ROSPlan: Planning in the Robot Operating System

What is it?

Specifically created for use with the Robot Operating System, ROSPlan is a framework for robotics task planning (ROS). It offers a general-purpose interface for integrating planning algorithms with different robotic systems, enabling robots to operate independently by creating plans based on environmental knowledge.

What are the main contributions?

- Integration with the ROS ecosystem: ROSPlan is built to integrate easily with the other ROS packages, enabling, among other things, systems for manipulation, perception, and navigation.
- Support for PDDL: ROSPlan is based on the PDDL (Planning Domain Definition Language) formalization, which offers a clear and straightforward method of defining the objectives, restraints, and regulations of a planning domain.
- Management of the knowledge base: ROSPlan gives the robot access to a knowledge base that it can utilize to decide on the optimal course of action. Based on inputs from sensors and other sources, this knowledge base is updated in real-time.
- Flexible planning algorithms: ROSPlan supports a variety of planning algorithms, allowing users to select the most appropriate algorithm for a given task based on its specifications.
- Robust execution: ROSPlan provides real-time execution of plans, enabling the robot to observe its environment continually and make adjustments as needed to finish its task.
- It is simpler for developers to incorporate intelligent behavior into their robotic systems thanks to ROSPlan, which offers a comprehensive solution for task planning in robotics.

What are other research on the same topics?

Here are a few of these:

- This planning system, called FAST-DOWNWARD, is based on heuristic search and is able to resolve issues specified in the PDDL language.
- Constrained programming and heuristic search are the foundations of the planning system known as PR-OHSP. It offers a versatile and effective method for resolving planning issues.
- Robot motion can be planned continuously and incrementally using the CHIMP (Continuous Histogram-Based Incremental Motion Planning) motion planning framework.
- A planning system called MOtoNMS (Motion and Task Planning for Non-Holonomic Mobile Robots) combines motion planning with task planning to solve the issue of autonomous navigation in challenging situations.
- STAN: The stochastic trajectory and navigation (STAN) planning framework offers a probabilistic approach to planning robot movements.

These are only a few of the numerous research studies that have been done on job planning in robotics. Each of these strategies has advantages and disadvantages of its own, and the ideal strategy for a particular issue depends on the demands of the work and the capabilities of the robot.

What are the similar/related applications?

- Robots that have the ability to navigate their surroundings without human assistance are said to have autonomous navigation. It entails the robot creating motion plans based on its understanding of the environment and its objectives.
- Motion planning involves figuring out the best route for a robot to take in order to arrive to its destination. Finding a practical and ideal motion for the robot while avoiding obstacles and meeting requirements is what this task entails.
- Manipulation planning is the process of deciding the order in which a robot should carry out certain operations in order to manipulate an object in its surroundings. To accomplish the task, plans must be created for the robot's joints and end effector.
- Planning a robot's work flow is the process of figuring out the best order in which a robot should carry out different tasks in order to finish a job. It entails creating strategies for the robot's movement, manipulation, and other behaviors to carry out a series of tasks effectively.
- Human-Robot Interaction: This is the capacity of a robot to communicate with people in a straightforward and natural way. It entails creating strategies for the robot's conduct based on its comprehension of the intents and deeds of the human.

These are just a handful of the numerous related or analogous uses for task planning in robotics. By enabling robots to carry out activities autonomously and intelligently, these applications hope to increase the usefulness and adaptability of robots in the real world applications.

The current limitations?

- Scalability: It can be difficult for robots to develop plans in real-time due to the computationally expensive and potentially unscalable nature of planning for big and complex environments.
- Uncertainty: Because the real world is fundamentally uncertain, robots need to be able to manage it when making plans. Currently, planning algorithms are unable to manage probabilistic and uncertain information, which results in less-than-ideal plans.
- Interference: When several robots are working together in the same space, their plans may conflict, which could result in strange behavior. Addressing this can be difficult, and the solutions that are in place today might not always work.
- Robustness: Robots need to be capable of adjusting to sudden changes in the environment or the objectives of the activity. The ability of robots to respond to these changes and produce new plans in real-time is still restricted.
- Human Interaction: Robots must be able to communicate with people in a way that feels natural and intuitive. It is now difficult to develop successful plans that take into account the behaviors of humans since robots are still limited in their understanding of human intentions and goals.

