

# Road-Centered Map-Aided Localization for Driverless Cars Using Single-Frequency GNSS Receivers

**Zui Tao, Philippe Bonnifait, and Vincent Frémont**

*Sorbonne Universités, Université de Technologie de Compiègne, CNRS, Heudiasyc UMR 7253, CS 60319, 60203 Compiègne cedex, France*

*e-mail: zui.tao@hds.utc.fr*

**Javier Ibanez-Guzman**

*Renault S.A.S, Technocentre 78084 Guyancourt, France*

**Stéphane Bonnet**

*Sorbonne Universités, Université de Technologie de Compiègne, CNRS, Heudiasyc UMR 7253, CS 60319, 60203 Compiègne cedex, France*

Received 8 April 2016; accepted 19 December 2016

Accurate localization with high availability is a key requirement for autonomous vehicles. It remains a major challenge when using automotive sensors such as single-frequency Global Navigation Satellite System (GNSS) receivers, a lane detection camera, and proprioceptive sensors. This paper describes a method that enables the estimation of stand-alone single-frequency GNSS errors by integrating the measurements from a forward-looking camera matched with lane markings stored in a digital map. It includes a parameter identification method for a shaping model, which is evaluated using experimental data. An algebraic observability study is then conducted to prove that the proposed state vector is fully observable in a road-oriented frame. This observability property is the basis to develop a road-centered Extended Kalman filter (EKF) that can maintain the observability of every component of the state vector on any road, whatever its orientation. To accomplish this, the filter needs to handle road changes, which it does using bijective transformations. The filter was implemented and tested intensely on an experimental vehicle for driverless valet parking services. Field results have shown that the performance of the estimation process is better than solutions based on EKF implemented in a fixed working frame. The proposed filter guarantees that the drift along the road direction remains bounded. This is very important when the vehicle navigates autonomously. Furthermore, the road-centered modeling improves the accuracy, consistency, and robustness of the localization solver. © 2017 Wiley Periodicals, Inc.

## 1. INTRODUCTION

Autonomous ground vehicles (AGVs) have existed as prototype and demonstration vehicles since the 1970s. Their widespread use promises increased comfort, safety, reduced traffic congestion, energy conservation, and pollution reductions (Litman, 2013). Usually, an AGV needs to perform four kinds of tasks: localization, perception, path planning, and control. This paper focuses on the localization system. Localization generally refers to determining the vehicle's position  $(x, y)$  and heading  $(\psi)$  with respect to a map that includes the vehicle's goals. For autonomous navigation, the real-time vehicle localization is required to be accurate, fully available, and reliable. Expensive sensor suites, such as Real-Time Kinematic Global Positioning System and high-grade Inertial Measurement Unit (IMU), are often adopted to achieve good performance. The abil-

ity to simultaneously estimate the pose of a vehicle and give reliable confidence indicators using only low-cost sensors remains a challenging problem, particularly in outdoor environments.

Recently there have been impressive demonstrations on both rural and urban routes of self-driving cars using close-to-market sensors and enhanced maps (Furgale et al., 2013; Schreiber, Knoppel, & Franke, 2013; Ziegler et al., 2014). In most cases, an on-road assumption and the use of an informative digital map make low-cost autonomous driving possible. In the research community, there has been significant progress in the generation of detailed digital maps (Guo, Meguro, Kojima, & Naito, 2014), to the point that map-aided perception (Cui, Xue, Du, & Zheng, 2014; Kurdej, Moras, Cherfaoui, & Bonnifait, 2014) and map-aided localization (Jo, Chu, & Sunwoo, 2013; Miller, Campbell, & Huttenlocher, 2011; Rose, Britt, Allen, & Bevly, 2014; Schreiber, Knoppel, & Franke, 2013) are now energetically being considered for autonomous navigation.

Direct correspondence to: Zui Tao, email: zui.tao@hds.utc.fr

A detailed and highly accurate map often contains lane polylines, carriageway boundaries, lane markings with associated types, speed limits, and additional information useful for navigation. The use of such a map makes the vehicle able to localize itself precisely in order to maneuver correctly with a path planner.

In order to benefit from a highly accurate map, a vehicle needs to measure its relative position with respect to features of the road on which it is traveling, by integrating perception information such as lane detections. An up-to-date survey of lane detection can be found in Bar Hillel, Lerner, Levi, and Raz (2014). The sensing technologies involved are mainly camera and lidar. Lidar uses active light and is therefore not affected by certain natural light issues. However, it can face problems in snow, rain, and fog. Lane and road boundary detection using only laser-based sensors has been discussed in Morales, Tsubouchi, and Yuta (2009), Ogawa and Takagi (2006), Reyher, Joos, and Winner (2005), and Wijesoma, Kodagoda, and Balasuriya (2004). Cameras are attractive sensors, as they are cheap and provide high-resolution information with lower operating power. Multi-sensor lane-finding systems have also been proposed for autonomous driving (Huang, Moore, Antone, Olson, & Teller, 2008). The obvious major drawback of lidar is the relatively high cost of sensors, which has prevented their use becoming widespread in automotive applications. Camera-based lane recognition systems are relatively mature nowadays and have already been introduced to market for lane departure warning systems (LDWSs). Many LDWSs provide, on the Controller Area Network (CAN) bus of a modern vehicle, several attributes of the detected lane markings, including lane type, position, curvature, curvature derivative, and heading. These local measurements are useful for improving vehicle localization when coupled with a lane marking map.

In order to get localization at high frequency and with high availability, proprioceptive sensors such as wheel speed sensors and yaw rate gyros are particularly relevant. These sensors are already available in modern cars. In Europe, every new vehicle has had to be equipped with an Electric Stability Program (ESP) since January 2012. Anti-lock Braking System (ABS) has been part of standard equipment even longer. These two systems contain a wealth of proprioceptive sensors that are capable of measuring vehicle yaw rate and individual wheel speed. The sensors can easily send their measurements to electronic control units via a CAN bus, to enable dead reckoning (DR). Even if drift is unavoidable, DR provides accurate short-term estimates with high frequency.

The most widely used outdoor localization systems are Global Navigation Satellite Systems (GNSSs). A GNSS provides an estimation of absolute position, and can therefore be useful when initializing a system, compensating for DR drift or calibrating sensors. The fusion of GNSS and DR has been widely studied (Bar-shalom, Li, &

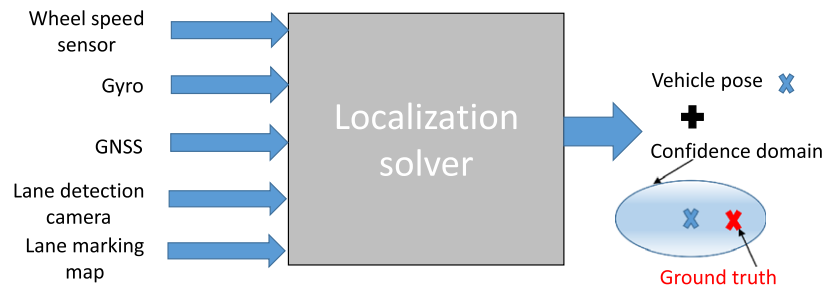


**Figure 1.** Driverless valet parking vehicle used in the PAMU (Plateforme avancée de Mobilité Urbaine) project (Laboratoire Heudiasyc, 2015).

Kirubarajan, 2002; Bonnifait, Bouron, Crubille, & Meizel, 2001; Gao, Petovello, & Cannon, 2006; Sukkariéh, Nebot, & Durrant-Whyte, 1999). However, measurements are often affected by strong biases when using a stand-alone GNSS receiver. These biases are caused mainly by satellite position errors from real-time broadcast ephemeris, atmospheric propagation delays, and multipath effects. Although the manufacturers of GNSS receivers are providing increasingly reliable solutions with the development of multiconstellation satellite technology, it remains hard to compensate systematic errors for a stand-alone solution without additional sensors or differential corrections.

In this paper, we develop a localization system for providing an accurate real-time pose estimate to be used as a feedback for the autonomous navigation of a full-sized car in valet parking operations (see Figure 1). The estimation of pose is used by a motion planner that enables the car to follow a predetermined path. We focus on map-aided methods and explore the feasibility of using low-cost automotive sensors to achieve this goal.

Figure 2 displays a systemic view of the proposed localization system. We use automotive sensors with CAN bus interfaces. The available sensor information sources are wheel speed sensors and a gyro for DR, GNSS for global positioning, and a lane detection camera coupled with a lane marking map for accurate cross-track positioning. The GNSS, when coupled with DR, provides vehicle pose estimation with high availability but low accuracy. Conversely, the lane detection camera with the lane marking map provides estimates with low availability but high accuracy. Availability is low because the lane detection camera from the LDWS works well only in areas where the lane marking is well defined, without any ambiguity, and reliable measurements cannot be obtained at intersections and on highly curved roads. The two groups of sensors are quite complementary. Therefore, our objective is to design a localization solver that merges all these information sources to achieve GNSS/DR-like availability and camera-like accuracy. In addition, we expect to have a highly consistent localization



**Figure 2.** Systemic view of the proposed approach.

solver, which means that the estimated confidence domain contains the ground truth as often as possible.

Research approaches in the literature have focused on tightly coupling vision sensors and enhancing the feature detection algorithm such as in Gruyer, Belaroussi, and Revilloud (2016), Hara and Saito (2015), Levinson and Thrun (2010), Pink (2008), and Wolcott and Eustice (2014). In this paper, the lane detection camera is an off-the-shelf system. The availability of lane marking detection will sometimes be low, and it outputs only the lane marking parameters of the host lane of the vehicle. The good aspect is that the host lane detection function is proved to be quite robust. The detection of the closest markings usually leads to less complex and more robust detection as argued in Gruyer et al. (2016). We focus on improving the GNSS/DR accuracy by modeling the measurement errors. Integrating the camera with the lane marking map makes this possible. Meanwhile, the camera measurements are related to the local geometry of the road. Generally, cross-track measurements (e.g., lane marking measurements) will be made more frequently than along-track measurements (e.g., stop line measurements). The observability of the GNSS/DR error is therefore weak in the along-track direction, whereas it is high in the cross-track direction.

We adopt a loosely coupled scheme, meaning that high-level information (e.g., position fixes of the GNSS receiver and clothoids of the lanes detected by the camera) and not raw data are fed into the localization solver. This option is often chosen because of its simplicity, given that position fixes are available in any standard GNSS unit. In this paper, the localization solver is realized using an Extended Kalman filter (EKF). It is difficult to model the error of a loosely coupled GNSS receiver, because it is affected by both the external environment and internal filtering effects of the receiver. Our solution is to identify a model that can represent effectively the major properties of GNSS errors, namely systematic bias, white noise, and autoregressive/Gauss–Markov processes. Navigation system errors can be approximated by this modeling in most cases, provided that the models adequately fit the real errors (Groves, 2013). In particular, we propose to enhance the GNSS error model in the cross-track direction by integrating camera cross-track measurement.

The state vector is then augmented by adding the sensor error models. To maintain the observability of each state element, we propose a road-centered Extended Kalman filter (RC EKF) algorithm for handling the enhanced GNSS error model. By “road-centered,” we mean that the working frame of the localization solver is adjusted so that it is always aligned with the road that the vehicle is traveling along. This idea is inspired from the invariant EKF proposed by Bonnabel, Martin, and Salaun (2009). Our localization problem possesses natural invariance with respect to road rotations, and geometrical transformations can be found to make the system invariant. The nonlinear observability of the augmented state vector is studied here in an algebraic framework.

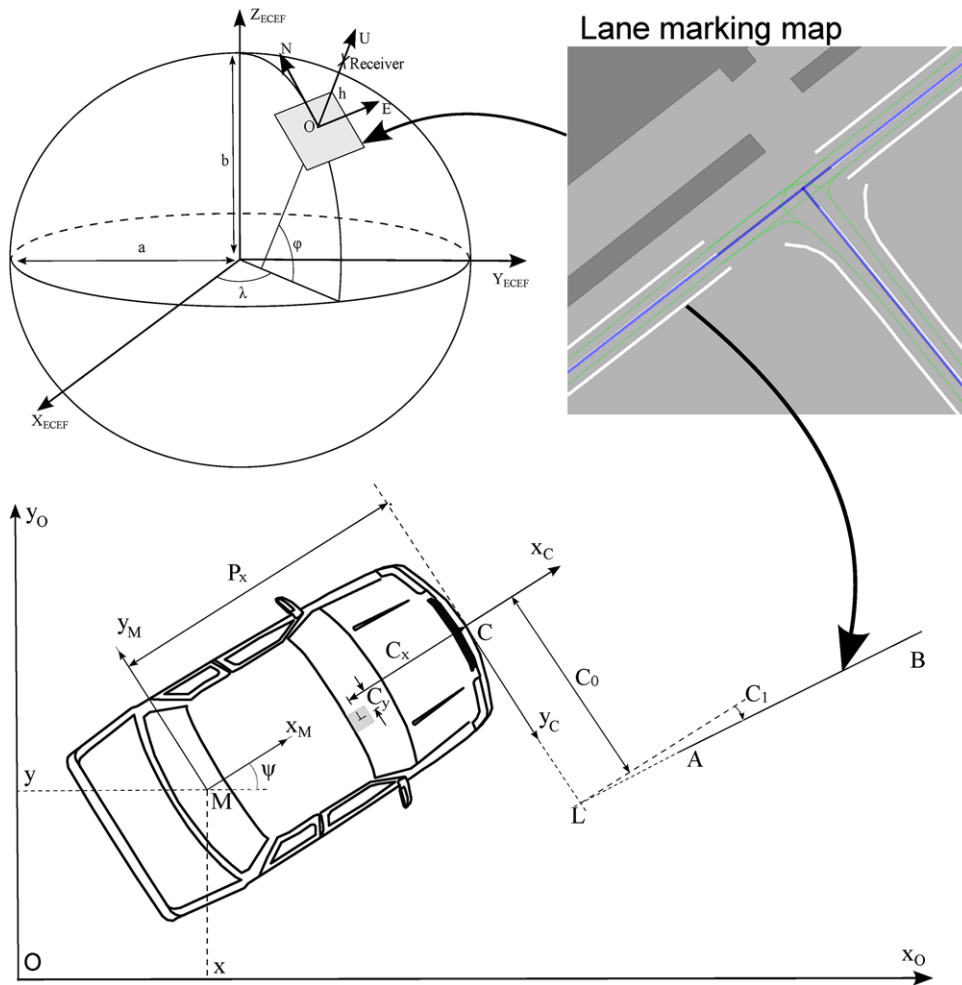
The main contribution of this paper is to propose a new modeling of GNSS error that is particularly well adapted for a new road-centered filtering scheme integrating cross-track lane marking measurements in real time. We do an algebraic observability analysis to support the design of this approach. The filter is implemented as an EKF that works in a frame that follows the road. A number of real outdoor experiments were carried out to evaluate the method in terms of localization accuracy and consistency.

