3D Local Planning for a Forestry UGV based on Terrain Gradient and Mechanical Effort

Dora S. B. Lourenço, João F. Ferreira and David Portugal[1]

***Abstract*— Planning feasible paths in 3D environments is a challenging problem, mainly in forestry environments due to, for instance, the rough and slippery terrain or the challenges for perception, as those caused by the high amount of trees, wind, and general unstructured nature of the environment. This paper presents a work in progress to propose an innovative method for 3D local planning in outdoor environments, to facilitate autonomous navigation of a forestry Unmanned Ground Vehicle. The proposed method builds on the ROS navigation stack integrating a module that analyses the gradient of the terrain to quantify slopes on the robot's path, and taking into consideration its mechanical effort when planning paths to traverse.**

I. INTRODUCTION

Reliable outdoor autonomous navigation, particularly in forestry scenarios, is still a major challenge due to the dynamic conditions of outdoor environments such as weather, illumination and vegetation characteristics changes make it difficult to build a system that can robustly navigate all the time in all conditions [1]. The forest environment is unstructured making it difficult to perceive and to correctly localize in at all times [2]; the estimation of wheel odometry that is often a useful source for localization in most environments may not be usable in forests, because of the rough terrain conditions and slippage [3]. Forestry vehicles are also typically large heavy-duty machines, making their deployment process a hard and tedious task with safety concerns.

The main goal of this work is to propose local planning methods that allow a forestry robot to safely navigate from an initial to a target configuration while avoiding obstacles. Specifically, we:

1) Present a technique that uses a 3D pointcloud from the robot's sensors to estimate the cost of traversing each individual point in space, producing a costmap for navigation;
2) Test the feasibility and applicability of existing local planning methods in forestry environments in light of the mechanical costs of traversal;
3) Demonstrate that existing techniques do not take mechanical cost into account, and can thus be refined to produce more economical trajectories.

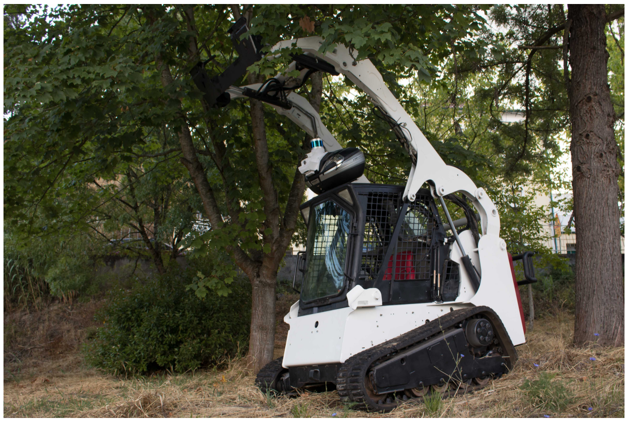


Fig. 1. The Ranger UGV.

We tested the approach in a forestry robot simulator, developed for the SEMFIRE R&D project[1], allowing us to perform repetitive testing without the related costs of using the real large-scale rig, presented in Fig. 1. This robot – the Ranger UGV – is the main actor of the SEMFIRE (Safety, Exploration, and Maintenance of Forests with the Integration of Ecological Robotics) project, which proposes the development of a robotic system to reduce fuel accumulation in forests, by eliminating flammable material for wildfire prevention, thus assisting in landscape maintenance tasks [4].

This paper is structured as follows: In Section II a literature review focused in 3D navigation is presented. In Section III is presented the approach proposed by this paper. In Section IV-A is presented experimental setup for the the tests and in IV-B the obtained results and their discussion.

II. RELATED WORK

Robot navigation is a common ability that enables the robot to reach the destination required by a certain job, planning and executing trajectories safely, both for itself, but also for humans and other entities that might be sharing its workspace [5].

Planning feasible paths in fully three-dimensional environments is a challenging problem [3]. Existing algorithms typically require the use of limited 3D representations that discard potentially useful information. In the literature, we can find several works on 3D navigation not just for ground

[1] D. Lourenço, J. F. Ferreira and D. Portugal are with the Institute of Systems and Robotics, University of Coimbra, 3030-290 Coimbra, Portugal; {dora.lourenco, jfilipe, davidbsp} at isr dot uc dot pt.

