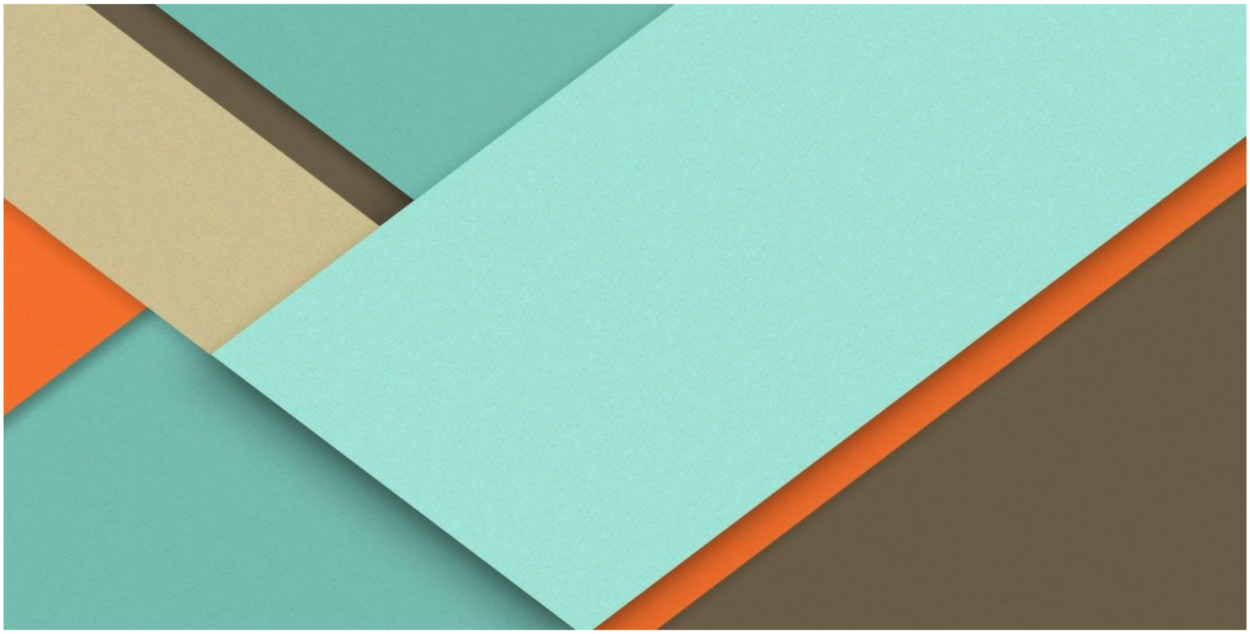

Robot Navigation Aimed for Uneven and Rough Agricultural Terrains

An Initial Report

13.02.2021

Fetullah Atas
PhD Candidate at REALTEK

Overview

This document aims to describe and give background information related to PhD research in focus to vision aided navigation of agricultural robots in uneven terrains. The document is splitted to a few sections to describe the Goals, Specifications and targeted Milestones of research.

Goals

This section is intended to clarify absolute goals of this research. The outcomes expected from this research are listed as follows;

1. A robot that performs on uneven and rough outdoors environments needs a 3D aware navigation system. Current popular navigation frameworks in Robotics society are based on 2D environment representation making the navigation of agricultural robots in non-flat environments infeasible.
2. In an uneven terrain , slopes, ramps and elevations are natural. The proposed method is expected to to understand that a slope or a ramp is not necessarily an obstacle instead depending on their size they could be traversable.
3. An outdoor robot's most reliable localization source is a RTK-GPS. The proposed method will take the RTK-GPS as its primary source of localization, opposed to 3D/2D scan matchers(AMCL,NDT) in other frameworks. Other available sensors such as IMU, wheel encoders will be included in localization. . In addition to sensors, if available, results of SLAM localization will also be fused in order to improve relative localization of robot W.R.T to the map.
4. For robust collision free motion planning, the environment that the robot will operate in, needs to be mapped with a 3D SLAM algorithm. According to current experimental results, LIDAR based 3D SLAM approaches perform poorly in repetitive patterns of agricultural environments. A vision based SLAM such as ORB-SLAM2 has yielded better results in a simulated agricultural environment. The usage of this 3D map will be optional however availability of such map will boost robustness of the system in face of collision free motion planning and will help to determine traversability of the environment. Additionally this map will be used to get relative localization of robots W.R.T this map. In case where the map cannot be created for some reason the collision checking should be relied on live sensor data such as LIDAR point clouds.

