Localization and Navigation Assessment of a Heavy-Duty Field Robot

Ahmad Kamal Nasir[1], André G. Araújo[1], and Micael S. Couceiro[1]

***Abstract*— Autonomous robots for challenging domains, for instances transportation, agriculture, forestry and construction, face several technical challenges inherent to their ability, or lack of it, in operating under the unstructured, harsh and dynamic nature of outdoor environments. This paper presents preliminary results of system integration toward autonomous navigation of a heavy-duty ground mobile robot designed for such challenging outdoor domains. The main contribution of this paper is to incorporate existing state-of-the-art Robot Operating System (ROS) based algorithms for localization, mapping, traversability, navigation, and exploration in real world unknown and unstructured environments. This paper focuses on assessing the localization and navigation ability of the robot by using proposed methodology and evaluating it under real-world outdoor tests.**

I. INTRODUCTION

Some particularly challenging domains, for example, transportation, agriculture, forestry and construction, have been left nearly inhabited by robots [1], [2]. Yet, the deployment of robots under these domains, here identified as heavy-duty applications (HDA), seems inevitable. There has been significant progress in the development and deployment of unmanned outdoor robotics systems for those applications, for instance autonomous vehicles, however, the required level of automation still requires efficient solutions to various technical challenges to be surmounted. Such as all terrain locomotion capability, long-term and large-scale localization under GPS denied environment, perception challenges due to dynamic environment conditions, scene understanding, autonomous navigation capability under complex terrain morphology, and multiple vehicles cooperation for SLAM and navigation. GPS has brought the localization problem tractable, however, under partial sky visibility, due to clouds or forest cover, the vehicle position estimate deteriorates. Fusion of various vision, range, and inertial sensors provides satisfactory results, however, their large-scale and long-term estimates require further improvements for the demanding field applications.

Fully autonomous driving capability is not only required for intelligent vehicles on the road but also for machines working in unstructured, dynamic, and harsh environments, for instance in forestry. During the last few decades, damages due to wildfires have increased dramatically all over the world and its impact can be seen in the economic stability of many nations, and ecology and society in general [3]. The need to address this issue even in the face of increasing rural abandonment and inadequate manpower brings forth the opportunity of introducing robots to assist in both fire prevention and firefighting applications. The SEMFIRE project aims to solve these problems by developing autonomous navigation capability for heavy-duty outdoor mobile robots [4].

Despite many advances in key areas, the adoption of fully autonomous driving robots for field applications is still in an early stage. This stems from the navigational challenges imposed by the unstructured setting presented by the woodland environment, but also from limited perception capabilities [5], and reasoning and planning under a high-level of uncertainty [6]. Artificial perception for robots operating in outdoor natural environments has been studied for several decades. For robots operating in forest scenarios, in particular, there is research dating from the late 80s-early 90s – see, for example, [7]. Nevertheless, despite many years of research, as described in surveys over time [8], a substantial amount of problems has yet to be robustly solved.

Our solution for autonomous navigation in precision forestry applications consists of a heterogeneous robotic team for cooperative localization, mapping, and navigation. The general framework composed of two types of robots: the *Ranger*, a 4000 kg autonomous robot, based on the Bobcat T190, equipped with a mechanical mulcher for forest

Fig. 1. SEMFIRE use case overview.

This work was supported by the Safety, Exploration and Maintenance of Forests with Ecological Robotics (SEMFIRE, ref. CENTRO-01-0247-FEDER-03269), the Semi-Autonomous Robotic System for Forest Cleaning and Fire Prevention (SAFEFOREST, ref. CENTRO-01-0247-FEDER-045931) and the Railway Autonomous Inspection and Landscape Operations with RoBOTs (RAIL-O-BOT, ESMERA H2020-ICT-2017-1).

