Asynchronous Sensor Fusion of GPS, IMU and CAN-based Odometry 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 Measurement Unit (IMU), Global Positioning System (GPS) and linear and angular speed readings from wheel tachometers on the internal Controller Area Network (CAN). These signals have different noise variance, bandwidth and sampling rate (being the latter, possibly, irregular). In this paper we present a non-linear sensor fusion algorithm allowing asynchronous sampling and non-causal smoothing. It is applied to achieve accuracy improvements when incorporating odometry measurements from CAN bus to standard GPS+IMU kinematic estimation, as well as the robustness against missing data. Our results show that this asynchronous multi-sensor (GPS+IMU+CAN-based odometry) fusion is advantageous in low-speed manoeuvres, improving accuracy and robustness to missing data, thanks to 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; it can provide data to subsequent processing steps at arbitrary time instants, not necessarily coincident with the original measurement ones. Given the extra information available in the smoothing case, its estimation performance is less sensitive to the noise-variance parameter setting, compared to causal filtering. Working Matlab™ code is provided at the end of this work.**

***Index Terms*—Sensor fusion; Asynchronous sampled-data; Extended Kalman Filter; Rauch-Tung-Striebel Smoother; Heavy-duty vehicles; SAE J1939**

I. Introduction

This paper addresses sensor fusion techniques in vehicles with data coming from industrial communication buses, such as a Controller Area Network (CAN), combined with measurements from several sensors such as GPS (Global Positioning System) and IMU (Inertial Measurement Unit), as shown in Figure 1. The goal is providing an accurate state estimate by combining traditional fusion of IMU and GPS sensors 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].

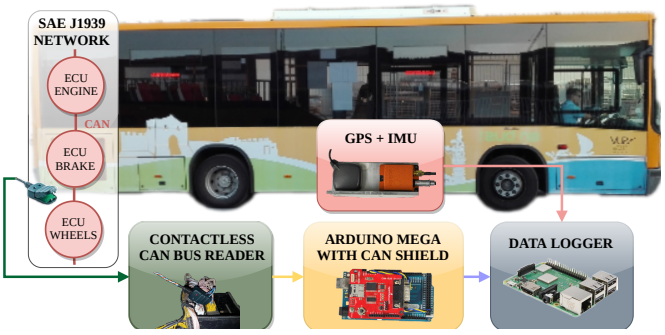


Fig. 1: Schematic representation of sensors and embedded devices used on the vehicle and their connections. The internal data of the sensors of the vehicle is accessed via CAN Network (SAE J1939 Protocol) using a contactless reader and processed by a micro-controller that sends it to an embedded PC.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

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 had to access J1939 bus, based on the protocol defined by the Society of Automotive Engineers (SAE) [6], to measure internal vehicle variables in order to control the vehicle. In [7] 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 J1939 protocol to access the information of the distributed control system of electric city buses [8], [9].

CAN bus in heavy-duty vehicles implements SAE J1939 protocol [6], 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], this method updates the estimation of a state, usually containing position and orientation of an object, based on sensor measurements. The state variables which are typically used in these filters for

[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 U. 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

vehicle applications are: position, linear and angular velocities and accelerations [12]; there are situations where sensor biases are also estimated to get a better overall accuracy [13], [14].

In tracking systems, IMU and GPS data fusion is a conventional solution for general purposes and, particularly, for estimation of kinematic state variables in land vehicles [15], [16] or UAVs (unmanned aerial vehicles) [17]. IMUs allow estimation of fast accelerations or angular rates; this is suitable to track movements with fast dynamics. Fusion with GPS data provides long-term stability, compensating offsets of IMU's measurements in navigation systems when integrated [15].

Although continuous-time observers may be mathematically devised [18], [19], discrete-time implementations are needed. Numerical integration of the continuous-time variance equation gives rise to the sampled-data Kalman filter and its Extended version [11] for causal applications. In a non-causal setup, Rauch-Tung-Striebel Smoothing (RTSS) [20] is used; incorporating model Jacobians enable extended versions for nonlinear cases [21], [22], [23]. The advantage of non-causal smoothing techniques is that they can be used for off-line data analysis, i.e., for vehicle modelling and identification; of course, in real-time applications for tracking and control, causal filters are needed.

In many applications, fusion techniques with multiple sample rates or even irregular (non-periodic) sampling instants become a relevant tool. For instance, a high-gain Extended Kalman Filter (EKF) for continuous-discrete systems with asynchronous measurements was introduced in [24]. Asynchronous multi-rate multi-sensor system estimation with unreliable measurements was proposed in [25]; in [26] the authors developed a distributed version for multi-rate systems with correlated noises. Distributed fusion filter may be a good alternative for multi-sensor systems, but a decentralised approach is not part of our problem statement. In multi-sensor target-tracking applications, the problem of out-of-sequence-measurements in cluttered environments has also been extensively studied [27], [28]. However, time-stamped data is being smoothed in our particular setting (CAN bus buffers overwrite old data if new untransmitted data reaches the bus); thus, the out-of-sequence problem is not that relevant in our approach and it is not in the scope of this work.

