sensors

MDPI

Article

High Precision Outdoor and Indoor Reference State Estimation for Testing Autonomous Vehicles †

Eduardo Sánchez Morales [1,*], Julian Dauth [1], Bertold Huber [2], Andrés García Higuera [3] and Michael Botsch [1]

[1] Technische Hochschule Ingolstadt, Esplanade 10, 85049 Ingolstadt, Germany; j-dauth@gmx.de (J.D.); michael.botsch@thi.de (M.B.)
[2] GeneSys Elektronik GmbH, In der Spöck 10, 77656 Offenburg, Germany; Huber@genesys-offenburg.de
[3] European Parliamentary Research Service, Rue Wiertz 60, B-1047 Brussels, Belgium; andres.garcia@europarl.europa.eu
* Correspondence: eduardo.sanchezmorales@thi.de
† This paper is an extended version of our paper published in : Sánchez Morales, E.; Botsch, M.; Huber, B.; García Higuera, A. High precision indoor positioning by means of LiDAR. In Proceedings of the 2019 DGON Inertial Sensors and Systems (ISS), Braunschweig, Germany, 10–11 September 2019.

Abstract: A current trend in automotive research is autonomous driving. For the proper testing and validation of automated driving functions a reference vehicle state is required. Global Navigation Satellite Systems (GNSS) are useful in the automation of the vehicles because of their practicality and accuracy. However, there are situations where the satellite signal is absent or unusable. This research work presents a methodology that addresses those situations, thus largely reducing the dependency of Inertial Navigation Systems (INSs) on the SatNav. The proposed methodology includes (1) a standstill recognition based on machine learning, (2) a detailed mathematical description of the horizontation of inertial measurements, (3) sensor fusion by means of statistical filtering, (4) an outlier detection for correction data, (5) a drift detector, and (6) a novel LiDAR-based Positioning Method (LbPM) for indoor navigation. The robustness and accuracy of the methodology are validated with a state-of-the-art INS with Real-Time Kinematic (RTK) correction data. The results obtained show a great improvement in the accuracy of vehicle state estimation under adverse driving conditions, such as when the correction data is corrupted, when there are extended periods with no correction data and in the case of drifting. The proposed LbPM method achieves an accuracy closely resembling that of a system with RTK.

Keywords: machine learning; autonomous vehicles; Inertial Navigation System; Satellite Navigation; Real-Time Kinematic; indoor navigation; reference state

Citation: Sánchez Morales, E.; Dauth, J.; Huber, B.; García Higuera, A.; Botsch, M. High Precision Outdoor and Indoor Reference State Estimation for Testing Autonomous Vehicles. *Sensors* **2021**, *21*, 1131. https://doi.org/10.3390/s21041131

Academic Editor: Aboelmagd Noureldin
Received: 5 January 2021
Accepted: 3 February 2021
Published: 6 February 2021

1. Introduction

Autonomous driving has become a popular trend in automotive research. The motivation for autonomous driving ranges from comfort or practicality functions to safety critical applications. Independently of the final use, the fact that the vehicles are machines that can cause serious harm to humans in the event of a malfunction has to be taken into consideration. Therefore, autonomous driving functions need to be subjected to an extensive process of testing and validation which requires a highly accurate reference vehicle state.

A common practice to generate such a highly accurate reference vehicle state is the fusion of data from Inertial Measurement Units (IMU) with measurements from external sensors. Given the extensive scientific community working on these systems, their practicality, and especially their high accuracy, Satellite Navigation (SatNav) receivers have become the most common choice of sensor to fuse with IMUs. Even consumer-grade receivers are capable of acquiring information from various GNSS (such as GPS, GLONASS, Galileo, Beidou, etc.), which enables the receivers to accurately estimate several state variables. Unfortunately, there are common driving situations where satellite signals are either absent

or corrupted to an extent that renders them unusable. Some examples of these driving situations are tunnels, bridge underpasses, parking structures or testing halls.

The objective of the present work is to reduce the dependency of INSs on external sensors, while still being able to generate a reference vehicle state. Six key aspects of the estimation of the vehicle state are addressed: (i) the standstill recognition (Section 2.1), (ii) the horizontation of IMU measurements (Section 2.2), (iii) sensor fusion (Section 2.3), (iv) drift detection (Section 2.4), (v) outlier detection for correction data (Section 2.5), and (vi) highly accurate indoor navigation (Section 2.6). It will be shown that the proposed methods are able to provide important improvements in the estimation of the vehicle state even when used individually. Furthermore, if used in combination, the proposed methods allow for an accurate enough estimation of the vehicle state to test and validate autonomous driving functions even in indoor environments.

For an adequate estimation of the vehicle state by means of IMUs, it is essential to ascertain when the vehicle is not moving because it is at this time that both the velocity over ground and the proper acceleration are equal to zero [1]. Even though, historically, vehicle speed sensors were designed to measure velocity rather than when a vehicle is at a standstill [2], they are now able to accurately measure a standstill. Nonetheless, there could be disadvantages if external devices continuously read the vehicle sensor data in real time (for example, through the OBDII interface). Even when various methods [3–5] and tools used by manufacturers [6] theoretically allow a CAN-bus use of ≈80% [7], recent studies suggest that CAN-bus messages can still miss their deadlines [8–10]. The traffic within a CAN-bus would increment if an external device continuously accesses the vehicle sensor data in real time, which could further increase the risk that the CAN-bus messages miss their deadlines. Consequently, more research is required before considering fusing the vehicle on-board sensors with external IMUs. Alternative sensors for measuring the velocity over ground, such as the Wheel Pulse Transducer (WPT) [11] or the Correvit [12,13] increase testing costs and complexity.

To the best of authors' knowledge, the only method specifically designed for standstill detection is patented by Robert Bosch GmbH [14]. This method compares the position of an object in consecutive frames of a video provided by a camera in the vehicle, and an algorithm estimates the velocity of the vehicle relative to the object. As with all camera-based methods, there is a trade-off between the spatial resolution and the distance to the recorded objects, which negatively affects the standstill recognition.

The method proposed in this work for standstill recognition is based on a machine learning classifier, and has the key advantage that all required inputs are generated solely with IMU measurements. No additional sensors or infrastructural elements are required, which reduces testing complexity, improves the robustness of the method and makes it highly adequate for indoor navigation.

The state of the vehicle can be estimated once the algorithm has reliably established that the vehicle is moving. If Gauss-Markov stochastic processes are assumed, the most computationally efficient method to estimate the *optimum* state of moving objects is the Kalman Filter (KF) [15]. The KF algorithm can fuse sensor measurements with the vehicle state predicted by means of a motion model. Since the aim of this work is to generate a reference vehicle state for testing and validation of autonomous driving functions, the motion model used is especially designed for accurate performance under tractive driving.

Two critical parameters for the KF are the measurement noise and the system noise, because they are required to calculate the Kalman gain. Therefore, two methods are proposed to monitor both values, and thus to improve the robustness of the KF: a drift recognition and an outlier detector.

The drift recognition tackles the task of monitoring the vehicle state to detect when the vehicle is drifting. This information is extremely valuable because the motion model used in this research work is optimized for tractive-driving situations. When the vehicle starts to drift, the previously estimated system noise is no longer valid. Knowing when the

vehicle is drifting enables the possibility of adjusting the system noise or switching to a better suited motion model.

Other works found in the literature address the issue of drifting too, however, they tend to limit the scope of their research to simulation environments [16,17] or require additional parameters that cannot easily be estimated with an IMU [18]. Conversely, the drift recognition in this work is designed to operate by performing consistency checks between two estimated state variables: the lateral acceleration and the sideslip angle. As demonstrated later, these state variables provide enough information to perform a robust drift detection.

The outlier detector serves as a complement to the drift detector. Although the drift detector monitors the validity of the motion model, the outlier detector does the same for the measurements. This is especially beneficial for SatNav measurements due to the multi-path effect. This effect appears when the signal received from the satellites does not follow a straight path, but bounces off other objects before reaching the SatNav receiver. The multi-path effect can occur as the vehicle drives near objects that are able to reflect the SatNav signal, such as trailers, tall buildings or metallic structures, and causes problems for the methods that measure the vehicle state (trilateration, carrier-phase measurements, etc.). A relevant consequence of the multi-path effect is that it reduces the accuracy of the measured vehicle state. It is quite common to see significant jumps in the measured state variables while the multi-path effect is present. There are several approaches in the literature to deal with the multi-path problem, such as [19–23]. However, to implement these approaches would require access to the pseudo-ranges of the SatNav receivers which is not possible for most consumer-grade devices. Therefore, the problem is addressed by filtering outliers. This filtering is achieved by a consistency check of the measurement vector before it is used as correction data.

In this research work a dynamic low-pass filter is designed to refine the outlier detection method by means of taking the measurement noise into account. This dynamic low-pass filter is highly beneficial for all sensors that estimate a quality indicator for their own measurements, as is the case of the "dilution of precision", "diminution of precision" [24,25] or standard deviation of the SatNav receivers. Many sensors can track the accuracy associated with the measurements that they perform, but are not always capable of detecting estimation errors of this self-reported accuracy.

The investments made in GNSS [26,27] have provided various advantages, such as (i) high accuracy [28,29], (ii) a low cost for end users [30], and (iii) an extensive scientific community that works on refining its measurements and overcoming its limitations [31]. Arguably, these three advantages make the GNSS the best choice for external sensors in open-sky use-cases. However, there are still situations where the GNSS are not available or usable, as is the case in tunnels, parking structures or urban canyons. Moreover, there is an interest in testing in indoor halls because this allows engineers to test sensors, algorithms and systems under simulated and reproducible weather conditions such as day, night, rain, fog or even SatNav outages.

Due to the aforementioned reasons, it is necessary to investigate approaches for acquiring feedback in closed-sky situations. Comprehensive surveys on indoor positioning systems can be found in [32,33]. These approaches are based on the combination of diverse methods and sensors, such as WiFi, sonar, radar, or proprietary technologies such as the iBeacon [34].

The LiDAR-based approach with the best performance is presented on [35] with a mean localization error of 11.5 cm and standard deviation of 5.4 cm. The accuracy is evaluated by comparing the outputs of the algorithm to "human-labeled" ground-truth data. The indoor positioning system with the best performance that was found is the "Active Bat" [36,37] with a mean localization error of 3 cm, but the evaluation methodology is not clearly specified [38]. The biggest disadvantage of the Active Bat is the numerous receivers required which must be placed with separations of 1.2 m and must cover the complete measurement area. Therefore, the installation of such a system might not be

practical. Other indoor positioning systems already on the market [39–41] are designed for applications that do not require a centimeter-precise localization. Alternative navigation methods that use LiDARs are presented in [42–44]. However, the existing methods still face a series of challenges, such as the positioning accuracy, the number of state variables they can measure or the algorithm runtime.

2. Materials and Methods

The present work deals with the measurements provided by different sensors, such as IMUs, LiDARs, SatNav receivers, etc. For simplification purposes, it is assumed that all sensors output their measurements in the Local Car Plane (LCP). The LCP is the vehicle reference frame and is defined analogue to the ISO 8855:2011 norm [45]. The LCP is composed by the mutually perpendicular x_{LCP}, y_{LCP}, and z_{LCP} axes, with $\overrightarrow{z_{\mathrm{LCP}}} = \overrightarrow{x_{\mathrm{LCP}}} \times \overrightarrow{y_{\mathrm{LCP}}}$ and origin o_{LCP} at the Center of sprung Mass (CoM) of the vehicle. The x_{LCP} axis is parallel to the longitudinal axis of the vehicle and points towards the hood, the z_{LCP} axis points upwards, and the y_{LCP} axis is given according to the right-hand rule. The LCP is illustrated in Figure 1.

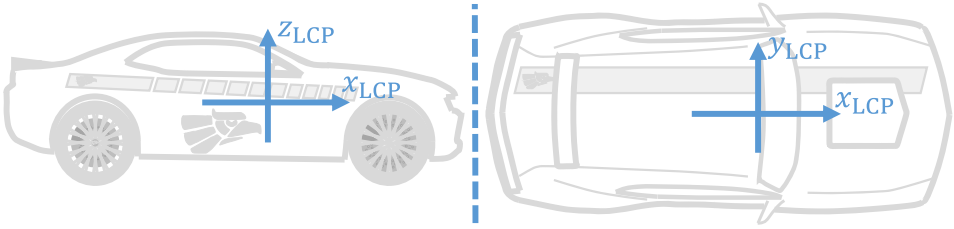

Figure 1. Graphical depiction of the Local Car Plane.

The vehicles move on the Local Tangent Plane (LTP). The LTP is a Cartesian reference frame, composed by the mutually perpendicular x_{LTP}, y_{LTP} and z_{LTP} axes, with $\overrightarrow{z_{\mathrm{LTP}}} = \overrightarrow{x_{\mathrm{LTP}}} \times \overrightarrow{y_{\mathrm{LTP}}}$; and with origin o_{LTP} at an arbitrary location on the surface of the Earth. The axis orientation of the LTP is similar to the East-North-Up (ENU) reference frame, where the $x_{\mathrm{LTP}} \times y_{\mathrm{LTP}}$-plane is perpendicular to the gravitational pull of the Earth, the y_{LTP} axis points to the true north of the Earth, and the z_{LTP} axis is parallel to the gravitational pull of the Earth, and points upwards.

2.1. Standstill Recognition

The first step for estimating the vehicle state is the standstill detection. The strategy to achieve this, is to generate features from the IMU signals that can describe whether the vehicle is standing still or not. The generated features are then classified by means of machine learning. The proposed method uses the Random Forest (RF) [46] for classifying standstill or motion according to the features generated from the IMU signals as inputs. The classes used are shown in Table 1. False positives are unacceptable because they can affect greatly the estimation of the vehicle state, especially if they occur at highway velocities. False negatives can be tolerated because, as shown later, their effect is negligible for the estimation of the vehicle state. If this holds, the standstill classification is considered robust.

Table 1. Standstill classification.

True State \ Classified State	Standstill	Motion
Standstill	True positive	False negative
Motion	False positive	True negative

The use of the RF is justified mainly because it is not possible to manually determine thresholds for the IMU signals that allow a robust standstill classification. This happens because some driving conditions are almost equal to a standstill, such as cruising on a

highway or driving at walking velocity. This fact is further strengthened when considering the numerous configurations possible for the same vehicle chassis (engine, suspension, tires, etc.), and that even small components, such as the engine mounts, can have a significant impact on the vibrations and rotations sensed by the IMU.

The RF not only enables the implementation of a robust standstill recognition, but a 10 min dataset is also enough for a robust classification, and the training process is done within seconds. Also, given that the RF is real-time capable, it allows the online use of the proposed method. Further details about the RF can be found in [46]. The procedure to recognize a standstill consists of three steps: (i) generation of the training dataset, (ii) training of the RF, and (iii) classification of the vehicle state. The specifics applicable to the present research work is detailed below.

The first step to recognize a standstill, is to generate a training dataset for the RF. The 10 min-long dataset that is required to generate the features for the RF is composed by the proper acceleration vector $\boldsymbol{a}_{\mathrm{LCP}}$ and the composition of rotations $\dot{\boldsymbol{\theta}}_{\mathrm{LCP}}$ that are measured by the IMU, these are such that

$$\boldsymbol{a}_{\mathrm{LCP}} = \begin{bmatrix} a_{x,\mathrm{LCP}} & a_{y,\mathrm{LCP}} & a_{z,\mathrm{LCP}} \end{bmatrix}^{\mathrm{T}}, \tag{1}$$

$$\dot{\boldsymbol{\theta}}_{\mathrm{LCP}} = \begin{bmatrix} \dot{\theta}_{\mathrm{roll}} & \dot{\theta}_{\mathrm{pitch}} & \dot{\theta}_{\mathrm{yaw}} \end{bmatrix}^{\mathrm{T}}. \tag{2}$$

The training dataset for the RF is equally distributed between three conditions: standstill, moving at walking pace and random driving.

For the standstill condition, the vehicle should be as motionless as possible, but ready to drive off. In the case of vehicles with automatic transmission, "drive" should be selected; and vehicles with manual transmission should be either in neutral or in gear and with the clutch engaged. The reason for this, is to let the IMU sense engine and drivetrain vibrations as well. During the standstill, all dead loads are negligible, and all live loads are undesirable. Except for engine and drivetrain, all actions that provoke chassis vibration or motion should be avoided: passenger movement, vehicle loading, opening or closing doors, trunk or hood, etc.

For the condition of moving at walking pace, the vehicle is driven as slow as possible and on a straight line. Contact with pedals or steering wheel should be avoided. This allows one to generate samples with the "motion" label but that are very similar to a "standstill", which is highly relevant for differentiating between both.

For the condition of random driving, the vehicle is driven with random accelerations, velocities and directions. The more diverse, the better for the dataset.

Next, the features for the RF are generated in the time and frequency domain. The features in the time domain are: (i) $a_{x,\mathrm{LCP}}^2 + a_{y,\mathrm{LCP}}^2$, (ii) $a_{z,\mathrm{LCP}}^2$, (iii) $\dot{\theta}_{\mathrm{roll}}^2 + \dot{\theta}_{\mathrm{pitch}}^2$, and (iv) $\dot{\theta}_{\mathrm{yaw}}^2$.