J. F. Ferreira is also with Nottingham Trent University, Computational Neuroscience and Cognitive Robotics Group, School of Science and Technology, Nottingham Trent University, Nottingham NG11 8NS, United Kingdom; joao.ferreira at ntu dot ac dot uk.

This work was supported by the Safety, Exploration and Maintenance of Forests with Ecological Robotics (SEMFIRE, ref. CENTRO-01-0247-FEDER-03269) research project co-funded by the "Agência Nacional de Inovação" within the Portugal2020 programme.

[1] http://semfire.ingeniarius.pt/, last accessed 2020/09/16.

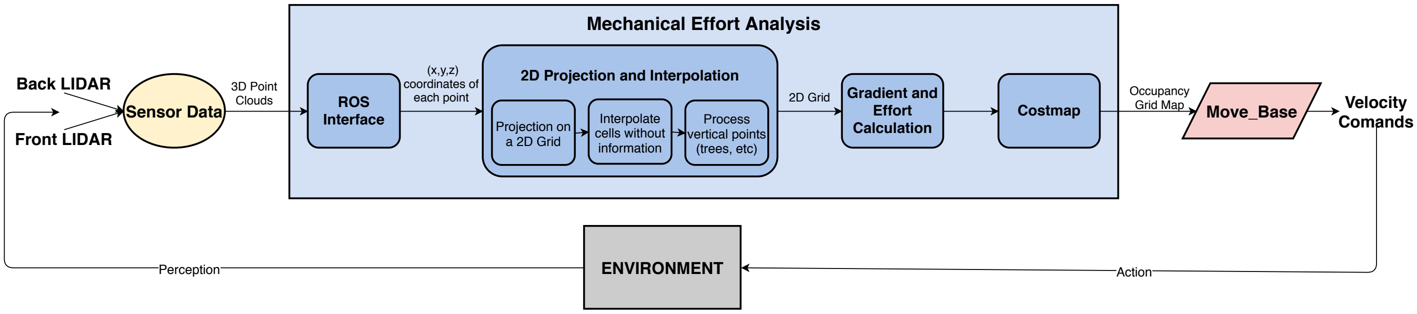

Mechanical Effort Analysis
Back LIDAR
Front LIDAR
Sensor Data
3D Point Clouds
ROS Interface
(x,y,z) coordinates of each point
2D Projection and Interpolation
Projection on a 2D Grid
Interpolate cells without information
Process vertical points (trees, etc)
2D Grid
Gradient and Effort Calculation
Costmap
Occupancy Grid Map
Move_Base
Velocity Comands
ENVIRONMENT
Perception
Action

Fig. 2. Flow diagram of the implemented algorithm to estimate the cost of traversing each individual point in space.

robots, such as [6], [7], [8], but also aerial [9], [10], [11] and underwater robots [12].

As mentioned in Section I, we aim to allow a robot to safely navigate in a 3D forestry environment. Benefiting from existing local planning methods, we propose an algorithm that distinguishes traversable areas and integrates the cost of the mechanical effort of the robot in a `2D Grid`, to be considered for local planning. In a nutshell, we consider the mechanical effort as a quantity that accounts for the additional burden of the UGV when climbing hills in rough terrain environments. In [13], a similar approach to ours is followed. The authors represent the traversability map in a `2D Grid`, but instead of each cell representing the mechanical cost of that point they represent the probability that the vehicle can successfully drive over that cell.

Other seminal research works on traversability analysis methods are available in the literature [14], [15], and in this work, we make use of the Robot Operating System (ROS) to develop our own method to estimate the cost of traversing the environment. ROS includes some ready-to-use methods, such as `navigation_mesh`[2], adopted in [16]. The authors refer to this method as a 3D representation of estimation of distances, height differences, and roughness. If specific safety thresholds are violated, these areas will be marked as lethal obstacles. Similarly to [13], they only evaluate if the area is traversable, not taking into account the mechanical effort of the robot. In [17], the authors use a “perceive-decide” paradigm on an Elevation Map of the terrain to identify traversable areas, and based on this map the best trajectory is selected. There are other packages in ROS, such as `traversability_estimation`[3], which use the elevation map and traversability estimation filters to generate a traversability map.

