

Coco

Rodrigo Arias Mallo
rodrigo.arias@udc.es

1. Descripción

Coco es un robot diseñado para aprender los principios básicos de la robótica. Ha sido creado con la intención de reutilizar componentes de antiguos aparatos y darles un nuevo uso. Así como para comprender los fundamentos teóricos aplicados en un experimento práctico.

1.1. Sensores

Cuenta con varios sensores que le permiten obtener información del entorno, así como información sobre sí mismo. A cada sensor se le asigna un nombre corto, terminado en un número, que permite identificarlo. Los sensores son los siguientes:

1. **sonar0**: Sensor de ultrasonidos HC-SR04
2. **mouse0**: Ratón de bola
3. **ldr0**: Sensor de luz LDR ambiental
4. **ldr1**: Sensor de luz LDR dirigido hacia delante

1.2. Actuadores

El robot dispone de dos motores DC a ambos lados que le permiten moverse por el entorno. También puede emitir señales a través de sus leds. Incluye tres LEDS en la parte frontal, que conjuntamente con el sensor **ldr1**, le permiten apreciar el color del objeto que tiene delante. A continuación:

1. **mtr**: Motor derecho
2. **mtl**: Motor izquierdo
3. **ledr**: LED frontal rojo.
4. **ledg**: LED frontal verde.
5. **ledb**: LED frontal azul.
6. **led0**: LED de información.

2. Calibración de sensores

2.1. **sonar0**: Ultrasonidos HC-SR04

El sensor de ultrasonidos, permite averiguar de forma estimada la posición de un obstáculo. Para ello dispone de un emisor y un receptor de ultrasonidos orientados hacia delante, y posicionados en la parte frontal del robot.

Para calcular la distancia al objeto, se mide el tiempo que tarda un pulso ultrasónico en viajar desde el emisor al objeto, y tras rebotar, volver al receptor.

Ya que el sonido ha de rebotar en el objeto, la superficie del mismo influirá en gran medida en la información captada por el sensor. Las superficies de tela o espuma, absorben casi todo el sonido que emite el sensor, por lo que son indetectables para el robot.

La velocidad del sonido, depende de la temperatura del aire T , además de otros factores¹. Sin embargo, para una simplificación de los cálculos, se empleará la siguiente fórmula, asumiendo que la temperatura T en grados Celsius es la misma en todas partes:

$$v_s = 331,3 + 0,606T \quad (1)$$

Por lo tanto, si t_e es el tiempo que tarda el sonido en llegar desde el emisor al obstáculo, y t_r el tiempo desde que rebota en el obstáculo hasta el receptor:

$$s_1 = v_s t_e \quad s_2 = v_s t_r$$

s_1 es la distancia del sensor al objeto, en el momento de lanzar el pulso. s_2 es la distancia desde el objeto al robot, que puede estar en movimiento y ser diferente a s_1 . El sensor mide t que es $t_e + t_r$.

Si se supone que el robot está parado cuando se realiza la medición, $s_1 = s_2$ y por tanto $t_e = t_r$. Finalmente, s es la distancia del sensor al obstáculo:

$$s = s_1 = s_2 = v_s \frac{t}{2} \quad (2)$$

Es importante mencionar que s no indica la posición exacta del obstáculo, si no la distancia del mismo al sensor.

En el caso de que el robot se mueva con una velocidad \vec{v} , si esta velocidad es pequeña comparada con la del sonido, se puede despreciar. Una velocidad de 1m/s, provocaría una diferencia en la distancia estimada de un objeto a 3m, de aproximadamente 9mm.

2.1.1. Ángulo de apertura θ_s

Para calibrar el sensor, es necesario determinar el ángulo de apertura θ_s , que indica hasta qué punto el sensor puede recibir el eco del sonido que rebota en los objetos. De modo que se realizarán una serie de experimentos, que indicarán de forma experimental dicho ángulo. En la figura 1 se observa un esquema del experimento.

Primero un objeto se coloca en el campo de visión del sensor, en C_1 . Luego se desplaza hacia A_2 , siguiendo una trayectoria circular centrada en O , y se observa el ángulo θ_1 en el que deja de observarse dicho objeto. De igual modo se repite en el otro sentido, y se calcula el ángulo θ_2 .

