

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. Son los siguientes:

1. Sensor de ultrasonidos HC-SR04
2. Ratón de bola
3. Sensor de luz LDR dirigido hacia delante
4. Sensor de luz LDR ambiental

1.2. Motores

Para efectuar el movimiento, dispone de dos motores Mabuchi RF-500TB-12560, colocados a ambos lados.

2. Calibración de sensores

2.1. 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 colocados en la misma dirección, y posicionados en la parte delantera 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.

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

$$v_s = 331,3 + 0,606T_c$$

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 el obstáculo al receptor:

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

Sin embargo, medimos t que es $t_e + t_r$. Si suponemos que el robot está parado cuando haga la medición, $s_1 = s_2$ y por tanto $t_e = t_r$. Finalmente, s es la distancia del sensor al obstáculo:

$$s = s_1 = s_2 = v_s \frac{t}{2} \quad (1)$$

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

2.1.1. Ángulo de apertura θ_{sonar}

Para calibrar el sensor, es necesario determinar el ángulo de apertura θ_{sonar} , que indica hasta qué punto el sensor puede recibir el eco del sonido. 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_{sonar} .

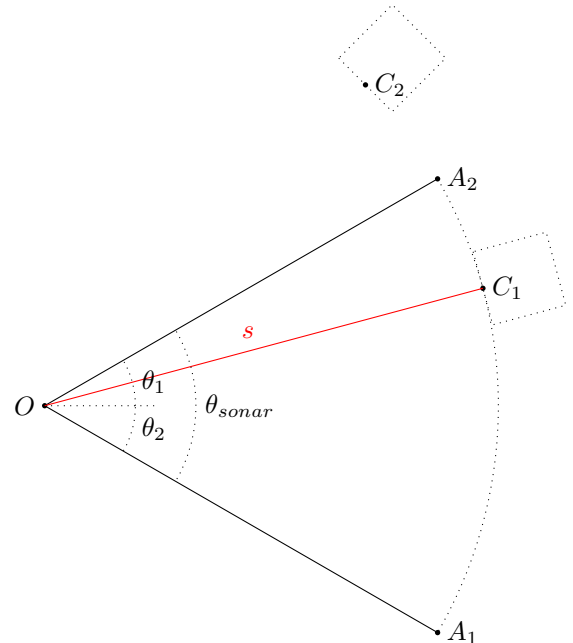


Figura 1: Sensor de ultrasonido.

Tras realizar las mediciones, se obtienen los siguientes resultados.

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

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

Analizamos ahora la media aritmética, la desviación típica y la dispersión muestral o varianza de los valores de θ_1 y θ_2 , respectivamente:

$$\bar{\theta}_1 = 29,875 \quad s(\theta_1) = 0,99103 \quad s^2(\theta_1) = 0,98214$$

$$\bar{\theta}_2 = 27,75 \quad s(\theta_2) = 1,0351 \quad s^2(\theta_2) = 1,0714$$

De forma que se aproxima el valor a:

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

$$\mu_{\theta_s} = \mu_{\theta_1} + \mu_{\theta_2} = 57,625$$

$$\sigma_{\theta_s} = \sqrt{\sigma_{\theta_1}^2 + \sigma_{\theta_2}^2} = 2,0261$$

Finalmente, aproximando θ_s :

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

Se empleará la media de los valores de los ángulos, para reducir los errores que se pudieran producir en las medidas de los mismos, quedando $\theta_1 = 29,875^{\circ}$ y $\theta_2 = 27,75^{\circ}$.

Por lo tanto ya se puede obtener el ángulo de apertura:

$$\theta_{sonar} = \theta_1 + \theta_2 = 57,625^{\circ}$$

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 1.

s_r	s_t	t
m	m	μs
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

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 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:

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

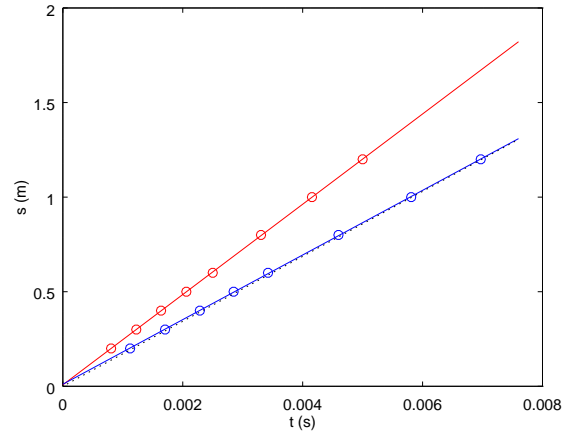


Figura 2: Valores experimentales erróneos (rojo) y correctos (azul). La recta punteada es el valor teórico.

