

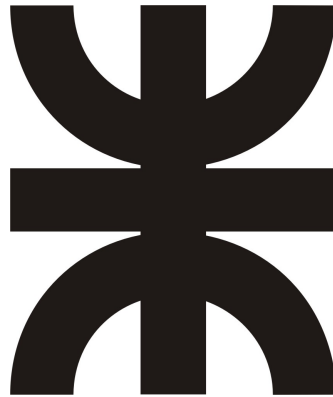
UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL CÓRDOBA

Fundamentos de Robótica móvil

Trabajo Práctico N°3:

Mediciones de sensor de barrido láser



Profesores:

- Prof. Dr. Ing Gaydou, David A.
- Prof. Dr. Ing. González Dondo, Diego.
- Prof. Dr. Ing. Perez Paina, Gonzalo F.

Alumnos:

- Dogliani, Matías / Legajo: 72.152
- Nicolodi, Juan / Legajo: 66875
- Pistelli, Franco / Legajo: 70.735



Índice

1. Introducción	1
1.1. Rosbag	1
1.2. Transformación de puntos	2
2. Actividades	2
2.1. Actividad 1-5	2
2.2. Actividad 6	4
2.3. Actividad 7-8	4
2.4. Actividad 9	5
3. Conclusión	6

1. Introducción

Todo robot móvil necesita sensores, tanto para extraer información relevante de su propio estado como información del entorno, para así poder tomar decisiones basada en estos datos. En este trabajo se utiliza el modelo del robot TurtleBot3 y se extraen los datos de sensores **propioceptivos** y **exteroceptivos**. Del primero extraemos los datos de odometría del robot y del segundo, un sensor de barrido láser, obtendremos una nube de puntos para luego reconstruir el mapa del entorno *world* simulado, de la figura 1

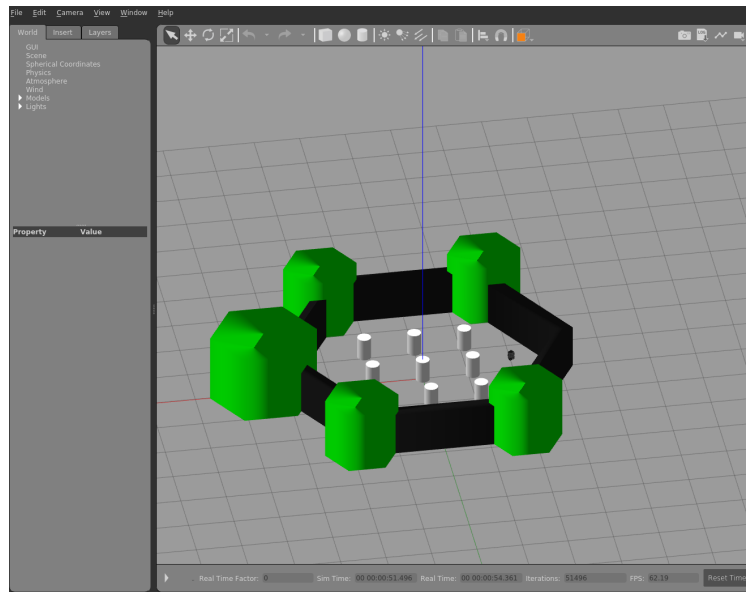


Figura 1: Simulación de entorno *world*

Los datos se almacenan utilizando la herramienta **rosvbag** de ROS y luego se los procesa para obtener como resultado final el mapa del entorno. En primer lugar se realiza una conversión del archivo de registro de ROS a un archivo de texto, luego se enlazan las muestras del sensor de barrido láser y odometría tomadas en tiempo lo suficientemente cercanos como para considerarlos iguales, y por último se realiza una transformación para ubicar a los puntos en un mismo sistema de referencia.

1.1. Rosbag

Es una herramienta que permite guardar la información de los tópicos activos o los que se les pase por argumento para crear un registro de acciones y valores y además permite, luego, reproducir los mismos mensajes almacenados. Esto es útil si se quieren realizar varios test con una trayectoria y velocidad constante, entre otras muchas cosas.

En este práctico utilizamos el comando de esta herramienta para grabar los mensajes de los tópicos de sensor láser, comandos de velocidad y odometría.

1.2. Transformación de puntos

Todos los sistemas de referencias fijos en el robot (su propio sistema de referencia, lo de sus sensores, sus diferentes partes mecánicas, etcétera) son no inerciales y se desplazan con el desplazamiento del robot.