Dado que el sonar puede no estar centrado, se anotarán las medidas de θ_1 y θ_2 , para luego estimar θ_s .

Tras realizar las mediciones, se obtienen los siguientes resultados:

¹http://en.wikipedia.org/wiki/Speed_of_sound#Practical_formula_for_dry_air

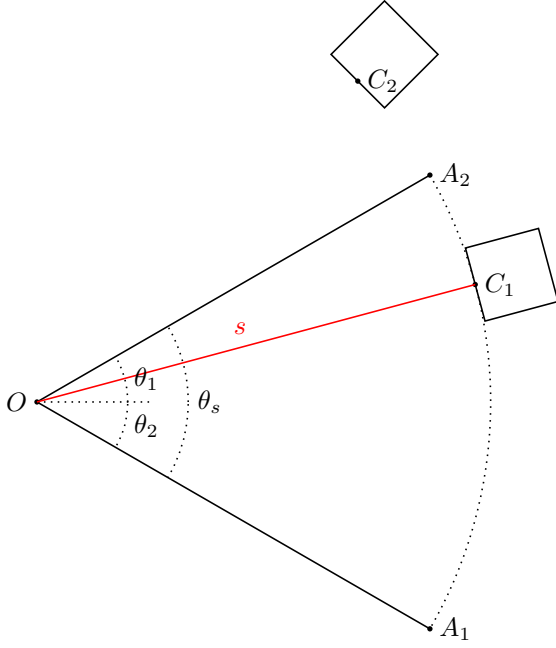


Figura 1: Experimento para calcular el ángulo θ_s en el sensor de ultrasonido.

s (m)	$\theta_1(^{\circ})$	$\theta_2(^{\circ})$
$\pm 5\text{mm}$	$\pm 1^{\circ}$	$\pm 1^{\circ}$
0,20	30	27
0,30	30	27
0,40	31	28
0,50	28	26
0,60	30	28
0,80	31	29
1,00	29	29
1,20	30	28

De forma que se aproxima el valor empleando la media aritmética a:

$$\theta_1 = 30 \pm 1^{\circ} \quad \theta_2 = 28 \pm 1^{\circ}$$

Finalmente, para θ_s :

$$\theta_s = 58 \pm 2^{\circ}$$

El fabricante indica un ángulo de apertura de 60° , centrado, con 30° hacia cada lado. De forma que el resultado obtenido experimentalmente se acerca bastante al resultado esperado.

2.1.2. Distancia medida

Para completar la calibración, es necesario poder calcular la distancia s , sabiendo el tiempo t . Para ello, se realizará otro experimento, midiendo las distancias de forma manual s_r con una regla, y comparándolas con el valor teórico s_t , calculado con la ecuación 2.

Se observa que el valor teórico s_t , no coincide con el valor real medido s_r .

Sería posible solucionar el problema empleando una corrección del error. Para ello se emplea una regresión

s_r (m)	s_t (m)	t (μs)
$\pm 0,05$		± 20
0,20	0,138	805
0,30	0,210	1225
0,40	0,281	1638
0,50	0,353	2060
0,60	0,429	2500
0,80	0,568	3305
1,00	0,713	4155
1,20	0,859	5000

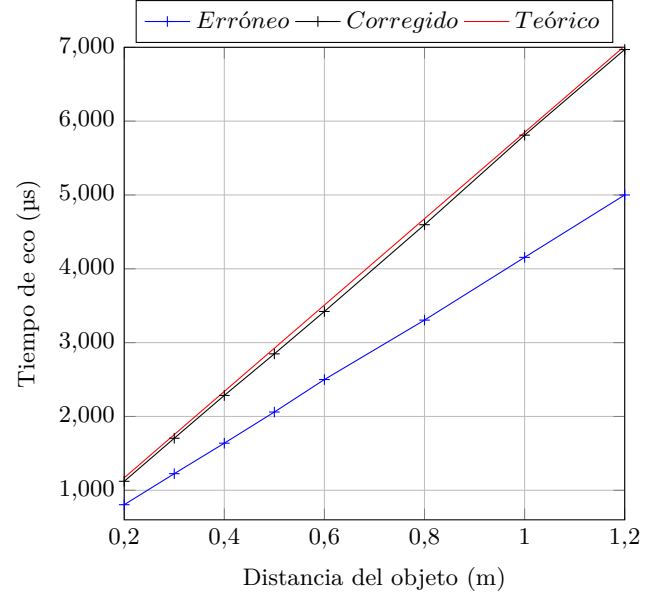


Figura 2: Valores experimentales del sensor de ultrasonidos.