[1] The authors are affiliated with Ingeniarius Ltd., Rua Coronel Veiga Simao, Edificio B CTCV, 3025-307, Coimbra, Portugal {ahmad, andre, micael}@ingeniarius.pt

clearing; a swarm of *Scouts*, small UAVs equipped with additional perceptual abilities to assist the ranger in its efforts, as shown in Fig. 1. The *Ranger* will function as a marsupial robot, as it can carry the swarm of *Scouts* via a small trailer, while recharging their batteries.

The presented work focuses on the Ranger localization and navigation capability without the assistance of scouts. Implementation work is currently ongoing on the SEMFIRE perception pipeline for the *Ranger*, namely the supporting ROS-based framework, sensor drivers, semantic segmentation modules, and the registration. In the following section, we shall describe some related works, proposed methodology, real-world test results toward developing autonomous navigation capability for the *Ranger*.

II. RELATED WORK

Localization and navigation in open and dynamic environments inherent to HDA is challenging due to the intermittent availability of GPS, uneven topography and difficult terrain traversability. Localization is the key for mobile robotics. However, due to poor availability of GPS and related technologies, autonomous robots have to rely on other localization sources, for example wheel, inertial, visual or range odometry. More importantly, autonomous vehicles have to feature the ability to merge data coming from these and other sources to robustly estimate their pose despite dynamic environment conditions.

The research work [9] proposed a graph SLAM method based on generalized iterative closest point matching technique for mining applications. The proposed solution takes in to account the roadway planes as optimization constraints. Another recent real-time LiDAR six Degree of Freedom (DoF) odometry approach is presented in [10] for ground vehicles. The proposed approach also leverages the presence of a ground-plane in segmentation and optimization steps, however, the proposed framework can be extended for unstructured terrain by removing the ground plane constraints.

Google Cartographer is a state of the art real-time SLAM solution, which can fuse information from multiple sensors and provide a highly accurate robot pose estimation [11] without environment feature constraints. This approach has been widely deployed in several applications, including agriculture robotics [12]. Despite the promising results achieved by Google Cartographer, as it relies on a graph-based SLAM, it fails to properly generate the map whenever adjacent submaps do not match as expected, thus inevitably leading to localization errors.

Real-Time Appearance Based Mapping (RTAB-Map) is an open-source Light Detection And Ranging (LiDAR) and visual SLAM library for large-scale and long-term online operation [13]. RTAB-Map can be adopted to provide visual odometry from the depth images to feed other approaches, including Google Cartographer. While RTAB-Map integrates a memory efficient loop closure detection approach [14], it suffers from the inability to properly handle rotations under several situations [15].

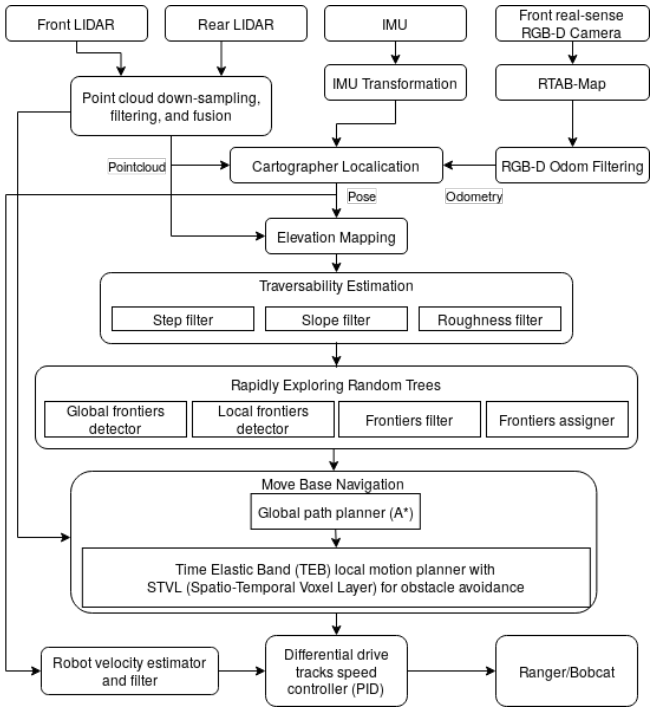

Fig. 2. ROS system architecture overview.

