Vermessung & Geoinformation 1/2024, pp. 6 – 15, 10 Figs.

Increasing Localization Robustness in a LiDAR-focused SLAM with a combined IMU and wheel odometry model

Erhöhung der Lokalisierungsrobustheit in einem LiDAR-basierten SLAM mit einem kombinierten IMU- und Rad-Odometrie-Modell



Eva Reitbauer, Christoph Schmied und Fabian Theurl, Graz

Dieser Beitrag wurde als „reviewed paper“ angenommen.

Abstract

Precise positioning and mapping are key technologies for autonomous robots. Most autonomous and semi-autonomous systems use a LiDAR-focused Simultaneous Localization and Mapping (SLAM) in combination with MEMS[1] Inertial Measurement Units (IMU). When LiDAR fails and only IMU data are used to compute the state vector of the robot, errors accumulate. After a few seconds without LiDAR data, LiDAR-inertial SLAM systems can no longer register the scan after the outage to the already created map. This paper proposes a novel approach for fusing LiDAR, inertial, and wheel odometry in a factor graph for SLAM. A combined IMU and wheel odometry model is used as an initial guess for LiDAR scan matching and to bridge LiDAR outages. The algorithm is evaluated and tested for two different robot models. The results show that with the proposed IMU and wheel odometry model localization accuracy improves during LiDAR outages. After 30 seconds without LiDAR data, the LiDAR point clouds after the outage can still be matched to the previously created map.

Keywords: SLAM, factor graph optimization, LiDAR, IMU, wheel odometry

Kurzfassung

Präzise Positionierung und Kartenerstellung sind Schlüsseltechnologien für autonome Roboter. Die meisten autonomen und halbautonomen Systeme verwenden ein LiDAR-basiertes Simultaneous Localization and Mapping (SLAM)-System in Kombination mit MEMS[2] inertialen Messeinheiten (IMU). Wenn LiDAR ausfällt und nur mehr IMU-Daten zur Berechnung des Zustandsvektors des Roboters herangezogen werden, werden Messfehler akkumuliert und die Position des Roboters driftet weg. Nach einigen Sekunden ohne LiDAR-Daten sind LiDAR-inertiale SLAM-Systeme nicht mehr in der Lage, den Scan nach dem Ausfall in der bereits erstellten Karte zu registrieren. In diesem Beitrag wird ein neuartiger Ansatz zur Fusionierung von LiDAR, Inertial- und Rad-Odometrie in einem Faktorgraphen für SLAM vorgestellt. Ein kombiniertes IMU- und Rad-Odometriemodell wird als erste Schätzung für den LiDAR-Scanabgleich und zur Überbrückung von LiDAR-Ausfällen verwendet. Der Algorithmus wird für zwei verschiedene Robotermodelle validiert und getestet. Die Ergebnisse zeigen, dass mit dem neuen IMU- und Rad-Odometrie-Modell die Lokalisierungsgenauigkeit während LiDAR-Ausfällen verbessert wird. Nach 30 Sekunden ohne LiDAR-Daten können die LiDAR-Punktwolken nach dem Ausfall immer noch zur zuvor erstellten Karte registriert werden.

Schlüsselwörter: SLAM, Faktorgraph-Optimierung, LiDAR, IMU, Rad-Odometrie

1) Micro-Electro-Mechanical System
2) Mikro-Elektro-Mechanisches System

1. Introduction

In the last two decades, a trend towards automation and autonomous platforms has been observable across many sectors. In logistics, automated mobile robots are used in warehouses [1–3]. In the agricultural sector, mobile robots are used for smart farming [4–6]. In the automotive sector, advances have been made for Advanced Driver Assistance Systems [7] and for autonomous vehicles [8]. Moreover, mobile robots can support first responders in their daily work and in extreme emergency situations. Use cases range from fire extinction to the search and rescue of injured persons or the identification of chemical, biological, radiological, nuclear, and explosive (CBRNE) substances [9]. In order for the robot to complete these tasks autonomously, precise positioning and mapping are key technologies.

In Simultaneous Localization and Mapping (SLAM), the robot constructs a map of its environment using onboard navigation sensors while simultaneously estimating its own position. Common navigation sensors that perceive the environment and are used in SLAM systems are LiDAR, monocular RGB cameras, stereo cameras [10], or RGB-Depth (RGB-D) cameras [11]. Observations from these sensors are typically fused with angular rates and accelerations measured by an Inertial Measurement Unit (IMU). Examples for LiDAR-inertial SLAM systems can be found in [12–15], monocular visual-inertial SLAM systems in [16, 17], stereo visual-inertial SLAM systems in [18, 19], and RGB-D-inertial SLAM systems in [20, 21].

Early SLAM systems rely on Bayesian filters such as the Extended Kalman filter (EKF) [22] or the Particle filter [23]. However, all Bayesian filters are limited by the complete state assumption or Markov assumption, which assumes that our robot's current state is conditionally independent from past observations [24]. A promising approach for SLAM which has received a lot of attention in the last years is Factor Graph Optimization (FGO) [25]. FGO allows modelling the joint density more generally, i.e., it is not limited by the Markov assumption and can take past observations into account when estimating the current state and the map. Why this is especially useful when working with exteroceptive navigation sensors that perceive the environment can be illustrated with the following example: Imagine a situation where a mobile robot performs SLAM as it drives into a building. It simultaneously estimates its own position along with the positions of specific features of the map that it creates. When the robot is close to an area it has already mapped, it may observe the same features again. If it does so, it can use the previous observations as well as the current observations to these features to better estimate its own position and to update the map.