The features in the frequency domain are created from the features in the time domain by generating the Discrete Fourier Transform (DFT) of each feature in the time domain. In the present work, the DFT is generated with a MATLAB Toolbox that is based on [47–49]. The frequency range that the DFT uses is given as in Equation (3).

$$\left[0 : \frac{F_{\mathrm{s}}}{n_{\mathrm{DFT}}} : F_{\mathrm{s}} \right], \tag{3}$$

where F_{s} is the sampling frequency and n_{DFT} is the number of samples used for the DFT. The highest meaningful frequency from the DFT is $\frac{F_{\mathrm{s}}}{2}$ [50,51], and a 0 Hz frequency (direct current) is not relevant in this work. Therefore, the used frequency range is given by Equation (4).

$$\left[\frac{F_{\mathrm{s}}}{n_{\mathrm{DFT}}} : \frac{F_{\mathrm{s}}}{n_{\mathrm{DFT}}} : \left\lfloor \frac{n_{\mathrm{DFT}}}{2} \right\rfloor \cdot \frac{F_{\mathrm{s}}}{n_{\mathrm{DFT}}} \right]. \tag{4}$$

The IMU sampling rate is set to $F_{\mathrm{s}} = 100$ Hz, and a time window of $\tau_{\mathrm{DFT}} = 170$ ms is used. Therefore, $n_{\mathrm{DFT}} = 17$ samples are used each time for the DFT generation: the latest

values of the features in the time domain, and their values from the previous $n_{\mathrm{DFT}} - 1$ samples. Therefore, the used DFT frequencies are (i) 5.8824 Hz, (ii) 11.7647 Hz, (iii) 17.6471 Hz, (iv) 23.5294 Hz, (v) 29.4118 Hz, (vi) 35.2941 Hz, (vii) 41.1765 Hz, and (viii) 47.0588 Hz. The single-sided amplitude of these 8 frequencies are the features in the frequency domain for a total of 36 features: 4 in the time domain and 32 in the frequency domain. It should be noted that due to the sliding window, it is not possible to generate a DFT for the first $n_{\mathrm{DFT}} - 1$ samples. In this case, the amplitudes of their frequencies are set to zero.

By analyzing the features of the training datasets of different vehicles, it can be noticed that the dominant frequencies at standstill can be associated with the vehicle motorization. For the vehicles with internal combustion engines, to the idling Revolutions Per Minute (RPM) and the number of cylinders. When the vehicles move at walking velocity, the dominant frequency can be associated mainly with the vehicle differential. Finally, when the vehicle moves faster than walking velocity, the dominant frequencies come from the live loads of the vehicle, such as road imperfections and suspension travel, among others.

As shown in Table 1, "standstill" and "motion" are the two labels used for classification. The labeling of the training dataset can be done by using other sensors as reference, by manually analyzing the IMU signals, or a combination of both. Whichever way, it is important to notice that the labeling directly affects the classification performance of the RF.

A best practice for labeling the training datasets is to include the jerks that appear at drive-off into the "motion" class. When the vehicle comes to a stop, the "standstill" class should start only once the chassis has stopped moving and there are no more jerks or rotations present in the IMU signals. In case of uncertainty, it is preferable to label a sample as "motion" rather than "standstill" because, as is shown later, false negatives can be tolerated, but false positives are unacceptable.

The algorithm initialization and the transition between states are two effects of the proposed method to keep in mind. As for the initialization, $n_{\mathrm{DFT}} - 1$ samples are needed for algorithm initialization. This is because, as explained above, the DFT can be generated only once n_{DFT} samples from the IMU are obtained. The single-sided amplitudes for the previous samples are set to zero, which might not be representative of the vehicle state.

Regarding the transition between states, as a vehicle goes from one state to the other (motion to standstill or vice versa), it might happen that the samples used for the DFT belong to different classes. The smaller the sliding window τ_{DFT}, the faster all samples for the DFT fall within a single class. Otherwise, as the sliding window τ_{DFT} gets bigger, more features can be used for the classification, but this also increases the lapse where samples from both classes are fed to the RF. The current size of τ_{DFT} is found to be adequate for a robust standstill recognition, and is determined after testing the RF with several hours of test datasets, and by comparing the classification performance of the RF with different values for τ_{DFT}.

Once the features in time and frequency domain are generated and labeled, the second step to recognize a standstill, is to train the RF. The training parameters used in this work are (i) number of trees: 12, (ii) stopping criteria: minimum leaf size, and (iii) minimum leaf size: 5. These parameters provide a robust classification and can be used for a variety of land vehicles.

It should be noted that the trained RF is not specific to the vehicle used to generate the training dataset, but to all vehicles with the same configuration (chassis, engine, transmission, suspension, etc.). This means that the training occurs only once, and the trained RF can be used for as long as the vehicle configuration remains the same.

Once the RF is trained, the third step to recognize a standstill, is to use the RF for the standstill recognition on test data. For this, the 36 features required as inputs for the RF are generated as the vehicle drives. The first step to do so is to buffer n_{DFT} IMU samples, i.e., the most recent IMU measurements and the previous $n_{\mathrm{DFT}} - 1$. The latest IMU measurements are the features in the time domain ($a^2_{x,\mathrm{LCP}} + a^2_{y,\mathrm{LCP}}$, $a^2_{z,\mathrm{LCP}}$, $\dot{\theta}^2_{\mathrm{roll}} + \dot{\theta}^2_{\mathrm{pitch}}$, and $\dot{\theta}^2_{\mathrm{yaw}}$), and the features in the frequency domain are generated with all n_{DFT} IMU buffered

samples. When all 36 features (4 in the time domain and 32 in the frequency domain) are generated, they are used as input for the RF.

If a standstill is recognized, all change rates (velocities, accelerations and rotation rates) are set to zero. Only the position and orientation are carried forward from the previous time step. Otherwise, if the vehicle moves, its state is estimated as is detailed later.

2.2. Horizontation of IMU Measurements

The horizontation is understood as the transformation of the IMU measurements, as if the $x_{\mathrm{LCP}} \times y_{\mathrm{LCP}}$ and $x_{\mathrm{LTP}} \times y_{\mathrm{LTP}}$ planes were parallel, and the z_{LCP} axis had the same orientation and direction as the z_{LTP} axis. The horizontation is required mainly because most vehicle motion models found in the literature depict the movement of cars from a bird's-eye perspective. Also, it is not a practical solution to physically maintain the $x_{\mathrm{LCP}} \times y_{\mathrm{LCP}}$ and $x_{\mathrm{LTP}} \times y_{\mathrm{LTP}}$ planes parallel when the vehicles are in motion. The horizontation requires tracking of the LTP axes relative to the LCP, so that the IMU measurements can be projected on the tracked LTP axes. This process varies depending on whether the vehicle is standing still or not. Both cases are detailed in what follows.

With exception of the engine vibrations, there are no live loads present in a vehicle during a standstill. Therefore, the gravitational pull G is the only force sensed by the IMU. This is given at standstill by

$$G = \|\boldsymbol{a}_{\mathrm{LCP}}\| = \sqrt{a_{x,\mathrm{LCP}}^2 + a_{y,\mathrm{LCP}}^2 + a_{z,\mathrm{LCP}}^2} \approx 9.81\,\frac{\mathrm{m}}{\mathrm{s}^2}. \tag{5}$$

By using the gravitational pull as reference, the roll and pitch angles are calculated as follows

$$\theta_{\mathrm{roll}} = \arccos\left(\frac{a_{y,\mathrm{LCP}}}{G}\right), \tag{6}$$

$$\theta_{\mathrm{pitch}} = \arccos\left(\frac{a_{x,\mathrm{LCP}}}{G}\right). \tag{7}$$

Then, the easiest manner to obtain an initial value of the yaw angle θ_{yaw}, is to equip the vehicle with a sensor that measures the Course Over Ground (COG), such as a SatNav receiver, and to drive the vehicle on a straight line. This is so because driving on a straight line results in $\theta_{\mathrm{yaw}} = \mathrm{COG}$.

Two rotations are necessary to transform the LTP to the LCP: one rotation around an arbitrary axis and one rotation around the z_{LTP} axis. The first rotation aligns the z_{LCP} axis with the z_{LTP} axis. The second rotation aligns the x_{LCP} axis with the x_{LTP} axis. The two parameters of the first rotation are the arbitrary axis $\boldsymbol{r}_{\mathrm{h}_0}$ and the rotation angle θ_{h}. They are expressed as follows

$$\theta_{\mathrm{h}} = \arccos\left(\frac{a_{z,\mathrm{LCP}}}{G}\right), \tag{8}$$

$$\boldsymbol{r}_{\mathrm{h}_0} = \begin{bmatrix} a_{x,\mathrm{LCP}} & a_{y,\mathrm{LCP}} & a_{z,\mathrm{LCP}} \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{x,\mathrm{h}} & r_{y,\mathrm{h}} & r_{z,\mathrm{h}} \end{bmatrix}. \tag{9}$$

The rotation matrix $\boldsymbol{R}_{\mathrm{LCP}}^{\mathrm{LCP}'}$ that aligns the z_{LCP} axis with the z_{LTP} axis is then given by:

$$\boldsymbol{R}_{\mathrm{LCP}}^{\mathrm{LCP}'} = \begin{bmatrix} \frac{r_{x,\mathrm{h}}^2 + \left(r_{y,\mathrm{h}}^2 + r_{z,\mathrm{h}}^2\right)\cos(\theta_{\mathrm{h}})}{r_{\mathrm{m}}} & \frac{r_{x,\mathrm{h}} r_{y,\mathrm{h}} r_{\mathrm{c}} - r_{z,\mathrm{h}}\sqrt{r_{\mathrm{m}}}\sin(\theta_{\mathrm{h}})}{r_{\mathrm{m}}} & \frac{r_{x,\mathrm{h}} r_{z,\mathrm{h}} r_{\mathrm{c}} + r_{y,\mathrm{h}}\sqrt{r_{\mathrm{m}}}\sin(\theta_{\mathrm{h}})}{r_{\mathrm{m}}} \\ \frac{r_{x,\mathrm{h}} r_{y,\mathrm{h}} r_{\mathrm{c}} + r_{z,\mathrm{h}}\sqrt{r_{\mathrm{m}}}\sin(\theta_{\mathrm{h}})}{r_{\mathrm{m}}} & \frac{r_{y,\mathrm{h}}^2 + \left(r_{x,\mathrm{h}}^2 + r_{z,\mathrm{h}}^2\right)\cos(\theta_{\mathrm{h}})}{r_{\mathrm{m}}} & \frac{r_{y,\mathrm{h}} r_{z,\mathrm{h}} r_{\mathrm{c}} - r_{x,\mathrm{h}}\sqrt{r_{\mathrm{m}}}\sin(\theta_{\mathrm{h}})}{r_{\mathrm{m}}} \\ \frac{r_{x,\mathrm{h}} r_{z,\mathrm{h}} r_{\mathrm{c}} - r_{y,\mathrm{h}}\sqrt{r_{\mathrm{m}}}\sin(\theta_{\mathrm{h}})}{r_{\mathrm{m}}} & \frac{r_{y,\mathrm{h}} r_{z,\mathrm{h}} r_{\mathrm{c}} + r_{x,\mathrm{h}}\sqrt{r_{\mathrm{m}}}\sin(\theta_{\mathrm{h}})}{r_{\mathrm{m}}} & \frac{r_{z,\mathrm{h}}^2 + \left(r_{x,\mathrm{h}}^2 + r_{y,\mathrm{h}}^2\right)\cos(\theta_{\mathrm{h}})}{r_{\mathrm{m}}} \end{bmatrix}, \tag{10}$$

$$r_{\mathrm{c}} = 1 - \cos(\theta_{\mathrm{h}}), \tag{11}$$

$$r_{\mathrm{m}} = r_{x,\mathrm{h}}^2 + r_{y,\mathrm{h}}^2 + r_{z,\mathrm{h}}^2. \tag{12}$$

The Equation (10) is known as the "Rodriguez' Formula", and its step-by-step derivation can be found in [52]. A graphical representation of a rotation around an arbitrary

axis is shown in Figure 2. It should be noted that Equation (10) can present a singularity if $\theta_{\text{roll}} = 0$ and $\theta_{\text{pitch}} = 0$, because in that case $r_{\text{m}} = 0$ as well. However, this would mean that the z_{LCP} axis and the z_{LTP} axis are aligned. Hence, $\boldsymbol{R}_{\text{LCP}}^{\text{LCP}'} = \boldsymbol{I}_{3\times3}$, where $\boldsymbol{I}$ is the identity matrix.

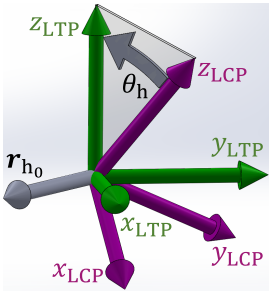

Figure 2. Shown is a rotation around an arbitrary axis to align the z_{LCP} axis with the z_{LTP} axis. The axes of the LTP are shown in green, the axes of the LCP are shown in magenta, and the arbitrary axis $\boldsymbol{r}_{\text{h}_0}$ and the rotation angle θ_{h} with grey.

The matrix that tracks the LTP with respect to the LCP is then given by

$$\boldsymbol{R}_{\text{LTP}}^{\text{LCP}} = \left(\begin{bmatrix} \cos(\theta_{\text{yaw}}) & -\sin(\theta_{\text{yaw}}) & 0 \\ \sin(\theta_{\text{yaw}}) & \cos(\theta_{\text{yaw}}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{R}_{\text{LCP}}^{\text{LCP}'} \right)^{\text{T}}. \tag{13}$$

Once the vehicle pose at standstill is determined, it must be updated as the vehicle moves. The update is done primarily with the gyroscopes because even when cruising in a straight line, random forces act on road vehicles. Some sources of these forces could be road irregularities, bumps, road joints or the wind. These forces are sensed by the IMU, which complicates the update of $\boldsymbol{R}_{\text{LTP}}^{\text{LCP}}$ with the accelerometers. Therefore, the composition of rotations is used instead. This is detailed in what follows.

With the instantaneous rotation rates $\dot{\theta}_{\text{roll}}$, $\dot{\theta}_{\text{pitch}}$ and $\dot{\theta}_{\text{yaw}}$ around the x_{LCP}, y_{LCP}, and z_{LCP} axes respectively, the instantaneous composition of rotations is expressed as follows

$$\boldsymbol{S}\left(\dot{\boldsymbol{\theta}}_{\text{LCP},\tau_2}^{\text{LCP},\tau_1}\right) = \begin{bmatrix} 0 & -\dot{\theta}_{\text{yaw}} & \dot{\theta}_{\text{pitch}} \\ \dot{\theta}_{\text{yaw}} & 0 & -\dot{\theta}_{\text{roll}} \\ -\dot{\theta}_{\text{pitch}} & \dot{\theta}_{\text{roll}} & 0 \end{bmatrix}, \tag{14}$$

$$\dot{\boldsymbol{\theta}}_{\text{LCP},\tau_2}^{\text{LCP},\tau_1} = \begin{bmatrix} \dot{\theta}_{\text{roll}} & \dot{\theta}_{\text{pitch}} & \dot{\theta}_{\text{yaw}} \end{bmatrix}. \tag{15}$$

Next, the instantaneous composition of rotations is integrated as if it was a vector and added to $\boldsymbol{R}_{\text{LTP}}^{\text{LCP}}$. The step-by-step derivation of this procedure is shown below.

$$\hat{\boldsymbol{R}}_{\text{LTP}}^{\text{LCP},\tau_2} = \boldsymbol{R}_{\text{LCP},\tau_1}^{\text{LCP},\tau_2} \boldsymbol{R}_{\text{LTP}}^{\text{LCP},\tau_1}, \tag{16}$$

$$\hat{\boldsymbol{R}}_{\text{LTP}}^{\text{LCP},\tau_2} = \left(\boldsymbol{R}_{\text{LCP},\tau_1}^{\text{LCP},\tau_1} + \Delta\tau \dot{\boldsymbol{R}}_{\text{LCP},\tau_1}^{\text{LCP},\tau_2} \right) \boldsymbol{R}_{\text{LTP}}^{\text{LCP},\tau_1}, \tag{17}$$

$$\hat{\boldsymbol{R}}_{\text{LTP}}^{\text{LCP},\tau_2} = \left(\boldsymbol{R}_{\text{LCP},\tau_1}^{\text{LCP},\tau_1} + \Delta\tau \boldsymbol{S}\left(\dot{\boldsymbol{\theta}}_{\text{LCP},\tau_1}^{\text{LCP},\tau_2}\right) \boldsymbol{R}_{\text{LCP},\tau_1}^{\text{LCP},\tau_2} \right) \boldsymbol{R}_{\text{LTP}}^{\text{LCP},\tau_1}, \tag{18}$$

$$\hat{\boldsymbol{R}}_{\text{LTP}}^{\text{LCP},\tau_2} = \left(\boldsymbol{R}_{\text{LCP},\tau_1}^{\text{LCP},\tau_1} + \Delta\tau \boldsymbol{S}\left(-\dot{\boldsymbol{\theta}}_{\text{LCP},\tau_2}^{\text{LCP},\tau_1}\right) \boldsymbol{R}_{\text{LCP},\tau_1}^{\text{LCP},\tau_2} \right) \boldsymbol{R}_{\text{LTP}}^{\text{LCP},\tau_1}, \tag{19}$$

$$\hat{\boldsymbol{R}}_{\text{LTP}}^{\text{LCP},\tau_2} = \left(\boldsymbol{I}_{3\times3} + \Delta\tau \left(\boldsymbol{S}\left(\dot{\boldsymbol{\theta}}_{\text{LCP},\tau_2}^{\text{LCP},\tau_1}\right) \right)^{\text{T}} \boldsymbol{I}_{3\times3} \right) \boldsymbol{R}_{\text{LTP}}^{\text{LCP},\tau_1}, \tag{20}$$

$$\hat{\boldsymbol{R}}_{\mathrm{LTP}}^{\mathrm{LCP},\tau_2} = \left(\Delta\tau \boldsymbol{S}\left(\dot{\boldsymbol{\theta}}_{\mathrm{LCP},\tau_2}^{\mathrm{LCP},\tau_1}\right)^{\mathrm{T}} + \boldsymbol{I}_{3\times 3}\right)\boldsymbol{R}_{\mathrm{LTP}}^{\mathrm{LCP},\tau_1}, \tag{21}$$

where $\Delta\tau = \tau_2 - \tau_1$. Equation (21) is used each time that a new IMU measurement is available to keep updating $\boldsymbol{R}_{\mathrm{LTP}}^{\mathrm{LCP}}$.