In the research work [16] the author presented a navigational approach for a heavy-duty wheeled mobile robot for multiple agricultural applications. However, their navigational approach is limited to a planer environment and the mapping is used only for visualization purposes.

The research work [17]–[19] presented an approach for a quadruped robot autonomous navigation in complex outdoor terrain. The proposed framework [17] is used to create elevation maps, estimating traversability, and planning safe and efficient paths in rough and unstructured terrain. The proposed framework is centered around a single robot system. However, the suggested approach is suitable for navigating the ground vehicle in uneven topography.

Another important aspect for the autonomous navigation of the ground robot in complex terrain is traversability. Traversability is not only the property of the terrain but also depends on the footprint of the platform. Wermelinger et al. [17] proposed an approach to calculate terrain characteristics, for instance slope, roughness and gaps to build a traversability map. The traversability of the environment can continuously vary between traversable, hardly traversable and non-traversable locations. The proposed approach uses an elevation map to assign each grid cell in the traversability map a value between [0-1] considering neighboring cells. One downside of the approach is that it fails to take in account the traversability of the robot between upward and downward slopes.

III. METHODOLOGY

Our autonomous navigation strategy consists of, first robust localization and then mapping of the unstructured outdoor environment followed by estimating the traversability of

the terrain and then exploration. The inputs to the processing pipeline are front and rear 3D LiDAR point clouds, robot attitude and inertial measurements from an Inertial Measurement Unit (IMU), and RGB-D camera images from a front facing depth camera. These sensor measurements are fed to various components of the autonomous navigation system. Fig. 2 shows the overview of the proposed methodology.

The first stage of the processing pipeline is a point clouds fusion node. This node simplifies the process of inserting multiple point clouds to the elevation mapping and traversability estimation process. It also reduces the computational load by removing points in the overlapping region of the two point clouds. The front and rear point clouds are transformed from their respective coordinate frames into the robot coordinate frame. Furthermore, a voxel filter is used to down-sample the point cloud to 0.2 cm/voxel and removes points above and below 3 m with respect to the robot coordinate frame's z-axis. The voxel filter size and height threshold are selected considering the computational load. The resolution of the elevation, traversability, and navigation cost maps are also equal to the voxel size.

The localization system relies on one RGB-D sensor, one IMU and two LiDARs. For the robust localization of the robot, cartographer [11] is adopted. The cartographer is selected because it can perform range odometry using point clouds and it can fuse the odometry from multiple sources. Fused point clouds are used internally to perform the range odometry using scan matching technique. The robot visual odometry is performed by the RTAB-map package [13], which uses Intel Real-sense RGB-D depth camera images and a feature-matching approach. The RGB-D sensor outputs a stream of depth images which are fed to RTABMap. RTABMap performs the visual odometry and provides an estimation of the robot pose. However, the output pose needs to be filtered in order to remove invalid robot orientations. The fusion of the range and visual odometry are complementary to each other due to different sensor modalities. The cartographer provides the 3D robot pose by using fused point clouds, inertial measurements and visual odometry. The IMU is placed inside the sensing kit mounted on the top of the robot. The cartographer uses the IMU frame for tracking purposes, therefore, the IMU measurements are transformed from the IMU frame to the robot coordinate frame to provide odometry with respect to the robot frame. The cartographer uses IMU measurements to determine the gravity vector to estimate the correct robot pose using Structure from Motion (SfM) algorithm [20].