For local planning, some of the works mentioned earlier use existing methods like [6], in which the authors integrate their system with ROS, using `base_local_planner`[4] with the Rollout Trajectory Planner (RTP) and `move_base`[5] as the low-level controller.

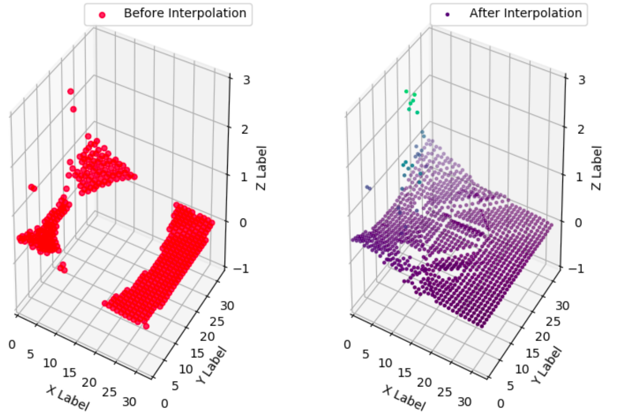


Fig. 3. 3D representation of the LIDAR information projected in a 2D grid with dimension 10 meters and resolution 0.3 meters/cell before and after the interpolation process.

III. Proposed Approach

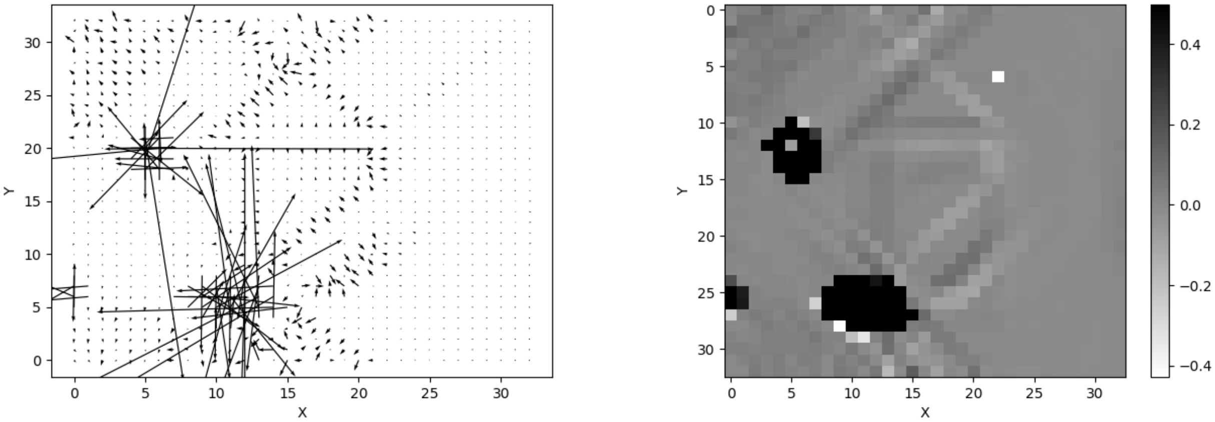
We have developed an algorithm that uses a 3D pointcloud from the robot’s sensors to estimate the cost of traversing each individual point in space, producing a costmap for navigation. This allows us to quantify the mechanical effort of traversing each cell of a map, enabling the robot to decide to traverse paths that are sub-optimal according to traditional metrics, such as total distance or time, while optimizing its use of other resources, such as energy. An overview of this process is shown in Fig. 2.

The first step is a ROS Interface, in which we subscribe to the front and back 3D LIDAR Point Clouds available in the Ranger UGV. The points of each LIDAR are transformed from their respective frame into the robot base frame and concatenated in a single matrix.

We used the ROS Navigation Stack, a framework designed to generate minimal but complete autonomous navigation solutions, thus speeding-up the implementation processes required to obtain a robot navigation solution.

