



PRÁCTICA 1

SENSORES Y ACTUADORES EN ROS

1 Introducción a la práctica.

El objetivo de esta práctica es realizar una revisión completa de los conceptos aprendidos durante el Tema 2 "Sistemas de Percepción", empleando para ello la plataforma de desarrollo robótico ROS, el simulador STDR y el robot real Amigobot.

Concretamente se va a trabajar con suscripciones a los topics de los sensores de distancia (**sonar**, **laser**) y odometría (**odom**) para poder leer la información de los sensores. Además, se trabajará con una publicación en el topic de los motores (**cmd_vel**) que permite enviar comandos a los actuadores del robot y conseguir así desplazarlo por el entorno.

Se empleará como entorno de prácticas un laberinto de prueba como el mostrado en la Figura 1, con una resolución de 1.000x1.000 píxeles que equivalen a unas dimensiones reales de 10x10 metros.

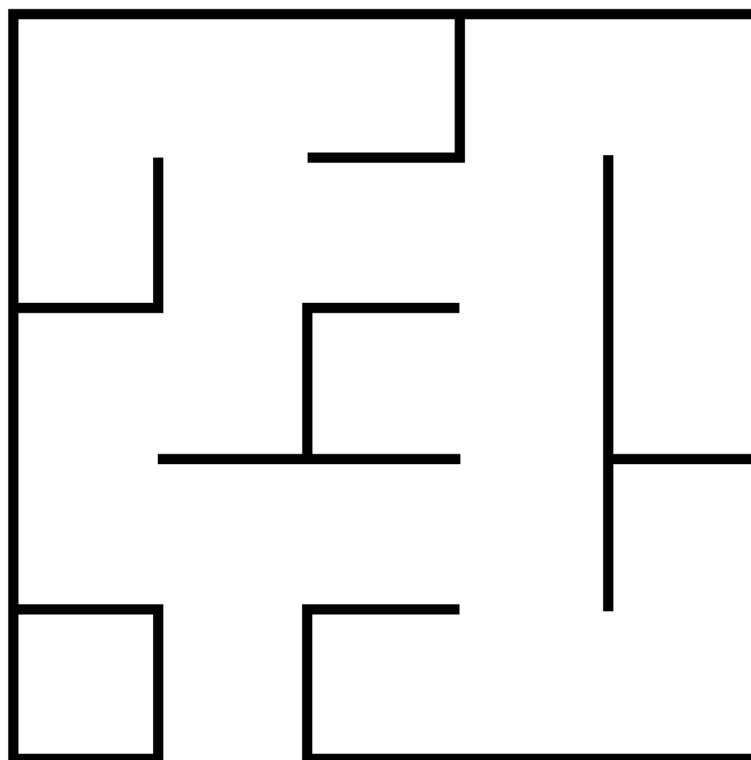


Figura 1. Entorno de movimiento del robot (mapa.png)



Las condiciones en las que se van a desarrollar las diferentes funciones son:

1. Se utilizará en primer lugar el simulador STDR y, una vez comprobado el funcionamiento correcto de las funciones, se realizarán las pruebas con el robot real Amigobot.
2. El objetivo final es conseguir disponer de funciones que permitan al robot percibir el entorno que le rodea correctamente por medio de sus sensores y poder moverse por el mismo con funciones sencillas de avanzar y girar.

2 Sensores en ROS.

2.1 Caracterización de los sensores de odometría

Se desea caracterizar los sensores de odometría del robot con la finalidad de obtener información sobre cómo se desplaza del robot por el entorno. Para ello será necesario emplear una suscripción al topic **odom**, tal y como se muestra en la guía rápida de ROS disponible en la documentación de la asignatura (véase ejemplo de la Figura 2).

```
%% DECLARACIÓN DE SUBSCRIBERS
odom=rossubscriber('/robot0/odom'); % Suscripción a la odometría

%% Nos aseguramos recibir un mensaje relacionado con el robot "robot0"
while (strcmp(odom.LatestMessage.ChildFrameId,'robot0')~=1)
    odom.LatestMessage
end

%% Obtenemos la posición actual
pos=odom.LatestMessage.Pose.Pose.Position;
```

Figura 2. Ejemplo de suscripción al topic **odom** y lectura de la última posición