In order to determine the traversability of the robot on uneven 3D terrain for navigation purposes, elevation mapping [21] has been adopted. The fused 3D pose provided by the cartographer is fed to the elevation map and a velocity estimator node. An elevation map can be considered as a 2.5D map. Each grid cell is assumed to contain a height field which represents the height of the obstacle at that grid cell. One downside of using an elevation map compared to the 3D voxel map is that it approximates all the vertical space above the grid cell using one height values, therefore, if there is a bridge like structure present at some location, the empty space within the object can not be correctly represented. For the Scouts, a 3D voxel map is more appropriate for path planning, however, for the ground robot elevation map is sufficient to determine robot traversability. The ROS elevation mapping package is adopted to create a $40m \times 40m$ elevation map. The size of the map is selected considering overall computational requirements. The elevation mapping node outputs two elevation maps, a raw and a filtered elevation map.

The traversability estimation process requires an elevation map, therefore, the filtered elevation map is used to estimate the traversable spaces [18] for the robot in complex 3D terrain as shown in Fig. 3. It also takes in account the robot 2D footprint information while estimating traversable regions. The estimated traversability cost-map is a 2D cost-map which is calculated by a weighted average of three different 2D cost-maps, namely, slope, step and roughness cost-map. The slope map is used to determine the slope at each grid cell within a circular vicinity of the robot. The roughness map is determined by calculating entropy within a circular region around the robot. The step map uses the maximum traversable gap width information to determine the step cost-map.

The traversable cost-map is used to identify frontiers for the robot exploration. For exploration, [22] proposed Rapidly Exploring Random Trees (RRT-Exploration). The proposed framework is designed to be modular, therefore, it can be used by a Multi-Robot System (MRS). It consists of four main components, a global frontier detector, local frontiers detectors in-case of MRS, a filter for pruning old, redundant and invalid frontiers, and the assigner node to send goals for robot navigation. The user specifies a four-point polygon region for exploration on the global traversable cost-map which is passed to global and local frontiers detectors. In the case of a single robot, only a global frontier node can be used. In case of a MRS, each robot uses its local frontiers detector and publishes the frontiers position information on a frontier topic. The frontiers filter process clusters and remove the duplicate and old frontiers received from global and local detectors. The assigner process is used to assign the optimal frontier as a goal to the available robot. The assignment strategy is modified to select the frontier with maximum information gain taking into account the robot position and orientation.

$$\begin{aligned} J(x_f) = &\lambda_1 h(x_f, x_r) I(x_f) - \lambda_2 ||x_f - x_r|| - \\ &\lambda_3 atan2(x_{r,y} - x_{f,y}, x_{r,x} - x_{f,x}) \end{aligned} \tag{1}$$

Where x_f is the frontier position, x_r is the robot position, $\lambda_{1,2,3}$ are the weights, $I(x_f)$ is the information gain, and $h(x_f, x_r)$ is the hysteresis gain.

For the robot close-loop velocity control, a velocity estimator node is implemented to calculate robot linear and angular velocity using the cartographer estimated 3D pose. Before sending the estimated velocities to the low-level PID velocity controller, they are filtered using a low-pass filter to remove the noise due to numerical differentiation. The

output of the velocity controller are sent to the Ranger CAN bus for the actual track motion. The ROS navigation [23] framework is configured to us A* [24] as a global planner on the traversability cost-map and Time Elastic Band (TEB) [25], [26] as a motion planner on a local cost-map. For the obstacle avoidance purpose, Spatio-Temporal Voxel layer (STVL) [27] is used to update the local cost-map. There are two recovery behaviors in the navigation architecture. The first behavior is to clear the local costmap with the same data as the global costmap and the second behavior is to perform a 360 degrees in-place rotation. If both behaviors are exhausted, the robot stays still and waits for a new goal location.

In order to evaluate the proposed methodology on the real-robot, the processing pipeline is first tested on a Unity based simulator designed for a 3D forest like environment, as shown in the Fig. 3. To assess the performance of the overall system following Key Performance Indicators (KPI) are defined.

- Localization failure rate (loss of position estimate).
- Navigation failure rate (inability to move).
- Ratio between robot operation time and human operation time.
- Number of times the robot fell back to human control in difficult situations.

