

A Novel Method for Land Vehicle Positioning: Invariant Kalman Filters and Deep-Learning-Based Radar Speed Estimation

Paulo Ricardo Marques de Araujo , Graduate Student Member, IEEE, Mohamed Elhabiby, Sidney Givigi , Senior Member, IEEE, and Aboelmagd Noureldin, Senior Member, IEEE

Abstract—Autonomous and intelligent vehicles are multi-sensor systems operating in various environments and conditions. Due to their characteristics, inertial measurement units (IMUs) are typically the core component of such systems. However, these sensors rapidly accumulate errors due to biases and noise, degrading the positioning solution. Therefore, this article presents a positioning solution that only uses three gyroscopes and one radar. The proposed method was tested using low-cost sensors in different scenarios, such as open-sky, urban and indoor areas. The key components of the method are the invariant Kalman filters and the use of deep neural networks to estimate the forward speed of the car using the radar readings. The method was tested on a custom dataset, and our integrated solution accurately estimates the vehicle’s position, velocity, and orientation. We achieved, on average, a 1.45% translational error in the tested scenarios, making the proposed method a robust alternative to current IMU-based positioning methods.

Index Terms—Autonomous vehicles, deep learning, invariant extended Kalman filter, inertial navigation.

I. INTRODUCTION

ACCURATE and reliable positioning systems are imperative for autonomous vehicles (AVs) to function safely and effectively. Positioning systems provide real-time information on the AV’s position, orientation, and velocity, which enables the vehicle controller to make informed decisions and precisely control its movements [1].

Typically, inertial measurement units (IMUs) are the core component of positioning systems [2]. IMUs generally have three orthogonal gyroscopes and three orthogonal accelerometers. The former provides angular velocity measurements, while

Manuscript received 27 April 2023; revised 19 May 2023 and 5 June 2023; accepted 15 June 2023. Date of publication 20 June 2023; date of current version 20 October 2023. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grants RGPIN-2020-03900 and ALLRP-560898-20. (Corresponding author: Paulo Ricardo Marques de Araujo.)

Paulo Ricardo Marques de Araujo and Aboelmagd Noureldin are with the Department of Electrical and Computer Engineering, Queen’s University, Kingston, ON K7L 2N8, Canada (e-mail: paulo.araujo@queensu.ca; nourelda@queensu.ca).

Mohamed Elhabiby is with the Micro Engineering Tech. Inc., Calgary, AB T2M 0L7, Canada (e-mail: elhabiby@microengineering.ca).

Sidney Givigi is with the School of Computing, Queen’s University, Kingston, ON K7L 2N8, Canada (e-mail: sidney.givigi@queensu.ca).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TIV.2023.3287790>.

Digital Object Identifier 10.1109/TIV.2023.3287790

the latter provides linear acceleration measurements. By using both information, the system can infer the position and orientation of the vehicle. Commercial IMUs can vary in quality and accuracy, which can be especially critical for low-cost sensors as they cause significant drifts in the final solution [3].

An alternative for positioning using an IMU is the Reduced Inertial Sensor System (RISS) [4]. It uses a reduced set of sensors and an odometer to estimate the state of the vehicle more accurately. Recent publications have achieved compelling results with this method [5], [6], [7], [8], [9]. However, RISS has limitations. It relies on odometers, which operate at low data rates. Therefore, RISS may not be the ideal solution for AVs. Also, some of the attitude angles are computed using accelerometers, which results in noisy and unreliable estimates.

Using a different reduced set of sensors, deep neural networks, and signal-processing tools, we propose a new method to serve as the core component for positioning systems embedded in modern vehicles and future AVs. Our contributions are as follows:

- we introduce the kinematic model and the strategy to estimate the forward speed of a car using low-cost radar, deep learning and signal-processing tools;
- we implement invariant extended Kalman filters that exploit the proposed model and combine it with relevant sources of position corrections for ground AVs (e.g. GNSS and 3D maps (Fig. 1));
- we demonstrate the performance of the proposed approach against IMU-based methods on a custom dataset that explored different urban environments.

The results shown in this article demonstrate that the proposed approach outperformed the IMU-based method in all tests. All equations of the filters used in this work are detailed, and adaptations or extensions may be made in a smooth way.

The remainder of the document is organized as follows. Section II offers a review of recent related work, giving an idea of the potential applications of the proposed method. Section III explains how an IMU-based positioning algorithm works and discusses its limitations. Next, Section IV presents the proposed method, the challenges associated with low-cost radars and forward speed estimation, and the strategy employed to tackle them. In Section V, we model invariant Kalman filters for relevant applications aiming AVs. All the experiments and achieved results are detailed in Section VI. Finally, Section VII concludes the article.

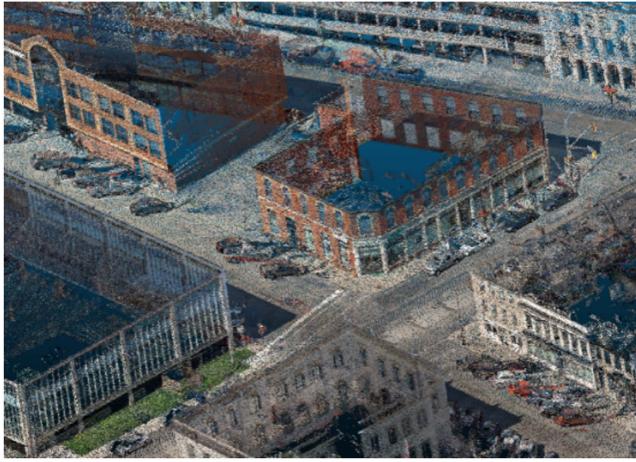


Fig. 1. 3D map example (Kingston, ON).

II. RELATED WORK

IMUs are ubiquitous in positioning systems for land vehicles. For example, they can be used with vision [10], [11], LiDAR [12], radar [13], global navigation satellite system (GNSS) [14], and 3D maps [15], [16]. As maps have become important for positioning systems in areas where GNSS fails, researchers also use IMUs to aid the 3D map creation while navigating [17]. Recently, some researchers proposed a remarkable approach for accurate positioning using only the IMU. The method leverages deep-learning tools and state-of-the-art Kalman filters [18].

Another exciting research path investigates methods to enhance the quality of IMU measurements. For example, exploring conventional signal-processing tools [19], supervised learning [20], and self-supervised learning, which diminishes the need for specialized datasets [21].

All the aforementioned techniques and sensors have associated uncertainties. Combining estimates from these diverse systems requires filtering techniques such as Kalman filters. Recently, the application of geometry theory for position estimation has been introduced [22]. The proposed framework gave rise to a class of invariant filters that rapidly became state-of-the-art for positioning problems. Invariant Kalman filters (IKFs) have been applied to different mobile platforms such as autonomous underwater vehicles (AUVs) [23], quadcopters [24], bipedal robots [25], aircrafts [26], and cars [27]. The properties of IKFs helped to solve various challenges in mobile navigation, such as Simultaneous Localization and Mapping (SLAM) [28] and visual-inertial odometry [29].

III. PROBLEM DEFINITION

As previously mentioned, IMUs are the core of positioning systems applied to autonomous vehicles due to their high operating data rates and ability to function in all weather conditions. The accelerometers and gyroscopes that compose an IMU provide noisy and biased linear accelerations, $\tilde{\mathbf{a}}_n \in \mathbb{R}^3$, and angular

velocities, $\tilde{\boldsymbol{\omega}}_n \in \mathbb{R}^3$. These sensors are usually modelled as:

$$\tilde{\mathbf{a}}_n = \mathbf{a}_n + \mathbf{b}_n^a + \mathbf{w}_n^a, \quad (1)$$

$$\tilde{\boldsymbol{\omega}}_n = \boldsymbol{\omega}_n + \mathbf{b}_n^\omega + \mathbf{w}_n^\omega, \quad (2)$$

where $\mathbf{a}_n \in \mathbb{R}^3$ is the true linear acceleration, $\boldsymbol{\omega}_n \in \mathbb{R}^3$ is the true angular velocity, $\mathbf{b}_n^a \in \mathbb{R}^3$ is the bias drift associated with the accelerometers, $\mathbf{b}_n^\omega \in \mathbb{R}^3$ is the bias drift associated with the gyroscopes, and $\mathbf{w}_n^a, \mathbf{w}_n^\omega \in \mathbb{R}^3$ are zero-mean Gaussian noises. The biases can be modelled as random walk processes:

$$\mathbf{b}_{n+1}^a = \mathbf{b}_n^a + \mathbf{w}_n^{b_a}, \quad (3)$$

$$\mathbf{b}_{n+1}^\omega = \mathbf{b}_n^\omega + \mathbf{w}_n^{b_\omega}. \quad (4)$$

The vehicle has three states: attitude ($R_n \in \text{SO}(3)$), velocity ($\mathbf{v}_n \in \mathbb{R}^3$) and position ($\mathbf{p} \in \mathbb{R}^3$). Typically, The evolution of the states is modelled as follows:

$$R_{n+1} = R_n \exp \left(\left[(\tilde{\boldsymbol{\omega}}_n - \mathbf{b}_n^\omega - \mathbf{w}_n^\omega) dt \right]_{\times} \right), \quad (5)$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + (R_n (\tilde{\mathbf{a}}_n - \mathbf{b}_n^a - \mathbf{w}_n^a) + g) dt, \quad (6)$$

$$\mathbf{p}_{n+1} = \mathbf{p}_n + \mathbf{v}_n dt, \quad (7)$$

where $g \in \mathbb{R}^3$ is the gravitational field, \exp is the $\text{SO}(3)$ exponential map, and dt is the discrete sampling time. Finally, $\left[x \right]_{\times} \in \mathbb{R}^{3 \times 3}$ denotes the skew matrix associated with the cross product with a three-dimensional vector, i.e., for $a, x \in \mathbb{R}^3$, we have $\left[x \right]_{\times} a = x \times a$. This is an IMU-based positioning system to estimate the state of the vehicle. The main challenge of such a solution is the noisy and biased measurements of the sensors. For example, one can easily see that \mathbf{p}_n is a double integration of the linear accelerations provided by the IMU, making it susceptible to rapid error propagation and substantial drift in the estimates over time.

