Universidad de Alcalá

# Práctica Final

Diseño de Controladores Borrosos y Neuronales para la maniobra de aparcamiento de un vehículo

# Sistemas de Control Inteligente

**Grado en Ingeniería Computadores**
**Grado en Ingeniería Informática**
**Grado en Sistemas de Información**
**Universidad de Alcalá**

# 1. Objetivo

El objetivo de la práctica es el diseño del control de velocidad (lineal y angular) de un vehículo para que éste realice la maniobra de aparcamiento, de la mejor forma y en el menor tiempo posible.

El entorno en el que debe realizar la maniobra se muestra en la Figura 1.

- El entorno donde debe realizarse el aparcamiento en línea es de 24 x 13,4m (contando las paredes)
- El carril por el que circula el vehículo es de 4,3m de ancho.
- El vehículo empieza de espaldas a la plaza de aparcamiento.
- La plaza de aparcamiento tiene un tamaño de 7,8 x 4,3m y se encuentra a la izquierda del vehículo.
- Se considera *posición ideal de aparcamiento* si el vehículo termina centrado en la plaza (tanto longitudinalmente como lateralmente) y alineado con el borde.

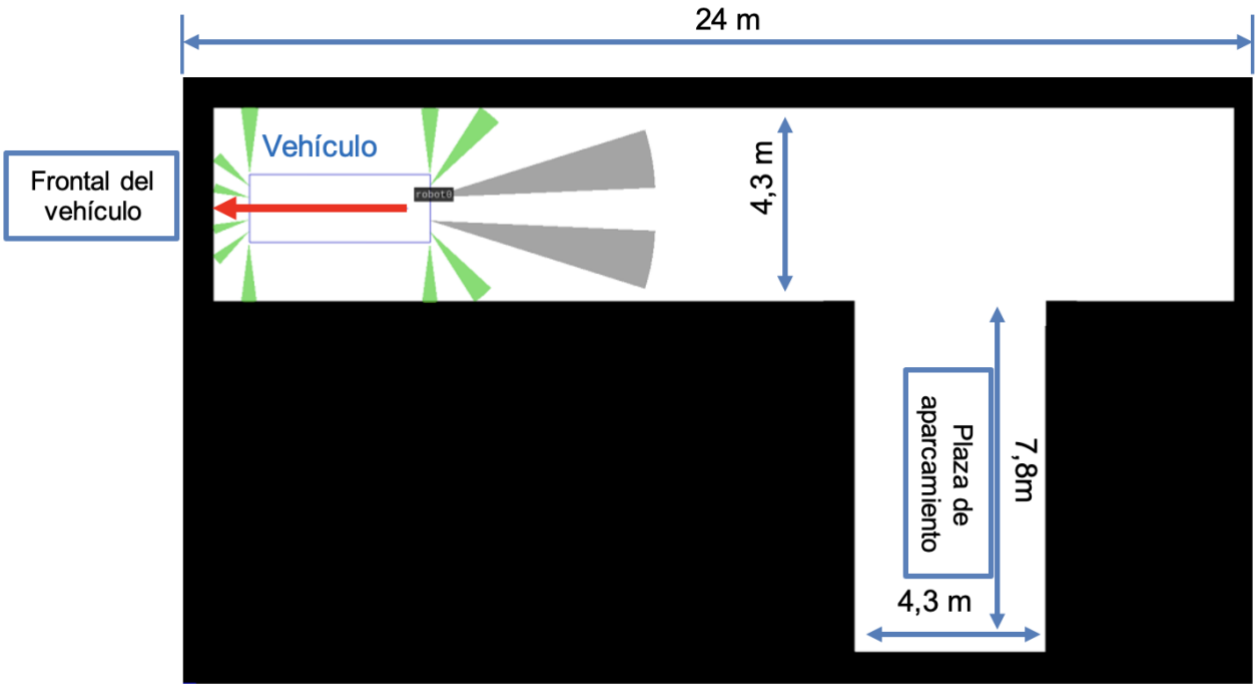



Figura 1. Problema de control propuesto.

# 2. Plataforma robótica - Vehículo

El vehículo sobre el que se probarán los controladores tiene las siguientes características:

| | |
|---|---|
| Longitud | 4.5 m |
| Anchura | 1.5 m |
| Velocidad lineal máx. | +-30 km/h |
| Giro de volante máx. | +-90º |

La configuración del vehículo es tipo Ackerman, es decir, no se desplaza únicamente con el giro del volante, siempre necesita tener velocidad lineal ni es capaz de cambiar su orientación girando sobre sí mismo. El vehículo dispone de 12 sensores de ultrasonidos dispuestos como se muestra en la Figura 2. Los sensores de ultrasonidos tienen un rango de medida de distancia de [0,5] m.

