Robust Localization, Mapping, and Navigation for Quadruped Robots

Dyuman Aditya[1] , Junning Huang[2] , Nico Bohlinger[2] , Piotr Kicki[3,4] , Krzysztof Walas[3] ,
Jan Peters[2,5,6] , Matteo Luperto[7] and Davide Tateo[2]

Abstract— Quadruped robots are currently a widespread platform for robotics research, thanks to powerful Reinforcement Learning controllers and the availability of cheap and robust commercial platforms. However, to broaden the adoption of the technology in the real world, we require robust navigation stacks relying only on low-cost sensors such as depth cameras. This paper presents a first step towards a robust localization, mapping, and navigation system for low-cost quadruped robots. In pursuit of this objective we combine contact-aided kinematic, visual-inertial odometry, and depth-stabilized vision, enhancing stability and accuracy of the system. Our results in simulation and two different real-world quadruped platforms show that our system can generate an accurate 2D map of the environment, robustly localize itself, and navigate autonomously. Furthermore, we present in-depth ablation studies of the important components of the system and their impact on localization accuracy. Videos, code, and additional experiments can be found on the project website.[1]
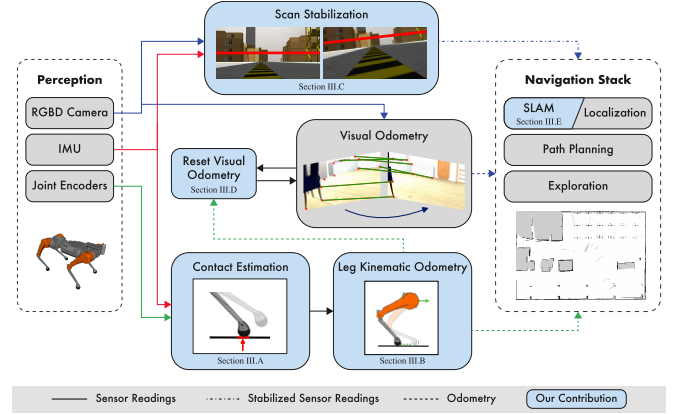
Fig. 1. Overall system design for robust navigation in low-cost quadruped robots. Our contributions are highlighted in light blue, different colors indicate different types of data.

I. INTRODUCTION

The widespread availability of robust and reliable robotics platforms, and the major advances in locomotion controllers based on Reinforcement Learning [1], [2], [3], [4], [5], [6], has enabled the deployment of quadruped robots in challenging environments, both outdoor and indoor [7], [8]. Concurrently, there is an increasing interest in quadruped robots from the industry for surveillance, inspection, and search and rescue tasks, owing to their ability to navigate diverse settings—including stairs, cluttered spaces, and uneven terrain.

Unfortunately, practical industrial solutions typically rely on expensive robotic platforms equipped with high-precision sensors that enable robust autonomous navigation and mapping. In low-cost platforms, however, these sensors are not available, and most of the localization capabilities must rely on inexpensive depth image cameras or 2D LiDARs, resulting in subpar performance. These issues are further exacerbated when employing policies learned with Reinforcement Learning, which can achieve very high speeds and reactive behavior, at the expense of trajectory smoothness. Such rapid trajectories may disrupt the SLAM pipeline, resulting in loss of localization, broken maps, and, consequently, the inability to navigate autonomously in the environment. These challenges are even more relevant when quadruped robots operate in indoor environments, thus in settings where fast movements change the robot's perceptions rapidly; in those environments, even the availability of expensive long-range LiDARS has a limited impact on performance, due to the proximity of obstacles to the robot's location.

In this paper, we focus on designing a simple and robust localization pipeline that enables low-cost quadruped robots to navigate autonomously in complex environments, thereby empowering these robots to tackle interesting real-world tasks. Our approach targets scenarios where the robot is equipped with a low-cost RGBD camera and an IMU. Due to the type and quality of sensors available for our target set-up, we restrict our localization and navigation pipeline to indoor settings. These environments, while being one of the target environments for quadruped robots, are particularly challenging as fast robot movements can cause abrupt changes in robot perception, thus causing localization and mapping issues. Moreover, we focus on localization, navigation, and mapping using 2D grid maps. Those maps are commonly used for path planning, localization, and path execution since widespread methods for 2D map-based navigation are still de facto the state-of-the-art for autonomous navigation [9], and since 2D maps can be combined with machine learning techniques and 3D and semantic maps [10].

In a 2D mapping setting, an occupancy map is generated using LiDAR scans or from depth data obtained by an RBGD camera [11]. While setup is widely used in classical wheeled mobile platforms with stable performance, its implementation with quadruped robots is still challenging, and

[1] École Centrale de Nantes, France
`dyuman.aditya@eleves.ec-nantes.fr`
[2] Technical University of Darmstadt, Germany
[3] Poznan University of Technology, Poland
[4] IDEAS NCBR, Warsaw, Poland
[5] German Research Center for AI (DFKI), SAIROL
[6] Hessian.AI
[7] Università degli Studi di Milano, Italy
[1] https://sites.google.com/view/low-cost-quadruped-slam

few robust solutions have been proposed yet. Due to rough, uneven, and slippery terrain, robot perception data are subject to jitters and noise, making 2D mapping and localization highly unreliable. Scans oscillate according to the terrain and the robot's own movement, especially when the movement involves sharp changes in roll and pitch angles. Several works used 3D map representation and vision-based methods for SLAM with quadruped robots. Unfortunately, while 3D maps can provide a more semantically rich representation of the environment, they are still more computationally expensive to obtain and maintain. Moreover, when low-cost sensors are used, 3D maps are heavily affected by noise, making them impractical for navigation.

