Toward Mobile Robot Navigation in Unstructured Environments Using Topology-Aware Efficiently Adaptive State Lattices*

Ashwin Satish Menon, Eric R. Damm, and Thomas M. Howard

***Abstract*— Contemporary mobile robot navigation architectures that employ a planning algorithm to provide a single optimal path to follow are flawed in the presence of dynamic and uncertain environments. As the environment updates and the robot's starting state changes, optimal plans often oscillate around discrete obstacles, which is problematic for path following controllers that are strongly biased to follow the planned route. In this paper, we reformulate the search process employed by Efficiently Adaptive State Lattices (EASL) to exploit homotopy classes extracted from an observed environment. This approach, which we call Topology-Aware Efficiently Adaptive State Lattices (TAEASL), performs heuristic search using multiple data structures to control expansion of nodes in the graph to provide multiple minimum-cost plans in distinct homotopy classes. Inspired by approaches such as Anytime Repairing A*, search continues until no further expansions can be performed or a maximum search time has been reached. To validate TAEASL's utility in field robotics, it was tested on real-world, off-road environment data that was collected by a Clearpath Warthog unmanned ground vehicle (UGV) and was able to generate multiple solutions. The paper concludes with a discussion of applications including high-speed off-road mobile robot navigation in cluttered obstacle fields.**

I. Introduction

Motion planning algorithms for mobile robots that search for the optimal route given an environment and robot model are pervasively applied across various domains. As illustrated in Figure 1, there are certain applications where a planning framework that can produce multiple distinct plans is beneficial. Mobile robot path planning is one such application. Since the cost map is dynamically updated from sensor observations, the planner may return single, optimal plans that periodically alternate around either sides of obstacles, which burdens the path-following controller with a more difficult obstacle avoidance problem [4] [6]. Recent research into algorithms that exploit homotopy classes has led to planning approaches that are capable of returning multiple routes for use cases such as the aforementioned one. As in [5], we adopt a definition of homotopy as a topological formalism to compare the similarity of paths. Given two paths that start and terminate at the same states, we describe the paths as belonging to the same homotopy class if one can be continuously deformed into the other without encroaching on any obstacle.

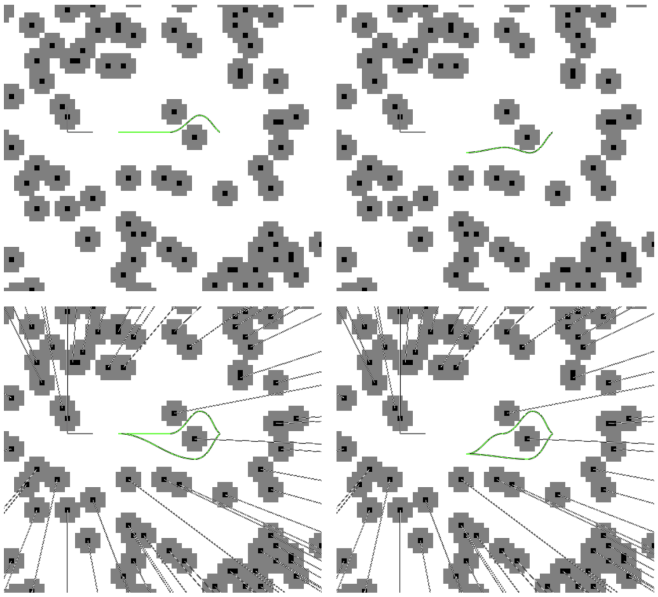

Fig. 1: Top row: Kinodynamic EASL [1] generating plans from (2.0, 0.0) to (6.0, 0.0) (left) and from (2.5, -0.75) to (6.0, 0.0) (right). Bottom row: TAEASL generating plans from the same respective states. The lines emanating from obstacles in the bottom row are reference frames. They are used to partition the world into separate homotopy classes (explained in detail in Section II). Note that as the start state differs from the left map to the right map, KEASL's single solution switches into a different homotopy class. However, TAEASL generates two solutions in both maps, providing a diversity of options for a path-following controller in obstacle-dense environments.