The paper is organized as follows. Section 2 describes the system modeling, in particular the frames and GNSS fix error modeling. The observability of the proposed state space modeling is studied in Section 3. Section 4 describes the RC EKF implementation. Outdoor real experimental results are presented and analyzed in Section 6. Section 7 concludes the paper.

## 2. SYSTEM MODELING

### 2.1. Definition of Frames

A local East, North, Up (ENU) frame is defined as a Cartesian coordinate system tangential to the Earth’s ellipsoid at an origin point close to the navigation area. The north axis is tangential to the meridian that passes through the origin in a northerly direction. The east axis is normal to the north axis and is in the positive longitudes direction. The Up axis is chosen so that the ENU is a right-handed coordinate system (see Figure 3). This ENU frame is defined as



**Figure 3.** Frames used: ECEF (Earth Centered Earth Fixed), ENU, vehicle body, and camera.  $\lambda$ ,  $\varphi$ , and  $h$  refer, respectively, to longitude, latitude, and altitude.

the local navigation frame. GNSS receivers usually provide geographical data ( $\lambda$ ,  $\varphi$ ,  $h$ ) in the WGS84 system.

When the navigation area is flat enough, the 3D ENU frame can be simplified into 2D coordinates by only considering easting and northing. The lower frame  $R_O$  in Figure 3 is the 2D ENU frame with  $x_O$  pointing east and  $y_O$  pointing north. The lane marking map is also defined in  $R_O$ . As shown in Figure 3, camera-relative measurement is done with respect to an absolute navigation frame.

$R_M$  denotes the mobile vehicle frame ( $x_M$  is the longitudinal axis pointing forward and  $y_M$  is such that  $z_M$  is upwards). Point  $C$ , the origin of the camera frame  $R_C$ , is located at the front of the vehicle, since camera systems will often autocalibrate. In order to stay consistent with vision system conventions,  $y_C$  is pointing to the right side of the vehicle. Even if the camera is located behind the windscreen with a position offset ( $C_x$ ,  $C_y$ ), every detected lane marking

is expressed in  $R_C$ .  $P_x$  refers to the translation between point  $M$  and the front bumper.

### 2.2. Dead Reckoning Kinematic Model

The linear velocity of each rear wheel is measured by the ABS speed sensors. Since the vehicle is front-wheel driven, the slippage of the rear wheels is neglected. In the following differential model, time is omitted for simplification:

$$\begin{cases} \dot{x} = v^m \cdot \cos\psi \\ \dot{y} = v^m \cdot \sin\psi \\ \dot{\psi} = \omega^m - \varepsilon_\omega \\ \dot{\varepsilon}_\omega = 0, \end{cases} \quad (1)$$

where the linear velocity is calculated by  $v^m = (v_{rl}^m + v_{rr}^m)/2$ ;  $v_{rl}^m$  and  $v_{rr}^m$  denote the measured linear velocity of the left and right rear wheels, respectively. The angular velocity of

the vehicle ( $\omega^m$ ) is measured by the ESP yaw rate gyro.  $\varepsilon_\omega$  is the gyro bias modeled by a random constant. An estimate of the pose  $(x, y, \psi)$  of the vehicle is given by the integration of the measurements from a known initial pose.

### 2.3. Camera Observation Model

We consider a camera system that provides the following lane marking parameters of a Taylor's expansion of a clothoid in the camera frame (Kluge, 1994):

$$y = C_3 \cdot x^3 + C_2 \cdot x^2 + C_1 \cdot x + C_0, \quad (2)$$

where  $C_0$ ,  $C_1$ ,  $C_2$ , and  $C_3$  are, respectively, the cross-track distance, the slope, the curvature, and the curvature derivative of the detected lane marking (see Figure 3 for  $C_0$  and  $C_1$ ).

Since the lane marking map is represented by polylines, the only parameter that is considered for vehicle localization is the cross-track distance  $C_0$ . The heading  $C_1$  is too sensitive to movements of the vehicle body.

In Figure 3, let  $L$  denote the lane marking detection located at ordinate  $C_0$  in  $R_C$ . The coordinates of point  $L$  in frame  $R_O$  are

$$\begin{bmatrix} x_L \\ y_L \end{bmatrix} = \begin{bmatrix} P_x \cdot \cos\psi + C_0 \cdot \sin\psi + x \\ P_x \cdot \sin\psi - C_0 \cdot \cos\psi + y \end{bmatrix}. \quad (3)$$

In Figure 3,  $[AB]$  represents the detected lane marking segment. The coordinates of points  $A$  and  $B$  are  $(x_A, y_A)$  and  $(x_B, y_B)$  in  $R_O$ . Let  $V = (x_{AB} = x_B - x_A, y_{AB} = y_B - y_A)^T$ . Point  $L$  on segment  $[AB]$  is such that

$$\begin{cases} x_L = x_A + \lambda \cdot x_{AB} \\ y_L = y_A + \lambda \cdot y_{AB} \end{cases} \quad \text{with } \lambda \in [0, 1]. \quad (4)$$

Plugging Eq. (3) into Eq. (4), we have

$$\begin{cases} P_x \cdot \cos\psi + C_0 \cdot \sin\psi + x = x_A + \lambda \cdot x_{AB} \\ P_x \cdot \sin\psi - C_0 \cdot \cos\psi + y = y_A + \lambda \cdot y_{AB} \end{cases}. \quad (5)$$

Through Eq. (5), we derive

$$C_0 = \frac{(P_x \cdot \sin\psi + y - y_A) \cdot x_{AB} - (P_x \cdot \cos\psi + x - x_A) \cdot y_{AB}}{x_{AB} \cdot \cos\psi + y_{AB} \cdot \sin\psi} \quad (6)$$

The camera observation model takes into account explicitly the camera position in the body frame. It is nonlinear and can be written as

$$\mathbf{y}_{\text{cam}} = g_{\text{cam}}(\mathbf{x}, \mathbf{l}) + \boldsymbol{\beta}_{\text{cam}}, \quad (7)$$

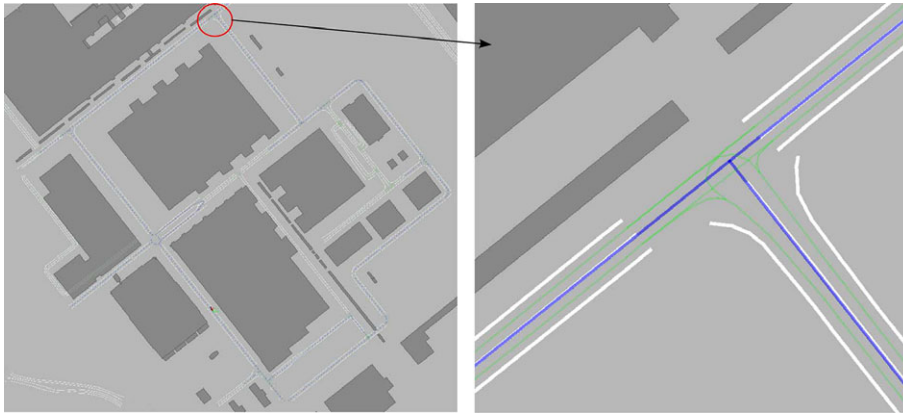
where  $\mathbf{y}_{\text{cam}} = [C_0]$  is the camera measurement,  $\mathbf{x}$  is the state vector of the vehicle,  $\mathbf{l}$  is the lane marking parameters extracted from the digital map, and  $\boldsymbol{\beta}_{\text{cam}}$  is a measurement noise that is centered when the camera is well calibrated intrinsically and extrinsically (Bouguet, 2008).

### 2.4. Road Feature Maps

Detailed maps provide prior information to navigation tasks and can provide a self-driving capability that is low cost in terms of onboard hardware equipment. In this section, we discuss relevant previous works.

Different kinds of maps have been considered in the literature. In Laneurit et al. (2006), a map was used that comprised a grid of rectangular facets representing the road-sides. The origin, orientation, length, and width of each facet are defined in the local navigation frame. Using a front view camera, an image processing algorithm is then implemented to give precise cross-track position and orientation of the vehicle with respect to the roadside. It corresponds to the parameters of the nearest facet. This work shows that it is possible and potentially useful to integrate lane boundary information, which often means detecting white lane markings. In Miller et al. (2011), the map is modeled by GNSS waypoints designating lane centers, stop lines, and lane markings. Dashed, solid, and unstructured lane types are denoted for each lane marking. In this approach,  $\mathbf{y}_{\text{cam}}$  contains the perpendicular distance of the camera from each of the lane boundaries, the camera heading with respect to the lane, the lane width, the lane type, and the distance to the stop line. The authors use a particle filter to augment the GNSS/DR solution with camera measurements coupled with a surveyed map. They give a demonstration of the stability of the localization solutions used to feedback the controller of a full-size AGV. Recently, other works with similar ideas but different configurations have been carried out, such as in Gruyer, Belaroussi, and Revilloud (2014), Jo et al. (2013), and Lee et al. (2015). Section 2.5.3 describes some research works. In GNSS error modeling (Jo et al., 2013), a front view camera is used to detect lane markings and zebra lines that have been charted in the map. In Gruyer et al. (2014), two cameras are used to provide an assessment of the cross-track distances between the vehicle and the ego-lane markings or borders. This information is then coupled with an accurate digital map of the road markings. However, the modeling of GNSS errors is not considered. Lee et al. (2015) propose, in particular, a localization method with GNSS/DR error estimation based on lane detection for curved lane models. They propose an along-track measurement from a curve matching. The curved parameter of the lane ahead is fitted using the waypoint map by supposing that the waypoints and the lane markings are parallel. The fitted waypoint curve is then transformed into the vehicle frame. However, the impact of the uncertainty of the estimated vehicle heading on this transformation (which introduces errors on the along-track measurement) is not discussed.

The lane marking map used in the present paper was the result of a surveying operation carried out by a private company. The mobile mapping system was equipped with a RIEGL VQ450 lidar with millimeter-level accuracy. The



**Figure 4.** Lane marking map of the test area. Blue line: road centerline; green line: lane centerline; white line: lane marking; gray box: building.

mobile mapping platform itself was localized using a post-processed kinematic GPS. The absolute error of the collected lane marking data is estimated to be in the order of 10 cm. Figure 4 shows the lane marking map of the navigation area. The gray boxes represent the buildings imported from OpenStreetMap<sup>1</sup>. The blue, green, and white lines, respectively, represent the road centerlines, the lane centerlines, and the lane markings.

## 2.5. GNSS Fix Error Model

GNSS is mandatory for initialization of the localization system, and GNSS information should be used as much as possible as long as it is consistent with the pose estimate. Where there are no lane markings in a navigation area, or where the camera fails to detect lane markings (for instance, at an intersection), GNSS can improve the accuracy of the pose information and compensate for DR errors. Moreover, since the camera observations provide only cross-track correction, where the vehicle is traveling on a long straight road, the along-track drift of a map-aided dead-reckoned estimate can become significant. The GNSS information can again be used to correct the along-track drift. We augment the state vector with shaping components so that data from GNSS fixes can be merged.

### 2.5.1. GNSS Position Fix Errors

Loosely coupling L1-GNSS with other sensors is a challenging task since GNSS positioning errors (PEs) are not white and can be affected by strong biases and multipath, particularly in urban areas. Figure 5 illustrates these issues on a real test with a low-cost L1-GNSS receiver: PEs can be as much as several meters, are strongly correlated and can have jumps within short time intervals. The solution that

we propose is to model the correlation to compensate for the biases and to reject fixes suffering from multipath effects. The GNSS error signals  $(\beta_x, \beta_y)$  were obtained using a ground truth system.

Autocorrelations of three different sequences of  $(\beta_x, \beta_y)$  produced by the same GNSS receiver (1000 samples each at 5 Hz) are shown in Figure 6. As the shape is clearly different from a Delta-Dirac at zero, errors are colored. Moreover, for short correlation times (less than 30 s), the different curves superimpose quite well, which indicates a repeatable behavior that can be modeled. Therefore, L1-GNSS errors can be modeled by zero-mean white noise that is put through shaping filters to yield an output statistically similar to the error under consideration.

The problem is to find a structure for the filter and then to estimate its parameters and the variance of the driving noise.