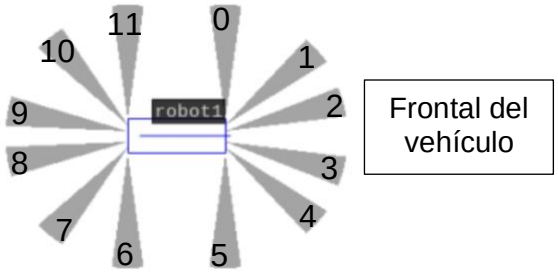



Figura 2. Distribución de sensores de ultrasonidos en el vehículo.

## 3. Entorno de simulación Matlab/ROS

El comportamiento del vehículo y su interacción con el entorno se simula en la plataforma ROS-STDR. Todos los detalles relativos a ROS y al simulador STDR, tanto su instalación como su configuración, desarrollo, ejecución, etc. se detallan en el documento "*Guía Rápida introducción ROS*", disponible en el aula virtual de la asignatura. Si no se tienen conocimientos de ROS, el estudio de dicho documento es indispensable para poder entender el contenido de esta práctica.

En la práctica se proporcionan los siguientes archivos para realizar la simulación:

- Máquina virtual con Linux y ROS (*Ubuntu 20.04 + ROS Noetic + Simulador STDR*)
- Archivo de descripción del mapa:
  `parking1600x895.yaml`
- Archivo *launch* que lanza el STDR, el mapa del entorno y el vehículo:
  `SCI-parking_ackerman_bateria.launch`
- Imágenes utilizadas para crear el mapa:
  `parking1600x895.png`
- Archivo de simulación del vehículo:
  `emulated_ackerman.xml`

Una vez guardados los archivos en los directorios correspondientes de la máquina virtual ROS+STDR, el simulador se inicia mediante el siguiente comando en la consola de Linux de la máquina virtual.

```
roslaunch stdr_launchers parking_ackerman_bateria.launch
```

La ventana de simulación del vehículo en el entorno con obstáculos se muestra en la Figura 3.

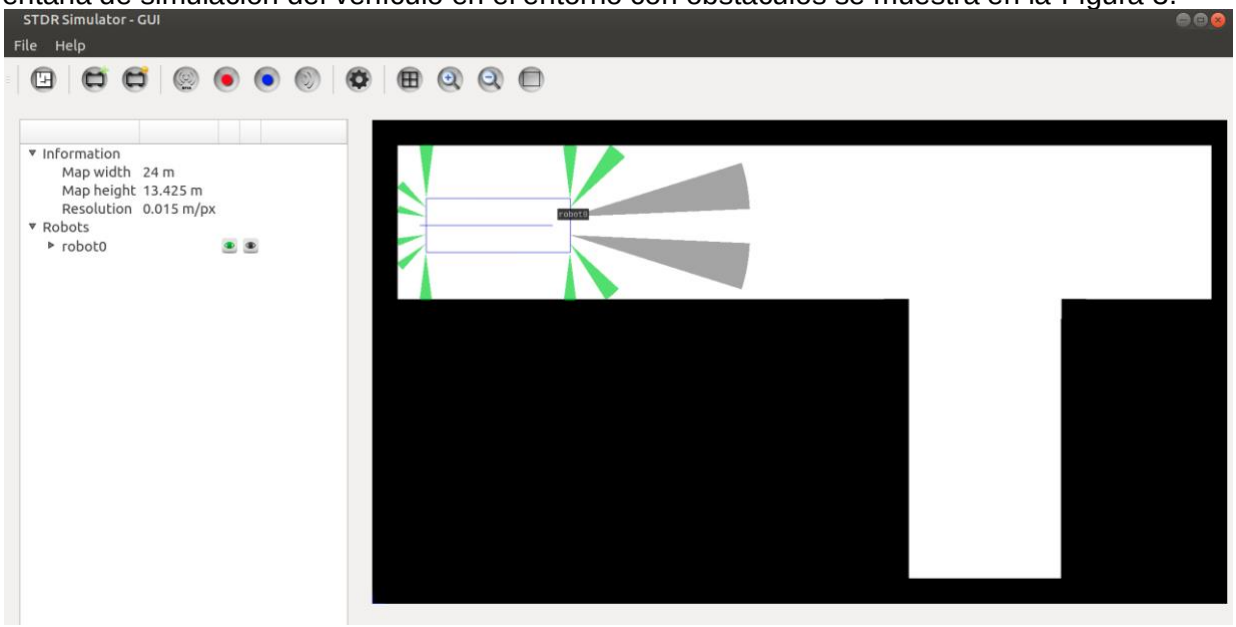



Figura 3. Entorno de simulación de la práctica.