To address these issues, the next section introduces our proposed method. It involves replacing the accelerometers of the IMU with a low-cost radar mounted on the front bumper of the vehicle. By leveraging the readings of the radar and deep learning methods, we significantly reduce the effects of noise and biases, resulting in more accurate and reliable estimates of the states of the vehicle.

IV. PROPOSED METHOD

A. Kinematic Model

The proposed method is a Radar Inertial Dead-Reckoning (RIDR) positioning system. Unlike the IMU-based method, the proposed method estimate the velocity in the IMU-frame, i , using measurements from a radar mounted on the front bumper of the car. This method alleviates the system from error propagation due to the integration of the linear accelerations as well as the presence of the gravity field.

Let us assume the vehicle is travelling following a generic path as shown in Fig. 2. The forward speed of the vehicle, v^f , is measured using the radar in the IMU frame, i . Let us further assume that the lateral and up velocities are null. Therefore, without loss of generality, the velocity vector in the i -frame is

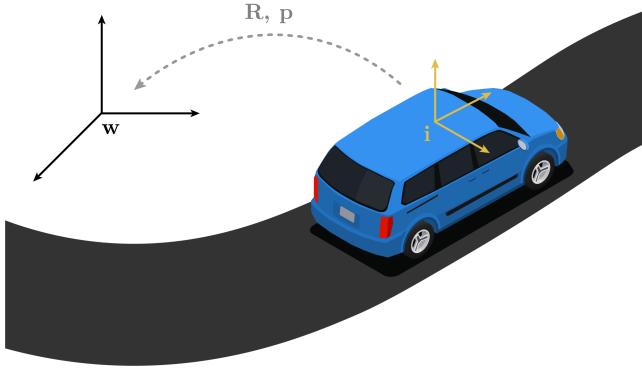


Fig. 2. Coordinate systems used in this article.



Fig. 3. Experimental setup.

defined as

$$\mathbf{v}_n^i = \begin{bmatrix} 0 & v_n^f & 0 \end{bmatrix}^\top. \quad (8)$$

The orientation, $R \in \text{SO}(3)$, and position, $\mathbf{p} \in \mathbb{R}^3$, of the vehicle are expressed in the w -frame (i.e., the world frame). Given R_0 and \mathbf{p}_0 , the proposed method estimates future states using the gyroscopes and the low-cost radar. The velocity vector in the i -frame is projected to the w -frame using R .

The radar also provides noisy measurements; therefore,

$$\tilde{\mathbf{v}}_n^i = \mathbf{v}_n^i + \mathbf{w}_n^{v^i},$$

where $\mathbf{w}_n^{v^i} \in \mathbb{R}^3$ is zero-mean Gaussian noise. Finally, the following equations govern the proposed kinematic model:

$$R_{n+1} = R_n \exp \left(\left[(\tilde{\boldsymbol{\omega}}_n - \mathbf{b}_n^\omega - \mathbf{w}_n^\omega) dt \right]_x \right) \quad (9)$$

$$\mathbf{p}_{n+1} = \mathbf{p}_n + R_n \left((\tilde{\mathbf{v}}_n^i - \mathbf{w}_n^{v^i}) dt \right), \quad (10)$$

B. Forward Speed Estimation

One of the advantages of the proposed system is the frequency of the integrated solution. Radars are often faster than odometers, e.g. the low-cost radar we use operates at 20 Hz, while the odometer available in the experimental setup operates at 3 Hz. Further technical details are presented in Section VI.

The radar is mounted facing the ground with an angle θ , as presented in Fig. 3. This configuration enables the measurement of the Doppler speed, v^D , between the car and the ground. As the ground is always stationary, the radar can estimate the forward speed of the vehicle. The mounting angle can be used to convert v^D to v^f using

$$v_n^D = v_n^f \cdot \cos \theta. \quad (11)$$

The mounting angle is unknown due to the configuration of the holding device and the shape of the bumper. Besides calibrating θ , it is crucial to consider the noise associated with the measurements. Radar data is usually very noisy due to several different reasons. One reason is that radars are susceptible to ground imperfections and changes in the geometry of the road. This generates inherently noisy measurements that may not be good for positioning solutions.

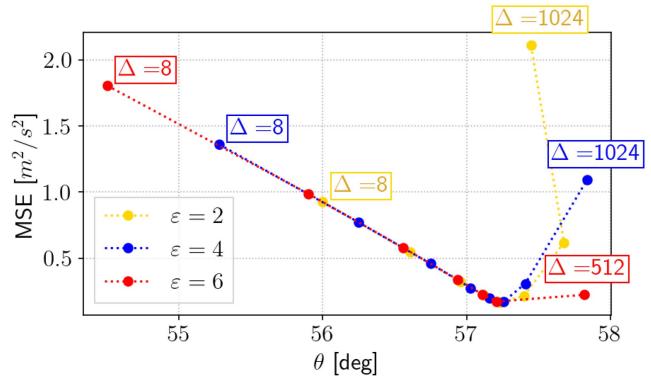


Fig. 4. Mounting angle calibration.

The mounting angle θ can be found using signal processing techniques and a calibration trajectory. The trajectory should cover an area with good road quality. Also, a reference solution is needed for the optimization routines. In this work, the reference was built using GNSS (see Section VI-A). Therefore, the calibration trajectory was recorded in an open-sky area.

Different methods can be applied, but this work uses the Savitzky-Golay filter (a.k.a. savgol) [30] to illustrate the calibration process. This is a digital signal processing technique used for smoothing and noise reduction of data. It fits a polynomial to a small window of neighbouring data points and uses it to estimate the value of the central data point. The filter is effective in preserving important features while reducing noise.

The calibration algorithm searches for the minimum mean square error between the filtered signal and the reference signal while varying θ . The designer can fix the window size, Δ , and the polynomial order, ε , during the search. The total number of parameters in the Savitzky-Golay filter can be calculated as $(\Delta + 1) \times (\varepsilon + 1)$. For example, if a window size of 5 and a polynomial order of 2 are chosen, the filter would have $(5 + 1) \times (2 + 1) = 18$ parameters. These parameters are coefficients of the polynomial function that are determined through the least-squares method to achieve the best fit for the data within the window.

Fig. 4 illustrates a study utilizing the Savitzky-Golay filter with different configurations: $\Delta = 2^n$, where $n = \{3,$

TABLE I
ESTIMATED MOUNTING ANGLES

Δ	ε	MSE [m^2/s^2]	θ [deg]	Parameters
128	2	0.168	57.239	387
256	4	0.172	57.259	1,285
256	6	0.172	57.209	1,799

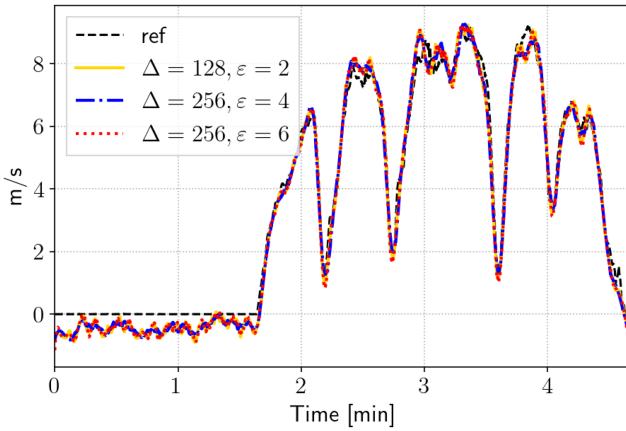


Fig. 5. Forward speed estimation using the Savitzky-Golay filter.

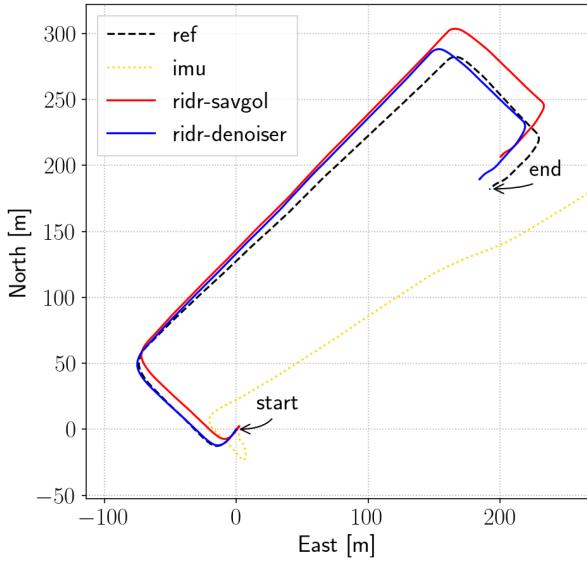


Fig. 6. Short testing trajectory.

