



UNIVERSIDAD DE  
GUANAJUATO

# Simulación de Planificación y Navegación Simultánea Multi-Robot en Cuadrícula de Ocupación

Verano UG 2024

Universidad de Guanajuato

Departamento de Ingeniería Electrónica

Campus Irapuato-Salamanca

## Participantes

- Carlo Leonardo Alcocer Soriano
- Jennifer González Ortega
- Gael Alexei García Chaire
- Francisco de Jesús Martínez González
- José Emmanuel Rangel Rodríguez

Asesor: Dr. Juan Pablo Ignacio Ramírez Paredes

Este proyecto presenta una simulación de navegación para múltiples robots en un entorno definido por una cuadrícula de ocupación. La simulación se centra en la generación de trayectorias suaves y la implementación de un controlador cinemático para el seguimiento de dichas trayectorias, con la capacidad de operar en entornos virtuales como CoppeliaSim.

El código fuente y los entornos de simulación para CoppeliaSim están disponibles en <https://github.com/jpiramirez/VeranoUG2024>

## Requisitos Técnicos

\* *Lenguaje de Programación*: Python 3.12.3 (versión sugerida)

\* *Bibliotecas*: NumPy, SciPy, Matplotlib, OpenCV (cv2)

\* *Software*: CoppeliaSim

\* *Módulos Personalizados*: **astargridMultiple**: Implementación del algoritmo A\* para planificación de rutas multi-robot. **coppeliastim\_zmqremoteapi\_client**: Cliente para la comunicación con CoppeliaSim a través de ZeroMQ.

## Resumen Técnico

El proyecto aborda el desafío de coordinar el movimiento de múltiples robots DDR en un espacio compartido, garantizando que cada robot alcance su objetivo individual sin

colisionar con otros robots o con obstáculos en el entorno. Para lograrlo, se utiliza una combinación de técnicas de planificación de rutas y control de movimiento:

1. **Planificación Global:** El algoritmo A\* se emplea para determinar la ruta óptima para cada robot en la cuadrícula de ocupación, considerando tanto la distancia a la meta como los obstáculos presentes.
2. **Interpolación de Trayectorias:** Las rutas obtenidas por A\* se someten a una interpolación spline cúbica, generando trayectorias suaves y continuas para cada robot.
3. **Control de Seguimiento:** Se implementa un controlador cinemático basado en la cinemática inversa de robots DDR para guiar a cada robot a lo largo de su trayectoria interpolada, teniendo en cuenta su velocidad y orientación.
4. **Simulación en CoppeliaSim:** La simulación se realiza en el entorno virtual de CoppeliaSim, donde se modela el comportamiento de los robots DDR y se obtiene información del entorno en tiempo real.
5. **Visualización y Evaluación:** Se generan gráficos para visualizar las trayectorias planificadas y las trayectorias seguidas por los robots en la simulación, permitiendo una evaluación cualitativa del rendimiento del sistema.

### **Características Destacadas**

- \* **Planificación de Rutas Óptimas:** Utilización del algoritmo A\* para encontrar rutas eficientes y libres de colisiones.
- \* **Generación de Trayectorias Suaves:** Interpolación spline cúbica para garantizar movimientos suaves y evitar cambios bruscos en la trayectoria.
- \* **Control Cinemático Preciso:** Implementación de un controlador basado en cinemática inversa para un seguimiento preciso de la trayectoria.
- \* **Simulación Realista en CoppeliaSim:** Integración con CoppeliaSim para validar el comportamiento de los robots en un entorno virtual realista.
- \* **Análisis Visual:** Visualización gráfica de trayectorias y resultados para facilitar el análisis y la evaluación del rendimiento.

### **Consideraciones para el Usuario**

- \* **Velocidad Constante:** Todos los robots se mueven a una velocidad constante durante la simulación.
- \* **Orientación Inicial:** Al inicio, cada robot está orientado directamente hacia su meta.
- \* **Formación de Colas:** Las colas se forman según la constante G (gasto) definida en el código.
- \* **Ordenamiento de Colas:** Las colas se ordenan de menor a mayor según el valor de la constante G.
- \* **Colisiones:** En esta versión inicial, no se verifica ni previene colisiones entre los agentes.

## Parámetros Ajustables para Pruebas

Los siguientes parámetros pueden ser modificados por el usuario para realizar diferentes pruebas y experimentos:

- **NoR:** Número de robots en la simulación.
- **map\_file:** Nombre del archivo de mapa (por ejemplo, mapaVacio.txt o un mapa personalizado).
- **robot\_radius:** Radio de los robots (en metros).
- **gap:** Distancia mínima deseada entre los robots (en metros).
- **cell\_size:** Tamaño de cada celda en la cuadrícula (en metros).
- **random\_seed:** Semilla para la generación aleatoria de posiciones.
- **topvel:** Velocidad máxima de los robots.
- **T, Kv, Kh:** Parámetros del controlador cinemático.
- **rr, L:** Parámetros del modelo del robot (radio de las ruedas y distancia entre ellas).

```
map_file = 'mapaVacio.txt'
robot_radius = 0.25 # meters
gap = 0.2 # meters
cell_size = 0.1 #meters
```

## Códigos de interés:

\* **pruebasAstarGridMultiple.py:** Código para probar algoritmos, y determinar la menor distancia alcanzada entre dos robots durante la ejecución a velocidad constante.

\* **pruebas\_sim\_robots.py:** Archivo para realizar la simulación de algoritmoMR (multiple robots) usando coppelia.

\* **simulacionDatosCsv.py:** Código para ejecutar algoritmo elegido y crear archivo de resultados.

## Instrucciones de Uso

### 1. Configuración:

\* Instalar las bibliotecas y dependencias necesarias (ver “Requisitos Técnicos”).

\* Preparar el archivo de mapa *mapaVacio.txt* o un mapa personalizado en formato compatible.

\* Configurar la escena de CoppeliaSim con los robots y el entorno adecuados.

## **2. Ejecución:**

- \* Ejecutar el script respectivo desde la línea de comandos.
- \* Observar la simulación en CoppeliaSim y los gráficos generados.

## **Contribuciones**

Se invita a la comunidad a contribuir al desarrollo de este proyecto. Las mejoras, sugerencias y correcciones son bienvenidas.