To enhance system autonomous navigation performance, we exploit joint encoder data to compute leg odometry [12]. This integration requires reliable contact information, typically obtained from contact sensors. However, these sensors are not always available or sufficiently accurate on legged platforms. Recognizing that contact sensors are not always available or sufficiently accurate on legged robotic platforms, we also demonstrate how to estimate foot contacts using only joint torque measurements. Additionally, we incorporate a scan stabilization technique to mitigate the impact of rapid, dynamic movements on the SLAM pipeline, ensuring more robust and accurate localization. The key contributions of this paper are i) a detailed description of the localization system, illustrating how it is possible improve localization and navigation performance integrating standard pipelines with specific problem-dependent solutions; our system involves a) an estimation of legged odometry and b) contact estimation, c) scan stabilization, and d) integration of leg-visual-inertial odometry to improve visual odometry. Then, we provide (ii) an extensive ablation study of the impact of each component of the localization system in simulation in several downstream tasks such as mapping, navigation, and exploration. Finally, we present (iii) an experimental real-world evaluation including a robot with an actuated spine.

II. RELATED WORKS

Quadruped robots, similarly to other autonomous mobile robots, need localization, mapping, and autonomous navigation capabilities [13]. While many different systems with such capabilities have been presented in literature, most of them rely on expensive robotic platforms, such as the Anymal and the Spot robots, mounting high-end, expensive long-distance sensors, thanks to their large form factor, allowing heavier payloads. An example of such systems can be found in [14] for the Anymal and in [15] for the Spot robot. Indeed, in [15], the authors make a step towards low-end platforms by reducing the total amount of sensors used by the system. However, their setup is still relatively expensive compared with the setup considered in this paper. On top of this, we consider much more reactive and fast neural-based policies. A similar setup is used in [13], where a long-distance 3D LiDAR and RGBD cameras are used for 3D SLAM in outdoor environments, while the recent work

of Li et al. [16] presents a framework to perform 3D SLAM integrating features extracted from a long-distance LiDAR.

In [17], the authors describe a framework designed to solve the task of autonomous exploration for object search, where a Spot robot is equipped with a Velodyne Puck LiDAR and an Azure Kinect RGBD camera. The robot plans its action using a 2D grid map by integrating a classifier to detect areas left to explore (labeled as *non-map*) where the target object is located. Interestingly, even if a long-range expensive LiDAR is used, since acquiring mapping data for training models is difficult with quadruped robots due to their mechanical limitations, a remotely controlled wheeled cart with the same sensor equipment as the quadruped robot is used instead for data acquisition.

The authors of [18] design an autonomy package that incorporates 3D LiDAR and IMU data for autonomous navigation of a Spot robot in GPS-constrained settings. The robot navigation is performed using a 2D occupancy map of the environment; to reduce the difficulty of building 2D maps of indoor environments with quadruped robots, a first run is made by manually moving the robot within the environment; the recorded data are later used to compute a 2D map, offline. In later runs, the map is used online for localization.

In [12], the authors compute Leg Odometry (LO) by modeling the forward kinematics and fuse it with IMU data to correct the robot position among different keyframes. To stabilize the perception, dynamic and moving objects are filtered from camera data using computer vision methods. As a result of this, they provide a robust visual-slam method for quadruped robots in dynamic environments with visually challenging conditions. Similarly, Kumar et al. [19] improve visual-intertial SLAM by exploiting the periodic predictability in the motion of legged robots by performing visual SLAM separately on each portion of the gait circle, thus leveraging a model of how quadruped robots move.

The work of [20] investigates how the combination of different sources of odometry, such as laser odometry (LO) and laser-inertial odometry (LIO), reduces the uncertainty in 2D and 3D SLAM that is due to the high noise of legged odometry derived from kinematics in quadruped robots. To do so, they estimate LIO from 3D maps, using a long-range (70 meters) LiDAR and creating both a 2D grid map and a 3D point cloud map of the environment.

Bouman et al. [21] present a system for the large-scale and long-duration mapping and exploration of a GPS-denied indoor environment using a Spot Robot. The proposed setup is based on five custom RealSense RGBD cameras and a high-end long-range LiDAR. Different sources of odometry, such as kinematic-based odometry (KO) and LO, are integrated using an odometry multiplexer so that the noise in different sources of odometry can be compensated by integrating different inputs. Even if the robot is designed to work in a multi-floor environment, a 2D grid map is built and used for local and global path planning and navigation.

In [22], the authors present a system that uses a 2D map to improve the process of building a 3D map of the same environment. Using a long-range LiDAR sensor, a 2D map is

used for navigation and to correct the expected localization within the 3D map, while the 3D map process uses LO and LIO derived from the 2D map. The decoupling of the process of 2D map-based navigation and 3D scene reconstruction allows focusing on the accuracy of the 3D reconstruction while having a robust 2D map.

Miller et al. [23] present a system for mapping large-scale GPS-denied environments using quadruped robots. The robot is equipped with a long-range LiDAR and a RealSense RGBD camera.

Fahmi et al. [24] present a terrain-aware locomotion framework for quadruped robots, where both proprioceptive and exteroceptive sensor readings are used to plan the feet contact location and the feet motion, and to plan the overall motion of the robot. This information is used to evaluate if the planned motion of the robot can cause a collision of a leg or place the robot's foot in an unstable position; the result of this model is used to improve motion control.

In [25], the authors show how robot perception about reinforcement learning can be used to allow quadruped robots to navigate on risky terrains with dynamic motion. However, the system assumes perfect perception and uses a ground truth map for state estimation. A similar work, where a model of the environment is used for local navigation with sensor constraints is presented in [26].

Recently, Wasserman et al. [27] developed a method to estimate the robot odometry of a quadruped robot by integrating previous action and command velocities, IMUs, proprioceptive sensing, roll and pitch estimates, and the angles and velocities of each joint of the robot. These inputs are processed by a neural network tasked to provide a probabilistic estimate of the robot's translation and rotation. Finally, several works proposed different approaches to integrate heterogeneous proprioceptive and exteroceptive sensor readings, including IMUs, joint encoders, contact sensors, force/torque measurements, camera, LiDAR, GPS, and motion capture, for state estimation in quadruped robots [28], [29], [30], [31].

