

Laboratorio 2

Evaluación de incertidumbre de un sensor

Ricardo E. Ramírez Heredia
Pedro-F Cardenas

2023-1S

Índice

1. Introducción	1
2. Consulta bibliográfica	2
2.1. Horario	2
3. Elementos y software	2
4. Lidar	3
4.1. Sensor HOKUYO	3
4.1.1. Preparación	3
4.1.2. Toma de datos	4
4.2. Sensor RPLIDAR	5
4.3. Procesamiento de datos y presentación de resultados	5
5. Sensor de ultrasonido	6
5.1. Preparación	6
5.2. Toma de datos	7
5.3. Procesamiento de datos y presentación de resultados	7
6. Sensores Lego	7

1. Introducción

Los sensores son elementos básicos para que los robots móviles cumplan con su función.

Los sensores tienen un conjunto de características técnicas, se clasifican en características estáticas y características dinámicas. Dentro de estas características o asociadas a ellas hay un conjunto de valores de error que conforman la

incertidumbre de medida.

En la medida que deseamos mejor desempeño de los robots, esos errores se hacen evidentes o cobran mayor importancia. Reconocer la existencia, magnitud y comportamiento de los errores nos permite hacer un control de esos errores mediante pre-procesamiento de señales o al menos identificar porqué el comportamiento de los robots es diferente al esperado.

2. Consulta bibliográfica

Hacer una consulta bibliográfica o en Internet y resolver las siguientes preguntas:

1. ¿Qué es el Vocabulario Internacional de Metrología (VIM)?
2. Según el VIM ¿Qué es?
 - exactitud de medida.
 - precisión de medida.
 - error de medida.
 - incertidumbre de medida.
3. Explique la diferencia entre un error sistemático y un error aleatorio.
4. De acuerdo a la teoría de estadística: ¿Qué es valor medio? ¿Cuáles magnitudes se usan para medir la dispersión de datos?

2.1. Horario

Programar horario para la práctica libre con el Ing. William Roa <awroag@una1.edu.co> en el laboratorio de Automatización de Máquinas.

3. Elementos y software

Elementos

- Computador
- Sensores LIDAR disponibles: [Lidar Hokuyo URG-04LX-UG01](#), con cable USB-mini y sensor RPLIDAR.
- Sensor de distancia por ultrasonido [HC-SR04](#).
- Cinta métrica.
- Graduator o transportador, escuadra de 45° y 60°.
- Microcontrolador Arduino con su cable de alimentación y comunicación.

- Protoboard
- Cables de conexión para protoboard.

Software

- Software Hokuyo. URG Benri data viewing tool [Download](#).
- Matlab [Download](#).
- Arduino IDE [Download](#).
- Lego EV3 [Download](#).

4. Lidar

Si dispone del Sensor Hokuyo continúe a la sección [4.1](#) o si dispone del Sensor RPLIDAR continúe a la sección [4.2](#)

En esta parte del laboratorio se hace uso del sensor Lidar. ¹

4.1. Sensor HOKUYO

4.1.1. Preparación

- **INICIALMENTE NO CONECTAR EL SENSOR AL PC.**
- Capture su área de trabajo con una foto y tome las dimensiones totales aproximadas.
- Abrir MATLAB.
- Desde MATLAB verificar con la instrucción `instrfind` que no hay puertos abiertos, la respuesta en la ventana de comandos debe ser:

```
ans=  
[ ]
```

Recomendación: Si aparecen instrumentos asociados es preferible cerrar el software, desconectar el sensor, volver a abrir MATLAB y reconectar el sensor.

- En la carpeta de trabajo con MATLAB copiar la carpeta [Lab02 - Matlab](#)
- Abrir el Administrador de dispositivos (Windows) y conecte el sensor a un puerto USB, observe los cambios en la carpeta Puertos COM y LPT. Registre en número de puerto asignado al sensor.

¹[Vídeo Lidar](#)

- Abra desde MATLAB el archivo SetupLidar.m y modifique la línea 6 con el número de puerto COM correspondiente.
- Organice su espacio de trabajo de manera que pueda colocar el sensor en 3 diferentes posturas en el área a mapear como aparece en la figura 1
- Establezca un punto de su área de trabajo y definalo cvomo origen de coordenadas y defina direcciones **X** e **Y**.
- Corra el algoritmo SetupLidar.m.