Después de repasar el programa que muestra el tiempo en Arduino, resulta que la función `pulseIn` calcula el tiempo que ha pasado, empleando un bucle, y luego multiplica las vueltas del bucle por el número de instrucciones. Sin embargo, el compilador `avr-gcc` puede alterar este número 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, y empleando las 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:

$$1.7084\text{e}+02 \quad 1.0857\text{e}-02$$

Y por tanto, la recta característica es:

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

Si la comparamos con la ecuación 1, se puede observar que es muy similar a lo esperado.

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:

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

Dado que el experimento se ha realizado por la noche, esa temperatura se encuentra dentro de un margen adecuado. La temperatura medida por un termómetro en la habitación es de 18°C . Por lo que se puede afirmar que el experimento muestra unos resultados esperados.

Existe un error fijo de aproximadamente un centímetro, que se debe 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.

Es importante mencionar, que un ajuste del error, empleando un método de regresión, permite esconder errores como el que ha sido detectado y corregido. Por ello, se ha decidido emplear los cálculos teóricos, y si 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, está colocado en la parte superior, orientado hacia arriba. Además está cubierto por un cilindro cerrado de 4cm de plástico blanco. 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 caucho negro mate. 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. LDR ambiental

Para calibrar el sensor ambiental, se realizarán algunas medidas comparando la distancia a un foco de luz, y la invarianza de la orientación.

964

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.

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

En la figura 3 se observa el montaje del LDR R_2 , junto con otra resistencia R_1 de un valor fijo de $10\text{k}\Omega$.

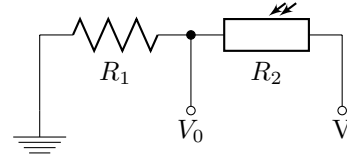


Figura 3: Circuito del sensor LDR.

Para calcular el voltaje en V_0 :

$$I = \frac{V}{R_T} = \frac{V}{R_1 + R_2}$$

$$V_0 = I \cdot R_1 = V \frac{R_1}{R_1 + R_2} = \frac{V}{1 + \frac{R_2}{R_1}}$$

Y posteriormente el valor de R_2 :

$$R_2 = R_1 \frac{V - V_0}{V_0}$$

La relación entre la iluminación I_V y la resistencia R viene dada por²:

$$\frac{E_2}{E_1} = \left(\frac{R_2}{R_1} \right)^\gamma$$

Y entre la distancia D y la iluminación:

$$\frac{E_1}{E_2} = \left(\frac{D_2}{D_1} \right)^2$$

De forma que, para calcular el parámetro γ :

$$E_2 = E_1 \cdot \left(\frac{R_2}{R_1} \right)^\gamma \Rightarrow \left(\frac{R_2}{R_1} \right)^{-\gamma} = \left(\frac{D_2}{D_1} \right)^2$$

$$\gamma = \frac{2 \ln \frac{D_2}{D_1}}{-\ln \frac{R_2}{R_1}}$$

s_r	0.20	0.30	0.40	0.60	0.80	1.00	1.20
ldr	500	394	330	255	202	160	140

2.3. Ratón

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.

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 4 un ejemplo de un giro de $\alpha = 45^\circ$.

²http://www.brighton-webs.co.uk/electronics/light_dependent_resistor.aspx

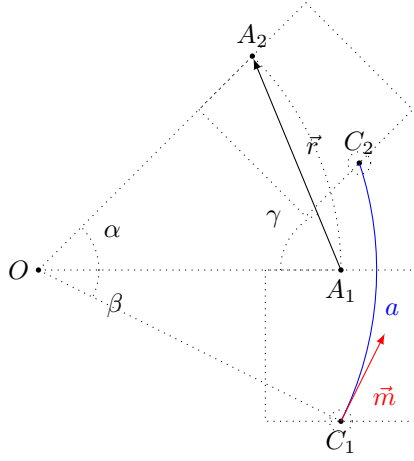


Figura 4: Diagrama de un giro del robot.