$4, \dots, 9, 10\}$, and $\varepsilon = \{2, 4, 6\}$. The three groups exhibit similar behaviour, with the mean squared error (MSE) decreasing as Δ increases until reaching a minimum. Beyond these points, the filter performance diminishes. Table I displays the estimated mounting angles, which are consistently around 57.2 degrees across the three groups. In terms of performance, the first group achieved comparable results with fewer parameters.

The performance of the filters is depicted in Fig. 5. As anticipated, the recovered signals exhibit close proximity to each other. However, there are instances where the filters struggle, such as during stationary periods, specifically from the beginning of the trajectory until approximately 1.7 min later, where

the filters fail to identify zero forward speed. While alternative techniques like clipping could potentially enhance the results, they require empirical adjustments and are time-consuming. Moreover, low-cost radar measurements are prone to noise that can be influenced by road quality and vehicle vibrations. These factors collectively render the signal-processing approach sub-optimal for this particular application.

To tackle all these challenges, the proposed approach leverages deep learning combined with signal-processing methods to better estimate the forward speed of the car. A deep-learning approach may be more efficient than traditional methods for denoising radar data due to uncertainties in the model. For example, the characteristics of the noise may not be known or may change over time due to road and/or environmental conditions. Deep learning has been shown to be effective to denoise images [31] and sound signals [32], and similar approaches may work well for radar data.

Fig. 7 depicts the denoiser used within the proposed framework. The core component of the denoiser is the autoencoder used to both inherently estimate θ and reduce the noise of a sequence of radar readings. Note that the length of the readings can vary according to any project requirement. In this work, a sequence of 128 readings is used to balance performance and initialization time for real-time systems (a radar operating at 20 Hz takes approximately 6.4 s to collect the required data to be sent to the denoiser).

Autoencoders are neural networks that can learn to remove noise from corrupted data [33]. They consist of an encoder and a decoder. The encoder transforms the noisy input into a lower-dimensional representation called the latent space, capturing important data features while removing noise. The decoder then maps the encoded representation back to the original input space. During training, the autoencoder is provided with pairs of original and noisy data and is optimized to minimize the difference between the decoded output and the original input. By iteratively adjusting the encoder and decoder parameters, the autoencoder improves its ability to remove noise and reconstruct the original input. Additionally, as a byproduct of the reconstruction process, the autoencoder automatically recovers the mounting angle, eliminating the need for extrinsic calibration of the radar mounting device.

To guarantee that the denoiser is effective, the architecture comprises a sequence of 6 one-dimensional convolutions with kernels of size 7 each for the encoder as well as the decoder. On the encoder side, the number of channels increases by 2^n , where n is the order of the layer starting at layer 2. As previously stated, the decoder mirrors the encoder, and the number of channels decreases until a single channel is achieved in the last layer. In total, the model has approximately 27,600 parameters, and its estimated size is 0.9 MB, a relatively small neural network. Each convolution layer is followed by a batch normalization layer [34] and a GELU activation layer [35]. At the end of the neural network, a median filter with a sliding window of size 3 is used to remove possible outliers.

To showcase the superior performance of the proposed model, Fig. 6 illustrates a short testing trajectory covering a distance of 0.6 km over a duration of 2.5 min. The kinematic model, as

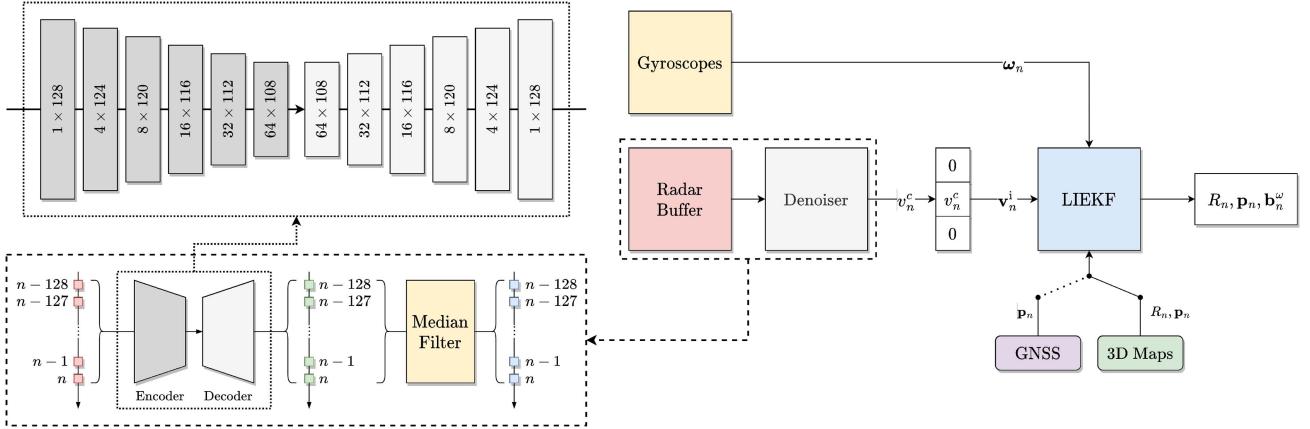


Fig. 7. Proposed integrated method architecture.

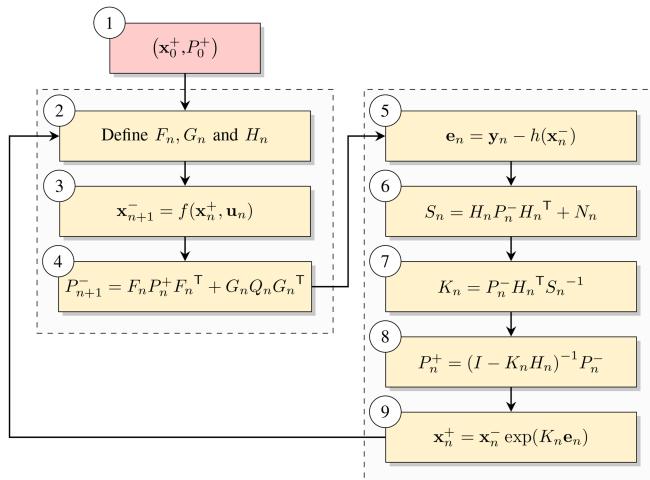


Fig. 8. Invariant Extended Kalman filter workflow.

explained in Section IV-A, was tested using both the Savitzky-Golay filter and the proposed denoiser.

Upon observation, both variations of the proposed model exhibit closer proximity to the reference solution, while the IMU-based solution loses accuracy after a few seconds. The introduced denoiser significantly enhances the positioning solution in this dead-reckoning example. The proposed model achieved an absolute trajectory error of 3.71% of the travelled distance, whereas the model utilizing the Savitzky-Golay filter achieved a performance of 4.02%. The following sections describe the utilization of invariant Kalman filters for robust positioning with GNSS and 3D maps, elucidate the training procedure of the denoiser, and present additional tests conducted with various trajectories.

V. INVARIANT EXTENDED KALMAN FILTER

The Invariant Extended Kalman filter (IEKF) has shown remarkable results for inertial navigation in recent years [18], [23], [25], [29], [36], [37], motivating the development of invariant filters for RIDR. In the IEKF, the state space is assumed to

be a matrix Lie group, where symmetry in the state space is exploited to derive state-invariant Jacobians [22], [38]. The IEKF workflow is summarized in Fig. 8.

The first step is to initialize the state variable \mathbf{x} and the covariance matrix P . Next, the algorithm enters a loop where a few equations are divided into two groups: prediction (left box) and correction (right box). In this work, $(\cdot)^-$ and $(\cdot)^+$ indicate the a priori and a posteriori states or covariances, i.e., before and after correcting the predictions at timestamp n .

Initially, the Jacobians matrices F_n , G_n , and H_n are updated accordingly. Then, the nonlinear function $f(\mathbf{x}_n, \mathbf{u}_n)$ is used to predict the states using the inputs, \mathbf{u}_n . The covariance is estimated in step 4, where Q_n is the process covariance matrix of the system. Exteroceptive and proprioceptive sensors can be used to measure part of or all of the states.

If measurements are available, the algorithm enters the correction stage. Firstly, the error e_n is computed using the measurement (\mathbf{y}_n) and the measurement model $h(\mathbf{x}_n)$. Steps 6 and 7 are used to compute the Kalman gain, where N_n is the measurement covariance matrix that can be defined following the sensors' datasheets, user experience, and/or dedicated experiments. Next, the covariance and the states are updated in steps 8 and 9, respectively. The last step presents the equation to correct the states using the left-invariant formulation.

The reader can refer to Appendix A for some background on Lie theory and to Appendix B for the equations used in the tested filters. Finally, to implement an IEKF, one needs to define the functions $f(\cdot)$ and $h(\cdot)$, the matrices F_n , G_n and H_n , and the covariance matrices Q_n and N_n .

