

PRACTICA 1

SENSORES Y ACTUADORES EN ROS

Sergio Sanz Cacho – 03201575K

Carlos Tejeda Martínez – 03148129G

Indice

[1.Caracterización de los sensores de odometría 2](#_Toc34906614)

[2.Caracterización de los sensores de distancia ultrasónicos 2](#_Toc34906615)

[3.Sensores de distancia láser 8](#_Toc34906616)

[4.Actuadores en ROS 11](#_Toc34906617)

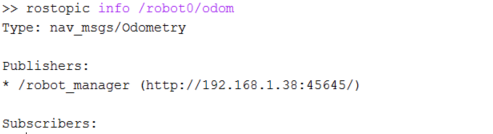
# 1.Caracterización de los sensores de odometría

1. Indique y describa la información que nos ofrece el mensaje disponible en el topic **odom**. Muestre algún ejemplo de captura.

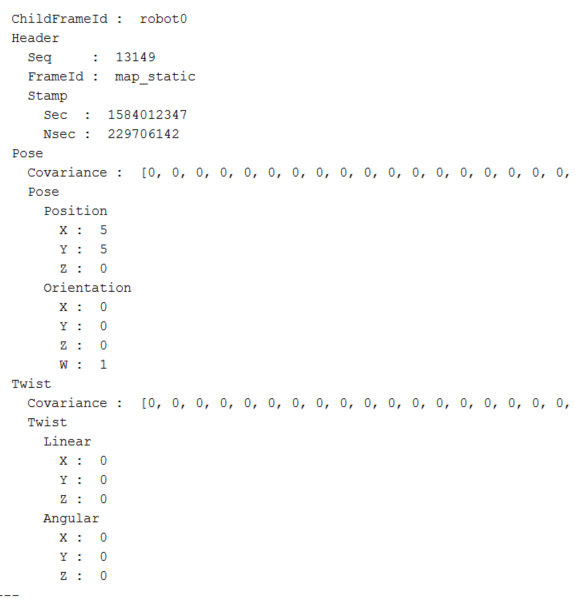
La información ofrecida por el topic de odom son los datos listados a continuación:

* ChildFrameId
* Header: Nos da la información identificatoria del sonar y el instante de tiempo
* Pose: Nos indica todo lo referente a la posición que esta ocupando en el plano y la orientación en este, en el momento de consulta del rossubscriber.
* Twist: Nos indica todo lo referente a la modificación de la posición que esta ocupando en el plano y la orientación en este, al mandarse una orden de movimiento.

En la captura se indica en el apartado “Type”:



Y a continuación se muestra la información ofrecida por el subscriber:



1. Mida la resolución (**q**) de la odometría lineal y angular máximas 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(m/s) | Ω(rad/s) | q\_lineal(m) | | | q\_angular(r) | | | |
| Observado | | Calculado | Observado | | | Calculado |
| 0.1 | 0 | 0.1095 | 0.1136 | | - | | - | |
| 0.3 | 0 | 0.3294 | 0.2754 | | - | | - | |
| 0.5 | 0 | 0.5459 | 0.5611 | | - | | - | |
| 0.7 | 0 | 0.7696 | 0.7815 | | - | | - | |
| 0.9 | 0 | 0.9893 | 1.0572 | | - | | - | |
| 0 | 0.3 | - | - | | 0.0809 | |  | |
| 0 | 0.7 | - | - | | 0.11380 | | 0.19254 | |
| 0 | 0.9 | - | - | |  |  | | |

1. Mida la resolución (**q**) de la odometría lineal y angular máximas con las diferentes combinaciones de velocidades propuestas en el robot real. Construya una tabla como la que se muestra a continuación con las 8 combinaciones propuestas

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| V(m/s) | Ω(rad/s) | q\_lineal(m) | | | q\_angular(r) | | | |
| Observado | | Calculado | Observado | | | Calculado |
| 0.1 | 0 |  |  | | - | | - | |
| 0.3 | 0 |  |  | | - | | - | |
| 0.5 | 0 |  |  | | - | | - | |
| 0.7 | 0 |  |  | | - | | - | |
| 0.9 | 0 |  |  | | - | | - | |
| 0 | 0.3 | - | - | |  | |  | |
| 0 | 0.7 | - | - | |  | |  | |
| 0 | 0.9 | - | - | |  |  | | |

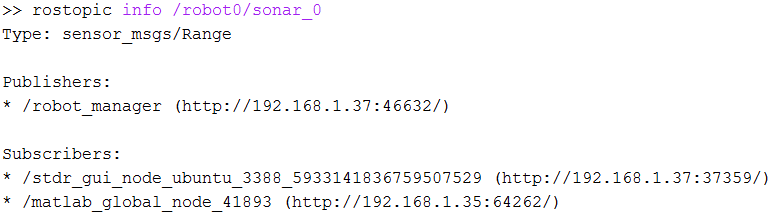
# 2.Caracterización de los sensores de distancia ultrasónicos

1. Indique y describa la información que nos ofrece el mensaje disponible en el topic **sonar**. Muestre algún ejemplo de captura.