The main contribution of this paper is the use of non-conventional sampling data fusion, proposing an asynchronous smoother to fuse data gathered from different sources, at various possibly irregular sampling rates. This approach provides signal reconstruction for further processing at any arbitrary time instant (i.e., arbitrary sampling frequency) as the underlying algorithm is based on integration of continuous-time models. Our results, on an urban bus vehicle, show that the proposed experimental data gathering and asynchronous smoothing algorithm allow better performance at low-speeds when CAN-based odometry is used. The improvement is obtained in terms of accuracy and robustness even with significant losses of GPS samples compared to synchronous GPS+IMU options. This has been shown to be relevant when using on-line (causal) estimation methods, such as EKF. In our previous work [29], a 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. An Appendix provides the code implementing the proposed algorithms.

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 EKF [16], [30] (or later modifications such as Unscented Kalman Filter (UKF) [31], [32] and Cubature Kalman Filter (CKF) [33], [34]). These filters give a robust, optimal, recursive state estimation to fuse redundant information from different sensors [15]. This can be improved when using non-causal smoothing techniques such as the Extended Rauch-Tung-Striebel Smoothing (ERTSS) [22], [23], which uses an EKF and backward smoothing to produce quasi-optimal state trajectory estimation in optimal control settings.

Many practical sensor systems provide data at different sampling periods due to the nature of their sensing technology or due to limitations of data transmission channels [35], [36]. There are situations in which it can be assumed that there is a periodicity between all sampling rates and, as a consequence, we can work with periodic sampled-data systems (perhaps multi-rate) using models at the greatest-common-divisor sampling period. However, there are other situations in which measurements are asynchronous (i.e., not taken at integer multiples of a base sampling period) and, thus, state estimation needs to be performed based on available measurements [37] at arbitrary time instants. The possibly irregular separation between sampling instants will be denoted as δt.

The basic underlying model will now be described. Let (1) be a non-linear stochastic dynamic system and let (2) be the output equation

$$\dot{x} = f(x, \epsilon_w) \tag{1}$$

$$y = h(x) + \epsilon_v \tag{2}$$

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 (sensor dynamics is assumed much faster than that of the measured signals, ϵ_v is assumed a finite-variance discrete-time noise at sampling instants). 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

$$\dot{x} \approx A_c x + B_c \epsilon_w \tag{3}$$

$$y \approx Hx + \epsilon_v \tag{4}$$

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:

$$A_c := \frac{\partial f(x, \epsilon_w)}{\partial x} \tag{5}$$

$$B_c := \frac{\partial f(x, \epsilon_w)}{\partial \epsilon_w} \tag{6}$$

$$H := \frac{\partial h(x)}{\partial x} \tag{7}$$

From the above continuous-time linear stochastic process, a time-varying discretization can be obtained 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 [38]:

$$x_{t+\delta t} = \Psi_1, \text{ with } \Psi = \begin{bmatrix} \Psi_1 \\ \Psi_2 \end{bmatrix} = e^{\begin{bmatrix} A_c & I \\ 0 & 0 \end{bmatrix} \delta t} \cdot \begin{bmatrix} x_t \\ f(x_t, \epsilon_w) - A_c x_t \end{bmatrix} \tag{8}$$

and the discretization of the evolution of the state covariance can be done, based on the same exponential idea, with [39]

$$A_t := e^{A_c \delta t} \tag{9}$$

$$Q_t := A_t \Phi_{12} \tag{10}$$

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 irregular sampling period.

III. Problem statement

Based on the above modelling framework, this paper will provide an integrated solution for sensor fusion that combines causal (EKF filtering) and non-causal (RTSS 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 or control design in, say, subsequent processing stages. Therefore, it is advantageous to consider the irregular sampling nature of many signals as part of the filtering/smoothing process.

In addition to this, we would like to study the advantages of combining CAN-based odometry, GPS and IMU data in urban scenarios with buses. A non-invasive data logging system can be used like 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. Combining GPS with IMU and speed data obtained from CAN will be useful to determine which information is more relevant and accurate for a candidate application.

IV. Asynchronous Extended Rauch-Tung-Striebel Smoother

Let us introduce an Asynchronous Extended Rauch-Tung-Striebel Smoother (AERTSS) algorithm for sampled data systems, which combines a continuous-discrete sampled-data EKF and a discrete RTSS.

In the first stage, the AERTSS algorithm performs a continuous in time prediction and a discrete measurement update, based on an EKF. Due to technological limitations of CAN-based measurements, data is assumed to be measured asynchronously, which implies that the EKF needs to integrate state predictions over time (using matrix exponential formulae, both in mean and variance) 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 (AEKF). In a second stage of the algorithm, the estimated state coming from the AEKF is smoothed to provide better estimations, given that RTSS is a non-causal filter that runs backward in time to provide estimations with data information from future time instants.