Para formular las ecuaciones del movimiento del robot, mapear un entorno, planificar caminos, etc. es necesario fijar un marco inercial de referencia, y para esto se debe realizar la transformación de un punto en un sistema dado al sistema de referencia.

La matriz de transformación para mapear un punto en el sistema de referencia O (odometría) a L (láser) puede representarse mediante la ecuación 1

$${}^O_L T = \begin{bmatrix} {}^O_L \mathbf{R} & {}^O_L \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix} \quad (1)$$

Donde \mathbf{R} es la matriz que estará conformada por todas las rotaciones posibles, en nuestro caso solo se consideran las rotaciones en plano xy que esta determinada por el ángulo θ . Y la matriz tiene la forma de la expresión 2

$${}^O_L R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Y el vector ${}^O_L \mathbf{p}$ son las coordenadas del sistema de referencia L en el sistema de referencia O que definirá la traslación de un sistema de referencia con respecto al mismo. Queda definido por la ecuación 3. (Como el sensor se ubica sobre el robot y su eje es el mismo que el eje del robot este vector está formado por los datos de odometría x,y posición del robot con respecto al sistema de referencia del plano)

$${}^O_L p = \begin{bmatrix} x_{odom} \\ y_{odom} \\ 0 \end{bmatrix} \quad (3)$$

Entonces, un punto p_1 en el sistema de referencia L se podrá transformar al sistema de referencia O mediante la multiplicación matricial:

$${}^O p_1 = [{}^O_L T] [{}^L p_1] \quad (4)$$

2. Actividades

2.1. Actividad 1-5

Se utilizaron los paquetes de simulación de TurtleBot3 ya descargados y contruidos en el práctico 1 de la materia. Se ejecutaron los archivos de lanzamiento para simulación del robot en gazebo y además el de teleoperación. Además se ejecutó la

herramienta `roslaunch` para grabar los mensajes de odometría, comandos de velocidad y mediciones del sensor de barrido láser.¹

```
roslaunch rosbag_record /cmd_vel /odom /scan
```

Con comandos de teclado movimos el robot por todo el entorno de la figura 1 para captar con el sensor láser todos los puntos del mapa. Una vez recorrido todo el mapa y terminado de ejecutar la grabación de los mensajes mencionados se convirtió el archivo de extensión `.bag` a dos archivos de texto. Para esto se utilizó un paquete de terceros `roslaunch`, descargado en el espacio de trabajo de ROS `catkin_ws/src` y construido con la herramienta `catkin make`

```
roslaunch rosbag2txt rosbag2txt.launch bagfile:=/home/<user>/
bagfile.bag
```

De esta forma obtuvimos dos archivos de texto `odom.txt` y `scan.txt`. El primero contiene los datos de odometría en columnas como se observa en la figura 2. En el segundo, en cambio, se encuentran los puntos del sensor de barridos láser en filas, separados por el correspondiente *timestamp*, como se puede ver en la figura 3

	Timestamps	x	y	θ	v	ω
0	6.670	-2.00002	-0.50000	0.00005	0.00000	0.00001
1	6.703	-2.00002	-0.50000	0.00005	0.00000	0.00001
...
5217	180.570	-0.93910	-1.29919	-1.89184	-0.00054	0.00910
5218	180.604	-0.93908	-1.29920	-1.89162	0.00026	0.00691

5219 rows × 6 columns

Figura 2: Datos de odometría

	0	1	2	...	358	359	360
0	6.738	inf	inf	...	inf	inf	inf
1	6.937	inf	inf	...	inf	inf	inf
...
868	180.348	1.02853	1.04619	...	1.01343	1.02998	1.04742
869	180.548	1.03649	1.04644	...	0.98313	1.00312	1.01272

870 rows × 361 columns

Figura 3: Datos de barridos láser

¹Todos los comandos utilizados se pueden ver en el cuaderno del [repositorio](#)

2.2. Actividad 6

Mediante un script en python, utilizando los archivos antes mencionados, graficamos 3 diferentes barridos láser. En la figura 4 se observan 3 barridos de tiempos consecutivos, en este caso particular, superpuestos. Esto se debe a que se seleccionaron los 3 primeros barridos, en donde el robot no se encontraba en movimiento aún.

Es importante mencionar que el gráfico puede generar una confusión por la forma del barrido, pudiendo pensar que el sensor láser se encuentra en el punto $(-1;0,5)$ pero se debe recordar que el sensor barre los 360 grados, y este se ubica en el punto $(0,0)$ de su propio sistema de referencia. La forma del barrido se debe a que se encontraba una esquina del mapa a espaldas del robot.