La conexión con el vehículo se realiza a partir de las librerías de Matlab/ROS. Se incluye en la práctica el archivo *ackerman_ROS_controller.slx* en el que se incluye un bloque Simulink (ROS Robot) de conexión a ROS (véase Figura 4) para la comunicación y el control con el robot simulado en STDR. Este bloque entrega en sus salidas la posición (salidas x e y), orientación (salida theta) y las lecturas de los 12 sensores de ultrasonidos del vehículo (salida sonar).

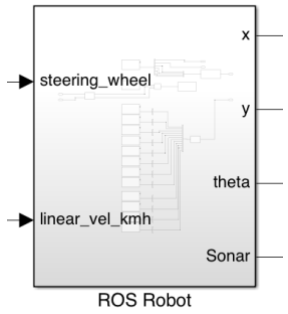



Figura 4. Bloque Simulink de acceso al robot y conexión a ROS.

Es necesario configurar correctamente la IP de la máquina virtual dentro del bloque del robot, haciendo doble click en cualquiera de los bloques de subscripción a ROS y haciendo click en ***"Configure network addresses"*** (véase Figura 5).

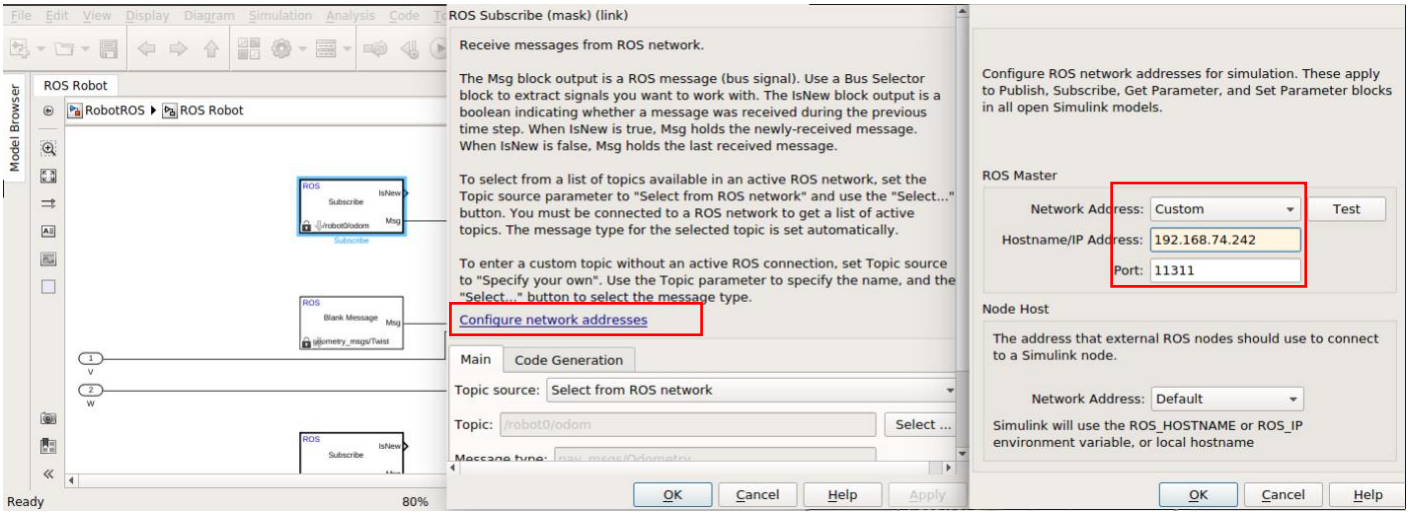



Figura 5. Configuración de la IP de la máquina virtual en el bloque ROS Robot.

# 4. Desarrollo del trabajo

El desarrollo de la práctica consta de dos partes que se describen a continuación.