A. Defining the Dynamical Model $f(\cdot)$

RIDR is flexible and easy to use within the IEKF workflow. Equations (2)–(4), (10) and (9) are used to propagate the states, \mathbf{x}_n . The Jacobian matrices help in approximating the nonlinear function $f(\cdot)$ as a linear function by providing local linearizations around the current estimate. Appendix B details the Jacobian matrices F_n and G_n used in the propagation of the covariance in the proposed filters.

B. Defining the Measurement Model $h(\cdot)$

Considering the arrangement presented in Fig. 2, a wide variety of measurement models can be admitted for Kalman filters employing RIDR, such as:

- 1) Position in the w -frame using GNSS: \mathbf{p}_n
- 2) Position in the w -frame using 3D maps: \mathbf{p}_n
- 3) Position in the i -frame using known landmarks \mathbf{r}_n^m : $R_n^T(\mathbf{r}_n^m - \mathbf{p}_n)$
- 4) Attitude in the i -frame using magnetometer: $R_n^T\beta$
- 5) Attitude in the w -frame using 3D maps: R_n

This work exploits two sources of information of utmost importance for autonomous vehicles: GNSS and 3D maps. The GNSS receiver measures the position, and the 3D maps are used with the LiDAR assembled in the car. The LiDAR generates a local point cloud that is compared to the city map using the iterative closest point (ICP) algorithm to measure the position and the orientation of the vehicle [15]. Appendix B presents the Jacobian matrix H_n developed for each case.

VI. EXPERIMENTAL RESULTS

A. Experimental Setup

The proposed method was evaluated using a dataset developed with the experimental setup available at the Navigation and Instrumentation laboratory of Queen's University (Fig. 3). The experimental car has various sensors, such as stereo cameras, GNSS receivers, LiDAR, and automotive radars.

For the filters we present in this work, we use the IMU inside the Stereolabs Zed2i camera (mounted on the roof of the car) and an OmniPresence OPS241-A radar. The low-cost radar operates at 20 Hz with a resolution of 0.3 m/s, and the low-cost IMU operates at 100 Hz, but it is dynamically downsampled to match the frequency of the radar. Experiments conducted in the laboratory reviewed a bias instability around 18 deg/h for the gyroscopes and 14 mg for the accelerometers. Finally, the reference solution was built using a Novatel PwrPak7 unit connected to a KVH 1750 IMU. The data was processed using the Novatel Inertial Explorer software along with base station data to generate Real-Time Kinematics positioning (RTK).

B. Radar Denoiser Training

The model was implemented using Pytorch and trained using the aforementioned dataset. It contemplates various trajectories recorded in Canadian cities such as Kingston, Toronto, Calgary, and Edmonton. Around 75 km of data was collected, generating more than 160,000 radar measurements. The data was then divided into small sequences of 128 readings.

The training spanned 400 epochs, using the Rectified ADAM as the optimizer ($\lambda = 10^{-3}$, $\delta = 10^{-5}$). The loss function was defined as

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (12)$$

where N is the total number of observations, y_i is the actual value of the i^{th} observation, and \hat{y}_i is the predicted value of the i^{th} observation. A scheduler was employed to reduce λ if the

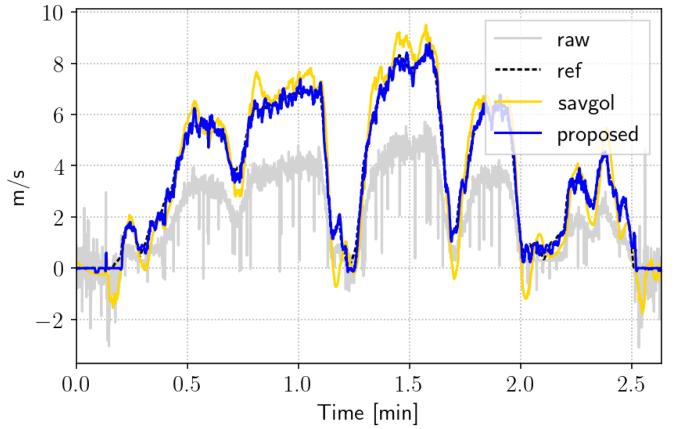


Fig. 9. Forward velocity results for the short testing trajectory.

validation loss does not improve after 100 epochs. The batch size was defined as 256.

The reference input of the encoder-decoder model was generated using the reference solution. After processing the GNSS and the IMU data inside Inertial Explorer, the standard deviations of the velocity were in the order of 10^{-3} m/s. The reference forward speed was computed using the reference velocity components as follows:

$$v_n^{\text{fref}} = \sqrt{v_n^x + v_n^y + v_n^z} \quad (13)$$

Fig. 9 presents the performance of the proposed model compared to the Savitzky-Golay filter with $\Delta = 128$, $\varepsilon = 2$ and $\theta = 57.239$ deg. It can be seen that the proposed method is closer to the reference. In this example, the denoiser obtained a mean square error of $0.056 \text{ m}^2/\text{s}^2$, while the Savitzky-Golay filter achieved $0.526 \text{ m}^2/\text{s}^2$. The proposed denoiser operates at high speed, with an approximate inference time of 1 ms. Furthermore, the acquired neural network (Fig. 7) can be analyzed as a filter. For example, the output of the encoder can be thought of as a projection onto a different space. A specialist can try to analyze these results in order to gain insight into the neural network parameters. Although more work is necessary to improve the interpretability of the architecture, this section also presents ablation studies to justify the proposed architecture.

Four different versions of denoisers are tested: (a) autoencoder only, (b) median filter only, (c) median filter before the autoencoder, and (d) the proposed architecture. For option (3), the filtered signal was scaled using $\theta = 57.239$ deg. Fig. 10 presents the results of the denoisers using data from the testing trajectory. In terms of mean square error, (a) achieved $0.059 \text{ m}^2/\text{s}^2$, (b) achieved $0.937 \text{ m}^2/\text{s}^2$, (c) achieved $0.065 \text{ m}^2/\text{s}^2$, and (d) achieved $0.056 \text{ m}^2/\text{s}^2$. The proposed method, i.e., (d), outperformed all other tested methods. While the difference between (a) and (d) is small, the addition of the median filter helps in smoothing noises that were not completely addressed by the autoencoder, particularly during stationary periods when the radar tends to produce erratic measurements.

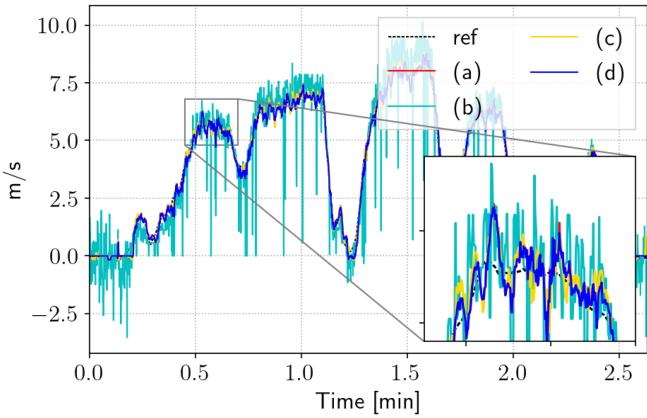


Fig. 10. Results of the four denoisers in the short testing trajectory.

C. Evaluation Metrics and Compared Methods

To evaluate the performance of RIDR with and without the IEKF, this work uses four metrics:

- 1) Relative Translation Error (t_{rel}): it is determined by calculating the average relative translation increment error for every possible sub-sequence of distances ranging from 100 m to 800 m and expressing this error as a percentage of the total distance travelled [39], [40];
- 2) Relative Rotational Error (r_{rel}): the average relative rotational increment error is calculated for all possible subsequences of distances ranging from 100 m to 800 m. This error is expressed in deg/m [39], [40];
- 3) Absolute Trajectory Error (ATE): it measures the root-mean-square error between the predicted 3D trajectory and the ground truth in m ;
- 4) Cumulative Distribution Function of Trajectory Error (CDF-TE): it is a statistical measure that describes the distribution of errors between the predicted 3D trajectory and the ground truth. We evaluate the errors at three percentiles: 1σ , 2σ and 3σ . The errors are expressed in m .

RIDR works as an alternative to the IMU-based method. Therefore, its performance is compared to the IMU-based method with and without the IEKF. The reader can refer to Appendix B for further details about IEKF implementation for RIDR and IMU-based methods. The tests were performed in real environments where GNSS and 3D maps were available. Some challenges, such as GNSS outages and 3D map gaps, are present in the presented trajectories.

