

Wheel-SLAM: Simultaneous Localization and Terrain Mapping Using One Wheel-mounted IMU

Yibin Wu^{1,2}, Jian Kuang¹, Xiaoji Niu¹, Jens Behley², Lasse Klingbeil², and Heiner Kuhlmann²

Abstract—A reliable pose estimator robust to environmental disturbances is desirable for mobile robots. To this end, inertial measurement units (IMUs) play an important role because they can perceive the full motion state of the vehicle independently. However, it suffers from accumulative error due to inherent noise and bias instability, especially for low-cost sensors. In our previous studies on Wheel-INS [1], [2], we proposed to limit the error drift of the pure inertial navigation system (INS) by mounting an IMU to the wheel of the robot to take advantage of rotation modulation. However, Wheel-INS still drifted over a long period of time due to the lack of external correction signals. In this letter, we propose to exploit the environmental perception ability of Wheel-INS to achieve simultaneous localization and mapping (SLAM) with only one IMU. To be specific, we use the road bank angles (mirrored by the robot roll angles estimated by Wheel-INS) as terrain features to enable the loop closure with a Rao-Blackwellized particle filter. The road bank angle is sampled and stored according to the robot position in the grid maps maintained by the particles. The weights of the particles are updated according to the difference between the currently estimated roll sequence and the terrain map. Field experiments suggest the feasibility of the idea to perform SLAM in Wheel-INS using the robot roll angle estimates. In addition, the positioning accuracy is improved significantly (more than 30%) over Wheel-INS. The source code of our implementation is publicly available (<https://github.com/i2Nav-WHU/Wheel-SLAM>).

Index Terms—SLAM, Localization

I. INTRODUCTION

STATE estimation is one of the most fundamental modules for autonomous vehicles. Therefore, modern ground mobile robots are commonly equipped with both exteroceptive sensors, e.g., Global Navigation Satellite System (GNSS) receiver, cameras, Light Detection And Ranging (LiDAR), and proprioceptive sensors, e.g., Inertial Measurement Unit (IMU) and odometer, to track the robot trajectories as well as perceive the environments. Among all these sensors, inertial sensors play a central role in robot navigation due to their self-contained characteristics, which means that it works independently without external signals and interaction with

Manuscript received: September 6, 2022; Revised November 4, 2022; Accepted November 28, 2022. This paper was recommended for publication by Editor Javier Civera upon evaluation of the Associate Editor and Reviewers' comments.

¹Yibin Wu, Jian Kuang, and Xiaoji Niu are with the GNSS Research Center, Wuhan University, Wuhan, China, {ybwu, kuang, xjniu}@whu.edu.cn.

²Yibin Wu, Lasse Klingbeil, Jens Behley, and Heiner Kuhlmann are with the Institute of Geodesy and Geoinformation, University of Bonn, Bonn, Germany, {firstname.lastname}@igg.uni-bonn.de. (Corresponding Author: *Jian Kuang*)

This work was funded in part by the National Key Research and Development Program of China (No. 2016YFB0501800 and No. 2016YFB0502202) and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy-EXC 2070-390732324.

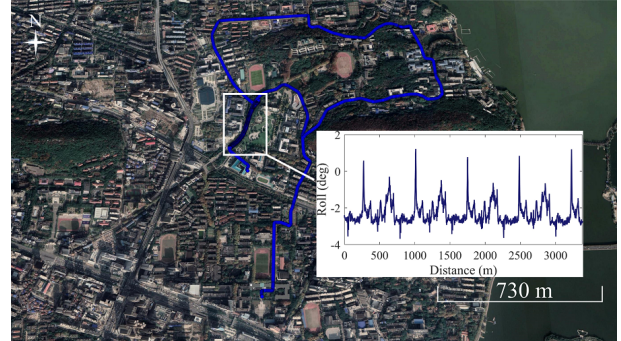


Fig. 1. Vehicle roll estimation and test trajectory in the car experiment of our prior work [1]. In the marked area, the car kept circling back and forth. It can be observed that the corresponding robot roll angle estimation (indicating the road bank angle) shows a repeating pattern that can be exploited to perform loop closure detection and correction.

the environment [3]. In most of the existing works, inertial sensors are used to perform Dead Reckoning (DR) to either complement other navigation techniques [4]–[6] or bridge the signal blockage of other sensors [7]. Although the advances in the microelectromechanical (MEMS) technique have made IMU ubiquitous in various devices, an Inertial Navigation System (INS) suffers from the curse of error drift due to the inherent sensor noise and bias instability. Therefore, it is both challenging and promising to improve the positioning performance of the stand-alone INS.

Inspired by the odometer-aided INS (ODO/INS) [8], we proposed Wheel-INS, a wheel-mounted IMU (Wheel-IMU)-based DR system, in our previous studies [1], [2]. Wheel-INS achieved competitive pose estimation performance with only one IMU. It was illustrated that the positioning and heading accuracy of Wheel-INS had been improved by 23% and 15% respectively over ODO/INS. Moreover, benefiting from the rotation modulation, Wheel-INS showed desirable resilience to the constant gyro bias error [1]. In summary, there are two major advantages of Wheel-INS. First, a similar information fusion scheme as ODO/INS is achieved by only one inertial sensor. Second, the continuous rotation of the Wheel-IMU significantly limits the heading error drift.

Although Wheel-INS exhibits excellent DR performance, it is only a relative positioning solution lacking the ability to limit long-term error accumulation. That is to say, the positioning error of Wheel-INS still drifts over time, especially when the stochastic error of the inertial sensor is significant. To correct the accumulated error without depending on absolute positioning techniques, a commonly used method is loop de-

tection and relocalization [9]. By extracting the environmental features with exteroceptive sensors, e.g., camera and LiDAR, the robot is allowed to recognize places that have been visited and the error drift can consequently be mitigated by, for example, performing a pose graph optimization using loop closure constraints [4]. Then, a natural question arises: *Can we only use the Wheel-IMU to achieve loop closure so as to further limit the error accumulation?*

From experiments of our prior work [1], we found that the robot roll estimation in Wheel-INS gave distinguishable and repeatable terrain-correlated responses, as shown in Fig. 1. Therefore, it would be possible to encode the robot roll angle as the road bank feature, so as to enable the loop closure detection in Wheel-INS. In addition, because the IMU is mounted on the wheel which is directly contacted with the ground, the roll angle estimated by Wheel-INS can represent the road bank angle precisely without being affected by the suspension system of the vehicle.