lineal. Con Octave, `polyfit` muestra los coeficientes de la recta:

$$2.3864\text{e}+02 \quad 7.8846\text{e}-03$$

Cuya recta (rojo) se puede observar en la figura 2. Su ecuación es:

$$s = 238,64t + 7,8846 \cdot 10^{-3}$$

Después de repasar el programa que muestra el tiempo que tarda el pulso de ultrasonidos en regresar, se ha detectado un error, provocado en una función de las bibliotecas de Arduino.

Resulta que la función `pulseIn` calcula el tiempo que dura un pulso, empleando un bucle, en el que se incrementa un contador. Luego multiplica las vueltas del bucle por el número de instrucciones que de antemano se han calculado. Sin embargo, el compilador `avr-gcc` puede alterar el número de instrucciones cuando se aplican optimizaciones. Este problema, provoca que el tiempo se vea alterado de la forma $t_{real} = t * K$. Siendo $K = 1,3898$, hallado de forma experimental.

Tras programar una nueva biblioteca para emplear el sensor de ultrasonidos, de forma independiente a las optimizaciones del código (empleando interrupciones), se vuelven a realizar las medidas.

s_r	s_t	t
m	m	μs
0.20	0.193	1122
0.30	0.293	1705
0.40	0.392	2285
0.50	0.489	2848
0.60	0.588	3422
0.80	0.789	4597
1.00	0.998	5810
1.20	1.197	6970

Obteniendo por regresión lineal, de la misma forma, los coeficientes:

$$1.7084e+02 \quad 1.0857e-02$$

Y por tanto, la recta característica es:

$$s = 170,84t + 1,0857 \cdot 10^{-2} = 341,68 \frac{t}{2} + 1,0857 \cdot 10^{-2}$$

Existe un error fijo de aproximadamente un centímetro (1.0857cm), que puede deberse a la dificultad de centrar la regla en el centro del sensor de ultrasonidos. Sin embargo, para determinar la distancia de un objeto al robot, esta distancia es despreciable.

Si comparamos la ecuación 1, con el resultado experimental, despreciando el desplazamiento de 1cm, se puede observar que son muy similares.

La velocidad del sonido calculada de forma experimental, resulta ser 341,68 m/s. Este valor puede oscilar, ya que depende de la temperatura. Para calcular la temperatura:

$$T = \frac{v_s - 331,3}{0,606} = 17,129^\circ C$$

La temperatura registrada de la sala es de 18° . Por lo que se puede afirmar que el experimento muestra unos resultados esperados.

El error medio de distancia cometido, en el peor caso (a 1.20m de distancia real) es de unos 4mm.

Es importante mencionar, que un ajuste del error, empleando un método de regresión, permite solventar errores como el que ha sido detectado, pero no los corrige. Por ello, se ha decidido que si los cálculos teóricos, no concuerdan con los esperados, averiguar que ha causado el error.

2.2. LDR

El sensor LDR permite obtener una medida de la cantidad de luz que incide sobre el mismo. El robot dispone de dos sensores LDR. Uno ambiental, que está colocado en la parte superior, orientado hacia arriba. Además está cubierto por un cilindro cerrado de 4cm de altura, de plástico blanco translúcido. Esto le permite obtener una medida de la luz ambiente sin importar la dirección de donde proviene.

El sensor LDR direccional, se encuentra en la parte delantera orientado al frente. Este sin embargo, está

encapsulado en un cilindro abierto, de plástico negro opaco. Lo cual le permite observar de forma direccional la luz que recibe por el frente, con el objetivo de encontrar una luz.

2.2.1. 1dr0: LDR ambiental

Para calibrar el sensor ambiental, se realizarán algunas medidas comparando la distancia a un foco de luz. También se compararán los valores con distintos ángulos, para observar si realmente es invariante a la orientación.

El valor medido por el sensor, está representado en un intervalo $[0, 1024)$ de valores discretos. El valor 0, corresponde con un voltaje de 0V en la entrada de Arduino, y 1023 con +5V. De forma que si l_0 es el valor discreto medido por Arduino:

$$V_0 = 5 \cdot \frac{l_0}{1023}$$

En la figura 3 se observa el montaje del LDR R_2 , junto con otra resistencia R_1 de un valor fijo de 10k Ω .