To add rotations as vectors is an approximation that is valid only for rotations that tend to zero [53]. This is because $\hat{\boldsymbol{R}}_{\mathrm{LTP}}^{\mathrm{LCP},\tau_2}$ is not an orthonormal matrix. Therefore, $\hat{\boldsymbol{R}}_{\mathrm{LTP}}^{\mathrm{LCP},\tau_2}$ must be adjusted to make its axes perpendicular to each other. The procedure is shown in the following.

$$\hat{\boldsymbol{R}}_{\mathrm{LTP}}^{\mathrm{LCP},\tau_2} = \begin{bmatrix} \boldsymbol{r}'_{\mathrm{h}_1} & \boldsymbol{r}'_{\mathrm{h}_2} & \boldsymbol{r}'_{\mathrm{h}_3} \end{bmatrix}, \tag{22}$$

$$\boldsymbol{r}''_{\mathrm{h}_3} = \boldsymbol{r}'_{\mathrm{h}_3}, \tag{23}$$

$$\boldsymbol{r}''_{\mathrm{h}_1} = \boldsymbol{r}'_{\mathrm{h}_2} \times \boldsymbol{r}''_{\mathrm{h}_3}, \tag{24}$$

$$\boldsymbol{r}''_{\mathrm{h}_2} = \boldsymbol{r}''_{\mathrm{h}_3} \times \boldsymbol{r}''_{\mathrm{h}_1}, \tag{25}$$

$$\boldsymbol{R}_{\mathrm{LTP}}^{\mathrm{LCP},\tau_2} = \begin{bmatrix} \frac{\boldsymbol{r}''_{\mathrm{h}_1}}{\left\|\boldsymbol{r}''_{\mathrm{h}_1}\right\|} & \frac{\boldsymbol{r}''_{\mathrm{h}_2}}{\left\|\boldsymbol{r}''_{\mathrm{h}_2}\right\|} & \frac{\boldsymbol{r}''_{\mathrm{h}_3}}{\left\|\boldsymbol{r}''_{\mathrm{h}_3}\right\|} \end{bmatrix}. \tag{26}$$

The next step is to project on the LTP the acceleration vector $\boldsymbol{a}_{\mathrm{LCP}}$ and the composition of rotations $\dot{\boldsymbol{\theta}}_{\mathrm{LCP}}$ measured by the IMU. This is done by using the tracker matrix $\boldsymbol{R}_{\mathrm{LTP}}^{\mathrm{LCP},\tau_2}$ as follows

$$\boldsymbol{a}_{\mathrm{LTP},\tau_2} = \left(\boldsymbol{R}_{\mathrm{LTP}}^{\mathrm{LCP},\tau_2}\right)^{\mathrm{T}} \boldsymbol{a}_{\mathrm{LCP}} = \begin{bmatrix} a_{x,\mathrm{LTP},\tau_2} & a_{y,\mathrm{LTP},\tau_2} & a_{z,\mathrm{LTP},\tau_2} \end{bmatrix}, \tag{27}$$

$$\dot{\boldsymbol{\theta}}_{\mathrm{LTP},\tau_2} = \left(\boldsymbol{R}_{\mathrm{LTP}}^{\mathrm{LCP},\tau_2}\right)^{\mathrm{T}} \dot{\boldsymbol{\theta}}_{\mathrm{LCP}} = \begin{bmatrix} \dot{\theta}_{x,\mathrm{LTP},\tau_2} & \dot{\theta}_{y,\mathrm{LTP},\tau_2} & \dot{\theta}_{z,\mathrm{LTP},\tau_2} \end{bmatrix}, \tag{28}$$

where $\boldsymbol{a}_{\mathrm{LTP},\tau_2}$ is the acceleration vector in LTP at time instance τ_2, $\dot{\boldsymbol{\theta}}_{\mathrm{LTP},\tau_2}$ is the composition of rotations in LTP at time instance τ_2, $a_{x,\mathrm{LTP},\tau_2}$, $a_{y,\mathrm{LTP},\tau_2}$ and $a_{z,\mathrm{LTP},\tau_2}$ are the accelerations at time instance τ_2 along the x_{LTP}, y_{LTP} and z_{LTP} axes accordingly; and $\dot{\theta}_{x,\mathrm{LTP},\tau_2}$, $\dot{\theta}_{y,\mathrm{LTP},\tau_2}$ and $\dot{\theta}_{z,\mathrm{LTP},\tau_2}$ are the rotations at time instance τ_2 around the x_{LTP}, y_{LTP} and z_{LTP} axes correspondingly.

The acceleration vector and composition of rotations are then projected on the $x_{\mathrm{LTP}} \times y_{\mathrm{LTP}}$-plane. The magnitude a_{OG} and direction θ_{aOG} of the acceleration over ground are calculated as follows

$$a_{\mathrm{OG}} = \sqrt{a_{x,\mathrm{LTP},\tau_2}^2 + a_{y,\mathrm{LTP},\tau_2}^2}, \tag{29}$$

$$\theta_{\mathrm{aOG}} = \arctan2\left(a_{y,\mathrm{LTP},\tau_2}, a_{x,\mathrm{LTP},\tau_2}\right). \tag{30}$$

The magnitude $\dot{\theta}_{\mathrm{OG}}$ and direction θ_{rOG} of the composition of rotations over ground are given by

$$\dot{\theta}_{\mathrm{OG}} = \sqrt{\dot{\theta}_{x,\mathrm{LTP},\tau_2}^2 + \dot{\theta}_{y,\mathrm{LTP},\tau_2}^2}, \tag{31}$$

$$\theta_{\mathrm{rOG}} = \arctan2\left(\dot{\theta}_{y,\mathrm{LTP},\tau_2}, \dot{\theta}_{x,\mathrm{LTP},\tau_2}\right). \tag{32}$$

Once the accelerations along and rotations around the LTP axes are known, they are projected on the projection on the $x_{\mathrm{LTP}} \times y_{\mathrm{LTP}}$-plane of the longitudinal axis of the vehicle,

i.e., the acceleration vector $\boldsymbol{a}_{\mathrm{LTP},\tau_2}$ and composition of rotations $\dot{\boldsymbol{\theta}}_{\mathrm{LTP},\tau_2}$ are projected on a vector located on the $x_{\mathrm{LTP}} \times y_{\mathrm{LTP}}$-plane and with θ_{yaw} orientation. Therefore,

$$\begin{bmatrix} a_{x,\mathrm{h}} \\ a_{y,\mathrm{h}} \end{bmatrix} = \begin{bmatrix} a_{\mathrm{OG}} \cos\left(\theta_{\mathrm{aOG}} - \theta_{\mathrm{yaw}}\right) \\ a_{\mathrm{OG}} \sin\left(\theta_{\mathrm{aOG}} - \theta_{\mathrm{yaw}}\right) \end{bmatrix}, \tag{33}$$

$$\begin{bmatrix} \dot{\theta}_{\mathrm{roll}} \\ \dot{\theta}_{\mathrm{pitch}} \end{bmatrix} = \begin{bmatrix} \dot{\theta}_{\mathrm{OG}} \cos\left(\theta_{\mathrm{rOG}} - \theta_{\mathrm{yaw}}\right) \\ \dot{\theta}_{\mathrm{OG}} \sin\left(\theta_{\mathrm{rOG}} - \theta_{\mathrm{yaw}}\right) \end{bmatrix}, \tag{34}$$

where $a_{x,\mathrm{h}}$ and $a_{y,\mathrm{h}}$ are called "horizontal accelerations". The measurement vector of the Extended Kalman Filter (EKF) that is designed in this research work (Section 2.3) includes $\dot{\theta}_{z,\mathrm{LTP}}$, $a_{x,\mathrm{h}}$ and $a_{y,\mathrm{h}}$ as measurement variables. The process from Equation (14) to Equation (34) is repeated with each new inertial measurement while the vehicle is in motion to keep updating the values of the measurement vector of the EKF.

2.3. Statistical Filtering

The next step is to estimate the vehicle state. This is done by using a motion model to predict the vehicle motion and by using exteroceptive sensors to get feedback.

If one assumes Gauss-Markov stochastic processes, the most computationally efficient method to estimate the *optimum* state of moving objects is the Kalman Filter. The Kalman Filter fuses the predicted vehicle motion with observed measurements. Its detailed derivation can be found in [15], and the specifics for the present research work are given in what follows.

The measurement vector is defined in the present work as

$$\boldsymbol{z}_{\mathrm{h}} = \begin{bmatrix} x_{\mathrm{h}} & y_{\mathrm{h}} & \theta_{z,\mathrm{LTP}} & \dot{\theta}_{z,\mathrm{LTP}} & v_{\mathrm{h}} & \beta_{\mathrm{h}} & a_{x,\mathrm{h}} & a_{y,\mathrm{h}} \end{bmatrix}^{\mathrm{T}}, \tag{35}$$

where $(x_{\mathrm{h}}, y_{\mathrm{h}})$ are the (x, y) coordinates of o_{LCP} in LTP, $\theta_{z,\mathrm{LTP}}$ is the yaw angle of the vehicle, v_{h} is the velocity over ground of the vehicle, and β_{h} is the sideslip angle of the vehicle. The used measurement noise covariance matrix $\boldsymbol{\sigma}_{\mathrm{z}}$ and state vector $\boldsymbol{x}_{\mathrm{s}}$ are defined as follows

$$\boldsymbol{\sigma}_{\mathrm{z}} = \mathrm{diag}\begin{pmatrix} \sigma^2_{x,\mathrm{h}} & \sigma^2_{y,\mathrm{h}} & \sigma^2_{\theta_z,\mathrm{LTP}} & \sigma^2_{\dot{\theta}_z,\mathrm{LTP}} & \sigma^2_{v,\mathrm{h}} & \sigma^2_{\beta,\mathrm{h}} & \sigma^2_{a_x,\mathrm{h}} & \sigma^2_{a_y,\mathrm{h}} \end{pmatrix}, \tag{36}$$

$$\boldsymbol{x}_{\mathrm{s}} = \begin{bmatrix} x_{\mathrm{s}} & y_{\mathrm{s}} & \theta_{\mathrm{yaw,s}} & \dot{\theta}_{\mathrm{yaw,s}} & v_{\mathrm{s}} & \beta_{\mathrm{sv}} & \dot{\beta}_{\mathrm{sv}} & \beta_{\mathrm{sp}} & a_{x,\mathrm{sh}} & a_{y,\mathrm{sh}} \end{bmatrix}^{\mathrm{T}}, \tag{37}$$

where $(x_{\mathrm{s}}, y_{\mathrm{s}})$ are the (x, y) coordinates of o_{LCP} in LTP, $\theta_{\mathrm{yaw,s}}$ is the yaw angle, $\dot{\theta}_{\mathrm{yaw,s}}$ is the yaw rate, v_{s} is the velocity over ground, β_{sv} is the first estimation of the vehicle sideslip angle, $\dot{\beta}_{\mathrm{sv}}$ is the rate of change of the vehicle sideslip angle, β_{sp} is the second estimation of the vehicle sideslip angle, and $a_{x,\mathrm{sh}}$ and $a_{y,\mathrm{sh}}$ are the vehicle horizontal accelerations. Two sideslip estimators are used to improve the prediction of the vehicle motion. The first sideslip estimator (β_{sv} and $\dot{\beta}_{\mathrm{sv}}$) is based on the balance of moments of inertia and provides a better performance for the estimation of the vehicle velocity. The second sideslip estimator (β_{sp}) is based on a geometrical model under the assumption of tractive driving, and gives a better performance for the position estimation.

With a distance l_r along the longitudinal axis of the vehicle from o_{LCP} to the rear transaxle, and additive system noise $\boldsymbol{\eta}_{\mathrm{s}}$, the used state equations are given by

$$\boldsymbol{x}_{\tau_{i+1},\mathrm{s}} = f(\boldsymbol{x}_{\tau_i,\mathrm{s}}) + \boldsymbol{\eta}_{\mathrm{s}}, \tag{38}$$

$$f_{1-10}(\boldsymbol{x}_{\tau_i,\mathrm{s}}) = \begin{bmatrix} x_\mathrm{s} + v_\mathrm{s}\mathrm{c}\big(\theta_{\mathrm{yaw,s}} + \beta_\mathrm{sp}\big)\Delta\tau + \big(a_{x,\mathrm{sh}}\mathrm{c}\big(\theta_{\mathrm{yaw,s}}\big) - a_{y,\mathrm{sh}}\mathrm{s}\big(\theta_{\mathrm{yaw,s}}\big)\big)\frac{\Delta\tau^2}{2} \\ y_\mathrm{s} + v_\mathrm{s}\mathrm{s}\big(\theta_{\mathrm{yaw,s}} + \beta_\mathrm{sp}\big)\Delta\tau + \big(a_{x,\mathrm{sh}}\mathrm{s}\big(\theta_{\mathrm{yaw,s}}\big) + a_{y,\mathrm{sh}}\mathrm{c}\big(\theta_{\mathrm{yaw,s}}\big)\big)\frac{\Delta\tau^2}{2} \\ \theta_{\mathrm{yaw,s}} + \dot{\theta}_{\mathrm{yaw,s}}\Delta\tau \\ \dot{\theta}_{\mathrm{yaw,s}} \\ v_\mathrm{s} + \big(a_{x,\mathrm{sh}}\mathrm{c}(\beta_\mathrm{sv}) + a_{y,\mathrm{sh}}\mathrm{s}(\beta_\mathrm{sv})\big)\Delta\tau \\ \beta_\mathrm{sv} + \dot{\beta}_\mathrm{sv}\Delta\tau \\ \frac{1}{v_\mathrm{s}}\big(a_{y,\mathrm{sh}}\mathrm{c}(\beta_\mathrm{sv}) - a_{x,\mathrm{sh}}\mathrm{s}(\beta_\mathrm{sv})\big) - \dot{\theta}_{\mathrm{yaw,s}} \\ \tan^{-1}\left(\frac{l_r\dot{\theta}_{\mathrm{yaw,s}}}{v_\mathrm{s}}\right) \\ a_{x,\mathrm{sh}} \\ a_{y,\mathrm{sh}} \end{bmatrix}. \quad (39)$$

From $f_7(\boldsymbol{x}_\mathrm{s})$ and $f_8(\boldsymbol{x}_\mathrm{s})$, one deducts that as the vehicle drives slower than 1.5 $\frac{\mathrm{m}}{\mathrm{s}}$, the sideslip estimators lead to a mathematical indetermination. To solve this, when $v_\mathrm{s} < 1.5\,\frac{\mathrm{m}}{\mathrm{s}}$, then $\boldsymbol{x}_\mathrm{s}(6)$, $\boldsymbol{x}_\mathrm{s}(7)$ and $\boldsymbol{x}_\mathrm{s}(8)$ are set to zero. Therefore, no sideslip is considered for the velocity or position estimation. However, given the low velocities where the indetermination happens, the integration error is negligible.

2.4. Drift Recognition

The geometrical model from which $f_8(\boldsymbol{x}_\mathrm{s})$ is derived, assumes tractive driving. Hence, the performance of the motion model to predict the vehicle motion decays during a drift. By recognizing a drift, it is possible to (i) identify when the performance of the motion model decays, and (ii) take measures such as adjusting the system noise, or switching to other motion models. Therefore, an objective of this research work is to create a drift detector. Other approaches achieve this with multiple SatNav receivers or with specific sensors, such as the Correvit. A drift is recognized in this work by checking the consistency between state variables to reduce the dependence of INSs on external sensors.

The tire longitudinal and lateral slips are a consequence of the forces that act on the tires as well as their mechanical properties. As such, both slips indicate how much force can the tires transmit to the ground. When classified by the tire slips, the driving state of a vehicle can be classified as either "tractive" or "non-tractive" [54]. The tractive driving is characterized by high lateral and longitudinal tire tractive forces and by tire slips usually less than 10° sideslip and less than 10% longitudinal slip. The non-tractive driving is characterized by low tire tractive forces and by high slips that are usually greater than 10° sideslip and greater than 10% longitudinal slip.

Hence, the first pair of state variables to compare to detect a drift, are both sideslip estimators. During tractive driving, $\beta_\mathrm{sv} \approx \beta_\mathrm{sp}$. However, as the vehicle starts drifting, the difference between both gets bigger because β_sp is derived under the assumption of tractive driving. Therefore, the difference between β_sv and β_sp is a strong indicator that shows whether the vehicle is drifting or not.

The second pair of variables that are used for the drift recognition are the estimated lateral acceleration $a_{y,\mathrm{sh}}$ and the expected lateral acceleration $a_{y,\mathrm{sc}}$. The latter results from assuming a tire sideslip of $\beta_t \approx 0$. This would mean that the path of a turning vehicle describes an arch with constant radius, so that $a_{y,\mathrm{sc}} = \dot{\theta}_{\mathrm{yaw,s}} v_\mathrm{s}$. Therefore, $a_{y,\mathrm{sh}}$ estimates the lateral acceleration in reality, and $a_{y,\mathrm{sc}}$ describes how the lateral acceleration should ideally be. During tractive driving, $a_{y,\mathrm{sh}} \approx a_{y,\mathrm{sc}}$. And as the vehicle starts drifting, the difference between $a_{y,\mathrm{sh}}$ and $a_{y,\mathrm{sc}}$ becomes statistically significant.