### 2.5.2. Autoregressive Processes for Modeling GNSS Colored Errors

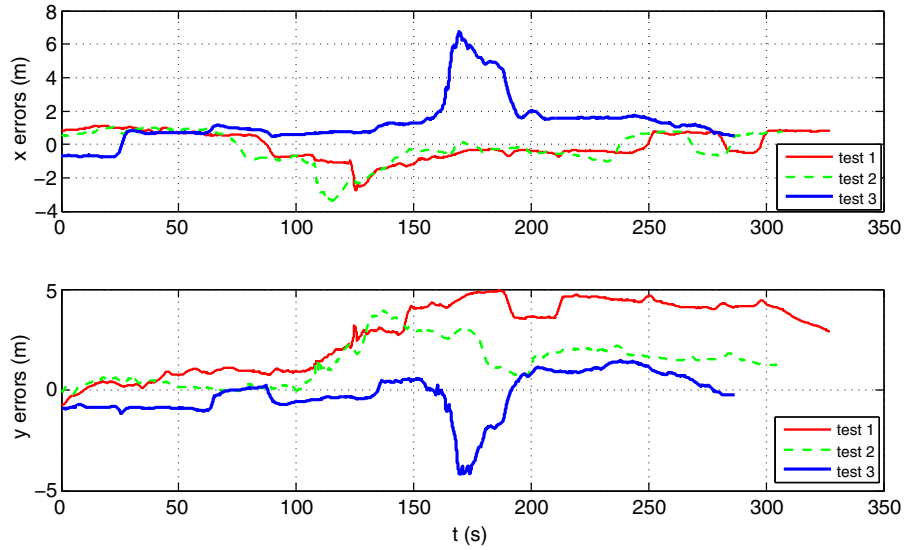
An autoregressive (AR) model is a representation of a type of random process. The name autoregressive comes from the fact that each signal sample is regressed on the previous values of itself. The AR process can be described using a discrete pole-zero transfer function system  $H(z)$  as follows:

$$H(z) = \frac{O(z)}{I(z)} = \frac{\beta_0}{1 + \sum_{n=1}^p \alpha_n z^{-n}}, \quad (8)$$

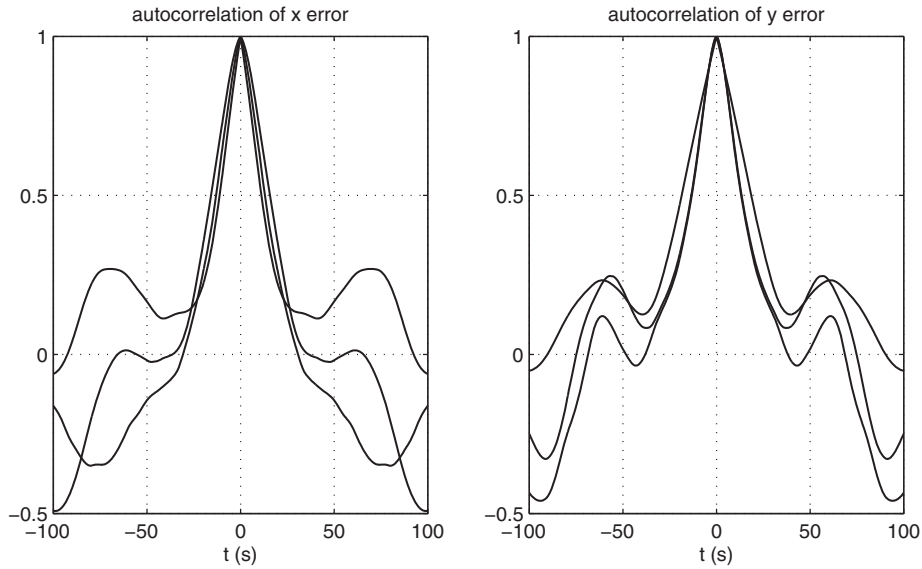
where  $I(z)$  is the  $z$ -transform of the input  $I_k$ , and  $O(z)$  is the  $z$ -transform of the output  $O_k$ .  $\beta_0^2$  represents the estimated variance of the white noise input to the AR model, and  $p$  is the AR order.

Applying the inverse  $z$ -transform to Eq. (8), the AR transfer function in the time domain can be obtained as

<sup>1</sup>www.openstreetmap.org



**Figure 5.** Errors of L1-GNSS solutions in the local navigation frame. The vehicle was manually driven up to 30 km/h with an RTK-GPS IMU for ground truth.



**Figure 6.** Autocorrelation of three different sequences. The shapes indicate that the GNSS error is colored and its behavior is repeatable between tests.

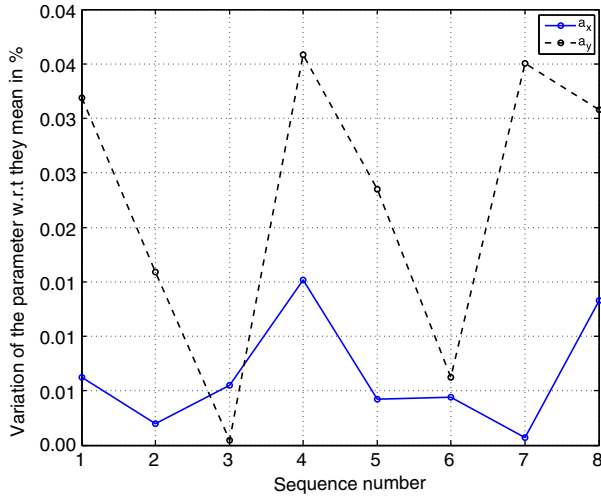
$$O_k = -\sum_{n=1}^p \alpha_n O_{k-n} + \beta_0 I_k. \tag{9}$$

In Nassar (2003), the AR processes are first used to model the randomness of the inertial sensor measurements. The residual random error component is modeled as a zero-mean white noise put through a shaping filter to yield a time-correlated output. In the present work, we suggest

using AR processes to model GNSS random errors. In our case, Eq. (9) becomes

$$\begin{cases} \varepsilon_{x,k} = -\sum_{n=1}^p \alpha_{x,n} \varepsilon_{x,k-n} + w_{x,k} \\ \varepsilon_{y,k} = -\sum_{n=1}^p \alpha_{y,n} \varepsilon_{y,k-n} + w_{y,k} \end{cases}, \tag{10}$$

where  $\varepsilon_x$  and  $\varepsilon_y$  are the biases on the GNSS position fix, and  $w_x$  and  $w_y$  are the input white noises.



**Figure 7.** Relative variation in AR1 parameters for eight different estimation sequences (Burg’s method).

We now look at how to determine the AR model parameters ( $\alpha_{x,n}$  and  $\alpha_{y,n}$ ), the input white noise ( $w_x$  and  $w_y$ ), and the AR order  $p$ .

Different methods exist for estimating AR parameters. Three methods are quite common, namely Yule–Walker, the covariance method, and Burg’s method (Nassar, 2003). The Yule–Walker method first determines the sample autocorrelation sequence of the input signal (GNSS residual bias), and the AR model parameters are then optimally computed by solving a set of linear normal equations in a least-squares sense. However, the Yule–Walker method performs adequately only for very long data records (Jackson, 1996), and it may introduce a large bias in the AR estimated coefficients, since it does not guarantee a stable solution of the model. The covariance method is similar to the Yule–Walker method in that it minimizes the prediction in the least-squares sense.

Burg’s method was introduced to overcome most of the drawbacks of the other modeling techniques by providing both stable resolution and high resolution, especially for short data records (Burg, 1975). Burg’s method attempts to use the data as fully as possible, by defining both a forward and a backward prediction error term.

In the present work, Burg’s method is adopted to estimate the AR parameters and the variance of the input white noise of the shaping filter.

The choice of the order of the AR model is also of importance.

Figures 7 and 8 show the variation of the estimated parameters of first- and second-order models (denoted AR1 and AR2 parameters, respectively) on eight different sequences acquired at speeds up to 50 km/h. The variation in the AR1 parameters is at most 0.06%. However, the

variation in the AR2 parameters is almost 50%. Although the sequences that we used to identify the parameters are probably too short to estimate the AR2 parameters correctly, an AR1 model clearly has better stability. Our strategy is therefore to use an AR1 filter that is more robust to the nonstationarity of the errors.

In order to validate this shaping model, the recurrence equations may be reversed in offline processing:

$$\begin{cases} w_{x,k} = \varepsilon_{x,k} + \alpha_x \varepsilon_{x,k-1} \\ w_{y,k} = \varepsilon_{y,k} + \alpha_y \varepsilon_{y,k-1} \end{cases} \quad (11)$$

The autocorrelations of  $w_{x,k}$  and  $w_{y,k}$  are given in Figure 9 (red curves). They approximate a Delta-Dirac function quite well. We conclude that these signals can be seen as white noise sequences with respect to the dynamics of our system, and the AR1 model defined by Eq. (11) is a good shaping filter.

### 2.5.3. GNSS Error Models

When using GNSS as a sensor in a loosely coupled fusion architecture, the position fix estimates ( $x_{\text{GNSS}}$ ,  $y_{\text{GNSS}}$ ) in the working frame are affected by errors ( $\beta_x$ ,  $\beta_y$ ). Neglecting the lever arm of the antenna for simplification, we have

$$\begin{cases} x_{\text{GNSS}} = x + \beta_x \\ y_{\text{GNSS}} = y + \beta_y \end{cases} \quad (12)$$

The errors ( $\beta_x$ ,  $\beta_y$ ) cannot be considered as zero-mean white noise in the way that was discussed above. The position fix of a GNSS receiver is affected by time-correlated errors caused by atmospheric effects and by the filter implemented in the receiver. Modeling of GNSS biases has been discussed in a number of publications, including Laneurit, Chapuis, and Chausse (2006), Clanton, Bevely, and Hodel (2009), and Miller et al. (2011).

The nonmodeled part of the error can be expressed by some random process  $\beta'_x$  and  $\beta'_y$ , assumed to be zero-mean white noise:

$$\begin{cases} x_{\text{GNSS}} = x + \varepsilon_x + \beta'_x \\ y_{\text{GNSS}} = y + \varepsilon_y + \beta'_y \end{cases} \quad (13)$$

where  $\varepsilon_x$  and  $\varepsilon_y$  are slow-variation errors in the ENU frame for which an evolution model exists. Some authors, such as Laneurit et al. (2006) and Lee et al. (2015), suppose that these errors are quite constant between two samples:

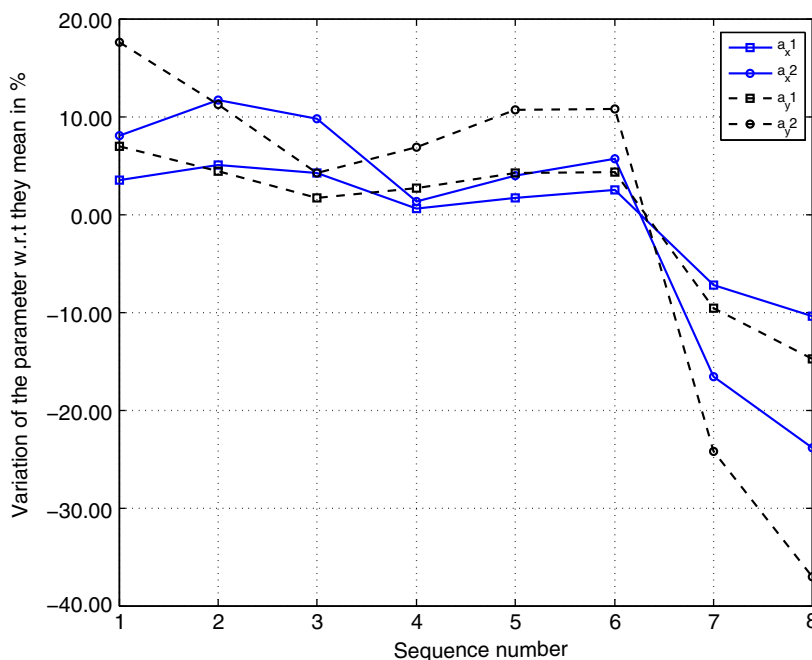
$$\begin{cases} \varepsilon_{x,k} = \varepsilon_{x,k-1} \\ \varepsilon_{y,k} = \varepsilon_{y,k-1} \end{cases} \quad (14)$$

In Laneurit et al. (2006), a bias management strategy is proposed to decide whether the bias has changed.

In Miller et al. (2011) and Jo et al. (2013), the GNSS biases are modeled by an AR1 process as proposed in Bar-shalom et al. (2002):

$$\begin{cases} \varepsilon_{x,k} = e^{-dt/\tau_x} \cdot \varepsilon_{x,k-1} + v_{x,k-1} \\ \varepsilon_{y,k} = e^{-dt/\tau_y} \cdot \varepsilon_{y,k-1} + v_{y,k-1} \end{cases} \quad (15)$$





**Figure 8.** Relative variation in AR2 parameters for eight different estimation sequences (Burg’s method).

where  $\tau_x$  and  $\tau_y$  are the bias autocorrelation time constants and  $dt$  is the elapsed time.  $v_x$  and  $v_y$  are zero-mean white noise. The vehicle dynamic equations are then augmented by Eq. (14) or (15).

In our work, we use a combination of the random constant process defined by Eq. (14) and the AR model defined by Eq. (15) to enhance the GNSS error modeling. This choice is motivated by the road-frame implementation presented hereafter.

## 2.6. Road-Oriented Frame Modeling of the Localization Problem

The working frame in which a localization solver is implemented can play an important role in terms of modeling and estimation performance. Often a local ENU frame is used but, since we are using a camera that is able to measure the cross-track distance with respect to known lane markings, we propose modeling the system in a road-oriented Cartesian frame.

A road-oriented frame  ${}^i x^i y$  (see Figure 10) is defined to have the same origin as the local ENU frame, and to have its  $x$ -axis pointing in the direction of the road  $i$  on which the vehicle is traveling.

As the vehicle uses DR sensors (yaw rate gyro and wheel speeds) and L1-GNSS fixes affected by non-white noise, the state vector contains different terms for the GNSS errors:

$$\mathbf{x} = [x, y, \psi, \varepsilon_\omega, \varepsilon_{x1}, \varepsilon_{x2}, \varepsilon_{y1}, \varepsilon_{y2}]^T, \quad (16)$$

where  $(x, y, \psi)$  is the 2D pose of the vehicle;  $\varepsilon_\omega$  denotes the gyro bias; and  $(\varepsilon_{x1}, \varepsilon_{x2}, \varepsilon_{y1}, \varepsilon_{y2})$  are GNSS errors on  $x$  and  $y$  in the road-oriented frame that are split into different components, as in Bar-shalom et al. (2002).

The proposed evolution model of the state vector is given by

$$\begin{cases} \dot{x} = v \cdot \cos\psi \\ \dot{y} = v \cdot \sin\psi \\ \dot{\psi} = \omega - \varepsilon_\omega \\ \varepsilon_\omega = 0 \\ \dot{\varepsilon}_{x1} = -\varepsilon_{x1}/\tau_1 \\ \dot{\varepsilon}_{x2} = -\varepsilon_{x2}/\tau_2 \\ \dot{\varepsilon}_{y1} = -\varepsilon_{y1}/\tau_1 \\ \dot{\varepsilon}_{y2} = 0 \end{cases} \quad (17)$$

In this model, first-order AR models with time constants  $\tau_1$  and  $\tau_2$  are used to model the nonwhiteness of the GNSS errors. The error in the  $x$ -direction is split into two components ( $\varepsilon_{x1}$  and  $\varepsilon_{x2}$ ) with different decorrelation time constants in order to manage the frame transformation when the road changes (a detailed explanation is given in Section 4.1). The time constant of  $\varepsilon_{y1}$  is the same as  $\varepsilon_{x1}$ .  $v$  is the linear velocity measured by the wheel speed sensors, and  $\omega$  is the angular velocity measured by the yaw rate gyro. The last equation of the model associated with  $\varepsilon_{y2}$  plays an important role in our localizer. It is a random constant model, as used in Lateurit et al. (2006), and as such it is well adapted to quickly estimate the cross-track bias of the GNSS fix in the road frame.