III. SYSTEM DESIGN

We present a system for robust localization, mapping, and navigation for quadruped robots in challenging environments by combining advances in visual-inertial odometry, contact-aided kinematic odometry, and an IMU-stabilized laser setup for mapping. The overall system is described in Fig. 1, where our contribution is highlighted in light blue. We introduce two key modules, one that performs legged-inertial odometry while also doing contact estimation (Section III-A-III-B), and another that generates stable scans from depth camera measurements and IMU attitude measurements (Section III-C). We then integrate Visual-Inertial Odometry (VIO), Legged-Inertial Odometry (LIO), and laser-stabilized scans into existing 2D localization frameworks that we adapt to the particular settings of quadruped navigation (Section IV-B). Furthermore, we use the leg odometry information to restart the visual odometry system when it gets lost (Section III-D). This allows us to maintain a reasonable pose estimate

even when the robot performs fast and/or abrupt movement, causing a loss of visual feature tracking (Section III-E).

A. Contact estimation

Low-cost quadruped robots are often not equipped with contact estimation sensors, as in the Dingo quadruped, and the RealAnt robot [32]. In the absence of external force sensing, we use a contact state detector based on the classic generalized momentum (GM) disturbance observer [33]. The GM-based observer is proposed as follows

$$\begin{bmatrix} \dot{\hat{\mathbf{p}}} \\ \dot{\hat{\mathbf{f}}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & -\mathbf{J}^T \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{p}} \\ \hat{\mathbf{f}} \end{bmatrix} + \begin{bmatrix} \bar{\boldsymbol{\tau}} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{L}_1(\mathbf{p} - \hat{\mathbf{p}}) \\ \mathbf{L}_2(\mathbf{p} - \hat{\mathbf{p}}) \end{bmatrix},$$

with $\mathbf{p} = \mathbf{M}(\mathbf{x})\mathbf{v}$, the measured generalized momentum, $\mathbf{M}$ denotes the joint space mass matrix, $\hat{\mathbf{p}}$ the estimated generalized momentum, and $\hat{\mathbf{f}} \in \mathbb{R}^{12}$ the measured contact force for four legs. $\bar{\boldsymbol{\tau}} = \boldsymbol{\tau}_m + \mathbf{C}^T\mathbf{v} - \mathbf{g}$, where $\boldsymbol{\tau}_m$ are the motor torques, $\mathbf{C}$ the Coriolis matrix, $\mathbf{g}$ the gravity, and $\mathbf{L}_1, \mathbf{L}_2$ the observer gains.

Since our goal is to detect contact states and not to reconstruct the full force vector, we impose two constraints on the observer: we estimate the force only on the z axis, and clip the estimated force to be always positive. Moreover, to alleviate the phase-lag inherent in high-gain disturbance observers, we introduce a mixed-mode observer that blends high-gain and sliding-mode strategies. This design requires only a global incremental affine bound on the force-signal nonlinearities—a strictly weaker assumption than that of pure high-gain observers [34]. Specifically, the observer dynamics are as follows

$$\begin{bmatrix} \dot{\hat{\mathbf{p}}} \\ \dot{\hat{\mathbf{f}}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & -\mathbf{J}^T \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{p}} \\ \hat{\mathbf{f}} \end{bmatrix} + \begin{bmatrix} \bar{\boldsymbol{\tau}} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{L}k_1(\mathbf{p} - \hat{\mathbf{p}}) \\ \mathbf{L}^2 k_2(\mathbf{p} - \hat{\mathbf{p}}) \end{bmatrix},$$

with $\hat{\mathbf{f}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \hat{f}_z \end{bmatrix}$ the estimated contact force on only z axis, and k_1, k_2 are defined as

$$k_1(s) \coloneqq q(s) \qquad k_2(s) \coloneqq \text{sign}(s) + q(s)$$

with $q(s) \coloneqq \text{sign}(s)|s|^{1/2} + s$, $\text{sign}(s) = 1$ if $s > 0$ and $\text{sign}(s) = -1$ if $s < 0$, $\mathbf{L}$ the gain parameter. To obtain the contact states, we filter the estimated contact force to reduce oscillations during the steady state period, and we use feet-specific thresholds to determine the contact states.

B. Legged Odometry as Least Squares

Using leg odometry [35], we can estimate the twist of the robot body, i.e., linear and angular velocities, as a function of the joint velocities of the legs in contact with the ground. This is given as an equation of the form

$$\mathbf{v}_b^l = -\omega_b \times \mathbf{p}_l - \mathbf{v}_l, \tag{1}$$

with $\mathbf{v}_b^l$ the velocity of the body due to leg l, ω_b the angular velocity of the body, $\mathbf{p}_l$ and $\mathbf{v}_l$ the position and velocity (in body frame) respectively of the leg in contact. Notice that the equation above represents a 3DoF linear constraint on the 6DoF robot twists. Thus, we require at least two linearly independent constraints to completely identify the twist of

the robot body, resulting in a system of linear constraints. To ensure that we always have at least two linearly independent constraints and improve the condition number of the linear system, we add the angular velocity measured by the IMU sensor as an additional constraint. The resulting system of constraints can be written as

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{A}_{\text{IMU}} \end{bmatrix} \mathbf{V}_b = \begin{bmatrix} \mathbf{b} \\ \boldsymbol{\omega}_{\text{IMU}} \end{bmatrix} \tag{2}$$

with the twist of the body $\mathbf{V}_b$, the IMU angular velocity measurements $\boldsymbol{\omega}_{\text{IMU}}$, and the matrices $\mathbf{A} \in \mathbb{R}^{3 \times 6}$, $\mathbf{A}_{\text{IMU}} \in \mathbb{R}^{3N \times 6}$, and $\mathbf{b} \in \mathbb{R}^{3N}$ defined as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_N \end{bmatrix} \quad \mathbf{A}_{\text{IMU}} = \begin{bmatrix} 0_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_N \end{bmatrix}$$

where each block $\mathbf{A}_i$ and $\mathbf{b}_i$ represent a single constraint and can be written as

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{I}_{3 \times 3} & -\mathbf{S}(\mathbf{p}_i) \end{bmatrix}$$
$$\mathbf{b}_i = -\mathbf{v}_i = -\mathbf{J}_i \dot{\boldsymbol{\theta}}_i$$

where $\mathbf{S}(\mathbf{p}_i)$ is the skew-symmetric matrix representation of the position of the i-th leg $\mathbf{p}_i$, while $\mathbf{J}_i$ and $\dot{\boldsymbol{\theta}}_i$ are the Jacobian and the joint velocities of the i-th leg.