In the present work, it is said that the vehicle is drifting if $|a_{y,\mathrm{sh}} - a_{y,\mathrm{sc}}| \geq 2.1\,\frac{\mathrm{m}}{\mathrm{s}^2}$ and $|\beta_\mathrm{sv} - \beta_\mathrm{sp}| \geq 0.2\,\mathrm{rad}$. These thresholds are determined after comparing the corresponding state variables as measured during drifting tests, and as estimated during tractive driving on open roads.

2.5. Outlier Detection

The next step is to fuse the predicted vehicle motion with measurements from external sensors (correction data). However, the sensor information could be corrupted, which

would affect negatively the estimation of the vehicle state. To avoid this, the state vector is used as reference to filter out flawed sensor measurements. This is because, by definition, the state vector represents the last known optimal state of the vehicle, and because, assuming that both the motion model and the sensors depict reality with 100% fidelity, the measurement vector is equal to the predicted vehicle state.

The consistency check is done with two consecutive sensor measurements made at time instances τ_1 and τ_2, which are compared to the corresponding predicted state variable. Therefore, the difference between two consecutive velocity measurements could be expressed as follows

$$v_{\tau_2,\mathrm{h}} - v_{\tau_1,\mathrm{h}} = \big(a_{x,\mathrm{sh}}\mathrm{c}(\beta_{\mathrm{sv}}) + a_{y,\mathrm{sh}}\mathrm{s}(\beta_{\mathrm{sv}})\big)\Delta\tau, \tag{40}$$

and to consider the inaccuracies of the sensor and the motion model, this can be rewritten as:

$$v_{\tau_2,\mathrm{h}} - v_{\tau_1,\mathrm{h}} \approx \big(a_{x,\mathrm{sh}}\mathrm{c}(\beta_{\mathrm{sv}}) + a_{y,\mathrm{sh}}\mathrm{s}(\beta_{\mathrm{sv}})\big)\Delta\tau, \tag{41}$$

Since the purpose is to filter outliers, an additional margin is given by means of a factor $\kappa = 3$. Also, only the absolute difference is taken to consider both positive and negative accelerations of the vehicle. Therefore, the measured velocity $v_{\tau_2,\mathrm{h}}$ at time instance τ_2 is used as correction data if

$$\left|v_{\tau_2,\mathrm{h}} - v_{\tau_1,\mathrm{h}}\right| \leq \left|\big(a_{x,\mathrm{sh}}\mathrm{c}(\beta_{\mathrm{sv}}) + a_{y,\mathrm{sh}}\mathrm{s}(\beta_{\mathrm{sv}})\big)\Delta\tau\kappa\right|. \tag{42}$$

The value of κ is determined after analyzing hours of real-world data, where the behavior of the SatNav measurements during regular operation is compared to the behavior of the measurements during periods of time where the multi-path effect is present.

Since $\theta_{\mathrm{yaw}} \approx \mathrm{COG}$ during tractive driving, the COG measured by the SatNav receivers can be used as correction data for $\theta_{\mathrm{yaw,s}}$ when $\dot{\theta}_{\mathrm{yaw,s}} \approx 0$. Analogue to the velocity measurement, the measured COG $\theta_{\tau_2,\mathrm{yaw,h}}$ at time instance τ_2 is used as correction data if

$$\left|\theta_{\tau_2,\mathrm{yaw,h}} - \theta_{\tau_1,\mathrm{yaw,h}}\right| \leq \left|\dot{\theta}_{\mathrm{yaw,s}}\Delta\tau\kappa\right|. \tag{43}$$

Analogue to the velocity and the COG, the SatNav location measurement $\left(x_{\tau_2,\mathrm{h}}, y_{\tau_2,\mathrm{h}}\right)$ at time instance τ_2 is used as correction data if

$$\sqrt{\left(x_{\tau_2,\mathrm{h}} - x_{\tau_1,\mathrm{h}}\right)^2 + \left(y_{\tau_2,\mathrm{h}} - y_{\tau_1,\mathrm{h}}\right)^2} \leq \left|\big(v_{\mathrm{s}} + \big(a_{x,\mathrm{sh}}\mathrm{c}(\beta_{\mathrm{sv}}) + a_{y,\mathrm{sh}}\mathrm{s}(\beta_{\mathrm{sv}})\big)\Delta\tau\big)\Delta\tau\kappa\right|. \tag{44}$$

The same method to detect outliers can be applied for β_{sv} and β_{sp} if necessary.

Some sensors can compute quality indicators for the own measurements, such as a dilution of precision or a standard deviation. The problem is that sometimes the sensors do not recognize when they miscalculate their quality measures, as is the case with the standard deviation of SatNav receivers.

To address this, the quality measure is modelled as a PT1 element to dampen it over time. Therefore, let $\sigma_{\tau_1,\mathrm{h}}$ and $\sigma_{\tau_2,\mathrm{h}}$ be an element of $\boldsymbol{\sigma}_z$ at time instances τ_1 and τ_2, $\sigma'_{\tau_1,\mathrm{h}}$ and $\sigma'_{\tau_2,\mathrm{h}}$ be the values corresponding to $\sigma_{\tau_1,\mathrm{h}}$ and $\sigma_{\tau_2,\mathrm{h}}$ that the sensor calculates, $t_{\mathrm{sat}} = 1.5\,\mathrm{s}$ a saturation parameter, t_{τ_1} and t_{τ_2} a timer at time instances τ_1 and τ_2. The update of the standard deviation is then modelled as follows

$$t_{\tau_2} = \begin{cases} 0, & \sigma_{\tau_1,\mathrm{h}} < \sigma'_{\tau_2,\mathrm{h}} \\ t_{\tau_1} + \Delta\tau & \text{otherwise} \end{cases}, \tag{45}$$

$$\sigma_{\tau_2,\mathrm{h}} = \begin{cases} \sigma'_{\tau_2,\mathrm{h}}, & \sigma_{\tau_1,\mathrm{h}} < \sigma'_{\tau_2,\mathrm{h}} \\ \sigma_{\tau_1,\mathrm{h}} e^{-\frac{t_{\tau_2}}{t_{\mathrm{sat}}}} + \sigma'_{\tau_2,\mathrm{h}}\left(1 - e^{-\frac{t_{\tau_2}}{t_{\mathrm{sat}}}}\right) & \text{otherwise.} \end{cases} \tag{46}$$

The saturation parameter t_{sat} is optimized for a SatNav receiver by analyzing its measurements with and without the multi-path effect, and can be optimized for other sensors as required.

From Equations (45) and (46), it can be deducted that the strategy is to quickly shift the Kalman gain towards the motion model if the quality of the sensor measurements decays, and to gradually shift the gain towards the sensor if the quality of its measurements continuously improve.

2.6. LiDAR-Based Positioning Method

As stated above, there are various situations where no SatNav is available, but where highly precise correction data is required. A highly precise source of correction data is generated in this work by means of a Velodyne LiDAR HDL-32E [55], and the method is detailed in what follows.

The first step is to identify references or "markers" with the LiDAR. Since recognizing markers by their shape is computationally intensive, they are identified by their reflectivity. It should be noted that the point cloud resolution decreases (i) as the distance d_{velo} between the LiDAR and the measured object increases, and (ii) as the rotational velocity ω_{velo} of the LiDAR head increases. The Euclidean distance $d_{\text{hor,velo}}$ between two horizontally adjacent points as a function of d_{velo} and ω_{velo} is given by

$$d_{\text{hor,velo}} = d_{\text{velo}} \cdot \sin\left(\omega_{\text{velo}} \cdot \frac{360}{60} \cdot 0.00004608\,\text{ms}\right), \tag{47}$$

where 0.00004608 ms is the time required for one firing cycle (a single shot of all lasers for a single LiDAR azimuth). From Equation (47), the relation between d_{velo}, ω_{velo} and the point cloud resolution can be deducted. This relation can be very relevant as the vehicle velocity increases.

The chosen LiDAR can measure the NIST calibrated reflectivity [56], which ranges from 0 for a lost reflection to 255 for a reflection with no loses. This allows differentiation of highly reflective objects from the rest of the world, but not to identify the highly reflective objects individually.

Once only the points with high reflectivity remain, they are clustered using a time threshold of $\Gamma_{\text{cluster}} = 1$ ms. Therefore, the points measured within Γ_{cluster} one another, are clustered together. Hence, all points that are within a clustering distance d_{cluster} from each other are clustered together, so that

$$d_{\text{cluster}} = d_{\text{velo}} \cdot \sin\left(\omega_{\text{velo}} \cdot \frac{360}{60} \cdot \Gamma_{\text{cluster}}\right). \tag{48}$$

The value of d_{cluster} is important when defining ω_{velo}, Γ_{cluster} and the spacing between markers.

The next step is to determine a single set of (x,y) coordinates for the cluster. For this work, the motion of the vehicle is limited to the $x_{\text{LTP}} \times y_{\text{LTP}}$-plane. Therefore, the i-th cluster point $\boldsymbol{p}_{i,\text{velo}}$ is defined as

$$\boldsymbol{p}_{i,\text{velo}} = \begin{bmatrix} x_{i,\text{velo}} \\ y_{i,\text{velo}} \end{bmatrix} = \begin{bmatrix} d_{i,\text{velo}} \cdot \cos(\theta_{i,\text{velo}}) \\ d_{i,\text{velo}} \cdot \sin(\theta_{i,\text{velo}}) \end{bmatrix}, \tag{49}$$

and the i-th cluster of LiDAR measurements is then defined as

$$\boldsymbol{P}_{i,\text{velo}} \in \mathbb{R}^{2,n_{\text{pts}}} = \begin{bmatrix} x_{1,\text{velo}} & x_{i,\text{velo}} & \cdots & x_{n_{\text{pts}},\text{velo}} \\ y_{1,\text{velo}} & y_{i,\text{velo}} & \cdots & y_{n_{\text{pts}},\text{velo}} \end{bmatrix}, \tag{50}$$

where $d_{i,\text{velo}}$ and $\theta_{i,\text{velo}}$ are respectively the range and azimuth angle of the i-th measured point, and n_{pts} is the number of points of the cluster.

The coordinates $\boldsymbol{p}'_{i,\mathrm{m}} = [x'_{i,\mathrm{m}}, y'_{i,\mathrm{m}}]^{\mathrm{T}}$ that express the measured position in LCP of the i-th cluster (marker) are then calculated by means of the mid-range arithmetic mean, so that

$$\boldsymbol{p}'_{i,\mathrm{m}} = \begin{bmatrix} x'_{i,\mathrm{m}} \\ y'_{i,\mathrm{m}} \end{bmatrix} = \begin{bmatrix} \frac{\max\left(\boldsymbol{P}_{i,\mathrm{velo}}(1,:)\right)+\min\left(\boldsymbol{P}_{i,\mathrm{velo}}(1,:)\right)}{2} \\ \frac{\max\left(\boldsymbol{P}_{i,\mathrm{velo}}(2,:)\right)+\min\left(\boldsymbol{P}_{i,\mathrm{velo}}(2,:)\right)}{2} \end{bmatrix}. \tag{51}$$

The mid-range arithmetic is applied as well to d_{velo}, θ_{velo} and the timestamp of the clustered measurements. This allows one to obtain the distance $d_{i,\mathrm{m}}$, the azimuth $\theta_{i,\mathrm{m}}$ and the time instance at which the i-th marker is seen. Next, the true location in LTP of the markers must be known. This is defined for the i-th marker as follows

$$\boldsymbol{p}_{i,\mathrm{map}} = [x_{i,\mathrm{map}}, y_{i,\mathrm{map}}]^{\mathrm{T}}. \tag{52}$$

The true location in LTP of all markers is stored in the marker map, which is assumed to be available beforehand. This map can be generated, for example, by means of a tachymeter or by stitching together LiDAR measurements of the test area. The marker map is then defined as follows

$$\boldsymbol{P}_{\mathrm{map}} \in \mathbb{R}^{3,n_{\mathrm{pts}}} = \begin{bmatrix} 1 & i & \dots & n_{\mathrm{pts}} \\ x_{1,\mathrm{map}} & x_{i,\mathrm{map}} & \dots & x_{n_{\mathrm{pts}},\mathrm{map}} \\ y_{1,\mathrm{map}} & y_{i,\mathrm{map}} & \dots & y_{n_{\mathrm{pts}},\mathrm{map}} \end{bmatrix}, \tag{53}$$

where n_{pts} is the total number markers, and the first row is a unique identifier for each marker.

In order to identify which marker is being measured, an approximate position $\boldsymbol{p}_{\mathrm{app}}$ and an approximate orientation θ_{app} of the vehicle in LTP is required. This approximate position can be, for example, $(x_{\mathrm{s}}, y_{\mathrm{s}})$, $(x_{\mathrm{h}}, y_{\mathrm{h}})$ or initialization values, and is defined as follows

$$\boldsymbol{p}_{\mathrm{app}} = [x_{\mathrm{app}}, y_{\mathrm{app}}]^{\mathrm{T}}. \tag{54}$$

The positioning error $\eta_{\mathrm{d,app}}$ is then defined as follows

$$\eta_{\mathrm{d,app}} = \left| \boldsymbol{p}_{\mathrm{app}} - \boldsymbol{p}_{\mathrm{true}} \right|, \tag{55}$$

were $\boldsymbol{p}_{\mathrm{true}} = [x_{\mathrm{true}}, y_{\mathrm{true}}]^{\mathrm{T}}$ is the true position of the vehicle in LTP. Since the location of the vehicle is tracked by means of an EKF, during tractive driving it holds that

$$\boldsymbol{p}_{\mathrm{true}} \approx \boldsymbol{p}_{\mathrm{app}} \approx \begin{bmatrix} x_{\mathrm{s}} \\ y_{\mathrm{s}} \end{bmatrix}. \tag{56}$$

The approximate orientation can be $\theta_{\mathrm{yaw,s}}$ or an initialization value, and during tractive driving it holds that

$$\theta_{\mathrm{true}} \approx \theta_{\mathrm{app}} \approx \theta_{\mathrm{yaw,s}}, \tag{57}$$

were θ_{true} is the true orientation of the vehicle in LTP. The orientation error η_{θ} is defined as follows

$$\eta_{\theta} = \left| \theta_{\mathrm{app}} - \theta_{\mathrm{true}} \right|. \tag{58}$$

So as to prevent ambiguities in the marker identification, $\eta_{\mathrm{d,app}}$ and η_{θ} are constrained as follows

$$\left| \boldsymbol{p}'_{i,\mathrm{m}} \right| \sin(\eta_{\theta}) + \eta_{\mathrm{d,app}} < \frac{\min\left(\boldsymbol{d}_{\mathrm{map}}\right)}{2}, \tag{59}$$

where $\boldsymbol{d}_{\mathrm{map}}$ is a vector with the Euclidean distance between all possible marker combinations. The markers do not have to be placed in specific patterns, as long as the constraint of

Equation (59) holds. The measured position $\boldsymbol{p}_{i,\mathrm{m}} = [x_{i,\mathrm{m}}, y_{i,\mathrm{m}}]^{\mathrm{T}}$ in LTP of the i-th marker is then given by

$$\boldsymbol{p}_{i,\mathrm{m}} = \boldsymbol{p}_{\mathrm{app}} + \begin{bmatrix} \cos(\theta_{\mathrm{app}}) & -\sin(\theta_{\mathrm{app}}) \\ \sin(\theta_{\mathrm{app}}) & \cos(\theta_{\mathrm{app}}) \end{bmatrix} \boldsymbol{p}'_{i,\mathrm{m}}. \tag{60}$$

The element of $\boldsymbol{P}_{\mathrm{map}}$ closest to $\boldsymbol{p}_{i,\mathrm{m}}$ is then the measured marker.

Once the markers can be individually identified, the measurement variables are calculated. The first one is the velocity over ground v_{h}. To calculate it, it is required that two LiDAR measurements (consecutive or not) point to the same marker. So, let $\boldsymbol{p}'_{\tau_1,\mathrm{m}}$ and $\boldsymbol{p}'_{\tau_2,\mathrm{m}}$ be the measurements in LCP of one marker done at time instances τ_1 and τ_2 accordingly, so that

$$\boldsymbol{p}'_{\tau_1,\mathrm{m}} = \begin{bmatrix} x'_{\tau_1,\mathrm{m}} \\ y'_{\tau_1,\mathrm{m}} \end{bmatrix} = \begin{bmatrix} d_{\tau_1,\mathrm{m}} \cdot \cos(\theta_{\tau_1,\mathrm{m}}) \\ d_{\tau_1,\mathrm{m}} \cdot \sin(\theta_{\tau_1,\mathrm{m}}) \end{bmatrix}, \tag{61}$$

$$\boldsymbol{p}'_{\tau_2,\mathrm{m}} = \begin{bmatrix} x'_{\tau_2,\mathrm{m}} \\ y'_{\tau_2,\mathrm{m}} \end{bmatrix} = \begin{bmatrix} d_{\tau_2,\mathrm{m}} \cdot \cos(\theta_{\tau_2,\mathrm{m}}) \\ d_{\tau_2,\mathrm{m}} \cdot \sin(\theta_{\tau_2,\mathrm{m}}) \end{bmatrix}. \tag{62}$$

From $\boldsymbol{p}'_{\tau_1,\mathrm{m}}$ and $\boldsymbol{p}'_{\tau_2,\mathrm{m}}$, the geometry of a cone is constructed as shown in Figure 3.

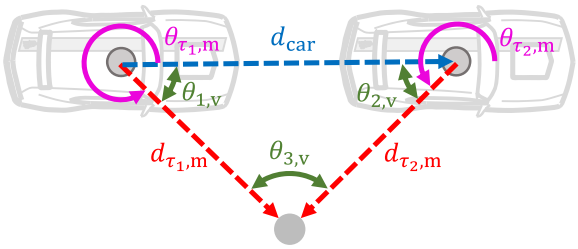

Figure 3. Shown is a graphical depiction of the cone geometry used for the velocity calculation from two LiDAR measurements that point to the same marker.