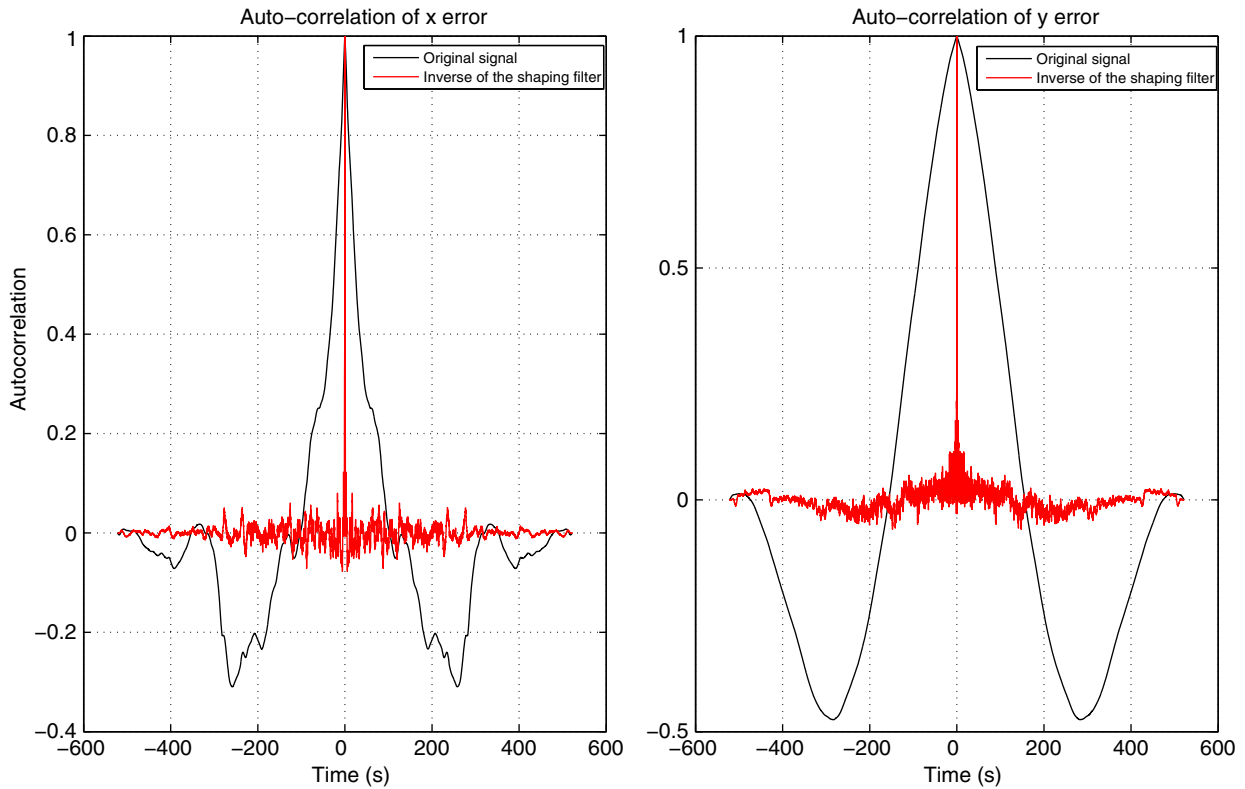


Figure 9. Autocorrelations of the experimental signal and of the inverse of the shaping filter.

### 3. OBSERVABILITY ANALYSIS IN THE ROAD FRAME

Observability is a necessary condition for any filtering algorithm to converge to an unbiased state estimate. In this section, the following question needs to be answered: are all the components of the state vector observable when using the exteroceptive measurements  $\mathbf{y} = [x_{\text{GNSS}}, y_{\text{GNSS}}, C_0]$  (the GNSS fix and the cross-track measure of the forward-looking camera that detects lane markings) and the proprioceptive measurements  $\mathbf{u} = [v, w]$ ?

The exteroceptive sensors that are considered are a GNSS receiver providing position fixes and a forward-looking camera that detects lane markings. To examine the structural properties of the modeling using equations that are easy to handle, let us suppose that the camera and the GNSS antenna coincide with point  $M$ , the origin of the body frame (see Figure 11).

At this stage, we consider that there is only one lane marking, locally represented by a line  $[AB]$ . The observation model in this case is given by

$$C_0 = (y - y_A) / \cos\psi, \tag{18}$$

where  $C_0$  is the cross-track distance measured by the camera in the body frame (Tao, Bonnifait, Frémont, & Ibañez-

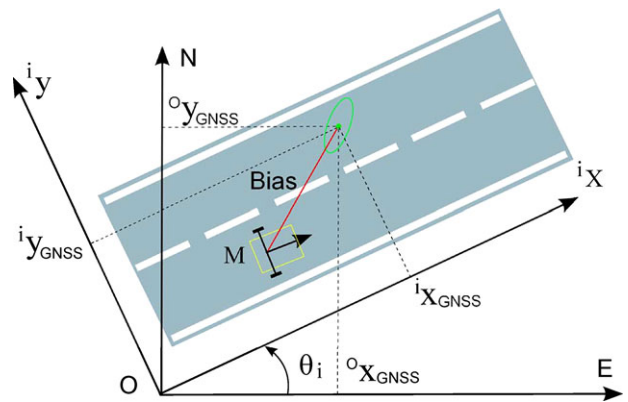
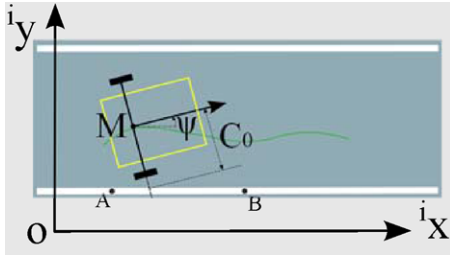


Figure 10. Road frame : the  $x$ -axis is chosen to be parallel to the road. GNSS fix uncertainty and bias are shown as the green ellipse and the red line, respectively.

Guzman, 2013).  $y_A$  is the ordinate of point  $A$  in the road-oriented frame. Figure 11 illustrates the simplification.

The GNSS fixes with their shaping errors are linked to the state by the following model:

$$\begin{cases} x_{\text{GNSS}} = x + \varepsilon_{x1} + \varepsilon_{x2} \\ y_{\text{GNSS}} = y + \varepsilon_{y1} + \varepsilon_{y2} \end{cases} \tag{19}$$



**Figure 11.** Bird's eye view in the road-oriented frame.  $i_x - i_y$  indicates the working frame along road  $i$ . The green line is the vehicle trajectory.  $\psi$  is the vehicle heading in the road-oriented frame. The camera is assumed to be located at the origin (point M) of the body frame for a simplified observation model defined by Eq. (18).

### 3.1. Algebraic Observability

We are dealing with a nonlinear system, and in such cases there are two main approaches to studying the observability of the state. The classical approach is local weak observability (Hermann & Krener, 1977) that relies on the study of a rank condition after linearization and the computation of Lie derivatives. There is another approach based on differential algebra, which is largely the work of Ritt (1950). Differential algebra was introduced into control theory through the works of Fliess, Glad, and Ljung (Fliess, 1989; Fliess & Glad, 1993; Glad & Ljung, 1990; Ljung & Glad, 1994). Algebraic observability can be expressed as follows (see Sert, Perruquetti, Kokosy, Jin, & Palos, 2012).

The state of a system with known internal dynamics is said to be observable if, and only if, there is an algebraic equation linking the state vector to the measured output  $\mathbf{y}$  and input  $\mathbf{u}$  and a finite number of their time derivatives. If a state component  $x_1$  is observable and another state component  $x_2$  can be expressed by an algebraic function of  $x_1$ ,  $\mathbf{y}$ ,  $\mathbf{u}$  and their derivatives, then  $x_2$  is also observable.

Algebraic observability is therefore a different way of studying observability. It has the advantage of providing a closed form for building a state observer in cases where the derivatives of the inputs and outputs can be estimated with a good quality. It is equivalent to local generic observability (Diop & Wang, 1993).

Eqs. (17) and (18) are analytic but not algebraic. A solution is to define an equivalent algebraic system (with algebraic functions) by introducing auxiliary state variables (Chatzis, Chatzi, & Smyth, 2015).

Let us define two auxiliary state variables:

$$\begin{cases} x_1 = \cos\psi \\ x_2 = \sin\psi \end{cases} \quad (20)$$

Since

$$\dot{x}_1 = -\dot{\psi} \cdot \sin\psi = -(\omega - \varepsilon_\omega) x_2$$

and

$$\dot{x}_2 = \dot{\psi} \cdot \cos\psi = (\omega - \varepsilon_\omega) x_1,$$

Eqs. (17) and (18) are complemented with two additional equations and  $\dot{\psi}$  is removed:

$$\begin{cases} \dot{x} = v \cdot x_1 \\ \dot{y} = v \cdot x_2 \\ \dot{x}_1 = -x_2(\omega - \varepsilon_\omega) \\ \dot{x}_2 = x_1(\omega - \varepsilon_\omega) \\ \dot{\varepsilon}_\omega = 0 \\ \dot{\varepsilon}_{x1} = -\varepsilon_{x1}/\tau_1 \\ \dot{\varepsilon}_{x2} = -\varepsilon_{x2}/\tau_2 \\ \dot{\varepsilon}_{y1} = -\varepsilon_{y1}/\tau_1 \\ \dot{\varepsilon}_{y2} = 0 \end{cases} \quad (21)$$

and

$$C_0 = (y - y_A)/x_1. \quad (22)$$

All the functions are now rational.

Below we examine the observability of the different components of the state in the algebraic framework. We first look at the observability of the vehicle heading and gyro bias, and then at vehicle position and GNSS biases.

### 3.2. Observability of the Vehicle Heading $\psi$

Taking the derivative of Eq. (22), we have

$$\dot{y} = \dot{C}_0 \cdot x_1 - \dot{\psi} \cdot C_0 \cdot x_2. \quad (23)$$

Plugging  $\dot{y} = v \cdot x_2$  and  $\dot{\psi} = \omega - \varepsilon_\omega$ :

$$v \cdot x_2 = \dot{C}_0 \cdot x_1 - (\omega - \varepsilon_\omega) \cdot C_0 \cdot x_2. \quad (24)$$

Now, taking now the derivative of Eq. (24), we have

$$\begin{aligned} [\dot{v} + 2\dot{C}_0(\omega - \varepsilon_\omega) + \dot{\omega} \cdot C_0] x_2 = [\ddot{C}_0 - C_0(\omega - \varepsilon_\omega)^2 \\ - v(\omega - \varepsilon_\omega)] x_1. \end{aligned} \quad (25)$$

If  $\psi$  is identically null (the vehicle is traveling parallel to the lane marking), then  $\dot{\psi} = 0$  and we have  $\varepsilon_\omega = \omega$ . The gyro bias is then observable. Moreover, in this case, we have  $\dot{C}_0$  that is identically null, and so it can be seen that  $\psi$  is identically null.

Now, suppose that  $\psi$  is not null. Consequently,  $x_2 \neq 0$ . From Eq. (24) we can work out

$$\varepsilon_\omega = \frac{v \cdot x_2 - \dot{C}_0 \cdot x_1}{C_0 \cdot x_2} + \omega, \quad (26)$$

where  $C_0$  is physically nonnull because the lane markings are on the sides of the lane.

Plugging Eq. (26) into Eq. (25) and noting that  $x_1 = (1 - x_2^2)^{1/2}$ , we obtain an implicit algebraic function:

$$\begin{aligned} (\dot{v} \cdot C_0 - v \cdot \dot{C}_0 + \dot{\omega} \cdot C_0^2) x_2^3 + (\ddot{C}_0 \cdot C_0 - \dot{C}_0^2) (1 - x_2^2)^{3/2} \\ + (2\dot{C}_0^2 - \ddot{C}_0 \cdot C_0) (1 - x_2^2)^{1/2} - v \cdot \dot{C}_0 \cdot x_2 = 0. \end{aligned} \quad (27)$$

From this expression, we can get an algebraic equation. A demonstration is given in Tao (2016).

$$x_2 = \Phi_{x_2}(C_0, \dot{C}_0, \ddot{C}_0, v, \dot{v}, \omega, \dot{\omega}). \quad (28)$$

So,  $x_2$  is observable.  $\psi$  is determined by  $\psi = \arcsin(x_2)$ .

$$x_1 = \sqrt{1 - x_2^2} = \Phi_{x_1}(C_0, \dot{C}_0, \ddot{C}_0, v, \dot{v}, \omega, \dot{\omega}) \quad (29)$$

is observable.

There is a particular situation when the trajectory is strictly parallel to the road. In this case,  $\psi$  is identically null and then  $\dot{C}_0$  is identically null. Therefore,  $\psi$  is seen to be identically null, and so  $\psi$  is observable.

### 3.3. Observability of the Gyro Bias $\varepsilon_\omega$

Using Eqs. (29) and (28) in Eq. (26), we obtain a function  $\Phi_{\varepsilon_\omega}$  which gives  $\varepsilon_\omega$ :

$$\varepsilon_\omega = \Phi_{\varepsilon_\omega}(C_0, \dot{C}_0, \ddot{C}_0, v, \dot{v}, \omega, \dot{\omega}). \quad (30)$$

$\varepsilon_\omega$  is therefore observable.