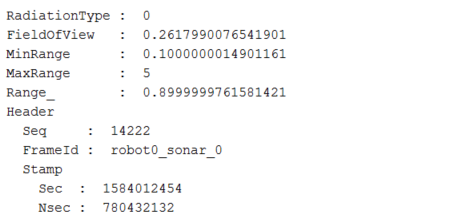
La información ofrecida por el topic de cualquier sonar son los datos listados a continuación:

* RadiationType
* FieldOfView: Nos indica el ángulo de visión del haz de ultrasonidos emitido por el sonar.
* MinRange: Nos indica la distancia mínima que puede medir eficazmente.
* MaxRange: Nos indica la distancia máxima que puede medir eficazmente.
* Range: Nos indica a la distancia donde se ha detectado un punto de colisión de los ultrasonidos.
* Header: Nos da la información identificatoria del sonar y el instante de tiempo

En la captura se indica en el apartado “Type”:



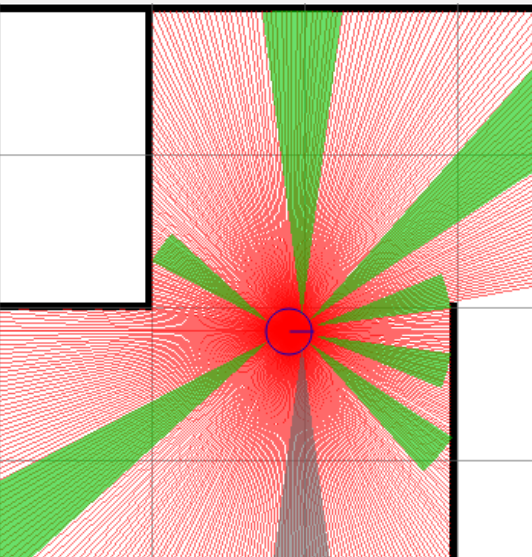
Y a continuación se muestra la información ofrecida por el subscriber:



1. En el simulador STDR, posicione el robot de tal forma que existiera una distancia de 2m de uno de los sensores sónar concretos del robot. ¿Qué posición y sónar se ha elegido?

Hemos seleccionado el sonar nº0 (“/robot0/sonar\_0”) y hemos posicionado el robot en las coordenadas [X:6.91, Y:7.85, Orientación: 0]

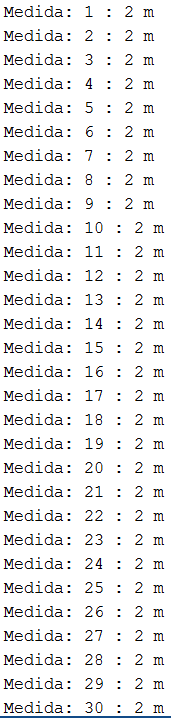
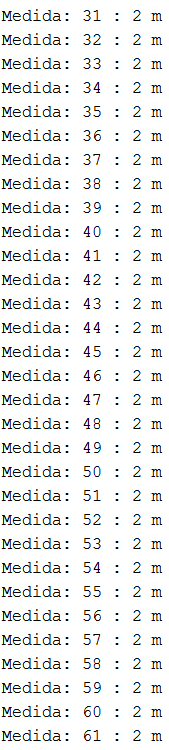
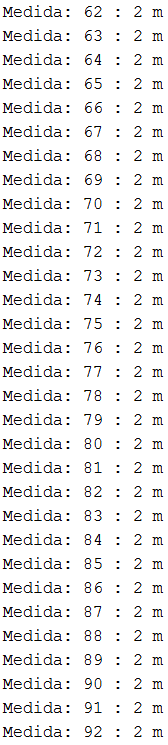
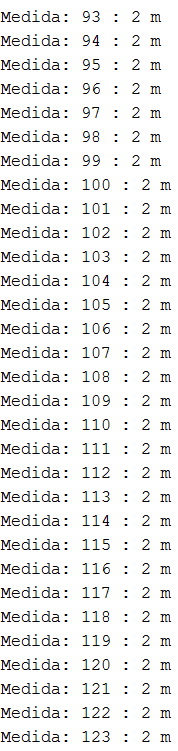
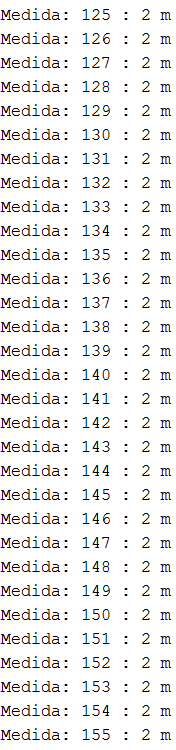