D. Implementation Details

The IMU was tested indoors for bias instability and noise density specification. Four hours of data were collected and analyzed using Allan Variance tools. The retrieved information was used to build the process covariance matrices. In this work, Q_n is kept constant throughout the trajectories, and dynamic tuning is left for future work. However, the measurement covariance matrices are set dynamically. The GNSS receiver provides RTK measurements with their associated standard deviations. This information is used to build the N_n whenever the GNSS readings

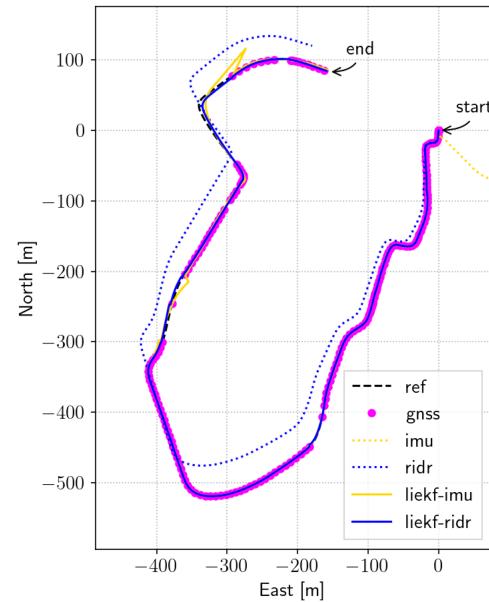


Fig. 11. Trajectory results in an open-sky area (seq. 01) of the Navinst dataset.

are used. On the other hand, ICP outputs similar information for the registration solution, which is used to build N_n whenever the 3D maps are used. GNSS and 3D maps measurements were given with respect to the w -frame.

E. Trajectory Results

All trajectories presented in this section followed the same protocol. The parameters were set as described in the previous section, and the trajectories started in open-sky areas to allow the initialization of the filters using GNSS. Also, predictions happen when the vehicle is moving.

This work explored four different real urban scenarios:

- *open-sky*: most of the trajectory happens in an open area with good GNSS coverage;
- *mixed*: the trajectory explores areas with good GNSS and 3D map coverage but faces challenges such as outages and gaps;
- *dense*: most of the trajectory happens in dense downtown areas with long periods of GNSS outages, but 3D maps are available;
- *indoor*: the trajectory starts outside before entering an indoor parking garage where 3D maps are available.

Fig. 11 shows the robustness of RIDR standalone and integrated with the left-invariant extended Kalman filter (LIEKF) in an open-sky scenario. As discussed in Section IV-A, RIDR does not use accelerometers as they cause significant drifts (e.g., IMU and LIEKF-IMU face this problem). As can be seen in Fig. 11, the IMU-based solution diverges from the beginning. RIDR accumulates errors, but the solution is bounded and closer to the real trajectory. As for the IEKF solutions, LIEKF-RIDR is more stable and robust, as it is more resilient to drifts in the sensors.

The second sequence explored a mixed area. In this trajectory, GNSS and 3D maps are available, but not continuously

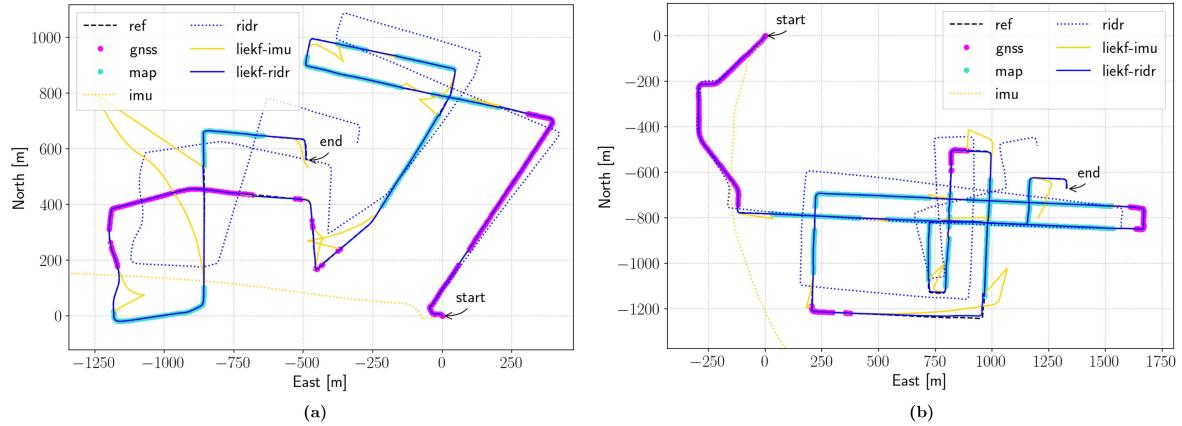


Fig. 12. Trajectory results in a mixed (seq. 02) and in a dense area (seq. 03) of the Navinst dataset.

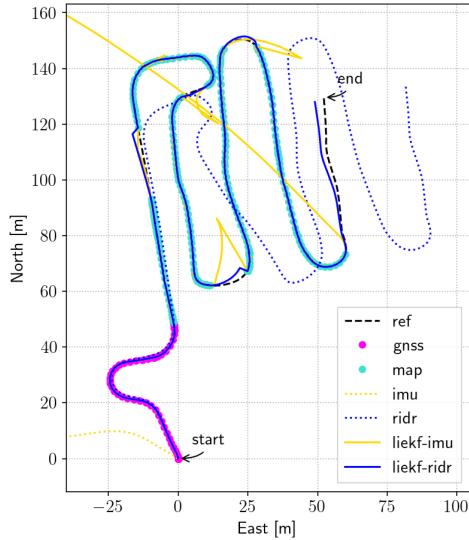


Fig. 13. Trajectory results in an indoor parking garage (seq. 04) of the Navinst dataset.

(Fig. 12(a)). Similarly, it can be seen that RIDR outperforms the IMU-based solution. This is a longer trajectory, and the biases of the sensors start to become more critical close to the end. Again, the IMU-based solution diverges from the start, while the RIDR can keep a bounded error with the trajectory. When evaluating the IEKF solutions, it is clear that LIEKF-RIDR showcases superior stability and robustness, especially around corners.

The third sequence was recorded in the dense area of downtown Calgary, where RTK GNSS measurements do not exist. Portions of the central area have 3D maps that are used for position and orientation corrections. The overall results obtained in this sequence are presented in Fig. 12(b). The same behaviour observed in the previous trajectories is noticed in this trajectory: the IMU-based solution diverges; RIDR keeps a bounded error; and, again, LIEKF-RIDR transits better between GNSS and 3D maps, while the LIEKF-IMU drifts quickly.

In Fig. 13, the last sequence explores an indoor parking garage. The limited time spent outside affects the initialization of the filters and biases estimation. This is particularly noticeable

in the RIDR and IMU-based solutions, with the former accumulating more errors and the latter diverging from the start. When the vehicle enters the garage ramp, GNSS measurements are no longer available, and only 3D maps are used for position and orientation corrections. The garage contains map gaps, as indicated by the cyan readings in Fig. 13, similar to other scenarios in this work. The speed profile in this sequence differs from others, with the vehicle moving slowly and intermittently, posing an additional challenge for the low-cost radar used in the experiment. Despite these challenges, the integrated solution with RIDR outperforms LIEKF-IMU.

Table II provides additional information on the trajectories and their corresponding results. The averaged trajectory results were calculated for easier analysis, revealing the superior performance of the proposed integrated method. Specifically, the LIEKF-RIDR achieved an impressive averaged relative translation error of 1.45% over a total distance of 16.48 km across all trajectories. This achievement is particularly noteworthy considering the challenges and the limitations of the employed sensors. The proposed method consistently maintained a closer alignment with the reference trajectory, resulting in more stable evaluation metrics. The averaged absolute trajectory error remained below 2 m throughout the experiments.

Table III illustrates the CDF-TE for all sequences. The results clearly demonstrate that the RIDR-based algorithms consistently outperformed their counterparts. For instance, at the 2σ percentile, the LIEKF-RIDR algorithm achieved an average error of 4.41 m, while the LIEKF-IMU algorithm exhibited an error of 43.75 m. Regarding standalone solutions, RIDR demonstrated superior performance with an error of 176.80 m, compared to the IMU-based solution with an error of 135.45×10^3 m. These findings highlight the effectiveness of the proposed method, which achieved the highest scores among the tested algorithms in this study.

F. Discussion

As presented in previous sections, the absence of accelerometers increases the robustness of RIDR. The integration of RIDR and the LIEKF proved to be a powerful tool for land vehicle navigation. This combination could bridge different challenges

TABLE II
EXPERIMENTAL RESULTS USING THE NAVINST DATASET

Sequence	Length (km)	Duration (min)	Environment	LIEKF-RIDR			LIEKF-IMU			RIDR			IMU		
				t_{rel} (%)	r_{rel} (deg/m)	ATE (m)	t_{rel} (%)	r_{rel} (deg/m)	ATE (m)	t_{rel} (%)	r_{rel} (deg/m)	ATE (m)	t_{rel} (%)	r_{rel} (deg/m)	ATE (m)
01	1.56	5.32	open-sky	1.54	0.0071	2.86	3.82	0.0113	12.02	5.28	0.0109	21.22	734.57	0.0109	6.45×10^3
02	6.05	16.84	mixed	2.53	0.0082	1.56	13.38	0.0098	79.66	12.05	0.0118	167.93	7.18×10^3	0.0118	246.42×10^3
03	8.29	31.16	dense	0.98	0.0035	2.16	4.34	0.0043	25.12	10.35	0.0084	113.92	2.19×10^3	0.0084	66.42×10^3
04	0.58	8.70	indoor	0.75	0.0081	1.12	6.42	0.0083	30.64	11.93	0.0625	26.30	2.19×10^3	0.0625	5.54×10^3
Average Scores				1.45	0.0067	1.92	6.99	0.0084	36.75	9.90	0.0234	82.34	3.07×10^3	0.0234	81.21×10^3

