CSE634: Algorithms in Robot Planning

Technical Report

Title: PRM-RL: Long-range Robotic Navigation Tasks by Combining Reinforcement Learning and Sampling-Based Planning

Authors: Aleksandra Faust, Kenneth Oslund, Oscar Ramirez, Anthony Francis, Marek Fiser, James Davidson and Lydia Tapia

Report By: Akanksha Singal (2021008)

December 13, 2024

Keywords

Reinforcement Learning, Sampling-Based Planning, Robot Navigation, Probabilistic Roadmaps, Hierarchical Planning

1 Impact of the Work

The paper[1] presents a novel hierarchical planning approach called PRM-RL, combining reinforcement learning (RL) and sampling-based planning for solving long-range robotic navigation tasks. The integration addresses the limitations of each method individually: RL's inefficiency in long-range navigation and sampling-based planners' inability to handle dynamic constraints and noise. The proposed approach is demonstrated on two challenging tasks: indoor navigation using a differential drive robot and aerial cargo delivery using a quadrotor UAV. PRM-RL achieves significant improvements in task success rates, showcasing robust and adaptable performance under noisy conditions and dynamic constraints.

Challenges

We will motivate the use of PRM-RL by discussing the challenges in long range navigation task and limitations of only PRM algorithms and only RL based agents for this task.

- **Long Range Navigation Tasks:** Long range path planning in complex environments with noisy data, task constraints and unstable dynamics that translate easily to physical robots is difficult.
- **Sampling Algorithms:** Most sampling algorithms are agnostic to the dynamics of the robots and the constraints of the task. They result in poor performance when evaluated on physical robots and cannot handle changes in the environment.
- **Reinforcement Learning:** RL agents are hard to train, take time and numerous runs to converge. Long-range navigation over complex maps have sparse rewards, making training the agent a difficult problem.

Significance

PRM-RL is designed combine the a path planning capabilities like most RPM algorithm and a controller which RL agents try to emulate. The PRM-RL algorithm incorporates a planner and controller to design the final path allowing it to be easily translated to the real world with high performance. PRM-RL overcomes the limitations of PRMs and RL agents by using them to address each other's shortfalls. This is proved by the emperical results shown in the paper. PRM-RL successfully navigates 215m long trajectory under noisy sensor conditions. The aerial cargo delivery successfully navigates 1000m without violating the task constraints. This approach was compared with baseline PRM Straight Line (PRM-SL) and PRM-RL consistently showed superior performance with comparable graph sizes.

Contributions

- Proposed a novel approach combining reinforcement learning (RL) and sampling-based planning for solving long-range robotic navigation tasks.
- Evaluated the approach on 2 case studies in simulation and physical world
  - Case Study 1: Indoor navigation - A differential drive robot to navigate inside buildings. Task - to avoid static obstacles using only its noisy LIDAR sensor. RL agent is trained on simulator with noisy sensors and dynamics designed to emulate the unprocessed, noisy sensor input of the actual robot.
  - Case Study 2: Aerial cargo delivery - A quadrotor with a suspended load to transport the cargo. Task - to minimize residual oscillations while maintaining load displacement below a given upper bound. This quadrotor-load system is non-linear and unstable.
- Their PRM-RL approach outperforms their individual components - PRM and RL.

Applicability

The work on PRM-RL presents a robust hierarchical method combining Probabilistic Roadmaps (PRMs) with Reinforcement Learning (RL) for long-range robotic navigation and showing significant potential in various real-world scenarios. Thes problems studied in this paper shows application in real world problems like cargo delivery using UAVs, navigation in environments where communication is unreliable, continued navigation post sensor damage are only to name a few. PRM-RL can be utilised in logistics, urban delivery systems, and disaster response, where robots must traverse complex terrains while responding to real-time disturbances and maintaining safety. Additionally, its modular design allows scalability across different robot types and environments.

2 Technicalities of the problem definition

Objective of Long Range Navigation Tasks

- The robot should move safely over a substantial distance.
- The robot should satisfy task constraint.
- Long range collision free path finding
- Local control of robot with noisy sensor data or unstable dynamics
- The control should keep the robot should near the obstacle-free path

If all of the above conditions are met, then we can state that the long range navigation task was successful.