Based on the discussion above, this letter proposes Wheel-SLAM, a simultaneous localization and terrain mapping system using only one Wheel-IMU. Specifically, we extend our previous study on Wheel-INS [1] by exploiting the robot roll angle estimates to encode the terrain feature which is used for loop closure detection to further limit the error drift of Wheel-INS. Within a particle filter (PF) framework, the uncertainty of the robot state is sampled by multiple particles which maintain their own trajectories and terrain maps. By matching the current roll angle estimation with the map, loop closure can be discovered to update the weights of the particles. In summary, our main contributions include:

- 1) A SLAM system with only one Wheel-IMU using the terrain feature (measured by the Wheel-IMU) is proposed and implemented.
- 2) We illustrate the feasibility of exploiting robot roll angle estimates to enable loop closure so as to limit the error drift effectively in Wheel-INS through extensive field experiments.
- 3) To the best of our knowledge, this is the first SLAM system using only one low-cost wheel-mounted IMU for the wheeled robot in the literature. We make our code publicly accessible.

Note that this letter mainly focuses on the feasibility of the idea (achieving loop closure with only one Wheel-IMU in Wheel-INS) instead of the development of a complete system that can be straightforwardly applied to practical applications.

II. RELATED WORK

A. Terrain Matching-based Localization

Terrain-based vehicle localization provides a usable alternative to GNSS to obtain absolute positioning results by exploiting road information. The repetitive, location-dependent nature of the terrain features allows them to be used for robot localization and mapping [10]. Usually, the terrain features are extracted with the in-vehicle inertial sensor, for example, an IMU can measure the road grades by vehicle pitch, road bank angles by vehicle roll, and road curvature by vehicle yaw rate [11]. The basic hypothesis is that the inertial sensor

signals imply vehicle responses to the terrain surface and the same terrain surface excites similar vehicle motions [12]. Existing literature mainly adopts vehicle pitch angles and pitch differences as the features for the terrain-based localization [13], [14], although the roll angle can play the same role [15], [16]. However, using onboard inertial signals to determine the terrain information would be affected by the vehicle maneuver, for example, the braking may induce unexpected pitch variation of the vehicle [10] and the centripetal acceleration may introduce the difference between the vehicle roll angle and the road bank angle [17].

Various methods have been investigated to integrate the terrain feature matching result into the localization pipeline [10], [12], while PF has attracted more attention in recent studies. In PF methods, the weights of the particles are updated by evaluating the difference between the in-vehicle measurements (e.g., roll, pitch, etc.) and the response in the map [10], [15], [18]. Martini et al. [19] used the Pearson-product correlation coefficient as a distance metric to compare the road grade measurements with the reference map.

However, a pre-built map is indispensable in all these methods which limits its application. In addition, the inertial sensors used to measure the terrain information are all placed on the vehicle body. By extending our previous studies [1], [2], this letter proposes to localize the robot and measure the road bank angles simultaneously without a prior map. As the IMU is mounted on the robot wheel, the terrain matching would not be affected by the vehicle maneuver which is the case when the IMU is mounted on the vehicle body.

B. SLAM Only Using Inertial Sensors

Recently deep learning-based inertial localization systems have shown promising results in both pedestrian [20], [21] and vehicular navigation [22]. These methods learn either motion information from raw inertial measurements [20], [21] or dynamic measurement noise [22] to solve the inertial odometry problem in a data-driven way but fail to exploit environmental signals to limit long-term positional drift.

Angermann et al. [23] proposed a pedestrian SLAM system using only foot-mounted IMU (FootSLAM) by taking advantage of human perception and cognition. A dynamic Bayesian network was employed to represent the fact that when a pedestrian walks in a constrained environment, e.g., an office building, he or she relies mainly on visual cues to avoid obstacles and determine accessible areas. Specifically, the algorithm was implemented based on a PF where the weights of the particles were updated by the probability of the pedestrian crossing transitions in a regular 2D grid of adjacent hexagons. After that, a probabilistic transition map implicitly encoding the environmental features that influence the pedestrian's visual impression and intention was constructed.

Different from FootSLAM where the building layout is implicitly used to perform SLAM by taking advantage of human cognition, Wheel-SLAM uses the Wheel-IMU to explicitly extract the terrain feature for loop closure detection.

In conclusion, Wheel-SLAM borrows ideas from both the terrain matching-based vehicle localization and FootSLAM.

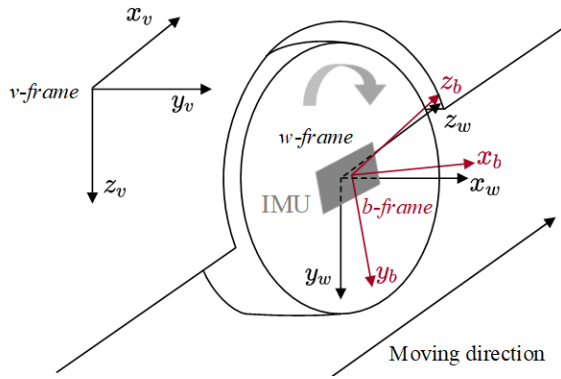


Fig. 2. Installation scheme of the Wheel-IMU and the definitions of the vehicle frame (*v*-frame), wheel frame (*w*-frame), and IMU body frame (*b*-frame) [1].