An algorithm recently developed for mobile robot motion planning in cluttered environments is the Efficiently Adaptive State Lattice (EASL) [3]. This method iteratively refines the search space over multiple planning cycles to increase the relative optimality of heuristic search. Further extensions, such as Kinodynamic EASL (KEASL) [1] extend the representation of the search space to satisfy position and orientation dependent velocity constraints to generate

*Research was sponsored by the Defense Advanced Research Projects Agency (DARPA), and was accomplished under Contract Number HR001121C0189. Any opinions, findings, conclusions or recommendations expressed herein are those of the author(s) and do not reflect the views of the Defense Advanced Research Projects Agency or Carnegie Mellon University. Additional research was also supported by the DEVCOM Army Research Laboratory (ARL) and was accomplished under Cooperative Agreement Number W911NF-20-2-0106. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Ashwin Satish Menon, Eric R. Damm, and Thomas M. Howard are with the Robotics and Artificial Intelligence Laboratory, Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY, USA amenon4@ur.rochester.edu

trajectories between boundary states. Although these approaches have improved the efficiency, fidelity, and optimality of heuristic search for mobile robots in unstructured environments, they are not formulated to exploit homotopy constraints to return multiple distinct solutions. In an attempt to remedy the oscillatory behavior seen in planners which output a single plan, this paper introduces Topology-Aware Efficiently Adaptive State Lattices (TAEASL). This contribution maintains the benefits of these recombinant motion planning search spaces but returns one or more optimal solutions subject to homotopy constraints informed by the cost map. Results indicate that TAEASL is able to generate multiple low-cost solutions in dense obstacle fields with feasible planning time constraints.

II. Method

Topology-Aware EASL makes use of the mathematical formalisms presented in [7] that describe how homotopy constraints are defined. This involves segmenting the robot's surrounding environment by drawing lines in cost maps which are anchored by a single center point and each obstacle's centroid. The lines are treated as separate "reference frames".

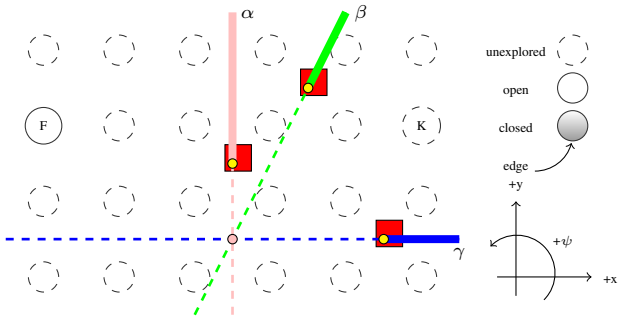

Fig. 2: TAEASL follows the procedure in [7] by sampling points in the obstacle and generating lines from the center point of the map that do not result in coincident lines.

In Figure 2, the cost map cells are parsed to find cells of lethal cost. If such cells exist, the 8-connected neighboring cells will be recursively searched and combined into the same obstacle cluster until no neighboring cells are lethal. The clusters are stored in a vector of obstacle clusters, where the cluster's centroid serves as the anchor point for the reference frame. The figure further illustrates how reference frames are then drawn with the two points being the obstacle centroid and the map's sampled center point. We depart from the method in [7] by including only sampled lines which are on the far side of the obstacle relative to the center of the map, which reduces the number of reference frame crossings to track. If the frame encounters a new obstacle on its way to the map boundaries, the frame ends. Each frame has a unique string identifier for identifying which reference frames a path may have crossed.

Figure 3 shows the first three expansions of the search process, which is based on A* search [2]. The path begins with no reference frame crossings, so we describe the path as belonging to the homotopy class $\varnothing$. As search expands from $(H, 0°)$ to $(I, 0°)$ the reference frame α is crossed, meaning that the path from $(F, 0°)$ to $(I, 0°)$ now contains edges that belong to two different homotopy classes, $\varnothing$ and α. To ensure that we continue searching along paths that go to the left and right of the first encountered obstacle, we add a second priority queue that only includes nodes belonging to the α homotopy class.

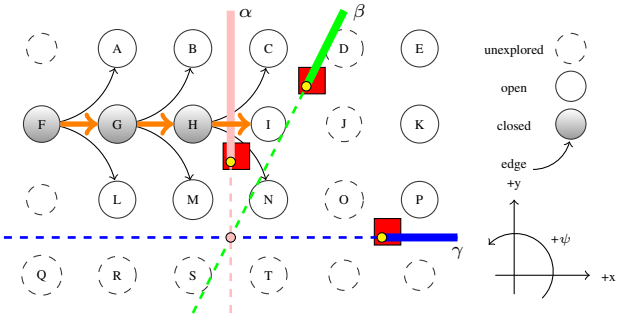

Fig. 3: The edge from $(H, 0°)$ to $(I, 0°)$ crosses reference frame α, meaning a new open list is associated with homotopy class α, whereas the previously expanded nodes belonged to an open list associated with homotopy class $\varnothing$.