Indoor and subterranean environments are particularly challenging for navigation. A study which reviewed state-of-the-art SLAM algorithms for subterranean environments [26] found that all teams which participated in the DARPA Subterranean Challenge, an international robotics competition, rely on a LiDAR-focused SLAM based on FGO for positioning and mapping. LIO-SAM [27] is a widely used and openly available algorithm which fuses LiDAR point clouds and IMU data using FGO. Like other algorithms that fuse exteroceptive navigation sensors with IMUs, it has the drawback that when LiDAR fails, only the IMU data are used to compute the position of the robot. However, MEMS IMUs are not suitable for stand-alone positioning as errors quickly accumulate. Therefore, if the LiDAR data are not usable due to a sensor failure or environmental conditions such as smoke or fog, the positioning solution starts to drift.

In a recent paper [28] we proposed LIWO-SLAM, an extension of LIO-SAM, which fuses LiDAR, IMU and wheel odometry in a factor graph. We showed that with LIWO-SLAM, a better positioning accuracy can be achieved to when an EKF [29] is used with the same sensors. Moreover, adding wheel odometry helps to increase the redundancy of the positioning solution. However, a limitation of using wheel odometry is that it can only account for motion changes in a 2D plane. This is where this paper comes in: it proposes a combined IMU and wheel odometry factor which accounts for 3D motion and attitude changes.

The key innovation of this paper is the development of a 3D wheel-inertial-odometry model for ground-based robots. The model is integrated into a state-of-the-art LiDAR-focused SLAM based on factor graph optimization.

The main aims of the paper are to:

1. Develop the mathematical framework for a wheel-inertial-odometry model and integrate it into a LiDAR-focused SLAM;
2. Evaluate the model using two different datasets;
3. Analyse the advantages of the combined wheel-inertial odometry model.

The paper is structured as follows: First, the mathematical framework is described. The general formulation of the factor graph as well as the combined IMU and wheel odometry model are explained. Second, the software framework is described. Third, the developed algorithm is evaluated using two different datasets: a real-world dataset collected with a small mobile robot and a simulated dataset created in Gazebo. Both datasets contain a LiDAR outage where no point clouds are available. The performance of the proposed 3D wheel-inertial-odometry model during the outage is analysed in detail. Finally, we conclude the paper and give an outlook for further research.

2. Mathematical framework

The multi-sensor SLAM algorithm proposed in this paper is based on factor graph optimization. Factor graphs allow modelling complex estimation problems by expressing the joint density as a product of factors [25]. The factorization is shown in the form of a bipartite graph with two types of nodes, factor nodes and variable nodes. The variable nodes represent the states of the mobile robot, and the factor nodes represent the measurements. The edges of the factor graph connect the factor nodes to the variable nodes and represent independence relationships: a factor is only a function of the variables it is adjacent to.

The aim of factor graph optimization is to estimate the unknown robot states $\mathbf{X}=\{\mathbf{x}_i\}$ given our measurements $\mathbf{Z}$, i.e. to find the maximum a posteriori (MAP) estimate [25]

$$\mathbf{X}^{MAP} = \underset{\mathbf{X}}{\operatorname{argmax}}\, p(\mathbf{X}\,|\,\mathbf{Z}) = \\ = \underset{\mathbf{X}}{\operatorname{argmax}}\, \frac{p(\mathbf{Z}|\mathbf{X})p(\mathbf{X})}{p(\mathbf{Z})}\,. \tag{1}$$

Factor graph optimization uses all available sensor data to optimally estimate the state nodes. Since the functional relationships between the observations or factor nodes and the state nodes are non-linear, we use the Levenberg-Marquardt-Algorithm to solve the least-squares problem.

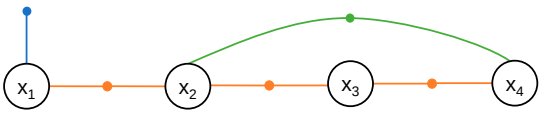

Fig. 1: Proposed factor graph for SLAM. The prior factor is shown in blue. The LiDAR factor (orange) connects consecutive state nodes. The loop closure factor is shown in green. Between two consecutive state nodes, the preintegration with the 3D wheel-inertial odometry model is used to compute the robot's state and provides an initial guess for LiDAR scan matching.

In our factor graph (Figure 1), a state node containing the robot's state $\mathbf{x}_i$ at epoch i is written as

$$\mathbf{x}_i = [\mathbf{R}_i, \mathbf{p}_i^m, \mathbf{v}_i^m], \tag{2}$$

with $\mathbf{R}_i \in SO(3)$ being the rotation matrix from the body frame to the map frame, $\mathbf{p}_i^m$ is the 3D position vector in the map frame, and $\mathbf{v}_i^m$ is the 3D velocity vector in the map frame. A new state node is added to the factor graph whenever a new LiDAR keyframe is introduced, i.e., when the position and attitude of the robot change more than a certain threshold.

The homogeneous transformation matrix, which describes the transformation from the origin of the map frame to the current pose of the robot, can be written as

$$\mathbf{T}_i = \begin{pmatrix} \mathbf{R}_i & \mathbf{p}_i^m \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix}. \tag{3}$$

The LiDAR odometry factor (shown in orange in Figure 1) links two consecutive state nodes, i.e., it contains information on how the robot's state changed from one epoch to the next. Two consecutive state nodes $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ are linked via the transformation

$$\Delta\mathbf{T}_{i,i+1} = \mathbf{T}_i^{-1}\mathbf{T}_{i+1}. \tag{4}$$

A loop closure factor (shown in green in Figure 1) is used when a feature is detected in a LiDAR scan which has already previously been mapped. In this case, the factor connects the two state nodes from which the same feature was observed.

The LiDAR odometry and loop closure factors are based on LIO-SAM [27]. For each new LiDAR scan, edge and planar features are extracted. When a new state node is added to the graph, the algorithm looks for other state nodes that are close to this state node and tries to match the keyframes using scan matching. If the matching is successful, the obtained relative transformation is added to the factor graph as a loop closure factor.