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

- A dataset consisting on triplets with the following information: $\mathcal{D} := \{(y_{t_1}, s_1, t_1), (y_{t_2}, s_2, t_2), \ldots\}$, being y_{t_i} 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. Triplets $\{[], [], t_i\}$ with *empty* fields y and s will denote points of interest for state prediction even if no measurement was taken at the stated time instant, which can be arbitrary.
- Expressions for the model non-linearities (1) and (2), 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 (process noise), measurement-noise covariance 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 a given sensor s will be denoted as R^{ss}.

The detail of the algorithm steps are as follows. First, lines 3 to 20 implement the AEKF. In particular, lines 5-10 perform

the asynchronous prediction step, where we compute mean using equations (5)-(8) and forward state covariance prediction using (9). It is worth mentioning that state covariance update is implemented using lines 11-18 with available measurements. If no measurement is available, then the update step is not performed.

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 RTSS (implemented on lines 22-27). Note that in Algorithm 1 the EKF estimate is $\hat{x}_t$, whereas $\hat{z}_t$ is the AERTSS state estimate.

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 feeding the algorithm a triplet with no available data (empty y_t), which will subsequently entail setting the Kalman gain to zero. For instance, we can provide state estimates at regularly spaced time instants at any desired sampling frequency, while actual 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 arbitrarily high sampling rate, as the underlying models are time-continuous: this feature allows to reconstruct the smoothed signal/state at any arbitrary time instant, regardless of its coincidence (or lack thereof) with any measurement.

Algorithm 1 AERTSS [AEKF IS STOPPED AFTER STEP 20]

1: **function** $\mathcal{Z}$=AERTSS($\mathcal{D}$,$\mathcal{X}$)
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 $\{x_{ini}, P_{ini}, t_0\}$.
[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: $A_c \leftarrow \frac{\partial f(x,\epsilon_w)}{\partial x}\Big|_{\substack{x=\hat{x}_t \\ \epsilon_w=0}}$, $B_c \leftarrow \frac{\partial f(x,\epsilon_w)}{\partial \epsilon_w}\Big|_{\substack{x=\hat{x}_t \\ \epsilon_w=0}}$
6: $\begin{bmatrix} \Psi_1 \\ \Psi_2 \end{bmatrix} \leftarrow e^{\begin{bmatrix} A_c & I \\ 0 & 0 \end{bmatrix}\delta t} \cdot \begin{bmatrix} \hat{x}_t \\ f(\hat{x}_t,0) - A_c\hat{x}_t \end{bmatrix}$
7: $\tilde{x}_t \leftarrow \Psi_1$
8: $\begin{bmatrix} \Phi_{11} & \Phi_{12} \\ 0 & \Phi_{22} \end{bmatrix} \leftarrow e^{\begin{bmatrix} -A_c^T & B_c Q_c B_c^T \\ 0 & A_c \end{bmatrix}\delta t}$
9: $A_t \leftarrow \Phi_{22}$, $Q_t \leftarrow \Phi_{22}^T \Phi_{12}$
10: $\tilde{P}_t \leftarrow A_t P_t A_t^T + Q_t$
[State and covariance update]
11: **if** y_t is not empty **then**
12: $H_t^s \leftarrow \frac{\partial h^s(x)}{\partial x}\Big|_{x=\tilde{x}_t}$
13: $K_t \leftarrow \tilde{P}_t (H_t^s)^T (H_t^s \tilde{P}_t (H_t^s)^T + R^{ss})^{-1}$
14: **else**
15: $K_t \leftarrow 0$
16: **end if**
17: $\hat{x}_t \leftarrow \tilde{x}_t + K_t(y_t - h_t^s)$
18: $P_t \leftarrow (I - K_t H_t^s) P_t$
19: Append $\{A_t, \hat{x}_t, \tilde{x}_t, P_t, \tilde{P}_t, t\}$ to $\mathcal{T}$
20: **end for**
[Smoothing (backward in time)]
21: $\hat{z}_t \leftarrow \hat{x}_t$, $\mathcal{Z} \leftarrow \{z_t\}$
22: **for** $i = \text{length}(\mathcal{T}) - 1$ to 1 **do**
23: $\delta t = \mathcal{T}_{i+1}.t - \mathcal{T}_i.t$
24: $L_t \leftarrow (\mathcal{T}_i.P_t)(\mathcal{T}_{i+1}.A_t)^T (\mathcal{T}_{i+1}.\tilde{P}_t)^{-1}$
25: $\hat{z}_t \leftarrow \mathcal{T}_i.\hat{x}_t + L_t(\hat{z}_t - \mathcal{T}_{i+1}.\tilde{x}_t)$
26: Prepend z_t to $\mathcal{Z}$
27: **end for**
28: **end function**

A. AERTSS vehicle data

In this section, we explain data used in AERTSS 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, since 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 [6] 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. AERTSS vehicle model

Given the absence of the wheel orientation measurement, it was estimated from the measurements of front wheels velocities, published in the CAN network, assuming an Ackermann steering geometry [40].

