

Article

Stretch-ICP: A Continuous-Trajectory Registration and Deskewing Algorithm in Scenarios of Aggressive Motions [†]

Simon-Pierre Deschênes , Veronica Vannini , Philippe Giguère  and François Pomerleau * 

Northern Robotics Laboratory, Université Laval, Quebec City, QC G1V 0A6, Canada;
simon-pierre.deschenes@norlab.ulaval.ca (S.-P.D.); veronica.vannini@norlab.ulaval.ca (V.V.);
philippe.giguere@ift.ulaval.ca (P.G.)

* Correspondence: francois.pomerleau@norlab.ulaval.ca

[†] This paper is an extended version of our paper published in the Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024.

Abstract

Robust robotic autonomy remains challenging in complex environments, where loss of stability on uneven or slippery terrain can induce extreme accelerations and angular velocities. Such motions corrupt sensor measurements and degrade state estimation, motivating the need for improved algorithmic robustness. To investigate this issue, we introduce the Tumbling-Induced Gyroscope Saturation (TIGS) dataset, which consists of recordings from a mechanical lidar and an Inertial Measurement Unit (IMU) tumbling down a hill. The dataset contains angular speeds up to four times higher than those in similar datasets and is publicly available. We then propose two complementary methods to improve Simultaneous Localization And Mapping (SLAM) robustness and evaluate them on TIGS. First, Saturation-Aware Angular Velocity Estimation (SAAVE) estimates angular velocities when gyroscope measurements become saturated during aggressive motions, reducing angular speed estimation error by 83.4%. Second, Stretch-ICP, a novel registration and deskewing algorithm, enables reconstruction of smoother 6-Degrees Of Freedom (DOF) trajectories under aggressive motions compared to *classical* Iterative Closest Point (ICP). Stretch-ICP reduces linear and angular velocity errors by 95.2% and 94.8%, respectively, at scan boundaries. Together, these contributions improve the robustness and consistency of lidar-inertial state estimation under aggressive motions.

Keywords: continuous trajectory; aggressive motions; SLAM; Stretch-ICP; SLAM robustness

1. Introduction

Mobile robots are often deployed in remote or hazardous environments where physical disturbances and recovery from failures are difficult [1]. Although recent hardware improvements have reduced the likelihood of mechanical failure due to collisions [2], state estimation for robot localization often remains vulnerable to falls, drops, and impacts [3]. Inspired by work in control [4], we refer to such events as *aggressive motions* for perception, defined as motion regimes characterized by abrupt changes, impacts, and loss of predictability that push the sensing and estimation pipeline close to or beyond its operational limits. With this definition, highway navigation would not cause aggressive motions despite high linear velocities, as the motion remains smooth and predictable. Similarly, sustained high angular velocities alone do not necessarily constitute aggressive motions when they occur without impacts or abrupt changes. On the other hand, a robot tumbling down a steep hill



Academic Editor: Zheng Chen

Received: 15 February 2026

Revised: 1 April 2026

Accepted: 15 April 2026

Published: 21 April 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

exemplifies aggressive motion, as it involves repeated collisions, rapid angular accelerations, and loss of motion predictability. In our context, these aggressive motions will greatly affect perception, causing a large skew in lidar scans [5] and saturation in gyroscope measurements [6]. Deskewing algorithms correct these distortions using an estimate of the intra-scan lidar motion. However, in many modern SLAM systems, the rotational motion prior to estimating this intra-scan trajectory is obtained by integrating IMU measurements [7–10], making the deskewing process directly dependent on gyroscope measurement quality.

Gyroscope sensitivity and performance generally degrade under realistic operation conditions such as temperature variations, mechanical vibration, and environmental stress, causing scale-factor drift and bias instability, even with a sensor's nominal rated range. These effects are especially pronounced in Micro-Electro-Mechanical Systems (MEMS) gyroscopes [11]. Although many commercial IMUs can report angular rates up to 35 rad/s or more, this specification alone does not guarantee high-quality measurements at those speeds. In MEMS and consumer-grade gyroscopes, increasing the full-scale measurement range directly reduces resolution due to coarser quantization. As sensitivity decreases, each digital count corresponds to a larger angular increment, amplifying quantization error and measurement noise [12,13]. Consequently, although a wider range may prevent saturation at high angular rates, it degrades the accuracy of state estimation in practice [14]. As a result, even if an IMU can nominally avoid saturation at high angular velocities, its practical utility for accurate state estimation degrades as range increases [14].

Therefore, gyroscope saturation will lead to inaccurate motion priors and deskewing, and thus to a potential SLAM failure [15]. Even when SLAM remains operational, erroneous intra-scan trajectories increase the correction required during registration, which introduces discontinuities in the reconstructed trajectory [16]. These discontinuities can be detrimental to other algorithms on a mobile robot, such as control algorithms, as they introduce incorrect velocity and acceleration estimates that may lead to instability and reduced system performance [17]. An example of a discontinuous trajectory produced by a SLAM algorithm using ICP as registration method is depicted in purple in Figure 1. This trajectory was obtained using a manually actuated sensor rig, with operators inducing linear and angular displacements across multiple directions. In this paper, we consider the reconstructed trajectory as including the full intra-scan motion within each lidar scan, rather than representing each scan by a single pose. When examining trajectories at this resolution, we observe that classical registration methods such as ICP break trajectory continuity by applying rigid transformations to register lidar scans. While such corrections can improve global pose consistency, they may do so at the expense of local trajectory smoothness and physically consistent motion estimates. To address this limitation, we propose a registration and deskewing approach that explicitly enforces trajectory continuity across scan boundaries, yielding physically consistent motion estimates even under aggressive motions, as illustrated by the green trajectory in Figure 1.

While a robot tumbling down a hill is not representative of nominal operation in most robotic platforms, it provides a controlled stress test for perception under impacts, abrupt angular accelerations, and loss of motion predictability. Similar estimation failures can arise, in less extreme but practically relevant situations, during slips, hard landings, collision recovery, aggressive traversal of uneven terrain, or emergency maneuvers. Our goal is therefore not to optimize for tumbling as an application in itself but to improve the robustness of lidar-inertial state estimation when robots are pushed beyond smooth-motion assumptions.

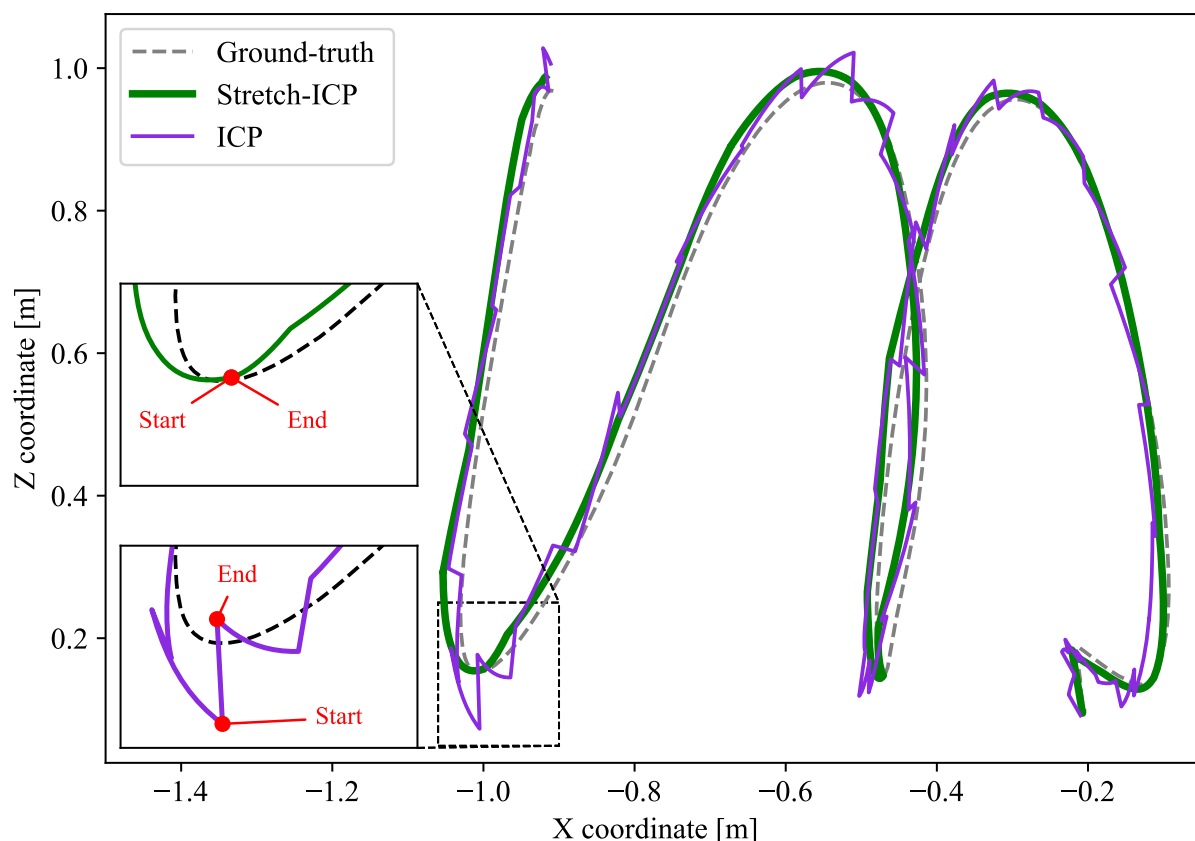


Figure 1. Example of reconstructed trajectories obtained with a SLAM framework using ICP (purple) and Stretch-ICP (green) as registration algorithms. For clarity, the 3D trajectories are projected onto the X-Z plane, and the dotted line indicates the ground-truth trajectory. The black arrows indicate the direction of motion through time. The zoomed-in views on the left show a scan boundary, where ‘End’ marks the end of one intra-scan trajectory and ‘Start’ marks the beginning of the next.

In our previous work [18], we introduced SAAVE, a method for estimating angular velocities during periods of gyroscope saturation. To address the limitations of existing robots and datasets, we also developed a collision-resistant lidar–inertial rig capable of sustaining large accelerations and angular velocities. Using this setup, we collected the TIGS dataset by tumbling the rig down a steep hill, capturing aggressive, collision-rich motions that lead to gyroscope saturation. By incorporating SAAVE into a lidar–inertial SLAM framework, we demonstrated improved robustness to aggressive motions on the TIGS dataset. However, the evaluation in that work was primarily limited to angular velocity recovery and final pose error, and did not allow analysis of trajectory continuity, scan-boundary artifacts, or velocity estimation accuracy along the full trajectory.

Building on this prior work, this journal extension expands the scope from sensor-level robustness to trajectory-level consistency. We introduce Stretch-ICP, a registration and deskewing algorithm that enforces continuity across scans, and present the High-Resolution Motion Capture (HRMC) dataset, providing high-frequency 6-DOF ground truth (i.e., 200 Hz). Together, these additions enable an evaluation of trajectory continuity, linear and angular velocity errors, and the trade-offs between motion consistency and localization accuracy under aggressive motions, which were not accessible in our previous work. In addition, we broaden the experimental comparison by evaluating SAAVE against Point-LIO [19], a lidar–inertial method designed to remain robust under gyroscope saturation, providing a stronger baseline for assessing robustness under aggressive motions.

In short, the contributions of our prior conference publication are:

1. SAAVE, a method to estimate robot angular velocities during gyroscope saturation periods;
2. the TIGS dataset, consisting of 32 distinct runs of a custom perception rig tumbling down a steep hill, reaching angular speeds of up to 18.6 rad/s.

Extending this prior work, this journal extension introduces the following additional contributions:

1. Stretch-ICP, a novel registration and deskewing algorithm that yields a continuous trajectory under aggressive motions, together with the HRMC dataset, which enables high-frequency trajectory and velocity error analysis;
2. an extended experimental evaluation that compares SAAVE against Point-LIO, a lidar-inertial method explicitly designed to remain robust under gyroscope saturation, providing a stronger state-of-the-art baseline.

Researchers can leverage our results in multiple ways. First, the SAAVE method can be seamlessly integrated into any system that relies on inertial measurements, requiring only knowledge of the IMU's position relative to the robot's center of mass and access to its measurements. Then, the TIGS dataset enables the evaluation of SLAM algorithms under extremely aggressive motions, without incurring the high repair costs typically associated with real-world mobile robotic platforms. Finally, for lidar-inertial SLAM systems that rely on classical registration algorithms for localization, the registration module can be replaced with Stretch-ICP to improve trajectory continuity under aggressive motions. This integration requires only minimal modifications to supply inertial data into the registration process.

2. Related Work

In this section, we describe recent work in the literature focused on localization and mapping under *aggressive motion*. In particular, we explain how these approaches were not tested or would not function in cases where gyroscope saturations occur. We then describe existing Gyro-free Inertial Navigation System (GF-INS) methods, aiming to estimate the angular velocity of a robot when gyroscope measurements are saturated. Next, we examine techniques for estimating *temporally high-resolution trajectories*, that is, trajectories reconstructed at the intra-scan scale to compensate for motion distortion in lidar data. Lastly, we analyze lidar SLAM datasets and demonstrate that they are not suited to test SAAVE, our angular velocity estimation method.

2.1. SLAM Algorithms Robust to Aggressive Motions

Several SLAM algorithms were proposed to overcome the challenges posed by aggressive motion. In the FAST-LIO2 SLAM algorithm [9], an Iterated Extended Kalman Filter (IEKF) back-propagates the estimated state to deskew the point cloud after the prediction step. FAST-LIO2 was tested at angular speeds up to 21.7 rad/s, without specifying accelerations and with no mention of gyroscope saturations. In DLIO [10], scans are deskewed using the lidar motion estimated by integrating IMU measurements with a constant jerk and angular acceleration model. After roughly aligning the scan with the map through deskewing, the scan alignment is refined using the Generalized Iterative Closest Point (GICP) [20] registration algorithm. Their method was tested at angular speeds up to 3.6 rad/s and linear accelerations up to 19.6 m/s², but was not tested under saturated gyroscope measurements. Although promising, the aforementioned methods use IMU measurements to compute the prior for their optimization process. If IMU measurements are incomplete because of saturations, they might lead the optimization to converge far

from the true solution. The LOCUS SLAM algorithm [21] addresses sensor failures by incorporating a health-monitoring module to detect malfunctions. In contrast, we propose an approach that not only *detects* but also *recovers* from gyroscope failures, as robots do not have a direct alternative for such measurements. To our knowledge, the only lidar-inertial SLAM framework explicitly designed to remain robust to IMU saturation is Point-LIO [19]. It consists of an on-manifold Extended Kalman Filter (EKF) that registers each individual point to the closest plane upon measurement, relying on a kinematic model to model IMU measurements as an output. As a result of its point-wise measurement update and kinematic IMU output model, Point-LIO avoids explicit scan deskewing. This design makes it inherently less sensitive to short periods of gyroscope saturation. However, the experimental evaluation of Point-LIO focuses on high-rate but dynamically smooth motions, in which saturation events occur without repeated impacts, loss of contact, or abrupt changes in angular velocity. Consequently, while effective under controlled saturation scenarios, the method is not evaluated under collision-rich, tumbling-like motions as defined in this work, where abrupt changes in dynamics and loss of motion predictability are the dominant failure modes. Since Point-LIO is the closest approach to ours in the literature, we include it as a baseline in Section 6. In our previous work [5], we addressed the limitations of scan deskewing under aggressive motion by introducing a SLAM algorithm that explicitly accounts for skewing uncertainty during registration. This uncertainty-aware registration allowed the system to assign greater importance to portions of a scan that were less affected by skewing, partially mitigating the effects of aggressive motion. However, since our method was not robust to gyroscope saturation, we limited our experiments to angular speeds up to 11 rad/s and linear accelerations up to 200 m/s². The aforementioned limitations highlight the need for an angular velocity estimation method that relies on alternative sensory inputs and remains reliable when gyroscope saturation occurs during aggressive motions.

