Sistemas Robóticos Autónomos

Trabajo de curso: Competición

Aitana Carreño Cabeza
Raúl Cruz Ortega
Laura González Suárez
Santiago Emilio Flores Reina

Foto grupal y nombre del robot

Nombre del robot: **Juan**

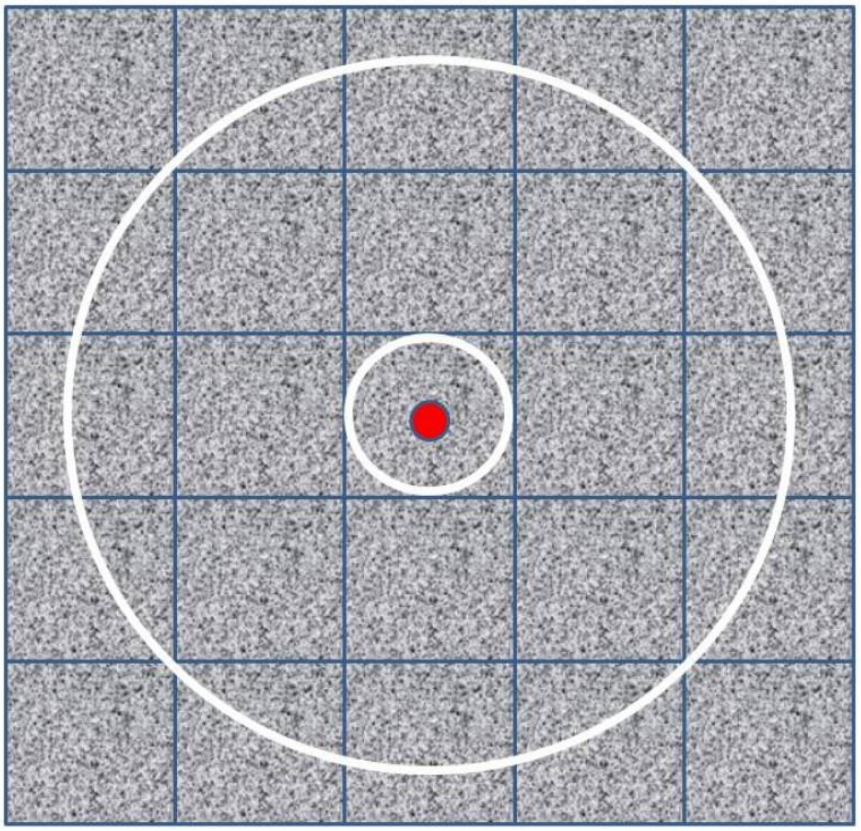
Introducción

El presente informe detalla el diseño, desarrollo y evaluación de un robot destinado a participar en una competición específica, en la cual será necesario el uso de sensores y corrección de errores.

El objetivo principal de este proyecto es que el robot, partiendo desde fuera de un círculo externo, a una distancia de aproximadamente un tercio de baldosa del mismo, con una desviación máxima de 45 grados con respecto a la posición de la lata ubicada en el centro del escenario, golpee la parte superior de ésta. Con el fin de indicar que el golpeo ha sido exitoso, el robot debe emitir un sonido y cambiar el color de los leds. En caso de que no se logre impactar la lata, el robot debe permanecer inmóvil durante 10 segundos antes de volver a intentar golpear la lata.

Una vez golpeada la lata con éxito, el robot deberá salir del círculo exterior emitiendo señales visuales y auditivas que indiquen la finalización de la tarea. En este punto, se detendrá el cronómetro, y el tiempo transcurrido se registrará para su evaluación.

Para lograr realizar este circuito con éxito, se deben utilizar distintos sensores, así como aplicar los elementos vistos en clase como el manejo de errores. Estos aspectos se tomarán en cuenta para la evaluación, no sólo el tiempo en el que el robot completa la actividad propuesta.



1. Algoritmo utilizado

El algoritmo utilizado para el desafío es el siguiente:

1. El robot avanza hasta detectar la línea blanca.
2. A partir de ahí, avanza 2 vueltas de rueda completas. Para este desafío, se han usado unas ruedas con las siguientes medidas:
   - Diámetro: 5,55 cm
   - Ancho: 2,8