We can find a solution to this linear system as a least squares problem

$$\hat{\mathbf{V}}_b = \arg \min_{\mathbf{V}_b} \|\mathbf{A}\mathbf{V}_b - \mathbf{b}\|^2 . \tag{3}$$

The problem described in (3) can be solved in closed form as follows

$$\hat{\mathbf{V}}_b = \left(\mathbf{A}^\top \mathbf{A} \right)^{-1} \mathbf{A}^\top \mathbf{b} \tag{4}$$

C. Scan Stabilization

In this Section, we show how we can exploit the information from depth images and the robot's orientation to generate a stabilized scan. To this end, we use the onboard IMU to select a slice of the image data, making it invariant to jitters and significant changes to roll and pitch angles. In this way, we can obtain more stable 2D depth readings that can be used for 2D mapping, localization, and, ultimately, navigation. An example of the effect of the scan stabilization can be seen in Fig. 1 and an example of the result in Fig. 2.

We extract the slice of image data along a line of pixels whose slope is determined by the roll r angle and whose intercept is determined by the pitch p angle. A standard width of pixels above and below this line is combined to form the depth scan. This allows us to compensate for any rotation along the x and y axes. The slope of the reference line is easily computed directly from the roll angle. However, the intercept position is slightly more complex to compute as it depends on these two quantities. We derive an expression for the intercept position in pixels Δv for an object's image in response to a pitch rotation θ of the camera.

Assuming that the z-axis of the camera frame lies along the camera's optical axis, a pitch rotation along the y-axis

in the robot frame corresponds to a roll rotation along the x-axis in the camera frame. Therefore, in the camera frame, the robot's pitch rotation is represented by the following roll rotation matrix:

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

For simplicity, we assume the initial coordinates of an object in the camera's frame are $(X, Y, Z) = (0, 0, Z)$, meaning the object is positioned along the optical axis at depth Z. After applying the rotation θ, the new coordinates of the object in the camera's frame are:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R_x(\theta) \cdot \begin{bmatrix} 0 \\ 0 \\ Z \end{bmatrix} = \begin{bmatrix} 0 \\ -\sin\theta \cdot Z \\ \cos\theta \cdot Z \end{bmatrix}$$

Let K be the camera's intrinsic matrix

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

where f_x and f_y are the focal lengths in pixels along the x- and y-axes, respectively, and (c_x, c_y) represents the principal point in pixel coordinates. Using the pinhole camera model, the 3D point in the camera frame projects onto the image plane as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \cdot \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \\ 1 \end{bmatrix}$$

Substituting $X_c/Z_c = 0$ and $Y_c/Z_c = -\tan\theta$, we obtain:

$$u = c_x + f_x \cdot 0 = c_x$$
$$v = c_y + f_y \cdot (-\tan\theta)$$

The vertical shift in pixels $\Delta v = v - v_0 = -f_y \cdot \tan\theta$ due to the pitch angle θ is the difference between the new v coordinate and the initial $v_0 = c_y$. This gives us the intercept value where the reference line equation as a function of pixel coordinates becomes

$$v = (u - x_c) \tan(-r) - f_y \tan(p), \tag{5}$$

where (u, v) represents the pixel coordinates, with the origin being on the top left, x_c is the x coordinate center of the image, and f_y represents the camera's focal length. Combining the pixels above and below this line, we get a stabilized scan using only depth and IMU data.

D. Integrating leg odometry with visual inerital odometry

When tracking is lost, we leverage the computed leg odometry to restart the visual odometry system. Due to the abrupt and jerky nature of quadruped locomotion compared to wheeled systems, sudden pose changes are frequent. This can cause visual odometry systems to lose track when consecutive frames lack sufficient feature correspondence. Such systems are typically reset to the last computed pose; however, with fast-moving platforms like ours, this approach often lacks accuracy and degrades localization performance.

To mitigate this issue, we integrate the last available pose with the body velocity computed by the leg odometry system over small time intervals, thereby generating a more accurate pose to reinitialize visual odometry. This mechanism is triggered when no correspondences are identified between two subsequent frames.

E. Integrating leg odometry with 2D SLAM

Finally, we use leg odometry to impose velocity constraints on the 2D SLAM factor graph. 2D SLAM factor graphs are typically formed with pose constraints based on laser scans. Loop closures occur, and the factor graph is optimized when scans are matched with ones nearby. However, when quadruped robots are used, due to noise and jitter that result from their fast motion, laser scans are often particularly difficult to match, thus affecting both the quality of mapping and localization and ultimately not allowing the robot to robustly navigate in challenging environments. To address this issue, in addition to the scan pose constraints, we add a velocity constraint between consecutive poses, thereby restricting sudden position changes due to mismatched scans. This ensures that the optimized poses remain consistent with the robot's estimated body velocity between poses. This is integrated with a standard 2D SLAM pipeline, as explained in Section IV-B.

IV. EXPERIMENTAL EVALUATION

In this Section, we perform an in-depth evaluation of our localization system by ablating its components. We carry out our experiments using two different robots: the Unitree Go2 and the MAB Silver Badger. The key difference between these platforms is that the MAB Silver Badger features an additional degree of freedom due to its actuated spine. This results in slightly more erratic camera movements, which makes the localization even more challenging. We also tested our system with three other robotic platforms, the Unitree Go1 and A1, and the MAB Honey Badger, with similar results. For sensing, we use an Intel RealSense D435i RGBD camera. Additional results can be found on our website.

