Interactive waypoint navigation for autonomous monitoring of vegetables in complex micro-farming

Sebastian Pütz[1,2], Malte Kleine Piening[1,2] and Lars Schilling[1,2]

Abstract: We present an interactive topological waypoint navigation system for autonomous robots integrated with the open source ROS, Move Base Flex (MBF), and Mesh Navigation stack (MeshNav). In the DFKI EXIST transfer-of-research project *PlantMap*, we develop a robotic software stack towards autonomous long-term navigation in market gardens, i.e., micro-farming. To monitor vegetable plants autonomously with the developed agricultural monitoring robot Lero in a market garden, we developed an interactive flexible waypoint navigation system.

Keywords: waypoint navigation, autonomous navigation, Move Base Flex, ROS, Mesh Navigation, agricultural monitoring robot

1 Introduction

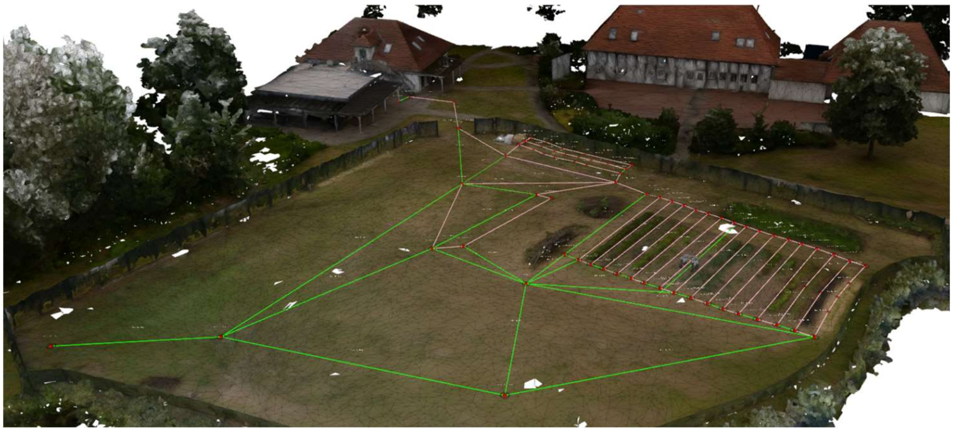

Fig. 1: Topological waypoint navigation graph and 3D AI Market Garden map from the DFKI test environment in Arenshorst, Bohmte, Germany

The path planning is performed using the underlying triangular mesh map representing the market garden environment as a surface in 3D space. In the paper, we present the whole

---

[1] Nature Robots GmbH, <first name>.<last name>@nature-robots.de
[2] German Research Center for Artificial Intelligence (DFKI), PBR, <first name>.<last name>@dfki.de

system commanding the agricultural robot Lero built within the PlantMap project to autonomously perform navigation using a mesh map which covers the market garden as shown in Figure 1. The robot is designed to record high resolution 3D data while it drives along a bed following the waypoint navigation plan. We show the novelty of this approach as it can enable long-term autonomy in a modular and well-structured way while it is fully integrated with ROS, MBF and MeshNav. Our approach is also integrated with RViz so that the developer can easily view current states, and add, remove or change waypoint positions or edges, and modify semantic meanings.

2 Related work

To get an overview of high-level execution logic architectures, Colledanchise and Ögren [Co18] list different architectures, such as subsumption architectures, teleo-reactive programs, decision trees as well as Behavior Trees (BTs), describe their advantages and disadvantages. Harel [Ha87] introduced the first hierarchical finite state machine (HFSM). The SMACH framework [B010] implements a modern HFSM library as a task level execution engine fully integrated with ROS topics, services, and actions. We use SMACH as task level architecture as described in detail below. It allows rapid prototyping of execution strategies in ROS and python. We designed such a SMACH to execute a high-level plan defined by a topological edge sequence generated by the user or a task.

In previous work, [Pü19; Pü20; Pü21; Pü22], we developed Move Base Flex (MBF) and integrated it for the use with 2D costmaps, as well as for the use with the 2.5D Grid Map representation, and finally for the use with a 3D mesh map representation. Furthermore, we demonstrated the advantages over other frameworks like move_base which is already successfully used for long-term navigation, for example, the robot office marathon [Ma10]. Conner et al. [Co17] proposed *flexible_navigation,* a more flexible framework compatible with the planner, controller and recovery plugins used for the standard navigation middle layer framework move_base. However, with MBF, we also eliminated the shortcomings of *flexible navigation*. These days, MBF is used in several projects in academia, industry and personal projects [Pü22].

In [Pü19; Pü20], we presented the integration for mesh maps and the interaction within RViz for labeling mesh segments and selecting goal poses on the mesh surface for navigation as well as autonomous navigation in the forest using MeshNav and the Continuous Vector Field Planner (CVP). This work forms the basis for the PlantMap project where we integrated the MeshNav software stack in the AI Market Garden robotic test environment in Arenshorst as described in detail below. To allow high-level applications in such domains, such as long-term monitoring of vegetable plants, the interactive waypoint navigation framework extends these software components and forms an interactive interface used towards long-term monitoring.