The localization error is defined as follows:

$$\varepsilon_{loc} = |\sqrt{P_{loc} - P_{gps}}| \tag{2}$$

Where P_{loc} is the position of the robot reported by the localization system and P_{gps} is the position of the robot reported by the RTK-GPS. The localization failure is defined as follows:

$$\varepsilon_{loc,fail} = \begin{cases} 1 & \text{if } |\varepsilon_{loc,t} - \varepsilon_{loc,t-1}| > 1m \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The localization failure rate is determined by comparing the estimated robot position provided by the cartographer and the robot position provided by the GPS during the experiment.

For the assessment of navigation failure rate, navigation failure is defined as follows:

$$\varepsilon_{nav,fail} = \begin{cases} 1 & \text{if } |d_{goal} - d_{travel}| < 1m \ and \ \text{t} \geq 60\text{sec} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

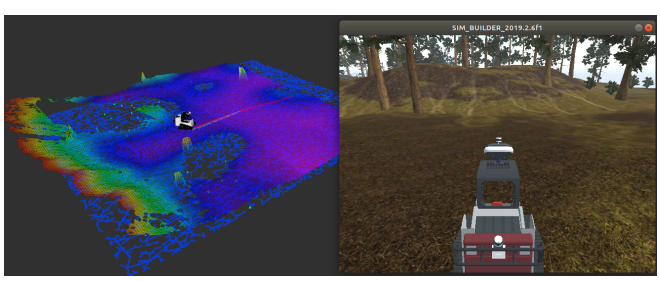

Fig. 3. Left: RRT-exploration using traversability cost-map, Right: Unity simulator.

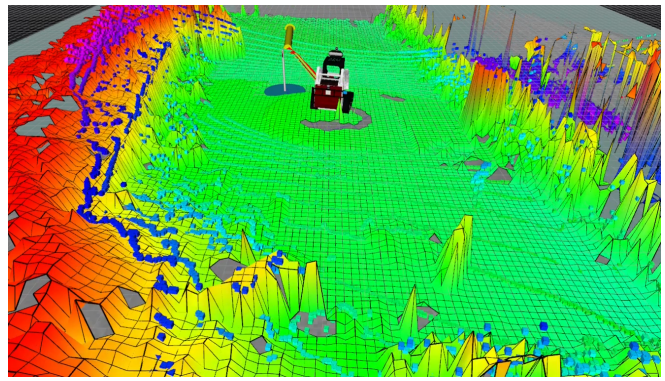

Fig. 4. Traversability estimation.

where $d_{goal} = |\sqrt{P_{loc} - P_{goal}}|$ and $d_{travel} = |\sqrt{P_{loc} - P_{prev}}|$. The navigation failure rate is assessed by considering that the robot was unable to move out of a circular region of one meter radius during a time window of 60 seconds.

IV. REAL-WORLD TEST

Two experiments were conducted in the outdoor parking area near the railway tracks near Ingeniarius, Ltd. headquarter on a cloudy day in the afternoon as shown in Fig.5. The objective of the real-world tests were to assess the localization and navigation failure rate of the complete robotic system. The experimentation area was approximately 40 m by 10 m in size. The ROS bags of the experiment were recorded and available in the public domain for downloading [28]. Due to the limited area and large detection range of the LiDARs the exploration package was only tested in the simulation as shown in Fig. 3. The blue lines in the Fig. 3 shows the randomly explored nodes, while the green points represent the filtered frontiers to be explored by the robot. The three cost-maps used to estimate traversability are assigned equal weights. The resolutions for the global and local cost-map are set to 0.2 m/grid cell, similar to the elevation map.