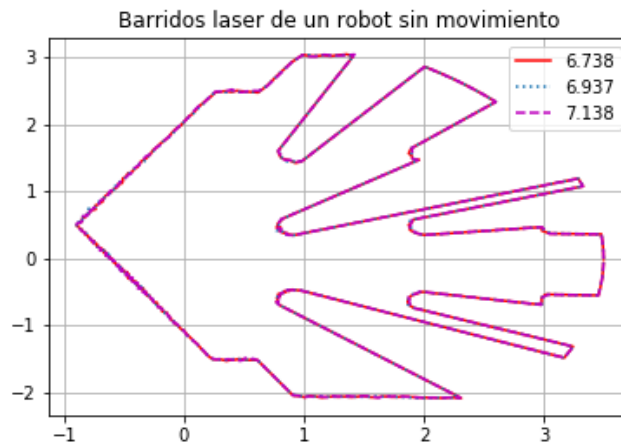


Figura 4: Barridos láser de robot estático

Eligiendo otros 3 barridos distribuidos entre todo el tiempo de simulación y, por ende, ya con el robot en movimiento se pueden observar los diferentes barridos de la figura 5 y en este caso, en cambio, se puede distinguir que el sensor láser se encuentra en el punto $(0,0)$ de su sistema de referencia ya que los 3 barridos se encuentran centrados en este punto.

2.3. Actividad 7-8

Como se puede observar en la figuras 2 y 3 los *timestamps* o tiempos de muestreo en la simulación no coinciden.

En la figura 6 podemos observar los puntos de muestreo para ambos sensores y los más cercanos. Los puntos azules corresponden a los tiempos de muestreo del sensor de barrido láser, los puntos en verde son los tiempos de muestreo del odometría, y los puntos en rojo son los tiempos de muestreo del sensor de odometría más cercanos a los tiempos de muestreo del láser.

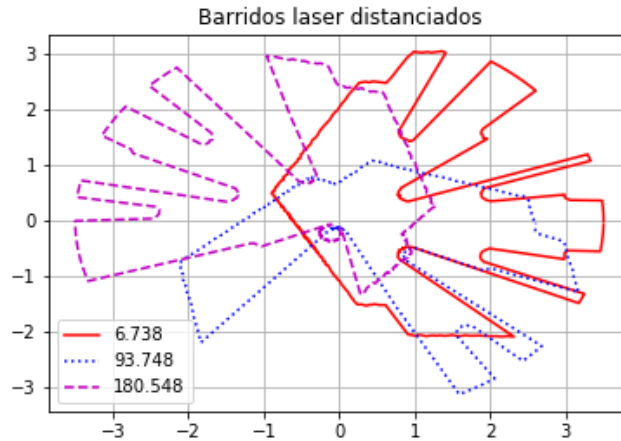


Figura 5: Barridos láser distanciados

El script que se utilizó para encontrar los tiempos de simulación más cercano entre los barridos del sensor láser y las muestras de odometría se encuentran en el cuaderno del [repositorio](#).

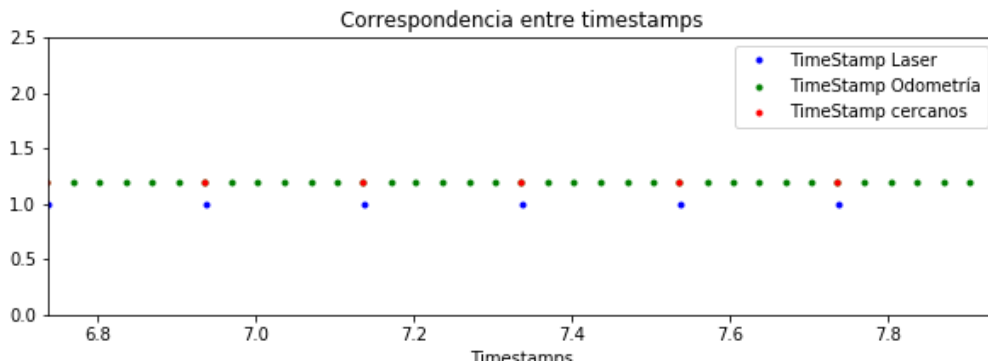


Figura 6: Correspondencia entre tiempos de muestreo

2.4. Actividad 9

Como se explicó anteriormente la matriz de transformación, que nos permite representar los puntos obtenidos del barrido láser (en coordenadas rectangulares) en el sistema de referencia de odometría queda representada por la expresión 2. Donde θ es el ángulo de rotación del sistema del láser con respecto al sistema de odometría (orientación del robot).