This framework traditionally operates in 2D, which is a disadvantage when it is necessary to plan for a 3D environment. To account for this, we can project the 3D LIDAR information into a 2D grid, so that information is represented by:

[2] `https://github.com/uos/mesh_navigation`
[3] `https://github.com/leggedrobotics/traversability_estimation`
[4] `http://wiki.ros.org/base_local_planner`
[5] `http://wiki.ros.org/move_base`
All URLs last accessed 2020/09/16.



(a) Gradient arrows of the environment.

(b) Resulting costmap based on the gradient and effort.

Fig. 4. Graphics representing a costmap with dimension 10×10 meters and resolution 0.3 meters/cell. In (a) we can see some outliers, that are points with a very high gradient norm, and in (b) we can see the represented obstacles, such as trees in the environment.

$$\Phi(x, y) = z, \tag{1}$$

where $\Phi(x, y)$ is one cell of the grid, and z is the height of detected LIDAR obstacles in the corresponding discretized cell x, y.

The main output of the technique is a spatial grid that contains a specific cost per cell. The size and the resolution of the grid can be chosen by the user as a parameter of the approach. Due to their discrete nature, each cell may contain more than one LIDAR point. In these cases, the median is used as a representative value for the whole cell, as shown in the left picture of Fig. 3. Because this procedure results in many empty cells, it is necessary to perform interpolation to obtain an estimation of a value for those empty cells. Our approach is to apply Nearest Neighbour Interpolation by analysing the neighbour cells in row, column and diagonals with information and apply the median, as before, to estimate the cell value.

Strong discontinuities in the resulting map cause undesirable artifacts when interpolated. Thus, we have opted to filter out these points from the interpolation procedure according to:

$$|z_i - z_{average}| < \rho, \tag{2}$$

where z_i is the height estimation of each of the neighbours of the current cell, $z_{average}$ is the weighted average of all the neighbours, and ρ the threshold applied. Note that the weight of each neighbour is inversely proportional to the euclidean distance between the current cell and the neighbour cell, and the ρ threshold has been empirically adjusted to 2 m allowing for appropriate smoothing of the interpolation, resulting in the representation seen on the right of Fig. 3.

After the interpolation we calculate the gradient (Eq. 3) on each cell to obtain the inclination in each direction, Fig. 4(a):

$$\nabla z(x, y) =< \frac{\partial z}{\partial x}(x, y), \frac{\partial z}{\partial y}(x, y) > \tag{3}$$

We define effort ξ, as:

$$\xi = \cos(\theta), \tag{4}$$

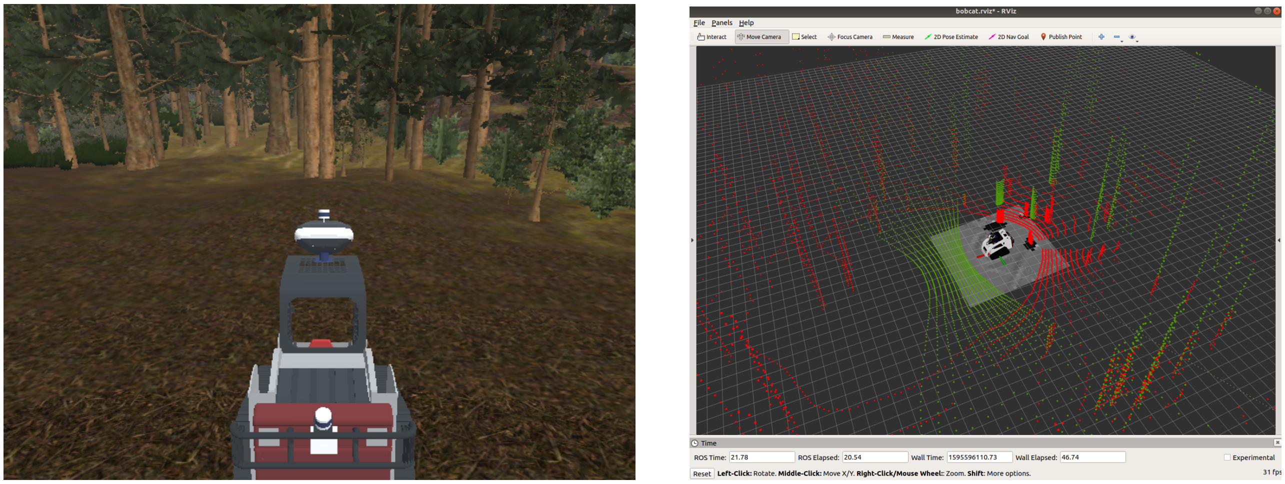
where θ is the angle between two vectors: the vector that connects the position of the robot with the position of the cell we are currently analyzing, and the gradient vector in that cell. Because the gradient direction is the direction in which the function increases more quickly, when the effort is maximum ($\xi = 1$) we are facing an upward slope, and when minimum ($\xi =-1$), a downward slope.