En primer lugar, se deberá realizar una conexión al topic **odom** y obtener información sobre todos los campos que nos ofrece este tipo de mensaje. Además, para poder caracterizar el sensor de odometría, se propone realizar una serie de movimientos con el robot registrando las posiciones que nos va proporcionando el topic **odom**. Los movimientos que se van a realizar son en línea recta ($\Omega=0\text{rads}^{-1}$): $V=0.1\text{ms}^{-1}$, $V=0.3\text{ms}^{-1}$, $V=0.5\text{ms}^{-1}$, $V=0.7\text{ms}^{-1}$ y $V=0.9\text{ms}^{-1}$; y girando sobre el mismo sitio ($V=0\text{ms}^{-1}$): $\Omega=0.3\text{rads}^{-1}$, $\Omega=0.7\text{rads}^{-1}$ y $\Omega=0.9\text{rads}^{-1}$. Para ello, es necesario publicar las diferentes combinaciones de velocidades en el topic **cmd_vel** para comprobar cuál es la máxima resolución de velocidad lineal y angular que podemos obtener (véase ejemplo de la Figura 3 para poder publicar en el topic **cmd_vel**)



```
%% DECLARACIÓN DE PUBLISHERS
pub = rospublisher('/robot0/cmd_vel', 'geometry_msgs/Twist');

%% GENERACIÓN DE MENSAJE
msg=rosmesssage(pub) %% Creamos un mensaje del tipo declarado en "pub"
(geometry_msgs/Twist)

% Rellenamos los campos del mensaje para que el robot avance a 0.2 m/s
% Velocidades lineales en x,y y z (velocidades en y o z no se usan en
robots diferenciales y entornos 2D)
msg.Linear.X=0.2;
msg.Linear.Y=0;
msg.Linear.Z=0;

% Velocidades angulares (en robots diferenciales y entornos 2D solo se
utilizará el valor Z)
msg.Angular.X=0;
msg.Angular.Y=0;
msg.Angular.Z=0;

%% Definimos la perodicidad del bucle (10 hz)
r = robotics.Rate(10);

%% Bucle de control infinito
while (1)
    send(pub,msg);
    % Temporización del bucle según el parámetro establecido en r
    waitfor(r)
end
```

Figura 3. Ejemplo de publicación en el topic **cmd_vel**

Se pide:

1. Indique y describa la información que nos ofrece el mensaje disponible en el topic **odom**. Muestre algún ejemplo de captura.
2. Mida la resolución máxima (**q**) de odometría lineal y angular máxima con las diferentes combinaciones de velocidades propuestas en el simulador STDR. Construya una tabla como la que se muestra a continuación con las 8 combinaciones propuestas.

v (ms^{-1})	Ω (rads^{-1})	q_lineal (m)	q_angular (r)
0.1	0.0
0.3	0.0
...

3. Mida la resolución máxima (**q**) de odometría lineal y angular con las diferentes combinaciones de velocidades propuestas en el robot real (en este caso, debe tener en cuenta las aceleraciones y deceleraciones del robot). Construya una tabla como la del apartado anterior con las 8 combinaciones propuestas.



2.2 Caracterización de los sensores de distancia ultrasónicos

Se desea caracterizar los sensores s3nar del robot con la finalidad de obtener informaci3n de distancia sobre el entorno que rodea al robot. Para ello, ser3 necesario emplear una suscripci3n al topic **sonar**, a semejanza de la realizada con el topic **odom** en el apartado anterior.

En primer lugar, se deber3 realizar una conexi3n al topic **sonar** y obtener informaci3n sobre todos los campos que nos ofrece este tipo de mensaje. Adem3s, para poder caracterizar el sensor de sonar, se propone posicionar el robot en unas zonas concretas del entorno que nos permitan caracterizar las medidas correctamente.

Se pide:

1. Indique y describa la informaci3n que nos ofrece el mensaje disponible en el topic **sonar**. Muestre alg3n ejemplo de captura.
2. En el simulador STDR, posicione el robot de tal forma que exista una distancia de 2m de uno de los sensores s3nar concretos del robot. 3Qu3 posici3n y s3nar se ha elegido?
3. Obtenga 1.000 medidas de distancia del sensor s3nar elegido y dibuje en una gr3fica (comando **plot** en Matlab) la distancia medida. 3Son estables las medidas? 3Hay ruido en la medida? En caso afirmativo, calcule el valor m3ximo, medio y la varianza del ruido.
4. Implemente un filtro media m3vil con los 3ltimos 5 valores de distancia y dibuje en una gr3fica. 3Son m3s estables las medidas que en el caso anterior? 3Ser3 3til este m3todo si el robot est3 en movimiento en lugar de permanecer est3tico?
5. Repita pasos 2-4 con el robot real.
6. En el simulador, posicione al robot en la casilla X. Estando el robot perfectamente paralelo a las paredes de la celda, seleccione las medidas de los sensores sonar del robot que podr3an resultar 3tiles para obtener las cuatro rectas que definen las paredes que lo rodean. Compruebe que las orientaciones de las rectas son paralelas dos a dos y perpendiculares entre ellas. Defina una **funci3n de calidad** para obtener el grado de confianza de dichas paredes empleando, por ejemplo, la relaci3n entre las diferentes pendientes.
7. Dise3e una **funci3n** que indique, mediante un c3digo, el n3mero de paredes que se encuentra el robot en sus laterales. La codificaci3n que se puede emplear para proporcionar la salida es la que se muestra en la Figura 4. Adem3s, indique el **grado de confianza** basado en la funci3n de calidad definida en el apartado anterior.