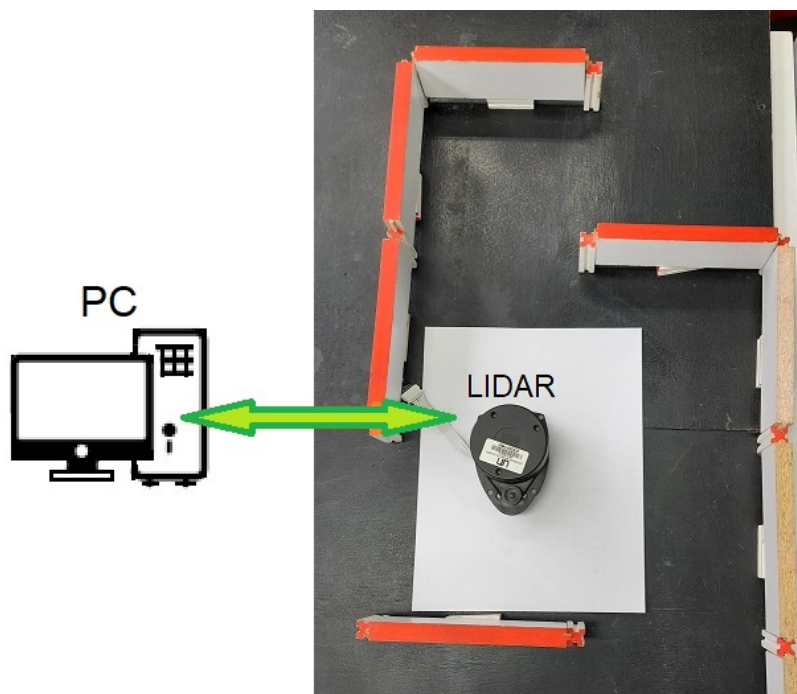


Figura 1: Montaje Lidar

4.1.2. Toma de datos

- Abra el algoritmo RunLidar2.m
- Modifique el algoritmo para que haga 3 escaneos con separación de tiempo de 1 a 3 segundos entre ellos.
- Coloque el sensor en la pose 1 mida y guarde los datos de esa pose $[x, y, \theta]_1$.
- Corra el algoritmo RunLidar2.

- Modifique el algoritmo en las líneas 12 y 16 para que genere una variable LidarSet2.
- Coloque el sensor en la pose 2 mida y guarde los datos de esa pose $[x, y, \theta]_2$.
- Corra el algoritmo RunLidar2.
- Modifique el algoritmo en las líneas 12 y 16 para que genere una variable LidarSet3.
- Coloque el sensor en la pose 3 mida y guarde los datos de esa pose $[x, y, \theta]_3$.
- Corra el algoritmo RunLidar2.
- Guarde los datos de las poses y de los archivos para posterior procesamiento.

4.2. Sensor RPLIDAR

- Instale, ejecute el software [URG Benri data viewing tool](#) .
- Realice la configuración del sensor y compuebe el funcionamiento adecuado del Lidar.
- Realice una preparación y toma de datos similar a la realizada para el sensor Hokuyo. Utilice los algoritmos disponibles en la carpeta [RPLIDAR](#)

4.3. Procesamiento de datos y presentación de resultados

- En el informe presente la foto de su área de trabajo modificada con las dimensiones totales y la pose aproximada del sensor en cada conjunto de medidas.
- Para cada pose informe los datos $[x, y, \theta]_i$ y presente los datos capturados con alguno de los métodos **plot(scan)** presentados en la página de ayuda de MATLAB: **lidarScan Create object for storing 2-D lidar scan**.
- Para cada pose presente los datos en forma de mapa de ocupación. Ayuda de MATLAB **occupancyMap Create 2-D occupancy map** ejemplo **Insert Laser Scans into Occupancy Map**.
- Usando los datos capturados en las 3 poses construya un mapa de ocupación de su área de trabajo, use como referencia la ayuda de MATLAB: **buildMap Build occupancy map from lidar scans**.
- Realice un análisis de los errores y fuentes de error evidenciados en la construcción del mapa en el punto anterior.