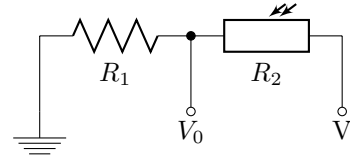


Figura 3: Circuito del sensor LDR.

Para calibrar el sensor ambiental, se realizarán algunas medidas observando el valor de la resistencia del LDR, mientras se varía la distancia del foco de luz, así como el ángulo.

En cada medida, el sensor realiza 5000 mediciones de forma continua, y luego envía el valor de la media aritmética al ordenador, que es registrado.

En la tabla, s indica la distancia a la fuente de luz. Los ángulos indican la orientación del robot:

$s(m)$	0°	-90°	$+90^\circ$	$+180^\circ$
0.20	936	926	933	943
0.40	823	784	831	843
0.60	719	675	729	742
0.80	627	576	628	639
1.00	555	505	560	531
1.20	500	445	497	489
1.40	429	396	446	454
1.60	396	340	407	440
1.80	378	296	383	405

En la figura 4, se puede observar en una escala logarítmica para el eje y, el valor del sensor para cada orientación.

Existe una variación bastante marcada en la orientación de -90° . Esta variación indica que el sensor recibe menos luz de la esperada. Dicha variación está relacionada con la superficie del robot. Dado que hay cables en la superficie, cuando se coloca a -90° , los

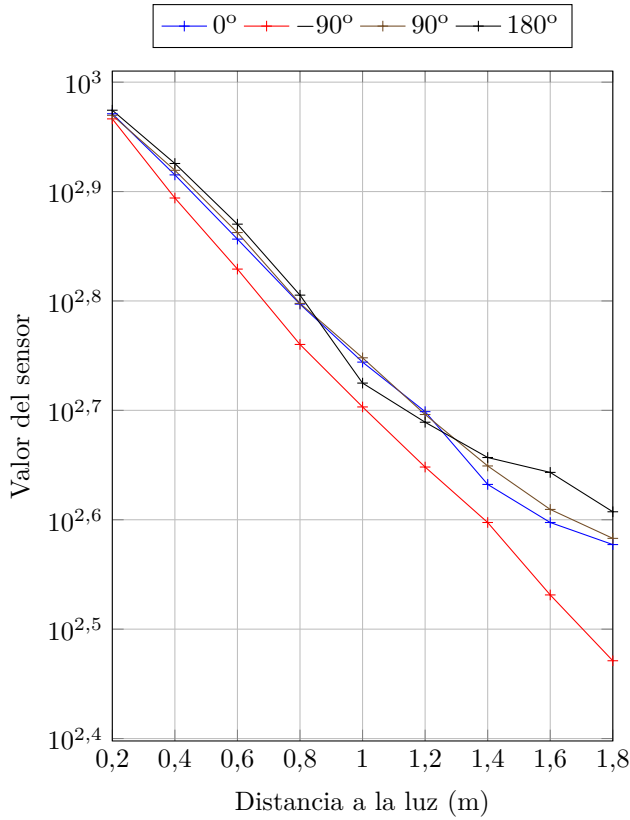


Figura 4: Resultados de calibración para ldr0.

cables se sitúan entre la luz y el sensor, y provocan una disminución de la intensidad.

En las demás orientaciones no se producen grandes cambios. Por lo que se asume que el sensor es realmente independiente de la orientación.

2.2.2. ldr1: LDR direccional

El sensor LDR direccional, por el contrario, trata de aumentar la variación de la luz conforme a la orientación. Para calibrar este sensor se realizará un experimento similar, dividido en 4 orientaciones. Se asume que los valores para los ángulos opuestos, dada la simetría, son iguales.

Se muestran en la siguiente tabla los resultados obtenidos:

$s(m)$	0°	30°	60°	90°
0.10	1015	1011	1004	977
0.20	1015	998	670	404
0.30	1013	915	520	414
0.40	1011	838	416	314
0.50	1008	684	308	254
0.60	1004	534	216	202
0.70	999	500	214	113
0.80	996	394	188	95
0.90	992	355	164	80
1.00	987	352	136	70
1.20	977	277	96	60
1.40	968	173	65	57

Finalmente, en la figura 5, se muestran también en una escala logarítmica para el valor del sensor.