5. The architecture of the proposed method should be abstract from the hardware, the sizes/shapes of robots used for agricultural tasks can be various, the proposed method needs to cope with such cases.

Specifications and Research Questions

This section focuses on details on the sought methods to be investigated in order to achieve goals described above.

Current consideration for 3D environment representation is octomap[1]. However this alone won't be enough to express constraints needed when planning motions in 3D for a ground robot. A ground robot still has limited states that can be controlled. For instance it's unusual for a ground robot to propel in the z-axis(upwards from ground plane) with attached actuators. The generated path plans should be 2.5D, such that the robot's wheels are always in contact with the ground but when necessary the plans can elevate through ramps and slopes. This will most likely require prior knowledge of the ground surface which is embedded in the octomap. A research question here can be arised as;

- How does one express all kinodynamic constraints of a ground robot within the octomap or another alternative map representation such that when planning, these constraints can be recognized and respected by the planner.

Additionally, the slopes and ramps and elevations should be properly represented within this 3D map. This means that the collision checking while planning should be able to distinguish slopes and ramps from actual obstacles. This most likely could be addressed by adding cost layers to the original octomap. Another research question then arises here;

- What are the algorithms, methods that can properly distribute the cost of cells in an octomap ?

Consider the case given in Figure.1, a simple agricultural environment and robot is simulated under Gazebo[2]. The ramp located in front of the robot is not really an obstacle, for many robots it is possible to go through such a ramp. The planner can plan around the ramp but if the ramp was wide such that the robot had no way but to pass through, then the planner should be capable of addressing such cases, with the help of octomap, expressed constraints and added cost layer(s).