## *Parte 1. Diseño manual de un control borroso de tipo MAMDANI.*

En esta parte de la práctica se diseña un controlador de la velocidad lineal y giro del volante del vehículo en el entorno para realizar la maniobra de aparcamiento de la mejor forma y en el menor tiempo posible a partir de la información obtenida de los sensores de ultrasonidos del robo. En esta parte, se dispone de libertad para elegir el número de entradas del controlador, los sensores utilizados, rangos de entrada y salida del controlador, funciones de pertenencia, y reglas del controlador, de manera que el vehículo sea capaz de cumplir con la tarea planteada.

Para la realización de esta práctica se hará uso de la herramienta **fuzzy** de Matlab utilizada en prácticas anteriores. Se recomienda al alumno el diseño de controladores que hagan uso del mínimo número de sensores posible para realizar la tarea encomendada. Una vez diseñado el controlador, se creará un bloque de control en Simulink donde se indique el archivo *.fis* que contiene el controlador y se conectará al bloque del vehículo. En la Figura 6 se muestra un ejemplo de controlador borroso que usa el sensor sonar_0 para realizar el control del vehículo.

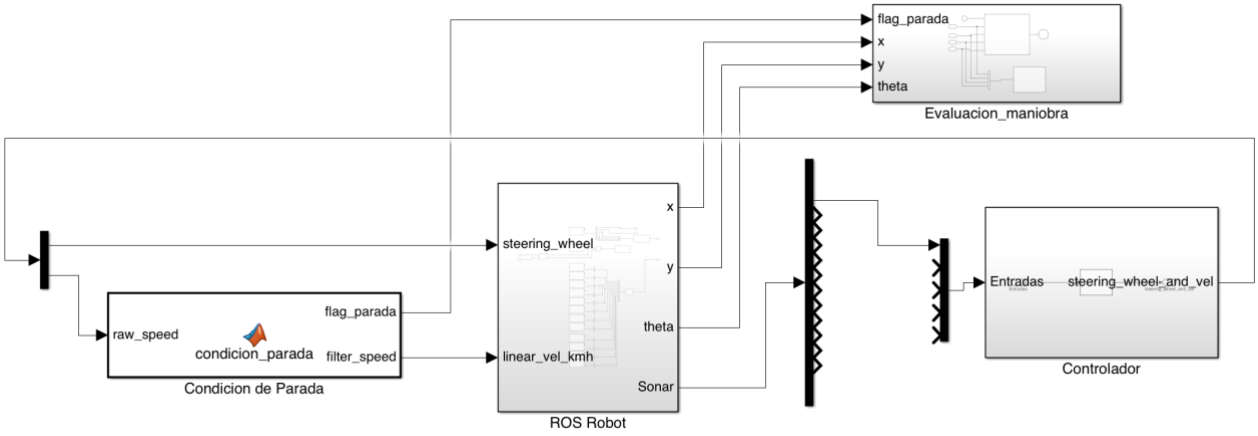



Figura 6. Ejemplo de control del vehículo (*archivo ackerman_ROS_controller.slx* incluido en la práctica)

**Condición de parada:**

Se debe decidir en base a la información disponible (velocidad lineal del vehículo, ángulo del volante e información de los sensores de ultrasonidos) una conducción de parada. Esta condición de parada debe realizar las siguientes acciones:

- Parar el vehículo: enviando a su salida una *velocidad_lineal = 0*.
- Indicar que se ha finalizado la maniobra, activando el flag *flag_parada=1.*

No se puede utilizar el bloque de Simulink “STOP” como en prácticas anteriores.

**Evaluación de la maniobra:**

Este subsistema (Evalucion_maniobra en la figura anterior) no debe ser modificado. Es el encargado de recolectar los datos de la maniobra y parar la simulación cuando le llega la señal *flag_parada=1:*

## Parte 2. Diseño automático de un controlador neuronal

Para la realización de esta parte se utilizará las herramientas de entrenamiento vistas en prácticas anteriores.

El proceso de generación de los controladores neuronales pasa por las siguientes etapas:

1. **Obtener los datos de entrenamiento:**
   Para obtener los datos de entrenamiento existen varias posibilidades:
   a. **Telecontrol**
   b. **Realizar un recorrido prefijado**
   c. **Usar un controlador borroso.**

a) **Telecontrol:** se dispone de una interfaz a través del teclado (script de Matlab `ControlManual_Ackerman_12_sensores_V_steering.m` incluido en los archivos de la práctica) con la que el alumno puede controlar manualmente el vehículo para realizar la maniobra y en el que se registrará la velocidad lineal, ángulo del volante, posición, orientación y lecturas de los sensores.