In contrast to Wheel-INS [1], we extend the approach by extracting the road features with the Wheel-IMU to enable loop closure, so as to further limit the error drift. In Wheel-SLAM, we maintain and update the grid map in real-time and detect the loop closure using the DR result from Wheel-INS. After that, we use the roll angle sequence matching results between current estimates and the map to update the weights of the particles for the sake of robustness.

III. METHODOLOGY

In this section, a brief introduction to Wheel-INS is first provided. Then we explain the details of the PF-based Wheel-SLAM algorithm, including the dynamic Bayesian model, the construction of the terrain map, and the update of the weights of the particles.

A. Background

Wheel-INS [1] is the foundation of Wheel-SLAM. It is used to provide the robot odometry and roll angle estimation. There are two major advantages of Wheel-INS. First, the wheel velocity can be calculated by the gyro output and wheel radius enabling the same information fusion as ODO/INS with only one IMU (no other sensors). Second, it can take advantage of rotation modulation to limit the error drift of INS.

Due to the space limitation, we just outline the algorithm of Wheel-INS here. Please refer to our earlier papers [1], [2] for details, e.g., the rotation modulation of the Wheel-IMU, the definition of the misalignment errors, etc.

Fig. 2 depicts the installation of the Wheel-IMU and the definition of related coordinate systems. The system overview of Wheel-INS is shown in Fig. 3. First, the forward INS mechanization is performed to predict the robot states. At the same time, the gyroscope outputs in the *x*-axis of the Wheel-IMU are used to calculate the wheel speed. Then, this vehicle velocity is treated as an external observation with non-holonomic constraint (NHC) [24] to update the state through an error-state extended Kalman filter (EKF) [25].

The forward wheel velocity calculated by the gyroscope data of the Wheel-IMU and wheel radius can be written as

$$\begin{aligned} \tilde{v}_{wheel}^v &= \tilde{\omega}_x r - e_v = (\omega_x + \delta\omega_x)r - e_v \\ &= v_{wheel}^v + r\delta\omega_x - e_v \end{aligned} \quad (1)$$

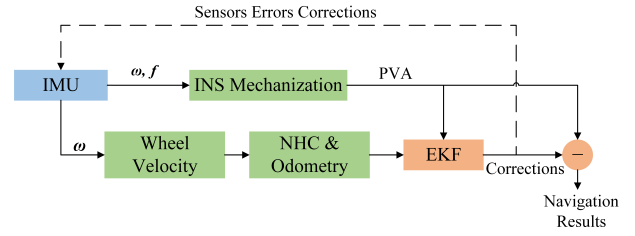


Fig. 3. Overview of Wheel-INS [1]. ω and f are the angular rate and specific force measured by the Wheel-IMU, respectively; PVA represents the position, velocity, and attitude of the Wheel-IMU. We use the output from the Wheel-IMU to perform INS mechanization to predict the robot state (PVA). The angular velocity measured in the *x*-axis of the Wheel-IMU and the wheel radius are used to calculate the forward speed. This speed is then integrated with NHC as a 3D velocity observation to update the robot state as well as correct the inertial sensor errors, e.g., gyro bias, through an EKF.

where \tilde{v}_{wheel}^v and v_{wheel}^v are the observed and true vehicle forward velocity, respectively; $\tilde{\omega}_x$ is the gyroscope output in the *x*-axis; ω_x is the true value of the angular rate in the *x*-axis of the Wheel-IMU; $\delta\omega_x$ is the gyroscope measurement error; r is the wheel radius, and e_v is the observation noise of the wheel velocity, modeled as Gaussian white noise.

Because the Wheel-IMU rotates with the wheel, the pitch of the robot is unknown in Wheel-INS. In other words, we cannot determine the robot's ascent and descent in Wheel-INS. Therefore, it is assumed that the robot is moving on a horizontal plane. However, experimental results [1] illustrated that this assumption does not cause significant navigation errors.

B. Dynamic Bayesian Network

A PF is a sequential Monte Carlo method where the basic idea is the recursive computation of relevant probability distributions using the concepts of importance sampling and approximation of probability distributions with discrete random measures [26]. In PF, the posterior distribution of the robot state is represented by a set of particles that evolve recursively with the integration of new information. Based on the technique of Rao-Blackwellization [27]–[29], Wheel-SLAM decomposes the SLAM problem into a robot localization problem and a terrain mapping problem that is conditioned on the robot pose estimate.

In Wheel-SLAM, we try to estimate the posterior

$$p(x_{1:t}, \mathbf{M} | z_{1:t}, u_{1:t}) \quad (2)$$

which is the distribution representing the robot state $x_{1:t}$ and the terrain map \mathbf{M} based on the set of control inputs $u_{1:t}$ which governs the robot motion and the road bank angle observations $z_{1:t}$. The conditional independence property of the Wheel-SLAM problem implies that the posterior in (2) can be factored as follows:

$$p(x_{1:t}, \mathbf{M} | z_{1:t}, u_{1:t}) = p(x_{1:t} | z_{1:t}, u_{1:t}) \prod_{i=1}^{N_f} p(m_i | z_{1:t}, x_{1:t}) \quad (3)$$

where m_i is the *i*-th terrain feature, and N_f is the total number of the features. In Wheel-SLAM, Wheel-INS is performed

Algorithm 1 Wheel-SLAM

Input: Robot pose x_{t-1} , terrain map, \mathbf{M}_{t-1} , control input, u_t and measurement, z_t .

Output: Robot pose x_t , terrain map, \mathbf{M}_t .

- 1: **for** each $i = 1 \rightarrow N_p$ **do**
 - 2: Predict the vehicle pose by the motion model $p(x_t|x_{t-1}, u_t)$.
 - 3: Update the terrain map $p(\mathbf{M}_t|x_t, z_t)$ if necessary.
 - 4: **if** Loop closure detected && Roll sequence matches **then**
 - 5: Update the weight of the particles according to (6).
 - 6: **end if**
 - 7: **end for**
 - 8: Normalize the weights of the particles $\omega^i = \omega^i / \sum_{i=1}^{N_p} \omega^i$.
 - 9: **if** Resampling is required ($N_{\text{eff}} = 1 / \sum_{i=1}^{N_p} (\omega^i)^2 < 0.75$) **then**
 - 10: Perform resampling.
 - 11: **end if**
-

for the robot state estimation as well as the road bank angle perception.

