimate the velocity at low-speeds, in GPS-denial environments and to improve initial performance. In general, the proposal allows fusing the information from the odometry system attached to the wheels with the standard inertial plus GPS navigation sources to improve accuracy. The underlying continuous-time theory allows signal reconstruction at an arbitrary time point, implementing adaptive discretization using exponential matrices for mean and variance equations. The algorithm has been tested in a public urban bus to estimate vehicle kinematics, but it is of course valid in many other scenarios, where asynchronous data is required to be fused.

Acknowledgements

This research was funded by Spanish Ministry of Economy, grant DPI2016-81002-R, by Generalitat Valenciana, grant



(a) AEKF algorithm (prediction)



(b) AERTS algorithm (prediction+smoothing)

Fig. 4. Initial performance comparison with the absence of CAN data for causal/non-causal filters.

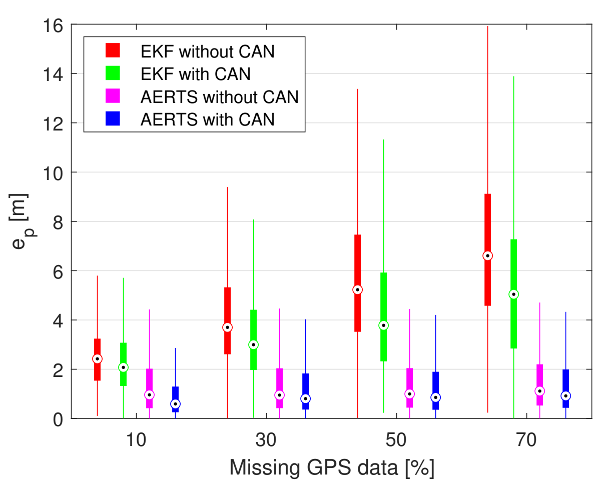


Fig. 5. Estimated position error e_p in the urban driving test for different percentages of missing GPS data.

APOSTD/2017/055, and by the European Union through the European Social Fund (ESF). The authors are also grateful to public transport operators Autos Vallduxense S.L. and EMT Valencia for their support to the project by lending their vehicles and drivers, and to Inst. de Diseño y Fabricación and SAFETRANS project (DPI2013-42302-R) for lending most of the electronic devices and sensors used in the experimentation.

References

[1] F. Cavallo, A. M. Sabatini, and V. Genovese, "A step toward gps/ins personal navigation systems: real-time assessment of gait by foot inertial sensing," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 1187–1191.

[2] Y. Jeong, S. Son, E. Jeong, and B. Lee, "An integrated self-diagnosis system for an autonomous vehicle based on an iot gateway and deep learning," *Applied Sciences*, vol. 8, no. 7, p. 1164, 2018.

[3] Yang Jiansen, Guo Konghui, Ding Haitao, Zhang Jianwei, and Xiang Bin, "The application of sae j1939 protocol in automobile smart and integrated control system," in *2010 Int. Conf. on Computer, Mechatronics, Control and Electronic Eng.*, vol. 3, Aug 2010, pp. 412–415.

[4] E. Türk and M. Challenger, "An android-based iot system for vehicle monitoring and diagnostic," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, May 2018, pp. 1–4.

[5] U. Ozguner, K. A. Redmill, and A. Broggi, "Team terramax and the darpa grand challenge: a general overview," in *IEEE Intelligent Vehicles Symposium, 2004*, June 2004, pp. 232–237.

[6] Y. Li and X. Ji, "Controller design for isg hybrid electric vehicle based on sae j1939 protocol," in *2nd Int. Conf. on Computer Science and Electronics Engineering*. Atlantis Press, March 2013. [Online]. Available: https://doi.org/10.2991/iccsee.2013.647

[7] Wang Dafang, Nan Jinrui, and Sun Fengchun, "The application of can communication in distributed control system of electric city bus," in *2008 IEEE Vehicle Power and Propulsion Conf.*, Sep. 2008, pp. 1–4.

[8] J. Hu, G. Li, X. Yu, and S. Liu, "Design and application of sae j1939 communication database in city-bus information integrated control system development," in *2007 Int. Conf. on Mechatronics and Automation*, Aug 2007, pp. 3429–3434.

[9] W. Voss, *A comprehensible guide to J1939*. Copperhill Technologies Corporation, 2008.

[10] X. Meng, H. Wang, and B. Liu, "A robust vehicle localization approach based on gnss/imu/dmi/lidar sensor fusion for autonomous vehicles," *Sensors*, vol. 17, no. 9, p. 2140, 2017.

[11] J. Hu, Z. Wu, X. Qin, H. Geng, and Z. Gao, "An extended Kalman filter and back propagation neural network algorithm positioning method based on anti-lock brake sensor and global navigation satellite system information," *Sensors (Switzerland)*, vol. 18, no. 9, pp. 1–15, 2018.