TABLE III
CUMULATIVE DISTRIBUTION FUNCTION OF TRAJECTORY ERRORS

Sequence	LIEKF-RIDR			LIEKF-IMU			RIDR			IMU		
	1 σ	2 σ	3 σ	1 σ	2 σ	3 σ	1 σ	2 σ	3 σ	1 σ	2 σ	3 σ
1	0.52	5.09	6.69	0.63	10.20	43.15	38.11	52.39	55.40	3.74×10^3	9.82×10^3	11.16×10^3
2	0.45	6.18	9.99	1.23	97.38	438.47	213.67	428.14	433.65	147.23×10^3	401.12×10^3	455.06×10^3
3	0.33	4.16	8.72	0.43	50.83	156.40	102.29	191.55	231.21	57.66×10^3	121.32×10^3	138.93×10^3
4	0.20	2.23	3.68	0.26	16.59	198.73	28.21	35.12	37.74	4.00×10^3	9.55×10^3	10.42×10^3
Average Scores	0.37	4.41	7.27	0.64	43.75	209.19	95.57	176.80	189.50	53.16×10^3	135.45×10^3	153.89×10^3

Note: values expressed in m .

in real urban environments. The key advantage of LIEKF for this kind of problem is the generation of state-invariant Jacobians. For example, by evaluating (34) and (48), it is possible to notice the absence of previous states in the linear models. This is a remarkable aspect of invariant filters, which certainly helps in the overall performance of the integrated solution, including the biases estimation.

We note that RIDR can be slower when compared to IMU-based algorithms, i.e., the data rate of the radar regulates the data rate of the solution, which is usually lower than the data rate of IMUs. For example, consider a vehicle travelling in a city at $50\text{ km/h} \approx 13.89\text{ m/s}$. Let us assume that RIDR is operating at 20 Hz, i.e., a new state estimate every 0.05 s, and the IMU is operating at 100 Hz, i.e., a new state estimate every 0.01 s. In this case, the car travelled 0.69 m without any prediction from RIDR, while the IMU would decrease this distance to 0.14 m. However, in terms of robustness, RIDR compensates for the speed sacrifice. For example, in this work, the IMU is 5 \times faster than RIDR. When evaluating the average CDF-TE at 3 σ , RIDR is 812 \times better than the IMU-based solution. Similarly, LIEKF-RIDR is 28 \times better than the LIEKF-IMU. This superiority is observed in the three percentiles investigated in this work, as presented in Table III.

With the rapid evolution of automotive radars, the data rate limitation is expected to be resolved in the near future. For instance, if radars or other forward speed sensors provide measurements at a frequency of 50 Hz, the travelled distance is significantly reduced to just 0.28 m. This advancement makes the RIDR system an even more valuable asset, offering enhanced safety benefits for autonomous vehicles.

RIDR, as the core component of a positioning system, can also be fused with other sensors. This work exploited and detailed filters for GNSS and 3D map measurements. However, it is perfectly possible to extend the proposed method to use cameras, magnetometers, electronic scan radars, and more. In fact, this strategy makes the system robust to various real circumstances.

For example, if the LiDAR fails, electronic scan radars can temporarily correct the integrated solution. Similarly, cameras can also be used to generate point clouds of the environment and estimate the change in the attitude of the vehicle. The robustness of RIDR translates into safety as it keeps the trajectory closer to the desired path while other sensors fail to provide reliable corrections.

Another important aspect to note is the presence of a deep learning module in the RIDR workflow. This technology may be hard to certify for commercial or industrial applications [41]. However, it can offer a baseline for developing more advanced or combined signal-processing techniques.

VII. CONCLUSION

The proposed method performed better than IMU-based methods in all trajectories, including the results using GNSS and 3D map measurements. The scenarios presented in this work represent real and critical situations that autonomous vehicles might face in urban environments. This work can be adapted to other applications to replace complete IMU-based methods. For example, based on the results presented in the previous section, RIDR can be used in surveying vehicles that map cities. GNSS outages have less effect on RIDR's positioning solution, potentially improving the accuracy of the map. In future work, we aim to address the biases issue using other deep learning methods to improve the integrated solution further and extend the filter to utilize camera point clouds and visual odometry to bridge the gaps when neither of the presented sources of correction will be available.

APPENDIX A

A matrix Lie group \mathcal{G} is a set of $n \times n$ matrices, verifying the following properties:

$$I \in \mathcal{G}, \quad (14)$$

$$\mathcal{X}^{-1} \in \mathcal{G} \quad \forall \mathcal{X} \in \mathcal{G}, \quad (15)$$

$$\mathcal{X} \cdot \mathcal{Y} \in \mathcal{G} \quad \forall \mathcal{X}, \mathcal{Y} \in \mathcal{G}. \quad (16)$$

\mathcal{G} is generally a curved space, and for a point, $\mathcal{X} \in \mathcal{G}$, $T_{\mathcal{X}}\mathcal{G}$ is denoted as the tangent space at X with dimension d equals to the degrees of freedom of \mathcal{G} . The tangent space at the identity element of the group is called Lie algebra, and it is denoted as \mathfrak{g} . It consists of the tangent vectors to smooth paths in \mathcal{G} where they pass through I . A system evolving through time can be seen as a point moving on the curved space of \mathcal{G} .

The general form of the tangent vectors living in this space can be found by differentiating the defining equation of the group. The tangent space can be locally identified with a Euclidean vector space in \mathbb{R}^d . Thus, for $\xi \in \mathbb{R}^d$, $\xi^\wedge \in \mathfrak{g}$ is its corresponding element of \mathfrak{g} . Therefore, for a Lie group \mathcal{G} and its associate Lie Algebra \mathfrak{g} , the following linear mapping is defined:

$$(\cdot)^\wedge : \mathbb{R}^{dim\mathfrak{g}} \mapsto \mathfrak{g}. \quad (17)$$

This linear map takes elements of the tangent space of \mathcal{G} at the identity to the corresponding matrix representation. The opposite mapping is defined as

$$(\cdot)^\vee : \mathfrak{g} \mapsto \mathbb{R}^{dim\mathfrak{g}}. \quad (18)$$

The matrix Lie group $SO(3)$ is the group of 3×3 rotation matrices that preserve orientation. Its Lie algebra is the space of skew-symmetry matrices:

$$SO(3) = \{R, RR^T = I_{3 \times 3}, \det(R) = 1\}, \quad (19)$$

$$\mathfrak{so}(3) = \{A, A = -A^T\}, \quad (20)$$

$$\xi^\wedge = [\xi]_\times = \begin{bmatrix} 0 & -\xi_3 & \xi_2 \\ \xi_3 & 0 & -\xi_1 \\ -\xi_2 & \xi_1 & 0 \end{bmatrix}. \quad (21)$$

The closed-form expression for the exponential map is given by

$$\begin{aligned} \exp(\xi) &= I_{3 \times 3} + \frac{\sin(\|\xi\|)}{\|\xi\|} [\xi]_\times \\ &\quad + 2 \frac{\sin(\|\xi\|/2)^2}{\|\xi\|^2} [\xi]_\times^2. \end{aligned} \quad (22)$$

The logarithmic map is computed using

$$\|\mathbf{t}\| = \cos^{-1} \left(\frac{\text{Tr}(R)-1}{2} \right), \quad (23)$$

$$\xi = \frac{1}{2 \sin(\|\mathbf{t}\|)} \begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix}. \quad (24)$$

The matrix Lie group $SE(3)$ is the group of rigid motions defined as

$$SE(3) = \left\{ \begin{bmatrix} R & \mathbf{p} \\ 0_{1 \times 3} & 1 \end{bmatrix}, R \in SO(3), \mathbf{p} \in \mathbb{R}^3 \right\}, \quad (25)$$

$$\mathfrak{se}(3) = \left\{ \begin{bmatrix} [\xi^R]_\times & \xi^p \\ 0_{1 \times 3} & 0 \end{bmatrix}, \xi^R, \xi^p \in \mathbb{R}^3 \right\}, \quad (26)$$

$$\begin{pmatrix} \xi^R \\ \xi^p \end{pmatrix}^\wedge = S = \begin{bmatrix} [\xi^R]_\times & \xi^p \\ 0_{1 \times 3} & 0 \end{bmatrix}. \quad (27)$$

Its exponential map is defined as

$$\begin{aligned} \exp \begin{pmatrix} \xi^R \\ \xi^p \end{pmatrix} &= I_{4 \times 4} + S + \frac{1 - \cos(\|\xi^R\|)}{\|\xi^R\|^2} S^2 \\ &\quad + \frac{\|\xi^R\| - \sin(\|\xi^R\|)}{\|\xi^R\|^3} S^3. \end{aligned} \quad (28)$$