Wheel-SLAM uses a PF to estimate the robot trajectory distribution. For each particle, the individual trajectory-based terrain map is independent of each other. As a result, every particle is composed of a robot pose and a terrain map; thus, the i -th particle at time t can be represented as

$$X_i^t = [x_i^t \quad \mathbf{M}_i] \quad (4)$$

where $i = 1, 2, 3, \dots, N_p$, is the index of the particle while N_p is the total number of the used particles. x_i^t is the pose of the robot estimated by the i -th particle at time t , and \mathbf{M}_i is the terrain map maintained by the i -th particle. Here the robot pose is represented by 2D translation $p_i^t \in \mathbb{R}^2$ and heading $\mathbf{R}_i^t \in SO(2)$ as Wheel-INS cannot estimate pitch angle, namely, the robot motion in the vertical direction [1].

The Wheel-SLAM algorithm consists of four main steps: 1) Sample new robot state by the motion model; 2) Update the terrain map; 3) Update the particle weights once a convinced loop closure is reported; 4) Resample the particles when it is necessary. Algorithm 1 shows an overview of Wheel-SLAM.

The first step is to generate a new pose for each particle by sampling from the robot probabilistic motion model:

$$p(x_t|x_{t-1}, u_t) \quad (5)$$

which means that the robot pose, x_t , is a probabilistic function of the robot control u_t and the previous pose x_{t-1} . Here, we adopt Wheel-INS to predict the robot state.

C. Grid Terrain Map Construction

Compared to the hexagon grid map used in FootSLAM [23], we simplify the grid to square due to the relatively simple robot motion mode. As we assume that the vehicle is moving

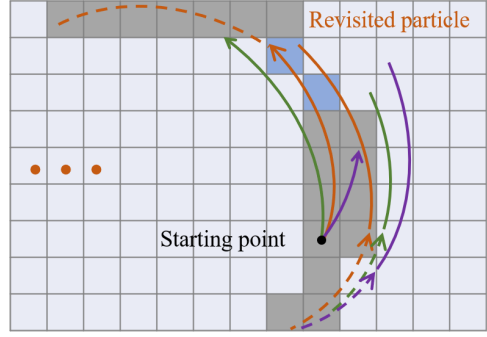


Fig. 4. Illustration of the grid map construction and revisit recognition. Different colors of curves represent the robot path sampled by different particles. The gray grids have been visited by the robot thus they have a road bank angle estimation. Once a particle detects that the robot has continuously returned to visited grids (blue), a potential loop closure is reported for further check (Please refer to Section III-D for details).

on the horizontal plane, we build a 2D grid map. Each grid holds the corresponding road bank angle estimated by Wheel-INS at that position. Fig. 4 illustrates the grid map constructed along with the robot pose evolution.

D. Particle Weight Update

In the beginning, all the particles are assigned the same weight. When the robot moves, each particle has a different trajectory and terrain map. To ensure the reliability of loop closure and reduce the impact of an outlier, we set three criteria. First, loop closures need to be continuously detected by the robot position in a window of length N_r . Second, we calculate the N_r roll sequence matching scores using Pearson correlation coefficient [19] and compare them with a threshold C_{thr} . In this N_r window, at least N_{thr} ($N_{\text{thr}} \leq N_r$) coefficients need to be larger than C_{thr} . Third, the correlation coefficient at the current position needs to be larger than the threshold. If all three requirements are met, we think it is a true loop closure and subsequently update the particle weights as follows:

$$\omega_k^i = \omega_{k-1}^i \frac{N_c}{N_r} \exp(\text{RMSE}(V_{\text{coeff}})) \quad (6)$$

where k indicates the time epoch; i indicates the number of the particle; N_c ($N_c \geq N_{\text{thr}}$) is the number of the correlation coefficients larger than the threshold in the window N_r ; RMSE indicates the root mean square error; V_{coeff} is the collection of the correlation coefficients larger than the threshold in the window. After that, normalization is performed to make the sum of the weights equal to one.

Note that although Wheel-SLAM requires the exact revisit of the robot, the vehicle doesn't need to always drive on the same road. The position error will accumulate when the robot explores an unknown environment, but it can be corrected once the robot is back on a previously-visited road. Please refer to Section IV for the experimental results and discussion. Moreover, as our algorithm requires a sequence of road bank angles for matching, it is not able to detect loop closure effectively at crossroads where the robot may enter a place perpendicular to its last entrance direction.

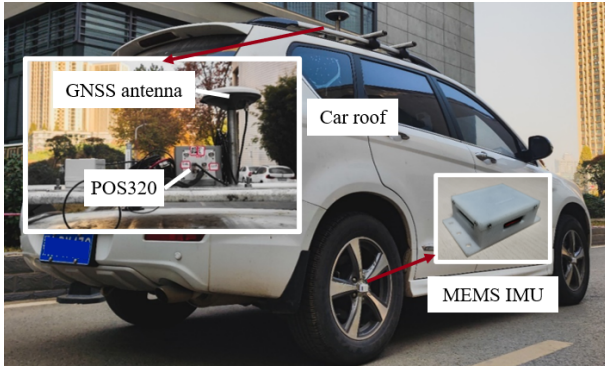


Fig. 5. Experimental platform. A GNSS antenna and a high-end IMU (POS320) were mounted on the car roof to provide the ground truth of the vehicle pose while a low-cost MEMS IMU was mounted on the right rear wheel to perform Wheel-SLAM.