Since the sum of the internal angles of a triangle is π rad, the internal angles of the constructed geometry are calculated as follows

$$\theta_{1,\mathrm{v}} = \begin{cases} \theta_{\tau_1,\mathrm{m}}, & \theta_{\tau_1,\mathrm{m}} \leq \pi \\ 2\pi - \theta_{\tau_1,\mathrm{m}}, & \theta_{\tau_1,\mathrm{m}} > \pi \end{cases}, \tag{63}$$

$$\theta_{2,\mathrm{v}} = \begin{cases} \pi - \theta_{\tau_2,\mathrm{m}} + \dot{\theta}_{\mathrm{yaw,s}} \cdot \Delta\tau, & \theta_{\tau_2,\mathrm{m}} \leq \pi \\ \theta_{\tau_2,\mathrm{m}} + \dot{\theta}_{\mathrm{yaw,s}} \cdot \Delta\tau - \pi, & \theta_{\tau_2,\mathrm{m}} > \pi \end{cases}, \tag{64}$$

$$\theta_{3,\mathrm{v}} = \pi - \theta_{2,\mathrm{v}} - \theta_{1,\mathrm{v}}. \tag{65}$$

The vehicle displacement d_{car} between the time instances τ_1 and τ_2 is calculated by means of the cosine law. In addition, since the LiDAR measurements have a timestamp, the measured velocity over ground of the vehicle v_{h} between τ_1 and τ_2 is calculated as follows

$$v_{\mathrm{h}} = \frac{d_{\mathrm{car}}}{\Delta\tau} = \frac{\sqrt{d_{\tau_1,\mathrm{m}}^2 + d_{\tau_2,\mathrm{m}}^2 - 2 \cdot d_{\tau_1,\mathrm{m}} \cdot d_{\tau_2,\mathrm{m}} \cdot \cos(\theta_{3,\mathrm{v}})}}{\Delta\tau}. \tag{66}$$

Nevertheless, the cone geometry is not completely defined as it can rotate around the marker. Therefore, it cannot be used to calculate the vehicle location. Instead, both LiDAR measurements must point to different markers. Also, to completely define the geometry shown in Figure 4, $\boldsymbol{p}'_{\tau_2,\mathrm{m}}$ is expressed in the LCP at time instance τ_1 as follows

$$\tilde{\boldsymbol{p}}_{\tau_2,\mathrm{m}} = \begin{bmatrix} \cos(\theta_{\tau_2,\mathrm{yaw,s}} - \theta_{\tau_1,\mathrm{yaw,s}}) & -\sin(\theta_{\tau_2,\mathrm{yaw,s}} - \theta_{\tau_1,\mathrm{yaw,s}}) \\ \sin(\theta_{\tau_2,\mathrm{yaw,s}} - \theta_{\tau_1,\mathrm{yaw,s}}) & \cos(\theta_{\tau_2,\mathrm{yaw,s}} - \theta_{\tau_1,\mathrm{yaw,s}}) \end{bmatrix}^{\mathrm{T}} \boldsymbol{p}'_{\tau_2,\mathrm{m}} + \begin{bmatrix} x_{\tau_2,\mathrm{s}} - x_{\tau_1,\mathrm{s}} \\ y_{\tau_2,\mathrm{s}} - y_{\tau_1,\mathrm{s}} \end{bmatrix} = \begin{bmatrix} \tilde{x}_{\tau_2,\mathrm{m}} \\ \tilde{y}_{\tau_2,\mathrm{m}} \end{bmatrix}. \tag{67}$$

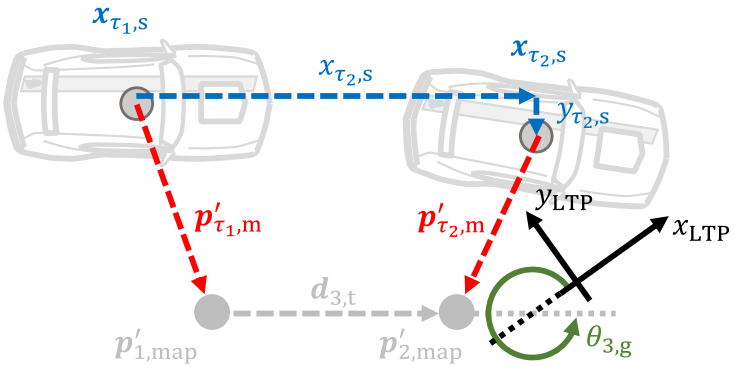

Figure 4. Shown is a graphical depiction of the geometry used for calculating the vehicle location.

Next, assuming that $\boldsymbol{p}'_{\tau_1,m}$ and $\boldsymbol{p}'_{\tau_2,m}$ point to $\boldsymbol{p}_{1,map}$ and $\boldsymbol{p}_{2,map}$ respectively, the angular offset $\Xi_{\theta,LCP}$ from the LTP to the LCP at time instance τ_1 is given by

$$\Xi_{\theta,LCP} = \arctan2\left(\tilde{y}_{\tau_2,m} - y'_{\tau_1,m}, \tilde{x}_{\tau_2,m} - x'_{\tau_1,m}\right) - \arctan2\left(y_{2,map} - y_{1,map}, x_{2,map} - x_{1,map}\right). \tag{68}$$

Afterwards, the marker measurements in LCP at time instance τ_1 are rotated as follows

$$\tilde{\boldsymbol{p}}^*_{\tau_1,m} = \begin{bmatrix} \cos(\Xi_{\theta,LCP}) & -\sin(\Xi_{\theta,LCP}) \\ \sin(\Xi_{\theta,LCP}) & \cos(\Xi_{\theta,LCP}) \end{bmatrix}^T \boldsymbol{p}'_{\tau_1,m} = \begin{bmatrix} \tilde{x}^*_{\tau_1,m} \\ \tilde{y}^*_{\tau_1,m} \end{bmatrix}, \tag{69}$$

$$\tilde{\boldsymbol{p}}^*_{\tau_2,m} = \begin{bmatrix} \cos(\Xi_{\theta,LCP}) & -\sin(\Xi_{\theta,LCP}) \\ \sin(\Xi_{\theta,LCP}) & \cos(\Xi_{\theta,LCP}) \end{bmatrix}^T \tilde{\boldsymbol{p}}_{\tau_2,m} = \begin{bmatrix} \tilde{x}^*_{\tau_2,m} \\ \tilde{y}^*_{\tau_2,m} \end{bmatrix}. \tag{70}$$

Finally, the linear offsets Ξ_{LCP} from the LTP to the LCP at time instance τ_1 can be estimated by comparing $\tilde{\boldsymbol{p}}^*_{\tau_1,m}$ and $\tilde{\boldsymbol{p}}^*_{\tau_2,m}$ with the true marker location $\boldsymbol{p}_{1,map}$ and $\boldsymbol{p}_{2,map}$ from the marker library. Since both measurements have the same accuracy, the average of both offsets is calculated as follows

$$\Xi_{LCP} = \frac{\left(\tilde{\boldsymbol{p}}^*_{\tau_1,m} - \boldsymbol{p}_{1,map}\right) + \left(\tilde{\boldsymbol{p}}^*_{\tau_2,m} - \boldsymbol{p}_{2,map}\right)}{2} = \begin{bmatrix} \Xi_{x,LCP} \\ \Xi_{y,LCP} \end{bmatrix}. \tag{71}$$

The location and orientation measurement variables at time instances τ_1 and τ_2 are then given by

$$\begin{bmatrix} x_{\tau_1,h} \\ y_{\tau_1,h} \\ \theta_{\tau_1,yaw,h} \end{bmatrix} = - \begin{bmatrix} \Xi_{x,LCP} \\ \Xi_{y,LCP} \\ \Xi_{\theta,LCP} \end{bmatrix}, \tag{72}$$

$$\begin{bmatrix} x_{\tau_2,h} \\ y_{\tau_2,h} \\ \theta_{\tau_2,yaw,h} \end{bmatrix} = - \begin{bmatrix} \Xi_{x,LCP} \\ \Xi_{y,LCP} \\ \Xi_{\theta,LCP} \end{bmatrix} + \begin{bmatrix} x_{\tau_2,s} - x_{\tau_1,s} \\ y_{\tau_2,s} - y_{\tau_1,s} \\ \theta_{\tau_2,yaw,s} - \theta_{\tau_1,yaw,s} \end{bmatrix}. \tag{73}$$

As can be deducted from the previous equations, the measurement variables for two consecutive time instances are calculated at each iteration, i.e., once the LiDAR measurement from the time instance τ_2 is acquired, the vehicle pose for the time instances τ_1 and τ_2 is calculated. When the LiDAR measurement from the time instance τ_3 is acquired, the vehicle pose for the time instances τ_2 and τ_3 is calculated; and so on. Since this creates an overlap of two pose measurements per time instance, both are averaged. Therefore, let $(x_{h,a}, y_{h,a})$ and $(x_{h,b}, y_{h,b})$ be the measured location of the vehicle at the i-th time instance that is calculated with the LiDAR measurements from the time instances $i-1$ to $i+1$. The measurement variables for the vehicle location at the i-th time instance are then given by

$$\begin{bmatrix} x_{\tau_i,h} \\ y_{\tau_i,h} \end{bmatrix} = \frac{\begin{bmatrix} x_{h,a} \\ y_{h,a} \end{bmatrix} + \begin{bmatrix} x_{h,b} \\ y_{h,b} \end{bmatrix}}{2}. \tag{74}$$

Analogously, let $\theta_{\text{yaw,h,a}}$ and $\theta_{\text{yaw,h,b}}$ be the measured orientation of the vehicle at the i-th time instance that is calculated with the LiDAR measurements from the time instances $i-1$ to $i+1$. The measurement variable for the vehicle orientation at the i-th time instance is then given by

$$\theta_{\tau_i,\text{yaw,h}} = \frac{\theta_{\text{yaw,h,a}} + \theta_{\text{yaw,h,b}}}{2}. \tag{75}$$

As can be deducted from Equations (74) and (75), to average the measurement variables that are obtained using different pairs of LiDAR measurements aids to compensate LiDAR measurement errors, and creates a time-smoothing effect for the measurement vector. A graphical depiction of this process is shown in Figure 5.

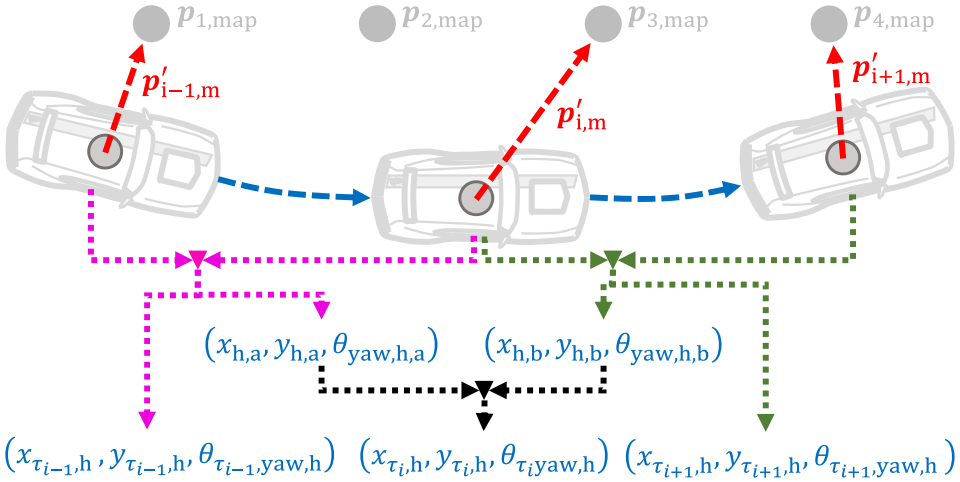

Figure 5. Shown is a graphical depiction of the generation of the measurement variables for the vehicle pose, as computed from LiDAR measurements.

To ensure that the measured object is a marker and not any other reflective object, outliers are filtered out. For this, the LiDAR measurements are translated to the LTP to compare them with the true marker position from the marker library. Therefore, the locations $\hat{\boldsymbol{p}}'_{\tau_i,\text{m}}$ and $\hat{\boldsymbol{p}}'_{\tau_{i+1},\text{m}}$ in LTP of the LiDAR measurements $\boldsymbol{p}'_{\tau_i,\text{m}}$ and $\boldsymbol{p}'_{\tau_{i+1},\text{m}}$ made at the i-th and $i+1$-th time instance are expressed as follows

$$\hat{\boldsymbol{p}}'_{\tau_i,\text{m}} = \begin{bmatrix} \cos(\Xi_{\theta,\text{LTP}}) & -\sin(\Xi_{\theta,\text{LTP}}) \\ \sin(\Xi_{\theta,\text{LTP}}) & \cos(\Xi_{\theta,\text{LTP}}) \end{bmatrix}^{\text{T}} \boldsymbol{p}'_{\tau_i,\text{m}} - \boldsymbol{\Xi}_{\text{LTP}}, \tag{76}$$

$$\hat{\boldsymbol{p}}'_{\tau_{i+1},\text{m}} = \begin{bmatrix} \cos(\Xi_{\theta,\text{LTP}}) & -\sin(\Xi_{\theta,\text{LTP}}) \\ \sin(\Xi_{\theta,\text{LTP}}) & \cos(\Xi_{\theta,\text{LTP}}) \end{bmatrix}^{\text{T}} \boldsymbol{p}'_{\tau_{i+1},\text{m}} - \boldsymbol{\Xi}_{\text{LTP}}. \tag{77}$$

Assuming that $\hat{\boldsymbol{p}}'_{\tau_i,\text{m}}$ and $\hat{\boldsymbol{p}}'_{\tau_{i+1},\text{m}}$ are closest to the i-th and $i+1$-th markers respectively, their measurement errors $\eta_{i,\text{m}}$ and $\eta_{i+1,\text{m}}$ are given by

$$\eta_{i,\text{m}} = \left|\hat{\boldsymbol{p}}'_{\tau_i,\text{m}} - \boldsymbol{p}_{i,\text{map}}\right| = \eta_{i+1,\text{m}} = \left|\hat{\boldsymbol{p}}'_{\tau_{i+1},\text{m}} - \boldsymbol{p}_{i+1,\text{map}}\right|. \tag{78}$$

The previous because, as shown in Equations (71) and (75), the angular and linear offsets are estimated with both LiDAR measurements.

A distance threshold is then used to separate outliers according to the following criteria:

$$\begin{cases} \text{inlier}, & \eta_{i,\text{m}} \leq \frac{\min(\boldsymbol{d}_{\text{map}})}{2} \\ \text{outlier}, & \eta_{i,\text{m}} > \frac{\min(\boldsymbol{d}_{\text{map}})}{2}. \end{cases} \tag{79}$$

If an outlier is detected, both measurements are discarded.

From what is detailed above, the LbPM can be used only in places with known markers, or an extra effort is needed to place and measure markers in new areas. However,

traffic signs have a fixed position and are highly reflective as well. Therefore, they too can be used as markers for Simultaneous Localization and Mapping (SLAM) instead of having a marker library *a priori*.

Ongoing work of the present research focuses on evaluating the adequacy of the LbPM for SLAM purposes. A critical aspect to evaluate an LbPM-based SLAM, is to identify the sources of error individually. Since the vehicle motion prediction between LiDAR measurements can be approximated as circular movement, the LbPM does not depend on the motion model of the EKF. Hence, it is possible to analyze the motion model with no correction data and the LbPM separately. However, the recursiveness of the prediction-correction process of the EKF and localization-mapping process of the SLAM complicate the isolation of the sources of error when the EKF and the LbPM are combined.

To study separately the sources of error, the mapping performance of the LbPM-based SLAM is inspected under two circumstances: (i) with position, velocity and COG correction data; and (ii) with only velocity correction data. The first variant excludes all possible LbPM feedback errors. The second variant excludes only the velocity feedback from the LbPM as a source of error. The presented method is based on [57–60], and the specifics applicable to this work are detailed in what follows.

First, some measurement variables are considered instead as inputs to perform the prediction of vehicle state. An input vector $\boldsymbol{u}_{\mathrm{h}}$ and the control covariance matrix $\boldsymbol{Q}_k$ are then defined as follows

$$\boldsymbol{u}_{\mathrm{h}} = \begin{bmatrix} a_{x,\mathrm{h}} & a_{y,\mathrm{h}} & \theta_{\mathrm{yaw,h}} & v_{\mathrm{h}} \end{bmatrix}^{\mathrm{T}}, \tag{80}$$

$$\boldsymbol{Q}_k = diag\begin{pmatrix} \sigma^2_{a_x,\mathrm{h}} & \sigma^2_{a_y,\mathrm{h}} & \sigma^2_{\dot{\theta}_{\mathrm{yaw}}} & \sigma^2_{v,\mathrm{h}} \end{pmatrix}. \tag{81}$$

With this, the vehicle state process covariance matrix can be updated as follows

$$\mathbf{P}_{\mathrm{xx,k}} = \mathbf{F}_x \mathbf{P}_{\mathrm{xx,k-1}} \mathbf{F}_x^{\mathrm{T}} + \mathbf{F}_u \mathbf{Q}_k \mathbf{F}_u^{\mathrm{T}}, \tag{82}$$

where $\mathbf{F}_u$ is the Jacobian matrix of $f(\boldsymbol{x}_{\mathrm{s}})$, derived after the control vector $\boldsymbol{u}_{\mathrm{h}}$.