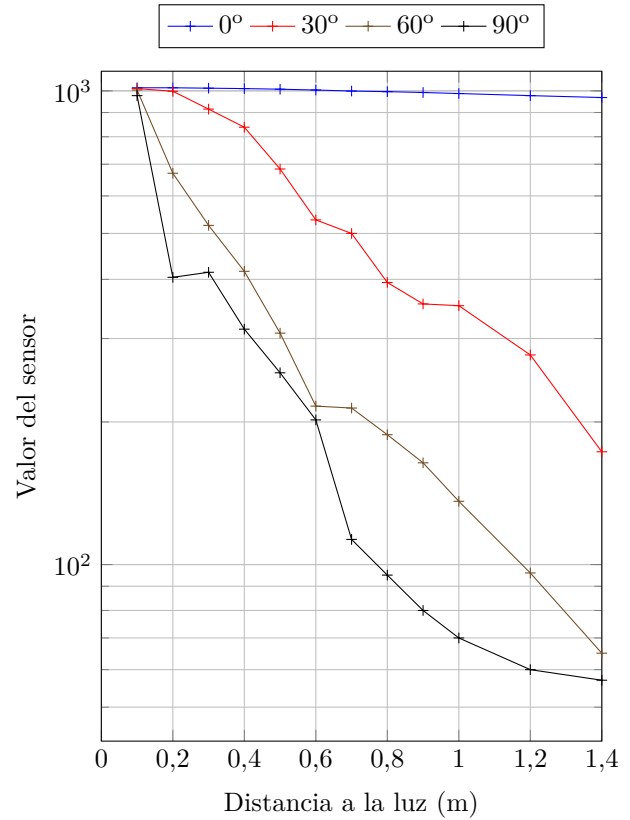


Figura 5: Resultados de calibración para ldr1.

2.3. mouse0: Ratón de bola

En la parte posterior, un ratón de bola permite obtener mediciones del movimiento del robot. Tiene un controlador EM83702BP, que dispone con dos contadores digitales que realizan la medición del desplazamiento en el eje x e y .

Una bola de goma se encuentra en contacto con el suelo. Al desplazar el ratón, la bola gira alrededor de un eje imaginario, que es perpendicular al sentido del desplazamiento y paralelo al suelo. Desplazándose en la dirección de \vec{m} .

Ya que los movimientos efectuados por el robot serán los que provoquen sus dos ruedas, se puede determinar la posición, midiendo sólo el desplazamiento del ratón. Para ello, se observa en la figura 7 un ejemplo de un giro de $\alpha = 45^\circ$.

2.3.1. Movimiento de la bola

Para poder analizar el movimiento que realizará el robot, mediante la información que el ratón proporciona, es necesario conocer como varía el vector \vec{m} que indica el desplazamiento real del ratón, con respecto a la información proporcionada por el mismo.

Para ello, es necesario averiguar el funcionamiento del ratón. Este consta de una bola de goma, que está si-

tuada en un zócalo, donde puede girar libremente, al mover el ratón por una superficie.

Dos rodillos de plástico rozan con la bola, situados en un ángulo de 90° . A continuación se analiza, como el movimiento de la bola, roza en los puntos B_1 y B_2 con los rodillos del ratón. En la figura 6 se observa con detalle el mismo giro que en la figura 7.

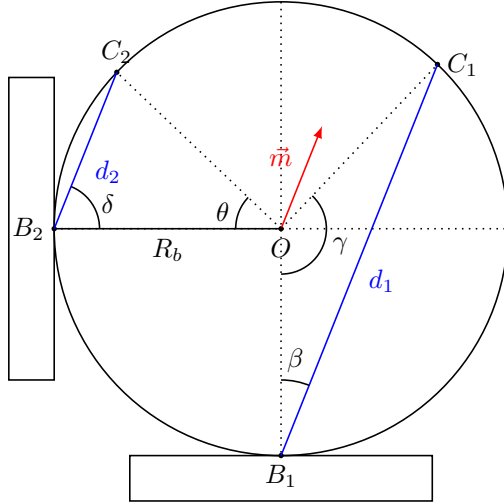


Figura 6: Diagrama de la bola del ratón.