TABLE I
VEHICLE MOTION INFORMATION IN THE EXPERIMENTS

Sequence	I	II	III	IV	V
Average speed (m/s)	5.41	5.47	4.93	5.16	4.89
Total distance (m)	≈ 2950	≈ 3791	≈ 2802	≈ 7235	≈ 9285

IV. EXPERIMENTAL RESULTS

This section presents and discusses the real-world experimental results to illustrate the positioning performance of Wheel-SLAM. As this work focus on the feasibility of the idea (achieving loop closure with only one Wheel-IMU) instead of the development of a complete system, we do not evaluate the computational efficiency of Wheel-SLAM. In addition, there is no paper in the literature studying SLAM with the same sensor set (wheel-mounted inertial sensors). We only compare the proposed Wheel-SLAM with its predecessor, Wheel-INS, to illustrate its effectiveness and discuss its characteristics. First, the experimental conditions are described. After that, we compare the positioning accuracy of Wheel-SLAM with Wheel-INS. The characteristics of Wheel-SLAM are also analyzed. Finally, we further discuss the key components which play a central role in Wheel-SLAM and some limitations in real-world applications.

A. Experimental Description

To demonstrate the feasibility and effectiveness of the proposed Wheel-SLAM system, we carried out five sets of field tests using a car on the campus of Wuhan University. The car was instrumented with one Wheel-IMU and reference system to provide the ground truth of the vehicle pose, as shown in Fig. 5. The characteristics of the vehicle motion in the tests are shown in Table I.

We used the same MEMS IMU from our prior works [1], [2]. The MEMS IMU contained four ICM20602 (TDK InvenSense) inertial sensor chips, a chargeable battery module, a microprocessor, an SD card for data collection, and a Bluetooth module for communication and data transmission. One can use an android phone to control the data collection.

TABLE II
TECHNICAL PARAMETERS OF THE IMUS USED IN THE TESTS

IMU	Gyro Bias ($^{\circ}/h$)	ARW ($^{\circ}/\sqrt{h}$)	Acc. Bias (m/s^2)	VRW ($m/s/\sqrt{h}$)
POS320	0.5	0.05	0.00025	0.1
ICM20602	200	0.24	0.01	3

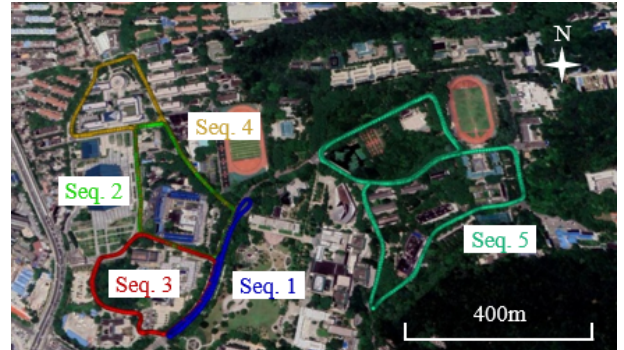


Fig. 6. Experimental trajectories. Seq. 1, Seq. 2, and Seq. 3 are circular trajectories where the vehicle moved several times in one direction while Seq. 4 and Seq. 5 are more complicated ones where the vehicle moved in different directions (not just turn around in one direction) in large-scale environments.

We collected the data of one chip of the Wheel-IMU for post-processing. The reference system used in the experiments was a high-accuracy position and orientation system with a tactical-grade IMU (POS320, MAP Space Time Navigation Technology Co., Ltd., China). The reference data were processed through a smoothed post-processed kinematic (PPK)/INS integration method. The main technical parameters of both the MEMS IMU and the high-end IMU are listed in Table II, where ARW denotes the angle random walk; Acc. denotes the accelerometer; VRW denotes the velocity random walk.

Fig. 6 shows the five experimental trajectories. In Sequence (Seq.) 1, the car was traveling back and forth on two parallel and opposite-direction roads. Seq. 2, and Seq. 3 are circular trajectories where the vehicle moved several times in one direction. Seq. 4 and Seq. 5 are more complicated ones with two circles in large-scale environments. In Seq. 4 and Seq. 5, the car not only traveled in the same lane in the same direction but moved in opposite directions on the same road. Note that in Seq. 1, Seq. 2, and Seq. 3, the car sometimes also changed the lane. Please refer to our source code webpage for the street views of the sequences.

The static IMU data before the car started moving were used to obtain the initial roll and pitch angle of the Wheel-IMU, as well as the initial value of the gyroscope bias. Other inertial sensor errors were set as zero. The key parameters of Wheel-SLAM set in the experiments are listed in Table III. Standard deviation is denoted as STD in Table III.

B. Performance Comparison and Analysis

1) *Performance Comparison*: Fig. 7 compares the positioning and heading errors of Wheel-SLAM and Wheel-INS in all five experiments, respectively. Please refer to our source code

TABLE III
KEY PARAMETERS OF WHEEL-SLAM IN THE EXPERIMENTS.

Parameters	Value
Particle number	100
Grid size of the 2D terrain map	1.5 m
Distance increment STD	0.025 m
Heading increment STD	0.05 °
Roll sample distance	0.5 m
Roll matching sequence length	25 m
Correlation coefficient threshold	0.4

webpage for the positioning results of Wheel-SLAM overlaid on Google earth. It is obvious that compared to Wheel-INS, Wheel-SLAM can suppress the error drift effectively, even in complicated scenarios where the car moves in different directions in large-scale environments, e.g., Seq. 4 and Seq. 5. However, the positioning error is increasing with drift of heading error in Wheel-INS. In addition, we can also notice that there are periodic decreases in the positioning error of Wheel-INS. This is due to the loop closure of the trajectory which cancels part of the cumulative error as discussed in our prior work [1].

