

Robótica 2021-2022

1st Andrés Ventura Valiente
aventurav@alumnos.unex.es

2nd Carlos Sánchez García
csancheznf@alumnos.unex.es

3rd Yeray Bravo Díaz
ybravodi@alumnos.unex.es

I. HERRAMIENTAS

En primer lugar, antes de comenzar a realizar los cambios necesarios para que la segunda práctica fuese correcta tuvimos que instalar el simulador de CoppeliaSim y la biblioteca PyRep para interactuar con él desde un script de Python. A parte de instalar estas dos herramientas, también hemos tenido que añadir algunos componentes. Principalmente el componente para representar el espacio de trabajo, para visualizar el espacio recorrido, para el correcto funcionamiento y detección del láser, para poder controlarlo con hardware externo y para controlar la velocidad de movimiento.

II. OBJETIVOS DE LA ENTREGA

Para esta entrega se precisa de que nuestro robot de limpieza sea capaz de que el robot avance y sea capaz de poder detectar obstáculos para así evitar colisiones, y el principal objetivo es que barra el mayor espacio posible en el menor tiempo posible. Para la obtención de esta práctica se han seguido unos pasos:

A. Creación del componente Controlador

Hemos tenido que crear el componente "Controlador" con el editor robocompds. Este componente es que condece al robot entre los obstáculos evitando colisiones. En el módulo compute de la clase SpecificWorker.cpp es donde hemos realizado cambios para que sea posible el objetivo. Tenemos una constante "threshold" declarada para que el robot se pare a la distancia especificada, que en nuestro caso es de 365 milímetros(const float threshold = 365). Una variable "rot" para la rotación del robot, que en nuestro caso es de 0.8 radianes por segundo(float rot = 0.8). Y otra variable "adv" para la velocidad de avance del robot, que en nuestro caso es de 220 milímetros por segundo(int adv = 220). Estas dos variables han sido declaradas previamente en el .h. Para el correcto funcionamiento hemos diferenciado dos casos, si el robot se choca o no, que lo explicamos a continuación.

B. Si el robot choca

Para poder controlar si el robot choca, tenemos en cuenta en todo momento los datos que obtiene el láser del propio robot(RoboCompLaser::TLaserData ldata = laserproxy-getLaserData()). Para este caso, hacemos la comprobación de si nuestra primera posición válida del vector de datos del láser "ldata" es menor a nuestra constante de "threshold". Si es menor, lo que hacemos es que a través de una estructura switch, que en el primer caso el robot retroceda(differentialrobotproxy-setSpeedBase(-120, 0)) y que después rote(differentialrobotproxy-setSpeedBase(0, rot)), si

no le es posible rotar a causa de una colisión, que entre en el siguiente caso del switch, y que haga exactamente lo mismo pero rotando en sentido contrario(differentialrobotproxy-setSpeedBase(0, -rot)). En ambos casos hacemos que el robot retroceda durante 2 segundos.

C. Si el robot no choca

En este caso tenemos controlado que la primera posición del vector de datos del láser sea mayor que nuestra constante "threshold". Como el objetivo de la entrega es que recorra el mayor espacio posible en el menor tiempo posible, el primer movimiento que hace el robot es una espiral(differentialrobotproxy-setSpeedBase(adv,0.8);adv=adv+16)) hasta que choque o hasta que alcance la velocidad de 1000(hemos decidido hacer una espiral ya que si hay un gran espacio libre, puede ser una buena opción para que barra el mayor espacio posible de ese hueco), el segundo y tercer movimiento es el mismo con la idea de evitar que se quede en alguna esquina o perdiendo eficacia entre objetos(differentialrobotproxy-setSpeedBase(800, 0)). Y el cuarto movimiento es que avance con una pequeña rotación para evitar siempre el avance en línea recta, y así puede avanzar con una diferente inclinación(differentialrobotproxy-setSpeedBase(800, -0.11)).

III. PRUEBAS

Tras el correcto funcionamiento del robot, hemos procedido a realizar varias pruebas. Hemos concluido tres pruebas de 5 minutos cada una. En la primera prueba hemos obtenido un recorrido de un 26.96. En la segunda prueba un recorrido de 27.2. Y en la tercera prueba un recorrido de 27.68. En el que de media hemos tenido un recorrido de 27.28