Este script necesita además los siguientes scripts también disponibles en la práctica:

```
Key_Down_sensores_V_steering.m
function_conversion_steering_to_linear_angular.m
```

Para la ejecución del control manual se deberán configurar correctamente:

- La dirección IP de la máquina virtual (**`ROS_MASTER_IP`**)
  - **`ROS_MASTER_IP =`** `'172.29.29.70'`
- Los incrementos de velocidad lineal (km/h) y ángulo del volante (grados) que queremos modificar con cada pulsación de una tecla:
  - incAngular = 5;
  - incLineal = 2;

Una vez iniciado el simulador STDR, se ejecuta el script `ControlManual_Ackerman_12_sensores_V_steering.m` desde la consola de Matlab. Para el control manual de las velocidades del vehículo se utilizan:

- Cada pulsación de las teclas DERECHA e IZQUIERDA incrementa o decrementa respectivamente el ángulo del volante, con incrementos determinados por la variable *incAngular.*
- Cada pulsación de las teclas ARRIBA ABAJO incrementa o decrementa respectivamente la velocidad lineal del vehículo en un valor determinado por la variable *incLineal.*
- Se deberá tener el foco de la ventana de la figura de Matlab que aparece al ejecutar el script activado para poder telecontrolar el vehículo.
- Mediante la pulsación de la BARRA ESPACIADORA se dará por finalizado el control manual (se recomienda grabar los datos de varios aparcamientos exitosos).

Se grabará cada 0.1 segundos la siguiente información en las filas de la matriz `training_data`:
`training_data(i,:)=[sonar_0, sonar_1, … , sonar_7, x, y , theta, ang_volante, vel_lineal, steering_wheel_angle, linear_vel_ackerman_kmh]`

donde,

- sonar_0, …, sonar_11 son las medidas instantáneas de los sensores de ultrasonidos
- x, y es la posición instantánea del vehículo.

- theta su orientación
- vel_angular y vel_lineal son los valores de velocidad angular y lineal del vehículo.
- steering_wheel_angle, linear_vel_ackerman_kmh son los valores de Angulo del volante y velocidad lineal del vehículo en km/h
- El número de filas de `training_data` dependerá del tiempo empleado en el control manual del vehículo.

b) **Realizar un recorrido prefijado:** se dispone de un script de Matlab `maniobra_park_ackerman_datos_entrenamiento_alumnos.m` (incluido en los archivos de la práctica) con la que el alumno puede programar un recorrido en base a diferentes avances y giros, modificando las siguientes líneas y añadiendo tantos avances/giros como considere:

```
distancia=3.5
vel_lineal_ackerman_kmh = -3      %(km/h)
steering_wheel_angle =  20        % desde -90 a 90 grados.
avanzar_ackerman
```

Este script necesita además los siguientes scripts también disponibles en la práctica:

`function_conversion_steering_to_linear_angular.m`
`avanzar_ackerman.m`

c) **Usar un controlador borroso:** se podrá usar un controlador borroso para obtener los datos. Puede ser el de la primera parte de esta práctica u otro.

2. **Preparar los datos de entrenamiento/training**. Estos son los datos generados previamente mediante el script `ControlManual_Ackerman_12_sensores.m` explicado anteriormente. Para diseñar cada uno de los controladores, será necesario crear una variable a partir de `training_data` con los datos que se necesiten.

   Solo está permitido usar como datos de entrenamiento:

   - Los sensores de ultrasonidos (sonar_0, …, sonar_11*)*,
   - Velocidad lineal (km/h) y el ángulo del volante (grados): steering_wheel_angle, linear_vel_ackerman_kmh

   Por ejemplo, si se quiere diseñar el controlador que genere el ángulo del volante a partir de los sensores de ultrasonidos 0 y 5, habrá que crear una variable nueva inputs que tome las columnas 1 (sonar _0) y 6 (sonar_5) y otra outputs con la columna 18 (ang_volante) de la variable `training_data`

```
inputs  = training_data( (:,[1,6])';
outputs = training_data( (:,[18])';
inputs(isinf(inputs)) = 5.0;
inputs = double(inputs);
outputs = double(outputs);
```