2.3.2. El tamaño de la bola

¿Cómo se mueven los rodillos del ratón, cuando la bola gira? Para responder a esta cuestión, se moverá el ratón un desplazamiento \vec{m} a una velocidad uniforme y en línea recta, y se intentará calcular de forma teórica el resultado esperado.

Los ejes están intercambiados con respecto al orden habitual de un plano. Esto es debido a que el ratón está colocado en el robot como en la figura 6. Sin embargo mantienen el sentido. Hacia arriba y derecha son positivos.

El rodillo inferior, girará rozando continuamente por el punto B_1 a medida que la bola gire. Dado que la dirección \vec{m} permanece constante durante este movimiento, el punto B_1 describirá una trayectoria circular en la superficie de la bola. La bola tiene un radio R_b .

El perímetro de esta circunferencia se denomina por p_1 , que es tangente a B_1 y C_1 y que tiene por diámetro d_1 . De modo que cuando la bola dé una vuelta completa, el rodillo habrá avanzado exactamente p_1 .

La distancia d_1 es la cuerda de ángulo γ , cuya longitud es, por su propia definición:

$$d_1 = R_b \text{ crd } \gamma = 2R_b \sin \frac{\gamma}{2}$$

Además, el ángulo $\gamma = 180^\circ - 2\beta$, pues forma un triángulo isósceles:

$$d_1 = 2R_b \sin(90^\circ - \beta) = 2R_b \cos \beta$$

De igual modo para d_2 :

$$d_2 = R_b \text{ crd } \theta = 2R_b \sin \frac{\theta}{2}$$

Sabiendo que $\theta = 180^\circ - 2\delta$ y que $\beta = 90^\circ - \delta$, puesto que $d_1 \parallel d_2$:

$$d_2 = 2R_b \sin(90^\circ - \delta) = 2R_b \sin \beta$$

Finalmente, para los perímetros:

$$p_1 = \frac{2\pi d_1}{2} = 2\pi R_b \cos \beta \quad p_2 = \frac{2\pi d_2}{2} = 2\pi R_b \sin \beta$$

Si se supone que no hay ningún deslizamiento entre la bola y el rodillo, cuando la bola avance \vec{m} , ésta girará un ángulo θ_b :

$$\theta_b = \frac{m}{R_b}$$

Si R_r es el radio de los rodillos, estos girarán un ángulo:

$$\theta_1 = \frac{\theta_b d_1}{2R_r} \quad \theta_2 = \frac{\theta_b d_2}{2R_r}$$

Sustituyendo d_1 y d_2 :

$$\theta_1 = \frac{\theta_b 2R_b \cos \beta}{2R_r} = \frac{R_b}{R_r} \theta_b \cos \beta$$

$$\theta_2 = \frac{\theta_b 2R_b \sin \beta}{2R_r} = \frac{R_b}{R_r} \theta_b \sin \beta$$

Recuperando el valor de θ_b , el radio R_b desaparece:

$$\theta_1 = \frac{m \cos \beta}{R_r} \quad \theta_2 = \frac{m \sin \beta}{R_r} \quad (3)$$

De modo que se obtiene una relación entre el ángulo con el que se desplaza el ratón y el giro de cada uno de los rodillos, como se deseaba. Además se observa que el tamaño no importa, de la bola, para calcular el giro de los rodillos.

2.3.3. Los optoreceptores

Los rodillos que rozan con la bola, tienen en los extremos unos discos con ranuras equiespaciadas, de modo que cortan un haz de luz infrarrojo al girar. Este efecto, provoca que un sensor registre el número de veces que se ha cortado el haz, y por lo tanto mide, de forma discreta, cuánto ha girado el rodillo, es decir θ_1 y θ_2 .

Cada disco tiene 27 dientes y R_r , el radio de los rodillos, es de 1.4 ± 0.1 mm.

Además, en cada disco se encuentran dos receptores, desfasados 90° , de forma que pueden detectar el sentido de giro. Por cada diente que se avanza, se cuentan 4 pasos en los contadores digitales.