### 3.4. Observability of Vehicle Position ( $x, y$ ) and of GNSS Biases

Taking the derivative of Eq. (19), we have

$$\dot{x}_{\text{GNSS}} = v \cdot x_1 - \varepsilon_{x1}/\tau_1 - \varepsilon_{x2}/\tau_2. \quad (31)$$

Similarly, for Eq. (31):

$$\ddot{x}_{\text{GNSS}} = \dot{v} \cdot x_1 - v \cdot x_2 \cdot (\omega - \varepsilon_\omega) + \varepsilon_{x1}/\tau_1^2 + \varepsilon_{x2}/\tau_2^2. \quad (32)$$

Since  $\psi$  is observable, and since we have a linear system with two unknowns and two equations, we have

$$\varepsilon_{xi} = \Phi_{\varepsilon_{xi}}(C_0, \dot{C}_0, \ddot{C}_0, \dot{x}_{\text{GNSS}}, \ddot{x}_{\text{GNSS}}, v, \dot{v}, \omega, \dot{\omega}) \quad i = 1, 2. \quad (33)$$

So  $\varepsilon_{x1}$  and  $\varepsilon_{x2}$  are observable. If  $\tau_1 = \tau_2$ , then  $\varepsilon_{x1}$  and  $\varepsilon_{x2}$  are not solvable by Eqs. (19) and (31), but  $(\varepsilon_{x1} + \varepsilon_{x2})$  is observable.

With  $x = x_{\text{GNSS}} - (\varepsilon_{x1} + \varepsilon_{x2})$  and Eq. (33),  $x$  is observable. With  $y = y_A + C_0 \cdot x_1$  and Eq. (29),  $y$  is observable.

The expression of  $\varepsilon_{y1}$  is given as follows:

$$\dot{y}_{\text{GNSS}} = v \cdot x_2 - \varepsilon_{y1}/\tau_1 \quad (34)$$

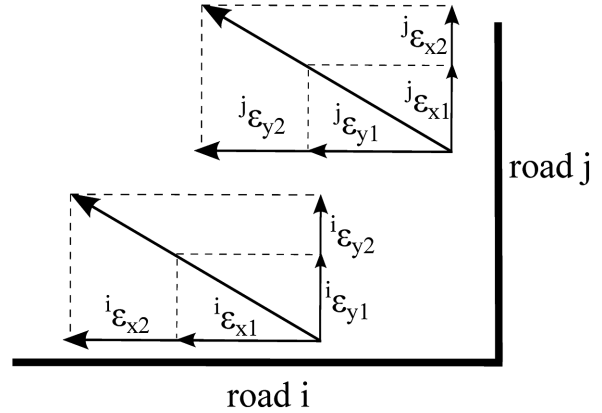
$$\varepsilon_{y1} = (v \cdot x_2 - \dot{y}_{\text{GNSS}}) \tau_1. \quad (35)$$

So  $\varepsilon_{y1}$  is observable.

With  $\varepsilon_{y2} = y_{\text{GNSS}} - y - \varepsilon_{y1}$ , we can derive that  $\varepsilon_{y2}$  is observable.

### 3.5. Observability: Conclusion

So far, we have proved that every element in the state vector  $\mathbf{x}$  can be expressed by an algebraic function of components of  $\mathbf{y}$  and  $\mathbf{u}$  and a finite number of their derivatives. We



**Figure 12.** Geometrical transformation of the GNSS errors. The estimated error remains unchanged. However, its components on  $x$  and  $y$  are transformed from road  $i$  to road  $j$ .

can consequently conclude that the state vector with its associated state space is observable, as long as the vehicle moves or accelerates.

## 4. ROAD-CENTERED EXTENDED KALMAN FILTER

The observability of the state has been demonstrated in a road-oriented frame. However, in real conditions, the orientation of the road changes as the vehicle moves from one road to another. We are now building an EKF that estimates the pose vector of the vehicle from one road to another, in a sequential way.

### 4.1. Geometrical Transformation

$R_O$  denotes the local ENU frame, and  $R_i$  is the working road-oriented frame, with its  $x$ -axis pointing in the direction of road  $i$ . When the vehicle moves from road  $i$  to road  $j$ , the working frame changes from  $R_i$  to  $R_j$ . Let  ${}^j\mathbf{x}$  denote the state vector in frame  $R_j$ :

$${}^j\mathbf{x} = [{}^jx, {}^jy, {}^j\psi, {}^j\varepsilon_\omega, {}^j\varepsilon_{x1}, {}^j\varepsilon_{x2}, {}^j\varepsilon_{y1}, {}^j\varepsilon_{y2}]^T. \quad (36)$$

The transformation from  ${}^i\mathbf{x}$  to  ${}^j\mathbf{x}$  is given by Eq. (37):

$$\begin{cases} {}^jx = {}^ix \cdot \cos\alpha + {}^iy \cdot \sin\alpha \\ {}^jy = -{}^ix \cdot \sin\alpha + {}^iy \cdot \cos\alpha \\ {}^j\psi = {}^i\psi - \alpha \\ {}^j\varepsilon_\omega = {}^i\varepsilon_\omega \\ {}^j\varepsilon_{x1} = {}^i\varepsilon_{x1} \cdot \cos\alpha + {}^i\varepsilon_{y1} \cdot \sin\alpha \\ {}^j\varepsilon_{x2} = {}^i\varepsilon_{x2} \cdot \cos\alpha + {}^i\varepsilon_{y2} \cdot \sin\alpha \\ {}^j\varepsilon_{y1} = -{}^i\varepsilon_{x1} \cdot \sin\alpha + {}^i\varepsilon_{y1} \cdot \cos\alpha \\ {}^j\varepsilon_{y2} = -{}^i\varepsilon_{x2} \cdot \sin\alpha + {}^i\varepsilon_{y2} \cdot \cos\alpha \end{cases}, \quad (37)$$

where  $\alpha = \theta_j - \theta_i$ ,  $\theta_i$ , and  $\theta_j$  are, respectively, the orientations of road  $i$  and road  $j$  in  $R_O$  (cf. Figure 10).

Figure 12 illustrates how the estimated biases change when the two successive roads are orthogonal.

Let  ${}^i P$  denote the covariance matrix estimated by the EKF in the working frame  $R_i$ . The transformation from  $({}^i \mathbf{x}, {}^i P)$  to  $({}^j \mathbf{x}, {}^j P)$  is described by the function given in Algorithm 1, where  $s\alpha$  and  $c\alpha$  denote  $\sin\alpha$  and  $\cos\alpha$ , respectively. The road directions being deterministic, this is simply the linear transformation of a random vector.

The reason why the bias on  ${}^i x$  has been modeled by two components now becomes clear. Our aim is to model the cross-track bias on  ${}^i y$  by an AR process plus a random constant in order to get a better estimation process (we have seen that these two components are observable thanks to the camera measurements, when working in the road-oriented frame). If the bias on  ${}^i x$  is modeled by only one component, then there is no way of finding a bijective transformation when the frame changes from  $R_i$  to  $R_j$ . When the vehicle pose is converted from one road frame to another, doing the inverse transformation has to give the same estimate. Mathematically, this means that matrix  ${}^j H_i$  (see Algorithm 1) has to be squared such that  ${}^j H_i \cdot {}^i H_j = I$  (identity matrix). It is straightforward to check that our proposal satisfies this property.

---

#### Algorithm 1 Function $R_i 2 R_j\_State$

---

**Input:**  ${}^i \mathbf{x}, {}^i P, \theta_i, \theta_j$

1:  $\alpha = \theta_j - \theta_i$

$$2: {}^j H_i = \begin{bmatrix} c\alpha & s\alpha & 0 & 0 & 0 & 0 & 0 & 0 \\ -s\alpha & c\alpha & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c\alpha & 0 & s\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & c\alpha & 0 & s\alpha \\ 0 & 0 & 0 & 0 & -s\alpha & 0 & c\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & -s\alpha & 0 & c\alpha \end{bmatrix}$$

3:  ${}^j b_i = [0, 0, -\alpha, 0, 0, 0, 0, 0]^T$

4:  ${}^j \mathbf{x} = {}^j H_i \cdot {}^i \mathbf{x} + {}^j b_i$

5:  ${}^j P = {}^j H_i \cdot {}^i P \cdot ({}^j H_i)^T$

**Output:**  ${}^j \mathbf{x}, {}^j P$

---

Since the output of the filter has to be given in the ENU frame  $R_O$ , the state transformation is performed using a function  $R_i 2 R_O\_State$ , which is constructed by replacing  $j$  in Algorithm 1 with  $O$ .  $\theta_O = 0$ .  $({}^O \mathbf{x}, {}^O P)$  denotes the Kalman filter estimates expressed in the ENU frame.

## 4.2. Implementation of Road-Centered EKF

The filter is implemented as a discrete EKF triggered by the proprioceptive sensors (typically at a sampling period  $Te = 0.01$  s). GNSS and camera measurements are used where they are available. The filter is described in Algorithm 2.  ${}^i A$  and  ${}^i B$  denote the coordinates of the detected lane marking  $[AB]$  in  $R_i$ .

---

#### Algorithm 2 Road-centred Extended Kalman filter

---

1:  $U = \text{Get}$  (proprioceptive sensor measurements)

2:  $({}^i \mathbf{x}, {}^i P) = \text{Predict}({}^i \mathbf{x}, {}^i P, U)$

3: **if** GNSS data is available **then**

4:  $({}^O \mathbf{y}_{GNSS}, {}^O R_{GNSS}) = \text{Get}$  (GNSS fix)

5:  $({}^i \mathbf{y}_{GNSS}, {}^i R_{GNSS}) = R_O 2 R_i ({}^O \mathbf{y}_{GNSS}, {}^O R_{GNSS}, \theta_i)$

6:  $({}^i \mathbf{x}, {}^i P) = \text{Update\_GNSS}({}^i \mathbf{x}, {}^i P, {}^i \mathbf{y}_{GNSS}, {}^i R_{GNSS})$

7: **end if**

8: **if** Camera data is available **then**

9:  $C_0 = \text{Get}$  (camera measurement)

10:  $({}^O \mathbf{x}, {}^O P) = R_i 2 R_O\_State({}^i \mathbf{x}, {}^i P, \theta_i)$

11:  $(j, {}^O A, {}^O B) = \text{Map\_Match}({}^O \mathbf{x}, {}^O P, C_0, \text{Map})$

12: **if**  $\theta_i \neq \theta_j$  **then**

13:  $({}^i \mathbf{x}, {}^i P) = R_i 2 R_j\_State({}^i \mathbf{x}, {}^i P, \theta_i, \theta_j)$

14:  $\theta_i = \theta_j$

15: **end if**

16:  ${}^i A = R_O 2 R_i\_Point({}^O A, \theta_i)$

17:  ${}^i B = R_O 2 R_i\_Point({}^O B, \theta_i)$

18:  $({}^i \mathbf{x}, {}^i P) = \text{Update\_Camera}({}^i \mathbf{x}, {}^i P, C_0, {}^i A, {}^i B)$

19: **end if**

20:  $({}^O \mathbf{x}, {}^O P) = R_i 2 R_O\_State({}^i \mathbf{x}, {}^i P, \theta_i)$

$\triangleright$ /\*System output\*/

21: Go to 1

$\triangleright$ /\*See Algorithms 1, 3 and 4\*/

---

#### 4.2.1. Prediction

When the proprioceptive sensors are available, the function  $\text{Predict}({}^i \mathbf{x}, {}^i P, \mathbf{u})$  consists in computing

$${}^i \mathbf{x} = f({}^i \mathbf{x}, \mathbf{u}) \iff \begin{cases} {}^i x = {}^i x + Te \cdot v \cdot \cos({}^i \psi) \\ {}^i y = {}^i x + Te \cdot v \cdot \sin({}^i \psi) \\ {}^i \psi = {}^i \psi + Te \cdot (\omega - {}^i \varepsilon_\omega) \\ {}^i \varepsilon_\omega = {}^i \varepsilon_\omega \\ {}^i \varepsilon_{x1} = a_1 \cdot {}^i \varepsilon_{x1} \\ {}^i \varepsilon_{x2} = a_2 \cdot {}^i \varepsilon_{x2} \\ {}^i \varepsilon_{y1} = a_1 \cdot {}^i \varepsilon_{y1} \\ {}^i \varepsilon_{y2} = {}^i \varepsilon_{y2} \end{cases} \quad (38)$$

and

$$\begin{cases} {}^i P = A \cdot {}^i P \cdot A^T + B \cdot N \cdot B^T + Q \\ A = \frac{\partial f({}^i \mathbf{x}, \mathbf{u})}{\partial {}^i \mathbf{x}}, B = \frac{\partial f({}^i \mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} \end{cases} \quad (39)$$

The measurement noises on  $v$  and  $\omega$  are assumed to be zero-mean-independent white noise.  $N$  denotes their covariance matrix.  $Q$  is the covariance matrix of the process noise  $a_1 = e^{-Te/\tau_1}$  and  $a_2 = e^{-Te/\tau_2}$ .

#### 4.2.2. GNSS Update

When a GNSS fix is available, the measurement vector  ${}^O \mathbf{y}_{GNSS} = ({}^O x_{GNSS}, {}^O y_{GNSS})$  in the ENU frame  $R_O$  is transformed to the working road-oriented frame  $R_i$  (see Figure 10) by Algorithm 3. The covariance matrix  ${}^O R_{GNSS}$  (typically given in the NMEA-0183 GST message) is also converted. Algorithm 3 shows the transformation.

**Algorithm 3** Function  $R_O2R_i$ **Input:**  ${}^O\mathbf{y}_{GNSS}, {}^OR_{GNSS}, \theta_i$ 

1:  ${}^iT_O = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix}$

2:  ${}^i\mathbf{y}_{GNSS} = {}^iT_O \cdot {}^O\mathbf{y}_{GNSS}$

3:  ${}^iR_{GNSS} = {}^iT_O \cdot {}^OR_{GNSS} \cdot ({}^iT_O)^T$

**Output:**  ${}^i\mathbf{y}_{GNSS}, {}^iR_{GNSS}$ 

A classical Kalman update step is then performed to update  $({}^i\mathbf{x}, {}^iP)$  with an innovation gating to reject the GNSS fix outliers (e.g., multipath on close buildings). For accurate data fusion, the level arm of the antenna with respect to the body frame has to be taken into account (see Tao et al., 2013, for details).

**4.2.3. Camera Update**

In order to update the filter, the map is used as it contains the coordinates of the lane marking. Map matching is therefore done when camera measurements are available.

For feature-based indoor localization, geometric constraints for data association can be classified as either location-independent constraints or location-dependent constraints (Arras, Castellanos, Schilt, & Siegart, 2003). For our purposes, the aim of map matching is to determine which lane marking segment  $[AB]$  (see Fig. 3) has been detected by the camera. We use the lane marking type and the orientation of the lane marking segment as location-independent constraints. As soon as a vehicle position is available, we utilize the distance between the estimated point  $L$  and the possible lane marking segment as a location-dependent constraint.