Computing the robot's state in between the keyframes or state nodes is referred to as *preintegration*. The preintegrated navigation solution is used as an initial guess for LiDAR scan matching. When no LiDAR data are available, the navigation solution is computed only from the preintegration.

Conventional algorithms use only IMU data in the preintegration. We propose a novel and more

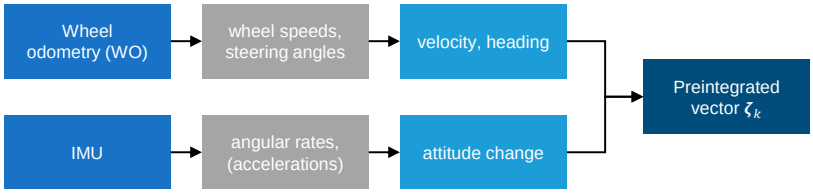

Fig. 2: Overview of the combined wheel odometry and IMU preintegration

robust approach which combines both wheel odometry and IMU data. In the following, this novel method will be explained.

2.1 Preintegration with a combined wheel odometry and IMU model

The task of the preintegration is to predict the navigation state from one state node of the factor graph $(\mathbf{x}_i = [\mathbf{R}_i,\ \mathbf{p}_i^m,\ \mathbf{v}_i^m])$ to the next state node $(\mathbf{x}_{i+1})$. To do so, observations from a sensor with a high data rate are used. IMU preintegration as proposed by [30] has become a standard and most systems use a low-cost MEMS IMU. However, when the main visual sensor fails, the preintegration with the MEMS IMU cannot be used as a standalone solution as errors quickly accumulate.

A method which is more robust and more suitable to bridge outages is wheel odometry [31]. Wheel odometry uses measured wheel speeds and steering angles of the wheels to compute how the robot's pose changed. However, wheel odometry only yields 2D information. Therefore, we propose a method that combines both observations from the IMU and from wheel odometry. An overview of this process is given in Figure 2. Wheel odometry is used to compute the 2D velocity and heading. The angular rates measured by the IMU are used to compute an attitude change. Both are combined to obtain the preintegrated navigation state.

2.1.1 Preintegrated navigation state in the tangent space

To predict the navigation state from one state node $(\mathbf{x}_i = [\mathbf{R}_i,\ \mathbf{p}_i^m,\ \mathbf{v}_i^m])$ to the next $(\mathbf{x}_{i+1})$, the IMU and wheel odometry observations are accumulated and integrated between the keyframes. In the following, the indices i will refer to the state nodes $\mathbf{x}_i = [\mathbf{R}_i,\ \mathbf{p}_i^m,\ \mathbf{v}_i^m]$ and the indices k refer to the preintegrated vector $\boldsymbol{\zeta_k} = [\boldsymbol{\theta}_k,\ \mathbf{p}_k,\ \mathbf{v}_k]$ in the tangent space.

Since the navigation state $\mathbf{x}_i$ contains a rotation matrix $\mathbf{R}_i$ and rotation matrices are manifolds, we lift the integration to the local tangent space and then perform a retraction back to the manifold.

The tangent space of a rotation matrix $\mathbf{R} \in SO(3)$ is its Lie algebra $\mathfrak{so}(3)$, the space of 3×3 skew-symmetric matrices. The logarithm map $\log(\mathbf{R})$ allows to compute the skew-symmetric matrix $\mathbf{S}$. Every skew-symmetric matrix $\mathbf{S}$ can be associated to a vector $\boldsymbol{\theta}$ via the hat-operator $\hat{}$ [30]

$$\boldsymbol{\theta}^{\wedge} = \begin{bmatrix}\theta_x\\ \theta_y\\ \theta_z\end{bmatrix}^{\wedge} = \begin{bmatrix}0 & -\theta_z & \theta_y\\ \theta_z & 0 & -\theta_x\\ -\theta_y & \theta_x & 0\end{bmatrix} = \mathbf{S} \in \mathfrak{so}(3). \quad (5)$$

The vector $\boldsymbol{\theta}$ points in the direction of the 3D rotation axis and its norm corresponds to the rotation angle. This 3D vector is used to describe attitude changes in the tangent space.

The 9D vector which describes the preintegrated navigation state in the tangent space is as follows:

$$\boldsymbol{\zeta_k} = [\boldsymbol{\theta}_k,\ \mathbf{p}_k,\ \mathbf{v}_k]^{\mathrm{T}} = \quad (6)$$
$$= [\theta_{x,k},\ \theta_{y,k},\ \theta_{z,k},\ p_{x,k},\ p_{y,k},\ p_{z,k},\ v_{x,k},\ v_{y,k},\ v_{z,k}]^{\mathrm{T}},$$

with $\mathbf{p}_k$ being the position and $\mathbf{v}_k$ the velocity at epoch k.

2.1.2 Wheel-inertial odometry model

To compute the preintegration, we use the gyroscope measurements of the IMU $\omega_{x,k}$, $\omega_{y,k}$ at epoch k, as well as the observations from wheel odometry. Which wheel odometry model to choose always depends on how the robot is steered. In our previous work [28], we proposed a four wheel independent steering and four wheel independent driving (4WIS4WID) wheel odometry model, which assumes that each wheel of the robot can be steered individually. The 4WIS4WID model uses measured wheel speeds v_{fl}, v_{fr}, v_{rl}, v_{rr} and steering angles δ_{fl}, δ_{fr}, δ_{rl}, δ_{rr} to compute how the position and attitude of the robot change.