In our experiments, we first show that our system can accurately estimate the state of the robot in simulation and real-world systems. While we do not strictly focus on accurate state estimation, this measure is useful for estimating the precision of map reconstruction. Indeed, a sufficiently good map is fundamental to perform autonomous navigation and exploration. Then, we present our results for autonomous navigation in known maps, showing that the system can navigate autonomously and robustly despite the low-cost sensors, without causing map corruption. We show that the system can be combined with an autonomous exploration algorithm. Finally, we report results made using two real-world settings where performance is measured using an OptiTrack system (Section IV-E). For all experiments, we use the evaluation pipeline from [36].

A. Experimental Setting

We develop our system in ROS2, and we integrate it with the ROS2 navigation stack. We provide both results obtained

in realistic simulations, using Gazebo, and in the real world with quadrupeds using RL-based control policies [2].

To compute precise metrics and rigorously evaluate our system, we conduct experiments in two large-scale simulated scenarios: the AWS small warehouse (a medium-scale furnished simulated warehouse) and the AWS small house environment (a three-room furnished simulated apartment). To demonstrate the effectiveness of our approach, we perform an ablation study comparing our full method with configurations that exclude scan stabilization or legged odometry, as well as a baseline that relies solely on visual-inertial odometry combined with 2D SLAM. For simulated environments, we do not use the contact estimation module to highlight the contribution of the other components, and we use the ground-truth contact instead. We test the full model, including contact estimation, in the more challenging experiments made with a real robot (Section IV-E). Full details can be found on our website.

Given that ground truth positions are available in simulation, we evaluate localization accuracy using the following six metrics [37]: **ATE** (Absolute Translation Error) and **ARE** (Absolute Rotational Error) measure, respectively, the Euclidean distance and the average angular distance between the estimated and ground truth positions over the entire trajectory. These two metrics are combined into the **APE** (Absolute Pose Error), which assesses the overall pose discrepancy in pose. Finally, the **RPE** (Relative Pose Error) evaluates the relative pose error over fixed intervals; we consider three values (2m), (5m), and (10m) representing different translations between poses.

B. Integrated Navigation Stack

Our navigation system is constructed entirely from ROS-compatible components and is integrated with the ROS2 navigation stack. We employ RTAB-Map [11] as the front-end for visual-inertial odometry (VIO), utilizing RGB-D imagery in conjunction with IMU data to estimate motion in real time. This front-end provides robust pose estimates in visually structured environments.

For the SLAM back-end, we integrate our pipeline with the ROS `slam_toolbox` [38], which provides efficient pose graph optimization and map management capabilities. This modular combination allows our system to benefit from
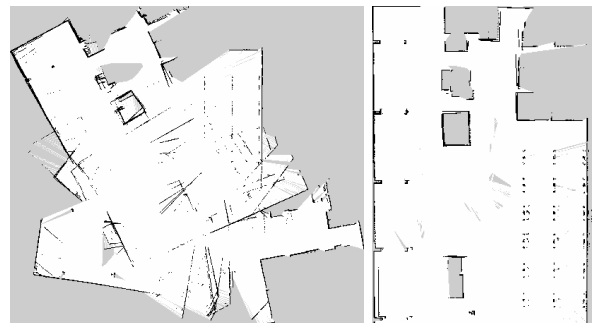

Fig. 2. Effects of scan stabilization in the map generation. Left: no scan stabilization. Right: with scan stabilization.

TABLE I

AWS SMALL HOUSE & WAREHOUSE ENVIRONMENT ABLATION STUDY FOR SB AND GO2 ROBOTS IN 5 INDEPENDENT RUNS.

Robot	Metric (RMSE)	AWS Small Warehouse				AWS Small House			
		Ours	**B+SS**	**B+LO**	**B**	**Ours**	**B+SS**	**B+LO**	**B**
SB	ATE	**0.34 ± 0.09**	5.19 ± 2.24	0.44 ± 0.11	5.00 ± 2.38	**0.42 ± 0.11**	2.54 ± 2.48	0.54 ± 0.22	1.38 ± 0.85
	ARE	**0.06 ± 0.02**	0.75 ± 0.21	0.08 ± 0.03	0.93 ± 0.37	**0.08 ± 0.05**	0.52 ± 0.49	0.10 ± 0.04	0.35 ± 0.18
	APE	**0.35 ± 0.02**	0.45 ± 0.11	7.90 ± 0.83	5.08 ± 2.41	**0.43 ± 0.12**	0.55 ± 0.22	2.59 ± 2.53	1.42 ± 0.86
	RPE (2m)	**0.18 ± 0.01**	0.18 ± 0.04	0.56 ± 0.04	0.51 ± 0.04	**0.18 ± 0.02**	0.21 ± 0.03	0.37 ± 0.04	0.27 ± 0.02
	RPE (5m)	**0.26 ± 0.02**	0.28 ± 0.11	1.15 ± 0.08	1.14 ± 0.16	**0.30 ± 0.08**	0.40 ± 0.11	0.70 ± 0.08	0.55 ± 0.04
	RPE (10m)	**0.31 ± 0.01**	0.37 ± 0.19	2.09 ± 0.16	2.20 ± 0.46	**0.45 ± 0.14**	0.64 ± 0.23	1.08 ± 0.28	0.94 ± 0.10
Go2	ATE	**0.44 ± 0.07**	0.85 ± 0.57	0.56 ± 0.20	0.53 ± 0.13	**0.43 ± 0.22**	0.43 ± 0.25	0.97 ± 0.63	0.93 ± 1.06
	ARE	**0.08 ± 0.02**	0.12 ± 0.05	0.10 ± 0.03	0.08 ± 0.04	**0.11 ± 0.06**	0.12 ± 0.06	0.21 ± 0.18	0.24 ± 0.29
	APE	**0.45 ± 0.07**	0.57 ± 0.19	0.86 ± 0.57	0.54 ± 0.13	**0.45 ± 0.23**	0.99 ± 0.65	0.44 ± 0.25	0.96 ± 1.10
	RPE (2m)	**0.19 ± 0.03**	0.23 ± 0.03	0.22 ± 0.06	0.23 ± 0.05	**0.19 ± 0.04**	0.23 ± 0.03	0.20 ± 0.05	0.23 ± 0.09
	RPE (5m)	**0.27 ± 0.03**	0.32 ± 0.05	0.38 ± 0.16	0.32 ± 0.07	**0.25 ± 0.07**	0.45 ± 0.18	0.26 ± 0.08	0.36 ± 0.16
	RPE (10m)	**0.37 ± 0.07**	0.41 ± 0.11	0.54 ± 0.24	0.42 ± 0.10	**0.36 ± 0.12**	0.80 ± 0.38	0.40 ± 0.16	0.58 ± 0.38