2.2. Angular Velocity Estimation Under Gyroscope Saturation

Several solutions have been proposed to estimate gyroscope measurements during saturation periods. In the work of Dang and Suh [22], the authors propose a smoothing algorithm to estimate saturated gyroscope measurements. They use an optimization algorithm based on the presence of zero-velocity intervals for motion tracking. Their method is well-suited in situations where short gyroscope-saturated time windows occur during a continuous motion contained between zero-velocity periods. However, their method assumes the existence of such zero-velocity intervals and was therefore not designed for scenarios involving sustained, repeated collisions (e.g., tumbling), where the motion does not regularly return to rest. Alternatively, Tan et al. [23] introduced an EKF exploiting the periodic structure of magnetometer measurements to estimate the angular velocity of a monocopter, despite gyroscope saturation. This approach relies on the assumption of an approximately stable rotational motion, which induces a sinusoidal pattern in the magnetometer measurements. In a situation where repeated collisions are sustained, such a structured pattern in the magnetometer measurements cannot be assumed. Moreover, in robotics, magnetometers are often disregarded as proximal magnetic sources bias their measurements [24]. Another approach is explored in the work of Pachter et al. [25], where GF-INS theory is applied to allow the estimation of the position, orientation, linear velocity, and angular velocity of an object in 3D using only accelerometers. Following this work, Lee et al. [6] proposed an EKF to estimate the angular velocity of a rotating plate using three accelerometers, which they validated experimentally. This solution was developed for aerospace applications and was not tested inside a SLAM framework. Since accelerometer-based methods have more potential than other work presented previously, we will build on these solutions to improve the robustness of SLAM algorithms under saturated gyroscope measurements.

2.3. Temporally High-Resolution Trajectory Estimation

An accurate estimate of the intra-scan lidar trajectory is required to deskew incoming point clouds affected by motion distortion [5]. The most influential and performing methods of recent years to estimate such a temporally high-resolution trajectory are presented in this section. In the work of Anderson and Barfoot [26], the authors introduce an efficient method to estimate the continuous trajectory of a robot while estimating landmark positions. They represent the estimated trajectory as a Gaussian Process (GP) and use its prior with a particular form to speed up the computations. To enforce trajectory smoothness, the authors employ a GP trajectory prior rather than a constraint external to the estimator. This smoothing method yields continuous and accurate results under normal conditions, but was not tested under aggressive motions. In their work, Gentil et al. [27] introduced IN2LAAMA. This SLAM algorithm starts by extracting planar and edge features from the point clouds, similar to what is done by Zhang and Singh [28], but using linear regression for more accurate results. Their method then computes Upsampled Pre-integrated Measurements (UPMs) [29] to allow querying inertial measurements at any time. A factor graph is then built and optimized, deskewing and registering the point clouds simultaneously. The authors also implemented loop closure detection and the estimation of extrinsic calibration between the lidar and IMU. Once again, IN2LAAMA was not tested under aggressive motions. It relies on lidar feature extraction, which can become fragile under rapid, collision-rich motions, particularly when using lidars with a low number of beams (e.g., 16). For this reason, we do not compare IN2LAAMA to our method on the datasets collected in this work. CT-ICP is a lidar-only odometry that was introduced in the work of Dellenbach et al. [16]. Their registration algorithm is one of the few approaches in the literature that enforces continuity of the poses within a scan, while explicitly allowing discontinuities between scans. This design choice improves robustness to high-frequency motion, but it does not prevent scan-boundary jumps in the reconstructed trajectory, which can produce unrealistic velocity and acceleration estimates when trajectories are analyzed at intra-scan resolution. In contrast, Stretch-ICP specifically enforces trajectory continuity by constraining the beginning of each intra-scan trajectory to match the end of the previous one while distributing registration corrections over time. In their work, Park et al. [30] introduce a continuous-time SLAM algorithm inspired by ElasticFusion [31], which optimizes a smooth trajectory in a map-centric surfel-based framework. Their method enforces consistency between consecutive scans and the global map while regularizing the trajectory using inertial constraints on linear and angular motion. By allowing non-rigid (deformable) adjustments of the map during optimization, the approach can compensate for registration errors over time. However, the method requires a camera for dense surfel fusion and relies on constraints that favor temporally smooth linear and angular velocities at the trajectory level to maintain consistency between consecutive scans and the evolving map. The repeated impacts and impulsive dynamics considered in this work violate these assumptions, and the method was therefore not evaluated under such aggressive motions. Lastly, in the work of Lang et al. [32], the authors introduce Coco-LIC, an efficient continuous-time lidar-inertial-camera odometry algorithm. Their method uses a factor graph to optimize the positions of non-uniform spline control points, with lidar, IMU, and camera factors in a local sliding window. The IMU bias and gravity vector are initialized under a no-motion assumption, after which the trajectory is extended at 0.1s intervals by dynamically selecting and optimizing spline control points using planar features, inertial measurements, and images. The authors evaluated their method on drones flying at high speeds and on a legged robot experiencing repeated ground contacts; it was therefore not tested under the sustained, collision-rich, and unpredictable motions that define aggressive motions in this work. Since the approach relies on a continuous-time spline trajectory opti-

mized using lidar, camera, and IMU measurements and favors temporally smooth linear and angular velocities to maintain cross-sensor consistency, the repeated impacts and impulsive dynamics considered here fall outside its underlying assumptions. As a result, this method is not evaluated in the experimental comparisons presented in this work.

2.4. Aggressive Motion Datasets

To investigate motion estimation in extreme scenarios, a dataset with aggressive motions and gyroscope saturations is required. We studied the publicly available lidar SLAM datasets that are the most commonly used and most aggressive, namely the Newer College [33] and Hilti-Oxford [34] datasets. Because of its importance in the literature, we also studied the KITTI dataset [35]. The maximum angular speed in all of these datasets combined is 4.7 rad/s and the maximum linear acceleration in all datasets combined is 30.7 m/s². Since the motion in these datasets is not aggressive enough to cause gyroscope saturation, we propose the Tumbling-Induced Gyroscope Saturation (TIGS) dataset, which consists of a perception rig tumbling down a hill, with angular speeds up to 18.6 rad/s and linear accelerations up to 157.8 m/s². These motion magnitudes are sufficient to induce gyroscope saturation and therefore make TIGS a suitable test bed for evaluating SLAM performance under extreme conditions. Additionally, the TIGS dataset includes ground-truth angular velocity measurements, which are essential for validating angular velocity estimation methods such as SAAVE. Collecting such data is challenging, as it involves a high risk of hardware damage, making datasets like TIGS rare. A detailed description of the TIGS dataset is provided in Section 5.

3. Saturation-Aware Angular Velocity Estimation (SAAVE)

This section presents the SAAVE method and the key assumptions behind it. While SAAVE was first introduced in our prior work [18], we include its complete formulation here to provide a self-contained description and to support the extended analyses conducted in this journal paper. We first describe the angular velocity estimation method in Section 3.1, followed by the SLAM framework in which it is integrated, presented in Section 3.2.

3.1. Angular Velocity Estimation

As an illustrative example, inertial measurements during an event of a robot tumbling down a hill are shown in Figure 2. As can be seen, a saturated gyroscope will read the same measurement for all angular speeds. From our experience, the observed saturation point differed from the nominal datasheet range. Nonetheless, gyroscope saturation usually occurs during the middle section of the tumbling, when the angular velocities are at their highest. Accelerometers, on the other hand, will not saturate due to the high angular velocities during tumbling. In our experiments, the maximum linear accelerations resulting from high angular velocities were less than 20% of the accelerometer's saturation threshold. The accelerometer saturations we did observe were caused by collisions with the ground, and these events occur over such short periods of time that they have little impact on state estimation overall. We therefore have two distinct cases during which to estimate saturated gyroscope measurements: (i) during free-fall and (ii) during collisions. Indeed, the plateaus in the angular speed curve correspond to free-fall periods, whereas the fast changes correspond to collisions, as indicated by the spikes in the acceleration curve.

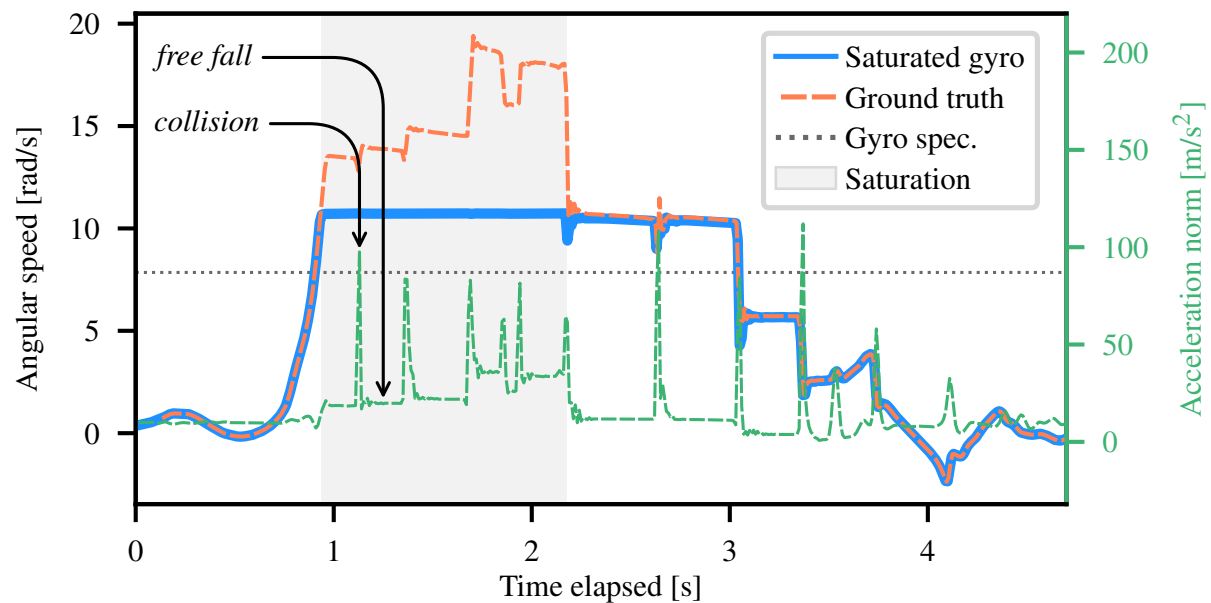


Figure 2. Angular speed over time for the saturated gyroscope axis during a tumbling event. Light gray zones indicate saturation periods. The saturated gyroscope measurements are shown in blue, the ground-truth angular speeds in orange, and the norm of the measured acceleration in green. The manufacturer-specified gyroscope saturation point is shown in dark gray. Examples of collision and free-fall events are highlighted.

Since modeling collisions requires complex simulations [36], it is challenging to estimate angular velocity from accelerometer measurements during ground impacts. As shown in Figure 2, collision events are significantly shorter than free-fall periods, indicating that during the gyroscope saturation period, the robot is free-falling for most of the time. SAAVE therefore starts by estimating the angular velocities under the assumption of free fall. To improve accuracy in situations where this assumption does not hold, our method smooths the estimated angular velocities using a GP with a physically-motivated motion prior. The following assumptions are made to estimate saturated gyroscope measurements:

Assumption 1. *The IMU is not located along the robot's rotation axis, enabling angular velocity estimation from the measured centripetal acceleration.*

Assumption 2. *The measured linear acceleration at the robot's Center Of Mass (COM) is zero because a body in free fall experiences no net acceleration. This conclusion assumes that the force due to air friction is negligible.*

Assumption 3. *The rotation axis remains unchanged between two IMU measurements, supported by the high acquisition rate of IMU measurements, typically at 100 Hz or more, and by angular momentum, which prevents rapid changes in the axis of rotation.*

Assumption 4. *The rotation axis passes through the robot's COM, relying on the principle that, when no external forces act on a body, it rotates about its COM.*

Assumption 5. *Only one axis of the gyroscope is saturated at once, allowing us to compute a simple and precise estimate of the angular speed of a saturated gyroscope axis.*

The important variables are illustrated in Figure 3, where an IMU is linked to the robot's COM by $\mathbf{t}$ and rotates at an angular speed $\omega = \|\omega\|$ around the unit rotation axis $\mathbf{e}$. The lever arm vector $\mathbf{r}$ orthogonally links the IMU to the rotation axis $\mathbf{e}$. To facilitate the estimation, we introduce a rotational frame $\mathcal{R}$ and explicitly distinguish it from the physical IMU body frame, which we denote as $\mathcal{A}$. The IMU frame $\mathcal{A}$ is attached to the physical sensor, and the angular velocities measured by the IMU are naturally expressed in this frame. Conversely, the rotational frame $\mathcal{R}$ is a geometric construct sharing the same origin as the IMU but rotated to align with the rotation axis $\mathbf{e}$. The x axis of the $\mathcal{R}$ frame is perpendicular and pointing to the rotation axis $\mathbf{e}$, and its z axis is in the same direction as $\mathbf{e}$. Quantities expressed in the IMU frame are denoted with left superscript $\mathcal{A}$, while those expressed in the rotational frame are denoted with left superscript $\mathcal{R}$. Quantities without a left superscript may be expressed in any frame, provided that all terms in the same equation are written in a common frame. As the axis-angle representation states, the rotation axis $\mathbf{e}$ can be recovered from the angular velocity $\omega = [\omega_x \ \omega_y \ \omega_z]^T$ such that $\omega = \omega \mathbf{e}$.