Finally, the matrix Lie group $SE_2(3)$ is an extension of $SE(3)$. It was named as the *group of double direct spatial isometries* [22],

$$SE_2(3) = \left\{ \begin{bmatrix} R & \mathbf{v} & \mathbf{p} \\ 0_{1 \times 3} & 1 & 0 \\ 0_{1 \times 3} & 0 & 1 \end{bmatrix}, R \in SO(3), \mathbf{v}, \mathbf{p} \in \mathbb{R}^3 \right\}, \quad (29)$$

$$\mathfrak{se}_2(3) = \left\{ \begin{bmatrix} [\xi^R]_\times & \xi^v & \xi^p \\ 0_{1 \times 3} & 0 & 0 \\ 0_{1 \times 3} & 0 & 0 \end{bmatrix}, \xi^R, \xi^v, \xi^p \in \mathbb{R}^3 \right\}, \quad (30)$$

$$\begin{pmatrix} \xi^R \\ \xi^v \\ \xi^p \end{pmatrix}^\wedge = S = \begin{bmatrix} [\xi^R]_\times & \xi^v & \xi^p \\ 0_{1 \times 3} & 0 & 0 \\ 0_{1 \times 3} & 0 & 0 \end{bmatrix}, \quad (31)$$

$$\begin{aligned} \exp \begin{pmatrix} \xi^R \\ \xi^v \\ \xi^p \end{pmatrix} &= I_{5 \times 5} + S + \frac{1 - \cos(\|\xi^R\|)}{\|\xi^R\|^2} S^2 \\ &\quad + \frac{\|\xi^R\| - \sin(\|\xi^R\|)}{\|\xi^R\|^3} S^3. \end{aligned} \quad (32)$$

APPENDIX B

The Invariant Extended Kalman Filter is an error state filter. The errors are linearized in the Lie algebra of the group used to represent the state of the system. It can have two variations named left- and right-invariant [22], [38]. This work exploits measurements in the w -frame, leading to the development of left-invariant filters. This appendix presents the left-invariant filters tested in this work.

A. Radar Inertial Dead-Reckoning System

The state of the system, i.e. $\mathbf{x}_n = (R_n, \mathbf{p}_n)$, is embedded in the matrix Lie group $SE(3)$. The gyroscope bias, \mathbf{b}_n^ω , is treated as a vector. The linearized error is defined as:

$$\mathbf{e}_n = [\xi_n^{R^T} \quad \xi_n^{p^T} \quad \mathbf{b}_n^{\omega T}]^\top \sim \mathcal{N}(0, P_n) \quad (33)$$

The states are propagated using the equations presented in Section V-A. The G_n matrix is a 9×9 identity matrix, and the F_n is defined as follows:

$$F_n = I_{9 \times 9}$$

$$\begin{aligned} &+ \begin{bmatrix} -[\tilde{\omega}_n - \mathbf{b}_n^\omega]_\times & 0_{3 \times 3} & -I_{3 \times 3} \\ -[\tilde{\mathbf{v}}_n^i]_\times & -[\tilde{\omega}_n - \mathbf{b}_n^\omega]_\times & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} dt \end{aligned} \quad (34)$$

The correction step exploits two measurement models. For position correction using GNSS readings, the model becomes:

$$\mathbf{y}_n = [\mathbf{p}_n \quad 1]^\top \quad (35)$$

leading to the Jacobian matrix H_n equals to:

$$H_n = [0_{3 \times 3} \quad I_{3 \times 3} \quad 0_{3 \times 3}] \quad (36)$$

The innovation is computed as follows:

$$\mathbf{z}_n = \mathbf{x}_n^{-1} \cdot \mathbf{y}_n - \mathbf{b}, \quad \mathbf{b} = [0 \quad 0 \quad 0 \quad 1]^T \quad (37)$$

In the case of position and attitude correction using 3D maps, the model becomes:

$$Y_n^R = R_n \quad (38)$$

$$\mathbf{y}_n^p = [\mathbf{p}_n \quad 1]^T \quad (39)$$

The innovation involves mapping the rotation matrix to the Lie algebra. The individual results are stacked into a final innovation vector to be used with the Kalman gain.

$$\mathbf{z}_n^R = \left(\log(\hat{R}_n^T \cdot Y_n^R) \right)^V \quad (40)$$

$$\mathbf{z}_n^p = \mathbf{x}_n^{-1} \cdot \mathbf{y}_n - \mathbf{b}, \quad \mathbf{b} = [0 \quad 0 \quad 0 \quad 1]^T \quad (41)$$

$$\mathbf{z}_n = \begin{bmatrix} \mathbf{z}_n^{R^T} & \mathbf{z}_n^{p^T} \end{bmatrix}^T \quad (42)$$

where \hat{R}_n is the rotation matrix extracted from the state matrix, and \log is the $\text{SO}(3)$ logarithmic map. Finally, the Jacobian matrix H_n becomes:

$$H_n = [I_{6 \times 6} \quad 0_{6 \times 3}] \quad (43)$$

The error is then computed and injected in the states as follows:

$$\mathbf{e}_n = K_n \mathbf{z}_n \quad (44)$$

$$\mathcal{X}_n^+ = \mathcal{X}_n^- \cdot \exp(\mathbf{e}_n^x) \quad (45)$$

$$\mathbf{b}_n^{\omega+} = \mathbf{b}_n^{\omega-} + \mathbf{e}_n^b \quad (46)$$

where \exp is the $\text{SE}(3)$ exponential map, and $\mathbf{e}_n^x \in \mathbb{R}^6$ and $\mathbf{e}_n^b \in \mathbb{R}^3$ are sub-vectors of \mathbf{e}_n regarding the matrix Lie group and the bias vector, respectively.

B. Inertial Measurement Unit System

The state of the system is embedded in the matrix Lie group $\text{SE}_2(3)$. The biases can be stacked to form a vector, i.e. $\mathbf{b}_n = [\mathbf{b}_n^{\omega^T} \quad \mathbf{b}_n^a]^T \in \mathbb{R}^6$. The linearized error becomes:

$$\mathbf{e}_n = \begin{bmatrix} \boldsymbol{\xi}_n^{R^T} & \boldsymbol{\xi}_n^v^T & \boldsymbol{\xi}_n^p^T & \boldsymbol{\xi}_n^{b^{\omega T}} & \boldsymbol{\xi}_n^{b^a T} \end{bmatrix}^T \sim \mathcal{N}(0, P_n) \quad (47)$$

The G_n matrix is a 15×15 identity matrix, and F_n is defined as follows:

$$F_n = I_{15 \times 15}$$

$$+ \begin{bmatrix} -[\bar{\omega}]_\times & 0_{3 \times 3} & 0_{3 \times 3} & -I_{3 \times 3} & 0_{3 \times 3} \\ -[\bar{\mathbf{a}}]_\times & -[\bar{\omega}]_\times & 0_{3 \times 3} & 0_{3 \times 3} & -I_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} & -[\bar{\omega}]_\times & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{6 \times 3} & 0_{6 \times 3} & 0_{6 \times 3} & 0_{6 \times 3} & 0_{6 \times 3} \end{bmatrix} dt \quad (48)$$

where $\bar{\omega} = \tilde{\omega}_n - \mathbf{b}_n^\omega$ and $\bar{\mathbf{a}} = \tilde{\mathbf{a}}_n - \mathbf{b}_n^a$. The measurement models are similar to those presented in the previous section.

However, due to the different state space, the Jacobian matrix H_n for position corrections using GNSS readings becomes:

$$H_n = [0_{3 \times 6} \quad I_{3 \times 3} \quad 0_{3 \times 6}] \quad (49)$$

In the case of position and attitude correction using 3D maps, the Jacobian matrix is defined as:

$$H_n = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 6} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 6} \end{bmatrix} \quad (50)$$

The error is then computed and injected in the states as follows:

$$\mathbf{e}_n = K_n \mathbf{z}_n \quad (51)$$

$$\mathcal{X}_n^+ = \mathcal{X}_n^- \cdot \exp(\mathbf{e}_n^x) \quad (52)$$

$$\mathbf{b}_n^+ = \mathbf{b}_n^- + \mathbf{e}_n^b \quad (53)$$

where \exp is the $\text{SE}_2(3)$ exponential map, $\mathbf{e}_n^x \in \mathbb{R}^9$ and $\mathbf{e}_n^b \in \mathbb{R}^6$ are sub-vectors of \mathbf{e}_n regarding the matrix Lie group and the bias vector, respectively.