For skid-steered robots, which will later be used to evaluate the algorithm in Chapter 4, all steering angles are zero. Furthermore, all wheels on one side of the robot are steered with the same velocity, i.e., all left wheels share the same wheel

speed v_l and all right wheels share the same wheel speed v_r. The wheel odometry equations therefore simplify to the following: The rotation rate $\dot{\theta}_{z,k}$ about the robot's z-axis at epoch k is obtained from

$$\dot{\theta}_{z,k} = \frac{v_{l,k} - v_{r,k}}{2b}, \tag{7}$$

where $v_{l,k}$ is the measured speed of the left wheels, $v_{r,k}$ is the measured speed of the right wheels and $2b$ is trackwidth. The velocity in the body frame of the robot can be computed from

$$v^b_{x,k} = \frac{v_{l,k} + v_{r,k}}{2}, \tag{8}$$

$$v^b_{y,k} = 0. \tag{9}$$

In the first step of the combined wheel odometry and IMU preintegration, the vector $\boldsymbol{\theta}_k$ is computed. For the x- and y-components, the method proposed by [30] is used: the gyroscope measurements $\omega_{x,k}$, $\omega_{y,k}$, which are corrected for their respective biases b_x, b_y, are integrated numerically from epoch $k-1$ to epoch k

$$\theta_{x,k} = \theta_{x,k-1} + (\omega_{x,k} - b_x) \cdot \Delta t_{k-1,k}, \tag{10}$$

$$\theta_{y,k} = \theta_{y,k-1} + (\omega_{y,k} - b_y) \cdot \Delta t_{k-1,k}. \tag{11}$$

For the z-component, the rotation rate obtained from wheel odometry (Equation 7) is integrated numerically

$$\theta_{z,k} = \theta_{z,k-1} + \dot{\theta}_{z,k} \cdot \Delta t_{k-1,k}. \tag{12}$$

In the next step, the retraction (exponential map) of $\boldsymbol{\theta}_k$ is computed to obtain the rotation matrix $\mathbf{R}_k$.

Next, Equation 8 from the wheel odometry model is used to compute the robots forward speed $v^b_{x,k}$ in the body frame. To obtain the 3D velocity vector $\mathbf{v}_k$, the velocity obtained from wheel odometry is rotated to the navigation frame with

$$\mathbf{v_k} = \mathbf{R_k} \begin{pmatrix} v^b_{x,k} \\ 0 \\ 0 \end{pmatrix} \tag{13}$$

Finally, the velocity vector $\mathbf{v_k}$ is integrated numerically to obtain the position vector $\mathbf{p_k}$

$$\mathbf{p_k} = \mathbf{p_{k-1}} + \mathbf{v_k} \cdot \Delta t_{k-1,k}. \tag{14}$$

3. Software Development

The software was developed in C++ using the Robot Operating System (ROS) framework. It builds on the previous work LIWO-SLAM [28] and on LIO-SAM [27] and uses the GTSAM library [32] for smoothing and mapping.

An overview of the program flow is given in Figure 3. Four ROS nodes are used that perform different tasks. The *3D Wheel Odometry Preintegration-Node* fuses data from the IMU and the wheel encoders to calculate the preintegration (preintegration_incremental), a three-dimensional wheel odometry increment between two LiDAR odometry poses. This increment is used in the *Cloud Deskewing*-node to deskew the point cloud of the LiDAR. Both the preintegration and the deskewed point cloud are passed on to the *Map Optimization*-node, where the actual factor graph is optimized. The *Mapping Odometry Fusion*-node combines the output of the *Map Optimization* with the preintegration to output a navigation solution with the same data rate as the preintegration.

4. Evaluation

To evaluate the developed algorithm, two different datasets were collected: a real dataset with a small mobile robot (scenario 1) and a simulated

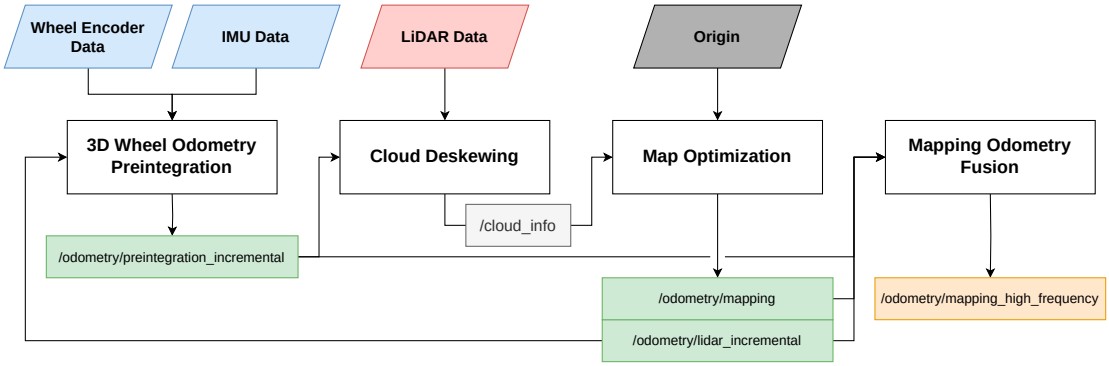

Fig. 3: Flow-graph of the proposed SLAM algorithm

dataset obtained in Gazebo (scenario 2). In the following, the collection of the datasets is explained, and the results obtained by processing these datasets with the newly developed algorithm are presented.

4.1 Scenario 1: Mapping the campus of TU Graz

In the first scenario, a small mobile robot was used to map an outdoor environment. During this scenario, connection issues between the LiDAR and the main PC of the robot occurred and no LiDAR data were available for approximately 30 seconds. In the following, first the collection of the dataset will be described. Then, the dataset will be evaluated with both a conventional LiDAR-inertial factor graph and the newly proposed method with the wheel-inertial preintegration. A special focus is put on the performance of both algorithms during the LiDAR outage.