El vector \vec{r} indica el movimiento que realizó el robot en línea recta. Conociendo el ángulo γ :

$$R' = \overline{OA_1} \quad R = \overline{OC_1}$$

$$r_x = \begin{cases} R'(1 - \cos \gamma) & \text{si } \gamma < 0 \\ -R'(1 - \cos \gamma) & \text{si } \gamma > 0 \\ 0 & \text{si } \gamma = 0 \end{cases}$$

$$r_y = \begin{cases} -R'(\sin \gamma) & \text{si } \gamma < 0 \\ R'(\sin \gamma) & \text{si } \gamma > 0 \\ m_x & \text{si } \gamma = 0 \end{cases}$$

2.3.1. Movimiento de la bola

Para poder analizar el movimiento que realizará el robot, mediante la información que este proporciona, es necesario conocer como varía el vector \vec{m} con respecto a la información proporcionada por los optocodificadores.

De modo que se analiza a continuación, como el movimiento de la bola, roza en los puntos B_x y B_y con los rodillos del ratón. En la figura 5 se observa con detalle el mismo giro que en la figura 4.

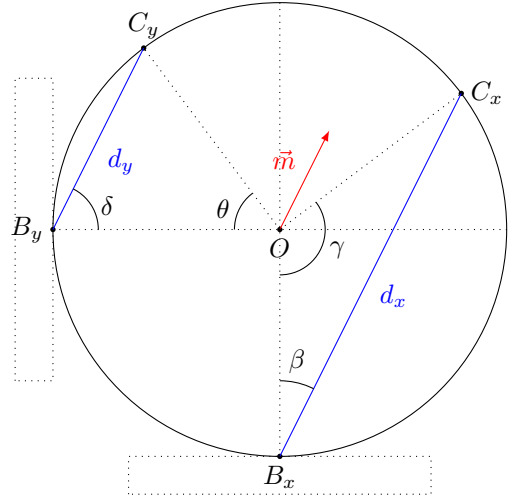


Figura 5: Diagrama de la bola del ratón. Los ejes están invertidos.

2.3.2. El tamaño de la bola

A continuación, trataré de averiguar cómo se mueven los rodillos del ratón, cuando la bola gira. Para ello moveré el ratón un desplazamiento \vec{m} a una velocidad uniforme y en línea recta.

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. Sin embargo mantienen el sentido. Hacia arriba y derecha son positivos.

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

Llamo p_x al perímetro de esta circunferencia, que es tangente a B_x y C_x y que tiene por diámetro d_x . De modo que cuando la bola dé una vuelta completa, el rodillo habrá avanzado exactamente p_x .

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

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

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

$$d_x = 2R_b \sin(90 - \beta) = 2R_b \cos \beta$$

De igual modo para d_y :

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

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

$$d_y = 2R_b \sin(90 - \delta) = 2R_b \sin \beta$$

Finalmente, para los perímetros:

$$p_x = \frac{2\pi d_x}{2} = 2\pi R_b \cos \beta \quad p_y = \frac{2\pi d_y}{2} = 2\pi R_b \sin \beta$$

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

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

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

$$\theta_x = \frac{\theta_b d_x}{2R_r} \quad \theta_y = \frac{\theta_b d_y}{2R_r}$$

Sustituyendo d_x y d_y :

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

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

Recuperando el valor de θ_b , el radio R_b desaparece:

$$\theta_x = \frac{m \cos \beta}{R_r} \quad \theta_y = \frac{m \sin \beta}{R_r}$$

Es decir, que el tamaño no importa, de la bola.

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_x y θ_y .

Cada disco tiene 27 dientes y R_r mide 1.35 ± 0.1 mm.

o_x y o_y son los valores medidos por los contadores que acumulan el número de veces que se ha cortado el haz, y por lo tanto el número de dientes que se ha desplazado el rodillo.

$$o_x = \frac{27\theta_x}{2\pi} \quad o_y = \frac{27\theta_y}{2\pi}$$

Sustituyendo el valor del ángulo θ_x y θ_y :

$$o_x = \frac{27}{2\pi R_r} m \cos \beta \quad o_y = \frac{27}{2\pi R_r} m \sin \beta$$

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.