Table of Symbols and Definitions

Symbol	Definition
S	State space, the set of all possible robot states.
s	Start state, $s \in S$.
g	Goal state, $g \in S$.
A	Action space, the set of all possible robot actions.
$p(s)$	Projection of the state s onto the configuration space.
C_{free}	Collision-free subset of the configuration space.
$\pi(s)$	Policy mapping states to actions.
γ	Discount factor for future rewards.
$R(s)$	Reward function for the state s.
$f(s, a)$	Transition function defining the next state s_{t+1}.
ε	Threshold for the proximity to the goal.

Table 1: Table of symbols and their definitions used in the paper.

Stages of PRM-RL Algorithm

- RL agent training
- Roadmap creation
- Roadmap querying

Problem Definition and Formal Expressions

State Space (S):

The robot operates in a continuous, multi-dimensional state space $S \subseteq \mathbb{R}^{d_s}$. Each state $s \in S$ represents the robot's observation space, combining:

- **Robot-specific parameters:** e.g., position, velocity.
- **Sensor inputs:** e.g., LIDAR observations or environmental feedback.

Action Space (A):

The robot's action space $A \subseteq \mathbb{R}^{d_a}$ defines all possible control actions. For instance:

- **Indoor navigation:** uses wheel speed vectors $a = (v_l, v_r)$.
- **Aerial cargo delivery:** uses acceleration vectors $a = (a_x, a_y, a_z)$.

Task Constraints ($L(s)$):

A state $s \in S$ is valid if:

$$L(s) = \text{True if } s \text{ satisfies all task constraints (e.g., collision-free, safe dynamics).}$$

Markov Decision Process (MDP):

The navigation task is modeled as an MDP (S, A, P, R):

- $P : S \times A \to S$: Transition probabilities between states, representing the robot's dynamics.
- $R(s) : S \to \mathbb{R}$: Reward function incentivizing task completion and adherence to constraints.

The goal is to find an optimal policy $\pi^*(s) : S \to A$ that maximizes cumulative rewards:

$$\pi^*(s) = \arg\max_{a \in A} \mathbb{E}\left[\sum_i \gamma^i R(s_i)\right],$$

where $\gamma \in [0, 1)$ is the discount factor.

Long-Range Navigation Objective:

Given start state s_{start} and goal state s_{goal}, the robot must find a trajectory:

$$\text{Trajectory } \{s_0, s_1, \ldots, s_n\} \text{ such that:}$$

$$\forall i,\ L(s_i) \text{ holds, and } \|p(s_n) - p(s_{\text{goal}})\| \leq \epsilon,$$

where $p(s)$ projects the state onto the configuration space (C-space), and ϵ is a goal tolerance threshold.

PRM-RL Integration:

- PRMs build a roadmap of states in C-space by sampling and connecting nodes. Two states are connected only if:

  $$\text{Success Rate} \geq p_{\text{success}},$$

  where the success rate is determined by running Monte Carlo trials using the RL agent as a local planner.
- The RL agent learns to perform short-range navigation between sampled states while adhering to task dynamics.

3 Approach

Solution

To resolve the challenges mentioned in section 1, the PRM-RL exploits their individual properties discussed below. The hierarchy can be visually interpreted from the Image **??**.

- **Sampling Algorithms:** builds a roadmap using RL agent for connectivity. Instead of a straight line path between the existing roadmap and a new sample point, PRM connects the points only if RL agent can perform collision free local point to point connection between them.
- **Reinforcement Learning:** used for local path planner to connect two points with task constraints, system dynamics and sensor noise. RL agent is independent of long-range map. It is not concerned with the C-space.

PRM-RL consists of three stages:

- **Reinforcement Learning Agent Training:** RL agents are trained to perform short-range, point-to-point navigation tasks. Algorithms like Deep Deterministic Policy Gradient (DDPG) and Continuous Action Fitted Value Iteration (CAFVI) are employed for training.
- **Roadmap Creation:** PRMs are constructed using the trained RL agent as a local planner. Nodes are sampled in the configuration space, and edges are added only if the RL agent consistently navigates between them under task constraints.