In order to build a costmap of the surrounding environment, all cells with gradient norm above a certain threshold are automatically considered as non-traversable; to other cells, we apply the product between the gradient norm and the so-called effort. Thus, we obtain a `2D Occupancy Grid Map` as in Fig. 4(b), where the black cells correspond to the non-traversable areas.

This approach was then integrated back into `move_base`. For local planning we are using the `base_local_planner` ROS package. This planner uses costmaps determine the optimal trajectory according to known costs between the target points, using a brute-force approach. To integrate our map with this package we made `move_base` subscribe the our new map topic, so that it could be included in the data structures that `base_local_planner` analyses to calculate the cost of traversing each cell, which it uses to score each possible trajectory.

IV. Experiments

We have carried out experiments to demonstrate that existing local planning methods in forestry environments do not take mechanical cost into account, and can thus be refined to produce more economical trajectories, in light of our definition of effort (see claims 2) and 3) in Section I).



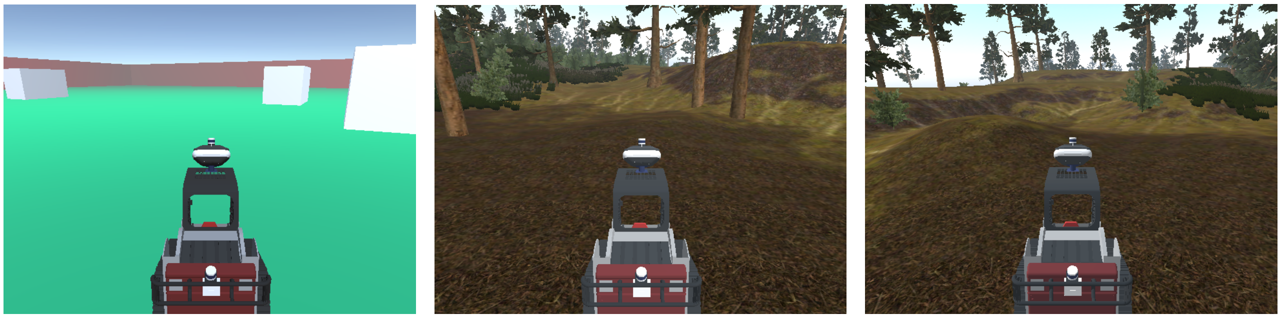
ROS Time: 21.78 ROS Elapsed: 20.54 Wall Time: 1595596110.73 Wall Elapsed: 46.74 ☐ Experimental

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options.

(a) Typical simulator scene.

(b) Simulated environment in `rviz`.

Fig. 5. Images representing the same scenario, in (a) within the Unity simulator, and in (b) the information that the sensors perceive from robot surroundings, as shown in `rviz`.



(a) First scenario.

(b) Second scenario.

(c) Third scenario.

Fig. 6. Simulator scenarios where the comparison tests of the approaches were carried out.

A. *Experimental Setup*

As mentioned in Section I, we have used a Unity-based simulator that realistically simulates a 3D forest environment. In Fig. 5(a) we show an image of a typical scene in the simulator. The simulated Ranger UGV is equipped with two 3D LIDARs, that are used as source data for our algorithm. In Fig. 5(b), we can see the information collected by the front (green dots) and back (red dots) LIDARs, as well as the local `2D Occupancy Grid Map` created with our algorithm.