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

$$\phi_w = 0.5 \arcsin \frac{4L(v_R - v_L)}{W(v_R + v_L)} \tag{11}$$

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 [40] 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 (11):

$$\omega^{CAN} = \frac{v \tan \phi_w}{L} \tag{12}$$

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:

$$\dot{x} = f(x, \epsilon_w) := \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ \omega \\ 0 \\ a \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + B_c \epsilon_w \tag{13}$$

$$y = h(x, \epsilon_v) := Hx + \epsilon_v \tag{14}$$

where B_c, H and the Jacobian A_c of the state equation (to be used in the AERTSS algorithm) are:

$$A_c = \begin{bmatrix} 0 & 0 & -v\sin(\theta) & 0 & \cos(\theta) & 0 & 0 & 0 & 0 \\ 0 & 0 & v\cos(\theta) & 0 & \sin(\theta) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

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

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).

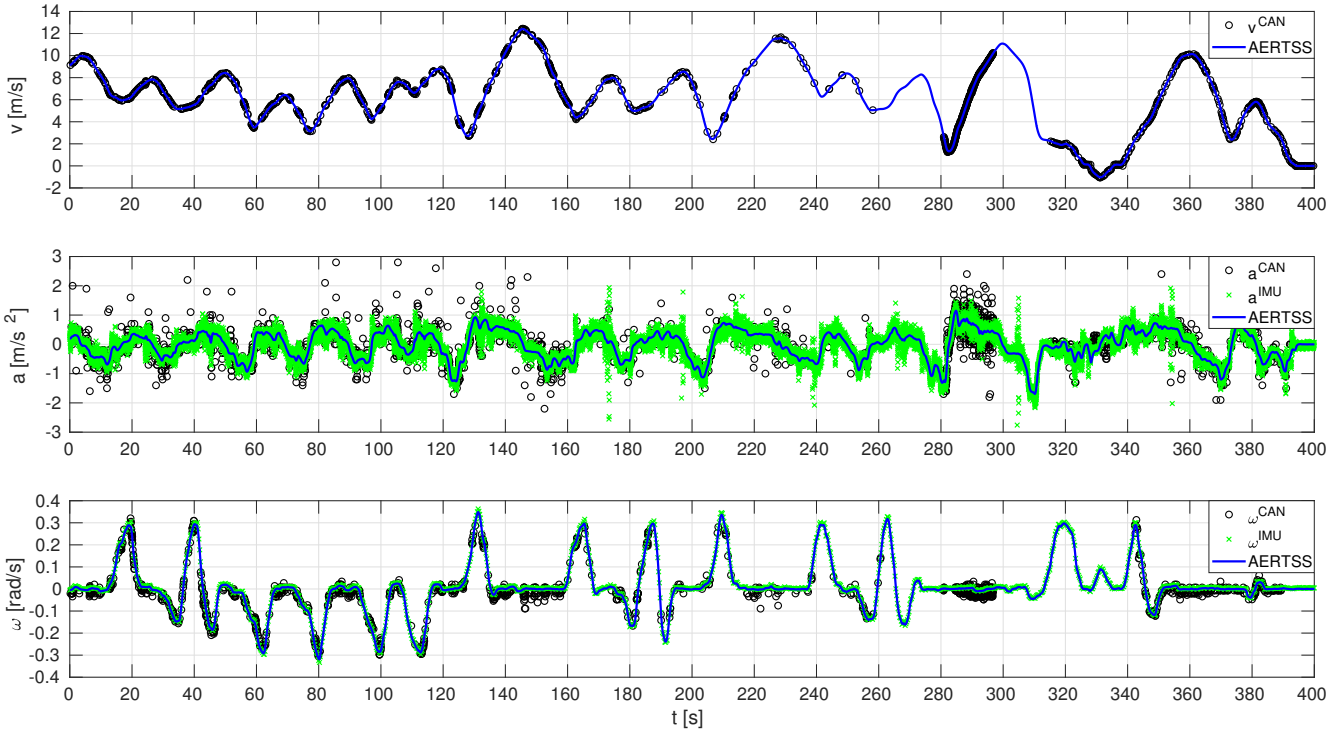
A contactless CAN connector was used to safely read data from vehicle CAN bus (see Figure 1). Using this device, data reading is non-invasive as it occurs without electrical connection and without damaging CAN wires. It works in "listen" mode only, i.e. it does not change original J1939 messages and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

A driving test was carried out in order to collect data. It was conducted in an urban scenario, where different manoeuvres were combined (straight driving, corners, roundabouts, etc.) and the bus driver faced everyday situations; the trip is depicted in Figure 2(a). The overall duration of data recording selected for this work was 420 s.

(a) Position on map



(b) Kinematic variables

Fig. 2: Driving test in urban scenario: (a) trace of vehicle's position estimated with AERTSS 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).