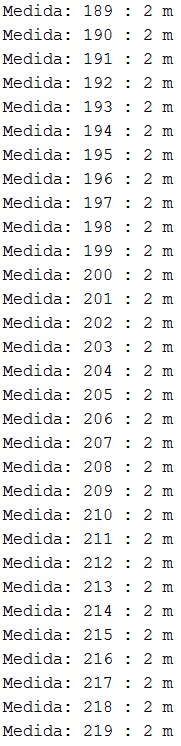
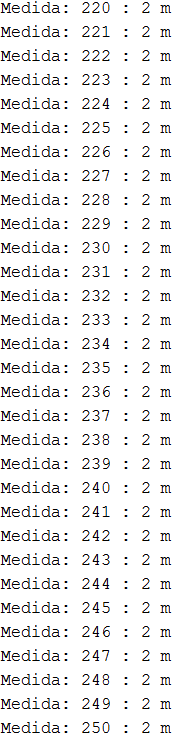
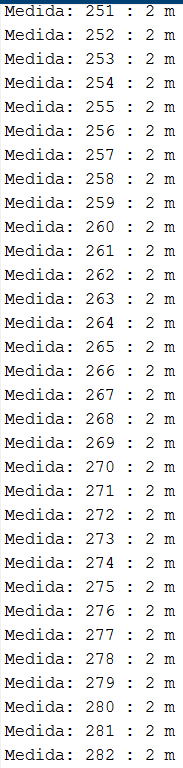
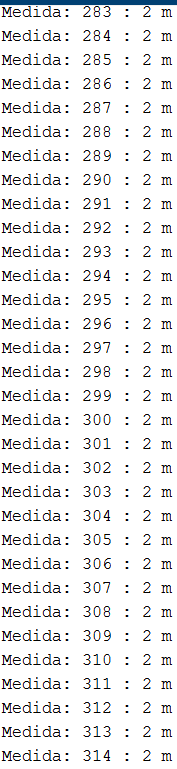
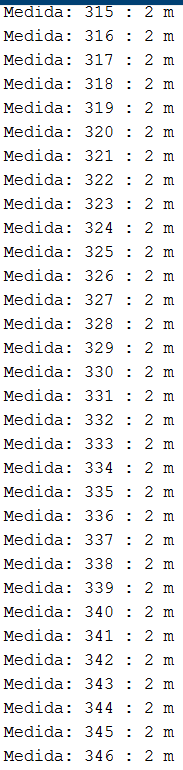
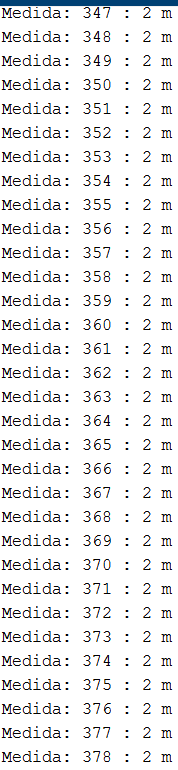


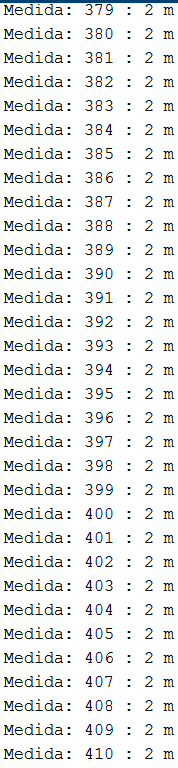
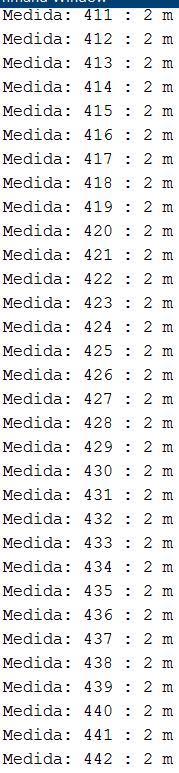
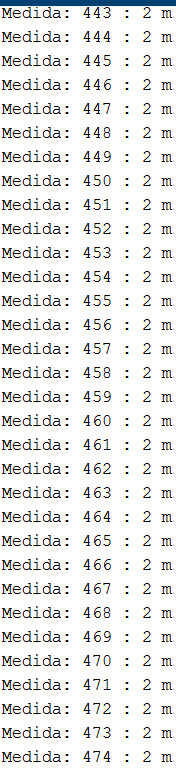
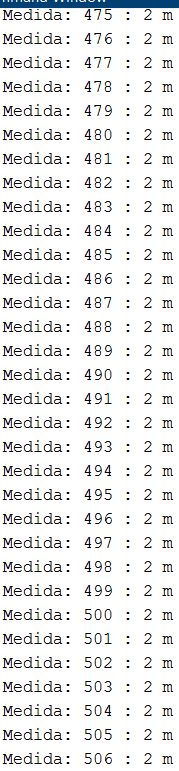
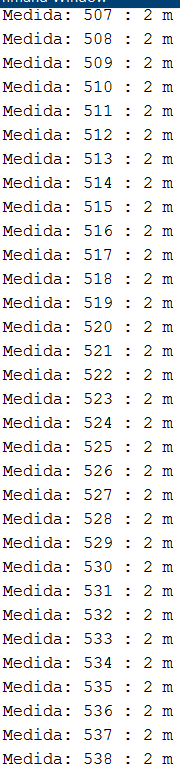
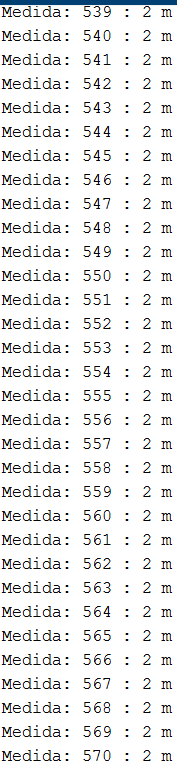
1. Obtenga 1000 medidas de distancia del sensor sónar elegido y dibuje en una gráfica la distancia medida. ¿Son estables las medidas? ¿Hay ruido en la medida? En caso afirmativo, calcule el valor máximo, medio y la varianza del ruido.