All the *Ranger* system software components executed on ROS Melodic, Ubuntu 18.04 (Bionic Beaver), and is

Fig. 5. Ranger operation during outdoor long-term localization and navigation experiment.

supported on the *Ranger* by the following computational resources: (1) a Mini-ITX computer equipped with a Geforce RTX 2060, an Intel Core i7-8700 CPU and 16 GB of DDR4 RAM; (2) a Tulipp FPGA+ARM Platform including a Xilinx XCZU4EV TE0820 FPGA and an ARM Quad-core Cortex-A53 CPU with a Mali-400 GPU. The *Ranger* is equipped with a sensor kit that includes Intel Real-sense 435i cameras (RGB-D+NIR sensors), a FLIR AX8 thermal camera, a Teledyne Dalsa Genie Nano C2420 multi-spectral camera, an UM7 IMU, an Emlid Reach RS2 RTK-GPS, and two LeiShen C16 Laser Range Finders. Fig. 5 shows the Ranger with front LiDAR and the sensing-kit on top of the robot. The low-level PID based velocity controllers for the ranger are implemented on an Arduino Mega2560.

To test the localization system accuracy, a RTK-GPS system was set-up before the experiments as shown in Fig. 5. The RTK base station was set up and configured to acquire measurements for 30 minutes in order to provide GPS fix status. The localization experiment was an hour long during which the robot was tele-operated to visit Geo-referenced way-points. During the second experiment the robot navigated autonomously for half hour to user assigned goals through ROS Rviz. The robot was given a new goal location whenever it reached near the previously sent goal. Thirty-six goals were sent in total to the robot during the navigation experiment. To assess the autonomy of the robot, an operator with the remote control was present all the time present in case of emergency or navigation failure situation. The goals sent along with the estimated trajectory is shown on the cartographer generated 2D map, Fig. 6, for the reference purpose.

V. RESULTS

Fig. 7 shows the result of the localization experiment. The ground truth robot trajectory is captured using the RTK GPS. During the experiment the RTK-GPS signals appear to be unreliable and switching between float and fix status, therefore, only the acquisitions with the fix status are used for further analysis. Due to this reason, for future experiments, Ultra Wide Band (UWB) positioning systems shall be employed for localization error assessment. The top graph in Fig. 7 shows both GPS and cartographer trajectories overlaid on each other. The GPS coordinates are converted

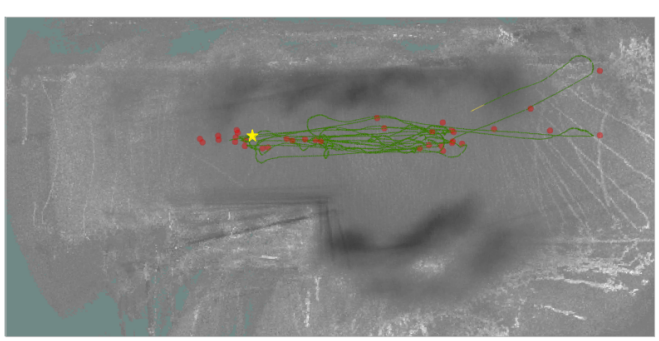

Fig. 6. Robot trajectory along with the goals locations on the occupancy grid map

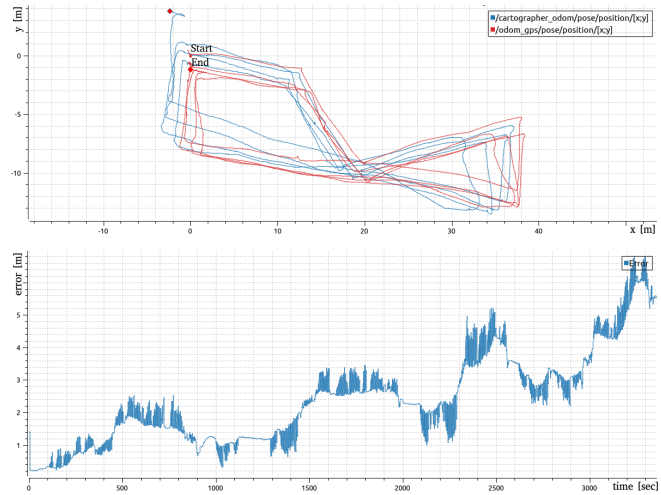

Fig. 7. Localization accuracy assessment. Top figure shows GPS and Localization system trajectories. Bottom figure shows the localization error over time.

