Project Title: Cleaning Robot Path Planning using Actor-Critic Reinforcement Learning

Author: Mahmoud M. Shoieb

Description:

This project implements an Actor-Critic reinforcement learning framework for autonomous cleaning robot navigation. The model learns optimal movement policies through continuous interaction with a simulated environment. The report discusses the training process, reward design, and performance comparison with traditional path-planning methods.

Contents

Introduction

This project enhances the robotic cleaning system developed in Assignment 1 by introducing an RL algorithm to optimize the agent's decision-making in a dynamic environment. In this upgraded environment, the robot needs to clean dust particles that are distributed over a 40x40 grid containing both static obstacles-for example, desks and couches-and dynamic obstacles, such as moving objects. In Assignment 2, the Actor-Critic RL algorithm substantially improved the productivity and flexibility of the robot by enabling it to learn an optimal cleaning path incrementally. This report compares the RL-based approach developed in Assignment 2 against the deterministic algorithms - primarily A* - from Assignment 1, details the enhancements to the environment, and discusses in detail the reinforcement learning methodology for improved cleaning by the robot.

Problem Statement

The main goal of this project is to enable a robot to efficiently navigate and clean a predefined environment, avoiding obstacles and optimizing its cleaning strategy. The robot has to make its way through various static obstacles-like desks and sofas-and dynamic obstacles, such as moving objects, for maximum dust collection with minimum redundant movements and no collisions. In Assignment 1, deterministic algorithms like A* and RRT were used to guide the robot along a predefined path. However, in Assignment 2, the reinforcement learning framework lets the robot adapt dynamically to changes in the environment, learn from its actions, and optimize the cleaning strategy in real time.

Environment Design

In the 40 x 40 grid environment, each cell corresponds to part of the workplace of a robotic servant. Dynamic obstacles traverse the grid randomly, whilst static obstacles (e.g. desks and couches) are placed at different positions to replicate real-world scenarios. This can be simulated quite realistically by populating the grid with dust particles in a random manner. Dust is collected when the robot enters a dirty cell, and location is updated based on its learned policy.

The environment is closely tied with reinforcement learning model, allowing the robot to dynamically adapt its actions to the positions of obstacles and dust particles. The state space would then be defined as the current position of the robot on the grid, whereas the action space would consist of four possible movements - up, down, left, and right. Thus, the robot can learn from the environment immediately.

Reinforcement Learning

An Actor-Critic reinforcement learning method helps the robot to make its decision. Here, the Critic measures the value of actions for the robot regarding being close to the goal. An Actor uses a policy for the next move by the robot according to the present state.This is developed iteratively by using the reinforcement learning methodology of Temporal Differences to improve the policy as well as the value function. It enables the robot to modify its knowledge after making an action without requiring the final outcome of that action.

The action space has four actions: up, down, left, and right. The state space is defined by the current location of the robot on the grid, which keeps changing as the robot moves in the environment and interacts with obstacles and dust. The robot receives rewards for picking up dust particles, and penalties for colliding with obstacles or for making wrong moves. Through this reward/penalty scheme, the exploration of better cleaning strategies is incentivized while keeping them adaptable to the dynamic changes in environmentally hazardous conditions.

Aspect	Assignment 1: Deterministic Algorithms	Assignment 2: Reinforcement Learning
Algorithm Used	A* (deterministic graph-based) and RRT (probabilistic sampling-based)	Actor-Critic (reinforcement learning algorithm using policy/value updates).
Pathfinding Approach	Predefined heuristic-driven path-finding algorithms.	Adaptive policy learned through exploration and exploitation.
Navigation Efficiency	Efficient in structured, static environments with minimal dynamic elements.	Adaptable in dynamic environments with frequent updates to policy
Obstacle Handling	Static obstacles: Very efficient. - Moving obstacles: Less efficient; required frequent recalculations.	Handles both static and dynamic obstacles effectively through learned behavior.
Exploration vs. Exploitation	Fixed paths focus on shortest or random feasible routes.	Balances exploration (trying new paths) and exploitation (using known optimal paths).
Optimality	Guarantees optimal paths with A* if heuristic is accurate; suboptimal with RRT.	Achieves near-optimal solutions after sufficient training.
	- High initial efficiency in static environments.	- Lower initial efficiency during training.

Performance Metrics	- Moderate collision avoidance in dynamic environments. - Relatively consistent performance.	- High long-term efficiency as it learns. - Improved collision avoidance and dust collection rates.
Output Performance	Dust cleaning was limited to predefined paths and heuristic goals.	Dust cleaning efficiency increased due to dynamic learning.
Exploration of Unvisited Areas	Limited; relies on shortest-path logic that might repeatedly visit cleaned areas.	Explores unvisited areas dynamically based on rewards and cleaning efficiency.
Initial Results	Quicker to start with predefined efficiency but lacks adaptability.	Requires training phase, but performance improves with experience.

Results and Observations

The reinforcement learning technique significantly improved the performance of the robot in dynamic conditions. Compared to the deterministic approaches, the robot removed a higher percentage of dust, especially when dynamic obstacles were present. The reinforcement learning system enabled the robot to effectively navigate less-explored areas by balancing exploration (trying new paths) and exploitation (using learned, optimal routes). However, during the initial training phase, the exploration process resulted in slower performance and hence delayed the optimal performance for some time. The main performance metrics, such as the proportion of dust collected, number of steps taken, and number of accidents avoided, reflected the gradually growing reliability and effectiveness of the learning-based method. Further training made the robot more efficient by reducing unnecessary explorations and enhancing cleaning efficiency.

Challenges and Solutions

Other main challenges while implementing the reinforcement learning model are exploring-exploitation trade-offs, optimization of learning rates, and reward shaping. To counter this, the grid-based model is implemented, providing efficient state-action mapping for easy maneuverability within the dynamic environment. This includes fine-tuning the hyperparameters of the learning rate and exploration factor to realize a better balance in exploration of new paths while exploiting optimal routes that are already learned. Moreover, the strategy was updated constantly by the robot to handle the dynamic obstacles. While this update is computationally expensive, it allowed for huge improvements in avoiding obstacles so that the robot could better navigate complex, dynamic environments.

References

Papers

Mnih, V., Kavukcuoglu, K., Silver, D. et al. (2015). Human level control through deep reinforcement learning. Nature 518, 7540, 529-533.
Link: https://www.nature.com/articles/nature14236

Lillicrap, T. P., Hunt, J. J., Pritzel, A., et al. (2015). Continuous control with deep reinforcement learning.
Link: https://arxiv.org/abs/1509.02971

Schulman, J., Wolski, F., Dhariwal, P., et al. (2017). Proximal Policy Optimization Algorithms. https://arxiv.org/abs/1707.06347

By Mnih, V., Badia, A. P., Mirza, M., and others (2016). Asynchronous Methods for Deep Reinforcement Learning. https://arxiv.org/abs/1602.01783

Videos

https://www.youtube.com/watch?v=2pWv7GOvuf0&list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

https://www.youtube.com/watch?v=fdY7dt3ijgY

https://www.youtube.com/watch?v=LawaN3BdI00