At the start of the experiments, the positioning results of Wheel-SLAM drift together with Wheel-INS because all the particles are sampled from Wheel-INS according to the Gaussian distribution and there is no external measurement to correct the position error. Once a credible loop closure is reported by some particles, they get a larger weight. Accordingly, the vehicle position would deflect to the previous trajectory when the vehicle visited the same place the last time. It can be observed that there are some sharp decreases in the positioning and heading error of Wheel-SLAM in the experiments, such as the 150 s in Fig. 7(d) and the 1000 s in Fig. 7(j). This is because we use the weighted average value of all the particles as the output. The increased weights of those particles which detect the loop closure would drag the vehicle position to the previous trajectory. In addition, we only correct the current robot state but not the historical trajectory. Using modern graph optimization tools, e.g., gtsam [30], can jointly optimize the history state so as to make the trajectory smoother and improve the overall accuracy, but this letter mainly focuses on the feasibility to extract road features to enable loop closure in Wheel-INS.

We can observe from the figures that both the positioning and heading error of Wheel-INS increase quickly over time, while the same value of Wheel-SLAM is constrained at a certain level. When there is no loop closure, Wheel-SLAM exhibits the same drift as Wheel-INS in the first circle of the trajectory. However, as the vehicle continues to revisit the roads previously visited, the position drift of Wheel-SLAM can be limited as that in the first circle because of the loop closure mechanism. This suggests that the proposed method is effective to limit the error growth of Wheel-INS by using terrain information perceived by the Wheel-IMU to perform the loop closure.

It is worth mentioning that in Seq. 5, there is a large

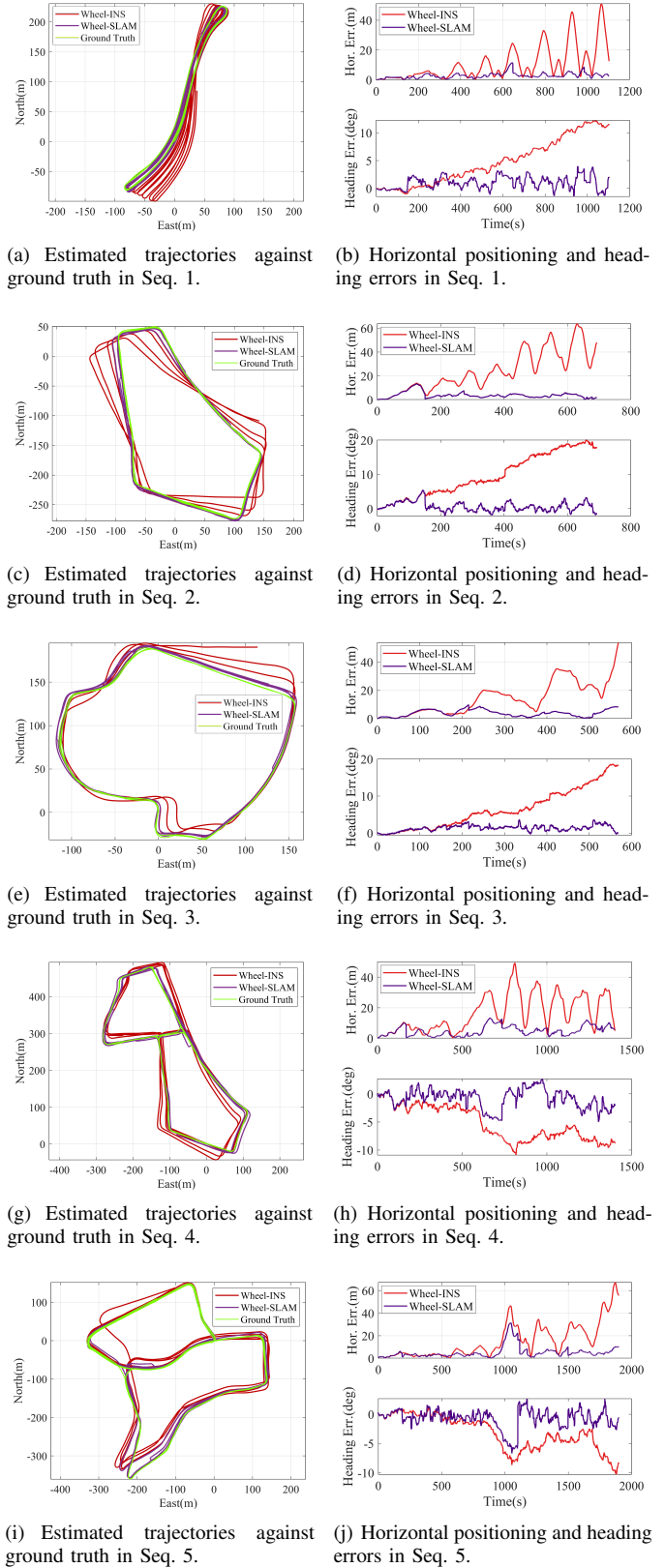


Fig. 7. The estimated trajectories and corresponding horizontal position and heading errors of Wheel-INS and Wheel-SLAM in the five experiments.

position and heading drift from about 860 s to 1100 s (see

TABLE IV
POSITIONING AND HEADING ERROR STATISTICS OF WHEEL-SLAM AND WHEEL-INS

Seq.	Horizontal Pos. RMSE(m)		Heading RMSE (°)	
	Wheel-INS	Wheel-SLAM	Wheel-INS	Wheel-SLAM
1	5.70	2.50	1.96	1.00
2	27.09	9.38	11.36	3.17
3	18.03	8.27	8.32	3.83
4	20.24	9.21	6.09	2.43
5	21.44	14.42	4.26	2.95

Fig. 7(j)). This is because the vehicle position error of Wheel-INS drifts quickly at that time. Meanwhile, there is no loop closure reported by the particles, so Wheel-SLAM exhibits a similar drift trend during this time period. Afterward, a loop closure is successfully detected in Wheel-SLAM at about 1100 s, resulting in a radical error correction.

Table IV lists the error statistics of the navigation results of Wheel-SLAM and Wheel-INS in all five experiments. We calculate the root mean square error (RMSE) of the horizontal position and heading (denoted as Horizontal Pos. RMSE and Heading RMSE in IV, respectively) as the indicators to evaluate the navigation performance. Additionally, given the stochasticity of the particle sampling, we run the algorithm multiple times to get the mean value of each test. It can be learned that thanks to the loop closure detection, Wheel-SLAM overwhelmingly outperforms Wheel-INS in terms of both position and heading estimation. Compared to Wheel-INS, the positioning and heading accuracy has been improved by 32.7% ~ 65.4% and 30.7% ~ 72.1%, respectively.

