

# Fundamentos de la Robótica Autónoma

## Laboratorio 1

### Objetivos

Reconocer las herramientas de trabajo en general. Utilizar ROS como plataforma de desarrollo del software de control. Interactuar con sensores y actuadores a través de los mecanismos estándares de ROS.

### Introducción

Para el desarrollo del laboratorio se utiliza:

- Hardware: una SBC Odroid N2, una placa Arduino, sensores de contraste, un sensor LIDAR LDS 01, motores AX-12 (y el módulo U2D2), alimentación (placa de alimentación para U2D2, batería LiPo y alarma), chasis Butiá 2 y elementos de interconexión.
- Software: paquete ROS básico, paquetes para interactuar con Ax-12 [1] y el LIDAR LDS 01 [2].

### Previos para trabajar con la placa

1. Cambiar nombre del host,
2. Chequear que el usuario usado pertenezca al grupo *dialout*.
3. Agregar reglas udev para que el nombre en el sistema de archivos sea único y razonable para el dispositivo conectado (e.i. laser, u2d2).

### Parte A – Interacción con sensores

1. Definir tres escenarios significativos a experimentar con el desempeño de los sensores.
2. Desarrollar un procedimiento o programa para obtener las lecturas de los sensores.
3. Elegir una de las posiciones de lectura y para cada escenario tomar una muestra de las lecturas generadas por el sensor. Deberá promediar el resultado habiendo descartado valores máximos y mínimos. Documentar según la siguiente tabla.

Rango operativo del sensor:

Condiciones del ambiente:

Cantidad de medidas a promediar:

Muestra	Ambiente <sup>1</sup>	Valor <sup>a</sup> máximo	Valor mínimo	Promedio	Desviación estándar

- Para las distintas condiciones ambientales, grafique (puede utilizar el script de gnuplot [6] dado) un histograma con los valores obtenidos en el experimento.

<sup>1</sup>Tipo e intensidad de luz ambiente

**Preguntas:**

1. ¿Cuál es la característica del ambiente que afecta más al resultado de este sensor? ¿Por qué?
2. Describa tres usos posibles para el sensor.
3. ¿Cuál es la frecuencia de muestreo del sensor? ¿Qué lo determina? Verifique con el comando `rostopic Hz`. Opcionalmente puede implementarse un servicio para ajustar dicha frecuencia.
4. En las partes anteriores realizó un promedio de los valores descartando el valor máximo y el mínimo.
  - i. ¿Por qué motivo se pidió que se descarten dichos valores?
  - ii. Desarrolle un nodo que re-publique la información en crudo del sensor utilizando el filtro pasa bajo dado en el teórico, para dos alphas diferentes.
  - iii. Grafique en una sola gráfica los datos sin filtrar y filtrados.
  - iv. Analice los resultados.

**Parte B – Interacción con actuadores**

1. Configure los motores adecuadamente utilizando la herramienta Dynamixel Wizard 2.0 [3]. Verifique que los valores de configuración del motor sean razonables (p.e identificador, temperatura y torque máximo, y rotación continua o ajuste angular).
2. Investigar sobre el uso y las capacidades de la biblioteca para controlar los motores AX-12
3. Ejecutar comandos o desarrollar un programa para controlar los motores AX-12.

**Parte C – Integración**

1. Ensamblar el robot usando el chasis de Butia 2, los motores AX-12 y el LIDAR.
2. Controlar los movimientos del robot con el teclado.
3. Programar al robot para que se acerque al objeto más cercano.
4. Publicar mensajes de odometría [4].
5. Programar el robot para que dibuje un cuadrado N veces. Intentar determinar el error. Observar el recorrido en rviz.

**Lenguaje de programación**

C++ [5] y Python [6].

**Instrucciones para configurar el ambiente de trabajo**

1. Instalar ROS. Se recomienda la versión Noetic.
2. Instalar GNU plot.
3. Instalar los paquetes .

**Fecha de entrega**

Jueves 28/04/2024 hasta las 23:30 hrs.

**Forma de entrega**

Se debe entregar lo pedido en cada parte de este laboratorio en formato electrónico a través del receptor que se habilitará en el EVA de la asignatura.

Consideraciones:

- Toda documentación (código fuente) debe entregarse en un archivo comprimido cuyo nombre debe seguir la siguiente nomenclatura: FRA-Lab1.zip.
- Cada parte deberá incluir una sección que indique los pasos a seguir para replicar su ejecución (p.e. nodos a levantar con sus parámetros y archivos launch).

## **Referencias**

[1] Robotis, "DYNAMIXEL Workbench", url:

[https://emanual.robotis.com/docs/en/software/dynamixel/dynamixel\\_workbench](https://emanual.robotis.com/docs/en/software/dynamixel/dynamixel_workbench), visitada marzo de 2022.

[2] Robotis, "Sensor LDS 01", url: [https://emanual.robotis.com/docs/en/platform/turtlebot3/appendix\\_lds\\_01](https://emanual.robotis.com/docs/en/platform/turtlebot3/appendix_lds_01), visitada marzo de 2022.

[3] Dynamixel Wizard 2.0, [https://emanual.robotis.com/docs/en/software/dynamixel/dynamixel\\_wizard2/](https://emanual.robotis.com/docs/en/software/dynamixel/dynamixel_wizard2/), visitada marzo de 2023.

[4] ROS wiki, "Publishing Odometry Information over ROS", url:

<http://wiki.ros.org/navigation/Tutorials/RobotSetup/Odom>, visitada marzo de 2022.

[4] Stroustrup, Bjarne. *The C++ Programming Language (Third ed.)*, 1997.

[5] Raúl González Duque, "Python PARA TODOS", Creative Commons.

[6] GNU Plot, "gnuplot homepage", url: <http://www.gnuplot.info/>, visitada marzo de 2021.