In a first stage, a set  $S$  of candidate segments is selected based on the following conditions.

- The lane marking type is consistent with the lane marking type reported by the camera.
- The orientation of the segment is close to the heading of the vehicle.
- The distance  $dist$  between point  $L$  and the candidate segment is less than the width of the road.

In a second stage, the segment  $s$  with the smallest  $dist$  is chosen as the map-matching result:

$$\text{Map\_matched} = \arg \min_{s \in \{S\}} \{dist\}.$$

The lane marking map used in this paper consists mainly of two-lane roadways with dashed lane markings in the center of the road, and solid lane markings on both sides of the road. In this case, the lane marking-type constraints are very good at distinguishing different lanes. This strategy works well in our tests.

Since the map is defined in  $R_O$ , the first step consists in converting the pose in the ENU frame. The algorithm checks

whether the vehicle is on a different road, in which case the road working frame is modified and the state with its covariance matrix is converted. It only remains to obtain the coordinates of the lane marking in the road frame described by Algorithm 4 and to update the state and covariance matrix by applying the estimation stage of the EKF. In practice, the location of the camera in the body frame is taken into account to get an accurate correction (see Tao & Bonnifait, 2016).

**Algorithm 4** Function  $R_O2R_i\_Point$ **Input:**  ${}^OA, \theta_i$ 

1:  ${}^iT_O = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix}$

2:  ${}^iA = {}^iT_O \cdot {}^OA$

**Output:**  ${}^iA$ **5. EXPERIMENTAL SETUP AND TRIAL CONDITIONS**

To evaluate the RC EKF, we used real data from outdoor experiments that were carried out near Paris, France. Three tests were performed on the same road with an experimental automotive vehicle (see Figure 13) in urban conditions.

The experimental vehicle was equipped with an IMU Oxford RT3000 coupled with RTK-GPS. It provided ground truth data at a rate of 100 Hz. A CAN-bus gateway was used to access the wheel speed sensors and the yaw rate gyro. The measured input  $[v, w]$  from the CAN bus was available at 100 Hz. A Mobileye camera installed behind the windscreen detected lane markings at 10 Hz (see Figure 13(c)). A low-cost u-blox 6T GPS receiver with a patch antenna was also part of the equipment (see Figure 13(b)), and this provided position measurements at 5 Hz without any correction.

Figure 14(a) shows the satellites in view for the first two tests and Figure 14(b) for the third test. It will be remarked that the GPS satellite geometries were quite different in the morning and in the afternoon.

Figure 15 shows the test area in the local ENU frame. The gray bounds represent buildings that are from OpenStreetMap to show the urban conditions of the test area. The black lines represent the lane marking map expressed by polylines. The test area consists mainly of two-lane roadways with dashed lane markings in the center of the road and solid markings on both sides of the road. The map has a centimeter-level precision, but a lower accuracy. As the mission goal of the car is given in the map frame, a small map offset has little effect once the GNSS errors have been compensated for in the map frame.

The traveling distance for each test was about 2 km, with a typical speed of 30 km/h. Taking test 1 as an example, the red line represents the test trajectory (see Figure 15). The vehicle started at  $t = 0$  s and halted at  $t = 327$  s. Between  $t = 80$  and 140 s, the vehicle was in a strong, 300-m-long



Figure 13. Experimental setup.

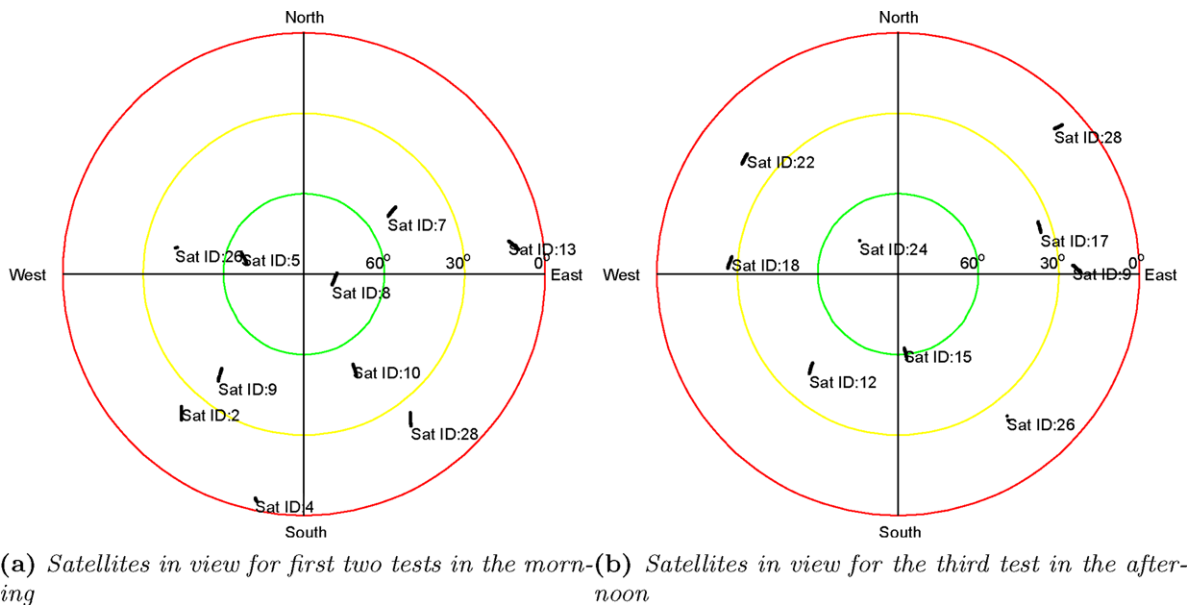
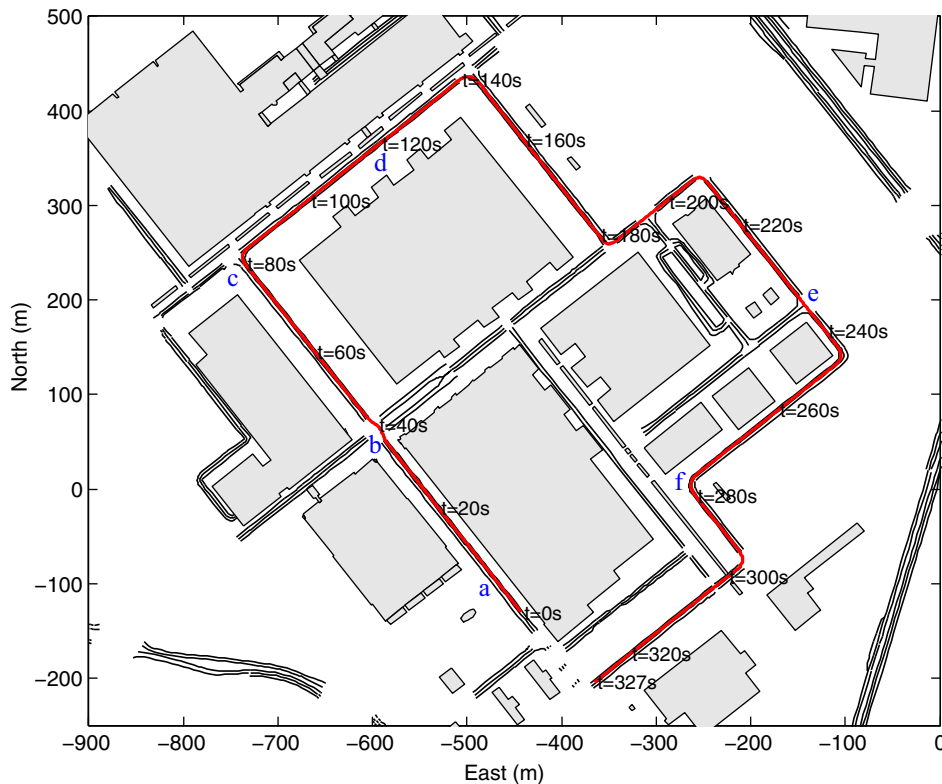


Figure 14. Azimuth-Elevation plot of the tracked GPS satellites during the experiments. In total, there are 10 satellites in view. The satellite geometries were different between the morning and the afternoon.

urban canyon. Around  $t = 123$  s, it encountered situations with GPS multipath and satellite masking. In addition, the camera was unable to detect lane markings at intersections or on roundabouts. Figure 16 shows some typical scenarios recorded by the experimental vehicle.

## 6. RESULTS

As described in Algorithm 2, the output of our localization solver is the vehicle state converted into frame  $R_O$ . Localization performance is studied in  $R_O$  by replaying recorded data. The method proposed in this paper is compared with



**Figure 15.** Test scene and trajectory in the local ENU frame. The red line is the vehicle trajectory in the first test. The black lines are lane markings. The gray boxes are buildings. The vehicle entered an urban canyon around  $t = 123$  s. Several typical locations (a, b, c, d, e, and f) are tagged along the trajectories. The corresponding scenarios are displayed in Figure 16.

a classical loosely coupled EKF implemented in the ENU frame  $R_O$ , in which GNSS bias on  $x$  and  $y$  are modeled as a first-order AR process to maintain the observability of every component of the state. The two filters were implemented in C++ so that they could be tested with the same development framework and with the same tuning parameters.

Figure 17 illustrates what happens when the vehicle moves from one road to another. The green bar indicates the orientation of the  $x$ -axis, and the red bar the  $y$ -axis of the road frame. The blue lines represent the road centerlines, and the white lines are the lane markings. The black ellipse represents the estimated confidence domain ( $3\sigma$ ).

### 6.1. Accuracy Analysis

Table I gives the global performance metrics for the three tests. The outputs of the u-blox receiver, the EKF in the ENU frame (ENU EKF), and the RC EKF are compared. The cross-track and along-track PEs are analyzed and compared. It can be seen that the RC EKF greatly improves localization accuracy, with 95% of the cross-track PEs less than 0.55 m.

Table II gives the relative improvement by the RC EKF with respect to the EKF in the ENU frame, in terms of me-

dian, 95th percentile and maximum of the cross-track and along-track PEs.

Figure 18 shows the cumulative distribution of the absolute PEs by the RC EKF and ENU EKF for the three tests. The RC EKF gives a better estimation in both the cross-track and the along-track directions.

Figure 19 shows the cross-track and along-track PEs over time, with  $\pm 3\sigma$  bounds estimated by the RC EKF. The uncertainty on the cross-track position increases greatly when the camera does not detect lane markings.

### 6.2. Consistency Analysis

Before looking at the consistency of the localization solver, let us examine the distribution of the size of the confidence domain estimated by the filters. For this purpose, we consider the determinant of  $P_{\text{HPE}}$ , denoted as  $|P_{\text{HPE}}|$ , since this is the usual measure of uncertainty where  $P_{\text{HPE}} = \begin{pmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_y^2 \end{pmatrix}$ , and  $\sigma_x$ ,  $\sigma_y$ , and  $\sigma_{xy}$  are estimated by the filter. In practice, confidence is compared with a threshold so that the client application is told either “use” or “don’t use.” It is important, in terms of the availability of the positioning



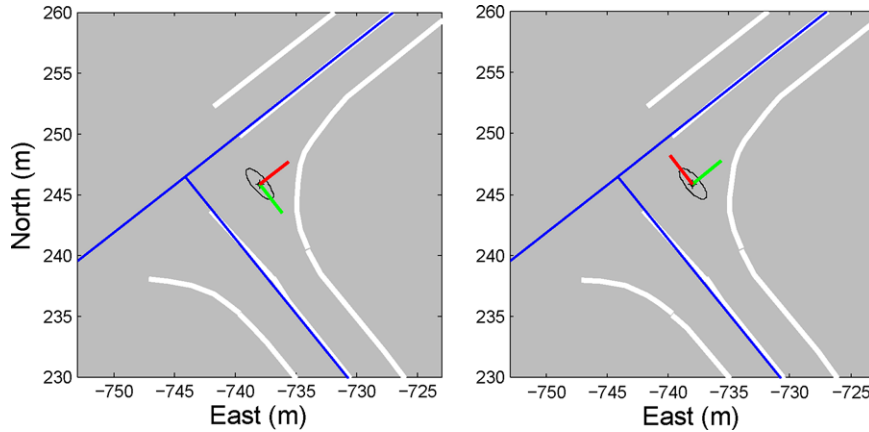


**Figure 16.** Scenarios recorded by a webcam mounted on the experimental vehicle. The roads in the trials had two lanes with mainly dashed lane markings in the center of the road and solid lane markings on both sides, as in (a). In (b), the vehicle encounters a roundabout with few lane markings. (c) A typical turning scenario: the lane marking detection always fails in such cases. (d) The most challenging situations in our experiments, where multipath effect and camera failure occur at the same time. In (e), the vehicle arrives at a fork where the lane detection fails. In (f), the camera fails to detect the highly curved lane markings. Their positions in the map are tagged in Figure 15.

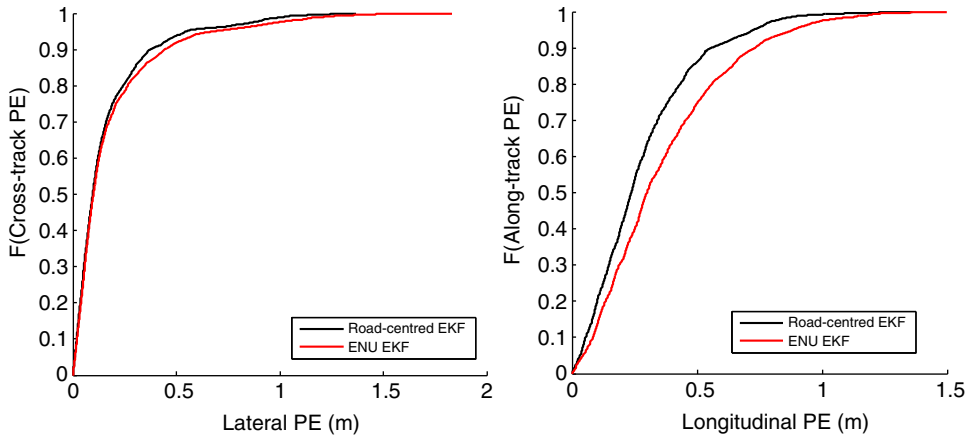
information, to provide confidence zones that are as small as possible. Figure 20 plots the cumulative distribution of the size of the uncertainty and shows that the confidence domain given by the ENU EKF is a little tighter than that given by the RC EKF, but with the same magnitude.