4.1.1 Collecting the dataset

Data was collected around the campus "Neue Technik" of Graz University of Technology on the 20th of October 2023. The robot used in this scenario is a Jackal by Clearpath Robotics™ (see Figure 4).

The robot was equipped with an IMU, a LiDAR, and wheel encoders. An overview of the specific navigation sensors used is given in Table 1. Data from the navigation sensors was recorded with ROS and stored in .bag files. ROS bag files have the advantage that they can be replayed at a later stage to simulate a real-time scenario. By replaying the same bag file several times, different algorithms can be tested with the same navigation data.

The collected trajectory in the first scenario has a duration of 12 minutes and the distance travelled was 820 meters. The robot started and stopped at the same location (see Figure 5).

During data recording, the connection between the LiDAR and the main PC of the robot was interrupted for approximately 30 seconds. This outage of the LiDAR data will be analysed in more detail in the evaluation.

Type	Description
IMU	XSens MTi-G-710
LiDAR	Velodyne ULTRA Puck VLP-32C
Wheel Encoders	78,000 pulses/m QUADRATURE

Tab. 1: *Navigation sensors used in the first scenario*

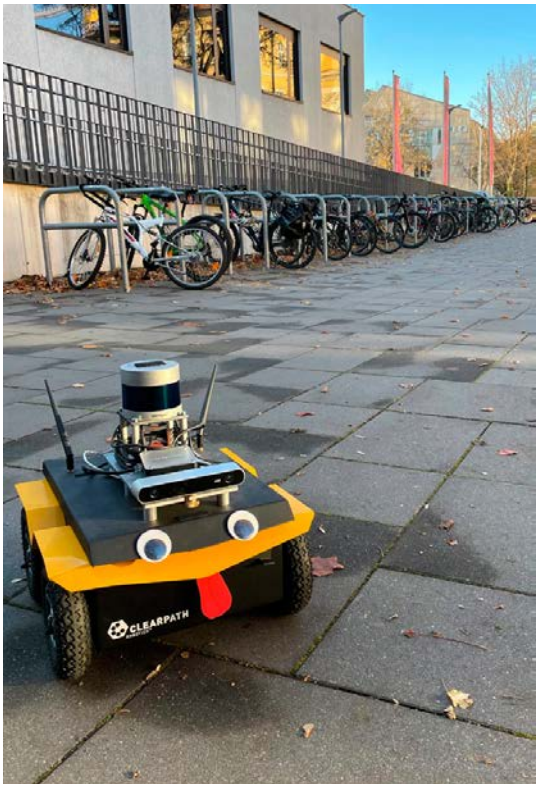

Fig. 4: *Jackal robot used in the first scenario. The picture was taken at the location of the LiDAR outage.*

Fig. 5: *The drain cover (indicated in red) was the start and stop point of the trajectory of scenario 1*

4.1.2 Evaluation of the first scenario

To evaluate the developed algorithm with the dataset collected in the first scenario, the ROS bag file was replayed and processed with two different algorithms: a conventional LiDAR-inertial factor graph (IMU preintegration), and the newly proposed factor graph which fuses LiDAR, wheel odometry and inertial data (WO/IMU-preintegration).

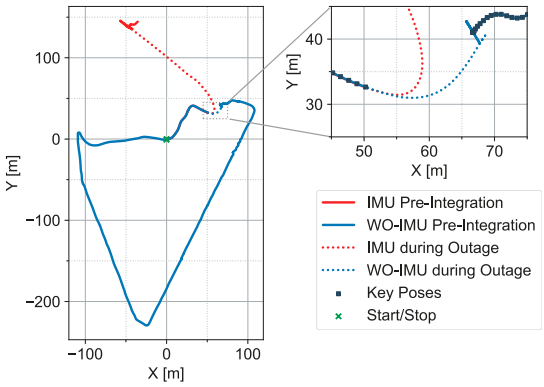

Fig. 6: 2D plot of the trajectory of the first scenario. Left: overview of the whole trajectory. Right: zoomed-in section where the LiDAR outage occurred.

Figure 6 shows the 2D horizontal plot of the trajectory which was recorded in the first scenario. The start and end points of the trajectory are marked as green crosses. The trajectory obtained from the factor graph with IMU preintegration is shown in red, the factor graph with the combined wheel odometry and IMU preintegration is shown in blue. During the LiDAR outage that occurred, both trajectories are depicted as dotted lines. The zoomed-in plot also shows the keyframes of the SLAM solution. These keyframes are computed every time the factor graph is optimized.

During the outage of the LiDAR data, which lasted around 30 seconds and during which the robot drove approximately 20 meters, no keyframes were computed as these are derived from the laser scans. During the LiDAR outage, the IMU errors accumulate quickly, and the trajectory obtained from IMU preintegration starts to drift. After 30 seconds without LiDAR data, the position computed only from IMU data is so far off that the LiDAR scan after the outage cannot be registered to the map anymore.

With the combined wheel odometry and IMU preintegration model, however, the outage can be bridged. The position error after 30 seconds without LiDAR data is small enough that the scan after the outage can be registered to the map.