Si o_1 y o_2 son los valores medidos por los contadores que acumulan el número de pasos que se ha desplazado el rodillo:

$$o_1 = \frac{27 \cdot 4\theta_1}{2\pi} \quad o_2 = \frac{27 \cdot 4\theta_2}{2\pi}$$

2.3.7. El ángulo global θ

θ indica en un sistema de referencia global S , el ángulo del robot. En cada giro cambia, y ha de actualizarse. Si $\Delta\theta$ es el giro que se produce en un intervalo Δt , tras cada actualización:

$$\theta(t + \Delta t) = \theta + \Delta\theta$$

Observando la figura 7, α indica el giro que describe el robot, en Δt , por lo que $\Delta\theta = \alpha$:

$$\theta(t + \Delta t) = \theta + \alpha \quad (10)$$

Siendo α el ángulo que gira el robot entre t y $t + \Delta t$.

2.3.8. La posición global \vec{s}

El vector $\vec{s}(t)$ indica la posición del robot en el instante t , tomando un sistema de referencia global S :

$$\vec{s}(t + \Delta t) = \vec{s}(t) + \Delta\vec{s}$$

Sin embargo, el vector $\vec{r}(t)$, indicará el desplazamiento en el instante t , pero en el sistema de referencia del robot S' .

Para transformar el vector \vec{r} de S' a S es necesario tener en cuenta el ángulo θ que poseía el robot al comenzar el desplazamiento:

$$\Delta\vec{s} = R_\theta \vec{r}$$

Substituyendo \vec{r} por la ecuación 8:

$$\Delta\vec{s} = R_\theta R_\phi \begin{bmatrix} r \\ 0 \end{bmatrix} = R_{\theta+\phi} \begin{bmatrix} r \\ 0 \end{bmatrix}$$

Realizando la multiplicación de la matriz:

$$\Delta s_x = r \cos(\theta + \phi) \quad \Delta s_y = r \sin(\theta + \phi)$$

De forma que para cada Δt :

$$\begin{aligned} s_x(t + \Delta t) &= s_x(t) + r \cos(\theta + \phi) \\ s_y(t + \Delta t) &= s_y(t) + r \sin(\theta + \phi) \end{aligned} \quad (11)$$

2.3.9. Odometría

Para determinar la ubicación del robot, bastará con conocer la posición y el ángulo del robot, usando el sistema de referencia global S . De esta forma, la tupla (\vec{s}, θ) contiene toda la información necesaria para ubicar al robot.

Cada vez que el sensor `mouse0`, indique el vector \vec{o} , ha de calcularse \vec{m} . Luego actualizar la posición global \vec{s} y finalmente el ángulo global θ .

2.4. Motores

Para efectuar el movimiento, el robot dispone de dos motores Mabuchi RF-500TB-12560, colocados a ambos lados. Estos motores eléctricos de corriente continua, son rápidos y ligeros, pero no tienen la precisión de los motores paso a paso. Así que para poder controlar la velocidad y aceleración, es necesario realizar una serie de pruebas de calibración.

2.4.1. Relación velocidad-voltaje

Aplicando diferentes configuraciones de ondas PWM, es posible variar la velocidad de los motores con arduino. De modo que se ha realizado un experimento en el que el robot ejecuta un movimiento en línea recta durante 2 segundos, manteniendo una alimentación constante en ambos motores. El resultado puede verse en la figura 8.

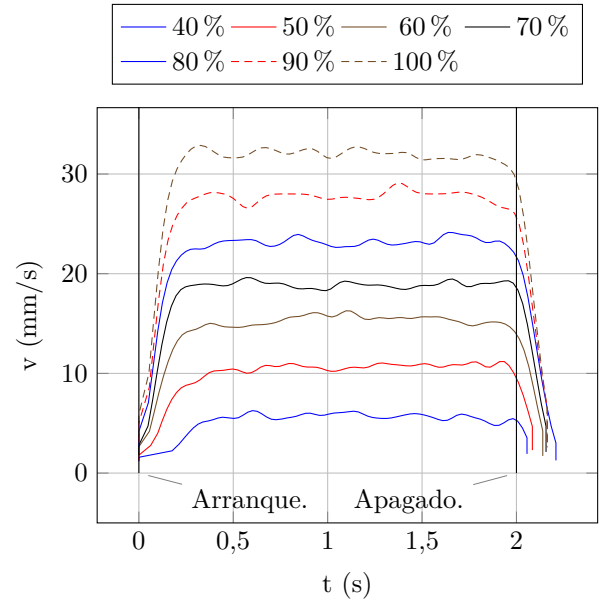


Figura 8: Arranque y parada de los motores, con diferente ciclo de trabajo, a 5V. Se observa una curva de arranque, y otra de detención, debido a la inercia.