Let us now look at the consistency. The consistency of a filter can be analyzed by constructing the normalized estimation error squared  $\delta^2$ , which has to follow a  $\chi^2$  distribution under the Gaussian assumption:

$$\delta^2 = (\mathbf{x} - \hat{\mathbf{x}})^T P^{-1} (\mathbf{x} - \hat{\mathbf{x}}). \quad (40)$$



**Figure 17.** Change of road frame from one road to another in a real experiment. The vehicle is changing from the southeast road to the northeast road.



**Figure 18.** Cumulative distribution functions (CDF) of the PEs. The performance gain using the RC EKF is particularly significant in the along-track direction.

The normalized horizontal positioning error (HPE) squared  $e^2$  is defined by the 2D position components of the state vector:

$$e^2 = \begin{pmatrix} e_x \\ e_y \end{pmatrix}^T P_{\text{HPE}}^{-1} \begin{pmatrix} e_x \\ e_y \end{pmatrix}, \quad (41)$$

where  $e_x = \hat{x} - x_{\text{ref}}$  and  $e_y = \hat{y} - y_{\text{ref}}$ , with  $(x_{\text{ref}}, y_{\text{ref}})$  the ground truth of the vehicle horizontal position and  $(\hat{x}, \hat{y})$  the estimated position.

If the model assumptions and the tuning filter are correct, then Eq. (41) follows a  $\chi^2$  distribution with two degrees of freedom (denoted as  $\chi^2(2)$  in the following).

In order to study the consistency of the HPE, we look at the percentage of samples exceeding a determined threshold given by a  $\chi^2(2)$  distribution.

$$\begin{pmatrix} e_x \\ e_y \end{pmatrix}^T P_{\text{HPE}}^{-1} \begin{pmatrix} e_x \\ e_y \end{pmatrix} > k^2, \quad (42)$$

where  $k^2$  is the predefined threshold linked to the chosen risk.

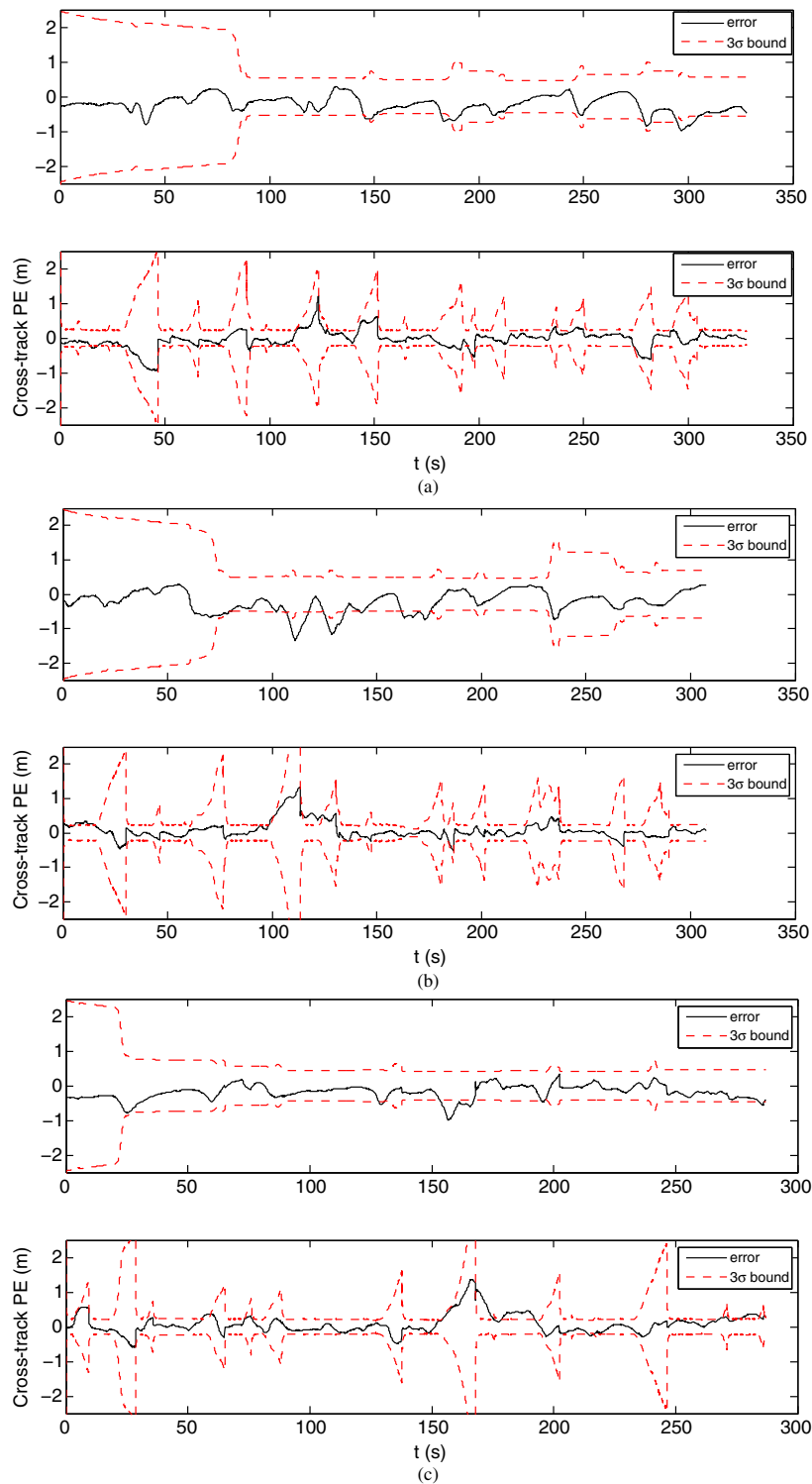
It is easy to check that Eq. (42) is equivalent to the following equation:

$$\sqrt{e_x^2 + e_y^2} > k \sqrt{\frac{1}{\mathbf{u}_e^T P_{\text{HPE}}^{-1} \mathbf{u}_e}}, \quad (43)$$

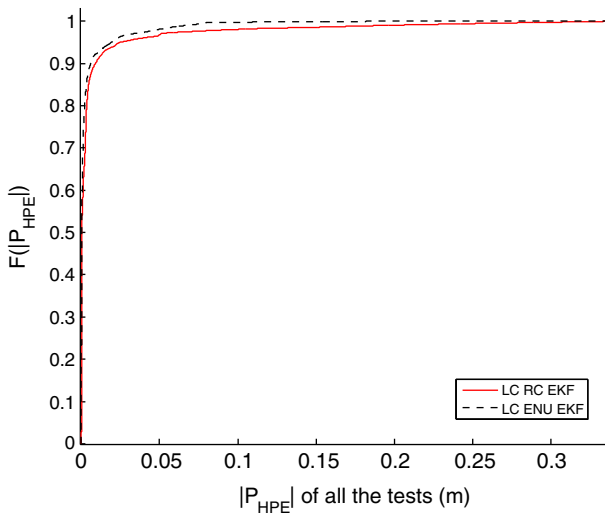
where  $\mathbf{u}_e = \begin{pmatrix} e_x \\ e_y \end{pmatrix} / \sqrt{e_x^2 + e_y^2}$  is the unit vector supporting the HPE.

Let us define  $\sigma_{\text{HPE}}$  as the standard deviation along the HPE vector:

$$\sigma_{\text{HPE}} = \sqrt{\frac{1}{\mathbf{u}_e^T P_{\text{HPE}}^{-1} \mathbf{u}_e}}. \quad (44)$$



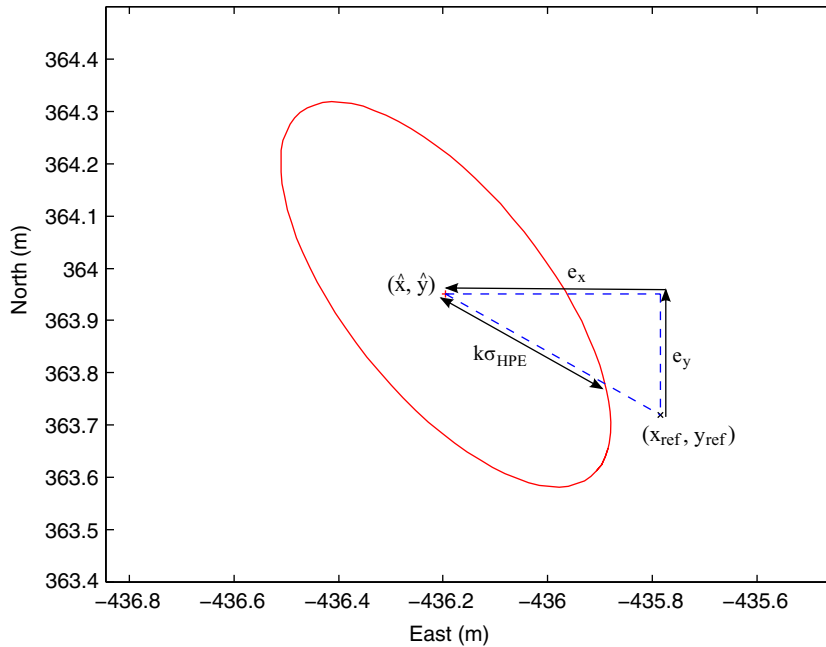
**Figure 19.** Cross-track and along-track PE obtained with the RC EKF.



**Figure 20.** Cumulative distribution of the estimated uncertainty defined by the determinant of the estimated covariance matrix  $|P_{HPE}|$ . LC RC EKF stands for the loosely coupled RC EKF.

Figure 21 illustrates the definition of  $k\sigma_{HPE}$ , where the equation of the ellipse is

$$\begin{pmatrix} x - \hat{x} \\ y - \hat{y} \end{pmatrix}^T P_{HPE}^{-1} \begin{pmatrix} x - \hat{x} \\ y - \hat{y} \end{pmatrix} = k^2.$$



**Figure 21.**  $(\hat{x}, \hat{y})$  is the estimated position.  $(x_{ref}, y_{ref})$  is the ground truth of the vehicle horizontal position. Here, the reference is located outside of the confidence domain.

**Table I.** Error statistics.

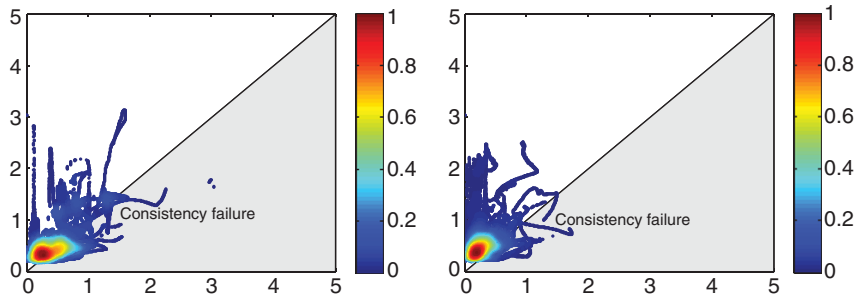
	Cross-track PE (m)			Along-track PE (m)		
	I	II	III	I	II	III
Mean	1.30	0.07	0.04	1.55	-0.32	-0.19
Std. dev.	1.12	0.29	0.26	1.18	0.32	0.29
Median	0.96	0.10	0.09	1.31	0.30	0.24
95th percentile	3.20	0.68	0.55	3.88	0.88	0.73
Mmax	6.78	1.83	1.37	4.69	1.50	1.36

PE: positioning error; I: u-blox; II: ENU EKF; III: RC EKF

We have chosen to set the consistency risk at  $10^{-2}$  (1%), which is a common choice in robotics, but this value can easily be adjusted to any specific requirement. According to the  $\chi^2(2)$  distribution,  $k^2 = 9.21$ . In this case, the corresponding bound of the 2D estimated position is  $3.035\sigma_{HPE}$ , with  $\sigma_{HPE}$  being estimated in real time by the solver. Therefore, a consistency failure occurs when the real error is beyond this  $3.035\sigma_{HPE}$  bound, which means

$$\sqrt{e_x^2 + e_y^2} > 3.035\sigma_{HPE}. \tag{45}$$

Figure 22 shows the consistency performance of the ENU and RC EKFs using a 2D histogram. The points within the gray area satisfy Eq. (45) and indicate consistency failures. The RC EKF is seen to be more consistent, since there are fewer points in the gray area.



**Figure 22.** Consistency plots for the two filters on the three tests (1% consistency risk). The density of the points for the RC EKF is better located above the black line than for the ENU EKF.

**Table II.** Improvement by the RC EKF with respect to the ENU EKF.

	Median	95th percentile	Max
Cross-track positioning	10%	19%	25%
Along-track positioning	20%	17%	9%

**Table III.** Consistency failure rate of the three methods.

	Consistency failure rate			
	Test 1	Test 2	Test 3	Global
ENU EKF	41.2%	31.8%	47.1%	39.9%
RC EKF	9.9%	22.9%	20.5%	17.6%

RC stands for road centered.

The consistencies for the three tests are reported in Table III. The consistency performance of the RC EKF dominates the ENU EKF in every test. The global failure rate of the loosely coupled method is 39.9%, which indicates that the filter is significantly overconfident. When using the RC EKF, the improvement in consistency is more than 55% (from 39.9% to 17.6%). The absolute values are not important here, but it is worth noting the relative improvement.

### 6.3. Robustness of the Road-Centered EKF

Robustness refers here to the ability of the localization system to withstand external perturbations such as GNSS multipath or lane marking large measurements errors. The solution proposed in this paper is to reject the exteroceptive measurements that are doubtful. In this kind of situation, the filter uses DR sensors and continues to provide estimates for the navigation of the car.