Figure 7 shows how the LiDAR outage is bridged in more detail. These four screenshots of the scan matching algorithm show the global

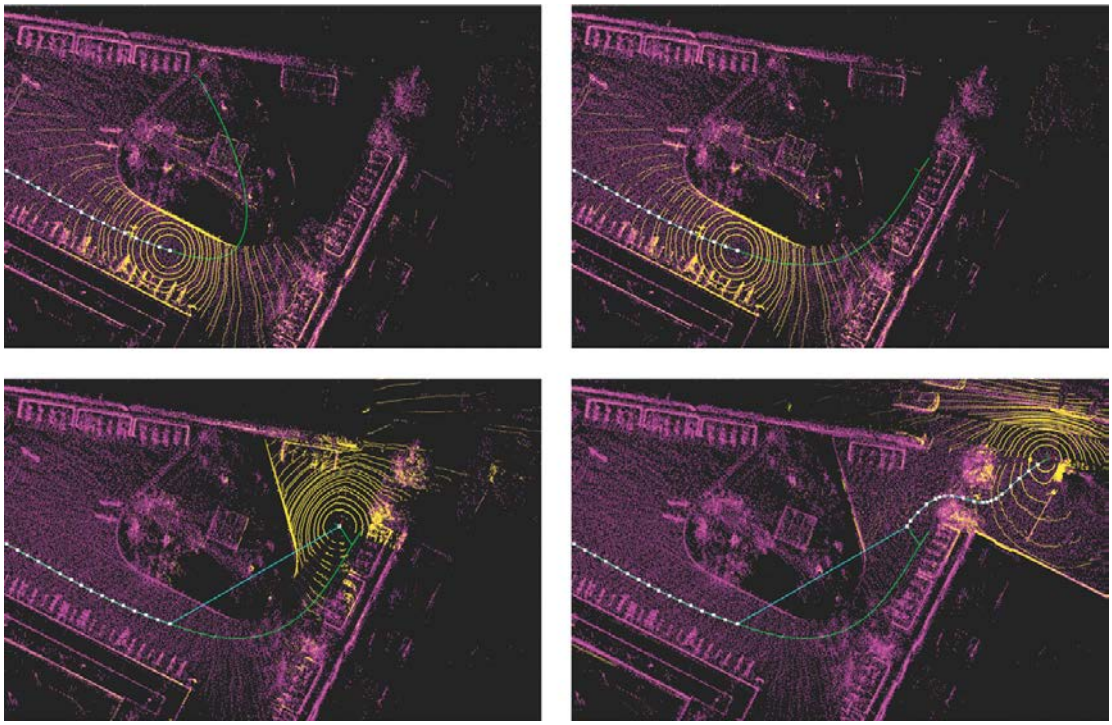

Fig. 7: Comparison of scan matching output at the LiDAR outage. Top left: IMU Pre-Integration failing to bridge the LiDAR outage. Top right: WO-IMU Pre-Integration bridges the LiDAR outage. Lower left: The scan matching is successful after bridging the outage with WO-IMU Pre-Integration. Lower right: Robot moving on after recovery of scan matching.

matched point cloud in purple and the current scan in yellow. The global path after factor graph optimization is shown in turquoise and the key poses are shown as white squares. The top left image shows the result of the preintegration (green) with the IMU, which drifts during the outage. The top right image depicts how the combined wheel odometry and IMU preintegration model bridges the outage. The lower left image shows how the first scan after the LiDAR outage is registered to the map and the lower right image shows how the SLAM algorithm continues after the outage.

The difference between the start and end point of the whole trajectory was 89 cm (when no loop closure is used at the end of the trajectory). With a trajectory length of 820 metres, this corresponds to an error of 0.1% of the distance travelled.

4.2 Scenario 2: Mapping a parking garage in a simulation environment

The purpose of the second scenario was to emphasize the importance of the combined wheel odometry and IMU model. Therefore, a scenario was simulated where a robot drives up a ramp in a parking garage and a LiDAR outage occurs. In the following, the simulation and the evaluation of this scenario will be described.

4.2.1 Simulating the dataset in Gazebo

The second scenario, shown in Figure 8, is based on a simulation of a parking garage with multiple levels in Gazebo. Gazebo is the simulation environment for the Robot Operation System (ROS). The sensors used in this study are simulated in Gazebo and are of the same type as in the previous scenario shown in Table 1. The noise characteristics of each sensor type were considered in the simulation to be as close to the real world as possible.

Fig. 8: RTE robot by Rosenbauer in the simulation environment created with Gazebo

A trajectory was recorded where the robot traversed between two levels of a parking garage to also show the change in the Z component of the location of the robot. Before the robot reached the ramp, an artificial outage of the LiDAR data was simulated.

4.2.2 Evaluation of the second scenario

The recorded data was processed with three different preintegration models: IMU preintegration, a 2D wheel odometry preintegration (WO), and the combined wheel odometry and IMU preintegration (WO-IMU).

Figure 9 shows the 2D plot of the second scenario. Since a simulation environment was used, the ground truth is known and shown in blue. Up to the simulated LiDAR outage, marked with the black triangle, all solutions yield comparable results. This is due to the fact that LIWO-SLAM

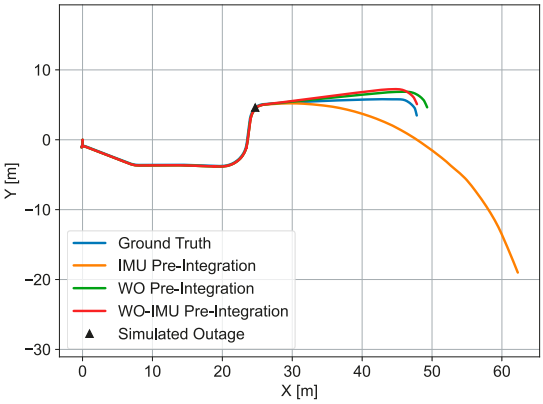

Fig. 9: 2D plot of scenario 2. The black triangle indicates the start of the LiDAR outage.

