

Práctica 2. Carrera de robots

Francisco Gómez, Diego Viejo, Bessie Domínguez
Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante

10 de abril de 2024

La práctica consiste en el desarrollo del sistema de navegación de un coche que le permita moverse por un circuito y completarlo. El objetivo principal es dar una vuelta completa al circuito en el menor tiempo posible.

En esta parte de la asignatura vamos a utilizar la plataforma robótica **Unibotics**. Esto es una plataforma web para programación de robots que permite editar, ejecutar y depurar código directamente desde el navegador sin necesidad de instalaciones locales, gracias a su backend robótico multiplataforma. Por debajo utiliza ROS y Gazebo de la forma que hemos visto anteriormente en la asignatura, pero nos permite interactuar con estas herramientas a más alto nivel, de forma más abstracta.

La plataforma ofrece diversos ejercicios sobre robots móviles, drones, conducción autónoma y visión por computadora. Soporta programación en Python y es el lenguaje en el que desarrollaremos la práctica.

1. Reglas de la práctica

La práctica consta de dos partes bien diferenciadas. Por un lado, el objetivo principal es que el robot complete de forma totalmente autónoma una vuelta al circuito. Después, hay una segunda parte en la que se debe implementar un pipeline de detección de objetos. La práctica podrá ser llevada a cabo por grupos de dos.

1.1. Navegación (6.5 puntos)

Para solucionar el problema de navegación se pueden usar todas las herramientas que se precisen oportunas. Se podrá acceder a todos los sensores que integra el turtlebot y se podrá usar cualquier tecnología, siempre que con ello se logre el objetivo de forma autónoma. Como propuestas para solucionar el problema se la navegación se sugieren las siguientes soluciones:

- Solución basada en Deep Learning
- Solución basada en robótica tradicional
- Solución basada en GMapping o Cartographer
- Otras soluciones (consultar)

Cada una tiene ventajas y desventajas por lo que se aconseja un estudio previo a la elección de la aproximación. El problema está abierto a que los alumnos propongan formas alternativas de solucionarlo, previa consulta con el profesor.

1.2. Detección de objetos (3.5 puntos)

La detección de obstáculos en vehículos autónomos es de vital importancia, sin embargo nosotros lo vamos a simplificar a la detección de objetos (otros robots u objetos dentro el circuito). En este apartado se sugiere el uso de las siguientes aproximaciones:

- Solución basada en Deep Learning
- Solución basada en Feature Matching

2. Posibles ampliaciones (1 punto adicional)

Se valorarán la proposición e implementación de ampliaciones relacionadas con el objetivo de la práctica y de la asignatura tales como, por ejemplo, la reconstrucción del mapa 2D o 3D del circuito, métodos avanzados de navegación, exploración de diferentes métodos de detección de otros robots, soluciones híbridas o suavizado de decisiones teniendo en cuenta el factor temporal. La realización de este apartado supondría hasta 1 punto adicional.

3. Documentación a entregar

La documentación de la práctica es una parte muy importante en la puntuación final (un 60 % de ella). El código debe estar debidamente comentado, indicando qué se hace en cada punto. Además, se debe entregar un vídeo donde se demuestre el funcionamiento de cada una de las partes de la práctica y una documentación (PDF) con los siguientes puntos:

1. Introducción
2. Propuesta
3. Experimentación
4. Conclusiones
5. Referencias

Normas de entrega de la práctica:

- La práctica se podrá realizar por parejas.
- La práctica se entregará antes de las 24 horas del 29 de mayo del 2024.
- La entrega se realizará a través del Moodle de la asignatura.