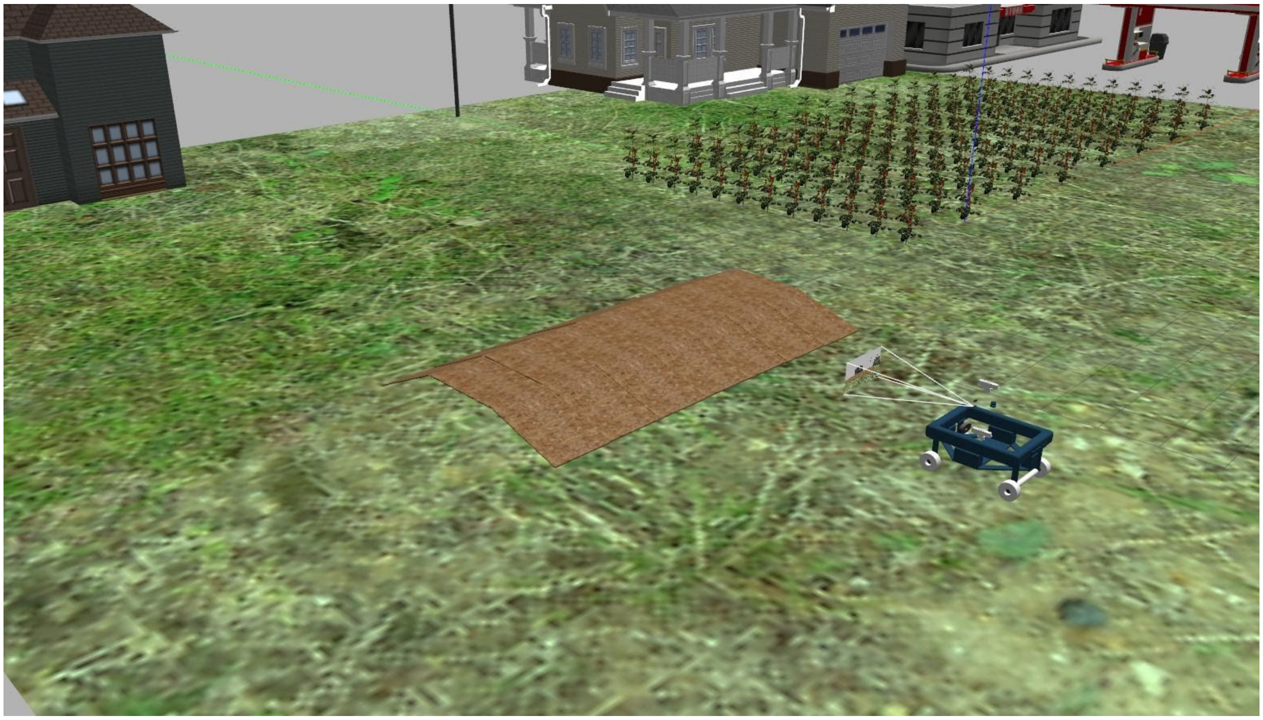

Figure.1

A simulated agricultural environment and simulated agricultural robot in Gazebo simulator.

In Figure.2, a map corresponding to the case in Figure.1 was created with OpenVSLAM[3]. The map is represented by an RGB cloud, starting from blue to magenta colors. The yellow color cloud comes from live LIDAR, so it is not part of octomap. This map is georeferenced, meaning that the origin of the map is aligned to earth coordinate frames, so the live sensor data and map data match correctly.

The RGB points are points clouds that represent octomap built for the case in Figure.1. OpenVSLAM is a visual SLAM framework which is mainly built on top of ORB-SLAM2[4] in the background. Using this octomap one can generate collision free paths. But the generated paths might not necessarily meet our requirements to deal with ramps and slopes. See Figure.3 for a path generated to a goal given in front of the robot.