Figure 4 shows two expansions for each of the two priority queues defined by homotopy classes $\varnothing$ and α. A solution is found from $(F, 0°)$ to $(K, 0°)$ in homotopy class α. Search continues in the other priority queues to explore alternative routes.

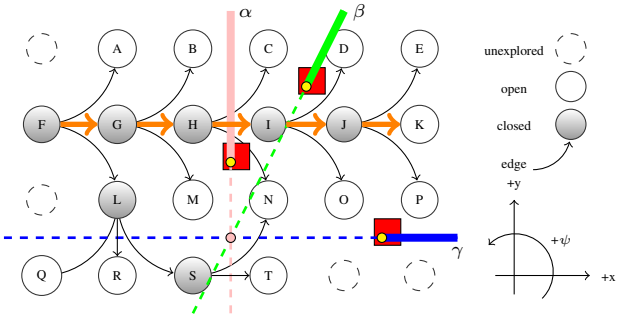

Fig. 4: TAEASL search continues in two directions by expanding two nodes in each homotopy class ($(L, 270°)$, and $(S, 0°)$ in $\varnothing$ and $(I, 0°)$, and $(J, 0°)$ in α) and finds a route from $(F, 0°)$ to node $(K, 0°)$ through $(G, 0°)$, $(H, 0°)$, $(I, 0°)$, and $(J, 0°)$. Since this route crossed the α boundary, we describe this solution as belonging to homotopy class α.

Figure 5 shows a second route belonging to homotopy class $\varnothing$ after two additional expansions of that priority queue. TAEASL continues until no further priority queues are capable of expanding further or a maximum time of search is reached.

III. Experiments

TAEASL was tested on multiple real-world, off-road environments in a Mid-Atlantic forest biome, with one second to

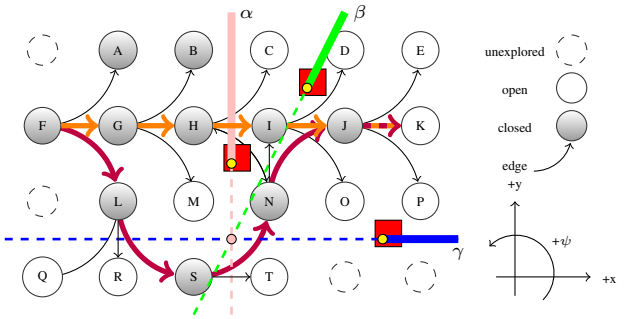

Fig. 5: After two additional expansions, a second route is found from $(F, 0°)$ to node $(K, 0°)$ through $(L, 270°)$, $(S, 0°)$, $(N, 90°)$, and $(J, 0°)$. Since this solution crossed no reference frames, we describe this solution as belonging to homotopy class $\varnothing$.

generate plans. There were two types of hardware validation performed. First, the environment data was collected on-robot and then TAEASL was used to generate plans offline in the logged maps. This is the data shown in Figures 6 and 7. Second, TAEASL ran live on the robot in the same test environments where the original map data was collected.

Figure 6 shows two and Figure 7 shows three resulting trajectories from TAEASL during physical field testing experiments. Red cells are obstacles, gray cells are traversable space, and black cells are unobserved space. Green segments of the solution indicate faster planned robot velocities while blue segments indicate slower velocities. In both examples, as the robot moves away from the most clearly observed part of the map towards the edge of the map, the planned velocity decreases. This is done by the kinodynamic search space being used to plan slower speeds where the perceptual horizon is filled with greater uncertainty. In the same way a human driver would slow down if they did not know their surroundings, the robot reduces speed when its environment is largely unknown.

The multicolored lines emanating from the centers of the maps are the reference frames. Note that only a few obstacles in the center of each map have associated reference frames. Specifically, in Figure 6, there are only four obstacles with frames emanating from them. In this map, the *frame_radius* parameter was set to 5.0 meters. Figure 7 contains many more reference frames due to the *frame_radius* being set to 10.0 meters. This parameter dictates which obstacles are are chosen for frame initialization. Only those within the specified radius from either the center of the map or the robot's position are selected. Figure 6 is measured from the map center and Figure 7 is measured from the robot's position.