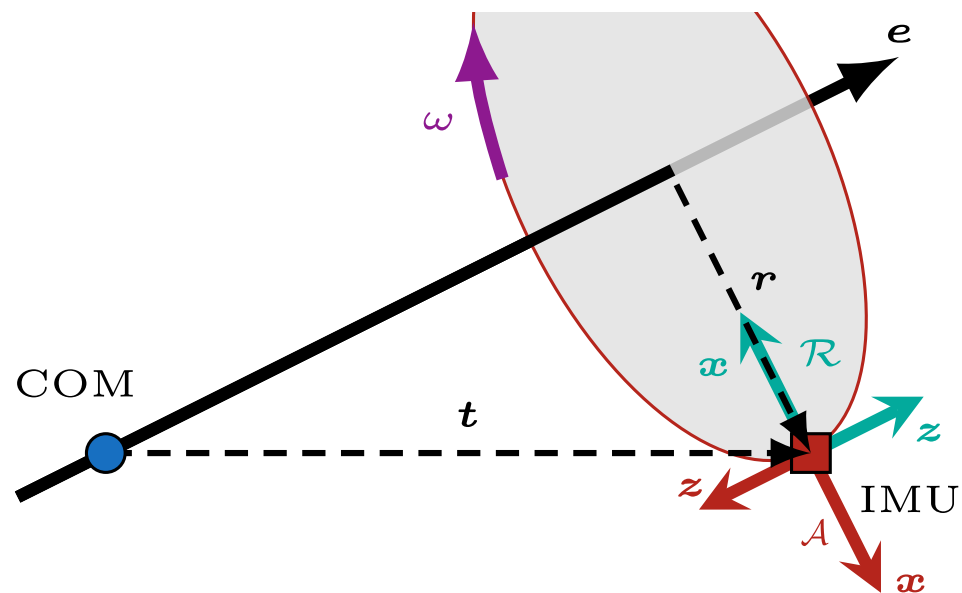


Figure 3. Illustration of the main quantities used in the angular velocity estimation method. The COM is shown as a blue dot and the IMU as a red square. The rotation axis $\mathbf{e}$ is assumed to pass through the COM; it is shown as a solid line. The IMU rotates around $\mathbf{e}$ along the red circle. The vector $\mathbf{t}$ joins the COM to the IMU, and the vector $\mathbf{r}$ joins the rotation axis to the IMU; both geometric constructs are shown with dashed lines. The angular speed ω is shown in purple. The x and z axes of the rotational frame $\mathcal{R}$ are shown in green, while those of the IMU frame $\mathcal{A}$ are shown in red.

Drawing from the work of Pachter et al. [25], the Coriolis formula states that

$$\mathbf{a}_I = \mathbf{a}_C + \dot{\omega} \times \mathbf{r} + \omega \times (\omega \times \mathbf{r}), \quad (1)$$

where $\mathbf{a}_I$ is the linear acceleration at the location of the IMU, $\mathbf{a}_C$ is the linear acceleration at the robot's COM, and $\dot{\omega}$ is the angular acceleration of the robot. All angular velocity and linear acceleration measurements are expressed in a common coordinate frame. Since accelerometers measure proper acceleration (i.e., the experienced acceleration), the measured acceleration $\tilde{\mathbf{a}}_I$ at the location of the IMU is equal to

$$\begin{aligned} \tilde{\mathbf{a}}_I &= \mathbf{a}_I - \mathbf{g} \\ &= (\mathbf{a}_C - \mathbf{g}) + \dot{\omega} \times \mathbf{r} + \omega \times (\omega \times \mathbf{r}) \\ &= \tilde{\mathbf{a}}_C + \dot{\omega} \times \mathbf{r} + \omega \times (\omega \times \mathbf{r}), \end{aligned} \quad (2)$$

where $\mathbf{g}$ is the gravity force vector and $\tilde{\mathbf{a}}_C$ is the measured acceleration at the robot's COM. Using Assumption 2, Equation (2) simplifies to

$$\tilde{\mathbf{a}}_I \approx \dot{\boldsymbol{\omega}} \times \mathbf{r} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}). \quad (3)$$

Equation (3) is the key to allowing the computation of angular velocity during gyroscope saturation periods. The first term of the sum is the tangential acceleration of the IMU and is oriented into the page in Figure 3. The second term of the sum is the centripetal acceleration and is oriented in the same direction as the x axis of the rotational coordinate frame $\mathcal{R}$ in Figure 3. Therefore, expressing the accelerometer measurements $\tilde{\mathbf{a}}_I$ in the coordinate frame $\mathcal{R}$ and using Equation (3), we can deduce without further approximation that

$${}^{\mathcal{R}}\tilde{\mathbf{a}}_I \approx \begin{bmatrix} \|\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r})\| \\ -\|\dot{\boldsymbol{\omega}} \times \mathbf{r}\| \\ 0 \end{bmatrix} = \begin{bmatrix} \omega^2 r \\ -\dot{\omega} r \\ 0 \end{bmatrix}, \quad (4)$$

where $\omega = \|\boldsymbol{\omega}\|$, $r = \|\mathbf{r}\|$ and $\dot{\omega} = \|\dot{\boldsymbol{\omega}}\|$. The y component of ${}^{\mathcal{R}}\tilde{\mathbf{a}}_I$ is negative because the y axis of $\mathcal{R}$ and the IMU tangential acceleration are in opposite directions. The last equality in Equation (4) holds because $\mathbf{r}$ is orthogonal to $\boldsymbol{\omega}$ by definition and to $\dot{\boldsymbol{\omega}}$ due to Assumption 3. From here, the angular velocity can be estimated from either the x or y component of the acceleration vector. However, computing the angular velocity via the angular acceleration $\dot{\boldsymbol{\omega}}$ would lead to integrating noise and thus lead to a less accurate estimate. In order to compute the magnitude of the angular velocity vector $\|\boldsymbol{\omega}\|$, the magnitude of the lever arm $\|\mathbf{r}\|$ must be determined. Using Assumption 4, as can be seen in Figure 3, $\mathbf{r}$ can be retrieved with

$$\mathbf{r} = \mathbf{t} - (\mathbf{t} \cdot \mathbf{e})\mathbf{e}. \quad (5)$$

The axis of rotation $\mathbf{e}$ is usually determined using the angular velocity $\boldsymbol{\omega}$, but this is not possible in the present case, since the measurement of one of the gyroscope axes is saturated. Using Assumption 3, the axis of rotation of the previous estimated angular velocity is used instead. Lastly, without loss of generality, let us assume that the gyroscope is saturated on its x axis (i.e., the ${}^{\mathcal{A}}\omega_x$ measurement is saturated). Since the magnitude of the angular velocity is invariant to the choice of frame, we have $\omega^2 = {}^{\mathcal{A}}\omega_x^2 + {}^{\mathcal{A}}\omega_y^2 + {}^{\mathcal{A}}\omega_z^2$, where ω^2 , although obtained from accelerometer measurements expressed in the rotational frame $\mathcal{R}$, is a scalar and can therefore be expressed equivalently in the IMU frame $\mathcal{A}$. Using Assumptions 1 and 5, we can retrieve the saturated measurement ${}^{\mathcal{A}}\omega_x$ using

$${}^{\mathcal{A}}\omega_x = \sqrt{\frac{\tilde{a}_x}{\|\mathbf{t} - (\mathbf{t} \cdot \mathbf{e})\mathbf{e}\|} - {}^{\mathcal{A}}\omega_y^2 - {}^{\mathcal{A}}\omega_z^2}, \quad (6)$$

where $\tilde{a}_x$ is the x component of ${}^{\mathcal{R}}\tilde{\mathbf{a}}_I$ and ${}^{\mathcal{A}}\omega_y$, ${}^{\mathcal{A}}\omega_z$ are the unsaturated gyroscope measurements. Due to the noise in accelerometer measurements, the computed angular speed might be below the saturation point, which is not possible. To solve this, we conserve the maximum between the estimated angular speed magnitude and the saturation point. The sign ambiguity of the computed angular speed can be resolved by considering the sign of the saturated gyroscope measurement. Again, due to the noise in accelerometer measurements, the term under the radical in Equation (6) can be negative. In that case, we simply reject the estimate. We are left with Equation (6) to estimate the angular velocity when a saturation is detected using a threshold on gyroscope measurements.

We now smooth the angular velocity estimates computed previously using a GP with a physically-motivated motion prior. Similarly to what was done by Tang et al. [37], a white-noise-on-jerk motion prior is used. To account for the possibly abrupt changes in

angular velocity, the diagonal entries of the angular jerk power spectral density matrix are set to a high value $q_{\ddot{\omega}}$. The unsaturated gyroscope measurements are assigned a variance of $\sigma_{\ddot{\omega}}^2$, which is computed using the IMU specifications. The valid angular speed estimates are given a higher variance, $\sigma_{\dot{\omega}}^2$, which is a parameter of our method. Employing a GP for smoothing has the advantage of yielding both the mean and covariance of the estimated angular velocity as functions of time. The STEAM library, from Anderson and Barfoot [26], was used to carry out these computations. The covariance of the estimated angular velocity can then be fed into the SLAM framework. The overall estimation, including the orientation estimation used to recover the gravity direction, operates at the full IMU sampling rate of 100 Hz. This high-frequency operation enables tracking of rapid orientation changes during extreme motions such as tumbling.

3.2. Using SAAVE in a SLAM Framework

The SAAVE method introduced in Section 3.1 is integrated into a lidar-inertial mapping framework implemented with the `norlab_icp_mapper` library [38]. The pipeline processes lidar scans sequentially in a scan-to-map manner. Each incoming scan undergoes three main steps. (1) *Scan pre-processing*: We start by applying a series of filters to the incoming scan to pre-process it before registration. (2) *Deskewing and registration*: Raw lidar scans are expressed in the sensor's local frame and implicitly assume a fixed sensor pose over the duration of a scan [5]. As a result, point positions must be corrected to account for intra-scan motion. Using Stretch-ICP, which is described in Section 4, the lidar scans are deskewed and registered into the map simultaneously. This registration algorithm also outputs the continuous intra-scan trajectory of the current scan, the end of which is used as the starting pose for the intra-scan trajectory of the next scan. (3) *Merge and map maintenance*: Once filtered, deskewed, and registered, the scan is merged into the map of the environment, and maintenance operations are performed on the resulting map. The specific SLAM configuration used in our experiments, including the filters and map maintenance operations involved in steps (1) and (3), is detailed in Section 5.2. Among these three stages, step (2) is the one most directly influenced by the introduction of SAAVE, since it relies on angular velocity inputs to estimate the intra-scan trajectory.

4. Stretch-ICP

In this section, we first provide an overview of Stretch-ICP, our continuous-trajectory registration and deskewing algorithm, in Section 4.1. Next, we introduce our notation in Section 4.2, followed by a description of Stretch-ICP in Section 4.3. Finally, we detail the key component of Stretch-ICP, the Data Stretcher, in Section 4.4.

4.1. Overview

An illustration of our registration algorithm in action, compared to ICP, is shown in Figure 4. As can be seen, when motion prediction is imperfect, ICP applies a corrective *rigid* transformation to align the scan with the map. This correction repositions the intra-scan trajectory estimated with the IMU preintegration, introducing a discontinuity at the scan's start, as illustrated by the gap between the blue square and the orange x in the upper right trajectory. In contrast, Stretch-ICP treats registration as a continuous-time trajectory deformation problem. Rather than correcting the scan with a rigid transformation, it distributes the registration correction over the intra-scan trajectory by stretching it in a manner consistent with inertial constraints and scan-boundary continuity. As a result, the scan is deskewed and aligned with the map while preserving continuity between consecutive scans. It is also possible to see in Figure 4 that, similarly to ICP, Stretch-ICP

relies on point matching errors computed at each iteration to guide the optimization, but applies these corrections to the trajectory itself instead of to the scan as a rigid body.

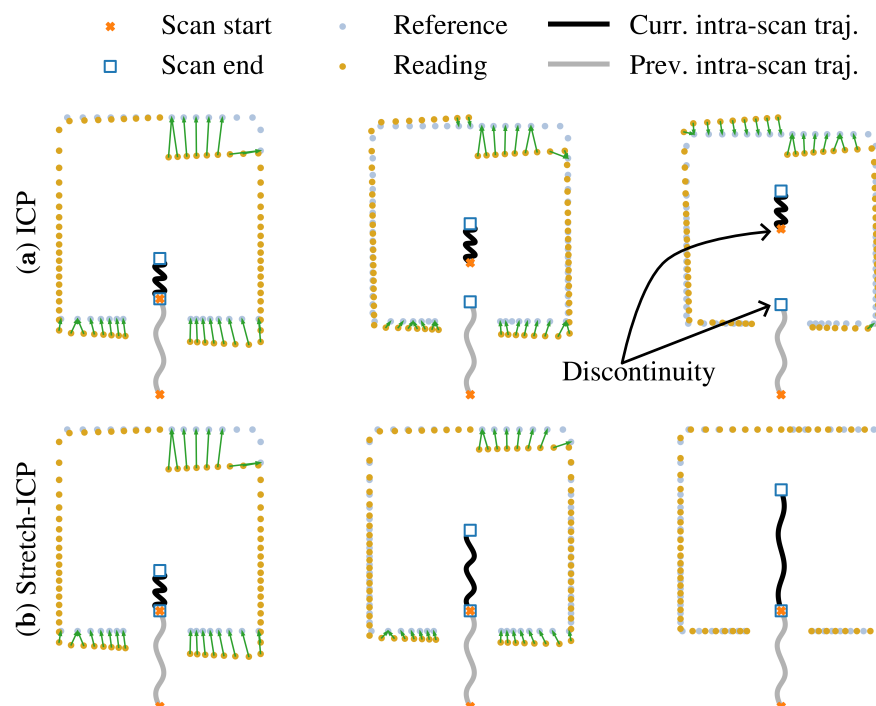


Figure 4. Toy example comparing Stretch-ICP with ICP. The light blue points represent a map, while the yellow points represent a scan acquired by a moving lidar. The orange dot and blue square represent the start and end positions of two consecutive intra-scan trajectories. The gray line represents the previous intra-scan trajectory, and the black line the current one. The green arrows indicate the matches used by the registration algorithms. Subfigure (a) shows three iterations of ICP, and subfigure (b) shows three iterations of Stretch-ICP.

4.2. Notation

In the following sections, the scan will be referred to as the reading point cloud $\mathcal{P}$; the map will be referred to as the reference point cloud $\mathcal{Q}$; the state of the lidar, which includes its position, its orientation and its linear velocity, will be denoted as $\mathbf{x}$; a gyroscope measurement will be denoted as ω ; an accelerometer measurement will be denoted as $\mathbf{a}$; the time between scans will be denoted as s ; and the time between IMU measurements will be denoted as q . We will use left superscripts to index a variable in time (e.g., the state at time t will be denoted ${}^t\mathbf{x}$) and left subscripts and superscripts to express the conversion from one coordinate frame to another (e.g., the transformation T from the lidar frame at time $t + s$ to the lidar frame at time t will be denoted as ${}_{t+s}^tT$). Lastly, we will use right subscripts to index the elements of a sequence (e.g., the transformation T computed at the i^{th} iteration of the optimization will be denoted as T_i).

