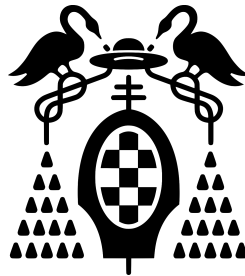


Sistemas de Control Inteligente

Aparcamiento Inteligente

Grado en Ingeniería Informática
Universidad de Alcalá



Pablo García García
Valeria Villamares Félix

14 de enero de 2024

Índice general

Introducción	2
Problema objetivo	2
Condición de parada	3
1. Controlador borroso	5
1.1. Variables	5
1.2. Reglas	5
2. Controlador neuronal	6

Introducción

El problema que se plantea y resuelve a lo largo de esta práctica de laboratorio es el diseño de dos controladores basados en técnicas populares de Inteligencia Artificial como son la Lógica Borrosa y las Redes Neuronales, que sean capaces de realizar el estacionamiento en batería de un vehículo.

Problema objetivo

Se cuenta con un escenario en el simulador STDR de ROS que simula el mapa en el que se deberá mover y aparcar el vehículo. Este cuenta con un carril de $24 \times 4,3$ metros, y ligeramente hacia la mitad de este, en perpendicular, se encuentra una plaza de aparcamiento de $7,8 \times 4,3$ metros. El frontal del vehículo se encuentra al principio de este carril, por lo que la maniobra deberá comenzar marcha atrás.

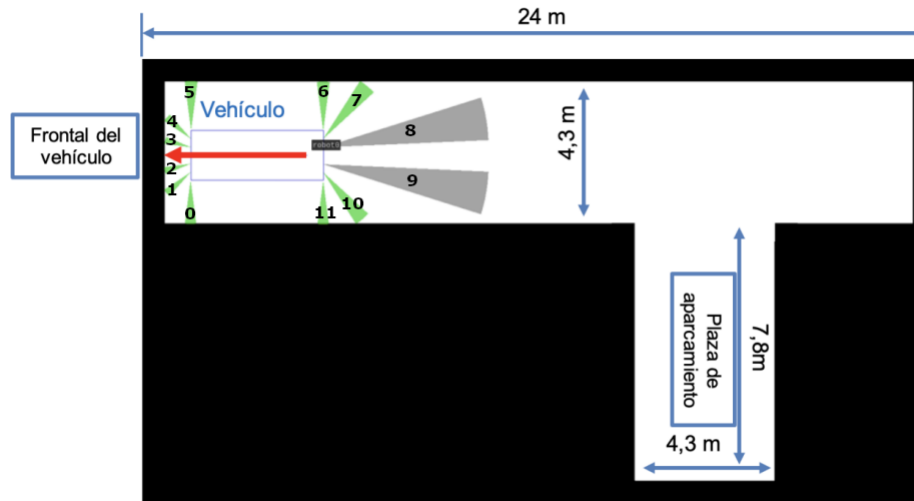


Figura 1: Escenario del problema

El vehículo a controlar tiene unas dimensiones de $4,5 \times 1,5$ metros, puede alcanzar velocidades de ± 30 km/h, y el volante puede girar $\pm 90^\circ$. Como es evidente, no es capaz de girar si no se está moviendo. Además, cuenta con doce sensores de ultrasonidos numerados del 0 al 11, capaces de medir distancias de hasta 5 metros. Finalmente, para el desarrollo de la práctica se empleará ROS, MATLAB, Simulink, y las toolboxes de *Fuzzy Logic*, *Deep Learning*, y *ROS*.

Condición de parada

Como el diseño de la condición de parada del simulador será común a los diferentes modelos y situaciones, será explicada en esta sección introductoria. Esta tarea consiste en que cuando el coche se encuentra en una posición similar a la mostrada en la Figura 2, fijando una distancia a la pared trasera, el vehículo deje de moverse y se pare la ejecución de Simulink.

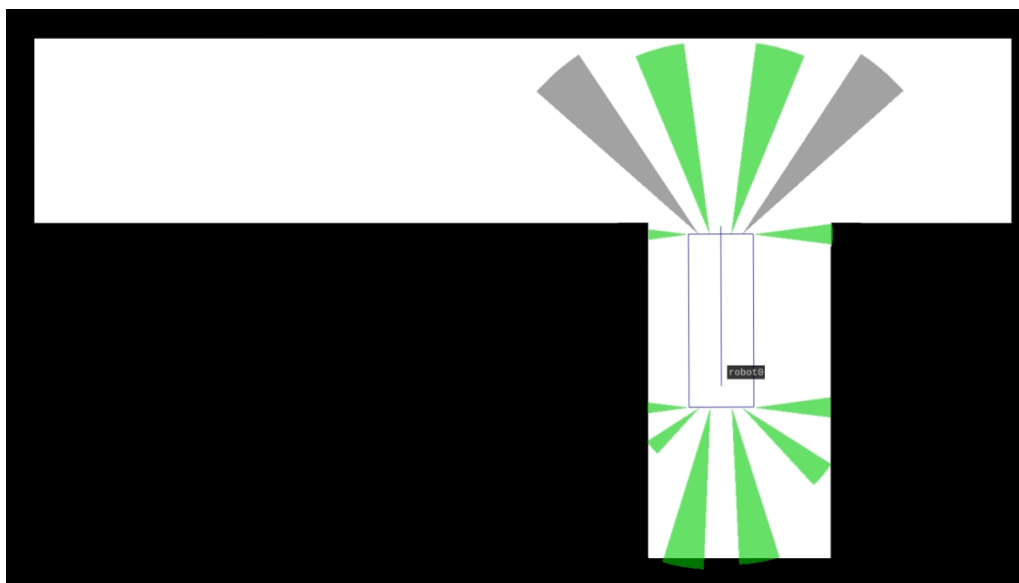


Figura 2: Posición de parada

Para ello, se introduce un bloque en Simulink que representa la ejecución de una función MATLAB, llamado “Condición de parada”. Para fijar un umbral de distancia, se utilizarán los sensores 8 y 9, que son aquellos que se ubican en la parte trasera del coche. Como salida se tiene la velocidad modificada del coche y un *flag* con valor binario para avisar a Simulink de si la ejecución debe continuar o no.

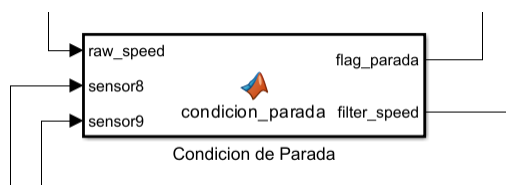


Figura 3: Bloque de condición de parada

El trabajo de este bloque es verificar si ambos sensores reciben una distancia menor a, por ejemplo 0,6 metros (este es el umbral que se fija), y en dicho caso activa el *flag* de parada y pone la velocidad a 0, y en caso contrario devuelve la velocidad que recibe del controlador. Se intenta dejar cerca de la pared, aprovechando el espacio de la plaza, pero sin que llegue a chocarse. El código que se encuentra dentro del bloque es el que se muestra a continuación.

```
1  function [flag_parada, filter_speed] = condicion_parada(raw_speed,
2      sensor8, sensor9)
3      condicionS8 = sensor8 < 0.6;
4      condicionS9 = sensor9 < 0.6;
5
6      if condicionS8 && condicionS9
7          filter_speed = 0.0;
8          flag_parada = 1;
9      else
10         filter_speed = raw_speed;
11         flag_parada = 0;
12     end
```

Código 1: Código del bloque de parada

Parte 1

Controlador borroso

1.1. Variables

1.2. Reglas

Parte 2

Controlador neuronal