This parameter exists to limit the number of solutions TAEASL can generate. Ignoring winding topologies, there are 2^n possible plans our approach can return, where n is the number of obstacles initialized with reference frames in the map. In dense, forested areas such as the ones TAEASL was tested in, there would be too many hypothetical solutions (if every obstacle in the map had an associated frame), often with not enough topological deviation between them. For a path-following controller, three solutions across three sufficiently different homotopy classes is more useful than 20 or more solutions where many of those solutions are very similar to each other. TAEASL and KEASL [1] were integrated and used to enable topologically aware, kinodynamic motion planning through an unstructured forest environment.

IV. Discussion

The motivation behind this contribution emerged when observing EASL [3] run on an Ackermann-steered robot in tandem with a path-following controller in an unstructured, off-road environment. As the robot got progressively closer to a large, lethal obstacle, the optimal plan would "jump" back and forth across the obstacle, essentially switching homotopy classes every second. The perceptual horizon would change due to the robot moving through the world, meaning the cost map was dynamically updated, yielding minimum-cost-to-goal plans which did not necessarily maintain a solution within the same homotopy class.

While the path was oscillating across homotopy classes, the controller would still attempt to track it. Due to the vehicle's steering angle constraints, it would drive closer to the obstacle it was trying to avoid in the first place until it was close enough that a manual safety intervention was necessary. In this same scenario but with TAEASL as the planner instead of EASL, the controller would have had various options to choose from, including the option to track a path that exists in the same homotopy class the controller's current action was in, hypothetically avoiding the need for a manual intervention.

Thus, the TAEASL approach presented here is the first of two steps in creating a planner-controller interface that can be inserted into an autonomous navigation stack which has existing perception and state estimation pipelines. As mentioned in Section III, TAEASL has been run live on a Clearpath Warthog in obstacle-dense forested areas. Initial field testing has been successful in generating optimal plans in distinct homotopy classes, indicating that hardware validation has been shown via two different modalities: collected data logs and live robot runs. The second step will be to develop a path-following controller that has a multi-plan interface (MP-PFC) such that it can consider several TAEASL plans at once.

In order for the MP-PFC to correctly make use of TAEASL, it must have a parameterized *cost_threshold* value. We define $\Delta_{cost_{ab}}$ as the difference between the cost to goal of the robot's current controller action, in homotopy class a, to the cost to goal of the action needed to track a TAEASL plan in homotopy class b. The robot will cease to track the path in a and begin to track the path in b if and only if $\Delta_{cost_{ab}} > cost\_threshold$. If $\Delta_{cost_{ab}}$ is very small and only yields a marginal improvement in cost to goal ($<$ *cost_threshold*), the robot will continue tracking the path in a. This threshold is necessary to enable TAEASL's utility — if no such value was supplied, the robot would greedily select

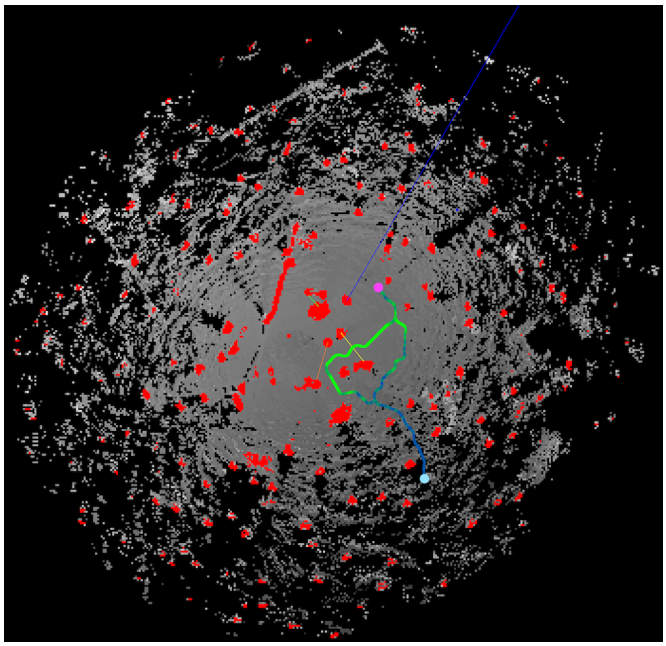

Fig. 6: TAEASL generates two kinodynamic solutions from the start state (magenta) to the goal state (cyan). Green path segments = higher robot velocities, blue path segments = lower robot velocities. Red cells = obstacles, gray cells = traversable space, black cells = unobserved space. Note the reduction in planned speed as the robot moves towards an area with more uncertainty.