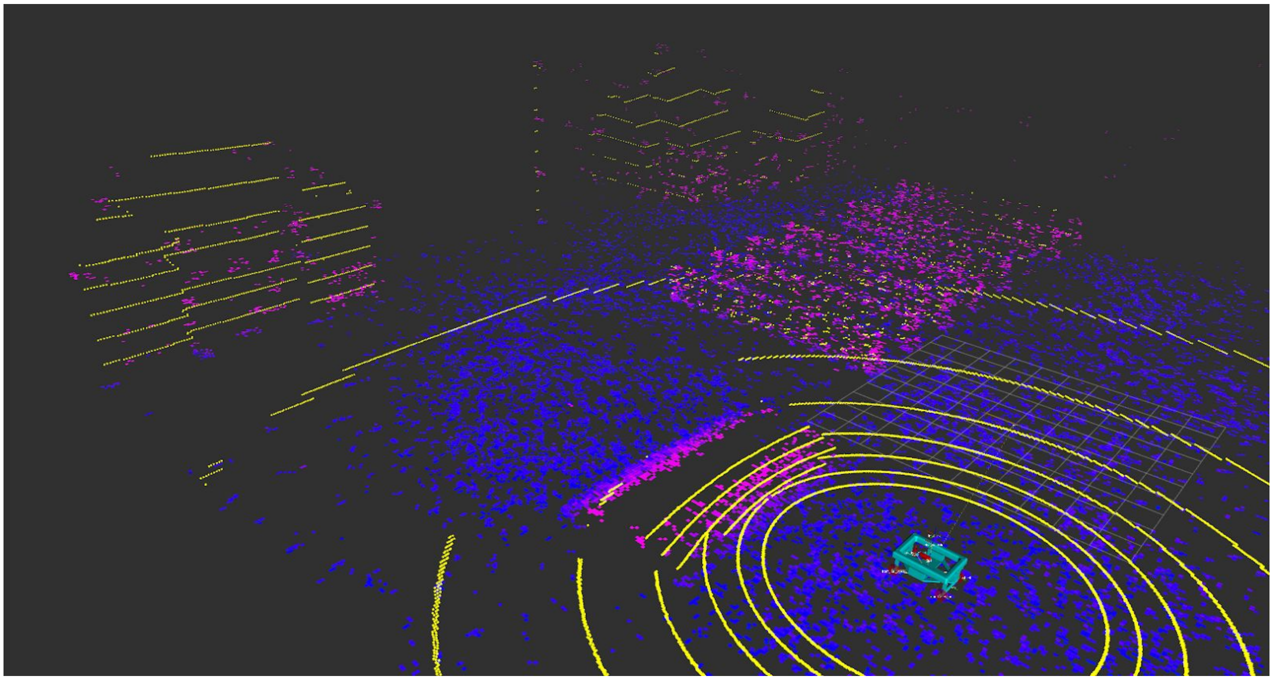

Figure.2

RGB point cloud representing octomap built by ORB-SLAM2, the yellow point clouds are from live LIDAR

In Figure.3 PRMStar probalistic planner from OMPL[5] was used to generate a plan for robot to pass to the other side of the ramp. The planner does output a valid plan but this might not be optimal or desired for the cases where we just want the robot to get into the goal with the shortest path/time. Since this plan was generated in SE2 Space(x,y,θ), the generated plan has no movement in the z axis, therefore it has to plan around the ramp. Extending SE2 Space to SE3 Space(6-DOF), seen in Figure.4 does yield a somewhat desired path but with the elevation in z axis not respecting the robot's constraints.