3 Architecture

3.1 Waypoint navigation system

The topological waypoint graph is defined on the mesh map. Each edge can hold multiple semantic attributes used during the task level execution. This way, we can assign attributes to associate edges with a bed in the garden which can later be used by the planner to observe certain beds. Semantic information such as path-blocked, trafficability or marker readings can be added to the system in order to perform replanning to further optimize the operation outcome or even to enable a robust and smart autonomous system.

For the market garden use case, we define *free-space*, *in-row* and *inter-row* edges in order to use different navigation strategies, i.e. MBF path planners, controllers and recovery behaviors to master the given navigation tasks. Later, this will be extended with additional navigation strategies, i.e., *parking*, and *docking*. The *in-row* edges are defined over beds connecting waypoints at the beginning and end of each bed, while intermediate waypoints can be used to cover beds with curvature or to trigger certain events. In addition, each two *in-row* segments lying next to each other are connected by *inter-row* edges. Lastly, different parcels of beds, a base station with the charging station, or special places can be connected by a sequence of *free-space* edges if the robot should use a path planner suitable for free space.

The user interaction is realized using interactive markers [Go11] in RViz and ROS. It allows us to change the used planner or controller, to block the edge, to define a plant type for an edge, and to change the trafficability which is used for the planner and could also be used for the controller speed. Furthermore, it also allows you to simply add intermediate waypoints, delete an edge or node, or to save the waypoint graph. Internally, the modules use the python library networkx and communicate using ROS services, e.g. to add, update, and delete edges and nodes. Additionally, the menu allows you to select a sequence of edges for the current path which can later be executed as explained next.

3.2 Task level waypoint navigation execution

In the following, we describe the architecture and connection to MBF using a hierarchical state machine implemented using SMACH, see Figure 2.

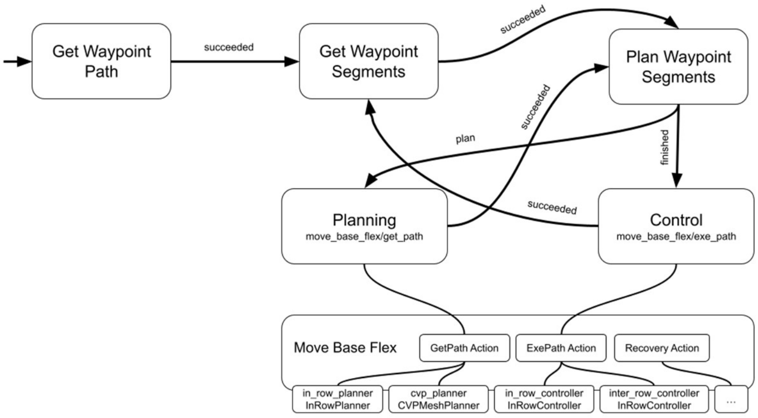

Fig. 2: Waypoint execution: performs the waypoint navigation using the MBF actions GetPath and ExePath

Figure 2 shows an implemented hierarchical state machine. The state *GetWaypointPath* waits until a topological path, i.e. a list of waypoints, is received. Next, it transitions to the State *GetWaypointSegments* which finds consecutive waypoint edges with the same controller type. Here, we combine these path segments in order to enable a continuous driving using one controller over a combined path, i.e. an array of poses which covers the sequence of topological edges. Thus, this combined list of waypoints is handed over to the *PlanWaypointSegments* state which calls the respective *in_row_planner* or *cvp_planner* in a sequence and concatenates the resulting low level plans, i.e. the lists of poses, to one plan. If the planner succeeds, the path is handed over to *PlanWaypointSegments* for the concatenation step. As soon as all paths of the waypoint segments are computed and concatenated, the concatenated path is handed over to the respective controller which computes velocity commands in order to follow the given path. Our four wheel hardware controller then translates these velocity commands to motor commands. If the robot arrives, the system transitions to *GetWaypointSegments* in order to process the next segments of the topological path.