[12] K. Kim, J. Lee, and C. Park, "Adaptive Two-Stage EKF for INS-GPS Loosely Coupled System with Unknown Fault Bias," *Journal of Global Positioning Systems*, vol. 5, no. 1&2, pp. 62–69, 2006.

[13] B. Gersdorf and U. Frese, "A Kalman Filter for Odometry using a Wheel Mounted Inertial Sensor," Tech. Rep., 2013.

[14] M. Spangenberg, V. Calmettes, and J. Y. Tourneret, "Fusion of GPS, INS and odometric data for automotive navigation," in *European Signal Processing Conference*, no. Eusipco, 2007, pp. 886–890.

[15] S. Zihajehzadeh, D. Loh, T. J. Lee, R. Hoskinson, and E. J. Park, "A cascaded Kalman filter-based GPS/MEMS-IMU integration for sports applications," *Measurement: Journal of the International Measurement Confederation*, vol. 73, pp. 200–210, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.measurement.2015.05.023

[16] B. S. Cho, W. sung Moon, W. J. Seo, and K. R. Baek, "A dead reckoning localization system for mobile robots using inertial sensors and wheel revolution encoding," *Journal of Mechanical Science and Technology*, vol. 25, no. 11, pp. 2907–2917, 2011.

[17] S. Sukkarieh, E. M. Nebot, and H. F. Durrant-Whyte, "A high integrity IMU/GPS navigation loop for autonomous land vehicle applications," *IEEE Trans. Robotics and Automation*, vol. 15, no. 3, pp. 572–578, 1999.

[18] K. Chiang, G. Tsai, H. Chu, and N. El-Sheimy, "Performance enhancement of ins/gnss/refreshed-slam integration for acceptable lane-level navigation accuracy," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 2463–2476, 2020.

[19] L. Armesto, J. Tornero, and M. Vincze, "On multi-rate fusion for non-linear sampled-data systems: Application to a 6d tracking system," *Robotics and Autonomous Systems*, vol. 56, no. 8, pp. 706–715, 2008.

[20] J.-H. Kim, S. Sukkarieh, and S. Wishart, "Real-time navigation, guidance, and control of a uav using low-cost sensors," in *Field and Service Robotics*. Springer, 2003, pp. 299–309.
[21] C. T. S. H. E. Rauch, F. Tung, "Maximum likelihood estimates of linear dynamic systems," pp. 1445–1450, 1965.
[22] M. Zima, L. Armesto, V. Girbés, A. Sala, and V. Smidl, "Extended rauch-tung-striebel controller," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, 2013, pp. 2900–2905.
[23] L. Armesto, V. Girbés, A. Sala, M. Zima, and V. Šmídl, "Duality-based nonlinear quadratic control: Application to mobile robot trajectory-following," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1494–1504, 2015.
[24] V. Girbés, D. Hernández, L. Armesto, J. F. Dols, and A. Sala, "Drive force and longitudinal dynamics estimation in heavy-duty vehicles," *Sensors*, vol. 19, no. 16, August 2019.
[25] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robot. Automation*, vol. 17, no. 3, pp. 229–241, 2001.
[26] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, Oct 2000, pp. 153–158.
[27] W. Zhang, Z. Wang, C. Zou, L. Drugge, and M. Nybacka, "Advanced vehicle state monitoring: Evaluating moving horizon estimators and unscented kalman filter," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5430–5442, 2019.
[28] J. Tornero and L. Armesto, "A general formulation for generating multi-rate models," in *Proceedings of the American Control Conference*, vol. 2, 2003, pp. 1146–1151.
[29] H. Tan, B. Shen, Y. Liu, A. Alsaedi, and B. Ahmad, "Event-triggered multi-rate fusion estimation for uncertain system with stochastic nonlinearities and colored measurement noises," *Information Fusion*, vol. 36, pp. 313–320, 2017.
[30] L. Armesto, G. Ippoliti, S. Longhi, and J. Tornero, "Probabilistic self-localization and mapping-an asynchronous multirate approach," *IEEE robotics & automation magazine*, vol. 15, no. 2, pp. 77–88, 2008.
[31] N. Wahlström, P. Axelsson, and F. Gustafsson, "Discretizing stochastic dynamical systems using lyapunov equations," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 3726 – 3731, 2014, 19th IFAC World Congress. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667016421841
[32] C. Van Loan, "Computing integrals involving the matrix exponential," *IEEE Trans. Automatic Control*, vol. 23, no. 3, pp. 395–404, 1978.
[33] L. A. Johnston and V. Krishnamurthy, "Derivation of a sawtooth iterated extended kalman smoother via the aecm algorithm," *IEEE Transactions on Signal Processing*, vol. 49, no. 9, pp. 1899–1909, 2001.
[34] S. SÄrkkÄ, "Unscented rauch–tung–striebel smoother," *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 845–849, 2008.
[35] Y. Hu, H. Su, L. Zhang, S. Miao, G. Chen, and A. Knoll, "Nonlinear model predictive control for mobile robot using varying-parameter convergent differential neural network," *Robotics*, vol. 8, no. 3, p. 64, 2019.
[36] W. Chung and K. Iagnemma, *Wheeled Robots*. Cham: Springer International Publishing, 2016, pp. 575–594. [Online]. Available: https://doi.org/10.1007/978-3-319-32552-1_24
[37] Ching-Yao Chan, "On the detection of vehicular crashes-system characteristics and architecture," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 1, pp. 180–193, 2002.
[38] V. Girbés, L. Armesto, J. Dols, and J. Tornero, "An active safety system for low-speed bus braking assistance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 2, pp. 377–387, 2017.
[39] H. Wang, Y. Huang, A. Khajepour, Y. Zhang, Y. Rasekhipour, and D. Cao, "Crash mitigation in motion planning for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3313–3323, 2019.
[40] Y. Chen, C. Hu, and J. Wang, "Human-centered trajectory tracking control for autonomous vehicles with driver cut-in behavior prediction," *IEEE Trans. Vehicular Technology*, vol. 68, no. 9, pp. 8461–8471, 2019.
[41] H. M. Choset, S. Hutchinson, K. M. Lynch, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