REFERENCES

- [1] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annu. Rev. Control Robot., Auton. Syst.*, vol. 1, pp. 187–210, 2018.
- [2] N. El-Sheimy and Y. Li, "Indoor navigation: State of the art and future trends," *Satell. Navigation*, vol. 2, no. 1, pp. 1–23, 2021.
- [3] A. Noureldin, T. B. Karamat, and J. Georgy, *Fundamentals of Inertial Navigation, Satellite-Based Positioning and Their Integration*. Berlin, Germany: Springer, 2013.
- [4] U. Iqbal, A. F. Okou, and A. Noureldin, "An integrated reduced inertial sensor system-RISS/GPS for land vehicle," in *Proc. IEEE/ION Position, Location Navigation Symp.*, 2008, pp. 1014–1021.
- [5] A. Abosekeen, A. Noureldin, and M. J. Korenberg, "Improving the RISS/GNSS land-vehicles integrated navigation system using magnetic azimuth updates," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1250–1263, Mar. 2020.
- [6] Q. Zhang, L. Guan, and D. Xu, "Odometer velocity and acceleration estimation based on tracking differentiator filter for 3D-reduced inertial sensor system," *Sensors*, vol. 19, no. 20, 2019, Art. no. 4501.
- [7] A. Aboutaleb, A. S. El-Wakeel, H. Elghamrawy, and A. Noureldin, "Lidar/RISS/GNSS dynamic integration for land vehicle robust positioning in challenging gnss environments," *Remote Sens.*, vol. 12, no. 14, 2020, Art. no. 2323.
- [8] Y. Sun, L. Guan, M. Wu, Y. Gao, and Z. Chang, "Vehicular navigation based on the fusion of 3D-RISS and machine learning enhanced visual data in challenging environments," in *Electronics*, vol. 9, no. 1, 2020, Art. no. 193.
- [9] X. Song et al., "Enhanced map-aided GPS/3 D RISS combined positioning strategy in urban canyons," *Math. Problems Eng.*, vol. 2022, pp. 1–17, 2022.
- [10] M. Ramezani and K. Khoshelham, "Vehicle positioning in GNSS-deprived urban areas by stereo visual-inertial odometry," *IEEE Trans. Intell. Veh.*, vol. 3, no. 2, pp. 208–217, Jun. 2018.
- [11] G. Huang, "Visual-inertial navigation: A concise review," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 9572–9582.
- [12] T. Shan, B. Englöt, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5135–5142.
- [13] Y. Z. Ng, B. Choi, R. Tan, and L. Heng, "Continuous-time radar-inertial odometry for automotive radars," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 323–330.
- [14] R. Sun, Y. Yang, K.-W. Chiang, T.-T. Duong, K.-Y. Lin, and G.-J. Tsai, "Robust IMU/GPS/VO integration for vehicle navigation in GNSS degraded urban areas," *IEEE Sensors J.*, vol. 20, no. 17, pp. 10110–10122, Sep. 2020.
- [15] A. Chalvatzaras, I. Pratikakis, and A. A. Amanatiadis, "A survey on map-based localization techniques for autonomous vehicles," *IEEE Trans. Intell. Veh.*, vol. 8, no. 2, pp. 1574–1596, Feb. 2023.

- [16] Z. Zhang, J. Zhao, C. Huang, and L. Li, "Learning visual semantic map-matching for loosely multi-sensor fusion localization of autonomous vehicles," *IEEE Trans. Intell. Veh.*, vol. 8, no. 1, pp. 358–367, Jan. 2023.
- [17] V. Ilci and C. Toth, "High definition 3D map creation using GNSS/IMU/Lidar sensor integration to support autonomous vehicle navigation," *Sensors*, vol. 20, no. 3, 2020, Art. no. 899.
- [18] M. Brossard, A. Barrau, and S. Bonnabel, "AI-IMU dead-reckoning," *IEEE Trans. Intell. Veh.*, vol. 5, no. 4, pp. 585–595, Dec. 2020.
- [19] M. Karaim, A. Noureldin, and T. B. Karamat, "Low-cost IMU data denoising using savitzky-golay filters," in *Proc. IEEE Int. Conf. Commun., Signal Process., Appl.*, 2019, pp. 1–5.
- [20] M. Brossard, S. Bonnabel, and A. Barrau, "Denoising IMU gyroscopes with deep learning for open-loop attitude estimation," *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4796–4803, Jul. 2020.
- [21] K. Yuan and Z. J. Wang, "A simple self-supervised IMU denoising method for inertial aided navigation," *IEEE Robot. Automat. Lett.*, vol. 8, no. 2, pp. 944–950, Feb. 2023.
- [22] A. Barrau and S. Bonnabel, "The invariant extended Kalman filter as a stable observer," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 1797–1812, Apr. 2017.
- [23] E. R. Potokar, K. Norman, and J. G. Mangelson, "Invariant extended Kalman filtering for underwater navigation," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5792–5799, Jul. 2021.
- [24] H. Chen, H. Bai, and C. N. Taylor, "Invariant-EKF design for quadcopter wind estimation," in *Proc. Amer. Control Conf.*, 2022, pp. 1236–1241.
- [25] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle, "Contact-aided invariant extended Kalman filtering for robot state estimation," *Int. J. Robot. Res.*, vol. 39, no. 4, pp. 402–430, 2020.
- [26] J. Wang and Z. Chen, "Variational Bayesian iteration-based invariant Kalman filter for attitude estimation on matrix lie groups," *Aerospace*, vol. 8, no. 9, 2021, Art. no. 246.
- [27] Z. Zhang, J. Zhao, C. Huang, and L. Li, "Precise and robust sideslip angle estimation based on INS/GNSS integration using invariant extended Kalman filter," *Proc. Inst. Mech. Eng., Part D: J. Automobile Eng.*, to be published, doi: [10.1177/09544070221102](https://doi.org/10.1177/09544070221102).
- [28] T. Zhang, K. Wu, J. Song, S. Huang, and G. Dissanayake, "Convergence and consistency analysis for a 3-D invariant-EKF SLAM," *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 733–740, Apr. 2017.
- [29] M. Brossard, S. Bonnabel, and A. Barrau, "Invariant Kalman filtering for visual inertial slam," in *Proc. IEEE 21st Int. Conf. Inf. Fusion*, 2018, pp. 2021–2028.
- [30] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Anal. Chem.*, vol. 36, no. 8, pp. 1627–1639, 1964.
- [31] L. Gondara, "Medical image denoising using convolutional denoising autoencoders," in *Proc. IEEE 16th Int. Conf. Data Mining Workshops*, 2016, pp. 241–246.
- [32] Y. Cai, R. B. Santos, S. N. Givigi, and A. M. F. Fileti, "A pipeline leak classification and location estimation system with convolutional neural networks," *IEEE Syst. J.*, vol. 14, no. 3, pp. 3072–3081, Sep. 2020.
- [33] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," 2021, *arXiv:2003.05991*.
- [34] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
- [35] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," 2023, *arXiv:1606.08415*.
- [36] S. Heo and C. G. Park, "Consistent EKF-based visual-inertial odometry on matrix lie group," *IEEE Sensors J.*, vol. 18, no. 9, pp. 3780–3788, May 2018.
- [37] P. R. M. d. Araujo, E. Dawson, M. Elhabiby, S. Givigi, and A. Noureldin, "Evaluation of invariant extended Kalman filters applied to multi-sensor land vehicle navigation in GNSS challenging environments," in *Proc. 35th Int. Tech. Meeting Satell. Division Inst. Navigation*, 2022, pp. 1631–1649.
- [38] A. Barrau and S. Bonnabel, "Invariant Kalman filtering," *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 1, pp. 237–257, 2018.
- [39] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.
- [40] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [41] N. Sünderauf et al., "The limits and potentials of deep learning for robotics," *Int. J. Robot. Res.*, vol. 37, no. 4/5, pp. 405–420, 2018.



Paulo Ricardo Marques de Araujo (Graduate Student Member, IEEE) received the M.Sc. degree in engineering and innovation management from the Federal University of ABC, Santo André, Brazil. He is currently working toward the Ph.D. degree with the Electrical and Computer Engineering Department, Queen's University, Kingston, ON, Canada. His research interests include autonomous systems and robotics, machine learning, and digital manufacturing.



Mohamed Elhabiby was the Treasurer of the Geodesy Section, Canadian Geophysical Union from 2008 to 2014. He is currently an Associate Professor with the Faculty of Engineering, Ain Shams University, Cairo, Egypt. He is also the Executive Vice President and Co-Founder of Micro Engineering Tech Inc., Calgary, AB, Canada, a high-tech international company specialized in high-precision engineering and instrumentation, mobile mapping, laser scanning, deformation monitoring, and GPS/INS integrations. He is a Leader of an Archaeological Mission with the Area of the Great Pyramids, Cairo. He was the recipient of the Astech Awards. Avenue Magazine names him as one of the Top 40 under 40. He is the Chair of WG 4.1.4: Imaging Techniques, Sub-Commission 4.1: Alternatives and Backups to GNSS. He chaired the Geocomputations and Cyber Infrastructure Oral Session with the Canadian Geophysical Union annual meeting from 2008 to 2012.



Sidney Givigi (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Carleton University, Ottawa, ON, Canada. He is currently an Associate Professor with the School of Computing of Queen's University, Kingston, ON. His research interests mainly include machine learning, autonomous systems, and robotics.



Aboelmagd Noureldin (Senior Member, IEEE) received the B.Sc. degree in electrical engineering and the M.Sc. degree in engineering physics from Cairo University, Giza, Egypt, in 1993 and 1997, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Calgary, Calgary, AB, Canada, in 2002. He is currently a Professor with the Department of Electrical and Computer Engineering, Royal Military College of Canada, Kingston, ON, Canada, with Cross Appointments with the School of Computing and the Department of Electrical and Computer Engineering, Queen's University, Kingston. He is also the Founder and the Director of the Navigation and Instrumentation Research Lab at RMCC. He has authored or coauthored two books, four book chapters, and more than 270 papers in journals, magazines, and conference proceedings. His research interests include global navigation satellite systems, wireless positioning and navigation, indoor positioning, and multi-sensor fusion targeting applications related to autonomous systems, intelligent transportation, road information services, crowd management, and the vehicular Internet of Things. His research led to 13 patents and several technologies licensed to the industry in position, location, and navigation systems.