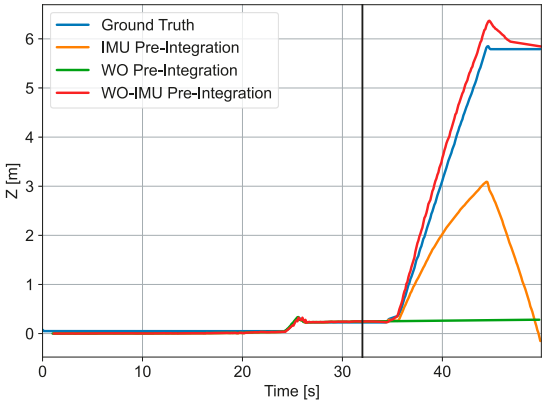

Fig. 10: Height time series of scenario 2. The start of the LiDAR outage is indicated by a black line.

strongly relies on the precision and accuracy of the LiDAR when it is available.

As in the previous scenario, the IMU-only preintegration (orange trajectory) starts to drift away shortly after the outage. The WO-only solution (green trajectory) shows comparable results to the combined WO-IMU solution (red trajectory) in the 2D case, as the distance travelled can be determined quite accurately with wheel odometry. Since the WO-only solution has no information about changes in the height component, the trajectory shows a longer horizontal distance travelled than the WO-IMU model, which takes height changes into account.

Figure 10 shows the height coordinate time series of the trajectories computed with the three different preintegration models compared to the ground truth (blue). The start of the LiDAR outage is indicated by a black vertical line. The wheel odometry preintegration (green) has no information about the changing height, as it always assumes planar movement. The IMU-only preintegration (orange) can bridge the outage for around 2 seconds, then the accumulation of the errors starts to increase, and the solution begins to drift away from the ground truth. Our proposed solution, the combined WO-IMU preintegration, can bridge the outage over the whole distance travelled up the ramp to the second floor.

5. Conclusion and outlook

This paper proposed a novel approach for fusing LiDAR, IMU and wheel odometry data in a factor graph for SLAM. In contrast to conventional algorithms which use only IMU data in the preintegration, a combined wheel odometry and IMU model was proposed. In the following, the main findings of this work are summarized.

The first aim was to develop the mathematical framework for a wheel-inertial-odometry model and integrate it into a LiDAR-focused factor graph for SLAM. The combined wheel odometry and IMU model uses measured wheel speeds from wheel odometry and angular rates measured by the IMU to compute how the robot's pose changes. In a LiDAR-focused factor graph, state nodes are added to the graph whenever a new LiDAR keyframe is introduced. Between the keyframes, the navigation solution is computed using a preintegration method, in our case with the combined wheel odometry and IMU model.

The second aim was to evaluate the model using two different datasets. Therefore, two different datasets were generated. The first scenario was a 820 m long trajectory collected with a small robot and the second scenario was obtained from a simulation in Gazebo, where a tracked robot was simulated. Both datasets contain a LiDAR outage. In both scenarios, the conventional IMU-based preintegration fails during the LiDAR outage as errors accumulate fast. With the proposed preintegration method that combines both wheel odometry and IMU, the LiDAR outage can be bridged. In the first scenario, the proposed preintegration method bridges a 30 second LiDAR outage with sufficient accuracy that the first scan after the outage can be registered to the map successfully.

The third aim of the paper was to analyse the advantages of the combined wheel-inertial odometry model compared to using only inertial or only wheel odometry data. To demonstrate the advantages of the combined model, a LiDAR outage was simulated in the second scenario where the robot moved up a ramp in the parking garage. As in the first scenario, the IMU-only preintegration method shows a large position error after a few seconds. The preintegration method that uses only wheel odometry shows a lower horizontal positioning error, however, it cannot account for the height change. The combined model combines the advantages of both sensors: it provides an accurate 3D position estimate during the LiDAR outage.

The developed algorithm will be used in the research project SURUx2, where a semi-autonomous robot supports military specialists in decontamination. In the future, the algorithm will be extended by adding more sensors that can perceive the environment to the factor graph. This allows to not only increase localization robustness but to also allow for continuous mapping when one sensor fails.

Acknowledgements

This research was conducted within the project SURUx2 (895151), which is funded by the Austrian defense research programme FORTE of the Federal Ministry of Finance (BMF). The authors would like to thank the project partners of the SURUx2 project at the AIT Austrian Institute of Technology GmbH - Center for Digital Safety and Security, Rosenbauer International AG, CBRN Protection GmbH and the Austrian Ministry of Defence.

References

[1] *H. Lee and J. Jeong, "Mobile Robot Path Optimization Technique Based on Reinforcement Learning Algorithm in Warehouse Environment," Applied Sciences, vol. 11, no. 3, p. 1209, 2021.*

[2] *R. Bernardo, J. M. Sousa and P. J. Gonçalves, "Survey on robotic systems for internal logistics," Journal of Manufacturing Systems, vol. 65, pp. 339–350, 2022.*

[3] *A. Yildirim, H. Reefke and E. Aktaş, Mobile Robot Automation in Warehouses*: A Framework for Decision Making and Integration, 1st ed., Cham, Springer International Publishing; Imprint: Palgrave Macmillan, 2023.

[4] *E. Reitbauer, Multi-Sensor Positioning for the Automatic Steering of Tracked Agricultural Vehicles, Verlag der Technischen Universität Graz, 2022.*

[5] *E. Reitbauer, C. Schmied and M. Wieser, "Autonomous Navigation Module for Tracked Compost Turners," in 2020 European Navigation Conference (ENC), Dresden, Germany, 2020, pp. 1–10.*

[6] *S. Saha et al., "A Vision-based Navigation System for an Agricultural Autonomous Tractor," IFAC-PapersOnLine, vol. 55, no. 32, pp. 48–53, 2022.*

[7] *A. Ziebinski et al., "A Survey of ADAS Technologies for the Future Perspective of Sensor Fusion," in Computational Collective Intelligence, vol. 9876, N. T. Nguyen et al., Eds. Cham*: Springer International Publishing, 2016, pp. 135–146.