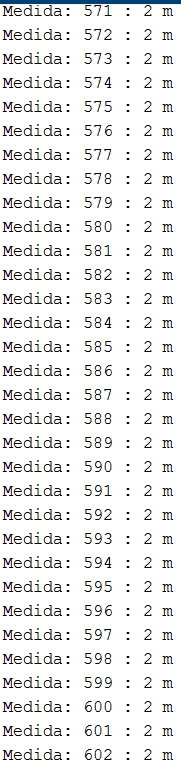
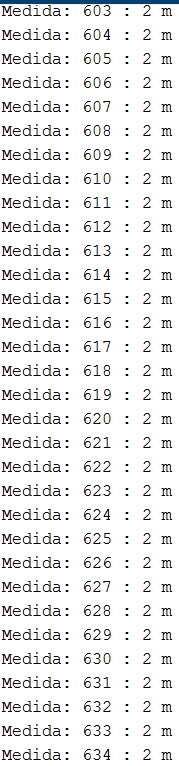
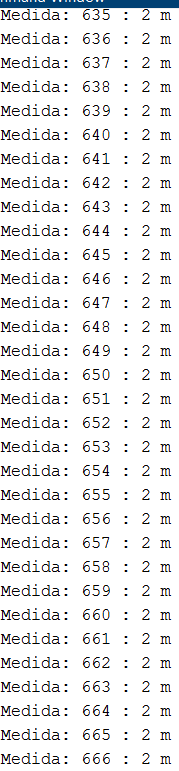
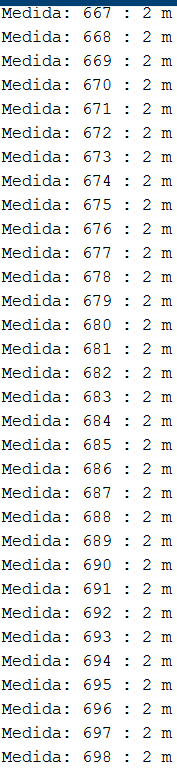
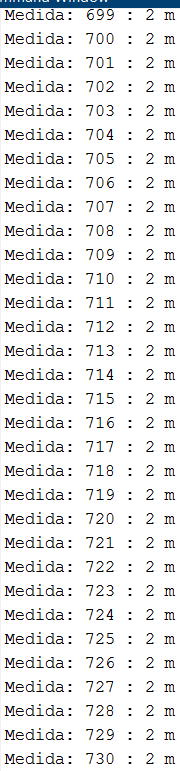
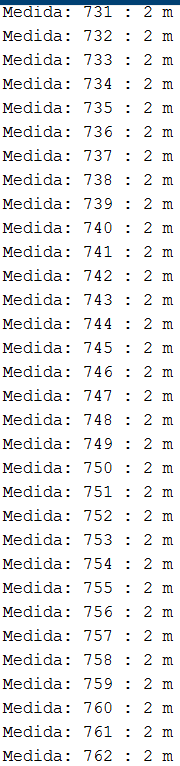
Al realizarse la medición en una simulación las medidas son estables y todas coinciden en el mismo valor (2m).