5. Sensor de ultrasonido

5.1. Preparación

- En los sitios referenciados en la carpeta Material de laboratorio 2 identifique la forma de conectar el ARDUINO y el sensor [HC-SR04](#) y la forma de conexión a su computador. Haga las conexiones correspondientes. Tenga en cuenta los números de pines del ARDUINO a los cuales conectó los pines de *trigger* y de *echo* del HCSR04.
- Abra el IDE de Arduino, abra el archivo usound3.ino, modifique las instrucciones

```
const int pinecho = 11;  
const int pintrigger = 12;
```

para que correspondan con los pines del Arduino usados.

- Compile y cargue el algoritmo al ARDUINO.
- En la IDE de Arduino, abra la ventana de terminal y verifique que se están leyendo datos y se están registrando. Cierre la ventana de terminal.
- Instalar el montaje experimental similar mostrado en la figura 2.

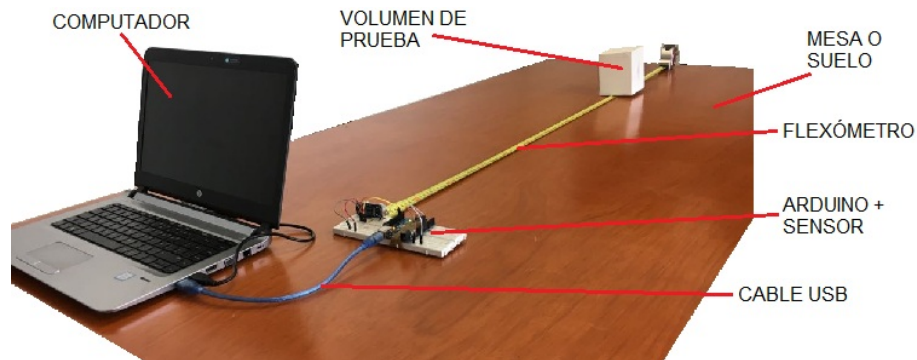


Figura 2: Montaje sensor ultrasonido

- En MATLAB abra el archivo ultrasound3.m.
- En la ventana de comando use la instrucción `serialportlist` para identificar el puerto asignado al ARDUINO.
- Modifique en el archivo la instrucción `port=serialport()` la configuración de comunicación para que el puerto COM corresponda con el que está asignado para el Arduino. El programa recibe datos del puerto serie y los almacena en la variable "dist".

- Coloque un objeto (caja de cartón, tabla pequeña o similar), que se constituya en el objeto con respecto al que se va a medir la distancia.

5.2. Toma de datos

- Coloque el objeto a una distancia de 1 m del sensor, verifique la distancia con la cinta métrica y guarde el dato.
- Corra el algoritmo `ultrasound3` y guarde los datos de distancia.
- Repita el procedimiento para otras 3 posiciones del objeto con distancias al sensor entre 1 y 2,5 m.
- Para cada distancia corra el algoritmo `ultrasound3.m` y guarde los datos en un archivo convenientemente identificado para su posterior procesamiento.

5.3. Procesamiento de datos y presentación de resultados

- Para cada grupo de datos calcule la distancia media, la desviación estándar, el error absoluto y el error relativo respecto a la medida de distancia con flexómetro.
- En el informe presente la gráfica de distancia contra índice de muestra.
- Para cada distancia presente un histograma de datos.
- Presente gráficas de comportamiento de desviación estándar y errores absoluto y relativo en función de la distancia media.
- Haga los análisis y conclusiones pertinentes.

6. Sensores Lego

Realizar una evaluación simple de la incertidumbre de los sensores y actuadores que son utilizados en los kits de LEGO.

1. Programe el LEGO EV3 para que realice arranque, desplazamiento de 100 cm. y parada con el robot. Programe la velocidad de desplazamiento al 30 % de la máxima. Tome los datos inicial y final de desplazamiento usando:
 - a) Encoder sensor de rueda.
 - b) Sensor de Ultrasonido del EV3.
 - c) Cinta métrica.
2. Repita el procedimiento anterior , pero ahora con la velocidad al 100 % de la máxima.
3. Realice un análisis comparativo de las medidas y calcule los errores de desplazamiento tomando como patrón la cinta métrica.

4. Construya un algoritmo para girar la rueda del EV3 una vuelta en intervalos 30° y 45° . Implemente un setup experimental para medir los ángulos girados con un método externo (transportador, apuntador láser y cinta métrica, otro). Repita la medición.
5. Realice un análisis comparativo de las medidas y calcule los errores de desplazamiento tomando como patrón el método de medida externo.