We compare the default version of `base_local_planner` and `dwa_local_planner` available in ROS, measuring distance traveled, the time it took to get to the target point, the total cost (T), defined as:

$$T = \sum_{i}^{N} (\nabla z_i), \tag{5}$$

where ∇z_i corresponds to the z gradient value every 0.1s of the trajectory and T the sum of all the N z gradient values. The Upward Cost (τ), defined as

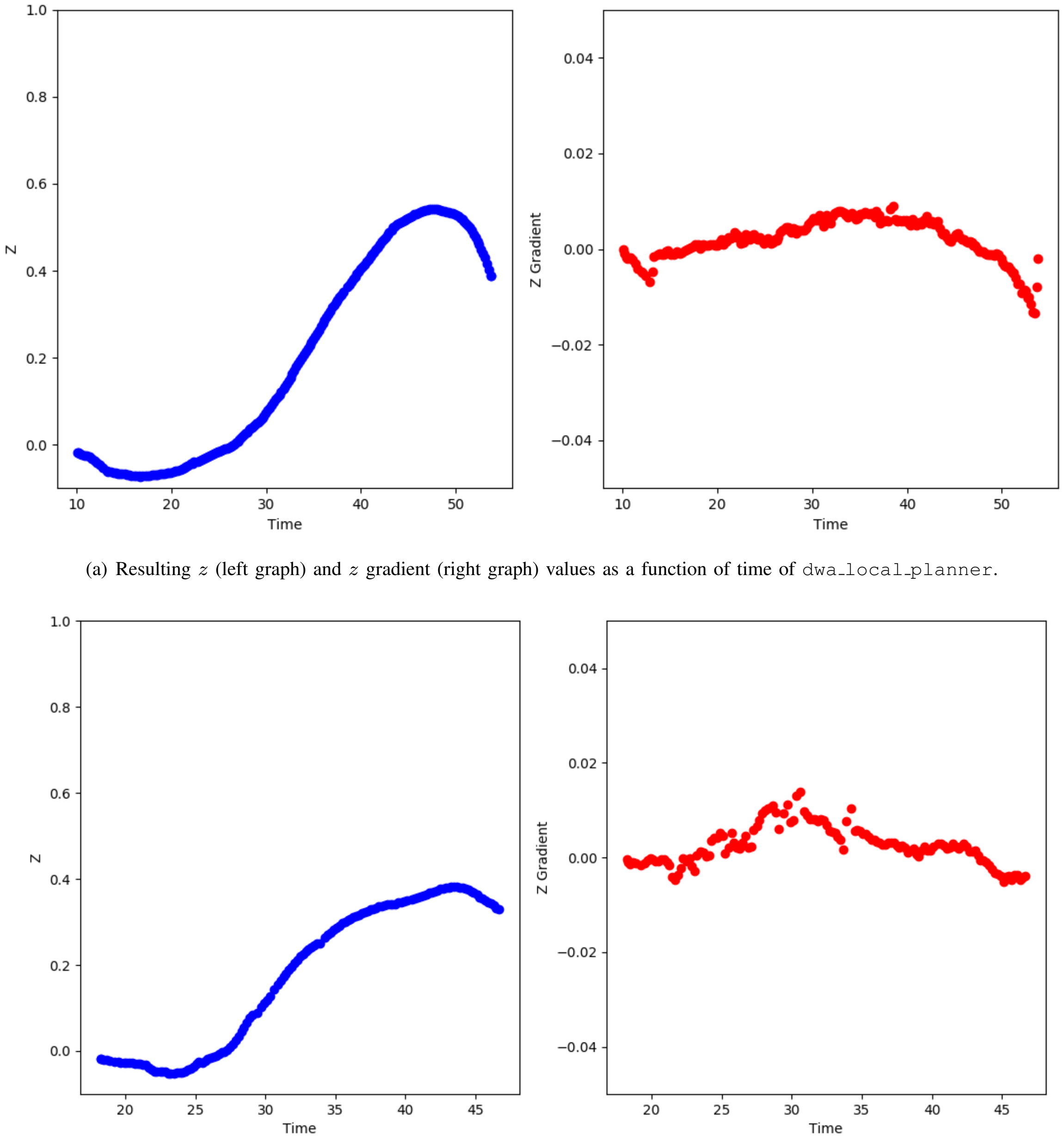
$$\tau = \begin{cases} \sum_i^n (\nabla z_i), & \text{if } \nabla z_i > 0 \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