#### 6.3.1. Robustness to GNSS Multipath

In Figure 23, the yellow box indicates a period during which a GNSS multipath occurs because of an urban canyon. For

instance, in test 1, the GNSS receiver is affected by the multipath during the time interval  $t = 120\text{--}128$  s (Figure 16(d) shows the scenario). The green and blue points in Figure 23 indicate lane marking detections on the left and right sides, respectively. Unfortunately, there are no lane marking measurements when the multipath effect begins. The cross-track PE rises to 1.2 m for the RC EKF and to 1.83 m for the ENU EKF, and the cross-track PE of the RC EKF is smaller overall than for the ENU EKF during the multipath effect. The camera provides a lane marking measurement on the right side of the lane at  $t = 122.2$  s. The cross-track PE of the RC EKF quickly falls to less than 0.5 m.

#### 6.3.2. Robustness to Outages of the Camera Lane Marking Measurements

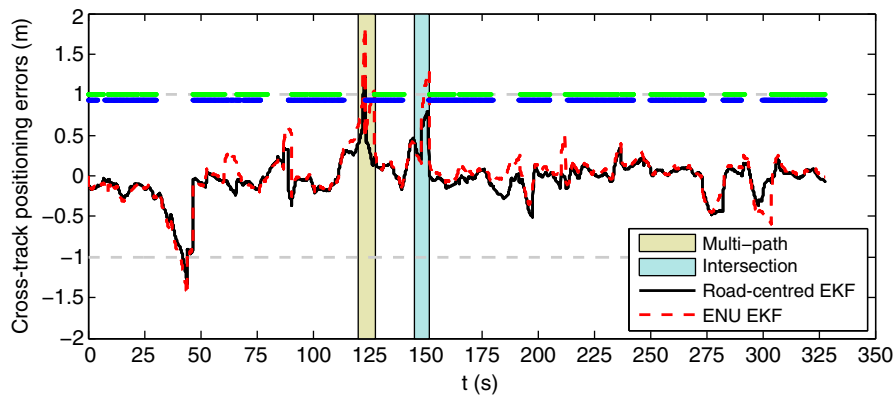
Now, let us see what happens at an intersection without any camera detection.

In Figure 23, the blue box indicates a period during which the vehicle is crossing an intersection. There is no lane marking detection in this kind of situation. The cross-track positioning performance of the RC EKF is seen to be much better during this period, and the cross-track errors remain bounded by 1 m.

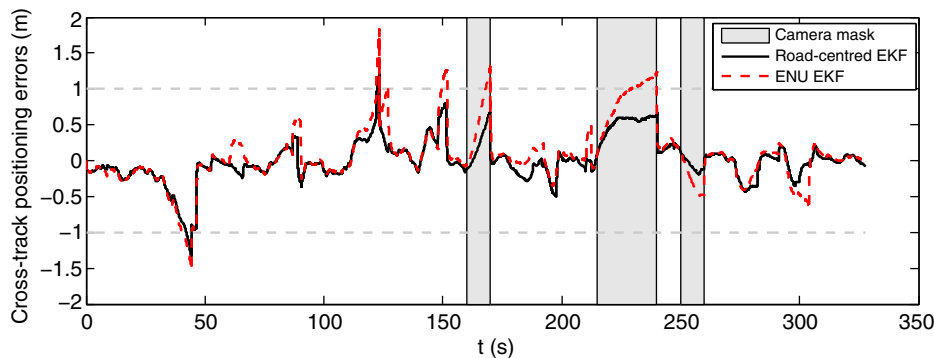
In order to further validate the robustness to outages of lane marking measurements, we simulated camera outages at  $t = 160\text{--}170$  s,  $215\text{--}240$  s, and  $250\text{--}260$  s in test 1 (the gray boxes in Figure 24) when the vehicle was moving in a straight line (cf. Figure 15). Results show that the cross-track positioning by the RC EKF is again much better than for the ENU EKF. The PEs are less than 0.5 m, which indicates a lane level positioning capability.

## 7. CONCLUSIONS

An enhanced model of GNSS errors for use in a multisensor data fusion algorithm that includes GNSS estimates, camera measurements on geo-referenced lane markings, and vehicle odometry was developed in this paper. The proposed error model addresses time correlation and bias of GNSS errors by using a road frame that is always aligned with



**Figure 23.** Cross-track PE during multipath and intersection. The green and blue points indicate lane marking detections on the left and right sides, respectively.



**Figure 24.** Cross-track PEs with camera measurement masks.

the main direction of the carriageway. The observability of this model was demonstrated within an algebraic observability framework. The proposed state space model also allows a bijective transformation between two road frames that guarantees the continuity of the filter estimates when the working frame changes from one road to another. An RC EKF algorithm integrates the proposed shaping models and manages the changes in the working frame. This was tested using experimental data. In all situations, the proposed method works better, in terms of accuracy and consistency, than a localization solver implemented in an ENU frame. During the experiments, GNSS multipath and satellite masking were encountered, as well as outages of camera measurements. Results show that the proposed method is robust to these incidences.

#### ACKNOWLEDGMENTS

This work was carried out within the framework of the Equipex ROBOTEX (ANR-10-EQPX-44-01) and the PAMU project which was supported by the Conseil Général des Yvelines (France).

#### REFERENCES

- Arras, K. O., Castellanos, J. A., Schilt, M., & Siegwart, R. (2003). Feature-based multi-hypothesis localization and tracking using geometric constraints. *Robotics and Autonomous Systems*, 44, 41–53.
- Bar Hillel, A., Lerner, R., Levi, D., & Raz, G. (2014). Recent progress in road and lane detection: a survey. *Machine Vision and Applications*, 25(3), 727–745.
- Bar-shalom, Y., Li, X.-R., & Kirubarajan, T. (2002). *Estimation with applications to tracking and navigation*. New York: John Wiley & Sons.
- Bonnabel, S., Martin, P., & Salaun, E. (2009). Invariant extended Kalman filter: theory and application to a velocity-aided attitude estimation problem. In *Proceedings of the 48th IEEE Conference on Decision and Control* (pp. 1297–1304).
- Bonnifait, P., Bouron, P., Crubille, P., & Meizel, D. (2001). Data fusion of four ABS sensors and GPS for an enhanced localization of car-like vehicles. In *IEEE International Conference on Robotics and Automation* (Vol. 2, pp. 1597–1602).
- Bouguet, J. Y. (2008). *Camera calibration toolbox for Matlab*. [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/)
- Burg, J. (1975). *Maximum entropy spectral analysis*. Stanford Exploration project. Stanford, CA: Stanford University.

- Chatzis, M. N., Chatzi, E. N., & Smyth, A. W. (2015). On the observability and identifiability of nonlinear structural and mechanical systems. *Structural Control and Health Monitoring*, 22(3), 574–593.
- Clanton, J., Bevely, D., & Hodel, A. (2009). A low-cost solution for an integrated multisensor lane departure warning system. *IEEE Transactions on Intelligent Transportation Systems*, 10(1), 47–59.
- Cui, D., Xue, J., Du, S., & Zheng, N. (2014). Real-time global localization of intelligent road vehicles in lane-level via lane marking detection and shape registration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 4958–4964).
- Diop, S., & Wang, Y. (1993). Equivalence between algebraic observability and local generic observability. In *Proceedings of the 32nd IEEE Conference on Decision and Control* (Vol. 3, pp. 2864–2865).
- Fliess, M. (1989). *Automatique et corps différentiels*. *Forum Mathematicum*, 1(3), 227–238.
- Fliess, M., & Glad, S. (1993). An algebraic approach to linear and nonlinear control. In Trentelman, H., & Willems, J., eds, *Essays on control*, volume 14 of *Progress in systems and control theory* (pp. 223–267). Boston: Birkhäuser.
- Furgale, P. et al. (2013). Toward automated driving in cities using close-to-market sensors: An overview of the v-challenge project. In *Intelligent Vehicles Symposium (IV)* (pp. 809–816). IEEE.
- Gao, J., Petovello, M., & Cannon, M. (2006). Development of precise GPS/INS/wheel speed sensor/yaw rate sensor integrated vehicular positioning system. In *Proceedings of ION National Technical Meeting* (pp. 780–792).
- Glad, S., & Ljung, L. (1990). Model structure identifiability and persistence of excitation. In *Proceedings of the 29th IEEE Conference on Decision and Control* (Vol. 6, pp. 3236–3240).
- Groves, P. (2013). *Principles of GNSS, inertial, and multisensor integrated navigation systems*, 2nd edition: GNSS/GPS. Artech House: Boston/London.
- Gruyer, D., Belaroussi, R., & Revilloud, M. (2014). Map-aided localization with lateral perception. In *IEEE Intelligent Vehicles Symposium Proceedings* (pp. 674–680).
- Gruyer, D., Belaroussi, R., & Revilloud, M. (2016). Accurate lateral positioning from map data and road marking detection. *Expert Systems with Applications*, 43, 1–8.
- Guo, C., Meguro, J.-I., Kojima, Y., & Naito, T. (2014). Automatic lane-level map generation for advanced driver assistance systems using low-cost sensors. In *IEEE International Conference on Robotics and Automation* (pp. 3975–3982).
- Hara, K., & Saito, H. (2015). Vehicle localization based on the detection of line segments from multi-camera images. *Journal of Robotics and Mechatronics*, 27(6), 617–626.
- Hermann, R., & Krener, A. J. (1977). Nonlinear controllability and observability. *IEEE Transactions on Automatic Control*, 22(5), 728–740.
- Huang, A., Moore, D., Antone, M., Olson, E., & Teller, S. (2008). Multi-sensor lane finding in urban road networks. In *Proceedings of Robotics: Science and Systems IV*, Zurich, Switzerland.
- Jackson, L. B. (1996). *Digital filters and signal processing*. Kluwer Academic Publishers. Norwell, MA, USA.
- Jo, K., Chu, K., & Sunwoo, M. (2013). Gps-bias correction for precise localization of autonomous vehicles. In *International Vehicles Symposium* (pp. 636–641).
- Kluge, K. (1994). Extracting road curvature and orientation from image edge points without perceptual grouping into features. In *International Vehicles Symposium* (pp. 109–114).
- Kurdej, M., Moras, J., Cherfaoui, V., & Bonnifait, P. (2014). Controlling remanence in evidential grids using geodata for dynamic scene perception. *International Journal of Approximate Reasoning*, 55(1), 355–375.
- Laboratoire Heudiasyc (2015). *Démonstration voiturier autonome renault*. [Video file]. Accessed April 8, 2016, from [https://www.youtube.com/watch?v=3rAILkHys\\_k](https://www.youtube.com/watch?v=3rAILkHys_k).
- Laneuric, J., Chapuis, R., & Chausse, F. (2006). Accurate vehicle positioning on a numerical map. *International Journal of Control, Automation, and Systems*, 3(1), 15–31.
- Lee, B.-H., Song, J.-H., Im, J.-H., Im, S.-H., Heo, M.-B., & Jee, G.-I. (2015). Gps/dr error estimation for autonomous vehicle localization. *Sensors*, 15(8), 20779.
- Levinson, J., & Thrun, S. (2010). Robust vehicle localization in urban environments using probabilistic maps. In *IEEE International Conference on Robotics and Automation* (pp. 4372–4378).
- Litman, T. (2013). *Autonomous vehicle implementation predictions: implications for transport planning*. Victoria, BC, Canada: Victoria Transport Policy Institute.
- Ljung, L., & Glad, S. (1994). On global identifiability for arbitrary model parametrizations. *Automatica*, 30(2), 265–276.
- Miller, I., Campbell, M., & Huttenlocher, D. (2011). Map-aided localization in sparse global positioning system environments using vision and particle filtering. *Journal of Field Robotics*, 28(5), 619–643.
- Morales, Y., Tsubouchi, T., & Yuta, S. (2009). Vehicle 3d localization in mountainous woodland environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3588–3594).
- Nassar, S. (2003). *Improving the Inertial Navigation System (INS) Error Model for INS and INS/DGPS Applications*. PhD thesis, University of Calgary.
- Ogawa, T., & Takagi, K. (2006). Lane recognition using on-vehicle lidar. In *IEEE Intelligent Vehicles Symposium* (pp. 540–545).
- Pink, O. (2008). Visual map matching and localization using a global feature map. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (pp. 1–7).
- Reyher, A. V. R., Joos, A., & Winner, H. (2005). A lidar-based approach for near range lane detection. In *IEEE Intelligent Vehicles Symposium* (pp. 147–152).
- Ritt, J. (1950). *Differential algebra*. American Mathematical Society: Colloquium publications. New York, U.S.A.

- Rose, C., Britt, J., Allen, J., & Bevly, D. (2014). An integrated vehicle navigation system utilizing lane-detection and lateral position estimation systems in difficult environments for gps. *IEEE Transactions on Intelligent Transportation Systems*, 15(6), 2615–2629.
- Schreiber, M., Knoppel, C., & Franke, U. (2013). Lane marking based localization using highly accurate maps. In *IEEE Intelligent Vehicles Symposium* (pp. 449–454).
- Sert, H., Perruquetti, W., Kokosy, A., Jin, X., & Palos, J. (2012). Localizability of unicycle mobile robots: An algebraic point of view. In *IEEE International Conference on Intelligent Robots and Systems* (pp. 223–228).
- Sukkarieh, S., Nebot, E., & Durrant-Whyte, H. (1999). A high integrity IMU/GPS navigation loop for autonomous land vehicle applications. *IEEE Transactions on Robotics and Automation*, 15(3), 572–578.
- Tao, Z. (2016). Autonomous road vehicles localization using satellites, lane markings and vision. PhD thesis, Université de technologie de Compiègne.
- Tao, Z., & Bonnifait, P. (2016). Sequential data fusion of GNSS pseudoranges and Dopplers With map-based vision systems. *IEEE Transactions on Intelligent Vehicles*, 1(3), 254–265.
- Tao, Z., Bonnifait, P., Frémont, V., & Ibanez-Guzman, J. (2013). Mapping and localization using gps, lane markings and proprioceptive sensors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 406–412).
- Wijesoma, W. S., Kodagoda, K. R. S., & Balasuriya, A. P. (2004). Road-boundary detection and tracking using lidar sensing. *IEEE Transactions on Robotics and Automation*, 20(3), 456–464.
- Wolcott, R., & Eustice, R. (2014). Visual localization within lidar maps for automated urban driving. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 176–183).
- Ziegler, J. et al. (2014). Making bertha drive-an autonomous journey on a historic route. *Intelligent Transportation Systems Magazine, IEEE*, 6(2), 8–20.