Next, a measurement vector $\boldsymbol{z}_{\mathrm{M}}$ is defined as

$$\boldsymbol{z}_{\mathrm{M}} = \begin{bmatrix} d_{i,\mathrm{m}} & \theta_{i,\mathrm{m}} \end{bmatrix}^{\mathrm{T}}, \tag{83}$$

where $d_{i,\mathrm{m}}$ and $\theta_{i,\mathrm{m}}$ are respectively the range and azimuth angle of the i-th cluster.

The task of the observation model is to calculate the estimated measurement $\tilde{\boldsymbol{z}}_{\mathrm{M}}$, for a given vehicle state $\boldsymbol{x}_{\mathrm{s}}$ and one specific marker $\boldsymbol{p}'_{i,\mathrm{m}}$. For this, two cases can be distinguished. If the current marker library is empty, then the marker is added to it with the location derived from $\boldsymbol{x}_{\mathrm{s}}$ and z_{M}. If the marker library is not empty, then an association check is required to compute the probability that the current observation corresponds to an existing marker. For this, the Mahalanobis Distance [61] Individual Compatibility (IC) check is performed for each marker in the library. The one yielding the minimum Mahalanobis Distance is the marker from the library with the highest probability to be the observed one. For this, the innovation term $\mathbf{y}_{\mathrm{M}}$ is calculated as follows

$$\mathbf{y}_{\mathrm{M}} = \boldsymbol{z}_{\mathrm{M}} - \tilde{\boldsymbol{z}}_{\mathrm{M}}. \tag{84}$$

The Mahalanobis Distance is then given by

$$d_{\mathrm{M}} = \sqrt{\mathbf{y}_{\mathrm{M}}^T \mathbf{S}_{\mathrm{M}}^{-1} \mathbf{y}_{\mathrm{M}}}, \tag{85}$$

with $\mathbf{S}_{\mathrm{M}}$ being the corresponding covariance matrix for the innovation term $\mathbf{y}_{\mathrm{M}}$. Finally, the marker from the library is associated with the observation according to the following criteria

$$\begin{cases} \text{associated}, & d_{\mathrm{M}} \leq \Gamma_M \\ \text{not associated}, & d_{\mathrm{M}} > \Gamma_M, \end{cases} \tag{86}$$

$$\Gamma_M = 3. \tag{87}$$

This is because a threshold $\Gamma_M = 3$ means a probability of 98.9 % that the sensor measurement and the estimated measurement refer to the same marker [62].

3. Results

To validate the performance of the From methods that are detailed above, a series of tests are designed and performed. The From methods are tested with several hours of real-world data. In what follows, the testing details, the evaluation metrics and the most relevant results are presented.

3.1. Standstill Recognition

The performance of the standstill recognition is measured in terms of its classification performance according to Table 1, as well as its robustness against false positives. The four most relevant tests and their results are detailed in what follows.

The results of the 1st test are shown on the Figure 6. Here, a 3rd generation Smart ForTwo electric drive is placed on a test track. The INS is mounted in the trunk of the vehicle. The vehicle is turned on, the front tires are set to point straight ahead, and the gear selector placed on "D". Some seconds after the data recording starts, the brake pedal is released. The car is driven on the test track for approximately 530 s. The accelerator pedal is not touched at any point, and the brake pedal is used only at the beginning to let the car roll and at the end to stop it. The steering wheel is used only to realign the vehicle towards paths that allow prolonged straight-line driving.

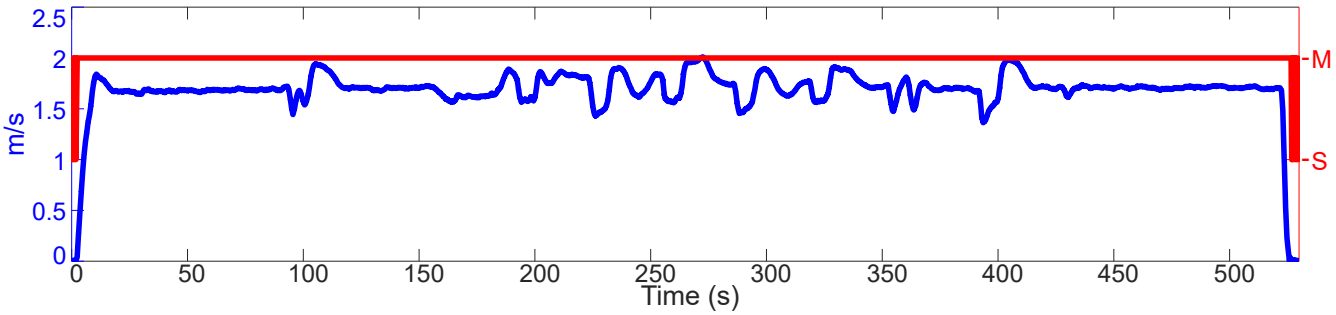

Figure 6. Shown is the RF output (red, S = standstill, M = motion) and the velocity over ground of the vehicle (blue). Vehicle: 3rd Gen. Smart ForTwo. Motorization: Electric Drive. Power source: Electricity.

What makes this test special is the combination of (i) the low driving velocity, (ii) the drivetrain of the vehicle (electrical motor and single gear reduction transmission), and (iii) the prolonged driving moments without rotations.

As can be seen, the RF is able to recognize that the vehicle is in motion, even when driving several seconds constantly at walking velocity on a straight line: (17.87–92.53 s) and (434.80–522.00 s). It is seen as well that the RF has no false positives.

The results of the 2nd test are shown on the Figure 7. Here, a 5th generation Audi A4 3.0L TDI with an S-Tronic 7-gear transmission is driven randomly in a city. The INS is mounted in the trunk of the vehicle. On three occasions, the gear selector is placed on "N", and the car is allowed to roll. On one of those occasions (22.26 s), the vehicle does come to a brief stop. On the other two occasions (138.10 s and 682.20 s), the car reaches very low velocities (0.18 $\frac{\mathrm{m}}{\mathrm{s}}$ and 0.12 $\frac{\mathrm{m}}{\mathrm{s}}$ respectively), but it does not come to a full stop. A prolonged standstill (302.10–652.00 s) is also included. During this period of time, the gear selector is kept in "S".

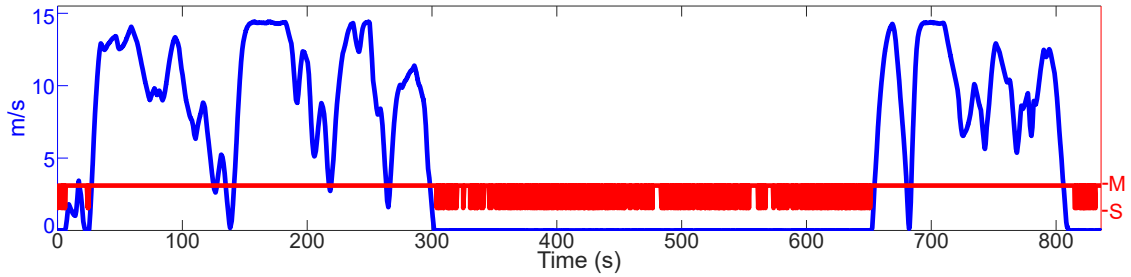

Figure 7. Shown is the RF output (red, S = standstill, M = motion) and the velocity over ground of the vehicle (blue). Vehicle: 5th Gen. Audi A4. Motorization: 6 cylinder, 3.0 L TDI. Power source: Diesel.

This test shows (i) three rolling instances, and (ii) a prolonged standstill. Diesel engines are known to produce more vibrations than their gasoline counterparts. Also, when the gear selector is placed on "N", much less vibrations are transmitted to the vehicle chassis because the transmission is not engaged, and because the engine idles. Therefore, the IMU signals when the vehicle rolls closely resemble those when the car is standing still, especially at low velocities. Contrarily, when the gear selector is on "S" instead of "D", the vibrations transmitted to the chassis increase notably.

As can be seen, in the first rolling instance (22.26 s) the RF is able to recognize the standstill first when the vehicle does come to a stop and not before. Also, the RF can recognize that the vehicle is still in motion at the other two rolling instances (138.10 s and 682.20 s), despite the velocity over ground almost reaching zero. The RF is also mostly able to recognize that the vehicle is standing still (302.10–652.00 s), despite the increased vibrations provoked by selecting "S". After 720.20 s of true standstill, the estimated total displacement of the vehicle is 0.008 m (8 mm) and the estimated total rotation of the vehicle is 0.011 rad (0.6282 deg). It is seen as well that the RF has no false positives.

The results of the 3rd test are shown on the Figure 8. Here, a Suzuki GSX-R750 K2 is driven on a test track. The INS is mounted by means of a metal plate directly on the chassis of the vehicle. On four occasions (251.80 s, 327.60 s, 391.20 s and 457.10 s), the "N" gear is selected, and the vehicle can roll to a full stop. Various instances of several seconds of standstill are included as well. In two of those instances (43.88–53.75 s and 94.08–108.20 s), the vehicle was let to rest on its side stand, with the motor idling and the gear "N" selected.

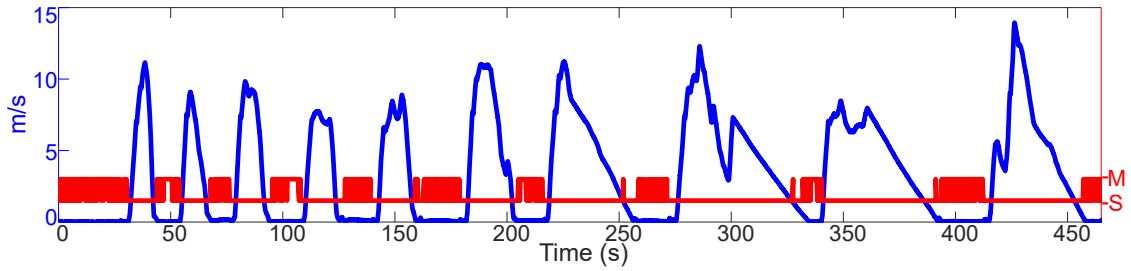

Figure 8. Shown is the RF output (red, S = standstill, M = motion) and the velocity over ground of the vehicle (blue). Vehicle: Suzuki GSX-R750 K2. Motorization: 4 cylinder, 749 cc. Power source: Gasoline.

What makes this test special are (i) the chassis-drivetrain configuration, and (ii) the rolling instances. By design, the engine mounts of motorcycles are not designed to dampen the engine vibrations as well as the engine mounts of cars. Thus, these vibrations are transmitted in a much more direct manner to the motorcycle chassis. Also, given that the INS is mounted on the chassis with no dampening, it senses the engine vibrations in a much more direct manner. Thus, the INS signals under these conditions, in combination with the rolling instances, echo the signals that are present at standstill.

As can be seen, on two of the occasions where the motorcycle can roll (251.80 s and 327.60 s), false positives appear. On the other two occasions (391.20 s and 457.10 s), the RF

detects a standstill first when the vehicle comes to a stop. It is seen as well that the RF has no false positives under regular driving conditions (driving always on gear).

The results of the 4th test are shown on the Figure 9. Here, a 2nd generation Audi Q7 is parked on a test track with the engine idling and the gear "P" selected. The INS is mounted in the trunk of the vehicle. The volume of the sound system is set to the maximum and music with high bass level is played for 254 s, to induce very strong vibrations for prolonged periods of time. The objective is to test if the RF can recognize the standstill regardless of the induced vibrations.

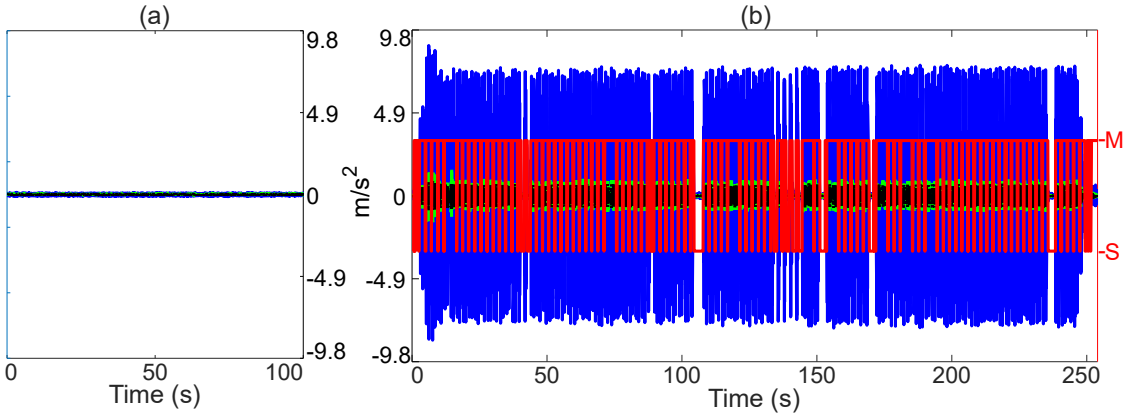

Figure 9. Shown are the vehicle accelerations along the z_{LCP} (blue), y_{LCP} (green) and x_{LCP} (black) axes. (**a**) Shows the vehicle accelerations of the RF training dataset with the label "Standstill". (**b**) Shows the vehicle accelerations while the music is playing and the RF output (red, S = standstill, M = motion). Vehicle: 2nd Gen. Audi Q7. Motorization: 6 cylinder, 3.0 L TFSI. Power source: Gasoline.

What makes this test special is the constant induction of vibrations which by magnitude is many times bigger than that of when the vehicle is standing still.

As can be seen, the output of the RF toggles much more than in the other tests. However, it is often still able to correctly detect a standstill. The maximum period of time where the vehicle state is wrongly classified as "motion" is 4.20 s, between 190.3 s and 194.50 s. In this test, after 254.01 s of true standstill, the estimated total displacement of the vehicle is 0.12 m (12.12 cm) and the estimated total rotation of the vehicle is 0.018 rad (1.08 deg).

3.2. Horizontation of IMU Measurements

The performance of this module is measured in terms of its capability to describe the pose (roll, pitch and yaw angles) of a vehicle. The motorcycle from the 3rd test of Section 3.1 is used because bigger roll angles can be achieved with it than with a car. The motorcycle is driven randomly on a test track, including a U-turn (772.70–780.00 s), a slalom (783.30–791.00 s), and "8" figures (791.00–843.70 s).

What makes this test special is that it serves as a practical demonstration of the mathematical methods detailed in Section 2.2. The results of the horizontation of the IMU measurements are shown on the Figure 10. As can be seen, the practical implementation of the mathematical methods is able to describe the vehicle pose.

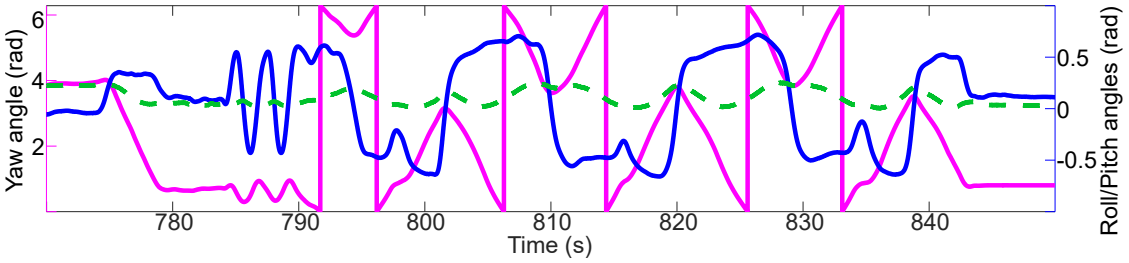

Figure 10. Shown are the roll (blue), pitch (green) and yaw (magenta) angles. Vehicle: Suzuki GSX-R750 K2. Motorization: 4 cylinder, 749 cc. Power source: Gasoline.

3.3. Statistical Filtering

The results of the statistical filtering are shown on the Figure 11. The performance of this module is measured in terms of its ability to (i) fuse the vehicle path (as estimated with the motion model) with correction data, (ii) estimate the vehicle path despite SatNav shortages or with corrupted SatNav measurements, and (iii) estimate the vehicle path solely by means of the motion model (with no correction data). For this, a vehicle is driven in various types of roads for 1904.15 s. The test starts at a country road (blue), followed by an express way (magenta, thick), then in-city driving (magenta, thin) and a parking structure (green). A standstill instance of 40.00 s inside the parking structure is included as well. The average velocity of the vehicle during the test, including the 40.00 s standstill inside the parking structure, is 8.19 $\frac{\mathrm{m}}{\mathrm{s}}$.

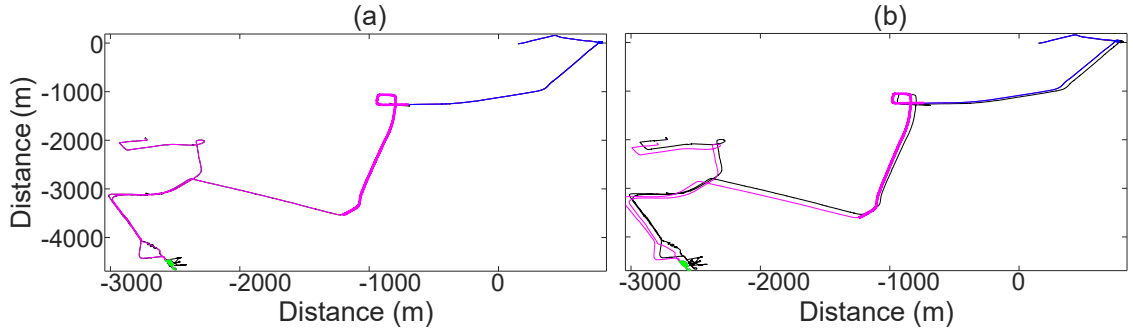

Figure 11. Shown is the vehicle position as measured by a SatNav receiver (black) and as estimated by means of statistical filtering (blue, magenta and green). (**a**) Shows the statistical filtering when SatNav correction data is used. (**b**) Shows the statistical filtering with no correction data.