VI. Results

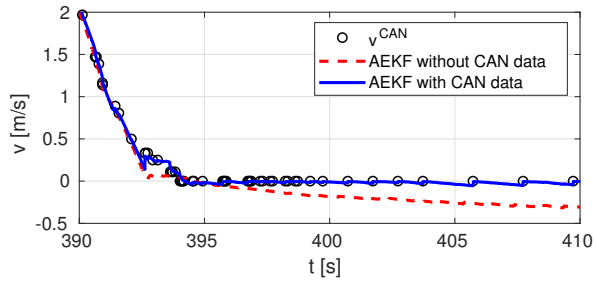
To estimate vehicle's kinematics, considering the measurement uncertainties, covariance matrices are initialised as follows in all driving tests:

$$Q_c = \mathrm{diag}(\{0.25, 0.25, 0.01, 0.0025, 10^{-6}, 10^{-6}, 10^{-6}\}),$$
$$R = \mathrm{diag}(\{0.25, 0.25, 0.01, 0.04, 0.0025, 0.25, 0.0001, 1\}).$$

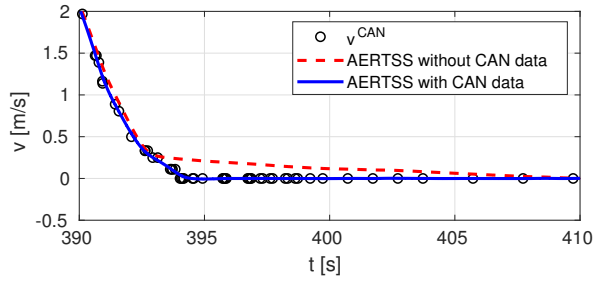
with initial state mean $x_{ini} = [0, 0, 0, 0, 0, 0, 0, 0, 0]^T$ and variance $P_{ini} = \mathrm{diag}(\{400, 400, 1, 4, 1, 1, 10^4, 10^4, 10^4\})$.

GPS/IMU measurement error was set from sensors' technical specifications and validated from measurement variance when the vehicle was at zero speed. CAN-odometry measurement error was estimated from segments of the dataset at grossly constant speed (variance of the difference from the linear regression estimate). Process-noise variance was determined by trial and error until estimated observer bandwidth and variance drift were deemed reasonable. Note that given the extra noncausal information available in the smoothing case, the AERTSS performance was less sensitive to the actual process-noise variance parameter setting than the AEKF setup.

To validate the AERTSS data fusion algorithm, it has been applied in a long driving test in an urban scenario, as shown in Figure 2. The position on map is depicted in Figure 2(a), whilst Figure 2(b) represents some kinematic variables (linear velocity, linear acceleration and angular velocity). These signals have been reconstructed using the proposed AERTSS.



(a) AEKF algorithm (prediction)



(b) AERTSS algorithm (prediction+smoothing)

Fig. 3: Time zoom of 20 seconds of vehicle's linear velocity after braking to analyse the effect of making the data fusion without 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; CAN sampling rate was markedly irregular, but poses no problem for AERTSS). It can be observed that the AERTSS algorithm is able to estimate correctly linear and angular velocities, being the acceleration noise reduced considerably and all sensor biases compensated.

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. AERTSS allows to have position estimates at 100 Hz sampling rate, even if the original GPS rate was 1 Hz.

Graphs in Figure 2(b) show linear velocity, linear acceleration and angular velocity, from top to bottom (both raw measurements and sensor-fusion estimates, to be latter commented upon). It can be observed that 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 and derivation of travelled distance computed from GPS position. 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: Figure 3 shows the vehicle's linear velocity estimation after braking. As shown in Figure 3(b), when the vehicle brakes the proposed AERTSS algorithm estimates that

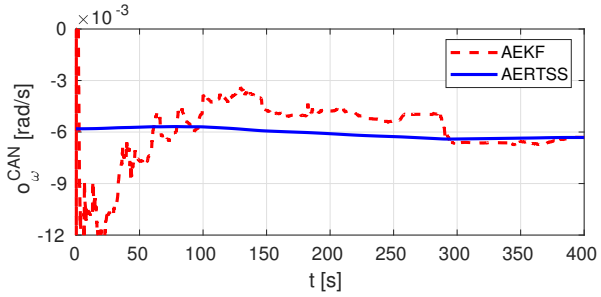


Fig. 4: Estimated offset of angular velocity from CAN bus.

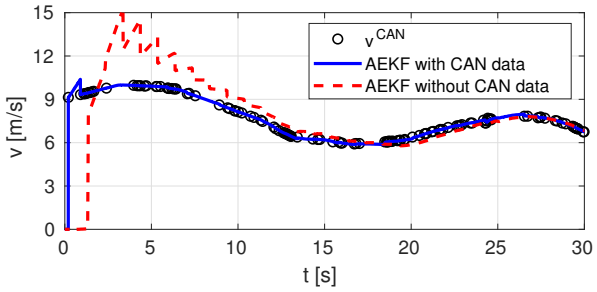
it is not completely stopped when it actually is, if CAN data is not used (dashed red line): it takes about 15 s for the AERTSS algorithm to estimate that the vehicle is not moving; note that causal AEKF drifts away to negative speeds without CAN data (Figure 3(a), dashed red). However, the AERTSS estimation using CAN data (blue line) is much more accurate, as it returns zero velocity thanks to the wheels' velocity reading from CAN bus messages; these CAN readings avoid, too, the AEKF drift (Figure 3(a), solid blue). Note that the AEKF estimation (i.e., using only a prediction step) implies having signal discontinuities (jump or reset effect) due to the filtering update after receiving new sensor data. AERTSS estimation uses a smoothing step to avoid such discontinuous behaviour, as observed in Figure 3(b).