These limits draw attention to the problems that still need to be solved in the area of robotic task planning and the need for additional study and development.

The current limitations?

The goal of ROSPlan is to overcome some of the issues and constraints with current planning systems in the robotics industry. ROSPlan is a good and distinctive solution since it has a number of advantages over other planning systems.

- Integration with ROS: Because ROSPlan is connected with the ROS framework, it is simple to use it with other ROS tools and packages. Thus, it becomes a more smooth and integrated method of organizing robotics tasks.
- Support for PDDL: ROSPlan has support for the Planning Domain Definition Language (PDDL), a popular language for describing planning issues. Due to this, ROSPlan can readily interact with current planning domains and issues.
- ROSPlan has a modular design, which makes it flexible and adaptable to a variety of planning applications. It is more adaptable as a result than other planning systems because it can be quickly changed and extended for particular uses.
- User-Friendly: ROSPlan offers a user-friendly interface for expressing planning problems and creating plans, making it accessible to a larger range of users, including those without substantial planning expertise.
- Robust Performance: ROSPlan has been demonstrated to operate effectively across a range of planning domains, offering solid and dependable solutions to planning issues.

Overall, ROSPlan provides a special and useful solution for robotics job planning, offering a versatile, approachable, and strong solution that interacts naturally with the ROS framework. Because of these benefits, ROSPlan is an excellent option for planning work in robotics and a significant advancement in the industry.

How does it work?

Finding a solution to a task planning issue specified in the Planning Domain Definition Language is how ROSPlan operates (PDDL). Robot Operating System (ROS), a well-known open-source robotics platform, is the framework within which the system is intended to operate. Following are the fundamental steps in the planning process for the ROSPlan:

- The user describes the task planning problem in PDDL, which defines the current world state, the actions that can be taken, and the desired state.
- Plan Generation: ROSPlan employs a planning algorithm to create a plan based on the problem statement in PDDL that fulfills the objective state. In order to change the current state of the world into the desired one, the planning algorithm looks for a series of activities.
- Execution of the Plan: ROSPlan, which communicates with the ROS system to implement the plan's instructions, executes the created plan.
- Feedback and Monitoring: ROSPlan continuously observes the plan's execution and informs the planning algorithm when an update is necessary due to changes in the environment or the availability of resources.

Until the objective state is attained or the planning problem is determined to be intractable, this process is repeated. Since it can be quickly modified and expanded to handle a variety of planning activities, ROSPlan offers a flexible and scalable solution for task planning in robotics.

Possible Improvements:

- Real-time planning is currently supported by ROSPlan offline planning, in which the plan is created ahead of time and then performed. ROSPlan could be expanded to include online planning algorithms that can create and update plans in real-time as the status of the world changes in order to allow real-time planning.
- ROSPlan is currently made for single-agent planning, in which a single robot executes the tasks in the plan. The ROSPlan framework could be expanded to incorporate algorithms that can produce plans for a group of cooperating robots in order to facilitate multi-agent planning.
- Better Plan Generation: To produce better and more effective plans, ROSPlan might incorporate more sophisticated planning algorithms, such as those based on reinforcement learning or heuristic search techniques.
- Better ROS Framework Integration: ROSPlan might be expanded to include more sophisticated capabilities, such as support for more complicated ROS message types and better ROS topic and service management, to improve the framework's integration.
- User-Friendly Interface: ROSPlan might have a more user-friendly interface for creating and carrying out plans, such as a graphical user interface or a web-based interface, to make it more accessible to a larger range of users.