4.3. Algorithm Description

Our novel registration algorithm, Stretch-ICP, is based on the ICP algorithm, as described in the work of Pomerleau et al. [39], but with an additional step. This step, which we named *Data Stretcher*, is in charge of deskewing the reading point cloud $\mathcal{P}$ to align it to the reference point cloud $\mathcal{Q}$. Roughly speaking, the intra-scan trajectory employed for deskewing is stretched using a factor graph that incorporates constraints from IMU measurements, from the previous intra-scan trajectory ending pose, and from the intra-scan displacement computed from ICP matching errors. The steps of our algorithm are illustrated in Figure 5. First, the reading $\mathcal{P}$ and reference $\mathcal{Q}$ point clouds are filtered using

customizable data filters, such as downsampling or other pre-processing filters. Then, the filtered reading point cloud $\mathcal{P}'$ is deskewed using a stretched intra-scan trajectory in the Data Stretcher. The points of this deskewed reading $\mathcal{P}''$ are matched with the points of the filtered reference point cloud $\mathcal{Q}'$. Next, the matching point pairs $\mathcal{M}$ are given weights $\mathcal{W}$ using outlier filters. These outlier filters, which are customizable, are typically based on the matching error percentile of the point pairs. Lastly, a rigid transformation T_i that aligns the deskewed reading $\mathcal{P}''$ with the filtered reference point cloud $\mathcal{Q}'$ is computed using the weighted matching errors and sent to the Data Stretcher to stretch the intra-scan trajectory at the next iteration of the algorithm. After convergence, the obtained stretched intra-scan trajectory aligns the reading $\mathcal{P}$ in the reference point cloud $\mathcal{Q}$ when using it for deskewing. Our algorithm can be seen as a generalization of the ICP algorithm, where the Data Stretcher is usually implemented by applying a rigid transformation to the filtered reading point cloud $\mathcal{P}'$, with no deskewing.

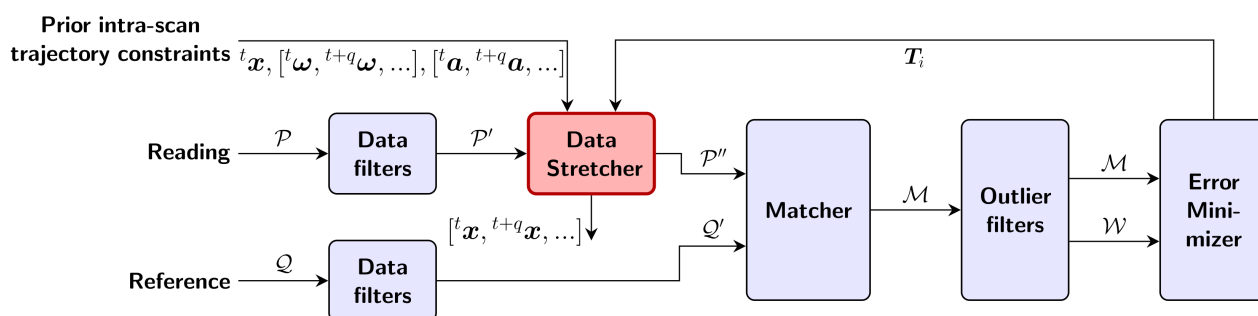


Figure 5. Illustration of the steps of Stretch-ICP. The algorithm takes as input the lidar state x_t at the start of the scan, the IMU measurements $[^t\omega, ^{t+q}\omega, \dots]$ and $[^ta, ^{t+q}a, \dots]$ recorded during the scan, the reading point cloud $\mathcal{P}$, and the reference point cloud $\mathcal{Q}$. The output is the lidar trajectory $[^tx, ^{t+q}x, \dots, ^{t+s}x]$ during the scan. Our novel Data Stretcher module is highlighted in red, while the classical ICP modules are shown in blue.

4.4. Data Stretcher

Our Data Stretcher module is composed of three steps: *S1* prior trajectory estimation, starting from the end pose of the previous intra-scan trajectory and predicting motion using IMU preintegration, *S2* stretching of this prior trajectory using the transformation computed by the Error Minimizer described previously, and *S3* deskewing of the reading point cloud with this stretched intra-scan trajectory. While steps *S1* and *S3* are straightforward, step *S2* requires more attention. At each iteration of Stretch-ICP, we stretch the intra-scan trajectory by optimizing the factor graph shown in Figure 6. This formulation turns the scan-to-map alignment update into a trajectory deformation problem constrained by IMU preintegration and scan-boundary continuity, rather than a rigid scan correction that can introduce discontinuities. The variables of this factor graph are the IMU states during a scan, and the factors belong to one of the following three categories: intra-scan trajectory beginning factor, IMU factor, and stretching factor. The trajectory beginning factor $\mathcal{B}$ forces the first pose of the intra-scan trajectory to be equal to the last pose of the previous intra-scan trajectory. The covariance of the trajectory beginning factor is set as close to zero as possible to make this constraint very rigid in the factor graph. The IMU factors $\mathcal{I}$ constrain consecutive IMU states (i.e., pose and linear velocity) in the intra-scan trajectory using IMU preintegration. The computation of error residuals for these factors follows the methodology described by Forster et al. [40]. Covariances are also determined according to their approach, with accelerometer covariances set based on manufacturer specifications and integration covariances assigned a very low value. The gyroscope covariances, on the other hand, are the covariances outputted by SAAVE, which are the result of a GP. At

the i^{th} iteration of Stretch-ICP, the stretching factor $\mathcal{S}$ enforces a rigid transformation ${}^{t+s}T_i$ between the first and last poses of the intra-scan trajectory, which is computed as follows:

$${}^{t+s}T_i = {}^tT^{-1} \Lambda_i {}^{t+s}\check{T}, \quad (7)$$

where tT is the beginning pose of the intra-scan trajectory, Λ_i is the composition of all the transformations computed by the error minimizer up to the i^{th} iteration (i.e., $\Lambda_i = T_{i-1}T_{i-2} \cdots T_1$) and ${}^{t+s}\check{T}$ is the initial guess on the intra-scan trajectory ending pose, estimated at step S1. The variance σ_S^2 of the stretching factor, which is the same in translation and in rotation, is a parameter of our method. By optimizing this factor graph, we obtain an intra-scan trajectory $[{}^tx, {}^{t+q}x, \dots, {}^{t+s}x]$ that deskews and aligns the scan in the map.

$\mathcal{B}$: Intra-scan trajectory beginning factor

${}^t\mathcal{I}$: IMU factor at time t

$\mathcal{S}$: Stretching factor

tx : State at time t

s : Delay between scans

q : Delay between IMU measurements

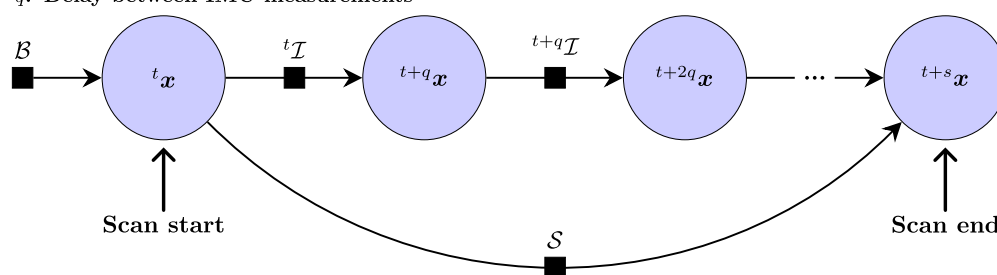


Figure 6. Factor graph used to optimize the intra-scan trajectory. The circles represent the intra-scan trajectory states x , and the black squares represent the factors $\mathcal{B}$, $\mathcal{I}$, and $\mathcal{S}$.

5. Experimental Setup

To validate the improvements reached through our methods while minimizing damage to a full robot, we created a rugged perception rig, shown in the top right of Figure 7. A RoboSense RS-16 lidar (RoboSense, Shenzhen, China) was used to record the 3D point clouds at a frequency of 10 Hz. The sensor has 16 beams, a vertical field of view of 30° , and a horizontal angular resolution of 0.2° at 10 Hz. According to the manufacturer, it provides a typical accuracy of ± 2 cm and a maximum range of up to 150 m. Although the RS-16 does not natively output per-point timestamps, the lidar driver was modified to compute and include per-point timing using the firing order and timing parameters provided in the manufacturer's documentation. The rig is also equipped with two IMUs for angular velocity measurements, with distinct gyroscope saturation points. The first IMU is an Xsens MTi-30 (Xsens, Enschede, The Netherlands), with a gyroscope saturating at 10.5 rad/s, despite the Xsens specification sheet stating a saturation point of 7.85 rad/s. The second IMU is a VectorNav VN-100 (VectorNav Technologies, Dallas, USA), with a gyroscope saturating at 34.9 rad/s according to its specification sheet. Because gyroscope saturation was not reached for the VN-100 in our experiments, its angular velocity measurements are used as ground truth. The MTi-30 model was chosen as many MEMS IMUs commonly used in mobile robotics operate with comparable full-scale gyroscope ranges, making saturation a realistic concern in aggressive-motion scenarios. All sensor measurements (lidar and IMUs) were timestamped upon reception and logged for offline post-processing using the onboard Raspberry Pi 4 (Raspberry Pi Ltd, Cambridge, UK) system clock, providing a common time reference across sensors. Because all sensors rely on the same clock source, no additional hardware synchronization (e.g., trigger signals or PPS) was used. We did not observe any performance degradation or require any compensation for temporal misalignment, suggesting that synchronization errors are not a limiting factor in the proposed pipeline.

Finally, the COM of the rig was estimated manually by balancing the system on a single point on each face.

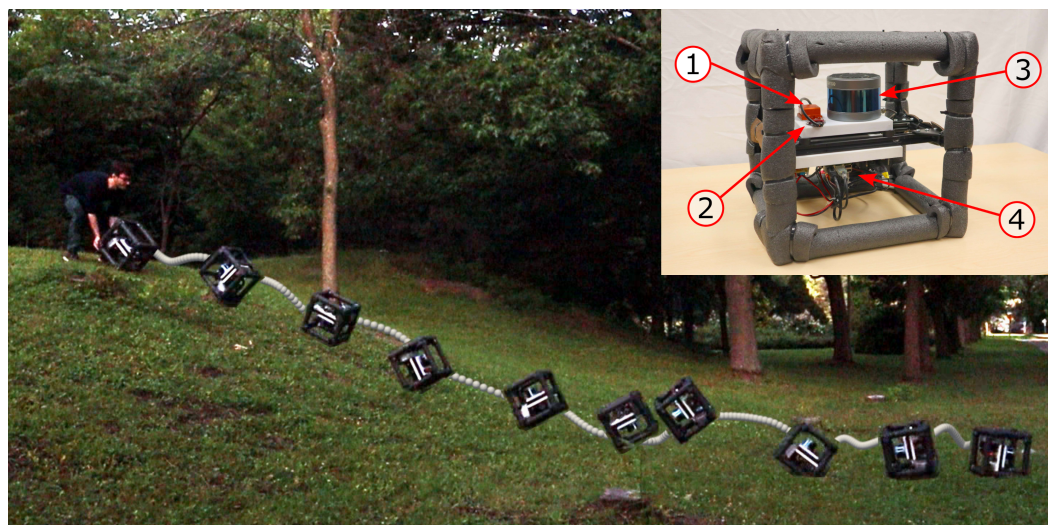


Figure 7. Composite image showing our localization system tumbling down a steep hill in one of the 32 runs of the TIGS dataset. The inset photo shows our rugged perception rig, with the numbers in the red circles corresponding to (1) XSens MTi-30 IMU, (2) VectorNav VN-100 IMU, (3) RoboSense RS-16 lidar, and (4) Raspberry Pi 4.

All experiments were executed offline on a Lenovo ThinkPad P52s laptop (Lenovo, Morrisville, USA) equipped with an Intel Core i7-8550U CPU (4 cores, 8 threads) and 16 GB of RAM. To ensure that each lidar scan could be fully processed before the next one was issued, all datasets were replayed at 10% of their real acquisition speed during evaluation. Consequently, the experiments reported in this paper do not constitute a real-time deployment study. Our goal here was to evaluate robustness and trajectory quality under aggressive motions rather than to optimize runtime. In the current implementation, SAAVE is computationally dominated by the Gaussian-process smoothing stage, while Stretch-ICP incurs additional cost compared with classical ICP because each registration iteration also optimizes a factor graph for the intra-scan trajectory. We therefore consider the present implementation to be an offline research prototype.

5.1. Datasets

To gather sensor measurements under aggressive motions, we recorded a unique outdoor dataset with our perception rig tumbling down a steep hill, as shown in Figure 7. This dataset (data available at https://github.com/norlab-ulaval/Norlab_wiki/wiki/TIGS-Dataset; accessed on 1 February 2026), which we named Tumbling-Induced Gyroscope Saturation (TIGS), includes a total of 32 distinct runs, consisting of pushing the rig to roll down a steep hill, mimicking a tumbling robot. A ground-truth map was built by moving the sensor rig slowly in the environment, thus limiting skew in the scans. The ground-truth 6-DOF displacement of the rig between the start and end of each run is provided in order to quantify localization error. The ground-truth transformations were found by registering the first and last scans of each run in the ground-truth map, as the perception rig is static at these times.

To show how our TIGS dataset covers a larger spectrum of aggressive motions than other mechanical lidar SLAM datasets, we present the distributions for observed linear accelerations and angular speeds in Figure 8, in comparison to the KITTI [35], Newer College [33], and Hilti-Oxford [34] datasets. Our dataset covers a significantly larger range of aggressive motions, characterized by high linear accelerations and angular speeds.

Indeed, the maximum recorded linear acceleration for the TIGS dataset is 414% higher than the highest linear acceleration observed in the compared datasets. Since the saturation point of the VN-100 accelerometer was reached for some collisions, the increase in linear acceleration that was sustained is probably higher than this number. Furthermore, the maximum angular speed for the TIGS dataset is 296% higher than the highest angular speed observed in the compared datasets. Notably, TIGS is the only dataset exhibiting angular speeds beyond the specified saturation point of the Xsens gyroscope, enabling the evaluation of SLAM pipelines under saturated gyroscope measurements. Angular speeds exceeding 20 rad/s were not observed in these experiments, reflecting the practical difficulty of generating higher angular velocities during uncontrolled aggressive motions in ground-based robotic systems.