[8] *I. Yaqoob et al., "Autonomous Driving Cars in Smart Cities*: Recent Advances, Requirements, and Challenges," IEEE Network, vol. 34, no. 1, pp. 174–181, 2020.

[9] *F. E. Schneider and D. Wildermuth, "Using robots for firefighters and first responders*: Scenario specification and exemplary system description," in 2017 18th International Carpathian Control Conference (ICCC), Sinaia, Romania, 2017, pp. 216–221.

[10] *M. Filipenko and I. Afanasyev, "Comparison of Various SLAM Systems for Mobile Robot in an Indoor Environment," in 2018 International Conference on Intelligent Systems (IS), Funchal - Madeira, Portugal, 2018, pp. 400–407.*

[11] *Y. Li and C. Shi, "Localization and Navigation for Indoor Mobile Robot Based on ROS," in 2018 Chinese Automation Congress (CAC), Xi'an, China, 2018, pp. 1135–1139.*

[12] *P. Geneva et al., "LIPS*: LiDAR-Inertial 3D Plane SLAM," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, 2018, pp. 123–130.

[13] *H. Li et al., "An Intensity-Augmented LiDAR-Inertial SLAM for Solid-State LiDARs in Degenerated Environments," IEEE Trans. Instrum. Meas., vol. 71, pp. 1–10, 2022.*

[14] *L. Yang et al., "A Tightly Coupled LiDAR-Inertial SLAM for Perceptually Degraded Scenes," Sensors (Basel, Switzerland), vol. 22, no. 8, 2022.*

[15] *K. Liu and H. Ou, "A Light-Weight LiDAR-Inertial SLAM System with High Efficiency and Loop Closure Detection Capacity," in 2022 International Conference on Advanced Robotics and Mechatronics (ICARM), Guilin, China, 2022, pp. 284–289.*

[16] *A. Concha et al., "Visual-inertial direct SLAM," in 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016, pp. 1331–1338.*

[17] *R. Mur-Artal and J. D. Tardos, "Visual-Inertial Monocular SLAM With Map Reuse," IEEE Robot. Autom. Lett., vol. 2, no. 2, pp. 796–803, 2017.*

[18] *H. Yin et al., "Dynam-SLAM*: An Accurate, Robust Stereo Visual-Inertial SLAM Method in Dynamic Environments," IEEE Trans. Robot., vol. 39, no. 1, pp. 289–308, 2023.

[19] *V. Usenko et al., "Direct visual-inertial odometry with stereo cameras," in 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016, pp. 1885–1892.*

[20] *X. Deng et al., "Robust 3D-SLAM With Tight RGB-D-Inertial Fusion," in 2019 Chinese Control Conference (CCC), Guangzhou, China, 2019, pp. 4389–4396.*

[21] *U. Qayyum, Q. Ahsan and Z. Mahmood, "IMU aided RGB-D SLAM," in 2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, Pakistan, 2017, pp. 337–341.*

[22] *J. Weingarten and R. Siegwart, "EKF-based 3D SLAM for structured environment reconstruction," in 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Alta., Canada, 2005, pp. 3834–3839.*

[23] *G. Grisetti, C. Stachniss and W. Burgard, "Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling," in Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005, pp. 2432–2437.*

[24] *S. Thrun, W. Burgard and D. Fox, Probabilistic Robotics, The MIT Press, 2005.*

[25] *F. Dellaert and M. Kaess, "Factor Graphs for Robot Perception," FNT in Robotics, vol. 6, 1-2, pp. 1–139, 2017.*

[26] *K. Ebadi et al., "Present and Future of SLAM in Extreme Underground Environments," Aug. 2022, Available*: http://arxiv.org/pdf/2208.01787v1.

[27] *T. Shan et al., "LIO-SAM*: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 10242020, pp. 5135–5142.

[28] *E. Reitbauer et al., "LIWO-SLAM*: A LiDAR, IMU, and Wheel Odometry Simultaneous Localization and Mapping System for GNSS-Denied Environments Based on Factor Graph Optimization," in Proceedings of the 36th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2023), Denver, Colorado, 2023, pp. 1669–1683.

[29] *E. Reitbauer, C. Schmied and H. Didari, "Subterranean positioning for a semi-autonomous robot supporting emergency task forces," in 2022 International Conference on Localization and GNSS (ICL-GNSS), Tampere, Finland, 672022, pp. 1–7.*

[30] *C. Forster et al., "On-Manifold Preintegration for Real-Time Visual--Inertial Odometry," IEEE Trans. Robot., vol. 33, no. 1, pp. 1–21, 2017.*

[31] *E. Reitbauer and C. Schmied, "Bridging GNSS Outages with IMU and Odometry*: A Case Study for Agricultural Vehicles," Sensors (Basel, Switzerland), vol. 21, no. 13, 2021.

[32] *Frank Dellaert and GTSAM Contributors, borglab/gt sam, Georgia Tech Borg Lab, Available*: https://github.com/borglab/gtsam).

Contacts

Ass.Prof. Dipl.-Ing. Dr. techn. Eva Reitbauer, BSc, Institut für Geodäsie, Technische Universität Graz, Steyrergasse 30/II, 8010 Graz.
E-Mail: eva.reitbauer@tugraz.at

Dipl.-Ing. Christoph Schmied, BSc, Institut für Geodäsie, Technische Universität Graz, Steyrergasse 30/II, 8010 Graz.
E-Mail: schmied@tugraz.at

Dipl.-Ing. Fabian Theurl, BSc, Institut für Geodäsie, Technische Universität Graz, Steyrergasse 30/II, 8010 Graz.
E-Mail: fabian.theurl@tugraz.at