In fact, the reason for the no-CAN velocity offsets in Figure 3 was that GPS was outputting non-constant positions originating a velocity offset; the same happened to the IMU, that returned a small value of acceleration offset that deviated the estimators towards a non-zero linear velocity. The presence of these offsets motivated its inclusion in the model in order to also estimate them, as later discussed below.

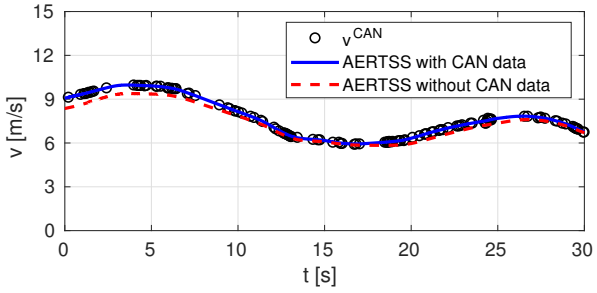
c) Angular kinematics: If we take a look at the bottom graph in Figure 2(b), the angular velocity from the IMU has a constant bias. Furthermore, the CAN measurement also has a negative offset, which is even bigger in magnitude. Nevertheless, thanks to the data fusion of these measures together with GPS position, the estimation is improved and the biases compensated.

d) Instrument offset identification: The average signal biases were identified based on the results obtained from the experimentation. On one side, data coming from the CAN network (angular velocity offset $o_\omega^{CAN} \approx -0.006$ rad/s); on the other, signals from the IMU sensor (gyroscope offset $o_\omega^{IMU} \approx 0.003$ rad/s and accelerometer offset $o_a^{IMU} \approx -0.015$ 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. A bias versus time plot is provided in Figure 4 for the angular velocity offset o_ω^{CAN}. Similar results are obtained for the IMU offsets, not shown for brevity.

The urban driving test has also been used to analyse robustness and accuracy of the proposed sensor fusion setting. Two different studies have been performed: effect of incorrect state space initialisation and robustness to missing data. Some aspects considered for such studies are described below.



(a) AEKF algorithm (prediction)



(b) AERTSS algorithm (prediction+smoothing)

Fig. 5: Initial performance comparison with the absence of CAN data for causal/non-causal filters.

e) Initial performance of AEKF vs. AERTSS: 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 AEKF without CAN data 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 5(a). The knowledge of future samples mitigates this initialization problem in the AERTSS case, as observed in Figure 5(b).

f) Study of robustness under missing CAN data: Table I shows the unexplained standard deviation of the vehicle's state estimation errors using different algorithms and CAN availability setups; the AERTSS output with CAN data is set as the reference value, i.e., error figures have been computed as the difference with respect to this smoother. The results show that the estimation of position p_x, p_y, orientation θ and angular velocity ω using AEKF is not significantly improved by availability of CAN data; improvement over angular speed accuracy is not very significant, possibly because inaccuracies in its estimation via Ackermann's formulae do not help improving the IMU's estimation of the said variable. However, CAN availability does significantly improve the accuracy of linear speed and acceleration estimates, as intuitively expected.

g) Study of robustness under missing GPS data: In this 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%). AEKF and AERTSS data fusion algorithms are compared, in both cases with and without in-vehicle CAN bus information. The estimation at very high frequency (1 kHz[1]) using the AERTSS algorithm with all available information (GPS, IMU and CAN data) is used as a reference value for comparison between other options, i.e., later error figures will be computed as the difference with respect to the estimation with 100% available data, at said rate.

TABLE I: Standard deviation of estimated variables minus those from AERTSS with CAN

Signal	AEKF		AERTSS
	without CAN	with CAN	without CAN
p_x [m]	0.8248543	0.7427177	0.0664604
p_y [m]	0.2924437	0.2741514	0.0918966
θ [rad]	0.0706022	0.0693579	0.0009166
ω [rad/s]	0.0604841	0.0574169	0.0004288
v [m/s]	0.6727803	0.2095957	0.2232302
a [m/s^2]	0.2733062	0.1174341	0.0615074

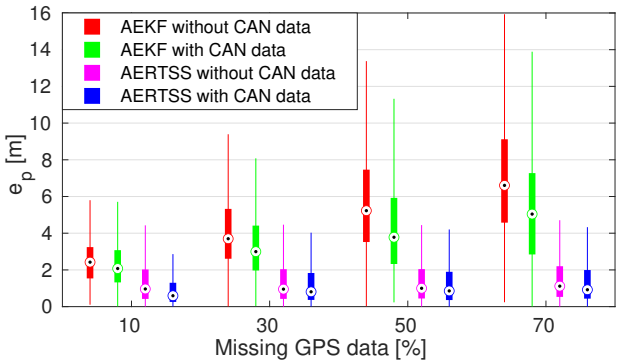


Fig. 6: Estimated position error e_p in the urban driving test for different percentages of missing GPS data and availability of CAN data.

