



Reto de Robótica Móvil y Aprendizaje Reforzado

Exploración Planetaria para la NASA

Trama:

Han sido contratados por la NASA para liderar la primera misión autónoma en un planeta desconocido. El objetivo principal es programar un robot móvil capaz de explorar y mapear el terreno de manera autónoma, identificar características del terreno y encontrar la ruta más eficiente desde cualquier punto del planeta hasta un punto objetivo, aplicando algoritmos de diferencia temporal.

Objetivo del Reto

Diseñar e implementar un sistema autónomo que utilice técnicas de aprendizaje por refuerzo para que el robot:

1. Reconozca y mapee el terreno.
2. Generar una tabla de estados que describa las posibles posiciones y acciones a ejecutar en cada estado del mapa.
3. Aprenda a navegar desde cualquier posición hacia la meta utilizando los algoritmos de **Q-Learning** y **SARSA**.
4. Optimice el proceso con un modelo basado en diferencias temporales

Instrucciones

1. Exploración del Terreno:

- Se tiene acceso a un telescopio con una cámara el cual es capaz de detectar mediante video en tiempo real generalidades del terreno (si es roca firme o lava de un volcán que hizo erupción recientemente). Es de aclarar que El terreno tiene forma de laberinto.
- El resultado de este mapa se encuentra en la variable “maze” del script anexo en el repositorio: <https://github.com/felipebuitragocarmona/recursos-practica-robotica-aprendizaje-por-refuerzo/blob/main/analisisMapa.py>
- El robot debe recorrer posición por posición del terreno para confirmar el resultado de las imágenes satelitales y re armar el mapa basado en su exploración. Es de aclarar que las imágenes en tiempo real permitirán saber la ubicación exacta del robot (retornando posiciones en ejes x,y. Además, se puede obtener el identificador de la zona en la que está, la cual para la primera casilla es igual a cero y la última es filas*columnas-1). El código para manejar el robot se encuentra en el siguiente enlace: <https://github.com/felipebuitragocarmona/recursos-practica-robotica-aprendizaje-por-refuerzo/blob/main/comunicacionArduino.py>
- Es de aclarar que el inicio y final del laberinto cada vez cambian y se calculan de manera aleatoria.



2. Diseño de la Tabla de Estados:

- Plantee la cantidad de estados y acciones que hay que tener en cuenta para resolver el problema
- Para cada estado, definan las posibles acciones (arriba, abajo, izquierda, derecha).
- Incluyan en la tabla de estados los costos asociados a cada tipo de terreno (ejemplo: Mover en terreno tipo roca da una pérdida de -1, caer a la lava -20 y llegar a la meta +30).

3. Implementación de Algoritmos de Aprendizaje por Refuerzo:

- Programe el algoritmo **Q-Learning** para que el robot aprenda la mejor estrategia de navegación en función del terreno.
- Apliquen el algoritmo **SARSA** como una alternativa y compare los resultados obtenidos en términos de convergencia y eficiencia.

Criterios de Evaluación

1. Precisión del reconocimiento y clasificación del terreno.
2. Correctitud y completitud de la tabla de estados generada.
3. Eficiencia del algoritmo en encontrar la meta desde cualquier posición.
4. Resultados comparativos entre **Q-Learning** y **SARSA** en términos de convergencia y eficiencia.
5. Implementación exitosa del modelo de aprendizaje profundo para manipular el robot de manera autónoma.

Nota: Para los puntos 1 y 5, se calificará que el robot lo haga en el menor tiempo posible con la mejor precisión

Entrega Esperada

- Código funcional que incluya la exploración, generación de tabla de estados, implementación de Q-Learning y SARSA.
- Gráficos que muestren la evolución de las recompensas y las rutas óptimas encontradas.
- Robot en funcionamiento operado de manera autónoma

Este desafío les permitirá aplicar y combinar conceptos avanzados de aprendizaje automático en un contexto altamente innovador. ¡Buena suerte en esta misión interplanetaria! 🚀