What makes this test special is (i) the inclusion of various driving conditions, (ii) the inclusion of extended periods of time without or with corrupted SatNav correction data (parking structure), and (iii) the dead-reckoning navigation.

As can be seen, the statistical filtering is able to fuse the estimation of the motion model with the SatNav measurements. It can also be seen that even in total absence of SatNav correction data, the motion model is able to accurately estimate the vehicle path. The test shown in Figure 11 is a representative case of the average accuracy of the motion model because it includes various driving situations. At the end of the test, the cumulative deviation of the position estimated by the motion model from the position measured by the SatNav receiver is 130.00 m, which accounts for a deviation of 245.79 m per driven hour at an average velocity of 8.19 $\frac{\mathrm{m}}{\mathrm{s}}$.

3.4. Outlier Detection

The performance of this module is measured in terms of its ability to recognize corrupted correction data, even when the sensors suggest otherwise. For this, two segments of the results of Section 3.3 are highlighted.

What makes this test special are the extended periods of time without or with corrupted correction data. As can be seen, the outlier detection recognizes the faulty correction data, thus dynamically adapting the Kalman gains accordingly. This greatly improves the estimation of the vehicle state. Some relevant results of the outlier detection are shown on the Figure 12.

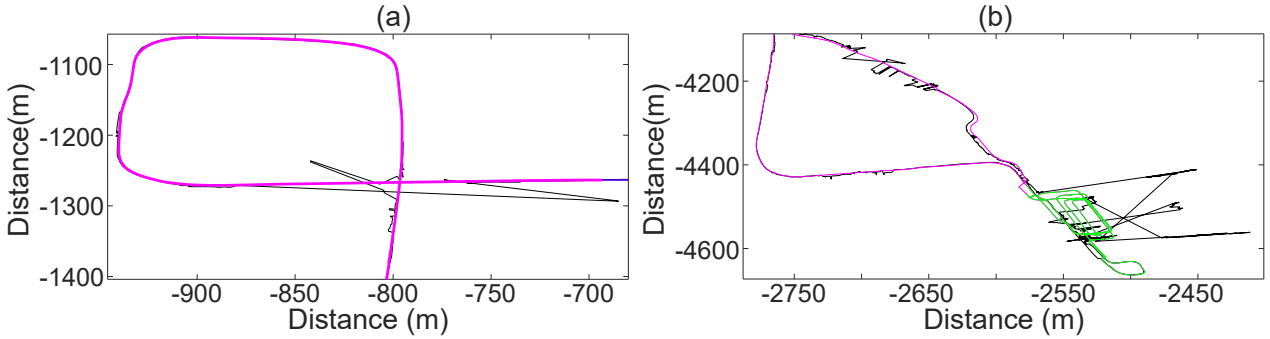

Figure 12. Shown is the vehicle position as measured by a SatNav receiver (black) and as estimated by means of statistical filtering (blue, magenta and green). (**a**) Shows the dataset section of a bridge underpass. (**b**) Shows the dataset section of a parking structure.

3.5. Drift Recognition

A representative result of the drift recognition is shown on the Figure 13. The performance of this module is measured in terms of its ability to recognize that the vehicle is drifting. For this, a 3rd generation BMW M5 is driven 51 times on a test track, alternating between tractive and non-tractive driving. The test vehicle is equipped with an INS and a Correvit S-Motion. No correction data is used in these tests to test the accuracy of the motion model under these conditions.

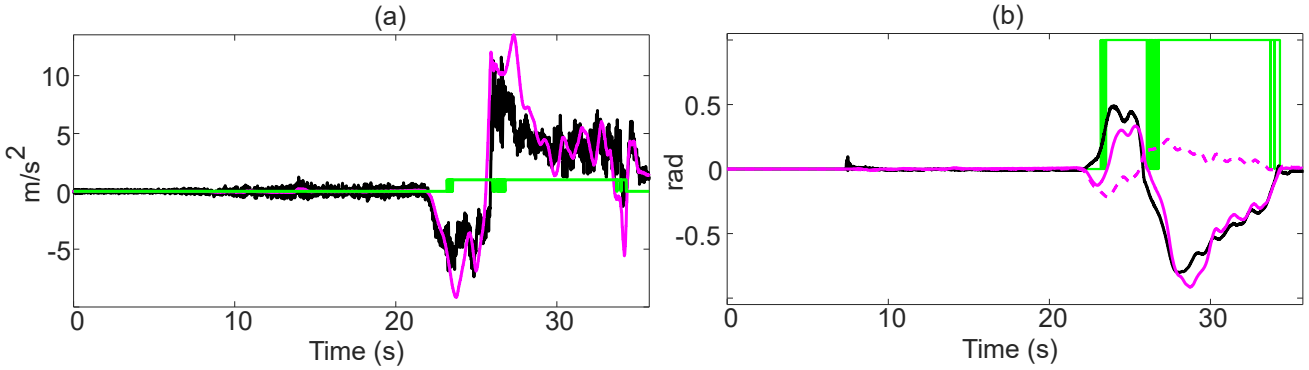

Figure 13. Shown are (**a**) the drift bit (green), the estimated (black) and expected (magenta) lateral acceleration, (**b**) the drift bit (green), the measured (black) and estimated (magenta solid, magenta dashed) sideslips. Drift bit = 1 means drifting.

The relevance of this test lies in (i) the extended periods of non-tractive driving, and (ii) the exclusion of correction data. As can be seen, when the vehicle enters the first curve, the module recognizes that the vehicle is not drifting. It is only when the sideslip starts to abruptly change (22.98 s) that the module starts to detect a drift. As the vehicle remains on a steady state drift (23.54–26.05 s), the drift bit stabilizes at "drifting". During the drift transition (26.05–26.83 s), when there are moments of tractive driving, the drift bit toggles between tractive and non-tractive driving. Once the transition is finished, and the vehicle enters again into a steady state drift, the drift bit does not toggle any more. As the vehicle ends the drift (34.31 s) the drift bit toggles for a few milliseconds before stabilizing at tractive driving.

The average deviation of the final position as estimated by the motion model when compared to the position measured by the SatNav receiver for all 51 tests is ≈24.47 s, what accounts for a deviation of 2465.30 m per driven hour at 6.35 $\frac{\mathrm{m}}{\mathrm{s}}$. However, it should be noted that (i) the average duration of these tests is ≈35 s, (ii) there is tractive and non-tractive driving in each test, (iii) around the first 20 s of each test are solely straight-line driving, and (iv) there is a full reset at the beginning of every test.

3.6. LiDAR-Based Positioning Method

The accuracy results of the LbPM are shown on the Table 2. The performance of this module is measured in terms of its accuracy to measure the vehicle state. For this, a vehicle is equipped with an INS with RTK and a LiDAR. An array of reflective markers is placed on a test track, and the marker library is generated by measuring the markers with a SatNav receiver with RTK. Two maneuvers (a slalom and a drive-by) are driven with various velocities ranging from 5 $\frac{\text{km}}{\text{h}}$ and up to 40 $\frac{\text{km}}{\text{h}}$. The outputs from the LbPM are then compared to those of the INS with RTK. It should be noted that the INS serves as an accurate and objective reference, but no correction data is used for the LbPM.

What makes this test special are (i) the exclusion of correction data, and (ii) the validation with both maneuvers and various velocities. As can be seen, the mean accuracy of the LbPM is almost the same as that of the INS with RTK for all the state variables that the LbPM can measure.

Table 2. Accuracy results for the proposed LbPM. Shown are the maneuvers, maneuverer velocity, mean deviation from the reference and Std. dev. of the corresponding errors.

Manoeuvrers	**Positioning Accuracy**		**Orientation Accuracy**		**Velocity Accuracy**	
Drive-by	**Mean (m)**	**Std. dev. (m)**	**Mean (deg)**	**Std. dev. (deg)**	**Mean** $\left(\frac{\text{m}}{\text{s}}\right)$	**Std. dev.** $\left(\frac{\text{m}}{\text{s}}\right)$
5 $\frac{\text{km}}{\text{h}}$	0.04	0.02	0.73	0.25	0.06	0.08
10 $\frac{\text{km}}{\text{h}}$	0.03	0.02	0.19	0.20	0.08	0.10
15 $\frac{\text{km}}{\text{h}}$	0.03	0.02	0.26	0.19	0.07	0.09
20 $\frac{\text{km}}{\text{h}}$	0.03	0.02	0.37	0.23	0.08	0.09
25 $\frac{\text{km}}{\text{h}}$	0.04	0.02	0.58	0.23	0.08	0.10
30 $\frac{\text{km}}{\text{h}}$	0.06	0.02	0.51	0.22	0.08	0.10
35 $\frac{\text{km}}{\text{h}}$	0.07	0.03	0.44	0.25	0.08	0.11
40 $\frac{\text{km}}{\text{h}}$	0.08	0.03	0.41	0.26	0.11	0.13
Slalom	**Mean (m)**	**Std. dev. (m)**	**Mean (deg)**	**Std. dev. (deg)**	**Mean** $\left(\frac{\text{m}}{\text{s}}\right)$	**Std. dev.** $\left(\frac{\text{m}}{\text{s}}\right)$
5 $\frac{\text{km}}{\text{h}}$	0.04	0.02	0.24	0.29	0.08	0.11
10 $\frac{\text{km}}{\text{h}}$	0.04	0.02	0.40	0.29	0.09	0.12
20 $\frac{\text{km}}{\text{h}}$	0.04	0.02	0.32	0.36	0.14	0.17
30 $\frac{\text{km}}{\text{h}}$	0.05	0.02	0.36	0.40	0.18	0.24
40 $\frac{\text{km}}{\text{h}}$	0.10	0.02	0.53	0.43	0.18	0.22

3.7. LbPM Adequacy for SLAM

The adequacy of the LbPM for SLAM is measured in terms of its accuracy to generate the marker library. For this, the same test setup from Section 3.6 is used. Two tests are performed. In the first test, the position, velocity and orientation of an INS with RTK is used as correction data. In the second test, only the velocity from the INS with RTK is used as correction data. In both tests, the location of the markers is computed with the estimated vehicle state and the LiDAR measurements. The observed location of the markers is then compared to their true position from the library.

The relevance of these tests lies in that (i) it allows identify some sources of error, and (ii) it allows evaluation of the identified sources of error individually. The results are presented in Table 3. As can be seen, the marker library that is generated when the INS-RTK correction data is present, closely resembles the marker library that is generated with the RTK SatNav receiver. As expected, the marker library that is generated with only velocity correction data is less accurate. However, this inaccuracy is still smaller than that of SatNav with no correction data.

3.8. Runtime

The runtime results of the presented methodology are shown on Table 4. These values are obtained by executing the Matlab code of each module for 1.1+ million cycles on an Intel i7-6820HQ CPU, and by using the Matlab Profiler to measure the runtime of each module. Given that the IMU horizontation, the outlier and the drift detectors are embedded in the statistical filtering module, the runtime of these four modules is considered to be a single one.

Table 3. Results of the mapping accuracy. Shown are the maneuvers, driving velocity, mean deviation from the observed to the true marker location, std. dev. of the errors and the maximum deviation. The columns (**a**) show the results when using position, velocity and orientation correction data. The columns (**b**) show the results when using only velocity correction data.

Manoeuvre	**Mapping Accuracy**			**Manoeuvre**	**Mapping Accuracy**		
Drive-by	**Mean (m)** (**a**)/(**b**)	**Std. dev. (m)** (**a**)/(**b**)	**Max (m)** (**a**)/(**b**)	**Slalom**	**Mean (m)** (**a**)/(**b**)	**Std. dev. (m)** (**a**)/(**b**)	**Max (m)** (**a**)/(**b**)
5 $\frac{km}{h}$	0.11/0.90	0.04/0.51	0.19/1.56	5 $\frac{km}{h}$	0.33/0.93	0.24/0.51	0.81/1.62
10 $\frac{km}{h}$	0.11/0.13	0.04/0.06	0.19/0.22	10 $\frac{km}{h}$	0.08/0.27	0.03/0.11	0.14/0.42
15 $\frac{km}{h}$	0.12/0.18	0.05/0.07	0.22/0.34	15 $\frac{km}{h}$	0.08/0.53	0.03/0.25	0.16/1.04
20 $\frac{km}{h}$	0.14/0.29	0.06/0.12	0.28/0.48	20 $\frac{km}{h}$	0.08/0.78	0.03/0.56	0.14/1.70
25 $\frac{km}{h}$	0.17/0.11	0.08/0.06	0.32/0.39	25 $\frac{km}{h}$	0.10/0.30	0.05/0.14	0.23/0.64
30 $\frac{km}{h}$	0.14/0.16	0.06/0.09	0.28/0.44	30 $\frac{km}{h}$	0.08/0.67	0.04/0.36	0.21/1.53
35 $\frac{km}{h}$	0.10/0.20	0.05/0.15	0.20/0.69	35 $\frac{km}{h}$	0.11/0.96	0.06/0.52	0.25/1.96
40 $\frac{km}{h}$	0.10/0.38	0.05/0.19	0.23/0.77	40 $\frac{km}{h}$	0.11/0.26	0.06/0.15	0.25/0.64

Table 4. Shown is the median and the std. dev. of the runtime of the modules presented above.

Module	**Median (µs)**	**Std. dev. (µs)**
Standstill classifier	123	50
Statistical filtering	403	98
LbPM-point clustering	43	20
LbPM-velocity estimation	11	14
LbPM-position estimation	42	54

The relevance of this test lies in that it gives a baseline to estimate the possibility of using the proposed methodology in real-time applications. As can be seen, the median runtime of all modules is always in the microsecond area. Even in a worst-case scenario (median plus 3σ), the runtime for the complete methodology is ≈1.33 ms, which remains under a typical IMU sampling time of 10 ms.

4. Discussion

As stated in Section 1, the focus of this research is to generate a reference state for ground vehicles, while reducing the dependency of the INSs on the SatNav. This allows the INSs to bridge SatNav outages for much longer periods of time, or even to function with no SatNav at all. Specifically, for the automotive industry, there are various use-cases that justify the use of SatNav-deprived INSs, such as the navigation in tunnels, underpasses or parking structures. A highly precise and prolonged navigation in places with no SatNav, such as testing halls, is also very relevant for the automotive research. As shown in Section 3, the methods developed in this research work address the aspects where the INSs profit the most.

Starting with the standstill recognition, it is shown that it is possible to classify whether a vehicle is moving or standing still by using only machine learning techniques and IMU measurements. The fact that no additional sensor is required, clearly implies important advantages, such as (i) reduced costs, (ii) less testing complexity, (iii) simplified information processing, and (iv) stand-alone functioning.

The proposed method uses the same features in the frequency domain, regardless of the vehicle. However, a first glance at the Laplace transformation of the features in the time domain, suggests that the choice of frequencies can be further refined if one considers them on a vehicle-specific basis. This could eventually lead to an improvement in the standstill recognition. Therefore, future work could include a vehicle-specific analysis of the features in the frequency domain. The use of the on-board vehicle sensors could also help refine and automate the data labeling.

As for the horizontation of INS measurements, modern INSs do use correction data (typically from the SatNav) to refine the IMU measurements. However, it is shown that the pose of the vehicle can be accurately described with the raw IMU measurements. This implies independence from the SatNav. The inclusion of the detailed mathematics to horizontate IMU measurements, implies an aid to scientists working on similar topics or trying to replicate the results presented here.

One challenging aspect of the horizontation of the IMU measurements is to detect the IMU mounting pose in the vehicle, i.e., it is possible to estimate the pose of the IMU with respect to the gravity vector and the true north of the Earth. However, it is not an easy task to relate the pitch and roll of the IMU with that of the vehicle because the IMU could be mounted in such a manner that its $x \times y$-plane is not parallel to the $x \times y$-plane of the vehicle. Knowing the offsets between both planes could be highly valuable to recognize certain driving situations, such as driving inside parking structures, wheelies or stoppies. Since the acquisition of these offsets on a vehicle-to-vehicle basis is a complicated and time-consuming task, future work could include the investigation of automated methods that allow estimation of these offsets.

The KF is a computationally efficient algorithm for fusing data. However, the accuracy of its output is limited by the accuracy of the information to fuse. Therefore, it is important to refine the sources of information as much as possible before using them as inputs for the KF. In this work, the EKF fuses the state vector estimated by the motion model with the measurements of the vehicle state made with external sensors. Therefore, a lot of effort is put on refining the motion model for tractive-driving applications. As shown above, the proposed motion model can accurately estimate the vehicle state for extended periods of time with no correction data, which implies a high confidence on the vehicle state, even when navigating by dead reckoning.

Given that the motion model is designed to perform best for tractive driving, its performance is reduced during non-tractive driving. Also, given that the drifting tests are designed to test the drift recognition, they are not long enough to objectively determine the accuracy of the motion model during a drift. Therefore, future work could include a deeper investigation of the accuracy of the motion model during non-tractive driving. This could help to estimate adequate values for the system noise, and so improve the performance of the KF during drifting.

As for the outlier detector, it further helps in refining the sources of information before they are fed to the KF. Its effect is especially noticeable when the sensors deliver quality metrics for their own measurements, as is the case of the standard deviation for the SatNav. As shown above, to include a smart outlier detector implies a very useful consistency check of the sensor measurements, thus avoiding fusing information that would negatively affect the estimation of the vehicle state.

This detector uses the latest state vector because it is the last known vehicle state. However, as time passes, the predicted state without correction data might differ so much from the true state that the constraints shown in Section 2.5 filter out meaningful sensor

measurements. Future work could include the combination of the system noise with the state vector to better adapt the outlier detector.

