

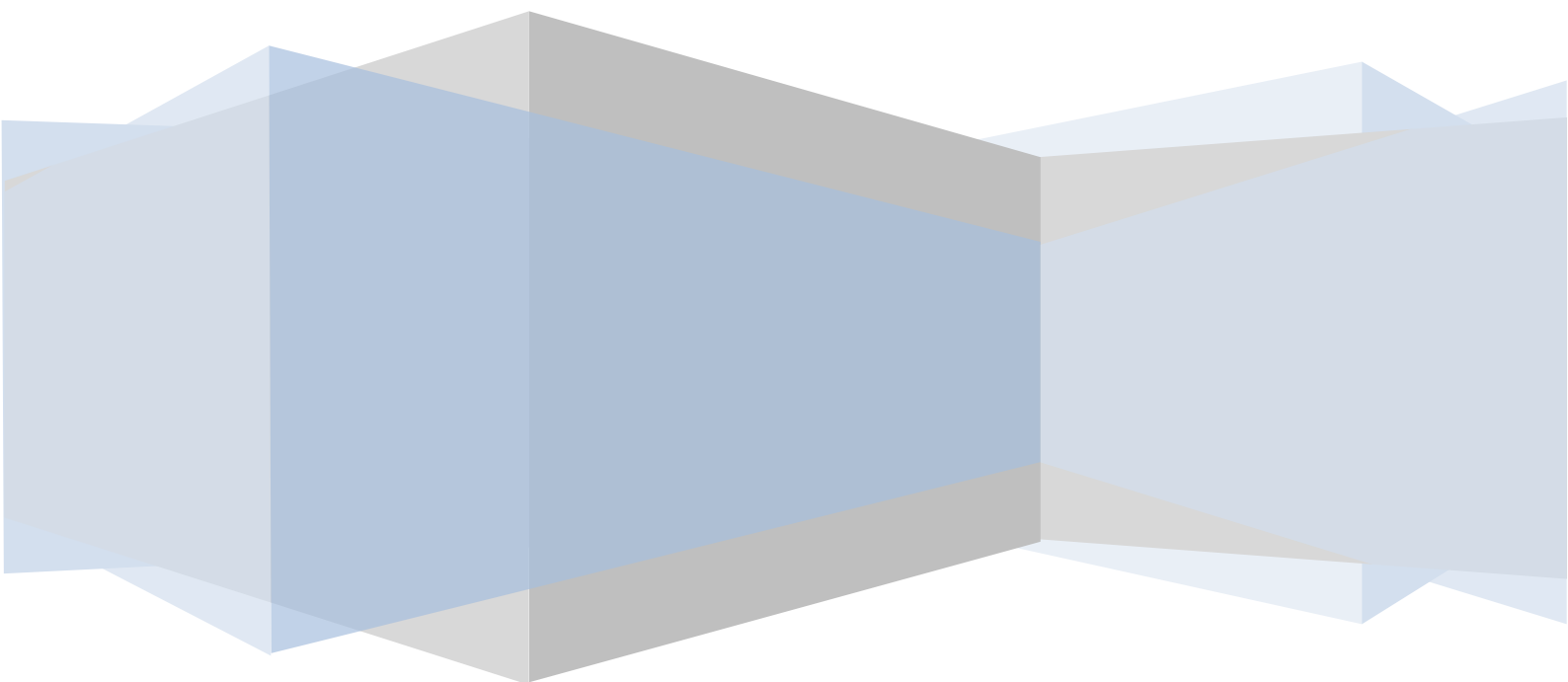
ROBÓTICA
CUARTO CURSO DEL GRADO EN
INGENIERÍA INFORMÁTICA



Universidad
de Huelva

GUÍA DE EJERCICIOS 8

Introducción al Control reactivo

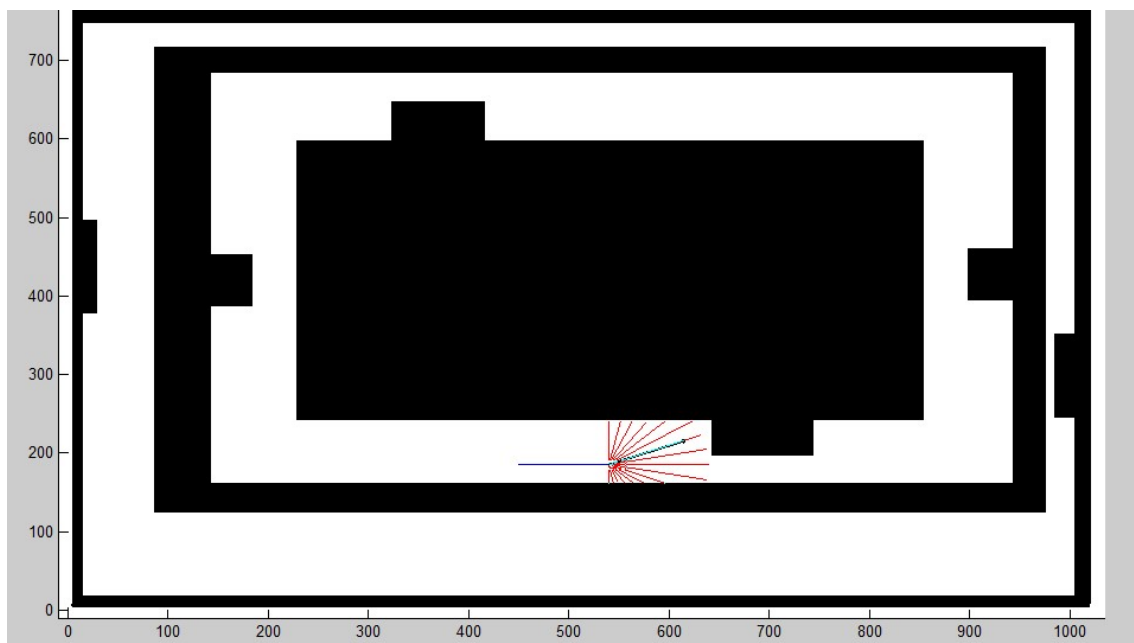


Introducción

En esta Guía se pretende trabajar con un ejemplo simulado en el que las medidas proporcionadas por un sensor de proximidad se fusionan para determinar la percepción que el robot tiene de su entorno. Posteriormente, dicha percepción se incorporará al bucle de control para conformar una estrategia de control reactivo.

Escenario de simulación

La simulación arranca ejecutando el script 'laser_reactivo.m'. Es necesario que en el directorio esté presente el fichero 'kuta.p', que como ya sabéis se trata de un fichero compilado que no puede ser modificado. Al correr la simulación, aparece una imagen como la siguiente:



En esta imagen podemos observar que se representa un vehículo con la cinemática de un automóvil (Ackerman) y mediante haces rojos la respuesta de un sensor LIDAR situado en la parte trasera del robot. Cada haz representa una de las direcciones en las que el sensor realiza las medidas. Cada haz tiene una longitud que representa la distancia desde el sensor hasta las paredes del entorno, medida en la dirección correspondiente. Igualmente, en la simulación también se representa con una línea azul la trayectoria del robot y un vector de color cian que será de utilidad para resolver las cuestiones planteadas en esta guía.

Control de la simulación

El tiempo final virtual que dura la simulación se determina con la variable `tf`. El movimiento del robot se controla mediante las variables `velocidad` y `volante`. En el código inicial tienen configurado un valor fijo. Variando el valor de `velocidad` el robot se mueve más rápido o más lento. Cambiando el valor de `volante` el vehículo se mueve hacia una dirección u otra. Los valores de `volante` están limitados entre $-\pi/4$ y $\pi/4$. La función `kuta()` es la encargada de realizar la integración numérica, realizar la representación gráfica y proporcionar las medidas del sensor LIDAR. Estas medidas se guardan en la variable `laser` que es un array de 17×2 en la forma:

$$\begin{bmatrix} d_1 & \theta_1 \\ d_2 & \theta_2 \\ \vdots & \vdots \\ d_n & \theta_n \end{bmatrix}$$

Siendo d_i cada una de las distancias captadas por los sensores y θ_i el ángulo en el que se realiza la medida.

La representación del vector color cian se controla mediante las variables `componentes_x` `componentes_y` que representan las componentes de dicho vector en el sistema de referencia local del robot.

En la llamada a la función `kuta()` se introducen como argumentos los valores de las variables de control y devuelve los valores del sensor LIDAR y de la configuración del robot en el instante $k+1$.

EJERCICIOS

- 1) Supervise el código, ejecútelo y comprenda su funcionamiento teniendo en cuenta todo lo descrito en los apartados anteriores.
- 2) Modifique los valores de las variables de control para que el vehículo se desplace en línea recta. Modifique igualmente los valores para cambiar la orientación y el módulo del vector cian.
- 3) Considere los haces del sensor LIDAR como vectores y realice la suma de los mismos en el sistema de referencia local del robot. Este vector será el vector de percepción que apuntará hacia la dirección con más espacio libre.
- 4) Haga que el vector cian represente al vector de percepción e identifique como varía el módulo y la dirección del vector según cambia el entorno del robot al moverse.
- 5) Diseñe una estrategia de control reactivo, donde la velocidad del robot dependa de la proximidad a un obstáculo y el ángulo del volante sea proporcional a la dirección del vector de percepción en el sistema de referencia local del robot. El robot deberá navegar alrededor del circuito evitando la colisión con el escenario y los obstáculos.
- 6) Cambie el fichero correspondiente al mapa por otro original generado libremente. Utilizando lo aprendido en las guías anteriores, genere una trayectoria que permita navegar al robot desde un punto a otro. Guarde la trayectoria en un fichero .txt. Modifique el mapa para incorporar uno o varios obstáculos nuevos que intercepten la trayectoria inicialmente planificada. Combine la estrategia de seguimiento, junto con el controlador reactivo desarrollado anteriormente para conseguir que el robot siga la trayectoria planificada y además esquivе los obstáculos no conocidos inicialmente.