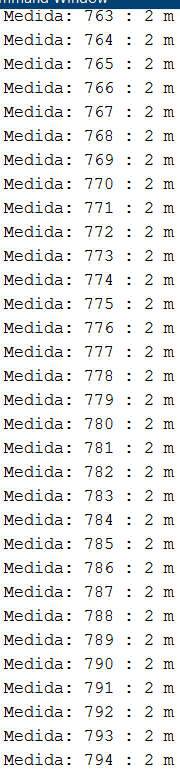
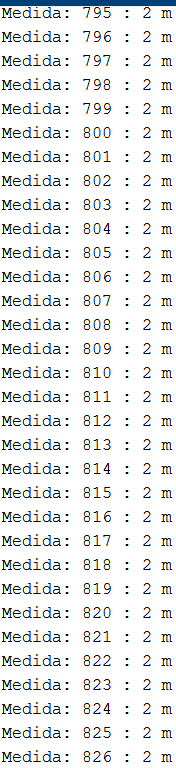
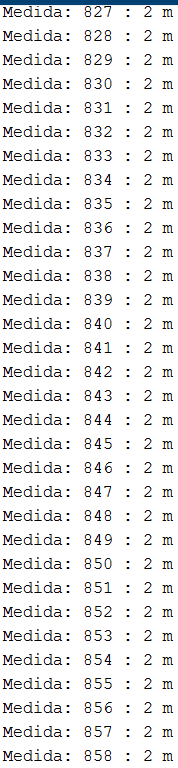
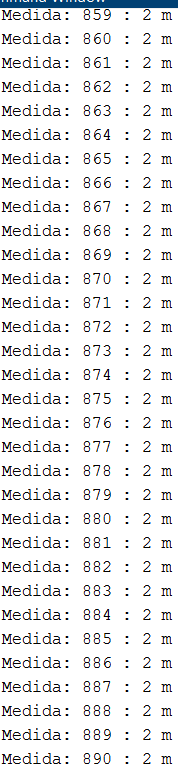
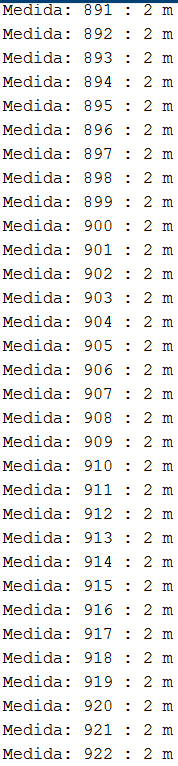
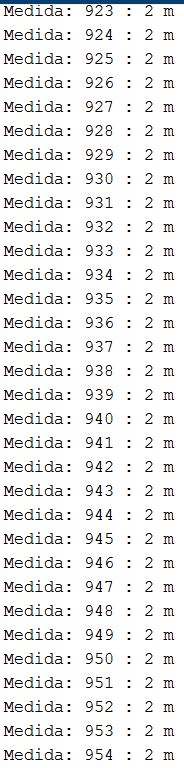
¿ruido del sonar que pone en el stdr?

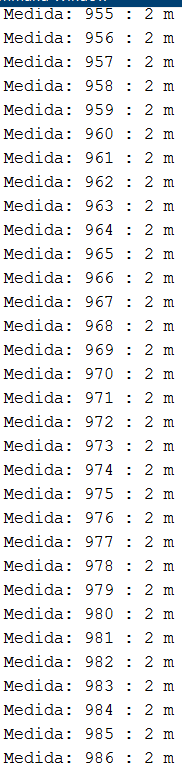
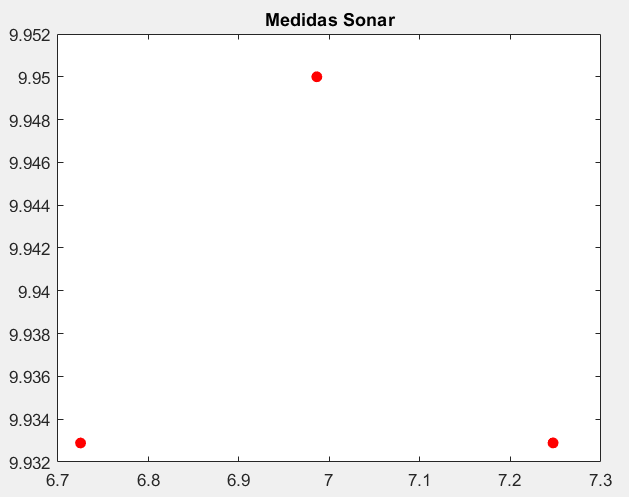
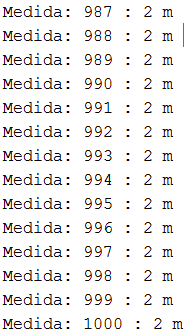
     

1. Implemente un filtro media móvil con los últimos 5 valores de distancia y dibuje en una gráfica. ¿Son más estables las medidas que en el caso anterior? ¿Sería útil este método si el robot está en movimiento en lugar de permanecer estático?