   Por lo que 2 vueltas de ruedas significan que el robot avanza 34,87 cm y luego se detiene.
3. Ahí comienza a girar 90º (grados) hacia la izquierda mientras escanea con el sensor ultrasonido.
4. Luego gira 90º hacia la derecha para volver a la posición antes del giro.
5. Gira 90º hacia la derecha mientras escanea con el sensor ultrasonido.
6. Posteriormente, gira 90º hacia la izquierda para volver a la posición anterior.
7. Luego, gira hacia el objeto más cercano que haya captado con el sensor ultrasonido durante los pasos 3 y 5.
8. Avanza hacia adelante hasta detectar la línea blanca y se detiene.
9. Realiza un escaneo similar al que realizó en los pasos 3-7, con la diferencia de que el ángulo de giro es menor, concretamente realiza un escaneo de 35º en ambas direcciones. Tras realizar el escaneo el robot se coloca en dirección al punto que menor distancia ha registrado durante el escaneo.
10. Avanza hasta que la distancia con respecto a la lata sea menor o igual a 8,5 cm, y en ese momento baja el brazo del robot para golpearla. Si durante el avance a la lata, deja de detectarla con el sensor de ultrasonido durante un periodo de aproximadamente un segundo, retrocede hasta detectar la línea blanca y realiza el paso 9 de nuevo.
11. Si falla el golpe, espera 10 segundos, retrocede hasta la línea blanca y vuelve al paso 9.
12. Si lo golpea, retrocede hasta salir del círculo exterior e informa de su finalización emitiendo un sonido y encendiendo los leds en color verde.

2. Uso de sensores

Se ha hecho uso de cuatro sensores para llevar a cabo correctamente la tarea:

Sensor de color: Este sensor detecta el color de la superficie a la que apunta. Es utilizado en el algoritmo para detectar el paso del robot por las líneas blancas.

Sensor de giro: Este sensor mide el ángulo en grados de cuánto ha girado el robot. Se usa durante los escaneos para hacer giros precisos y guardar el ángulo de giro en el que detecta el objeto más cercano.

Sensor de ultrasonido: Es un dispositivo que emite ondas ultrasónicas para medir la distancia entre el sensor y un objeto, lo hace midiendo el tiempo en el que dichas ondas vuelven al sensor. Se usa en el desafío para localizar la lata y determinar la distancia entre el robot y la lata.

Sensor de toque: Es un sensor que detecta el contacto físico con un objeto. En el trabajo, se utiliza para garantizar que se ha golpeado la lata correctamente.

3. Configuración del robot

El robot está equipado con tres motores: uno en cada rueda y otro en su brazo. Este brazo cuenta con un sensor de toque encargado de identificar el contacto con la lata. Además, el brazo está alineado con el sensor de ultrasonido, el cual es el encargado de detectar la posición de la lata. Por otra parte, el sensor de color está ubicado cerca del suelo para la correcta detección de las líneas blancas. Por último, el sensor de giro está ubicado en la parte superior.

4. Manejo de errores

Durante la ejecución del proyecto, nos hemos encontrado con ciertos errores los cuales hemos tenido que manejar para el correcto funcionamiento del robot. Los errores encontrados son los siguientes:

- Cuando el robot se aproximaba a la lata en la parte del círculo interno, se observó en diversas ocasiones que el sensor perdía momentáneamente la detección de la lata. Esto nos llevó a abordar este problema de la siguiente manera: si, durante el desplazamiento del robot hacia la lata en el círculo interior, la lata dejaba de ser detectada por el sensor de ultrasonido durante 0,8 segundos, el robot retrocedía hasta identificar la línea blanca y luego reiniciaba el escaneo en busca de la lata.

- Durante la fase de golpeo a la lata, el robot experimentaba en ocasiones fallos o quedaba atascado con la lata, impidiendo así la finalización de la acción. Para abordar este inconveniente, implementamos la siguiente solución: si, durante el descenso del sensor de toque, no se detecta el contacto con la lata en un lapso de 0.50 segundos, el sensor de toque regresa a su posición inicial, el robot retrocede hasta la línea blanca y posteriormente reinicia el escaneo en búsqueda de la lata.

5. Resultados

Tiempo (s) orientación: 45º hacia la izquierda	**Tiempo (s) orientación: Recto**	**Tiempo (s) orientación: 45º hacia la derecha**
23,28	21,28	29,38
35,40	21,14	29,33
34,08	21,07	28,65
47,50	21,19	21,19
22,94	28,19	23,73

Se han registrado mediciones de tiempo que representan el lapso que el robot emplea para completar la tarea. Los valores en rojo señalan los casos en los que el robot falló una vez al intentar golpear la lata. El tiempo más breve registrado fue de 21,07 segundos, logrado cuando el robot se orientó directamente hacia la lata. Por otra parte, el peor desempeño fue el tiempo de 47,50 segundos, cuando el robot se orientó 45º hacia la izquierda con respecto a la posición de la lata.