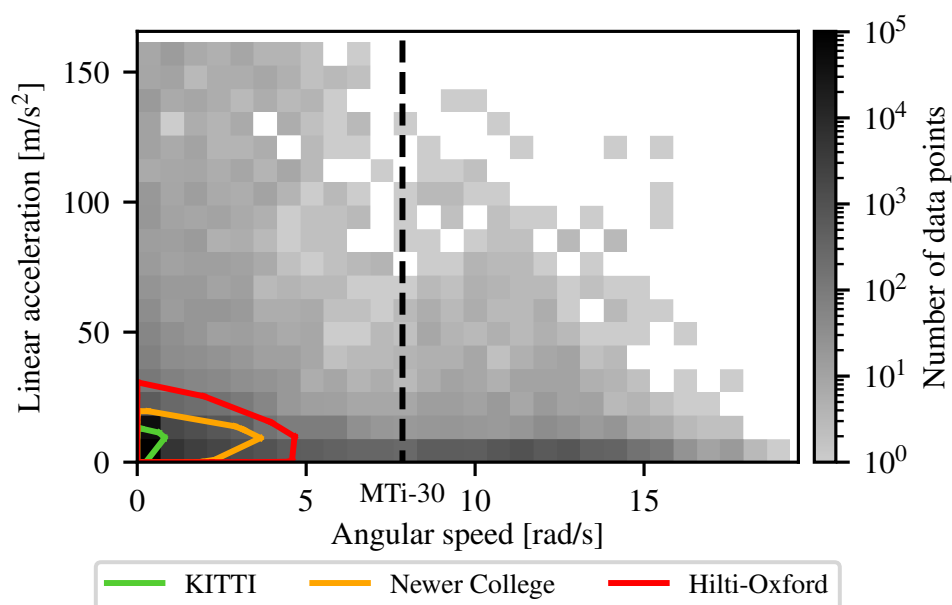


Figure 8. Density map of the TIGS dataset. The grayscale represents the number of data points acquired at the specific angular speeds and linear accelerations. The outlines represent the distributions in linear accelerations and angular speeds for similar datasets. The KITTI dataset is shown in green, the Newer College dataset is indicated in orange, and the Hilti-Oxford dataset is illustrated in red. The dashed line represents the manufacturer-specified saturation point of the MTi-30 gyroscope.

We additionally recorded the HRMC dataset (data available at https://github.com/norlab-ulaval/Norlab_wiki/wiki/HRMC-Dataset; accessed on 1 February 2026), which consists of experiments in which a perception rig is moved rapidly within a room equipped with a Vicon Motion Capture (Mo-Cap) system (Vicon, Yarnton, UK), as illustrated in Figure 9. This dataset was recorded with a similar perception rig as the TIGS dataset, except for its protective cage, whose shape was changed to a sphere. The HRMC dataset also contains a total of 32 runs, with the ground-truth positions and orientations of the rig recorded at a rate of 200 Hz by the Mo-Cap system. Each run consists of an approximately 15-s trajectory, generated by two operators manually actuating the rig rapidly via ropes attached to either side. Across the dataset, the operators followed a standardized set of motion primitives designed to excite different translational and rotational directions while keeping the rig within the Vicon tracking volume. Depending on the run, the trajectory focused either on a single repeated motion primitive, such as figure-eight motions or up-down zig-zags, or on a sequence of short straight-line pulls in varying directions. The motion design aimed to produce aggressive, diverse 6-DOF trajectories suitable for evaluating trajectory continuity and velocity estimation, rather than to maximize angular

speed at all costs. In practice, gyroscope saturation was not reached in this dataset because the achievable motion was constrained by two factors: (i) the need to remain within the limited indoor tracking space, and (ii) the requirement to preserve accurate Vicon ground truth. In particular, the limited sampling rate of the Vicon system, together with the risk of marker loss or degraded tracking quality, restricted the use of more rapidly rotating maneuvers. Nevertheless, the HRMC dataset enables trajectory-level evaluation that is impossible with the TIGS dataset, including linear velocity estimation error, scan-boundary artifacts, and continuity metrics along the full trajectory. Table 1 summarizes the properties of the two datasets introduced in this section.

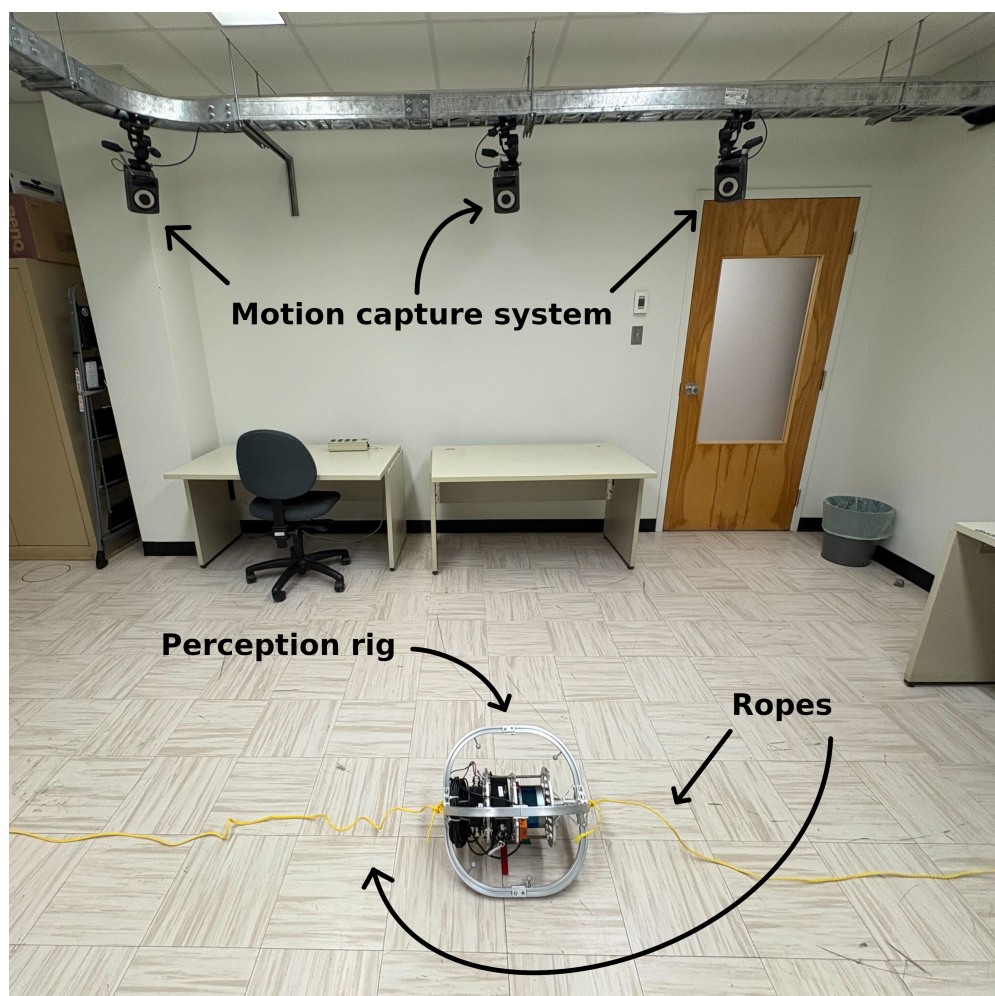


Figure 9. Photo of our perception rig inside the room where the HRMC dataset was recorded. Small reflective spheres were fixed on the rig to track its 6-DOF trajectory using a Vicon Mo-Cap system. Ropes were attached to both sides of the rig to allow the operators to move the rig rapidly while avoiding occluding the Mo-Cap system’s cameras with their bodies.

To better contextualize the motion regimes covered by our datasets from a sensor-selection perspective, Table 2 compares the maximum angular velocities observed in the TIGS and HRMC datasets with the gyroscope operating ranges of the inertial sensors used in our experiments and of several representative commercially available IMUs. The table shows that the angular velocities reached in TIGS exceed the operating range of several commonly used sensors, including the Xsens MTi-30 used in our setup, whereas those reached in HRMC remain within the range of a broader set of units. At the same time, the TIGS angular velocities remain within the range of higher-range sensors such as the VectorNav VN-100, which we used as angular-velocity ground truth. This comparison

provides practical context for selecting inertial sensors for robotic systems expected to operate under aggressive dynamic conditions.

Table 1. Properties of the TIGS and HRMC datasets.

Property	TIGS	HRMC
Environment	Outdoor	Indoor
Features	Grass, hill, trees, building	Walls, furniture
Scale	$1.24 \times 10^6 \text{ m}^3$	$2.05 \times 10^3 \text{ m}^3$
Number of runs	32	32
Max. acceleration	157.8 m/s^2	66.6 m/s^2
Max. angular speed	18.6 rad/s	6.0 rad/s
Saturations	Gyroscope, accelerometer	None
Ground truth	Angular velocity, total displacement	Angular velocity, 6-DOF trajectory

Table 2. Comparison of the gyroscope operating ranges of representative commercially available IMUs and the expected occurrence of saturation in the TIGS and HRMC datasets.

IMU model	Nominal Gyroscope Range	Saturation Expected in TIGS (Max. Ang. Speed: 18.6 rad/s)	Saturation Expected in HRMC (Max. Ang. Speed: 6.0 rad/s)
Xsens MTi-30 (Xsens, Enschede, The Netherlands)	$\pm 7.85 \text{ rad/s}$	Yes	No
VectorNav VN-100 (VectorNav Technologies, Dallas, USA)	$\pm 34.9 \text{ rad/s}$	No	No
Xsens Sirius (Xsens, Enschede, The Netherlands)	$\pm 5.24 \text{ rad/s}$	Yes	Yes
MicroStrain 3DM-GX5-AHRS (MicroStrain, Williston, USA)	$\pm 5.24 \text{ rad/s}$	Yes	Yes
Bosch BHI260AP (Bosch Sensortec, Reutlingen, Germany)	$\pm 34.9 \text{ rad/s}$	No	No

5.2. Method Parameters and SLAM Configuration

Throughout the experiments, the parameters of our methods were set and remained unchanged to allow a fair comparison. The specific values of the scalar parameters are listed in Table 3. The value of the gyroscope measurement variance σ_{ω}^2 was computed from the MTi-30 datasheet, and the values of the angular jerk noise $q_{\ddot{\omega}}$, of the gyroscope estimation variance $\sigma_{\hat{\omega}}^2$, and of the stretching factor variance σ_S^2 are hyperparameters of our method. The values for these parameters were selected manually after trial and error on our experimental data. Our SLAM results were computed offline, using the SLAM framework described in Section 3.2. The pre-processing filters in the first step of our SLAM algorithm, described in Section 3.2, are a bounding box filter [41] to remove the parts of the sensor rig that were obstructing the field of view of the lidar and a voxel grid subsampler [41] that reduces point density by keeping only one point per 15 cm cell to ensure computation efficiency. The map maintenance operations in the third step of our SLAM algorithm are the surface normal computation [42] for each map point using its 20 nearest neighbors. The outlier filters that were mentioned in Section 4.3 of Stretch-ICP are a variable trimmed distance outlier filter [43], which dynamically removes between 30% and 50% of the point pairs with the highest Euclidean residuals. Additionally we implemented a quadratic weighted filter, which assigns a weight $w_i = (t_i/T)^2$ to each point i , where t_i is the relative timestamp and T is the total scan duration. This weight increases quadratically from 0 to 1 throughout the scan. The quadratic outlier filter is used to give a higher importance to the

points at the end of the scan, since our registration algorithm has more influence on the end of the intra-scan trajectory when applying a stretch. Lastly, a point-to-plane [44] cost function was used in the error minimizer module introduced in Section 4.3.

Table 3. Scalar parameters of our methods. The values listed in this table remained unchanged throughout our experiments to allow fair comparisons with other methods.

Parameter	Symbol	Value	Unit
Angular jerk noise	$q\ddot{\omega}$	10^6	$\text{rad}^2/\text{s}^6 \text{ Hz}$
Gyroscope measurement variance	σ_{ω}^2	2.74×10^{-5}	rad^2/s^2
Gyroscope estimation variance	$\sigma_{\hat{\omega}}^2$	3.65	rad^2/s^2
Stretching factor variance	σ_S^2	10^{-24}	m^2, rad^2

6. Results

In this section, we evaluate a state-of-the-art lidar odometry algorithm on the TIGS dataset to illustrate its challenging nature. We then characterize the positive impact of SAAVE on SLAM algorithms using this dataset. Specifically, we evaluate the angular velocity estimation accuracy of SAAVE during gyroscope saturation, leveraging the ground-truth angular velocities provided by a separate, higher-range gyroscope in the TIGS dataset. Using the same TIGS dataset, we then assess how SAAVE's ability to estimate these angular velocities despite gyroscope saturation increases robustness within a SLAM pipeline. Then, we extend the analysis beyond final pose accuracy by comparing angular and linear velocity estimates with and without Stretch-ICP, using the TIGS and HRMC datasets, respectively, with the latter providing high-resolution 6-DOF ground-truth trajectories. Finally, using the TIGS dataset, we demonstrate that adding Stretch-ICP to SAAVE to increase trajectory smoothness does not negatively impact the localization accuracy of SLAM significantly.

6.1. Motion Aggressiveness of TIGS

To highlight the challenging nature of our TIGS dataset, even without consideration for the gyroscope saturation, we tested KISS-ICP [45], a state-of-the-art lidar odometry algorithm, on it. KISS-ICP explicitly relies on a constant-velocity motion model to deskew incoming scans, and does not use IMU measurements; as such, its performance is unaffected by gyroscope saturation. For all 32 runs of the TIGS dataset, KISS-ICP diverged and resulted in a localization and mapping failure. This behavior is expected and serves to illustrate the severity of the motion conditions present in the TIGS dataset: the aggressive motions violate the constant-velocity assumption underlying KISS-ICP's deskewing model. Additionally, the low number of channels of our lidar (i.e., 16) also leads to fewer geometric constraints in the scans, which might be another cause for the failure of KISS-ICP.

We can also quantify the aggressiveness of the motions in the TIGS dataset by estimating the magnitude of the motion-induced geometric distortions they produce in lidar scans. As we quantified in earlier work [5], sensor motion during a scan induces point cloud skewing that degrades registration accuracy and can lead to the failure of standard algorithms. Without accurate angular velocity compensation, these distortions directly translate into significant spatial errors. Following the deterministic model of Haider et al. [46], and considering a 10 Hz lidar, we can quantify the impact of gyroscope saturation on mapping integrity. For instance, if the angular velocity is underestimated at the manufacturer-specified saturation point of 7.85 rad/s while the actual speed reaches our experimental maximum of 18.6 rad/s, the residual spatial distortion for an object at a 10 m range remains as high as 10.75 m even after attempted deskewing. Such errors are several orders of magnitude larger than the typical sensor noise floor (on the order of 10 cm), fundamentally corrupting the geometric features required for reliable SLAM.