GRAFICA MEDIA MOVIL

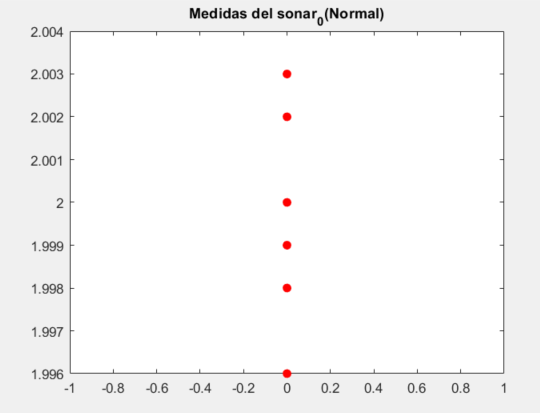
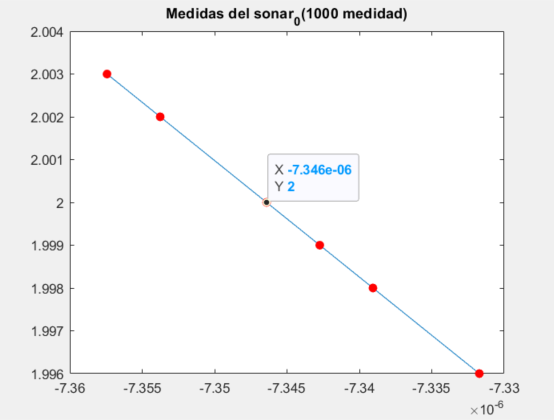
Las medidas van a ser igual de estables debido a que en la simulación no hay ruido evidente y todos los resultados de las medidas son iguales.

Este método no sería funcional ni útil en caso de que el robot este en movimiento ya que en el algoritmo desarrollado para el control y toma de medidas del robot no hemos contemplado este caso, obteniendo datos erróneos.

1. Repita pasos 2 – 4 con el robot real.

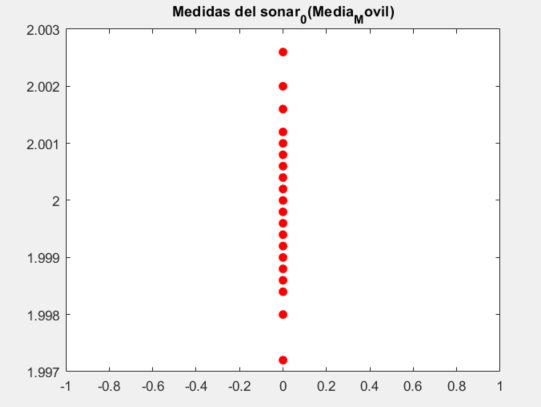
Las 1000 medidas tomadas por el sonar con el robot real nos dan los siguientes conjuntos de datos representados a continuación:

La primera muestra los puntos x y la segunda los puntos y



Al realizarse en con el robot real, ya sea por el entorno donde probamos el sonar o por dificultades del propio sonar, las medidas, aunque están agrupadas entorno un punto, se puede apreciar ruido entre medidas.

La media móvil realizada a partir de los datos anteriores resulta así:



Las medidas son más estables ya que mediante el filtro de la ventana móvil agrupamos resultados y obtenemos puntos medios, más próximos reduciendo el ruido aparente.

Este método no sería funcional ni útil en caso de que el robot este en movimiento ya que en el algoritmo desarrollado para el control y toma de medidas del robot no hemos contemplado este caso, obteniendo datos erróneos.

1. 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 podrían resultar útiles 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 **función de calidad** para la obtención de dichas paredes empleando, por ejemplo, la relación de entre las diferentes pendientes.

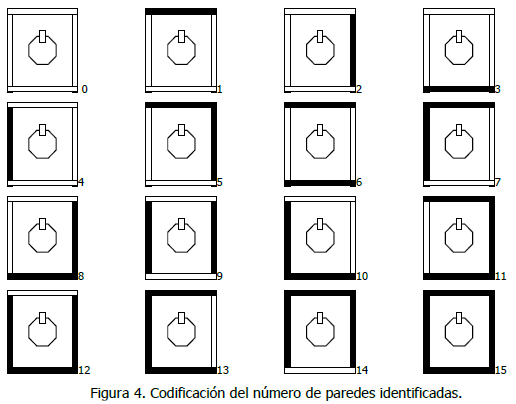
A la hora de detectar paredes, enfocándonos en conseguir resultados en el ejercicio inmediato a este también, hemos tomado como referencia los ejes de coordenadas y abscisas que pasan por el centro del robot.