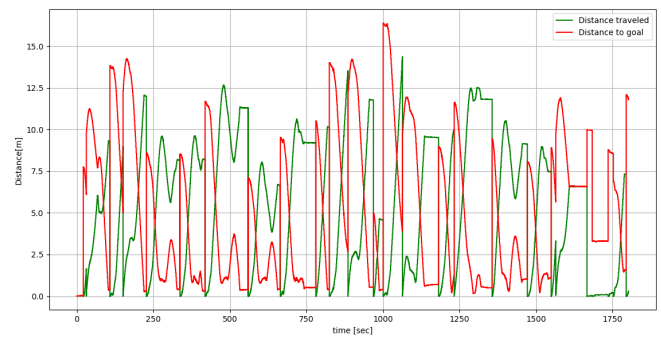

Fig. 8. Navigation failure rate.

from geodetic to Cartesian coordinate with respect to the first GPS measurement. The localization trajectory is translated and rotated manually in order to provide a visual match. The bottom graph of Fig. 7 shows the Euclidean distance between the corresponding GPS and localization system reported position. The localization reported by the cartographer appears to drifting over time. The sudden increases in the distance corresponds to the localization error.

Fig. 8 shows the result from the navigation experiment. The red line shows the Euclidean distance of the goal position from the current robot position, while the green line shows the distance traveled by the robot from the robot pose when a correct goal pose is received. For example, during the start of the experiment, the first goal was sent at $t = 51sec$ which is 7.5 m away and, therefore, the distance traveled started increasing as the robot moved towards the goal. A new goal was sent 25 sec after the previous goal, being this one 6 m away from the robot, thus resetting the distance covered to zero. It can be observed that at $t = 1650sec$ the robot appears to be stuck for about 100 seconds. It is noteworthy that the robot was able to move to a new goal location without the help of the remote operator. During the 30 minutes navigation experiment, the remote

operator intervention was not required. However, considering the navigation failure rate KPI defined in Eq. 4, it did fail once.

VI. CONCLUSIONS

This paper presented integration of various ROS based robotic components for localization, mapping, traversability, navigation and exploration of a tracked mobile robot for forestry applications. The integrated system was tested in both simulation and real-world experimentation and the results are presented. Furthermore, both the localization and navigational failure KPIs are defined and evaluated. Regarding localization and navigation ground truth, the RTK-GPS accuracy is dependent on weather conditions among other physical properties of the environment. Therefore, for future experiments, we shall also employ the UWB setup for ground truth estimation in addition to RTK-GPS. In future the proposed processing pipeline shall be extended toward a MRS.

ACKNOWLEDGMENT

The authors would like to thank Miguel Girão and the rest of the technical team, for their tireless work on the Ranger hardware, testing and experiments.

REFERENCES

[1] R. Parker, K. Bayne, and P. W. Clinton, "Robotics in forestry," *NZ Journal of Forestry*, vol. 60, no. 4, p. 9, 2016.

[2] M. Bergerman, J. Billingsley, J. Reid, and E. van Henten, "Robotics in agriculture and forestry," in *Springer handbook of robotics*. Springer, 2016, pp. 1463–1492.

[3] S. Gómez-González, F. Ojeda, and P. M. Fernandes, "Portugal and Chile: Longing for sustainable forestry while rising from the ashes," *Environmental Science and Policy*, vol. 81, pp. 104–107, 2018.

[4] M. S. Couceiro, D. Portugal, J. F. Ferreira, and R. P. Rocha, "Semfire: Towards a new generation of forestry maintenance multi-robot systems," in *2019 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2019, pp. 270–276.

[5] M. K. Habib and Y. Baudoin, "Robot-Assisted Risky Intervention, Search, Rescue and Environmental Surveillance," *International Journal of Advanced Robotic Systems*, vol. 7, no. 1, 2010.