6.2. SAAVE Angular Speed Accuracy

Using SAAVE, which was described in Section 3.1, the angular speed of the platform was estimated for all the runs in our TIGS dataset. An example is shown for a single run in the left subplot of Figure 10. The SAAVE method was applied to the MTi-30 measurements, while the VN-100 measurements were only used as ground truth. The angular speed estimation error during gyroscope saturation periods with and without SAAVE is shown for all runs in our dataset in the right part of Figure 10. Only periods of saturation are studied (i.e., the light gray area), as angular speeds are the same with or without SAAVE outside the saturation zones. When accounting for all runs, our approach reduces the angular speed error median by 83.4% when compared to saturated gyroscope measurements. The spikes observed in the SAAVE curve correspond to collision events during which the free-fall assumption assumed by the method is no longer valid.

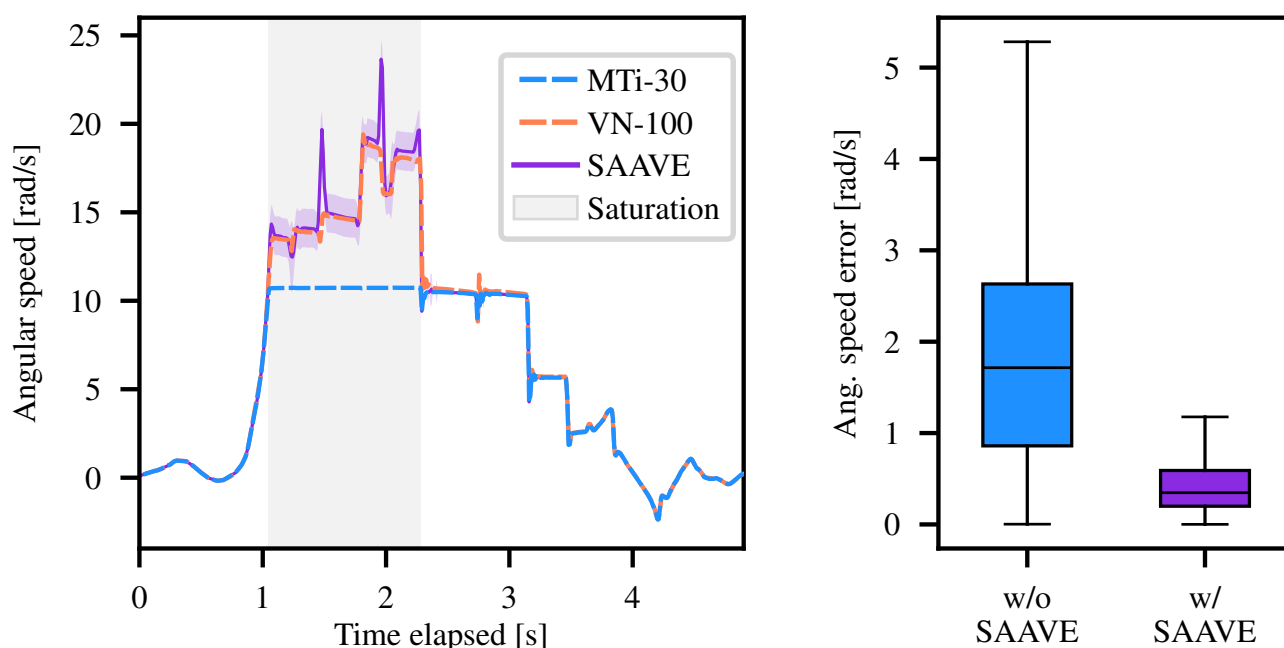


Figure 10. Qualitative and quantitative results on angular speed estimates for SAAVE. **(Left)** An example of the angular velocity over time for the saturated gyroscope (MTi-30) axis for one run of the TIGS dataset. The measurements from the MTi-30 gyroscope are shown in dashed blue, the ground truth measurements from a VN-100 gyroscope are indicated in dashed orange, and the angular speeds estimated with SAAVE using MTi-30 measurements are illustrated in purple. The purple-shaded area represents three standard deviations above and below the angular velocities estimated by SAAVE. The light gray zone indicates a gyroscope saturation episode. **(Right)** The error in angular velocities without (in blue) and with SAAVE (in purple) during saturation periods for all the runs of the TIGS dataset.

6.3. Reduction of SLAM Localization Error with SAAVE

To assess the impact of SAAVE on SLAM, we evaluate our SLAM framework on the TIGS dataset with raw gyroscope measurements and with velocities estimated by SAAVE as input. We isolate the effect of our angular velocity estimation method on the results by employing the state-of-the-art registration and deskewing algorithms described in our previous work [18] instead of Stretch-ICP in our SLAM framework. We refer to this framework as ICP-SLAM when used alone, and as SAAVE-ICP-SLAM when used in combination with SAAVE. We also evaluate Point-LIO [19] on our dataset, which is currently the only other method in the state of the art claiming robustness to gyroscope saturations. To ensure a fair comparison, all methods use the same time-synchronized lidar

and IMU streams, the same extrinsic calibration, the same lidar per-point timestamps, and were tuned to the best of our knowledge. Figure 11 illustrates the distribution of localization errors for the different methods by showing the cumulative probability of observing a given translation or rotation error. As a reminder, the localization error corresponds to the error in the estimated transformation between the initial and final poses of the rig for a run. As can be seen, the median localization error of SAAVE-ICP-SLAM is improved by 71.5% for translation and by 65.5% for rotation when compared to ICP-SLAM. It can also be observed that the lowest translational localization errors of Point-LIO are lower than those of the other methods, while its lowest rotational errors are comparable. However, due to the broader spread of its error distribution, the cumulative probability for Point-LIO increases more slowly than for the other methods. The large translational and rotational errors observed for Point-LIO on the right side of the plot result from a divergence of the algorithm on several runs. As a result, SAAVE-ICP-SLAM quickly outperforms Point-LIO in translational localization error and consistently achieves lower rotational localization errors. ICP-SLAM also achieves lower localization errors than Point-LIO, albeit at a slower rate, and it remains functional under saturated angular velocity measurements without SAAVE on multiple runs, as its registration algorithm can compensate for inaccurate motion priors.

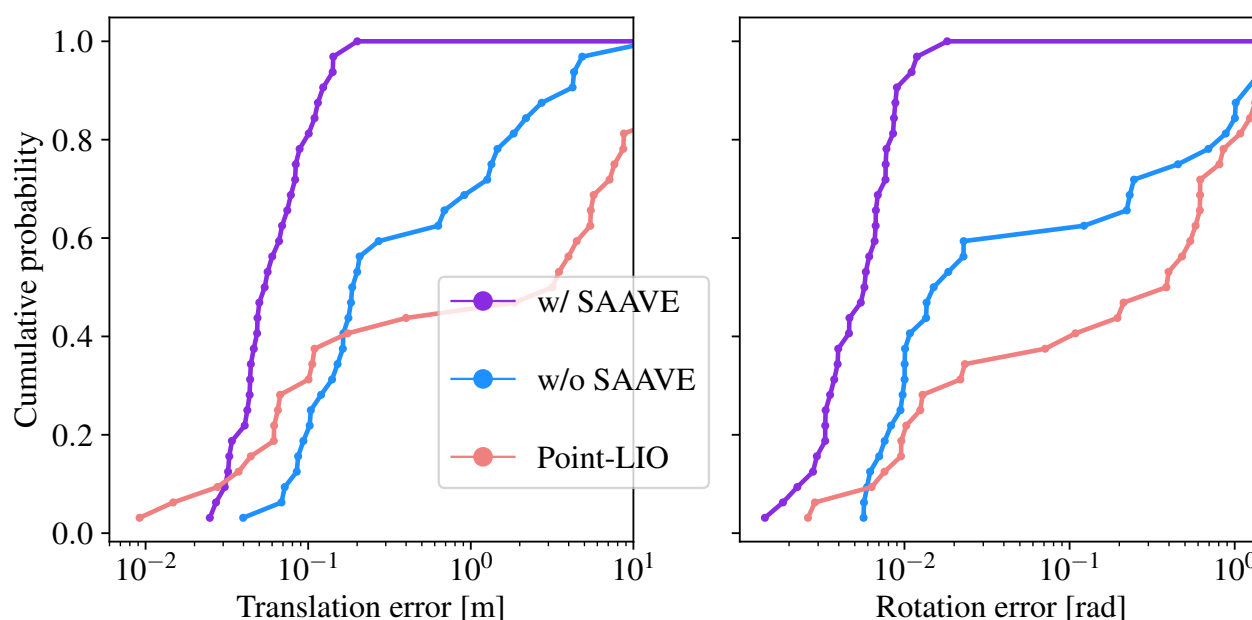


Figure 11. Localization error for all runs in the TIGS dataset. **(Left)** The cumulative probability of observing a given translation error is shown on the left plot. **(Right)** The cumulative probability of observing a rotation error. The blue, purple, and pink lines represent the cumulative probability of localization errors for ICP-SLAM, SAAVE-ICP-SLAM, and Point-LIO, respectively. Note that the errors on the x-axis are in log scale.

To investigate the mapping performance of the different methods, we analyze the maps built by each of them for all the runs of the TIGS dataset. We built on prior work by Chung et al. [47] for the Defense Advanced Research Projects Agency (DARPA) Subterranean Challenge to evaluate mapping quality. Our map overlap metric is the percentage of reconstructed map points that are within a threshold distance from a point belonging to the ground-truth map. In the present case, we chose the threshold distance to be 0.25 m as opposed to the 1 m from the work of Chung et al. [47] to reflect the much smaller scale of our experiments. Indeed, the distance travelled in our runs is between 5 m and 10 m, compared to between 150 m and 250 m in the case of the DARPA Subterranean

Challenge. Moreover, before computing the overlap metric for the different methods, a density filter was run on each of the reconstructed maps to uniformize their density and avoid a bias in the evaluations. The mean overlap of the maps built by ICP-SLAM is 77.2%, as opposed to 92.1% for SAAVE-ICP-SLAM. In the case of Point-LIO, the mean overlap of the reconstructed maps is 71.3%. The qualitative result of SAAVE-ICP-SLAM for a specific run is shown in Figure 12. We selected this run since the increase in mapping performance was significant when our SLAM algorithm relied on angular velocities estimated by SAAVE. Additionally, we define a metric of mapping failures such that the percentage of outliers in the reconstructed map (i.e., points farther than 0.25 m from their closest neighbor in the ground-truth map) is above 15%. In the case of ICP-SLAM, we observe a failure of the mapping for 12 out of the 32 runs, as opposed to no failure in the case of SAAVE-ICP-SLAM. For Point-LIO, we observed failure of the SLAM in 19 runs out of the 32. This difference in the number of failures demonstrates that SAAVE increases mapping robustness under aggressive motions, such as those induced by tumbling events.

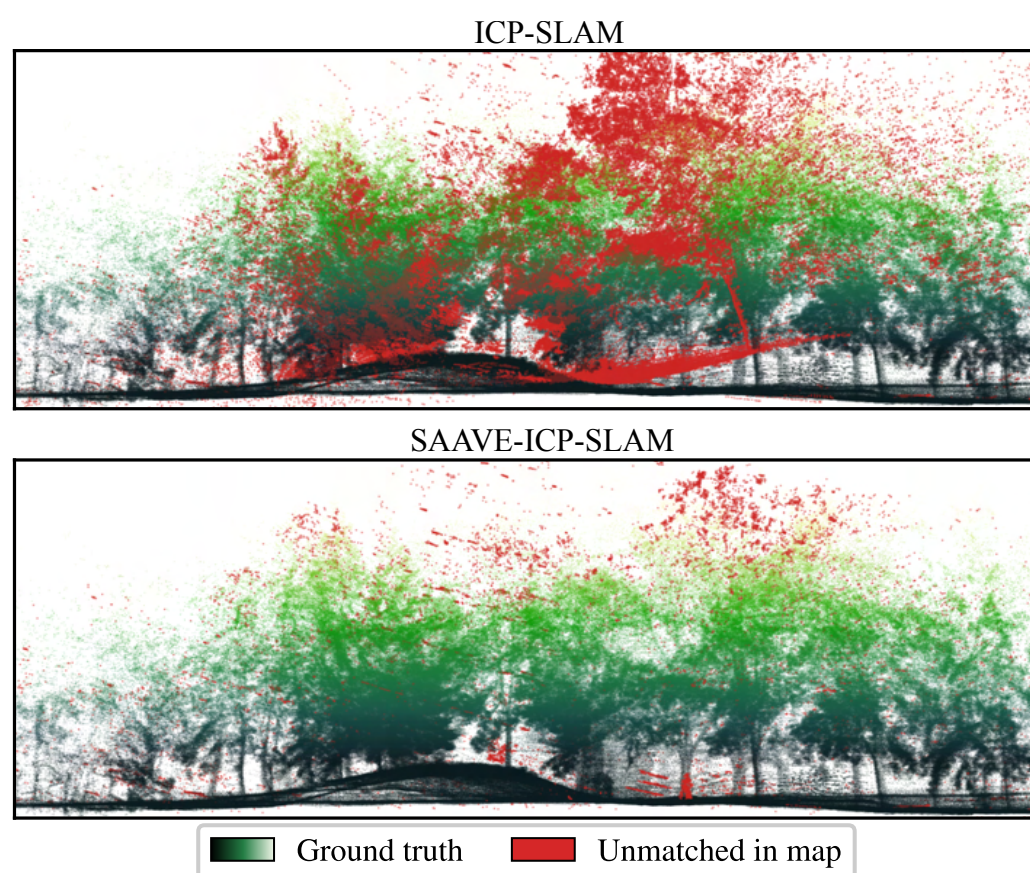


Figure 12. Side view of the ground-truth map built for the TIGS dataset. The color map is proportional to the point height. Mapping outliers from the fourteenth run of the TIGS dataset are displayed in red. **(Top)** A map showing the outliers of ICP-SLAM. **(Bottom)** The same map showing the outliers of SAAVE-ICP-SLAM. A point is considered an outlier if it is farther than 0.25 m from the ground-truth map.

6.4. Improving Trajectory Continuity with Stretch-ICP

In this section, we evaluate the continuity of trajectories reconstructed by our SLAM framework when relying on ICP or Stretch-ICP for point cloud registration. These two variants, referred to as SAAVE-ICP-SLAM and SAAVE-Stretch-SLAM respectively, both incorporate SAAVE angular velocity estimates. We first analyze angular speed estimates on the TIGS dataset by differentiating consecutive orientations and comparing them to the ground-truth angular velocities provided by the VN-100 gyroscope. To assess linear speed

estimation, we rely on the HRMC dataset, which provides accurate ground-truth positions over the full trajectory (as the TIGS dataset provides ground truth only for angular velocity and displacement). As shown in Figure 13, the angular speed error of SAAVE-ICP-SLAM increases with angular speed and exhibits a growing variance. In contrast, SAAVE-Strech-SLAM shows a markedly slower growth in error and maintains a nearly constant standard deviation. This behavior stems from the trajectory discontinuities introduced by ICP at scan boundaries, which induce overshoots in the estimated angular speeds. By enforcing trajectory continuity, Stretch-ICP mitigates these effects, resulting in more accurate and physically consistent motion estimates.