6. Código

Main:

```
#!/usr/bin/env python3

from getUltrasonicDistance import * from ev3dev2.led import Leds from
ev3dev2.sound import Sound from ev3dev2.sensor.lego import TouchSensor,
ColorSensor, GyroSensor from ev3dev2.motor import LargeMotor,
OUTPUT_B, OUTPUT_C, OUTPUT_D from time import sleep

leds = Leds() touch =
TouchSensor() sensor_color =
ColorSensor() sound = Sound()
gyro_sensor = GyroSensor() grua =
LargeMotor(OUTPUT_D) motor_i =
LargeMotor(OUTPUT_B) motor_d =
LargeMotor(OUTPUT_C)
leds.all_off()

def warning(color: str):
    leds.set_color("LEFT", color)
    leds.set_color("RIGHT", color)
    sound.play_tone(1500,1,0)
    leds.all_off()

def move_until_white(velocity: int, time_shutdown: float = 0):
    motor_i.on(velocity)
    motor_d.on(velocity)
    while True:
        if (sensor_color.color ==
            6):
            sleep(time_shutdown)
            motor_i.off()
            motor_d.off() break
def detectCan(angulo_maximo: int):
    distancia = getUltrasonicDistance()
    angulo = gyro_sensor.angle
    angulo_inicial = gyro_sensor.angle
    print(gyro_sensor.angle)
```

```
motor_i.on(-10)
motor_d.on(10)
while True:
    dist_actual = getUltrasonicDistance() print(dist_actual) if
    (abs(gyro_sensor.angle - angulo_inicial) >= angulo_maximo):
        motor_i.off()
        motor_d.off()
        break
    if dist_actual < distancia:
        print("Lo guarda 1") angulo
        = gyro_sensor.angle
        print(angulo) distancia =
        dist_actual

motor_i.on(10)
motor_d.on(-10)
while True:
    if gyro_sensor.angle >= angulo_inicial:
        motor_i.off()
        motor_d.off()
        break

motor_i.on(10) motor_d.on(-10)
angulo_inicial =
gyro_sensor.angle while True:
    dist_actual = getUltrasonicDistance() print(dist_actual) if
    (abs(gyro_sensor.angle - angulo_inicial) >= angulo_maximo):
        motor_i.off()
        motor_d.off()
        break
    if dist_actual < distancia:
        print("Lo guarda 2") angulo
        = gyro_sensor.angle
        print(angulo) distancia =
        dist_actual motor_i.on(-10)
        motor_d.on(10) while True:
    if gyro_sensor.angle <= angulo_inicial:
        motor_i.off()
        motor_d.off()
        break
```

```
    print(angulo - angulo_inicial)
    giro_recorrer = angulo -
    angulo_inicial angulo_tras_giro =
    gyro_sensor.angle if (giro_recorrer >
    0): motor_i.on(10) motor_d.on(-10)
    else:
        motor_i.on(-10)
        motor_d.on(10)
    while True:
        if (abs(gyro_sensor.angle - angulo_tras_giro) >= abs(giro_recorrer)):
            motor_i.off()
            motor_d.off()
            break


def goToCan():
    motor_i.on(20)
    motor_d.on(20)
    count = 0
    anterior = getUltrasonicDistance()
    while True:
        print(getUltrasonicDistance()) if
        getUltrasonicDistance() < 8.5:
            motor_i.off()
            motor_d.off()
            break
        if getUltrasonicDistance() > 40:
            count += 1
        else:
            count = 0
        if count == 40:
            motor_i.off()
            motor_d.off()
            move_until_white(-20)
            detectCan(40)
            goToCan() break

def hitCan(): position_grua =
    grua.position grua.on(35)
    count = 0 while True:
```

```
        if touch.is_pressed:
            grua.off()
            warning("GREEN") position_actual = grua.position recorrer =
            abs(position_actual - position_grua) grua.on_for_degrees(speed=40,
            degrees=-recorrer, brake=True, block=True) finish() break
        if count == 2: grua.off() position_actual = grua.position recorrer =
            abs(position_actual - position_grua) grua.on_for_degrees(speed=40,
            degrees=-recorrer, brake=True, block=True) sleep(10) tryAgain() break
        sleep(0.25)
        count += 1


def tryAgain():
    move_until_white(-10)
    detectCan(30)
    goToCan() hitCan()

def finish():
    motor_i.on_for_degrees(speed=60, degrees=-550, brake=False,
    block=False) motor_d.on_for_degrees(speed=60, degrees=-550,
    brake=False, block=True) move_until_white(-60, 0.25) warning("GREEN")
if __name__ == '__main__': warning("ORANGE") move_until_white(40)
    motor_i.on_for_degrees(speed=40, degrees=720, brake=True, block=False)
    motor_d.on_for_degrees(speed=40, degrees=720, brake=True, block=True)
    detectCan(90) move_until_white(40) sleep(0.5) detectCan(35) goToCan()
    hitCan()
```

getUltrasonicDistance:

```
#!/usr/bin/env python3 from

ev3dev2.sensor.lego import *

ultr_sonic = UltrasonicSensor()

def getUltrasonicDistance():
    dist = ultr_sonic.distance_centimeters
    return dist
```