



PRÁCTICA 3

Introducción al iRobot Framework Lectura y calibración de los sensores del iRobot

OBJETIVOS

- Aprender a usar el *iRobot Framework*¹
- Programar la lectura de los sensores de iRobot Create y su calibración en lenguaje C/C++.

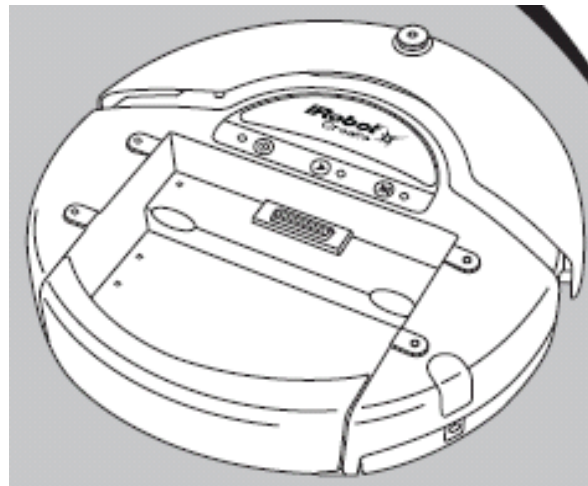
MATERIAL NECESARIO

Hardware

- iRobot Create
- Raspberry Pi2

Software

- NetBeans
- iRobot Framework



Manuales (disponibles en eGela)

- Guía del usuario: iRobot Create Owner's Guide
- Manual del lenguaje de comandos: iRobot Create Open Interface
- Lista de funciones iRobotConnection

1. Puesta en marcha

Para instalar y verificar el iRobot_Framework seguir el procedimiento detallado en **Instalacion iRobot_Framework.pdf** disponible en eGela.

2. Programación de sensores del iRobot Create

La práctica tiene por objetivo probar y calibrar los distintos sensores que existen en el robot iCreate. Para ello la lista de funciones de iRobotConnection dispone de la instrucción “**int iRobotConnection::updateSensor (char sensorId)**”. Los códigos **sensorId** se encuentran en el fichero **iRobotInstructionSet.h** dentro del namespace **iRobotSensors**. Si por ejemplo,

¹ La versión inicial del iRobot Framework fue creada por Gorka Montero como proyecto fin de carrera en 2012. En 2015 fue modificada por Borja Gamecho para funcionar sobre Raspberry Pi2.

queremos consultar el modo de funcionamiento en el que se encuentra iCreate, usaremos la siguientes instrucciones:

```
int modo = robot.updateSensor(iRobotSensors::OIMODE);  
cout << modo << endl;
```

Este código nos devolverá los siguientes valores:

- 1 si estamos en el modo Passive
- 2 si estamos en el modo Safe
- 3 si estamos en el modo Full

Tenemos que tener en cuenta que el método **updateSensor** siempre devuelve un **int** (entero con signo de 16 bits) donde se guarda el valor que recibe del robot iCreate (a través de Open Interface), dónde puede tener otro formato. Por ejemplo, si se leen los sensores *Bumps & WheelDrops* hay que interpretar bit a bit, mediante una máscara de bits, el entero de 16 bits que devuelve el framework.

Sensor	Codificación OI iRobot
Bumps & WheelDrops	Binaria (usar máscara de bits)
Wall, Cliffs x 4, Virtual Wall	1 bit value (el menos significativo)
Infrared	1 Byte [0,255]
Distance, Angle, Requested Radius	Signed 16 bit value [-32768 - 32768]
Wall Signal, Cliffs Signal x 4	Unsigned 16 bit [0 – 4095]
Requested Velocity x3	Signed 16 bit value [-500, 500]

Hay más información sobre los sensores disponibles en las páginas 24 y 25 del manual Create_Open_Interface_v2 (disponible en eGela).

3. Práctica

Nota: En todos los programas se valorará positivamente la cantidad y calidad de la información que devuelva el programa sobre los valores de los sensores que va leyendo.

3.1 Verificación de sensores on/off

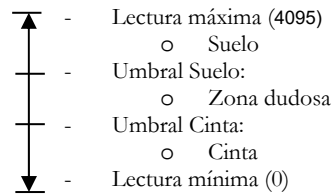
- Diseñar un programa que lea los valores de cada uno de los sensores siguientes en las situaciones de activado y desactivado. Deberá dar una señal acústica diferente para cada sensor cuando cambia de estado. Además, puede encender un LED cuando cualquiera de ellos está activado y apagarlo cuando todos estén desactivados.
- Este programa debe verificar el funcionamiento de los sensores:
 - Bumps (izquierdo y derecho)
 - Wheel drops (izquierdo, derecho y caster)
 - Cliff (derecho, frontal derecho, frontal izquierdo e izquierdo)
 - Botones Play y Advance

3.2 Calibración de sensores de barranco (Cliff Signal):

- Diseñar un programa capaz de recoger 5 muestras de cada sensor de barranco sobre

la cinta aislante negra y otros 5 sobre el suelo del laboratorio.

- Para cada sensor calcular los valores medios de lectura de cinta y suelo.
- Definir un umbral de suelo y un umbral de cinta que permitan diferenciar el suelo y la cinta aislante.



3.3 Calibración de sensores de distancia:

- Diseñar un programa que haga avanzar el robot en línea recta $(2+g/10)m$ [donde g es el número de grupo] usando el comando DRIVE DIRECT (iRobot Create Open Interface, pg. 9) y encuesta por distancia². Ejecutarlo 4 veces y medir las distancias recorridas realmente. Hacer una tabla con los valores medidos, su valor medio, el error absoluto y relativo del valor medio obtenido.
- Diseñar un programa que haga avanzar el robot en línea recta $(2+g/10)m$ usando el comando DRIVE³ y encuesta por distancia². Ejecutarlo 4 veces y medir las distancias recorridas realmente. Hacer una tabla con los valores medidos, los errores absolutos y relativos y calcular en error absoluto medio y el error relativo medio.
- Comparar los resultados obtenidos con ambos programas ¿Qué ventajas e inconvenientes presenta cada uno de ellos? ¿Cuál es más conveniente?