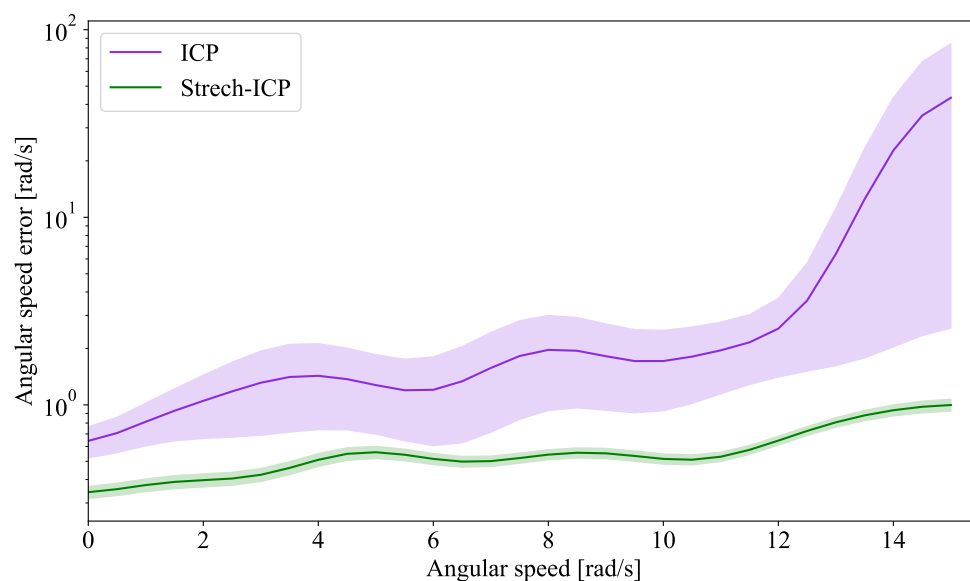


Figure 13. Error on the angular speed derived from estimated trajectories on all runs of the TIGS dataset. In purple is the angular speed error of the trajectory estimated by SAAVE-ICP-SLAM, and in green is the error of the trajectory estimated by SAAVE-Strech-SLAM. The curves represent the mean of the error distributions, and the shaded areas represent the standard deviation around the mean. As can be seen, the SAAVE-ICP-SLAM angular speed error grows significantly faster with angular speed than the error of SAAVE-Strech-SLAM. Note that the angular speed error is on a log scale.

We now compare the trajectory linear speeds of the methods on the HRMC dataset. The error on linear speed derived from the trajectories reconstructed by SAAVE-ICP-SLAM and SAAVE-Strech-SLAM is shown in Figure 14. As can be seen, the distribution of errors on the linear speeds estimated by both frameworks is asymmetric, with the mode of the distribution at 0.17 m/s for SAAVE-Strech-SLAM. In the case of SAAVE-ICP-SLAM, the mode of the distribution is approximately the same (i.e., 0.20 m/s), but represents fewer data points. The major difference between the error distributions of the two methods lies in the number of linear speed errors beyond 3 m/s. In the case of SAAVE-ICP-SLAM, the discontinuities in the estimated trajectory lead to a high number of linear speed errors above 3 m/s. On the other hand, because of the trajectory continuity constraints it enforces, SAAVE-Strech-SLAM has no such discontinuities in its estimated trajectory, and thus fewer outliers in the estimated linear speeds. Specifically, Figure 14 shows that the linear speed error distribution of SAAVE-ICP-SLAM exhibits a substantially heavier tail, with many more large errors, whereas SAAVE-Strech-SLAM strongly suppresses these high-error outliers.

To highlight the difference in how the methods handle continuity between scans, we computed the error of the estimated speeds at the boundaries between scans. The error on these estimated speeds for all runs of the HRMC dataset is shown in Figure 15. As can be

seen, the median linear speed estimation error of SAAVE-ICP-SLAM is at 4.51 m/s and its median angular speed estimation error is at 1.80 rad/s. In the case of SAAVE-Stretch-SLAM, the median of the linear speed error distribution is at 0.24 m/s and the median of the angular speed error distribution is at 0.10 rad/s. This decrease in error represents an improvement of 95.2% for the median linear speed estimation error and an improvement of 94.8% for the median angular speed estimation error.

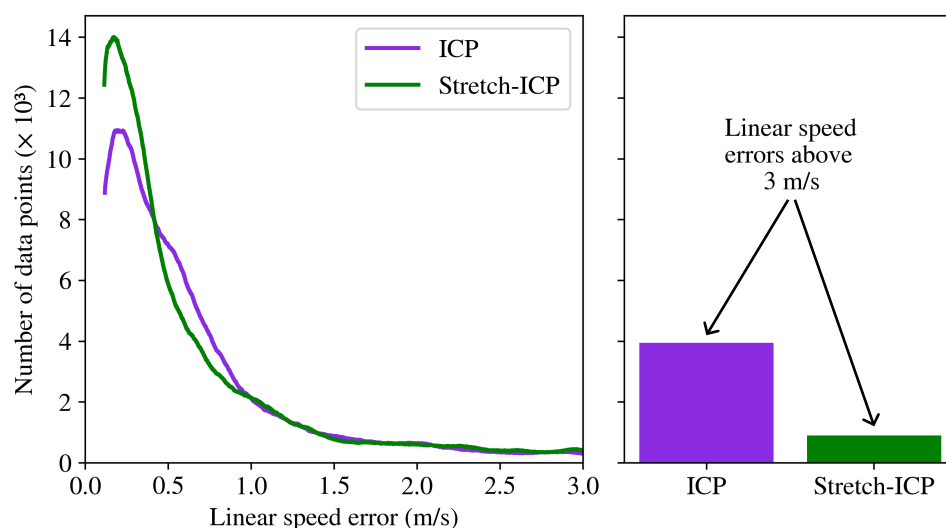


Figure 14. Distribution of errors on the linear speeds derived from trajectories estimated by the different methods on all runs of the HRMC dataset. In purple is the error of the linear speeds estimated by SAAVE-ICP-SLAM, and in green is the error of the linear speeds estimated by SAAVE-Stretch-SLAM. **(Left)** The curves represent the distribution of linear speed errors below 3 m/s. **(Right)** The bars represent the number of linear speed errors greater than 3 m/s for each method.

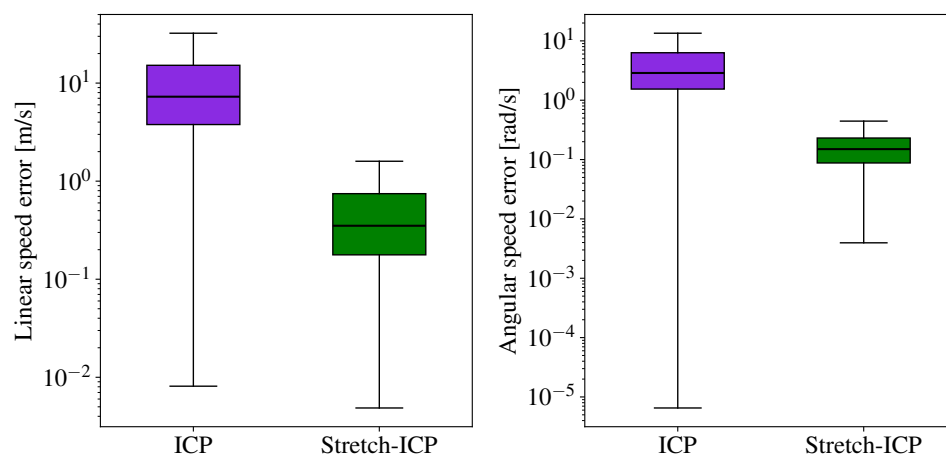


Figure 15. Errors created by discontinuities at the beginning of scans on all runs of the HRMC dataset. **(Left)** The linear speed errors. **(Right)** Angular speed errors. The purple boxes represent the error of SAAVE-ICP-SLAM, and the green boxes represent the error of SAAVE-Stretch-SLAM. The black line indicates the median of the error distributions, and the bottom and top of the boxes represent the first and third quartiles, respectively. Note that the errors are in log scale.

The previous results show that ICP repositions small portions of the trajectory with no regard to trajectory continuity, optimizing local alignment independently across scans. While this behavior can reduce the median localization error over the entire trajectory, as will be shown in the next section, it introduces significant discontinuities at scan boundaries.

These discontinuities lead to physically implausible overshoots in the estimated linear and angular speeds, resulting in substantially higher velocity errors. In contrast, Stretch-ICP explicitly enforces trajectory continuity, producing a smoother and more physically plausible motion estimate. This comes at the cost of a modest increase in final pose error, but yields significantly more reliable linear and angular velocity estimates along the trajectory.

6.5. Limited Impact of Stretch-ICP on SLAM Localization Accuracy

Stretch-ICP enforces continuity constraints during registration to prevent scan-boundary discontinuities and yield more physically consistent motion estimates under aggressive motions. However, it is important to ensure that this does not degrade the localization accuracy of a SLAM pipeline. To this effect, we compare the localization error of our SLAM framework when relying on ICP or Stretch-ICP for point cloud registration. These variations of our SLAM framework, which are referred to as SAAVE-ICP-SLAM and SAAVE-Stretch-SLAM respectively, include our SAAVE angular velocity estimates. The cumulative probability of localization errors for both methods on the TIGS dataset is reported in Figure 16, along with the results of Point-LIO [19] as a reference. As can be seen, the rightward shift of the SAAVE-Stretch-SLAM curve indicates slightly higher localization errors compared to SAAVE-ICP-SLAM. At the same time, the continuity constraints enforced by Stretch-ICP result in a substantial reduction of velocity estimation errors, particularly at scan boundaries, as discussed in Section 6.4. These results highlight a trade-off in which modest increases in final pose error are accompanied by significant gains in trajectory continuity and motion estimation reliability under aggressive motions. For reference, Point-LIO exhibits significantly higher localization errors on the TIGS dataset, highlighting the overall difficulty of the evaluation scenario. Overall, incorporating Stretch-ICP preserves robustness while enforcing trajectory continuity, yielding more physically consistent motion estimates with only a limited impact on final pose accuracy. However, because the TIGS dataset provides ground truth only for the initial and final poses, this dataset offers a partial view of localization performance and does not capture errors accumulated along the full trajectory. As a result, trajectories that differ substantially in realism between SAAVE-ICP-SLAM and SAAVE-Stretch-SLAM, as illustrated in Figure 1, may still produce similar final pose errors on the TIGS dataset.

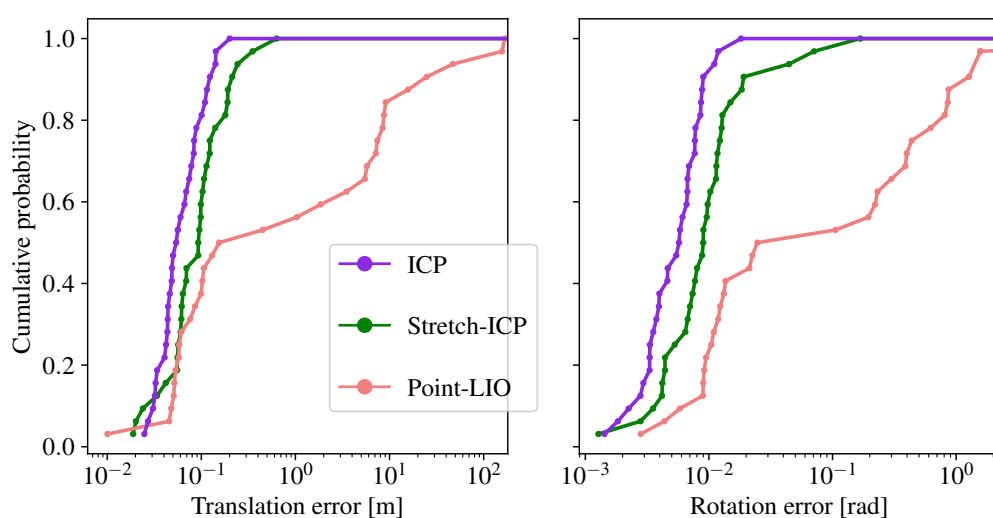


Figure 16. Localization error for all runs in the TIGS dataset. **(Left)** The cumulative probability of observing a given translation error is shown on the left plot. **(Right)** The cumulative probability of observing a rotation error. The purple, green, and pink lines represent the cumulative probability of localization errors for SAAVE-ICP-SLAM, SAAVE-Stretch-SLAM, and Point-LIO, respectively. Note that the errors on the x -axis are in log scale.

7. Conclusions

In this paper, we investigated the robustness of SLAM in the presence of aggressive motions. To this effect, we introduced the TIGS dataset, which consists of 32 runs mimicking a robot tumbling down a hill, with angular speeds reaching up to 18.6 rad/s and linear accelerations up to 157.8 m/s². This dataset, which we released publicly, has angular speeds up to four times higher and linear accelerations up to five times higher than other similar datasets. To increase the robustness of SLAM to these aggressive motions, we proposed a solution in two parts. First, we estimate angular velocity under saturated gyroscope measurements with our novel Saturation-Aware Angular Velocity Estimation (SAAVE) method. Using the TIGS dataset, we demonstrated that SAAVE reduces the angular speed estimation error by 83.4% during gyroscope saturation periods. Second, we presented Stretch-ICP, a novel algorithm to simultaneously register and deskew point clouds in scenarios involving aggressive motions. By enforcing continuity constraints across scans, Stretch-ICP mitigates the discontinuities commonly introduced by classical registration methods. The linear and angular speeds of the trajectories reconstructed by our SLAM system when using this registration algorithm were evaluated on the TIGS and HRMC datasets. This evaluation showed that Stretch-ICP reduced the error by 95.2% on linear speed estimation and by 94.8% on angular speed estimation at scan boundaries, while exhibiting only a limited impact on final pose error on the TIGS dataset.

As discussed previously, it was shown that SAAVE significantly improves the accuracy of angular velocity estimates in the presence of gyroscope saturation. However, even with improved angular velocity estimates, motion prediction remains a challenge in scenarios of aggressive motions. As a result, SLAM frameworks that rely on classical registration methods often produce discontinuous trajectories. Stretch-ICP addresses this by applying smoothness constraints across the entire trajectory, reducing discontinuities, and preventing downstream issues in components such as control algorithms. Therefore, the highest level of robustness is achieved through the combination of SAAVE and Stretch-ICP. Although tumbling is used here as an extreme stress-test scenario, the broader contribution of this work is to improve robustness to sensing and estimation failures that can also arise in practical robotic situations involving impacts, slips, hard landings, collision recovery, and aggressive traversal of uneven terrain.