corresponding to the sum of the n positive z gradient values, that is, the mechanical effort for the same goals given to the robot. We tested different scenarios (*cf.* Fig. 6):

1) One to validate the metrics, using a simple planar scenario with some obstacles, Fig. 6(a);
2) In the second one, we used a forestry scenario, providing a minimally flat goal, with small map elevations, Fig. 6(b);
3) In the third goal, we intend the robot to go through a more challenging forestry scenario, with large hills, to check whether the robot can overcome them, Fig. 6(c).

B. *Results and Discussion*

Given the results in the Table I, we can observe that existing techniques make efficient use of mobility resources: they plan trajectories that take the robot from A to B in a quick, efficient manner. In the first test, the robot just had to go through a planar scenario, so as expected both algorithms chose the shortest path, and the traveled distance is the same. Being a planar scenario it was expected that both algorithms obtained a total cost very close to zero, with a small variance. The `dwa_local_planner` even obtained a very small negative total cost T, meaning that



(a) Resulting z (left graph) and z gradient (right graph) values as a function of time of `dwa_local_planner`.

(b) Resulting z (left graph) and z gradient (right graph) values as a function of time of `base_local_planner`.

Fig. 7. Graphs obtained from the z values and the z gradient, with the two algorithms along the third scenario.

there was a slightly greater negative variation of the gradient than positive along the path. The `dwa_local_planner` tends to be slightly slower, taking more time to reach the goal, with only an average speed of 0.459 m/s, while the `base_local_planner` obtained an average speed of 1.180 m/s, close to the maximum speed of the robot (1.5 m/s).

However, they do not take mechanical cost into account in light of our definition of effort. As said in the last paragraph, both algorithms tend to choose the shortest path, not considering other costs. In Fig. 7, and in Tables I-III we can see that both algorithms prefer to go through a big climb, presenting a high Upward Cost (τ), instead of circumventing it. In Fig. 5(a) we can see that some hills can be circumvented, preventing the high mechanical effort.

Therefore, these can be refined to produce more economical trajectories, avoiding slopes by preventing the robot to just choose the shortest path. In order to maximize the autonomy or minimize the energetic costs, the robot should mainly avoid steep climbs to reduce substantially the mechanical effort involved in the planned trajectory path.

V. Conclusion

This paper presents a work in progress on a technique to estimate the cost of traversing each individual point in space. The method is grounded on prioritizing paths that minimize the mechanical effort to the robot. We have defined

TABLE I

RESULTS FOR THE FIRST SCENARIO

Approach	Traveled Distance (m)	Time (s)	T	τ
base_local_planner	38.680	32.779	0.00029	0.006
dwa_local_planner	38.647	79.719	-0.00026	0.001

TABLE II

RESULTS FOR THE SECOND SCENARIO

Approach	Traveled Distance (m)	Time (s)	T	τ
base_local_planner	23.366	25.559	0.2706	0.7899
dwa_local_planner	23.335	48.36	0.276	0.842

TABLE III

RESULTS FOR THE THIRD SCENARIO

Approach	Traveled Distance (m)	Time (s)	T	τ
base_local_planner	17.670	28.98	0.390	0.4646
dwa_local_planner	16.88	43.659	0.403	0.6117

metrics and tested competing techniques to determine how well they fared according to our standards. We can conclude that there is room for improvement and, thus, this research will continue.