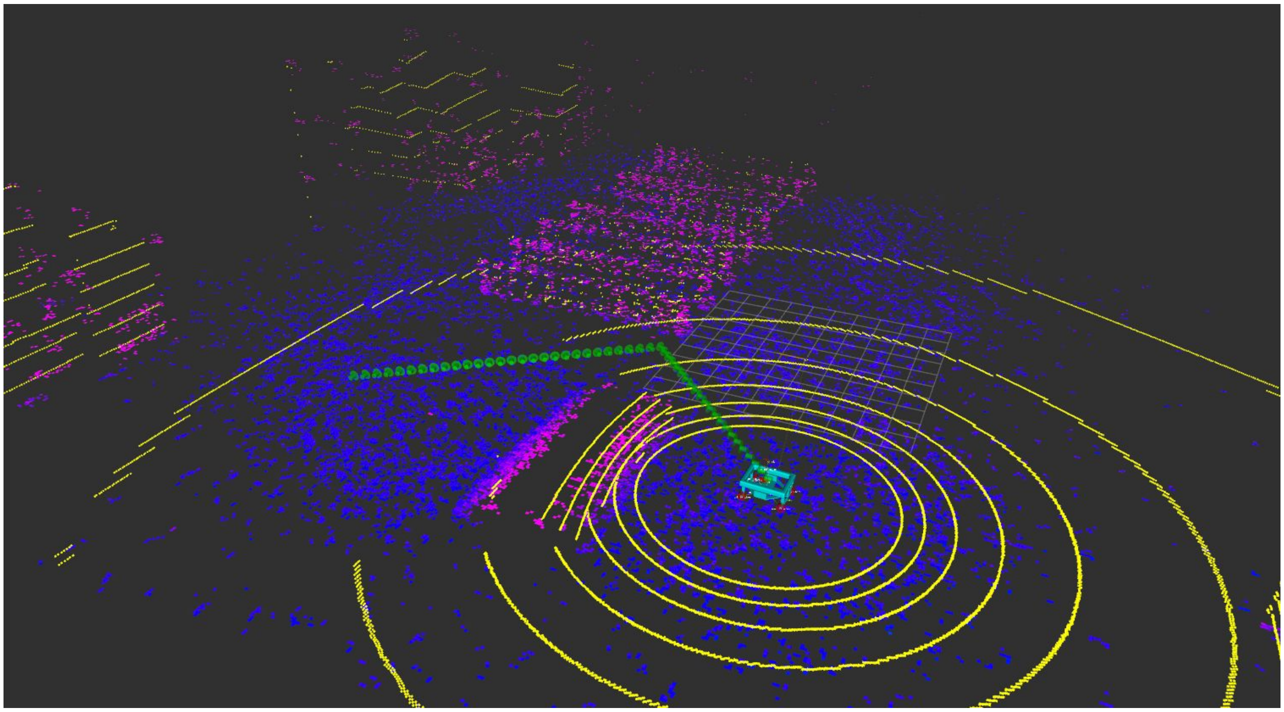

Figure.3

Planning in SE2 Space(x,y,θ) with OMPL's PRMStar. The plan is found to be around the ramp depicted in front of the robot.

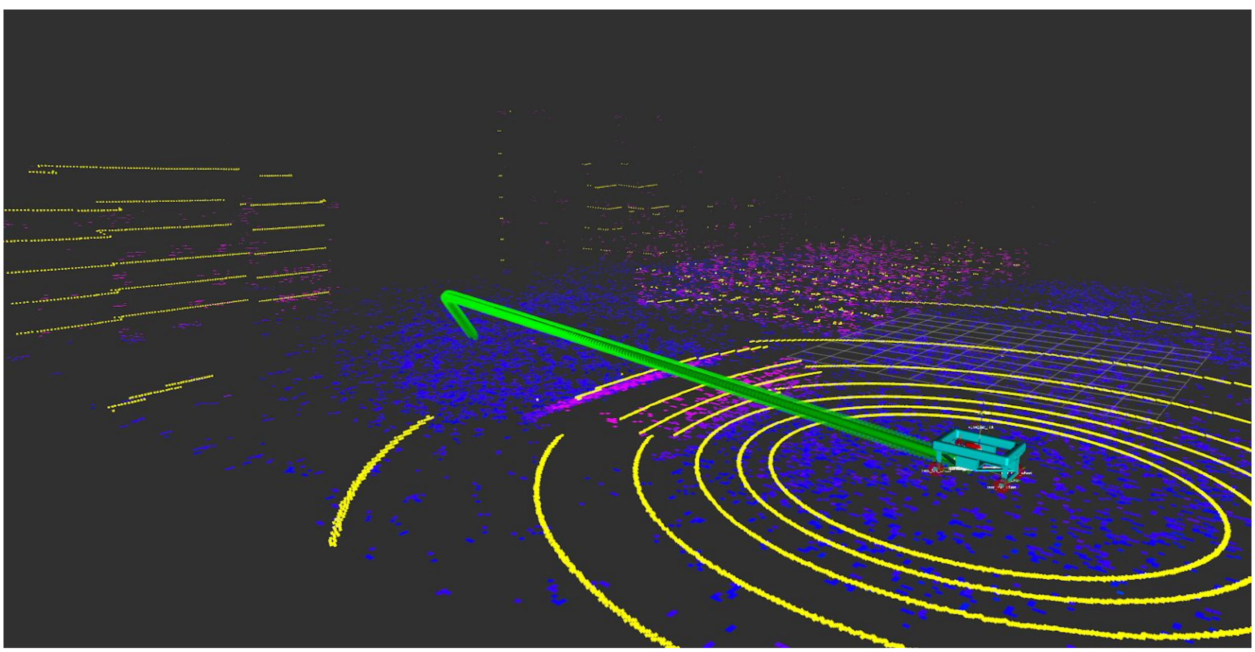

Figure.4

Planning in SE3Space with OMPL's RRTStar. The generated plan is shorter and more optimal, however the z axis of the path seems unreasonable considering kinodynamic constraints of the robot.

A combination of answers that can address the two research questions above should be able to project parts of the path generated in Figure.4 such that it is feasible in the z axis too.