Vicent Girbés-Juan received the B.Eng. and M.Sc. in Industrial Electronics and Control in 2009 and 2011, respectively; and the PhD. in Automation, Robotics and Computer Science in 2016, from the Universitat Politècnica de València (UPV), Spain. From 2009 to 2019 he worked in the Robotics and Automation Research Group of the Design and Manufacturing Institute at UPV. He has been visiting researcher at University of Manchester (6 months) and Imperial College London (12 months), both in the United Kingdom. Since October 2019 he is Assistant Professor in the Department of Electronic Engineering at Universitat de València, Spain. His fields of interest are collaborative robotics, human-robot interaction, haptics, advanced driving assistance systems, machine learning, path planning and obstacle avoidance for AGVs and UAVs.

Leopoldo Armesto received the B.Sc. degree in Electronic Engineering, the M.Sc. degree in Control Systems Engineering, and the Ph.D. in Automation and Industrial Computer Science from the Universitat Politècnica de València (UPV), Spain, respectively. He is member of the Department of Systems Engineering and Control at the same University. His current research interests are Mobile Robotics, Optimal Control, Advanced Driving Assistance Systems, 3D Printing and Reinforcement Learning. He has supervised 3 Ph.D. theses and 6 final M.Sc. Projects. He has participated in 23 Research Projects (leading 8 of them) and many other contracts with industry. He has published 16 JCR journal papers and more than 60 conference papers (google h-index 15).

Daniel Hernández-Ferrándiz received the B.Eng. and M.Sc. in Industrial Electronics and Automatics in 2016 and 2018 from the Universitat Politècnica de València (UPV). From 2018 to 2019 he worked (7 months) as a researcher in the Robotics and Automation Research Group of the Automatics and Industrial Infomatics Institute (AI2) at UPV. Since September 2019 he is working as a researcher and studying computer vision and deep learning in the Universidad Rey Juan Carlos (URJC). Some of his current fields of interest are the following: collaborative and industrial robotics, computer vision and deep learning.

Juan Dols is Industrial Engineer (Mechanics) since 1987 and in 1996 the Ph.D. degree at Universitat Politècnica de València (UPV), Spain, where he is now teaching Automotive and Transportation Engineering at the Mechanical and Materials Engineering Department of UPV. Professor Dols is also Director of the UPV Automobile Laboratory (LAUPV). His research activity is based in the design and development of driving simulators, wheelchair tiedown and occupant restraint systems, active and passive safety in vehicles, accessible transportation engineering, driving aids design and development, accident reconstruction technologies and vehicle dynamics. He has conducted as main researcher and coordinator several national projects and international EU funded projects.

Antonio Sala received a Ph.D. in Control Engineering from Universitat Politècnica de València, Spain. Since 1993, he has been teaching in UPV where he currently is full Professor. He has supervised 11 Ph.D. theses and 30 final M.Sc. projects. His current research interests are optimal, fault-tolerant, robust and LPV control. He has published one book, co-edited another two ones, more than 75 journal papers and more than 130 congress papers (Google h-index: 28). Antonio Sala has served as Associate Editor of IEEE Trans. on Fuzzy Systems, Fuzzy Sets and Systems, and Rev. Iberoamericana de Automática e Informática Industrial.