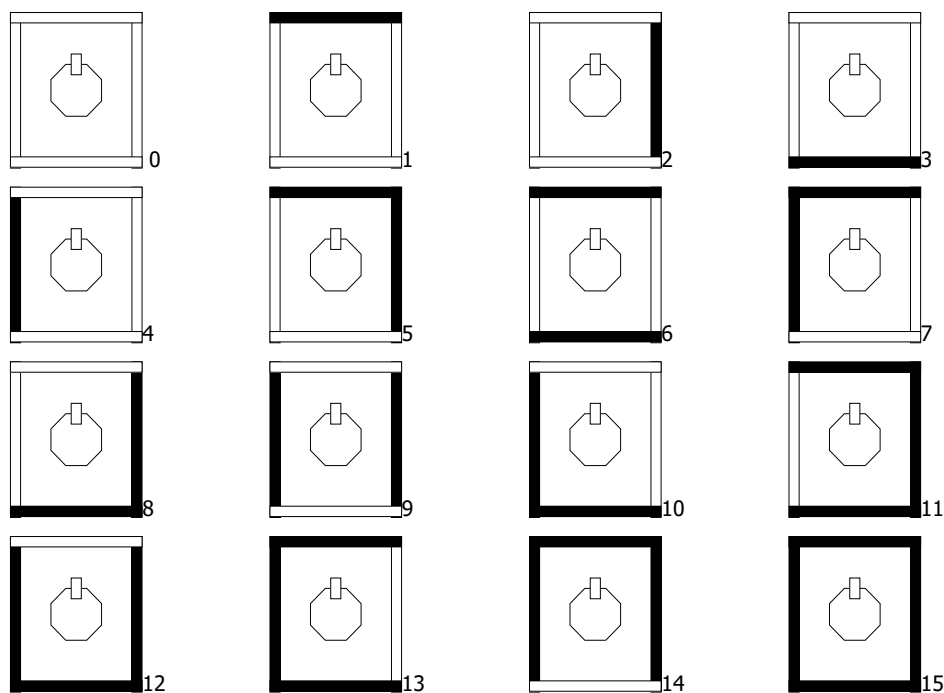


Figura 4. Codificación del número de paredes identificadas.

8. Comprueba los resultados de la función diseñada en el apartado anterior tanto con el simulado STDR como con el robot real y con las 16 posibles combinaciones. Complete la tabla siguiente indicando los resultados.

Robot_real/Simulador	Combinación real	Combinación identificada	Grado confianza
Simulador	0
Simulador	1
...

2.3 Sensores de distancia láser

Se desea caracterizar el sensor láser del robot con la finalidad de obtener información de distancia sobre el entorno que rodea al robot. Para ello, será necesario emplear una suscripción al topic **laser**, a semejanza de la realizada en el apartado anterior.

En primer lugar, se deberá realizar una conexión al topic **laser** y obtener información sobre todos los campos que nos ofrece este tipo de mensaje. Además, para poder caracterizar el sensor de láser, se propone posicionar el robot en unas zonas concretas del entorno que nos permitan caracterizar las medidas correctamente.

Se pide:

1. Indique y describa la información que nos ofrece el mensaje disponible en el topic **laser**. Muestre algún ejemplo de captura.
2. Repita los pasos 2-4 y 6-8 del apartado anterior con el sensor láser.
3. ¿Tiene más o menos ruido que el sensor sónar?
4. ¿Cómo es el grado de confianza empleando este sensor?



3 Actuadores en ROS.

La plataforma de desarrollo robótico ROS ofrece un topic de control con robots móviles que se desplazan por un terreno plano conocido como **cmd_vel** del tipo **geometry_msgs/Twist**. Se desea realizar unas funciones que nos permitan desplazar el robot por el entorno que le rodea, para ello será necesario publicar en el topic **cmd_vel**, tal y como se muestra en la guía rápida de ROS disponible en la documentación de la asignatura (véase ejemplo de la Figura 3).

Se pide:

1. Diseñe una función **avanzar** que reciba como parámetro de entrada la distancia a avanzar (2m, 4m...).
2. Diseñe una función **girar** que se reciba como parámetro de entrada el ángulo a girar (45°, 90°,...).
3. Empleando el simulador STDR, y utilizando las funciones **avanzar** y **girar**, navegue con el robot desde la esquina inferior derecha (4,-4,0) hasta la salida utilizando los puntos centrales de las casillas como coordenadas de destino.
4. Realice un recorrido con el robot real concatenando los siguientes tramos:
 - 1 tramo recto de 2m
 - 1 giro de 90°
 - 1 tramo recto de 1m
 - 1 giro de -90°
 - 1 tramo recto de 1m

Compruebe el error final que se ha obtenido con el robot real. ¿Cuál es el error de odometría global que se ha cometido?

4 Memoria

Se realizará una entrega en un único archivo (en un archivo comprimido .zip / .rar) de la práctica que contenga:

- Un informe/memoria (en .pdf). Esta memoria debe incluir los datos / tablas solicitadas en el guion, capturas de pantalla de los códigos programados, los resultados numéricos o gráficos obtenidos, los problemas encontrados y soluciones propuestas, las respuestas a aquellas preguntas / comparativas / razonamientos que se indican en el guion y unas conclusiones.
- Archivos fuente de los apartados implementados en Matlab (.m / .mlx)