A valid, optimal and constraints respecting planning is actually an important focus of this research. Once found, a valid path plan then can be fed to a trajectory tracking algorithm(AKA controller), finally navigating the robot to goal pose collision free. Existing algorithms such as TEB Controller[6], DWA controller[7] have proven to perform well for tracking pre-planned paths. These algorithms are coupled with 2D Occupancy Grids for collision checking(or state validity checking) while feeding reference states to the controller. However this research is focused on 3D, therefore a modification to these controllers might be necessary to utilize them. Another consideration for a controller is to reimplement Model Predictive Controller(MPC) from scratch. The ultimate goal is to come up with an easy to configure controller that can deal with multiple robot kinematics such as Ackermann, Differential-Drive.

Milestones

I. Design and Considerations for Octomap Cost Layer(s)

Addressing this will be crucial for the advance of this research hence it is a milestone that is aimed to be achieved in early stages of research. Cost layer will aim to determine penalties/costs to be considered while the planner generates plans. The costs corresponding to each of sampled states (by planner) can be queried. The cost layer(s) should be parameterized and configurable, this should allow the algorithms generating costs to be flexible, for instance for some applications going through a ramp could be desirable or vice versa. It is then up to desired behavior and with provided parameter interfaces one should be able to tune the system.

II. Generating Valid Plans Through Ramps, Slopes and Elevation

A complete octomap with its cost layer(s) can be utilized to get minimal cost plans from start to goal poses. This means that in some cases a minimal cost plan could be well going through all these nonlinear terrains. OMPL, a powerful motion planning library, luckily contains implementations of many state-of-the-art planning algorithms that can quickly be deployed. The path optimization objectives would be set considering costs in octomap layer(s). The returned paths then should be with minimal costs calculated by optimization objective function.

III. Constraint and Collision(in 3D) aware implementation of Optimal Controller for Path Tracking

This milestone will be the last piece of puzzle for navigating a robot from start to goal pose with collision free and optimal motion commands in 3D. Receding horizon control or MPC is a well known and studied problem of Control Theory. That is, given states, control inputs and robot dynamics constraints, find optimal control inputs that drive error between reference states and current states to minimum in a time horizon. This is most of the time a nonlinear optimization problem, several software tools including CasAdi[8] are known to provide optimal or suboptimal solutions in reasonable time for such optimization problems. At this milestone an MPC control shall be implemented for tracking of trajectories produced by previous milestone(II.)

IV. Utilize Latest Software Stack for Implementation

Implementation of the system shall be realized by up to date software tools. ROS2, an improved successor of ROS is targeted as middleware. Besides ROS2 and its utilities other methods such as Behaviour Trees shall be used. Behaviour trees will help to define complex robot behaviours in an intuitive way.

References

[1] Hornung, A., Wurm, K.M., Bennewitz, M. *et al.* OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Auton Robot* **34,** 189–206 (2013). https://doi.org/10.1007/s10514-012-9321-0

[2] Design and use paradigms for gazebo, an open-source multi-robot simulator

N Koenig, A Howard - ... on Intelligent Robots and Systems (IROS ..., 2004 - ieeexplore.ieee.org

[3] OpenVSLAM: A Versatile Visual SLAM Framework MM '19: Proceedings of the 27th ACM International Conference on MultimediaOctober 2019 Pages 2292–2295https://doi.org/10.1145/3343031.3350539

[4] ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras

Raul Mur-Artal, Juan D. Tardos

[5] Ioan A. Șucan, Mark Moll, Lydia E. Kavraki, The Open Motion Planning Library, IEEE Robotics & Automation Magazine, 19(4):72–82, December 2012.

[6] D. Fox, W. Burgard, S. Thrun: The dynamic window approach to collision avoidance. In: IEEE Robotics & Automation Magazine. 1997, pp. 23–33.

[7] C. Rösmann, F. Hoffmann, T. Bertram: Integrated online trajectory planning and optimization in distinctive topologies. In: Robotics and Autonomous Systems, vol. 88, 2017, pp. 142–153.

[8] {CasADi} -- {A} software framework for nonlinear optimization and optimal control

Joel A E Andersson and Joris Gillis and Greg Horn and James B Rawlings and Moritz Diehl journal of Mathematical Programming Computation