[6] S. Panzieri, F. Pascucci, and G. Ulivi, "An outdoor navigation system using GPS and inertial platform," *IEEE/ASME transactions on Mechatronics*, vol. 7, no. 2, pp. 134–142, 2002.

[7] F. A. Gougeon, P. H. Kourtz, and M. Strome, "Preliminary research on robotic vision in a regenerating forest environment," in *Proc. Int. Symp. Intelligent Robotics Systems*, vol. 94, 1994, pp. 11–15.

[8] S. Lowry and M. J. Milford, "Supervised and unsupervised linear learning techniques for visual place recognition in changing environments," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 600–613, 2016.

[9] Z. Ren, L. Wang, and L. Bi, "Robust gicp-based 3d lidar slam for underground mining environment," *Sensors*, vol. 19, no. 13, p. 2915, 2019.

[10] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4758–4765.

[11] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1271–1278.

[12] F. Roure, G. Moreno, M. Soler, D. Faconti, D. Serrano, P. Astolfi, G. Bardaro, A. Gabrielli, L. Bascetta, and M. Matteucci, "Grape: ground robot for vineyard monitoring and protection," in *Iberian Robotics Conference*. Springer, 2017, pp. 249–260.

[13] 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.

[14] I. Caminal, J. R. Casas, and S. Royo, "Slam-based 3d outdoor reconstructions from lidar data," in *2018 International Conference on 3D Immersion (IC3D)*. IEEE, 2018, pp. 1–8.

[15] N. Altuntaş, E. Uslu, F. Çakmak, M. F. Amasyalı, and S. Yavuz, "Comparison of 3-dimensional slam systems: Rtab-map vs. kintinuous," in *2017 International Conference on Computer Science and Engineering (UBMK)*. IEEE, 2017, pp. 99–103.

[16] P. Biber, U. Weiss, M. Dorna, and A. Albert, "Navigation system of the autonomous agricultural robot bonirob," in *Workshop on Agricultural Robotics: Enabling Safe, Efficient, and Affordable Robots for Food Production (Collocated with IROS 2012), Vilamoura, Portugal*, 2012.

[17] M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, and M. Hutter, "Navigation planning for legged robots in challenging terrain," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1184–1189.

[18] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3019–3026, 2018.

[19] D. Wooden, M. Malchano, K. Blankespoor, A. Howardy, A. A. Rizzi, and M. Raibert, "Autonomous navigation for bigdog," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 4736–4741.

[20] J. J. Koenderink and A. J. Van Doorn, "Affine structure from motion," *JOSA A*, vol. 8, no. 2, pp. 377–385, 1991.

[21] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart, "Robot-centric elevation mapping with uncertainty estimates," in *Mobile Service Robotics*. World Scientific, 2014, pp. 433–440.

[22] H. Umari and S. Mukhopadhyay, "Autonomous robotic exploration based on multiple rapidly-exploring randomized trees," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1396–1402.

[23] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," in *2010 IEEE international conference on robotics and automation*. IEEE, 2010, pp. 300–307.

[24] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[25] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram, "Trajectory modification considering dynamic constraints of autonomous robots," in *ROBOTIK 2012; 7th German Conference on Robotics*. VDE, 2012, pp. 1–6.

[26] ——, "Efficient trajectory optimization using a sparse model," in *2013 European Conference on Mobile Robots*. IEEE, 2013, pp. 138–143.

[27] S. Macenski, D. Tsai, and M. Feinberg, "Spatio-temporal voxel layer: A view on robot perception for the dynamic world," *International Journal of Advanced Robotic Systems*, vol. 17, no. 2, p. 1729881420910530, 2020.

[28] M. C. A. K. Nasir, "Wppmfr 2020," Jul 2020. [Online]. Available: https://bunker.ingeniarius.pt/owncloud/index.php/s/yMWFi3A1lnERKdt