Yet another step that is taken to improve the KF is the drift detector. Given that the motion model is designed to function best in tractive driving, it is very useful to differentiate whether the vehicle is drifting or not. As shown above, the drift recognition can make this distinction. This implies that the time instance to make changes in the KF is known. This can be, for example, to adjust the system noise accordingly or to use a more adequate motion model.

The drift recognition here presented is based on the thresholds of two features: the magnitude of the sideslip and the difference between the estimated and expected lateral acceleration. Future work on this method could include the analysis of a bigger dataset that includes vehicles with different drivetrains (front, rear and all-wheel drive). This could help to gain a better understanding on the ideal thresholds for vehicles according to their drivetrain.

One crucial module to reduce the dependency of INSs on the SatNav is the LbPM. The results clearly show that the accuracy of the vehicle state as measured by this method closely resembles that of an INS with RTK correction data. Considering the refresh rate of the LiDAR and the state variables that can be measured by means of the LbPM, this could already be enough for an accurate vehicle indoor navigation. Future work on this module could include the implementation of the method on dedicated hardware to analyze the computing requirements for a real-time implementation.

The results on Table 3, suggest that the LbPM is adequate for SLAM. Given that the LbPM is capable of delivering RTK-like correction data, it could enable the generation of an accurate marker library in unknown environments. It is not clear whether or not a lane-accurate indoor navigation could be possible in places with a reduced marker density. Future work on this module could include adapting the LbPM algorithm for SLAM, and implementing it on dedicated hardware for real-world testing. For this purpose, tailored-made hardware solutions, such as [63,64], could aid in the navigation of complex environments, as can be parking structures. The use of lightweight artificial intelligence techniques, as shown in [65,66] could aid in the implementation of the standstill recognition in hardware with limited computational resources, as are embedded systems. Finally, the sensor fusion with on-board sensors in the vehicles could improve the dead-reckoning navigation in indoor environments.

Regarding the methodology runtime, it should be noted that even with high-level programming code, such as a Matlab, its runtime never exceeded 2 ms. This implies that it is possible to implement the methodology in real-time applications. Future work could include the implementation using a more efficient programming language, such as C++, and the use of dedicated hardware.

5. Conclusions

The objective of generating a reference vehicle state for proper testing and validation of automated driving functions is achieved in a prototypical manner by developing novel methods and by adapting existing ones to the problems at hand. This while reducing the dependency of INSs on external sensors, such as the SatNav. The testing results demonstrate that the proposed methods greatly improve the accuracy of the estimated vehicle state. The measured runtime suggests the possibility of real-time implementation of the proposed methodology.

Supplementary Materials: The following are available online at https://www.mdpi.com/1424-8220/21/4/1131/.

Author Contributions: E.S.M. contributed with most of the presented methodology, its implementation and its validation. J.D. contributed with the methodology of the LiDAR-based SLAM. B.H. contributed with the conceptualization of the methodology. A.G.H. contributed with the supervision of the methodology. M.B. contributed with the conceptualization and supervision of the methodology, the funding acquisition and the project administration. All authors contributed in the writing, review and editing. Andrés García Higuera is a policy analyst in the European Parliamentary Research Service (EPRS), the internal research service and think-tank of the European Parliament. He is writing in a personal capacity and any views expressed do not represent an official position of the Parliament. All authors have read and agreed to the published version of the manuscript.

Funding: This research and the APC was funded by the Federal Ministry of Education and Research of Germany (Bundesministerium für Bildung und Forschung) in the framework of FH-Impuls, project number 03FH7I02IA.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Datasets of the IMU signals and SatNav measurements of the results shown are available within the article as Supplementary Materials. The data presented in this study are available within the article as Supplementary Materials.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SatNav	Satellite Navigation
IMU(s)	Inertial Measurement Unit(s)
INS(s)	Inertial Navigation System(s)
RTK	Real-Time Kinematic
LbPM	LiDAR-based Positioning Method
RF	Random Forest
DFT	Discrete Fourier Transform
RPM	Revolutions Per Minute
COG	Course Over Ground
EKF	Extended Kalman Filter
GPS	Global Positioning System

References

1. Einstein, A. *Relativity*; The University of Sheffield: Sheffield, UK, 1920.
2. Schulze, O. Wirbelstrom-Tachometer. German DRP146134, 7 October 1902.
3. Tindell, K.; Burns, A. Guaranteeing message latencies on Controller Area Network (CAN). In Proceedings of the 1st International CAN Conference, Mainz, Germany, 13–14 September 1994; pp. 1–11.
4. Tindell, K.; Hanssmon, H.; Wellings, A.J. Analysing Real-Time Communications: Controller Area Network (CAN). In Proceedings of the RTSS Citeseer, San Juan, Puerto Rico, 7–9 December 1994; pp. 259–263.
5. Tindell, K.; Burns, A.; Wellings, A.J. Calculating controller area network (CAN) message response times. *Control Eng. Pract.* **1995**, *3*, 1163–1169. [CrossRef]
6. Casparsson, L.; Rajnak, A.; Tindell, K.; Malmberg, P. *Volcano—A Revolution in On-Board Communications*; Technical Report; Volvo: Gothenburg, Sweden, 1998.
7. DeMeis, R. Cars sag under weighty wiring. *EE Times* **2005**. [CrossRef]
8. Davis, R.I.; Burns, A.; Bril, R.J.; Lukkien, J.J. Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised. *Real-Time Syst.* **2007**, *35*, 239–272. [CrossRef]
9. Davis, R.I.; Navet, N. Controller area network (CAN) schedulability analysis for messages with arbitrary deadlines in FIFO and work-conserving queues. In Proceedings of the 2012 9th IEEE International Workshop on Factory Communication Systems, Lemgo, Germany, 21–24 May 2012; pp. 33–42. [CrossRef]
10. Dziurzanski, P.; Davis, R.I.; Indrusiak, L.S. Synthesizing Real-Time Schedulability Tests using Evolutionary Algorithms: A Proof of Concept. In Proceedings of the 2019 IEEE Real-Time Systems Symposium (RTSS), Hong Kong, China, 3–6 December 2019; pp. 43–55. [CrossRef]
11. WPT Sensors: Wheel Pulse Transducers. Available online: https://www.kistler.com/files/document/000-811e.pdf (accessed on 11 November 2019).

12. Correvit S-Motion DTI: 2-axis Optical Sensors. Available online: https://www.kistler.com/files/document/003-395e.pdf (accessed on 1 January 2020).
13. Haus, J.; Lauinger, N. Optische Gitter: Die Abbildung der Realität—75 Jahre berührungslose dynamische Meßtechnik auf der Basis optischer Gitter. *Laser Tech. J.* **2007**, *4*, 43–47. [CrossRef]
14. Robert Bosch GMBH Vehicle Standstill Recognition. German PCT/US2013/074650, 12 December 2013.
15. Kalman, R. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng.* **1960**, *82*, 35–45. [CrossRef]
16. Zubov, I.; Afanasyev, I.; Gabdullin, A.; Mustafin, R.; Shimchik, I. Autonomous Drifting Control in 3D Car Racing Simulator. In Proceedings of the 2018 International Conference on Intelligent Systems (IS), Madeira, Portugal, 25–27 September 2018; pp. 235–241. [CrossRef]
17. Wang, H.; Zhao, L.; Yang, Y.; Yan, Y.; Liu, J. Modeling and analysis of wheel mobile robot for high-speed precise drift. In Proceedings of the 31st Chinese Control Conference, Heifei, China, 25–27 July 2012; pp. 5064–5069.
18. Bárdos, A.; Domina, A.; Szalay, Z.; Tihanyi, V.; Palkovics, L. MIMO Controller Design for Stabilizing Vehicle Drifting. In Proceedings of the 2019 IEEE 19th International Symposium on Computational Intelligence and Informatics and 7th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Sciences and Robotics (CINTI-MACRo), Szeged, Hungary, 14–16 November 2019; pp. 000187–000192. [CrossRef]
19. El-Saigh, A.I.; Macario, R.C.V. A review of anti-multipath techiques, past and present. In Proceedings of the IEE Colloquium on Multipath Countermeasures, London, UK, 23 May 1996.
20. Burr, A.G. The multipath problem: an overview. In Proceedings of the IEE Colloquium on Multipath Countermeasures, London, UK, 23 May 1996.
21. Cheng, L.; Chen, J.; Gan, M. Multipath error analysis of carrier Tracking Loop in GPS receiver. In Proceedings of the 29th Chinese Control Conference, Beijing, China, 28–31 July 2010; pp. 4137–4141.
22. Liu, L.; Amin, M.G. Comparison of Average Performance of GPS Discriminators in Multipath. In Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07, Honolulu, HI, USA, 15–20 April 2007; Volume 3, pp. 1285–1288.
23. Yedukondalu, K.; Sarma, A.D.; Kumar, A. Mitigation of GPS multipath error using recursive least squares adaptive filtering. In Proceedings of the 2010 IEEE Asia Pacific Conference on Circuits and Systems, Kuala Lumpur, Malaysia, 6–9 December 2010; pp. 104–107.
24. Groves, P.D. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*; Artech House: London, UK, 2013.
25. Farrell, J. *Aided Navigation: GPS with High Rate Sensors*; McGraw-Hill Inc.: London, UK, 2008.
26. Program Funding. Available online: https://www.gps.gov/policy/funding/ (accessed on 13 January 2021).
27. Annual Activity Report of the European GNSS Agency 2019. Available online: https://www.gsa.europa.eu/sites/default/files/annual_report_gsa_2019.pdf (accessed on 12 January 2021).
28. New Civil Signals. Available online: https://www.gps.gov/systems/gps/modernization/civilsignals/ (accessed on 13 January 2021).
29. Release No: CR-048-17. Available online: https://www.defense.gov/News/Contracts/Contract-View/Article/1112618/ (accessed on 16 April 2018).
30. Lam, K. Broadcom Introduces World's First Dual Frequency GNSS Receiver with Centimeter Accuracy for Consumer LBS Applications. Available online: https://www.broadcom.com/company/news/product-releases/2302120 (accessed on 16 April 2018).
31. Beidou Upgrades for Global Reach. Available online: https://en.unicorecomm.com/news/detail/18 (accessed on 16 April 2018).
32. Batistić, L.; Tomic, M. Overview of indoor positioning system technologies. In Proceedings of the 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 21–25 May 2018; pp. 473–478. [CrossRef]
33. Einsiedler, J.; Radusch, I.; Wolter, K. Vehicle indoor positioning: A survey. In Proceedings of the 2017 14th Workshop on Positioning, Navigation and Communications (WPNC), Bremen, Germany, 25–26 October 2017; pp. 1–6. [CrossRef]
34. iBeacon-Apple Developer. Available online: https://developer.apple.com/ibeacon/ (accessed on 20 April 2020).
35. Ibisch, A.; Stümper, S.; Altinger, H.; Neuhausen, M.; Tschentscher, M.; Schlipsing, M.; Salinen, J.; Knoll, A. Towards autonomous driving in a parking garage: Vehicle localization and tracking using environment-embedded LIDAR sensors. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, Australia, 23–26 June 2013; pp. 829–834. [CrossRef]
36. Harter, A.; Hopper, A.; Steggles, P.; Ward, A.; Webster, P. The Anatomy of A Context-Aware Application. *Wirel. Netw.* **2002**, *8*, 187–197. [CrossRef]
37. Ward, A.; Jones, A.; Hopper, A. A New Location Technique for the Active Office. *IEEE Personal Communications* **1997**, *4*, 42–47. [CrossRef]
38. Addlesee, M.; Curwen, R.; Hodges, S.; Newman, J.; Steggles, P.; Ward, A.; Hopper, A. Implementing a sentient computing system. *Computer* **2001**, *34*, 50–56. [CrossRef]
39. Indoor Positioning System-Data Drives Innovation. Available online: https://indoo.rs/solution/indoor-positioning-system/ (accessed on 12 May 2020).
40. Quick Start: Indoor Positioning Systems. Available online: https://www.infsoft.com/indoor-positioning (accessed on 1 May 2020).
41. What Is Indoor Positioning Systems? Available online: https://senion.com/indoor-positioning-system/#difference (accessed on 20 May 2020).

42. Gatesichapakorn, S.; Takamatsu, J.; Ruchanurucks, M. ROS based Autonomous Mobile Robot Navigation using 2D LiDAR and RGB-D Camera. In Proceedings of the 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), Bangkok, Thailand, 16–18 January 2019; pp. 151–154. [CrossRef]
43. Toth, C.; Grejner-Brzezinska, D.A.; Lee, Y. Terrain-based navigation: Trajectory recovery from LiDAR data. In Proceedings of the 2008 IEEE/ION Position, Location and Navigation Symposium, Monterey, CA, USA, 5–8 May 2008; pp. 760–765. [CrossRef]
44. Cong, D.; Zhang, L.; Su, P.; Tang, Z.; Meng, Y.; Wang, Y. Design and implementation of LiDAR navigation system based on triangulation measurement. In Proceedings of the 2017 29th Chinese Control And Decision Conference (CCDC), Chongqing, China, 28–30 May 2017; pp. 6060–6063. [CrossRef]
45. ISO 8855:2011. *ISO/TC 22/SC 33 Vehicle Dynamics and Chassis Components*; Technical Report; International Organization for Standardization: Geneva, Switzerland, 2011.
46. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
47. Frigo, M.; Johnson, S.G. The Design and Implementation of FFTW3. *Proc. IEEE* **2005**, *93*, 216–231. [CrossRef]
48. Johnson, S.G.; Frigo, M. A modified split-radix FFT with fewer arithmetic operations. *IEEE Trans. Signal Process.* **2007**, *55*, 111–119. [CrossRef]
49. Frigo, M.; Leiserson, C.E.; Prokop, H.; Ramachandran, S. Cache-oblivious algorithms. In Proceedings of the 40th Annual Symposium on Foundations of Computer Science, New York, NY, USA, 17–19 October 1999; pp. 285–297.
50. Nyquist, H. Certain Topics in Telegraph Transmission Theory. *Trans. Am. Inst. Electr. Eng.* **1928**, *47*, 617–644. [CrossRef]
51. Shannon, C.E. Communication in the Presence of Noise. In *Proceedings of the 1949 Proceedings of the Institute of Radio Engineers*; IEEE: New York, NY, USA, 1949; Volume 37, pp. 10–21. [CrossRef]
52. Murray, G. Rotation about an Arbitrary Axis in 3 Dimensions. Available online: https://tinyurl.com/y6rvtbtx (accessed on 17 November 2020).
53. Caicedo, M. Cinemática de las Rotaciones. Available online: http://hc09paa1.pbworks.com/f/FisII_rotation.pdf (accessed on 15 November 2020).
54. Abdulrahim, M. On the Dynamics of Automobile Drifting. *SAE Mobilus* **2006**. [CrossRef]
55. Velodyne LiDAR, Inc. *HDL-32E User Manual*; Velodyne LiDAR, Inc.: San Jose, CA, USA, 2017.
56. Laser Measurements Calibrations. Available online: https://www.nist.gov/calibrations/laser-measurements-calibrations (accessed on 6 August 2018).
57. Durrant-Whyte, H.; Bailey, T. Simultaneous localization and mapping: part I. *IEEE Robot. Autom. Mag.* **2006**, *13*, 99–110. [CrossRef]
58. Mochnac, J.; Marchevsky, S.; Kocan, P. Bayesian filtering techniques: Kalman and extended Kalman filter basics. In Proceedings of the 2009 19th International Conference Radioelektronika, Bratislava, Slovakia, 7 July 2009; pp. 119–122.
59. Matsebe, O.; Namoshe, M.; Tlale, N. *Basic Extended Kalman Filter-Simultaneous Localisation and Mapping*; INTECH Open Access Publisher: London, UK, 2010.
60. Dissanayake, M.W.M.G.; Newman, P.; Clark, S.; Durrant-Whyte, H.F.; Csorba, M. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Trans. Robot. Autom.* **2001**, *17*, 229–241. [CrossRef]
61. Mahalanobis, P.C. On the Generalized Distance in Statistics. *Proc. Natl. Inst. Sci. India* **1936**, *2*, 49–55.
62. Wang, B.; Shi, W.; Miao, Z. Confidence Analysis of Standard Deviational Ellipse and Its Extension into Higher Dimensional Euclidean Space. *PLoS ONE* **2015**, *10*, e0118537. [CrossRef] [PubMed]
63. Lindner, L.; Sergiyenko, O.; Rivas-López, M.; Ivanov, M.; Rodríguez-Quiñonez, J.C.; Hernández-Balbuena, D.; Flores-Fuentes, W.; Tyrsa, V.; Muerrieta-Rico, F.N.; Mercorelli, P. Machine vision system errors for unmanned aerial vehicle navigation. In Proceedings of the 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), Edinburgh, UK, 19–21 June 2017; pp. 1615–1620. [CrossRef]
64. Garcia-Cruz, X.; Sergiyenko, O.; Tyrsa, V.; Rivas-Lopez, M.; Hernandez-Balbuena, D.; Rodriguez-Quiñonez, J.; Basaca-Preciado, L.; Mercorelli, P. Optimization of 3D laser scanning speed by use of combined variable step. *Opt. Lasers Eng.* **2014**, *54*, 141–151. [CrossRef]
65. Gupta, N.; Khosravy, M.; Gupta, S.; Dey, N.; González Crespo, R. Lightweight Artificial Intelligence Technology for Health Diagnosis of Agriculture Vehicles: Parallel Evolving Artificial Neural Networks by Genetic Algorithm. *Int. J. Parallel Program.* **2020**. [CrossRef]
66. Gupta, N.; Khosravy, M.; Patel, N.; Dey, N.; Gupta, S.; Darbari, H.; González Crespo, R. Economic data analytic AI technique on IoT edge devices for health monitoring of agriculture machines. *Appl. Intell.* **2020**. [CrossRef]