As Seq. 1, Seq. 2, and Seq. 3 are rather simple, there are more opportunities for loop closures. Therefore, the performance improvements in these three tests are more significant than that in Seq. 4 and Seq. 5.

The terrain maps built by Wheel-SLAM in Seq. 1, Seq. 2, and Seq. 4 are shown in Fig. 8. Because the wheel of the vehicle contacts the ground directly, the mapping is not affected by the suspension system of the vehicle especially when the vehicle maneuvering is large which is the case when the IMU is mounted on the vehicle body. Furthermore, these maps can be used to provide valuable information to monitor the deformation and deterioration of the roads.

2) *Analysis on the characteristics of Wheel-SLAM*: To further evaluate the performance and stability of Wheel-SLAM, we set different particle numbers to compare the positioning performance. The algorithm was run 100 times for each configuration. Fig. 9 shows the results.

It can be observed that the position error of Wheel-SLAM is more centralized when there are more particles, which means that the stability of Wheel-SLAM is improved with the increment of particles. This is because it is more likely for the system to detect the real loop closure with more particles and the performance would also be less susceptible to anomalies. However, there is no obvious gain by adding particles from 1000 to 2000 because of the diminishing marginal effect. In addition, the accuracy of Wheel-SLAM also depends on the

position error when the vehicle visits the place for the first time. In consequence, continuing to add the particles only improve position accuracy lightly.

Furthermore, it can be learned that in a statistical sense, the overall positioning performance of Wheel-SLAM is also not significantly enhanced with the increase of the particles. Although there are some outliers when the particle number is small (= 100), the system is robust to recognize the loop closure thanks to the excellent DR ability of Wheel-INS.

C. Discussion

The core principles behind Wheel-SLAM can be summarized as follows: 1) Particles are spread to sample the possible state of the robot and detect the loop closure by the trajectory maintained by each particle; 2) The road bank angle sequence matching result is used to update the particle weights, so as to pick out the most trusted one(s). What plays a central role in Wheel-SLAM is the roll sequence matching strategy. It must be robust enough to keep the outstanding particles while filtering out false alarms. Therefore, we used a rather strict loop closure detection criteria to make the loop closure detection robust.

However, it can be realized that there are two major limitations in the application of Wheel-SLAM. First, the robot must strictly revisit the previous places with a certain length. It is not like the vision-based SLAM where the vehicle has the remote sensing ability by using exteroceptive sensors, e.g., camera and LiDAR. In Wheel-SLAM, the Wheel-IMU is used to extract the terrain features which can only be obtained by the exact arrival of the robot. Second, the success of the loop closure depends on the matching of the road bank angle sequence. If the robot is moving on extremely smooth roads without any fluctuations in the bank angle, it would be difficult to detect loop closure.

V. CONCLUSIONS

In this study, we propose to perform SLAM with only one Wheel-IMU by exploiting the environmental perception ability of the Wheel-IMU. To be specific, we extend our previous study on Wheel-INS to Wheel-SLAM by extracting the terrain feature from the robot roll angle estimates to enable loop closure detection. The system is implemented with a Rao-Blackwellized particle filter where each particle maintains its own robot state and grid map. Experimental results show that the proposed method can effectively suppress the error drift of Wheel-INS. The positioning and heading accuracy has been averagely improved by 52.6% and 53.2%, respectively, against Wheel-INS.

However, Wheel-SLAM has two major limitations. First, a certain level of variation in the bank angles of the road is needed. Second, the robot has to revisit the same place exactly. Wheel-SLAM would be suitable for those kinds of robots that move repeatedly in given areas, for example, the sweeping robots and patrol robots in restricted areas.

For future research, integrating Wheel-SLAM with other exteroceptive sensors (e.g., camera and LiDAR) would be promising to improve the robustness and applicability of the robot navigation system.

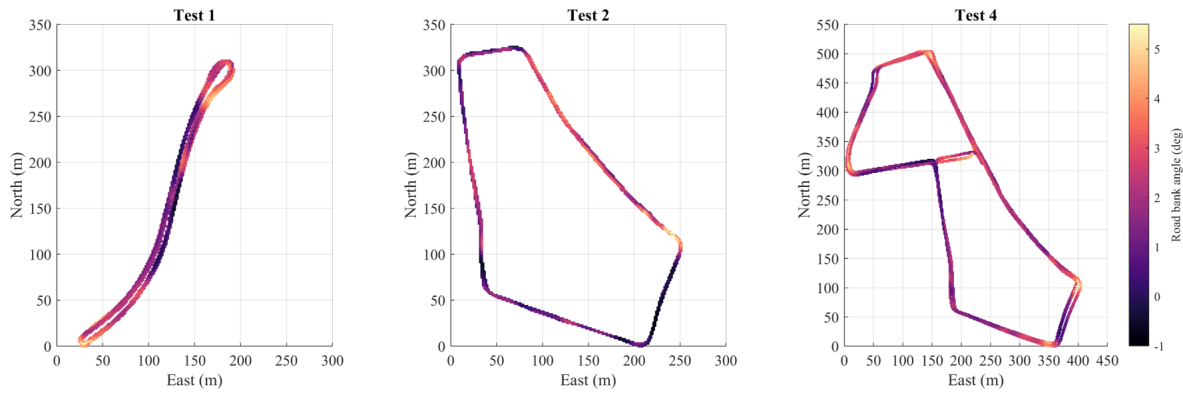


Fig. 8. The terrain maps estimated by Wheel-SLAM in Seq. 1, Seq. 2, and Seq. 4, respectively. The colors represent the values of the road bank angles. The larger the road bank angle, the lighter the color.