In the short term, besides testing additional interpolation methods, and to generally improve, document and make the proposed approach available to the community, we aim to obtain quantitative results comparing the travel time and energy spent when taking into consideration the mechanical effort. Also, we would like to propose metrics to answer questions such as: "Is circumventing a hill better than driving on it considering that a larger route will imply also additional power consumption to some extent?". Future work will tackle two main fonts: (1) we will compare our approach with the standard techniques; (2) when a stable approach is proposed, tests will be transferred to the real robot – the Ranger, a 4000 kg heavy-duty UGV, based on the Bobcat T190 (see Fig. 1).

ACKNOWLEDGMENT

The authors would like to thank Gonçalo S. Martins for the technical discussion and scientific guidance and Rui P. Rocha for invaluable help with technical support for carrying out this work.

REFERENCES

[1] Y. Morales, A. Carballo, E. Takeuchi, A. Aburadani, and T. Tsubouchi, "Autonomous robot navigation in outdoor cluttered pedestrian walkways," *Journal of Field Robotics*, vol. 26, no. 8, pp. 609–635, 2009.

[2] 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, p. 10, 2010.

[3] R. Reis, F. N. dos Santos, and L. Santos, "Forest robot and datasets for biomass collection," in *Iberian Robotics conference*, pp. 152–163, Springer, 2019.

[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)*, pp. 270–276, IEEE, 2019.

[5] A. Bonarini, S. Ceriani, G. Fontana, and M. Matteucci, "On the development of a multi-modal autonomous wheelchair," in *Handbook of Research on ICTs for Human-Centered Healthcare and Social Care Services*, pp. 727–748, IGI Global, 2013.

[6] À. Santamaria-Navarro, E. H. Teniente, M. Morta, and J. Andrade-Cetto, "Terrain classification in complex three-dimensional outdoor environments," *Journal of Field Robotics*, vol. 32, no. 1, pp. 42–60, 2015.

[7] A. Linz, A. Ruckelshausen, E. Wunder, and J. Hertzberg, "Autonomous service robots for orchards and vineyards: 3D simulation environment of multi sensor-based navigation and applications," in *Proceedings book of the 12th International Conference on Precision Agriculture (ICPA)*, pp. 2327–2334, 2014.

[8] A. Pfrunder, P. V. Borges, A. R. Romero, G. Catt, and A. Elfes, "Real-time autonomous ground vehicle navigation in heterogeneous environments using a 3D lidar," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2601–2608, IEEE, 2017.

[9] E. Koyuncu and G. Inalhan, "A probabilistic b-spline motion planning algorithm for unmanned helicopters flying in dense 3D environments," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 815–821, IEEE, 2008.

[10] J. L. Sanchez-Lopez, M. Wang, M. A. Olivares-Mendez, M. Molina, and H. Voos, "A real-time 3D path planning solution for collision-free navigation of multirotor aerial robots in dynamic environments," *Journal of Intelligent & Robotic Systems*, vol. 93, no. 1-2, pp. 33–53, 2019.

[11] C.-T. Lee and C.-C. Tsai, "3D collision-free trajectory generation using elastic band technique for an autonomous helicopter," in *FIRA RoboWorld Congress*, pp. 34–41, Springer, 2011.

[12] I. Tusseyeva, S.-G. Kim, and Y.-G. Kim, "3D global dynamic window approach for navigation of autonomous underwater vehicles," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 13, no. 2, pp. 91–99, 2013.

[13] J. Sock, J. Kim, J. Min, and K. Kwak, "Probabilistic traversability map generation using 3D-lidar and camera," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5631–5637, IEEE, 2016.

[14] S. Martin and P. Corke, "Long-term exploration & tours for energy constrained robots with online proprioceptive traversability estimation," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5778–5785, IEEE, 2014.

[15] R. O. Chavez-Garcia, J. Guzzi, L. M. Gambardella, and A. Giusti, "Learning ground traversability from simulations," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1695–1702, 2018.

[16] S. Pütz, T. Wiemann, J. Sprickerhof, and J. Hertzberg, "3d navigation mesh generation for path planning in uneven terrain," *IFAC-PapersOnLine*, vol. 49, no. 15, pp. 212–217, 2016.

[17] D. Bonnafous, S. Lacroix, and T. Siméon, "Motion generation for a rover on rough terrains," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 2, pp. 784–789, IEEE, 2001.