	Test 1	Test 2	Test 3	Test 4	Valor medio	Valor real ⁴	Error absoluto ⁵	Error relativo ⁶
DriveDirect								
Drive								

3.4 Calibración de sensores de giro:

- Diseñar un programa que haga girar al robot 180° a la derecha sin avanzar, usando DRIVE DIRECT y espera por ángulo⁷. Ejecutarlo 4 veces y medir los ángulos girados realmente. Hacer una tabla con los valores medidos, su media, los errores absoluto y relativo. Repetir el experimento girando 180° a la izquierda.
- Diseñar un programa que haga girar el robot 180° a la derecha sin avanzar usando DRIVE y espera por ángulo. Ejecutarlo 4 veces y medir el ángulo girado realmente⁸. Hacer una tabla con los valores medidos, los errores absolutos y relativos y calcular en error absoluto medio

² Cuando el iRobot Create avanza, la distancia medida se incrementa. Cuando el iRobot Create retrocede, la distancia medida se reduce. Si las ruedas giran pasivamente en cualquier dirección, la distancia medida se incrementa. Hasta que Create recorra la distancia especificada, su estado no cambia, ni reacciona ante ninguna entrada, en serie o de otro modo. *iRobot Create Open Interface* (pg. 16)

³ Un comando Drive con velocidad positiva y radio positivo hace que el robot avance mientras gira hacia la izquierda. Si el radio es negativo hace que Create gire hacia la derecha. Una velocidad negativa hace que el robot retroceda. Hay valores del radio especiales: 32768 (8000_{hex}) o 32767 (7FFF_{hex}) hacen que el robot avance recto. FFFF_{hex} hace que gire en su lugar en el sentido de las agujas del reloj. 0001_{hex} hace que gire en su lugar en el sentido contrario a las agujas del reloj. *iRobot Create Open Interface* (pg. 9)

⁴ Tomamos como valor real el medido con un metro.

⁵ $E_{\text{absoluto}} = |V_{\text{real}} - V_{\text{medido}}|$

⁶ $E_{\text{relativo}} = |V_{\text{real}} - V_{\text{medido}}| / V_{\text{real}} = E_{\text{absoluto}} / V_{\text{real}}$

⁷ Cuando el iRobot Create gira en el sentido contrario a las agujas del reloj, el ángulo se incrementa. Cuando iRobot Create gira en el sentido de las agujas del reloj, el ángulo disminuye. Hasta que el giro del robot no llegue al ángulo especificado, su estado no cambia, ni reacciona ante ninguna entrada, en serie o de otro modo. *iRobot Create Open Interface* (pg. 16)

⁸ Usar un transportador de ángulos para medirlo.

y el error relativo medio. Repetirlo girando 180° a la izquierda.

- Comparar los resultados obtenidos con ambos programas ¿Qué ventajas e inconvenientes presenta cada uno de ellos? ¿Cuál es más conveniente?

	Test 1	Test 2	Test 3	Test 4	Valor medio	Valor real ⁹	Error absoluto ⁵	Error relativo ⁶
DriveDirect Derecha								
Drive Derecha								
DriveDirect Izquierda								
Drive Izquierda								

3.5 Calibración del sensor de distancia a la pared (Wall Signal):

- Recoger 10 muestras del sensor Wall Signal a distintas distancias de la pared.
- Hacer una tabla con los resultados.
- Dibujar la recta de regresión y obtener su ecuación mediante el método de mínimos cuadrados.

4. Informe a presentar

1. Código bien estructurado y comentado de todos los programas

2. Verificación de sensores on/off

Para cada sensor decir si funciona bien o no, si tiene fallos esporádicos, si falla en determinadas condiciones, etc., de: Bumps (izquierdo y derecho), Wheel drops (izquierdo, derecho y caster), Cliff (derecho, frontal derecho, frontal izquierdo e izquierdo), Botones Play y Advance

3. Calibración de sensores de barranco (Cliff Signal):

- Tabla con los valores recogidos de cada sensor de barranco sobre la cinta aislante negra y sobre el suelo del laboratorio. Valores medios de lectura de cinta y suelo para cada sensor.
- Cálculo del umbral de suelo y del umbral de cinta para diferenciar con seguridad el suelo y la cinta aislante.

4. Calibración de sensores de distancia:

- Tabla con los valores medidos usando Drive Direct y Drive, las medias, los errores absolutos y relativos.
- Comparación de los resultados
 - ¿Qué ventajas e inconvenientes presenta cada uno de ellos?
 - ¿Cuál es más conveniente? ¿Por qué?

5. Calibración de sensores de giro:

- Tabla con los valores medidos usando Drive Direct y Drive con giro de 180° a la derecha y a la izquierda, las medias y los errores absolutos y relativos.
- Comparación de los resultados
 - ¿Qué ventajas e inconvenientes presenta cada uno de ellos?
 - ¿Cuál es más conveniente? ¿Por qué?

6. Calibración del sensor de distancia a la pared (Wall Signal):

- Tabla con los valores de Wall Signal medidos a distintas distancias de la pared.
- Ecuación y gráfica de la recta de regresión por el método de mínimos cuadrados.

⁹ Tomamos como valor real el medido con un transportador de ángulos. En el caso de 180° es el ángulo en línea recta en dirección contraria.