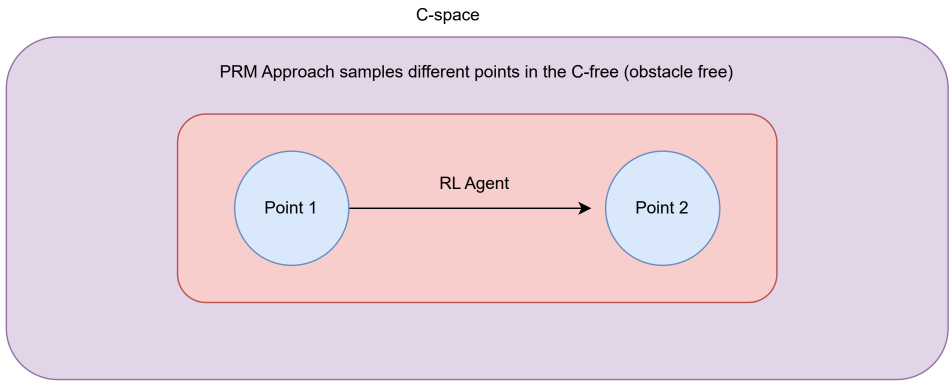

Figure 1: Hierarchical diagram of the approach

- **Roadmap Querying:** For long-range tasks, the roadmap is queried to generate feasible trajectories, which are executed using the RL agent.

Advantages: PRM-RL leverages the strengths of both RL and PRMs, which is the reason for its superior performance.

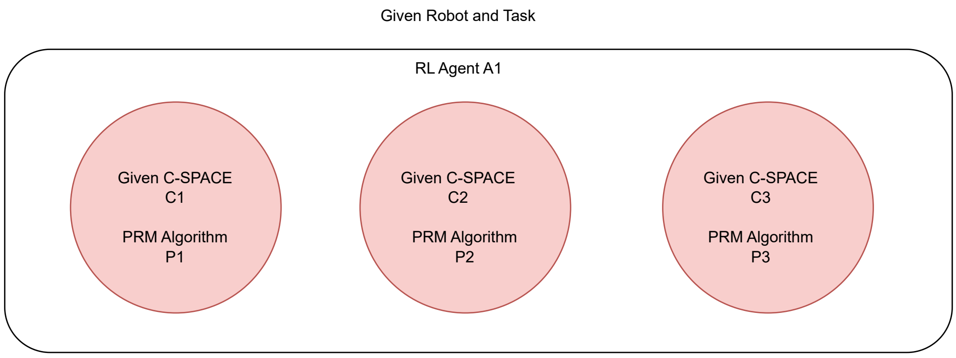

Figure 2: Scope of each component of PRM-RL

Algorithm Explanation

Inputs:

- Start (s) and goal (g) states.
- Success threshold (p_{success}), maximum attempts ($numattempts$), and trajectory parameters (e.g., maximum steps, goal tolerance).
- RL policy (π) and system dynamics (D).

State Sampling:

- Sample multiple variations of s and g in the state space to account for noise and variability.

Monte Carlo Trials:

- Use the RL policy π to attempt navigation from s to g, simulating dynamics and checking task constraints ($L(s)$).
- Record success if the RL agent reaches the goal within the specified tolerance (ϵ).

Edge Validation:

- Compute the success rate and average trajectory length from multiple trials.
- Add the edge to the roadmap if the success rate exceeds p_{success}; otherwise, reject the connection.

Output:

- Returns a boolean indicating if the edge is added, along with success rate and trajectory length.

4 Future Work

Some of the future work can include:

- Working on expanding the current RL based agent to be agnostic to the robot dynamics and the task constraints. Can we achieve a higher level of generalization by eliminating awareness of dynamics and task constraints for long range navigation?
- The paper uses Monte Carlo simulation process irrespective of the learning algorithm used. like DDPG, CAFVI etc. Monte Carlo process wait for the entire episode to complete to calculate the reward. It is only after the episode ends do the policy and state-action values are updated. However, we have seen that Temporal Difference Learning (TDL) approach like SARSA and Q-learning lead to faster updates with similar convergence guarantees as Monte Carlo. In TDL, we only need to wait for the next state to occur to update the policy and state-action values thus can be computationally faster than Monte Carlo.
- Given that we are using a RL agent as controller, future work can include on providing a mathematical analysis of the error bounds and proving state space stability that errors converge to zero and the errors should be low.
- Reinforcement Learning agents can be hard to train sometimes and require much longer training time and runs to converge. Imitation learning is a supervised learning objective that can easily generalize to robust and unseen scenarios.

References

[1] Aleksandra Faust, Oscar Ramirez, Marek Fiser, Kenneth Oslund, Anthony G. Francis, James Davidson, and Lydia Tapia. PRM-RL: long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning. *CoRR*, abs/1710.03937, 2017.