Acronyms: SB = Silver Badger, Go2 = Unitree Go2, B+LO = Baseline + Leg Odometry, B+SS = Baseline + Scan Stabilization, B = Baseline.

the real-time performance of RTAB-Map and the global consistency afforded by the graph-based SLAM framework.

For autonomous navigation on pre-mapped environments, we utilize the Adaptive Monte Carlo Localization (AMCL) in tandem with RTAB-Map's visual odometry to ensure accurate and drift-resilient localization. This hybrid localization strategy enhances robustness, particularly in visually repetitive indoor settings, where AMCL alone might struggle.

C. Mapping Results

Table I presents the results of our ablation study, with metrics averaged over five runs per environment and robot. The baseline configuration (B) relies solely on visual-inertial odometry and laser-based mapping without additional enhancements. The system variants include B (Baseline), B+LO (Baseline with Leg Odometry), and B+SS (Baseline with Scan Stabilization). Each subsequent variant isolates the contribution of a specific component to assess its impact. The results clearly demonstrate that the full system outperforms the alternatives and maintains consistency across all the possible metrics. Notably, scan stabilization proves to be the most critical component–a conclusion further supported by the qualitative results in Fig. 2. This component is essential for generating accurate maps, enabling effective loop closure, and maintaining localization even during rapid movements that may challenge the VIO pipeline.
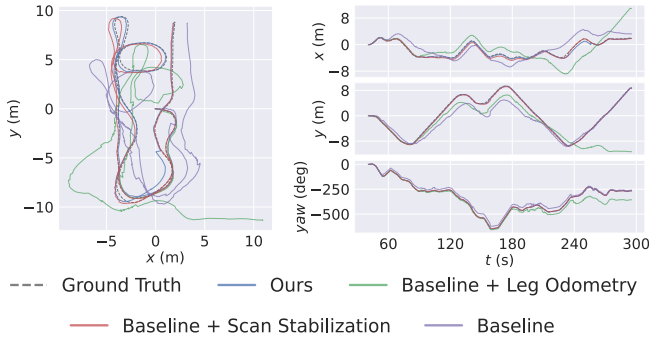
Our experiments further reveal that incorporating leg odometry into the pipeline enhances localization accuracy, yielding slightly better maps. However, when leg odometry is used without laser stabilization, performance may occasionally degrade relative to the baseline. We attribute this to inaccuracies in scan alignment within the 2D SLAM system, which result in erroneous pose refinements. When combined with additional constraints from the leg odometry system, these inaccuracies further deteriorate overall performance.

Fig. 3 illustrates that the full system accurately estimates trajectories that closely match the ground truth, underscoring the crucial role of scan stabilization in producing high-quality maps and enabling effective loop closure. More detailed results, including per-run localization performance, are available on our project website.

D. Robot autonomy

After assessing our system's map reconstruction capabilities, we proceeded to evaluate the robot's autonomy and the robustness of its navigation stack. First, we examined the navigation performance within a pre-established map across two environments using both the Silver Badger and Go2 platforms and using the previously described navigation setup (Section IV-B). The maps were generated before testing by teleoperating the robot through each environment, using our mapping system described in Section IV-C. In this experiment, the robot is sequentially commanded to navigate to five predefined poses, and we measure its success rate in reaching each target. The results, presented in Table II, clearly indicate that our system outperforms the baseline—achieving a higher

Fig. 3. Estimated trajectories of the Silver badger robot in the small warehouse environment

TABLE II

NAVIGATION SUCCESS RATE RESULTS ON AWS HOUSE AND AWS WAREHOUSE FOR SB AND GO2 ROBOTS.

Method	AWS House		AWS Warehouse	
	SB	**Go2**	**SB**	**Go2**
Baseline	0%	80%	60%	60%
Ours	**100%**	**100%**	**100%**	**100%**

Acronyms: SB = Silver Badger, Go2 = Unitree Go2

Robot	Metric (RMSE)	Environment 1 - IAS Lab				Environment 2 - IRIM Lab			
		Ours	B+SS	B+LO	B	Ours	B+SS	B+LO	B
SB (real)	ATE	**0.22 ± 0.36**	1.91 ± 0.33	2.34 ± 0.43	2.51 ± 0.50	**0.65 ± 0.28**	1.68 ± 1.27	1.74 ± 0.77	1.50 ± 0.89
	ARE	**0.33 ± 0.21**	1.38 ± 0.33	1.65 ± 0.23	2.07 ± 0.17	**0.48 ± 0.50**	0.54 ± 0.39	0.78 ± 0.48	0.60 ± 0.38
	APE	**0.64 ± 0.40**	2.38 ± 0.28	2.83 ± 0.40	3.27 ± 0.37	**0.84 ± 0.52**	1.74 ± 1.23	1.99 ± 0.61	1.65 ± 0.90

Acronyms: SB = Silver Badger, B+LO = Baseline + Leg Odometry, B+SS = Baseline + Scan Stabilization, B = Baseline.

success rate in reaching the commanded pose without losing track of its position w.r.t. the map. Scan stabilization plays a crucial role because the localization uses scans to localize itself. Even though the baseline method can navigate to a few poses successfully, the robot struggles to do so, and the navigation system has to recover several times. Videos of the results can be viewed on our project website.