**3.** **Generar la red** con la configuración deseada

```
net = feedforwardnet([neuronas_capa1, neuronas_capa2,… ]);
```

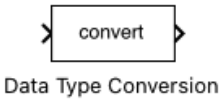
4. **Entrenar la red**

```
net = configure(net,inputs,outputs);
net = train(net,inputs,outputs);
```

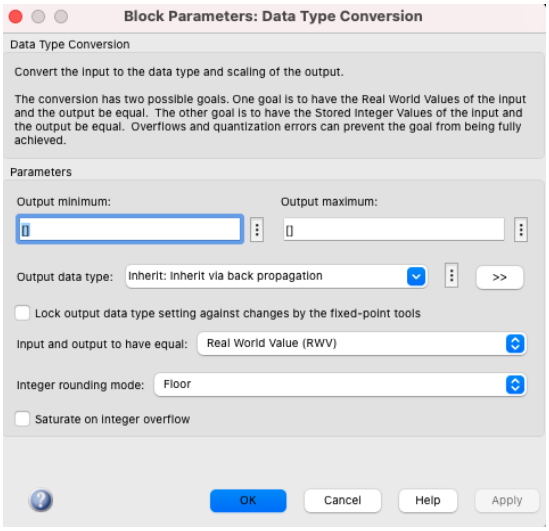
5. **Generar el bloque de Simulink** con la red entrenada para probar su funcionamiento en el simulador.

```
gensim(net,Ts)
```

6. **Modificar el diagrama de Simulink de la Figura 6 para incluir el nuevo controlador neuronal:**
   a. Cambiar el controlador por el generado en el paso anterior
   b. Añadir bloque "*Data Type Conversion*" entre la salida del multiplexor de los ultrasonidos y la entrada de la red:




Data Type Conversion



# 5. Evaluación de la práctica

La evaluación de la práctica se realizará sobre el simulador STDR, ejecutando los controladores desarrollados sobre dos entornos diferentes:

- ***Entorno original,*** es el mostrado en la Figura 1 con la posición inicial fija.
- ***Entorno definitivo***, será similar al mostrado en la Figura 1, pudiendo variar la posición inicial del vehículo en 0.5m en ambos ejes. Esta posición se publicará unos días antes de la entrega del trabajo.

Cada pareja de laboratorio deberá entregar una memoria y grabar un video (o varios si fuera necesario) mostrando el funcionamiento de su propuesta y haciendo una breve exposición de los métodos empleados con una duración máxima de 10 minutos (entre todos los vídeos), de los cuales:

- 5 minutos como máximo para la explicación del trabajo realizado.
- 5 minutos para mostrar la maniobra del vehículo en el mapa (mostrando la posición y orientación final del vehículo).

Esta exposición, lo que sería una presentación oral en clase, conviene que se apoye en transparencias. En este vídeo deben participar activamente los 2 miembros del grupo.

Se indican a continuación los porcentajes máximos de la nota final para cada apartado de la práctica. Para obtener la máxima nota en cada apartado se evalúa la calidad objetiva de los diseños realizados, así como la presentación de los resultados.

- **Desarrollo práctico de la práctica (80%)**
  - Parte 1. (50%)
    - Diseño de un control borroso capaz de realizar la maniobra de aparcamiento en el entorno original (25%).
    - Diseño de un control borroso capaz de realizar la maniobra de aparcamiento en el entorno definitivo (25%)
  - Parte 2. (30%)
    - Diseño de un control neural capaz de realizar la maniobra de aparcamiento en el entorno original (15%)
    - Diseño de un control neural capaz de realizar la maniobra de aparcamiento en el entorno definitivo (15%)
- **Memoria y presentación de la práctica final (20%).**
  - Correcta exposición de los algoritmos utilizados y los resultados obtenidos tanto en la memoria como en el vídeo de presentación.
- **Competición (1 punto adicional)**
  - Cada grupo propondrá un controlador de los diseñados en la práctica final para de realizar la maniobra de aparcamiento en el entorno definitivo y se medirá el tiempo empleado por el controlador en realizar la maniobra, así como lo cercano a la posición ideal de aparcamiento.
  - Con el objetivo de poder decidir qué grupo o grupos obtienen este punto extra, aquellos alumnos que quieran optar a esta recompensa deben grabar otro vídeo donde se muestre el recorrido de su vehículo con el mejor controlador diseñado ***sin cortes ni edición y mostrando claramente el cronómetro visible en el simulador STDR***.