Possible Improvements:

- As a valuable tool for robotic job planning, ROSPlan offers a scalable and adaptable framework for creating plans in the Robot Operating System (ROS). The system communicates with ROS to execute the plan's instructions after using a planning algorithm to solve a task planning problem defined in the Planning Domain Definition Language (PDDL).
- Numerous robotics applications, from straightforward single-agent tasks to intricate multi-agent scenarios, have made use of ROSPlan. Even with its achievements, ROSPlan still has a lot of room for improvement. These include real-time planning, multi-agent planning, enhanced plan generation, better interaction with ROS, and a more user-friendly interface.
- Overall, ROSPlan makes a significant contribution to the field of robotics and is an invaluable tool for academics, industry professionals, and developers engaged in task planning and control research.

References

Beetz, M., and McDermott, D. 1994. Improving Robot Plans During Their Execution. In *Proc. International Conference on AI Planning Systems (AIPS)*.

Bernardini, S.; Fox, M.; and Long, D. 2014. Planning the Behaviour of Low-Cost Quadcopters for Surveillance Missions. In *Proc. Int. Conf. on Automated Planning and Scheduling (ICAPS)*.

Carrera, A.; Palomeras, N.; Ribas, D.; Kormushev, P.; and Carreras, M. 2014. An intervention-auv learns how to perform an underwater valve turning. In *OCEANS - Taipei, 2013 MTS/IEEE*.

Cashmore, M.; Fox, M.; Larkworthy, T.; Long, D.; and Magazzeni, D. 2014. AUV Mission Control Via Temporal Planning. In *Proc. Int. Conf. on Robots and Automation (ICRA)*.

Coles, A.; Coles, A.; Fox, M.; and Long, D. 2010. Forward-Chaining Partial-Order Planning. In *Proc. Int. Conf. on Automated Planning and Scheduling (ICAPS)*, 42–49.

Della Penna, G.; Magazzeni, D.; and Mercorio, F. 2012. A universal planning system for hybrid domains. *Applic. Intell.* 36(4):932–959.

Dornhege, C.; Hertle, A.; and Nebel, B. 2013. Lazy Evaluation and Subsumption Caching for Search-Based Integrated Task and Motion Planning. In *Proc. IROS workshop on AI-based robotics*.

Eyerich, P.; Mattmüller, R.; and Röger, G. 2012. Using the context-enhanced additive heuristic for temporal and numeric planning. In *Towards Service Robots for Everyday Environments*. 49–64.

Firby, R. J. 1987. An investigation into reactive planning in complex domains. In *Proc. National conference on Artificial intelligence (AAAI)*, 202–206.

Fossen, T. I. 2011. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, Ltd.

Fox, M., and Long, D. 2003. PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. *Journal of AI Research (JAIR)* 20:61–124.

Fox, M.; Long, D.; and Magazzeni, D. 2012. Plan-based policies for efficient multiple battery load management. *J. Artif. Intell. Res. (JAIR)* 44:335–382.

Frank, J., and Jonsson, A. 2003. Constraint-based attribute and interval planning. *Journal of Constraints* 8(4).

Gaschler, A.; Petrick, R. P. A.; Giuliani, M.; Rickert, M.; and Knoll, A. 2013. KVP: A knowledge of volumes approach to robot task planning. In *Proc. Int. Conf. on Intelligent Robots and Systems (IROS)*, 202–208.

Geib, C.; Mourão, K.; Petrick, R.; Pugeault, N.; Steedman, M.; Krueger, N.; and Wörgötter, F. 2006. Object Action Complexes as an Interface for Planning and Robot Control. In *Proc. Humanoids-06 Workshop: Towards Cognitive Humanoid Robots*.

Gerevini, A., and Serina, I. 2002. LPG: A planner based on local search for planning graphs. In *Proc. Int. Conf. on Automated Planning and Scheduling (AIPS)*. AAAI Press.

Ghallab, M.; Nau, D.; and Traverso, P. 2004. *Automated Planning: Theory and Practice*. Morgan Kaufmann.

Graham, R.; Py, F.; Das, J.; Lucas, D.; Maughan, T.; and Rajan, K. 2012. Exploring Space-Time Tradeoffs in Autonomous Sampling for Marine Robotics. In *Proc. Int. Symp. Experimental Robotics*.