Second, we evaluated the robot's capability to solve the widely-known task of autonomous exploration for map building by using a frontier-based exploration strategy [39] using a nearest-frontier method. In this scenario, we use our mapping setup in conjunction with the navigation setup (Sections IV-C & IV-B) to autonomously map the environment. The exploration results, available on our project website and illustrated in Fig 4, demonstrate that the system can successfully navigate through diverse simulated environments. It covers over 90% of the AWS house and warehouse environments without getting lost and consistently achieves robust localization and high-quality map estimation.

E. Real robot experiments

We evaluate our system's performance in two different real-world environments over five independent runs each in Table III using a SB robot. Experiments were conducted in two room-scale environments—one heavily cluttered and the other mostly clear. Details about our settings can be found on the website. Robot poses are tracked via an Optitrack motion-capture system to provide high-precision ground truth. Robots are equipped with an Intel RealSense D435i RGBD camera, while the onboard IMU supplies high-frequency acceleration and angular-velocity measurements, and the joint encoders provide leg kinematics for proprioceptive odometry. We perform our experiments using these sensors along with the same ROS-based setup as in simulation. Robots are not equipped with contact estimation sensors, and thus rely on our method as in Section III-A.

Real-world results presented in Table III confirm the insights from simulation, showing that our method reduces
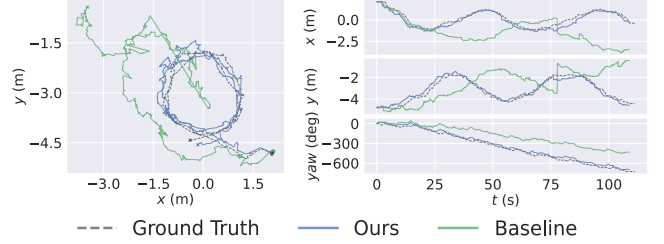

Fig. 5. Estimated trajectories using our system and the baseline of the Silver badger robot in a real-world environment (Environment 1, IAS Lab)

localization error and generates good-quality maps. The maps for both these environments using our system can be seen in Fig. 6, showing reasonable performance even in complex cluttered environments such as the IRIM lab. Fig. 5 compares our full system against the baseline for the IAS Lab. Our approach achieves consistently better overall performance and produces a trajectory that follows the ground truth closely. In general, real-world results confirm the need for the full pipeline: leg odometry alone introduces more noise, reducing the accuracy, and only scan stabilization has problems with loop closure, which is improved by the leg odometry edges in the factor graph. More details about the real-world experiments are available on the project website.

V. CONCLUSIONS

This paper shows how to develop a robust navigation system for low-cost quadruped robots, empowering them with the capability of autonomously navigating and exploring the environment. These platforms are particularly challenging to use due to the limited number of inexpensive sensors available and reactive Reinforcement Learning policies that may cause abrupt movements. Our system proves that it is possible to make these robots autonomous by using mostly off-the-shelf localization and mapping pipelines, by only fixing some key details. Our analysis shows that scan stabilization is the key issue preventing the application of
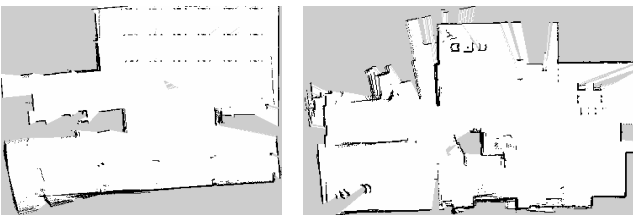

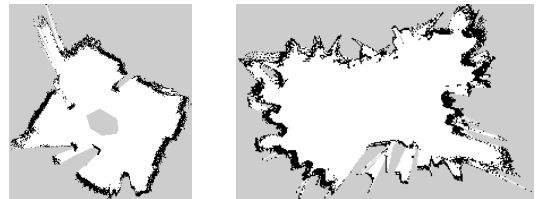Fig. 4. Autonomous exploration mapping results.

Fig. 6. Maps generated on two real-world environments. Left: IAS lab. Right: IRIM lab.

standard 2D localization pipelines. Furthermore, we can improve the mapping and the stability of VIO using the information coming from legged odometry. Both simulated and real-world results prove the effectiveness of our system in terms of map accuracy and autonomy.

In future works, we want to extend this system to more challenging 2.5D/3D mapping, possibly allowing quadruped robots to navigate and explore buildings with multiple floors.

ACKNOWLEDGMENT

REFERENCES

[1] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[2] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 91–100.

[3] G. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid locomotion via reinforcement learning," in *Robotics: Science and Systems*, 2022.

[4] G. B. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," in *Proc. CORL*, 2023.

[5] L. Smith, I. Kostrikov, and S. Levine, "Demonstrating a walk in the park: Learning to walk in 20 minutes with model-free reinforcement learning," *Proc. RSS*, vol. 2, no. 3, p. 4, 2023.

[6] L. Smith, Y. Cao, and S. Levine, "Grow your limits: Continuous improvement with real-world rl for robotic locomotion," in *Proc. ICRA*, 2024, pp. 10 829–10 836.

[7] D. Hoeller, L. Wellhausen, F. Farshidian, and M. Hutter, "Learning a state representation and navigation in cluttered and dynamic environments," *IEEE RA-L*, vol. 6, no. 3, pp. 5081–5088, 2021.

[8] C. Tang, B. Abbatematteo, J. Hu, R. Chandra, R. Martín-Martín, and P. Stone, "Deep reinforcement learning for robotics: A survey of real-world successes," in *Proc. AAAI*, 2025.

[9] J. A. Placed, J. Strader, H. Carrillo, N. Atanasov *et al.*, "A survey on active simultaneous localization and mapping: State of the art and new frontiers," *IEEE T-RO*, 2023.

[10] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Motion planning and control for mobile robot navigation using machine learning: a survey," *Autonomous Robots*, vol. 46, no. 5, pp. 569–597, 2022.