Introducción a Unibotics

Pasos a seguir para trabajar con Unibotics

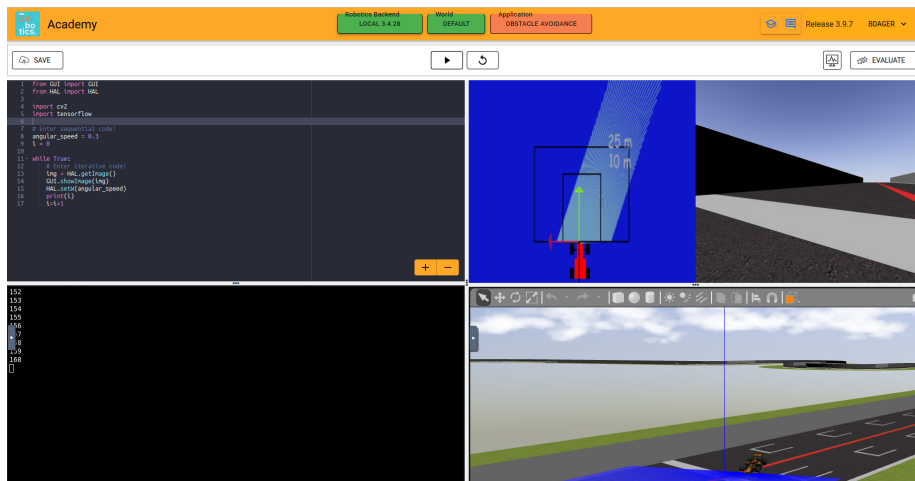
1. Entrar en la web de Unibotics: <https://unibotics.org/>
2. Crearse una cuenta y loggarse.
3. Definir el backend robótico. Escogemos **Local ROS1 (RADI 3)**, que corresponde con ROS1 Noetic. RADI significa: Robotics-Academy Docker Image.(Pasos bien explicados en la propia página de Unibotics).
4. Descargar la imagen de docker del backend robótico (3.4.28) desde DockerHub:

```
docker pull jderobot/robotics-academy:3.4.28
```

5. Una vez descargado, iniciar el backend robótico en local:

```
docker run --rm -it -p 7681:7681 -p 2303:2303 -p 1905:1905  
-p 8765:8765 -p 6080:6080 -p 1108:1108 -p 7163:7163 -p 7164:  
7164 jderobot/robotics-academy:3.4.28
```

6. Cuando ya tenemos la imagen corriendo en local, entramos en la opción de Academy dentro de la web de Unibotics y seleccionamos el problema de **Follow Line / Obstacle Avoidance**.
7. Si hemos entrado correctamente, se habrá conectado la web lanzada al docker local y se empieza a ejecutar el código sobre esa simulación. (Figura 1)



Robot API

Componentes claves de la plataforma:

- **HAL (Hardware Abstraction Layer)**: Es la capa de abstracción de hardware. Este componente de Unibotics actúa como un intermediario entre el hardware del robot (o una simulación de este) y el software de programación. Su función principal es abstraer la complejidad del hardware subyacente, proporcionándonos una interfaz de programación simplificada. Esto nos permite escribir código que pueda interactuar con diferentes tipos de robots o simulaciones sin tener que preocuparnos por los detalles específicos del hardware.

Ejemplos:

- `from HAL import HAL` - La clase contiene funciones que envían y reciben información desde y hacia el hardware (Gazebo).
- `HAL.getImage()`- Obtener imagen
- `HAL.setV()`- definir velocidad lineal
- `HAL.setW()`- definir velocidad angular

- **GUI (Graphical User Interface)**: Es la interfaz gráfica de usuario. Este componente se refiere a la parte visual y gráfica a través de la cual los usuarios interactúan con la plataforma Unibotics. La GUI incluye elementos visuales como botones, menús y ventanas que nos permiten editar código, ejecutar y depurar programas, y visualizar los resultados.

Ejemplos:

- `from GUI import GUI`- importar la clase de la biblioteca de Interfaz Gráfica de Usuario. La clase contiene las funciones utilizadas para ver la información de depuración, como los widgets de imagen.
- `GUI.showImage()`- permite ver imágenes para depurar o información relevante

Juntos, HAL y GUI son fundamentales para proporcionar una experiencia de usuario fluida y eficaz en Unibotics, permitiendo a los programadores centrarse en el desarrollo de algoritmos y soluciones de software para desafíos de robótica sin tener que lidiar con las complejidades del hardware o con interfaces de usuario complicadas.

Enlaces útiles

```
https://jderobot.github.io/RoboticsAcademy/user\_guide/  
https://jderobot.github.io/RoboticsAcademy/exercises/AutonomousCars/  
follow\_line/  
https://jderobot.github.io/RoboticsAcademy/exercises/AutonomousCars/  
obstacle\_avoidance
```