Se observa que al comienzo, los motores experimentan una aceleración, tras la cual, se estabilizan en una velocidad relativamente constante, y finalmente se detienen tras un período de frenado.

Es importante conocer la velocidad de los motores, para poder tener una mejor información de odometría. En la figura 9 se observa como aumenta la velocidad conforme se aumenta el ciclo de trabajo de la onda PWM.

Las ecuaciones que describen la aceleración del motor permitirían contrastar la forma de la función. Sin embargo, no serán analizadas aquí, debido a la gran complejidad que conforma un motor. Para estimar la gráfica, se realizará una aproximación lineal. La interpolación que se muestra en la figura 9 es la siguiente

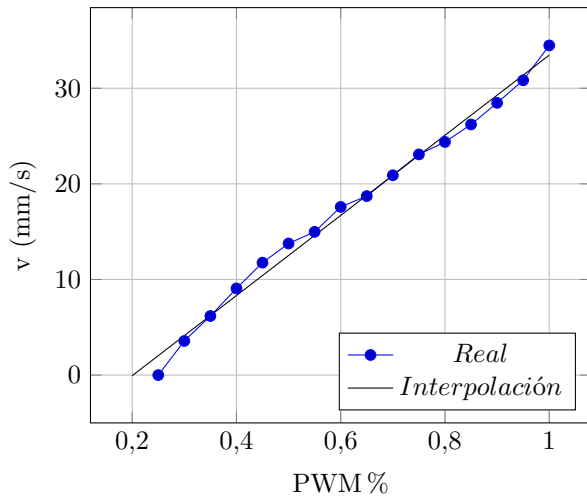


Figura 9: Arranque y parada de los motores, con diferente ciclo de trabajo, a 5V. Se observa una curva de arranque, y otra de detención, debido a la inercia.

recta:

$$f(x) = 41,9103x - 8,4458$$

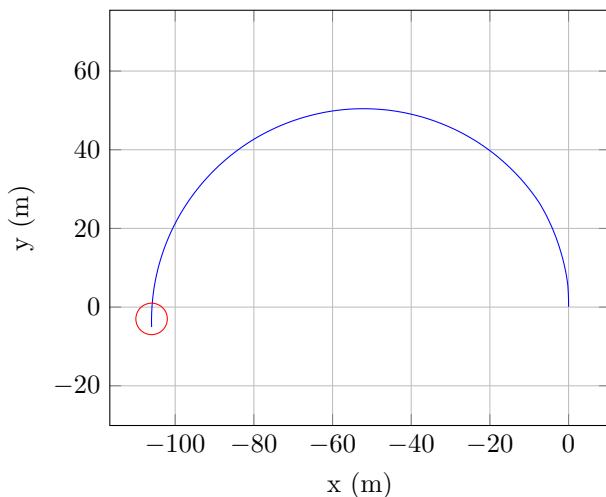


Figura 10: Giro de 180°. Se observa que excede el giro en unos 5,5°, debido a la inercia de los motores.

3. Posibles errores

La función `attachInterrupt` no borra las banderas (flags) `INTF0` y `INTF1` del registro `EIFR`, de forma que al colocar una interrupción por primera vez, es posible que se ejecute al comienzo sin haber recibido tal interrupción. Una posible solución es añadir esta línea al comienzo de la función:

```
EIFR |= _BV(INTF0) || _BV(INTF1);
```

La función `pulseIn` calcula el tiempo dependiendo de las optimizaciones que se produzcan. Es necesario que sea independiente. Además es conveniente sustituirla por alguna función no bloqueante que emplee interrupciones.

4. TODO

Este documento está incompleto. Faltan cálculos en la parte de la odometría, así como los experimentos que verifican la posición de robot.

Falta cambiar los gráficos a pgf. Corregir la ortografía. Usar la precisión en los experimentos. Encontrar la ecuación que describe la luz con la distancia, en el LDR. Y las ecuaciones que describen el movimiento de los motores, sabiendo el voltaje y tiempo de funcionamiento.

La última versión se encuentra en <https://github.com/rodarima/robotica>.