[11] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of field robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[12] C. Li, Y. Zhang, Z. Yu, X. Liu, and Q. Shi, "A robust visual slam system for small-scale quadruped robots in dynamic environments," in *Proc. IROS*. IEEE, 2024, pp. 321–326.

[13] Z. Zhou, C. Zhang, C. Li, Y. Zhang, Y. Shi, and W. Zhang, "A tightly-coupled lidar-imu slam method for quadruped robots," *Measurement and Control*, vol. 57, no. 7, pp. 1004–1013, 2024.

[14] J. Lee, M. Bjelonic, A. Reske, L. Wellhausen, T. Miki, and M. Hutter, "Learning robust autonomous navigation and locomotion for wheeled-legged robots," *Science Robotics*, vol. 9, no. 89, p. eadi9641, 2024.

[15] J. Y. Raheema, M. R. H. R. C. Provost, M. Bilinski, and H. I. Christensen, "Autonomous exploration and mapping payload integrated on a quadruped robot," in *Proc. ISRR*, 2022.

[16] J. Li, D. Zeng, Q. Luo, X. Luo, C. P. Chen, and C. Yang, "Feature assessment and enhanced vertical constraint lidar odometry and mapping on quadruped robot," *IEEE Transactions on Instrumentation and Measurement*, 2025.

[17] R. Gupta, K. Morgenstein, S. Ortega, and L. Sentis, "Fast lidar informed visual search in unseen indoor environments," in *2025 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2025, pp. 1170–1176.

[18] A. Koval, S. Karlsson, and G. Nikolakopoulos, "Experimental evaluation of autonomous map-based spot navigation in confined environments," *Biomimetic Intelligence and Robotics*, vol. 2, no. 1, p. 100035, 2022.

[19] H. Kumar, J. J. Payne, M. Travers, A. M. Johnson, and H. Choset, "Periodic slam: Using cyclic constraints to improve the performance of visual-inertial slam on legged robots," in *Proc. ICRA*, 2022.

[20] J. Cheng, M. Zhang, L. Ma, H. Chen, Y. Gan, and D. Huang, "A hybrid-dimensional laser slam framework for indoor quadruped inspection robots," *IEEE Sensors Journal*, 2024.

[21] A. Bouman, M. F. Ginting, N. Alatur, M. Palieri, D. D. Fan, T. Touma, T. Pailevanian, S.-K. Kim, K. Otsu, J. Burdick *et al.*, "Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion," in *Proc. IROS*, 2020.

[22] V. J. Gan, D. Hu, Y. Wang, and R. Zhai, "Automated indoor 3d scene reconstruction with decoupled mapping using quadruped robot and lidar sensor," *Computer-Aided Civil and Infrastructure Engineering*, 2025.

[23] I. D. Miller, F. Cladera, A. Cowley, S. S. Shivakumar, E. S. Lee, L. Jarin-Lipschitz, A. Bhat, N. Rodrigues, A. Zhou, A. Cohen *et al.*, "Mine tunnel exploration using multiple quadrupedal robots," *IEEE RA-L*, 2020.

[24] S. Fahmi, V. Barasuol, D. Esteban, O. Villarreal, and C. Semini, "Vital: Vision-based terrain-aware locomotion for legged robots," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 885–904, 2022.

[25] C. Zhang, N. Rudin, D. Hoeller, and M. Hutter, "Learning agile locomotion on risky terrains," in *Proc. IROS*. IEEE, 2024.

[26] C. Zhang, J. Jin, J. Frey, N. Rudin, M. Mattamala, C. Cadena, and M. Hutter, "Resilient legged local navigation: Learning to traverse with compromised perception end-to-end," in *Proc. ICRA*. IEEE, 2024, pp. 34–41.

[27] J. Wasserman, A. Agarwal, R. Jangir, G. Chowdhary *et al.*, "Legolas: Deep leg-inertial odometry," in *Proc. CORL*, 2024.

[28] J.-H. Kim, S. Hong, G. Ji, S. Jeon, J. Hwangbo, J.-H. Oh, and H.-W. Park, "Legged robot state estimation with dynamic contact event information," *IEEE RA-L*, 2021.

[29] Y. Kim, B. Yu, E. M. Lee, J.-h. Kim, H.-w. Park, and H. Myung, "Step: State estimator for legged robots using a preintegrated foot velocity factor," *IEEE RA-L*, 2022.

[30] D. Wisth, M. Camurri, and M. Fallon, "Vilens: Visual, inertial, lidar, and leg odometry for all-terrain legged robots," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 309–326, 2022.

[31] S. Yang, Z. Zhang, Z. Fu, and Z. Manchester, "Cerberus: Low-drift visual-inertial-leg odometry for agile locomotion," in *Proc. ICRA*, 2023.

[32] R. Boney, J. Sainio, M. Kaivola, A. Solin, and J. Kannala, "Realant: An open-source low-cost quadruped for education and research in real-world reinforcement learning," *arXiv preprint arXiv:2011.03085*, 2020.

[33] G. Bledt, P. M. Wensing, S. Ingersoll, and S. Kim, "Contact model fusion for event-based locomotion in unstructured terrains," in *Proc.ICRA*, 2018.

[34] V. Andrieu, D. Astolfi, and P. Bernard, "Observer design via interconnections of second-order mixed sliding-mode/linear differentiators," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 9, pp. 3631–3657, 2021.

[35] M. Camurri and M. Mattamala, "Leg odometry for slam," in *SLAM Handbook — From Localization and Mapping to Spatial Intelligence*, L. Carlone, A. Kim, F. Dellaert, T. Barfoot, and D. Cremers, Eds. Cambridge University Press, 2025.

[36] M. Grupp, "evo: Python package for the evaluation of odometry and slam." https://github.com/MichaelGrupp/evo, 2017.

[37] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. IROS*, 2012.

[38] S. Macenski and I. Jambrecic, "Slam toolbox: Slam for the dynamic world," *Journal of Open Source Software*, 2021.

[39] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. CIRA*, 1997, pp. 146–151.