Sabiendo que la resolución del sensor de barrido láser es un punto cada grado, es decir toma 360 muestras en un barrido de 360° y que las 360 muestras se encuentran ordenadas en filas según su *timestamp*, transformamos cada punto en coordenadas polar del barrido a coordenadas rectangulares, y luego multiplicamos a cada uno de

estos por la matriz de transformación, para obtener la nube de puntos del sensor láser que representa al mapa, y se puede observar en la figura 7

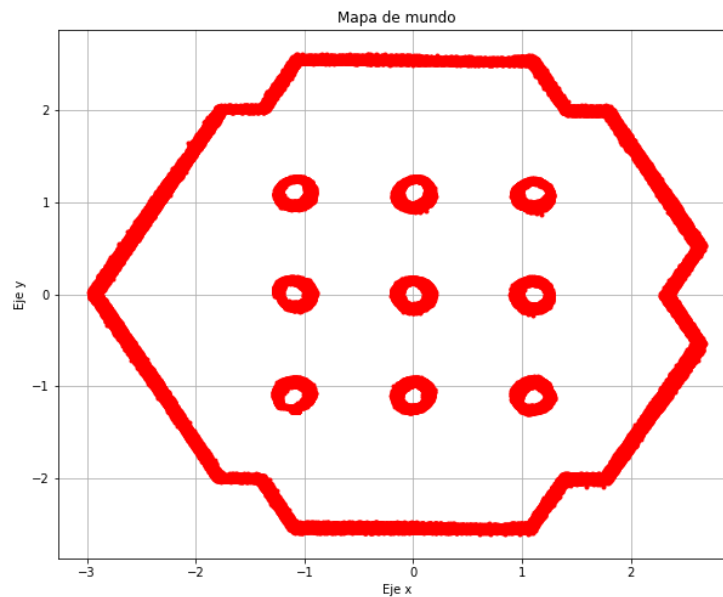


Figura 7: Mapa del entorno mundo

A este gráfico de puntos que forman el mapa, agregamos el gráfico de los puntos de las coordenadas x,y del robot, es decir, su trayectoria. Este se puede observar en la figura 8

Una animación de cómo el sensor láser va captando el mapa mientras el robot se desplaza se puede observar en el [repositorio](#).

3. Conclusión

Como puede observarse en la figura 6 tanto los los tiempos de muestreos de los sensores como la frecuencia de muestreo son diferentes, aproximadamente cada una muestra del sensor de barrido láser tenemos 6 muestras del sensor de odometría. Por esto, para la construcción del mapa se descartaron mediciones del sensor de odometría ya que se requería las mediciones más coincidentes en tiempo muestreo entre ambos sensores y solo estas son las necesarias para realizar la transformación de sistemas de referencias.

Para reconstruir el mapa es necesario la transformación entre los correspondientes sistemas de referencia. Si graficamos los puntos obtenidos del barrido láser en coordenadas rectangulares, solo vamos a observar un gráfico como el de la figura 5 sin

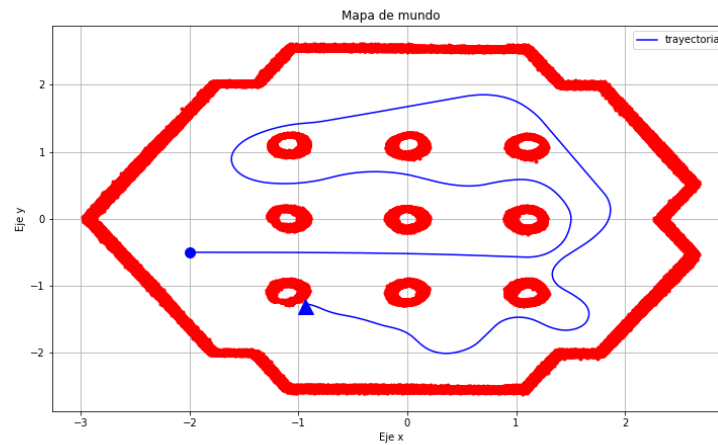


Figura 8: Mapa y trayectoria

importar en qué lugar del mapa este posicionado el robot, ya que este sensor solo arroja la distancia y ángulo de un punto medido desde su centro; y no se podría obtener un gráfico de los puntos del mapa referenciados al centro del mismo, como el de la figura 7

La matriz de rotación en este caso es la de la expresión 2 pero se debe tener en cuenta que en caso que existan más grados de libertad o ángulos de rotación la matriz estará conformada por la multiplicación de todas las matrices de rotación R .