The primary limitation of SAAVE lies in its computational cost. Due to the GP smoothing it employs, the method is currently not suitable for real-time applications. Reducing this overhead is a key area for future development to enable its integration into time-sensitive autonomous systems. For Stretch-ICP, the main constraint is its localization accuracy when used within a SLAM framework. While the error of Stretch-ICP remains relatively small when compared to the ICP algorithm, it can lead to faster drift accumulation, particularly during the exploration of previously unmapped areas. Future work will focus on optimizing SAAVE for real-time performance and evaluating its impact under tighter computational constraints. For Stretch-ICP, we plan to perform a detailed runtime analysis to assess its computational efficiency relative to classical ICP. Additionally, we aim to conduct a sensitivity analysis of both methods to identify the most influential parameters, which will help guide more effective tuning and deployment strategies.

Author Contributions: Conceptualization, S.-P.D., P.G. and F.P.; methodology, S.-P.D. and F.P.; software, S.-P.D.; validation, S.-P.D.; formal analysis, S.-P.D. and F.P.; investigation, S.-P.D.; resources, F.P. and P.G.; data curation, S.-P.D.; writing—original draft preparation, S.-P.D., V.V., P.G. and F.P.; writing—review and editing, S.-P.D., V.V., P.G. and F.P.; visualization, S.-P.D. and V.V.; supervision, P.G. and F.P.; project administration, P.G. and F.P.; funding acquisition, P.G. and F.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Fonds de recherche du Québec—Nature et technologies (FRQNT) through the Doctoral research scholarships, grant 304410. This research was also supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Canada Graduate Scholarships—Doctoral grant (CGS D) 456424915, through grant CRDPJ 527642-18 SNOW (Self-driving Navigation Optimized for Winter), and through grant ALLRP 585289-23 CRYOTIC (Challenging Robots in Year-around, Outdoor, and Time-critical missions In extreme Conditions).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The TIGS dataset analyzed during the current study is available in the TIGS Dataset repository, https://github.com/norlab-ulaval/Norlab_wiki/wiki/TIGS-Dataset (accessed on 1 February 2026). The HRMC dataset generated and analyzed during the current study is available in the HRMC Dataset repository, https://github.com/norlab-ulaval/Norlab_wiki/wiki/HRMC-Dataset (accessed on 1 February 2026).

Acknowledgments: The authors thank Dominic Baril, Matěj Boxan, and Johann Laconte for their contributions to the conference version of this work. They also thank Olivier Gamache, Jean-Michel Fortin, William Dubois, William Deschênes, and Sandrine Fauteux for their assistance with the experimental work. Finally, the authors thank Effie Daum for her contributions to the development of the methods.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Nagatani, K.; Kiribayashi, S.; Okada, Y.; Otake, K.; Yoshida, K.; Tadokoro, S.; Nishimura, T.; Yoshida, T.; Koyanagi, E.; Fukushima, M.; et al. Emergency response to the nuclear accident at the Fukushima Daiichi Nuclear Power Plants using mobile rescue robots. *J. Field Robot. (JFR)* **2013**, *30*, 44–63.
2. Dilaveroğlu, L.; Özcan, O. MiniCoRe: A Miniature, Foldable, Collision Resilient Quadcopter. In Proceedings of the 3rd IEEE International Conference on Soft Robotics (RoboSoft), New Haven, CT, USA, 15 May–15 July 2020; pp. 176–181.
3. Ebadi, K.; Bernreiter, L.; Biggie, H.; Catt, G.; Chang, Y.; Chatterjee, A.; Denniston, C.E.; Deschênes, S.P.; Harlow, K.; Khattak, S.; et al. Present and Future of SLAM in Extreme Environments: The DARPA SubT Challenge. *IEEE Trans. Robot. (T-RO)* **2023**, *40*, 936–959.
4. Williams, G.; Drews, P.; Goldfain, B.; Rehg, J.M.; Theodorou, E.A. Information-Theoretic Model Predictive Control: Theory and Applications to Autonomous Driving. *IEEE Trans. Robot. (T-RO)* **2018**, *34*, 1603–1622.
5. Deschênes, S.P.; Baril, D.; Kubelka, V.; Giguère, P.; Pomerleau, F. Lidar Scan Registration Robust to Extreme Motions. In *Proceedings of the 18th Conference on Robots and Vision (CRV)*; IEEE: Piscataway, NJ, USA, 2021; pp. 17–24.
6. Lee, J.; Kim, H.; Oh, S.H.; Do, J.C.; Nam, C.W.; Hwang, D.H.; Lee, S.J. Angular velocity estimation of rotating plate using extended Kalman filter with accelerometer bias model. *Microsyst. Technol.* **2019**, *25*, 2855–2867.
7. Shan, T.; Englot, B.; Meyers, D.; Wang, W.; Ratti, C.; Rus, D. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Online, 25–29 October 2020; pp. 5135–5142.
8. Reinke, A.; Palieri, M.; Morrell, B.; Chang, Y.; Ebadi, K.; Carlone, L.; Agha-Mohammadi, A.A. LOCUS 2.0: Robust and Computationally Efficient Lidar Odometry for Real-Time 3D Mapping. *IEEE Robot. Autom. Lett. (RA-L)* **2022**, *7*, 9043–9050.
9. Xu, W.; Cai, Y.; He, D.; Lin, J.; Zhang, F. FAST-LIO2: Fast Direct LiDAR-Inertial Odometry. *IEEE Trans. Robot. (T-RO)* **2022**, *38*, 2053–2073.
10. Chen, K.; Nemiroff, R.; Lopez, B.T. Direct LiDAR-Inertial Odometry: Lightweight LIO with Continuous-Time Motion Correction. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 3983–3989.
11. Gill, W.A.; Howard, I.; Mazhar, I.; McKee, K. A review of MEMS vibrating gyroscopes and their reliability issues in harsh environments. *Sensors* **2022**, *22*, 7405.
12. Yazdi, N.; Ayazi, F.; Najafi, K. Micromachined inertial sensors. *Proc. IEEE* **1998**, *86*, 1640–1659.
13. Passaro, V.M.; Cuccovillo, A.; Vaiani, L.; De Carlo, M.; Campanella, C.E. Gyroscope technology and applications: A review in the industrial perspective. *Sensors* **2017**, *17*, 2284.

14. Liu, Y.; Li, Z.; Zheng, S.; Cai, P.; Zou, X. An evaluation of MEMS-IMU performance on the absolute trajectory error of visual-inertial navigation system. *Micromachines* **2022**, *13*, 602.
15. Zhao, C.; Hu, K.; Xu, J.; Zhao, L.; Han, B.; Wu, K.; Tian, M.; Yuan, S. Adaptive-lio: Enhancing robustness and precision through environmental adaptation in lidar inertial odometry. *IEEE Internet Things J.* **2024**, *12*, 12123–12136.
16. Dellenbach, P.; Deschaud, J.E.; Jacquet, B.; Goulette, F. CT-ICP: Real-time Elastic LiDAR Odometry with Loop Closure. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 5580–5586.
17. Ardeshiri, R.R.; Gheisarnejad, M.; Tavan, M.R.; Vafamand, N.; Khooban, M.H. A robust intelligent controller-based motion control of a wheeled mobile robot. *Trans. Inst. Meas. Control.* **2022**, *44*, 2911–2918.
18. Deschênes, S.P.; Baril, D.; Boxan, M.; Laconte, J.; Giguère, P.; Pomerleau, F. Saturation-Aware Angular Velocity Estimation: Extending the Robustness of SLAM to Aggressive Motions. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 10711–10718.
19. He, D.; Xu, W.; Chen, N.; Kong, F.; Yuan, C.; Zhang, F. Point-LIO: Robust High-Bandwidth Light Detection and Ranging Inertial Odometry. *Adv. Intell. Syst.* **2023**, *5*, 2200459.
20. Segal, A.; Haehnel, D.; Thrun, S. Generalized-ICP. In Proceedings of the Robotics: Science and Systems (RSS) V. Robotics: Science and Systems Foundation, Seattle, WA, USA, 28 June–1 July 2009; p. 435.
21. Palieri, M.; Morrell, B.; Thakur, A.; Ebadi, K.; Nash, J.; Chatterjee, A.; Kanellakis, C.; Carlone, L.; Guaragnella, C.; Aghamohammadi, A.a. LOCUS: A Multi-Sensor Lidar-Centric Solution for High-Precision Odometry and 3D Mapping in Real-Time. *IEEE Robot. Autom. Lett. (RA-L)* **2021**, *6*, 421–428.
22. Dang, Q.; Suh, Y. Sensor Saturation Compensated Smoothing Algorithm for Inertial Sensor Based Motion Tracking. *Sensors* **2014**, *14*, 8167–8188.
23. Tan, C.H.; bin Shaiful, D.S.; Tang, E.; Khaw, J.Y.; Soh, G.S.; Foong, S. Flydar: Magnetometer-based High Angular Rate Estimation during Gyro Saturation for SLAM. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Online, 31 May–31 August 2020; pp. 8532–8537.
24. Silic, M.; Mohseni, K. Correcting Current-Induced Magnetometer Errors on UAVs: An Online Model-Based Approach. *IEEE Sens. J.* **2020**, *20*, 1067–1076.
25. Pachter, M.; Welker, T.C.; Huffman, R.E. Gyro-free INS Theory. *Navig. J. Inst. Navig.* **2013**, *60*, 85–96.
26. Anderson, S.; Barfoot, T.D. Full STEAM Ahead: Exactly Sparse Gaussian Process Regression for Batch Continuous-Time Trajectory Estimation on SE(3). In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 157–164.
27. Gentil, C.L.; Vidal-Calleja, T.; Huang, S. IN2LAAMA: Inertial Lidar Localization Autocalibration and Mapping. *IEEE Trans. Robot. (T-RO)* **2021**, *37*, 275–290.
28. Zhang, J.; Singh, S. LOAM: Lidar Odometry and Mapping in Real-time. In Proceedings of the Robotics: Science and Systems (RSS) X. Robotics: Science and Systems Foundation, Berkeley, CA, USA, 12–16 July 2014; Volume 41, pp. 401–416.
29. Le Gentil, C.; Vidal-Calleja, T.; Huang, S. 3D Lidar-IMU Calibration Based on Upsampled Preintegrated Measurements for Motion Distortion Correction. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 2149–2155.
30. Park, C.; Moghadam, P.; Williams, J.; Kim, S.; Sridharan, S.; Fookes, C. Elasticity Meets Continuous-Time: Map-Centric Dense 3D LiDAR SLAM. *IEEE Trans. Robot. (T-RO)* **2022**, *38*, 978–997.
31. Whelan, T.; Leutenegger, S.; Salas-Moreno, R.F.; Glocker, B.; Davison, A.J. ElasticFusion: Dense SLAM Without a Pose Graph. In Proceedings of the Robotics: Science and Systems (RSS) XI, Rome, Italy, 13–17 July 2015; Volume 11, p. 3.
32. Lang, X.; Chen, C.; Tang, K.; Ma, Y.; Lv, J.; Liu, Y.; Zuo, X. Coco-LIC: Continuous-Time Tightly-Coupled LiDAR-Inertial-Camera Odometry Using Non-Uniform B-Spline. *IEEE Robot. Autom. Lett. (RA-L)* **2023**, *8*, 7074–7081.
33. Ramezani, M.; Wang, Y.; Camurri, M.; Wisth, D.; Mattamala, M.; Fallon, M. The Newer College Dataset: Handheld LiDAR, Inertial and Vision with Ground Truth. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; pp. 4353–4360.
34. Zhang, L.; Helmberger, M.; Fu, L.F.T.; Wisth, D.; Camurri, M.; Scaramuzza, D.; Fallon, M. Hilti-Oxford Dataset: A Millimeter-Accurate Benchmark for Simultaneous Localization and Mapping. *IEEE Robot. Autom. Lett. (RA-L)* **2023**, *8*, 408–415.
35. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
36. Karapetkov, S.; Uzunov, H.; Dechkova, S.; Uzunov, V. Solving Cauchy Problem for Modelling the Dynamics of Vehicle-Fixed Barrier Collisions by the Finite Element Method and the Effect of Forces of Inertia on Passengers. *Proc. Eng.* **2023**, *5*, 667–678.
37. Tang, T.Y.; Yoon, D.J.; Barfoot, T.D. A White-Noise-on-Jerk Motion Prior for Continuous-Time Trajectory Estimation on SE(3). *IEEE Robot. Autom. Lett. (RA-L)* **2019**, *4*, 594–601.

38. Baril, D.; Deschênes, S.P.; Gamache, O.; Vaidis, M.; LaRocque, D.; Laconte, J.; Kubelka, V.; Giguère, P.; Pomerleau, F. Kilometer-scale autonomous navigation in subarctic forests: challenges and lessons learned. *Field Robot.* **2022**, *2*, 1628–1660.
39. Pomerleau, F.; Colas, F.; Siegwart, R. A Review of Point Cloud Registration Algorithms for Mobile Robotics. *Found. Trends Robot.* **2015**, *4*, 1–104.
40. Forster, C.; Carlone, L.; Dellaert, F.; Scaramuzza, D. IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation. In Proceedings of the Robotics: Science and Systems (RSS) XI, Rome, Italy, 13–17 July 2015.
41. Pomerleau, F.; Colas, F.; Siegwart, R.; Magnenat, S. Comparing ICP Variants on Real-World Data Sets. *Auton. Robot.* **2013**, *34*, 133–148.
42. Hoppe, H.; DeRose, T.; Duchamp, T.; McDonald, J.; Stuetzle, W. Surface reconstruction from unorganized points. In Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, Chicago, IL, USA, 26–31 July 1992; pp. 71–78.
43. Phillips, J.M.; Liu, R.; Tomasi, C. Outlier Robust ICP for Minimizing Fractional RMSD. In *Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM)*; IEEE: Piscataway, NJ, USA, 2007; pp. 427–434.
44. Chen, Y.; Medioni, G. Object Modeling by Registration of Multiple Range Images. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: Piscataway, NJ, USA, 1991; pp. 2724–2729.
45. Vizzo, I.; Guadagnino, T.; Mersch, B.; Wiesmann, L.; Behley, J.; Stachniss, C. KISS-ICP: In Defense of Point-to-Point ICP—Simple, Accurate, and Robust Registration If Done the Right Way. *IEEE Robot. Autom. Lett. (RA-L)* **2023**, *8*, 1029–1036.
46. Haider, A.; Haas, L.; Koyama, S.; Elster, L.; Köhler, M.H.; Schardt, M.; Zeh, T.; Inoue, H.; Jakobi, M.; Koch, A.W. Modeling of Motion Distortion Effect of Scanning LiDAR Sensors for Simulation-Based Testing. *IEEE Access* **2024**, *12*, 13020–13036.
47. Chung, T.H.; Orekhov, V.; Maio, A. Into the Robotic Depths: Analysis and Insights from the DARPA Subterranean Challenge. *Annu. Rev. Control. Robot. Auton. Syst.* **2023**, *6*, 477–502.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.