Figure 6 shows the box-and-whisker plot of the position estimation error e_p with randomly missing GPS data. The red and green boxes correspond to estimations using AEKF, whilst the magenta and blue whiskers diagrams represent estimations with AERTSS 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 wheel odometry from 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 AERTSS with 10% of GPS data missing) just by plugging information from CAN bus into the algorithm. Figure 6 also shows the accuracy advantage of the non-causal smoothing in missing-data situations compared to causal AEKF.

VII. Conclusion

This paper has presented an asynchronous smoothing algorithm combining data from IMU, GPS and CAN-based odometry at arbitrary (possibly irregular) sampling rates. Based on

[1]The chosen estimation frequency does not influence the accuracy (AERTSS can provide filtered data at arbitrary time instants). The chosen frequency is high enough to be considered as an approximation of the continuous-time error probability distribution.

the experimental results on an urban bus, it can be concluded that, as CAN messages provide more 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 situations with missing data and to improve initial performance. Besides, given the extra information available in the smoothing case, the AERTSS performance was less sensitive to the actual process-noise variance parameter setting than the AEKF setup. 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 is valid in many other scenarios, where asynchronous data is required to be fused. Working Matlab™ code for the AERTSS algorithm is also provided as an Appendix below in this work.

MATLAB™ CODE

```
function [Z]=AERTSS(D,X,f,h,Ac,Bc,H,Qc,R,causal)
    N=size(D,1);
    x=X.xini; P=X.Pini; n=size(x,1); tbar=X.t0;
    T(N,1)=struct('A',zeros(n,n), ...
        'x',zeros(n,1), 'xp',zeros(n,1), ...
        'P',zeros(n,n), 'Pp',zeros(n,n), 't',0);
    dt=zeros(N,1);
    for i=1:N
        y=D(i).y; t=D(i).t; s=D(i).s;
        dt=t-tbar; tbar=t;
        [xp,A,Q]=expm_pred(x,f,Ac,Bc,Qc,dt);
        Pp=A*P*A'+Q;
        if (¬isempty(y))
            H_x=H(xp); Hs_x=H_x(s,:);
            K=Pp*Hs_x'/(Hs_x*Pp*Hs_x'+R(s,s));
        else
            K=0;
        end
        h_x=h(xp);
        x=xp+K*(y-h_x(s));
        P=(eye(n)-K*Hs_x)*Pp;
        T(i)=struct('A',A,'x',x,'xp',xp, ...
            'P',P,'Pp',Pp,'t',t);
    end
    if(causal) Z=x; return; end
    Z=zeros(n,N); Z(:,end)=x;
    for i=N-1:-1:1
        x=T(i).x; P=T(i).P; dt=T(i+1).t-T(i).t;
        xp=T(i+1).xp; A=T(i+1).A; Pp=T(i+1).Pp;
        L=P*A'/Pp;
        Z(:,i)=x+L*(Z(:,i+1)-xp);
    end
end

function [x,A,Q]=expm_pred(x,f,Ac,Bc,Qc,dt)
    Ac_x=Ac(x); Bc_x=Bc(x);
    n=size(Ac_x,1);
    psi=expm([Ac_x eye(n);zeros(n,2*n)]*dt)* ...
        [x;f(x)-Ac_x*x];
    x=psi(1:n);
    phi=expm([-Ac_x' Bc_x*Qc*Bc_x';zeros(n) ...
        Ac_x]*dt);
    A = phi(n+1:end,n+1:end);
    Q = phi(n+1:end,n+1:end)'* phi(1:n,n+1:end);
end
```

This section presents Matlab code for Algorithm 1. Its inputs are: a struct array D, with fields 'y', 't' and 's', corresponding to the measurement value, sampling time and sensor number, respectively. Also, a struct X with fields 'Xini', 'Pini' and 't0' is required to provide the initialisation of the algorithm. Function handles 'f' and 'h', 'Ac', 'Bc' and 'H' need to be provided accordingly to what is described in Section IV-B, as well as the process and measurement covariance matrices Q_c and R. Finally, the 'causal' flag allows computing either AEKF (prediction) when true, or AERTSS (prediction+smoothing) when false.

ACKNOWLEDGMENTS

This research was funded by Agencia Española de Investigación (European Union), grants PID2020-116585GB-I00 and PID2020-118071GB-I00, and by Generalitat Valenciana, grant GV/2021/074. 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 AI2 and IDF institutes (CAROT DPI2016-81002-R, SAFETRANS DPI2013-42302-R projects) 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] W. Voss, *A comprehensible guide to J1939*. Copperhill Technologies Corporation, 2008.