Hitzler, P.; Krötzsch, M.; Parsia, B.; Patel-Schneider, P. F.; and Rudolph, S., eds. 2009. *OWL 2 Web Ontology Language: Primer*. W3C Recommendation. Available at *http://www.w3.org/TR/owl2-primer/* (Nov 2014).

Hoffmann, J., and Edelkamp, S. 2005. The Deterministic Part of IPC-4: An Overview. *Journal of AI Research (JAIR)* 24:519–579.

Howey, R.; Long, D.; and Fox, M. 2004. VAL: Automatic Plan Validation, Continuous Effects and Mixed Initiative Planning Using PDDL. In *Proc. Int. Conf. on Tools with AI (ICTAI)*, 294–301.

Ingrand, F.; Chatilla, R.; Alami, R.; and Robert, F. 1996. PRS: a high level supervision and control language for autonomous mobile robots. In *IEEE Int.l Conf. on Robotics and Automation*.

Kavraki, L. E.; P. Svestka, J.-C. L.; and Overmars, M. H. 1996. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. In *IEEE Trans. on Robotics and Automation*.

Kortenkamp, D., and Simmons, R. G. 2008. Robotic Systems Architectures and Programming. In *Springer Handbook of Robotics*. Springer. 187–206.

Lemai-Chenevier, S., and Ingrand, F. 2004. Interleaving Temporal Planning and Execution in Robotics Domains. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

Magazzeni, D.; Py, F.; Fox, M.; Long, D.; and Rajan, K. 2014. Policy learning for autonomous feature tracking. *Autonomous Robots* 37(1):47–69.

McGann, C.; Py, F.; Rajan, K.; Thomas, H.; Henthorn, R.; and McEwen, R. S. 2008. A deliberative architecture for AUV control. In *Proc. Int. Conf. on Robotics and Automation (ICRA)*.

Muscettola, N.; Dorais, G.; Fry, C.; Levinson, R.; and Plaunt, C. 2002. IDEA: Planning at the Core of Autonomous Reactive Agents. In *Proc. of AIPS Workshop on On-line Planning and Scheduling*.

Palomeras, N.; El-Fakdi, A.; Carreras, M.; and Ridao, P. 2012. COLA2: A Control Architecture for AUVs. *IEEE Journal of Oceanic Engineering* 37(4):695–716.

Piacentini, C.; Alimisis, V.; Fox, M.; and Long, D. 2013. Combining a Temporal Planner with an External Solver for the Power Balancing Problem in an Electricity Network. In *Proc. Int. Conf. on Automated Planning and Scheduling (ICAPS)*.

Ponzoni, C.; Chanel, C.; Lesire, C.; and Teichteil-Knigsbuch, F. 2014. A Robotic Execution Framework for Online Probabilistic (Re)Planning. In *Proc. Int. Conf. on Automated Planning and Scheduling (ICAPS)*.

Py, F.; Rajan, K.; and McGann, C. 2010. A systematic agent framework for situated autonomous systems. In *Proc. of Int. Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; and Ng, A. Y. 2009. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, volume 3.

Ribas, D.; Palomeras, N.; Ridao, P.; Carreras, M.; and Mallios, A. 2012. Girona 500 AUV: From Survey to Intervention. *Mechatronics, IEEE/ASME Transactions on* 17(1):46–53.

Simmons, R. 1992. Concurrent planning and execution for autonomous robots. *IEEE Control Systems* 12(1):46–50.

Srivastava, S.; Fang, E.; Riano, L.; Chitnis, R.; Russell, S.; and Abbeel., P. 2014. Combined Task and Motion Planning through an Extensible Planner-Independent Interface Layer. In *Proc. Int. Conf. on Robotics and Automation (ICRA)*.

Tenorth, M., and Beetz, M. 2009. KnowRob – Knowledge Processing for Autonomous Personal Robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4261–4266.

Tenorth, M.; Bartels, G.; and Beetz, M. 2014. Knowledge-based Specification of Robot Motions. In *Proc. European Conf. on AI (ECAI)*.

Tenorth, M.; Kunze, L.; Jain, D.; and Beetz, M. 2010. KNOWROB-MAP - knowledge-linked semantic object maps. In *Humanoids*, 430–435.