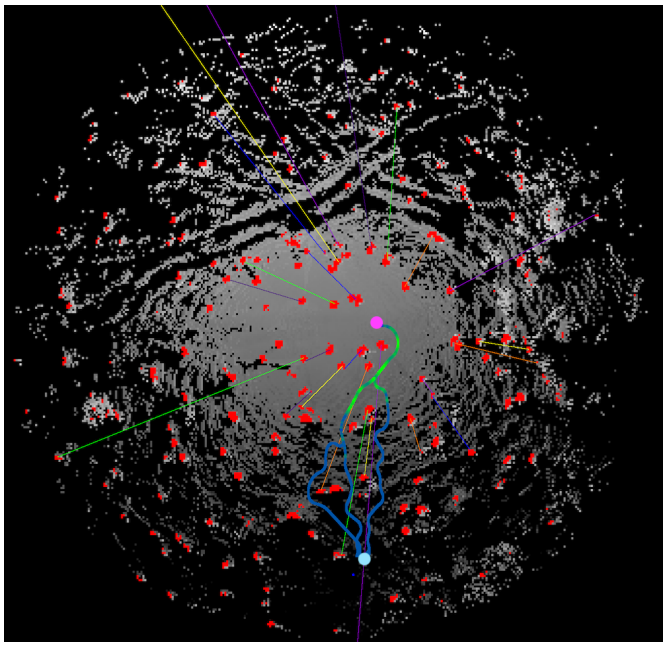

Fig. 7: TAEASL generates three kinodynamic solutions from the start state (magenta) to the goal state (cyan). Green path segments = higher robot velocities, blue path segments = lower robot velocities. Red cells = obstacles, gray cells = traversable space, black cells = unobserved space. Note the reduction in planned speed as the robot moves towards an area with more uncertainty.

the control action that would minimize cost at the risk of unsafe driving behavior. This would cause a reoccurrence of the very problem the TAEASL/MP-PFC paradigm attempts to solve. Put simply, *cost_threshold* must be large enough to be worth the risk that is associated with tracking a path in a different homotopy class.

V. Conclusion

In this paper, we have presented an extension of the Efficiently Adaptive State Lattice algorithm, termed the Topology-Aware Efficiently Adaptive State Lattice algorithm. By introducing homotopy constraints, we are able to enforce that our planner returns multiple solutions in a recombinant search space, each of which is the optimal path in a given homotopy class. This has been done primarily to ease the burden of a path-following controller in an autonomous navigation stack when negotiating through cluttered environments. The results of this contribution indicate that TAEASL is able to generate multiple high-quality solutions, even when restricted to aggressive search time constraints such as one second.

TAEASL was tested in a real-world environment and was successful in returning multiple solutions, even when incorporating kinodynamic constraints. Future work will consist of running TAEASL in tandem with a controller that has a multi-path interface in similar environments to the ones shown in Figures 6 and 7. By supplying a path-following controller with a diversity of solutions across homotopy classes, the goal is to both reduce the controller's burden with regard to collision avoidance and increase the likelihood of safer autonomous navigation in very obstacle-dense environments.

References

[1] Eric R. Damm, Jason M. Gregory, Eli S. Lancaster, Felix A. Sanchez, Daniel M. Sahu, and Thomas M. Howard. Terrain-aware kinodynamic planning with efficiently adaptive state lattices for mobile robot navigation in off-road environments. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9918–9925, 2023.

[2] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems, Science, and Cybernetics*, SSC-4(2):100–107, 1968.

[3] Benned Hedegaard, Ethan Fahnestock, Jacob Arkin, Ashwin Menon, and Thomas M Howard. Discrete optimization of adaptive state lattices for iterative motion planning on unmanned ground vehicles. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5764–5771. IEEE, 2021.

[4] Thomas M Howard, Colin J Green, and Alonzo Kelly. Receding horizon model-predictive control for mobile robot navigation of intricate paths. In *Field and Service Robotics: Results of the 7th International Conference*, pages 69–78. Springer, 2010.

[5] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.

[6] Grady Williams, Andrew Aldrich, and Evangelos Theodorou. Model predictive path integral control using covariance variable importance sampling. *arXiv preprint arXiv:1509.01149*, 2015.

[7] Daqing Yi, Michael A Goodrich, and Kevin D Seppi. Homotopy-aware rrt*: Toward human-robot topological path-planning. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 279–286. IEEE, 2016.