[7] 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.

[8] 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.

[9] 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.

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

[13] 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.

[14] 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.

[15] W. Quan, J. Li, X. Gong, and J. Fang, *INS/CNS/GNSS integrated navigation technology*. Springer, 2015.

[16] 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.

[17] 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.

[18] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York: Academic Press, 1970.

[19] G. González, D. Aligia, C. Pezzani, and C. D. Angelo, "Ciclist's torque observer in electric power assisted bicycles," *Revista Iberoamericana de Automática e Informática industrial*, vol. 17, no. 4, pp. 380–389, 2020.

[20] H. E. Rauch, C. T. Striebel, and F. Tung, "Maximum likelihood estimates of linear dynamic systems," *AIAA journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[21] Y. Xu, X. Chen, and Q. Li, "Autonomous integrated navigation for indoor robots utilizing on-line iterated extended rauch-tung-striebel smoothing," *Sensors*, vol. 13, no. 12, pp. 15 937–15 953, 2013.

[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] A. Feddaoui, N. Boizot, E. Busvelle, and V. Hugel, "High-gain extended kalman filter for continuous-discrete systems with asynchronous measurements," *International Journal of Control*, vol. 93, no. 8, pp. 2001–2014, 2020.

[25] L. Yan, X. R. Li, Y. Xia, and M. Fu, "Modeling and estimation of asynchronous multirate multisensor system with unreliable measurements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 3, pp. 2012–2026, 2015.

[26] H. Lin and S. Sun, "Distributed fusion estimator for multisensor multirate systems with correlated noises," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 7, pp. 1131–1139, 2018.

[27] I. Ullah, M. B. Qureshi, U. Khan, S. A. Memon, Y. Shi, and D. Peng, "Multisensor-based target-tracking algorithm with out-of-sequence-measurements in cluttered environments," *Sensors*, vol. 18, no. 11, p. 4043, 2018.

[28] Y. Shi, S. Qayyum, S. A. Memon, U. Khan, J. Imtiaz, I. Ullah, D. Dancey, and R. Nawaz, "A modified bayesian framework for multi-sensor target tracking with out-of-sequence-measurements," *Sensors*, vol. 20, no. 14, p. 3821, 2020.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] I. Arasaratnam and S. Haykin, "Cubature kalman filters," *IEEE Transactions on automatic control*, vol. 54, no. 6, pp. 1254–1269, 2009.

[34] H. Dai, S. Dai, Y. Cong, and G. Wu, "Performance comparison of ekf/ukf/ckf for the tracking of ballistic target," *Indones. J. Electr. Eng. Comput. Sci*, vol. 107, pp. 1692–1699, 2012.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] C. Van Loan, "Computing integrals involving the matrix exponential," *IEEE Trans. Automatic Control*, vol. 23, no. 3, pp. 395–404, 1978.

[40] W. Chung and K. Iagnemma, *Wheeled Robots*. Cham: Springer International Publishing, 2016, pp. 575–594.

Vicent Girbés-Juan received the B.Eng. and M.Sc. in Ind. Electronics and Control in 2009 and 2011, respectively; and the PhD. in Automation, Robotics and Comp. Science in 2016. From 2009 to 2019 worked at the Universitat Politècnica de València in the Robotics and Automation Research Group. He has been visiting researcher at University of Manchester and Imperial College London. Currently, he is Assistant Professor in the Electronic Eng. Dept. at Universitat de València. His fields of interest are HRI, autonomous driving and machine learning.

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 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 and Reinforcement Learning.

Daniel Hernández-Ferrándiz received the B.Eng. and M.Sc. in Industrial Electronics and Automatics in 2016 and 2018 from the UPV, and M.Sc. in Computer Vision from Universitat Rey Juan Carlos (2020). From 2018 to 2019 he worked as a researcher in the Automation and Industrial Infomatics Institute (AI2) at UPV, contributing to the experimental/coding tasks of this work.

Juan Dols is Ind. Eng. (Mechanics) since 1987 and in 1996 the Ph.D. degree at UPV, Spain, where he is now teaching Automotive and Transportation Eng. at the Mechanical and Materials Eng. Dept. of UPV. Professor Dols is also Director of the UPV Automobile Laboratory (LAUPV). His research activity is based on the design and development of driving simulators, wheelchair tiedown and occupant restraint systems, active and passive vehicle safety, accessible transportation, driving aids development, accident reconstruction and vehicle dynamics.

Antonio Sala received a Ph.D. in Control Eng. from UPV, Spain. Since 1993, he has been teaching in UPV where he currently is full Professor. His current research interests are optimal, fault-tolerant, robust and LPV control. Antonio Sala has served as Associate Editor of IEEE Trans. on Fuzzy Systems, Fuzzy Sets and Systems, and Rev. Iberoamericana Automática e Informática Industrial.