Debido a ello, considerando el caso de partida del robot con orientación 0 radianes y con el robot estático, nos basamos en el uso de los sonares 0 y 5 para obtener información del eje y, mientras que usamos los sonares 2,3,6 y 7 para obtener información del eje x.

En el caso del eje y, consideraremos que hay pared cuando los sonares 0 o 5 tengan un punto de colisión a menos de 2m. Mientras tanto en el eje x comprobaremos los pares de sonares 2, 3 y 6,7 para comprobar la existencia de una pared completa. En estos casos comparamos los puntos más cercanos al eje x de cada sonar con el de su pareja y si tienen un punto x parecido dentro del error asumido, se considera la existencia de pared.

El error asumido en las comparaciones de coordenadas es de 6cm después de diversas pruebas de medición con sonares.

1. Diseñe una **función** que indique, mediante un código, el número de paredes que se encuentra el robot en sus laterales. La codificación que se puede emplear para proporcionar la salida es la que se muestra en la Figura 4. Además, indique el **grado de confianza** basado en la función de calidad definida en el apartado anterior.



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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Amigobot/STDR | Combinación real | Combinación identificada | | | | | Grado confianza |
| Amigobot | 1 | - | - | - | - | - | 0% |
| STDR | 1 | 1 | 1 | 1 | 1 | 1 | 100% |
| Amigobot | 2 | - | - | - | - | - | 0% |
| STDR | 2 | 8 | 2 | 2 | 2 | 2 | 80% |
| Amigobot | 3 | - | - | - | - | - | 0% |
| STDR | 3 | 3 | 3 | 3 | 3 | 3 | 100% |
| Amigobot | 4 | - | - | - | - | - | 0% |
| STDR | 4 | 4 | 4 | 4 | 4 | 4 | 100% |
| Amigobot | 5 | - | - | - | - | - | 0% |
| STDR | 5 | 5 | 5 | 5 | 5 | 5 | 100% |
| Amigobot | 6 | - | - | - | - | - | 0% |
| STDR | 6 | 6 | 6 | 6 | 6 | 6 | 100% |
| Amigobot | 7 | - | - | - | - | - | 0% |
| STDR | 7 | 7 | 7 | 7 | 7 | 7 | 100% |
| Amigobot | 8 | - | - | - | - | - | 0% |
| STDR | 8 | 8 | 8 | 2 | 8 | 8 | 80% |
| Amigobot | 9 | - | - | - | - | - | 0% |
| STDR | 9 | 9 | 9 | 9 | 9 | 9 | 100% |
| Amigobot | 10 | - | - | - | - | - | 0% |
| STDR | 10 | 10 | 10 | 4 | 10 | 10 | 80% |
| Amigobot | 11 | - | - | - | - | - | 0% |
| STDR | 11 | 11 | 11 | 11 | 11 | 11 | 100% |
| Amigobot | 12 | - | - | - | - | - | 0% |
| STDR | 12 | 12 | 12 | 12 | 9 | 12 | 80% |
| Amigobot | 13 | - | - | - | - | - | 0% |
| STDR | 13 | 13 | 13 | 13 | 7 | 13 | 80% |
| Amigobot | 14 | - | - | - | - | - | 0% |
| STDR | 14 | 14 | 14 | 14 | 14 | 14 | 100% |
| Amigobot | 15 | - | - | - | - | - | 0% |
| STDR | 15 | 15 | 15 | 14 | 14 | 15 | 60% |

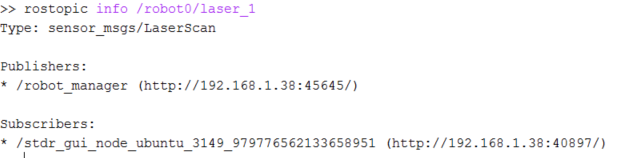
# 3.Sensores de distancia láser

1. Indique y describa la información que nos ofrece el mensaje disponible en el topic **laser**. Muestre algún ejemplo de captura.

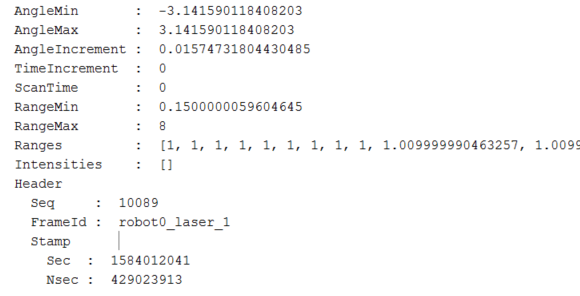
La información ofrecida por el topic láser son los datos listados a continuación:

* AngleMin: Nos indica el ángulo mínimo que mide el láser eficazmente.
* AngleMax: Nos indica el ángulo máximo que mide el láser eficazmente.
* AngleIncrement: Nos indica la distancia de separación entre emisión y emisión de un rayo.
* TimeIncrement
* ScanTime
* RangeMin: Nos indica la distancia mínima que puede medir eficazmente.
* RangeMax: Nos indica la distancia máxima que puede medir eficazmente.
* Ranges: Nos indica el conjunto de distancias tomadas por las colisiones de los rayos emitidos, siguiendo el orden de desde el primer rayo emitido (con el ángulo mínimo) hasta el último rayo emitido (con el ángulo máximo).
* Intensities
* Header: Nos da la información identificatoria del sonar y el instante de tiempo.