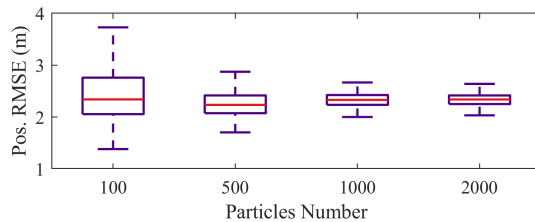


Fig. 9. The positioning RMSE of Wheel-SLAM with different particle numbers in Seq. 1. Medians are indicated by the red lines, while the bottom and top edges of the boxes indicate the first quartile and third quartile, respectively, and the whiskers show the maximum (upper) and minimum (lower).

REFERENCES

- [1] X. Niu, Y. Wu, and J. Kuang, "Wheel-INS: A wheel-mounted MEMS IMU-based dead reckoning system," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 9814–9825, 2021.
- [2] Y. Wu, X. Niu, and J. Kuang, "A comparison of three measurement models for the wheel-mounted MEMS IMU-based dead reckoning system," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 11 193–11 203, 2021.
- [3] N. El-Sheimy and Y. Li, "Indoor navigation: State of the art and future trends," *Satell. Navig.*, vol. 2, no. 1, pp. 1–23, 2021.
- [4] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [5] G. Huang, "Visual-inertial navigation: A concise review," in *Proc. Int. Conf. Robot. Automat.*, pp. 9572–9582. IEEE, 2019.
- [6] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct lidar-inertial odometry," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, 2022.
- [7] Y. Wu, X. Niu, J. Du, L. Chang, H. Tang, and H. Zhang, "Artificial marker and mems imu-based pose estimation method to meet multirotor uav landing requirements," *Sensors*, vol. 19, no. 24, p. 5428, 2019.
- [8] W. Ouyang, Y. Wu, and H. Chen, "INS/odometer land navigation by accurate measurement modeling and multiple-model adaptive estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 1, pp. 245–262, 2021.
- [9] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [10] T. Li, M. Yang, H. Li, L. Deng, and C. Wang, "A terrain-based vehicle localization approach robust to braking," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 2923–2932, 2019.
- [11] H. Zhang, W. Li, C. Qian, and B. Li, "A real time localization system for vehicles using terrain-based time series subsequence matching," *Remote Sensing*, vol. 12, no. 16, p. 2607, 2020.
- [12] J. Gim and C. Ahn, "Inertial-sensor-based non-dead reckoning localization for ground vehicles," *IEEE Access*, vol. 9, pp. 132 766–132 778, 2021.
- [13] P. K. Vemulapalli, A. J. Dean, and S. N. Brennan, "Pitch based vehicle localization using time series subsequence matching with multi-scale extrema features," in *Proc. Amer. Control Conf.*, pp. 2405–2410, 2011.
- [14] E. I. Laftchiev, C. M. Lagoa, and S. N. Brennan, "Vehicle localization using in-vehicle pitch data and dynamical models," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 206–220, 2015.
- [15] H. Ahmed and M. Tahir, "Terrain-based vehicle localization using low cost mems-imu sensors," in *Proc. IEEE Veh. Technol. Conf.*, pp. 1–5, 2016.
- [16] A. J. Dean, "Terrain-based road vehicle localization using attitude measurements," Ph.D. dissertation, Department of Mechanical and Nuclear Engineering, The Pennsylvania State University, Philadelphia, U.S., 2008.
- [17] J. Ryu and J. Gerdes, "Estimation of vehicle roll and road bank angle," in *Proc. Amer. Control Conf.*, vol. 3, pp. 2110–2115, 2004.
- [18] A. Dean, R. Martini, and S. Brennan, "Terrain-based road vehicle localisation using particle filters," *Veh. Syst. Dyn.*, vol. 49, no. 8, pp. 1209–1223, 2011.
- [19] R. D. Martini, "GPS/INS sensing coordination for vehicle state identification and road grade positioning," Master's thesis, Department of Mechanical and Nuclear Engineering, The Pennsylvania State University, Philadelphia, U.S., 2006.
- [20] C. Chen, X. Lu, A. Markham, and N. Trigoni, "Ionet: Learning to cure the curse of drift in inertial odometry," *Proc. AAAI Conf. on Artif. Intell.*, vol. 32, no. 1, 2018.
- [21] S. Herath, D. Caruso, C. Liu, Y. Chen, and Y. Furukawa, "Neural inertial localization," in *Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pp. 6604–6613, Jun. 2022.
- [22] M. Brossard, A. Barrau, and S. Bonnabel, "Ai-imu dead-reckoning," *IEEE Trans. on Intell. Veh.*, vol. 5, no. 4, pp. 585–595, 2020.
- [23] M. Angermann and P. Robertson, "FootSLAM: Pedestrian simultaneous localization and mapping without exteroceptive sensors-hitchhiking on human perception and cognition," *Proc. IEEE*, vol. 100, pp. 1840–1848, 2012.
- [24] G. Dissanayake, S. Sukkarieh, E. Nebot, and H. Durrant-Whyte, "The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications," *IEEE Trans. Robot.*, vol. 17, no. 5, pp. 731–747, 2001.
- [25] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.
- [26] P. Djuric, J. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. Bugallo, and J. Miguez, "Particle filtering," *IEEE Signal Process. Mag.*, vol. 20, no. 5, pp. 19–38, 2003.
- [27] K. Murphy and S. Russell, "Rao-Blackwellised particle filtering for dynamic Bayesian networks," in *Sequential Monte Carlo methods in practice*, pp. 499–515. Springer, 2001.
- [28] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with Rao-Blackwellized particle filters," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 34–46, 2007.
- [29] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit *et al.*, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. AAAI Conf. on Artif. Intell.*, pp. 593–598, 2002.
- [30] F. Dellaert, R. Roberts, V. Agrawal, and *et al.*, "borglab/gtsam," May. 2022. [Online]. Available: <https://doi.org/10.5281/zenodo.5794541>