4 Application

Our waypoint navigation system is implemented and tested in the DFKI AI Market Garden test environment in Arenshorst near Bohmte, Germany. The test environment contains several areas including 16 beds lined up, 3 heavily curved beds and free space. So far *in-row* and *inter-row* edges are being tested. The AI Market Garden mesh map is created using a Riegl VZ400i high resolution terrestrial laser scanner. Such mesh maps can also automatically be generated, see [Pü16; Pü22]. The interactive waypoints can be chosen arbitrarily as explained in Chapter 3.1. To localize the robot on the mesh map, several sensors are combined using an extended Kalman filter. We tested two localization methods, one based on RTK GPS and the other based on LiDAR, each combined with filtered IMU and wheel odometry measurements. When using GPS for localization, the mesh map must be georeferenced. We have successfully been using the navigation system in the DFKI AI Market Garden for 1.5 months now to record data of the plants twice a week. During this recording campaign, we came across no major issues with the navigation stack other than some parameters that required some tuning initially. A video of the waypoint navigation system in action can be seen here:

https://youtu.be/-HOqglSfhAc

5 Summary and future work

In this paper, we present an interactive waypoint navigation approach consisting of several software components. The *interactive waypoints* component allows the user to modify and define missions, as well as to modify the underlying graph and the strategies to execute. The *waypoint server* holds and persists the data and provides services to add, update and delete nodes and edges with its attributes. The *waypoint planner* component plans waypoint sequences using the selected edges by the user. The waypoint graph with its attributes, e.g. the used controller, plant type, and trafficability is connected to the *waypoint execution* component in order to perform navigation tasks and monitoring tasks. This modular system allows to model different use cases for autonomous robots with different kinematics navigating in complex outdoor terrains from small-scale micro-farming up to large-scale organic vegetable farms, or forest path networks with unstructured forest pathways with changing slope. We demonstrated the use in the AI Market Garden test environment in Arenshorst near Bohmte, Germany, with the developed robot Lero.

Acknowledgements: The DFKI Niedersachsen is sponsored by the Ministry of Science and Culture of Lower Saxony and the VolkswagenStiftung.

Bibliography

[Bo10] Bohren, J.; Cousins, S.: The SMACH High-Level Executive [ROS News]. IEEE Robotics & Automation Magazine, 17(4):18-20, 2010.

[Co17] Conner D. C., Willis, J.: Flexible Navigation: Finite State Machine-Based Integrated Navigation and Control for Ros Enabled Robots. In SoutheastCon 2017, p. 1-8, March 2017.

[Co18] Colledanchise M., Ögren, P.: Behavior Trees in Robotics and AI: An Introduction. CRC Press, 2018.

[Go11] Gossow, D. et al.: Interactive Markers: 3-D User Interfaces for Ros Applications [ROS Topics]. IEEE Robotics & Automation, Magazine, 18(4):14-15, December 2011.

[Ha87] Harel, D.: Statecharts: A Visual Formalism for Complex Systems. Science of Computer Programming, 8(3):231-274, June 1987.

[Ha17] Hawes, N. et al.: The STRANDS Project: Long-Term Autonomy in Everyday Environments. IEEE Robotics Automation Magazine, 24(3):146-156, September 2017.

[Hö20] Höllmann, M. et al.: Towards Context-Aware Navigation for Long-Term Autonomy in Agricultural Environments. In 12th IROS Workshop on Planning, Perception, Navigation for Intelligent Vehicle (IROS PPNIV-2020), IEEE Robotics and Automation Society, October 2020.

[Ma10] Marder-Eppstein, E. et al.: The Office Marathon: Robust Navigation in an Indoor Office Environment. In Ieee International Conference on Robotics and Automation, Icra 2010, Anchorage, Alaska, USA, 3-7 May 2010, p. 300-307. IEEE, 2010.

[Ni22] Niemeyer, M. et al.: A Spatio-Temporal-Semantic Environment Representation for Autonomous Mobile Robots equipped with various Sensor Systems, In: International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI-2022), September 20-22, Cranfield, United Kingdom, IEEE 9/2022.

[Pü16] Pütz, S. et al.: 3D Navigation Mesh Generation for Path Planning in Uneven Terrain. In Proceedings. IFAC Symposium on Intelligent Autonomous Vehicles (iav-2016), 9th, June 29-July 1, Leipzig, Germany. IFAC, 2016.

[Pü18] Pütz, S. et al.: Move Base Flex: A Highly Flexible Navigation Framework for Mobile Robots. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 2018. Software available at https://github.com/magazino/move base flex tex.ids= putz 2018 movebaseflexa ISSN: 2153-0866.

[Pü19] Pütz, S. et al.: Tools for Visualizing, Annotating and Storing Triangle Meshes in ROS and RViz. In Proceedings of the 9th European Conference on Mobile Robotics, Prague, Czech Republic, Prague, Czech Republic, September 2019. IEEE.

[Pü20] Pütz, S. et al.: The Mesh Tools Package – Introducing Annotated 3D Triangle Maps in Ros. Robotics and Autonomous Systems (RAS), Selected papers from the 9th European Conference on Mobile Robots (ECMR 2019):1-38, November 2020.

[Pü21] Pütz, S. et. al.: Continuous Shortest Path Vector Field Navigation on 3D Triangular Meshes for Mobile Robots. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 2021. IEEE.

[Pü22] Pütz, S.: "Navigation Control & Path Planning for Autonomous Mobile Robots." PhD-Thesis, Osnabrueck University, Feb. 2022.