En la captura se indica en el apartado “Type”:



Y a continuación se muestra la información ofrecida por el subscriber:



1. Repita los pasos 2 – 4 y 6 – 8 del apartado anterior con el sensor láser

MIAU

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Amigobot/STDR | Combinación real | Combinación identificada | | | | | Grado confianza |
| Amigobot | 1 | 1 | 1 | 0 | 1 | 1 | 80% |
| STDR | 1 | 1 | 1 | 1 | 1 | 1 | 100% |
| Amigobot | 2 | 2 | 2 | 2 | 2 | 2 | 100% |
| STDR | 2 | 2 | 2 | 2 | 2 | 2 | 100% |
| Amigobot | 3 | 3 | 3 | 0 | 0 | 3 | 60% |
| STDR | 3 | 3 | 3 | 3 | 3 | 3 | 100% |
| Amigobot | 4 | 4 | 4 | 4 | 4 | 4 | 100% |
| STDR | 4 | 4 | 4 | 4 | 4 | 4 | 100% |
| Amigobot | 5 | 5 | 5 | 5 | 5 | 2 | 80% |
| STDR | 5 | 5 | 5 | 5 | 5 | 5 | 100% |
| Amigobot | 6 | 6 | 6 | 1 | 1 | 6 | 60% |
| STDR | 6 | 6 | 6 | 6 | 6 | 6 | 100% |
| Amigobot | 7 | 7 | 7 | 7 | 7 | 4 | 80% |
| STDR | 7 | 7 | 7 | 7 | 7 | 7 | 100% |
| Amigobot | 8 | 8 | 8 | 8 | 2 | 8 | 80% |
| STDR | 8 | 8 | 8 | 8 | 8 | 8 | 100% |
| Amigobot | 9 | 9 | 9 | 4 | 9 | 9 | 80% |
| STDR | 9 | 9 | 9 | 9 | 9 | 9 | 100% |
| Amigobot | 10 | 10 | 3 | 3 | 10 | 3 | 40% |
| STDR | 10 | 10 | 10 | 10 | 10 | 10 | 100% |
| Amigobot | 11 | 3 | 8 | 5 | 1 | - | 0% |
| STDR | 11 | 11 | 11 | 11 | 11 | 11 | 100% |
| Amigobot | 12 | 12 | 10 | 12 | 12 | 12 | 80% |
| STDR | 12 | 12 | 12 | 12 | 12 | 12 | 100% |
| Amigobot | 13 | 13 | 13 | 3 | 10 | 13 | 60% |
| STDR | 13 | 13 | 13 | 13 | 13 | 13 | 100% |
| Amigobot | 14 | 7 | 14 | 14 | 14 | 14 | 80% |
| STDR | 14 | 14 | 14 | 14 | 14 | 14 | 100% |
| Amigobot | 15 | - | - | - | - | - | 0% |
| STDR | 15 | 15 | 15 | 15 | 15 | 15 | 100% |

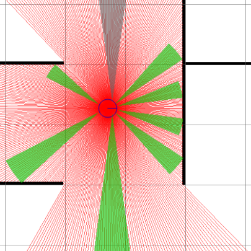
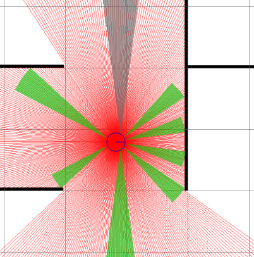
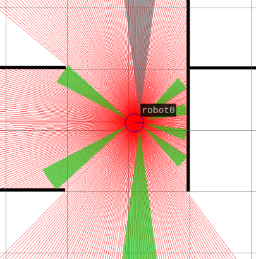
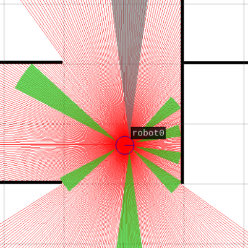
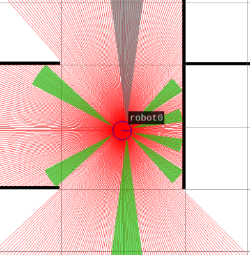
1. ¿Tiene más o menos ruido que el sensor sónar?

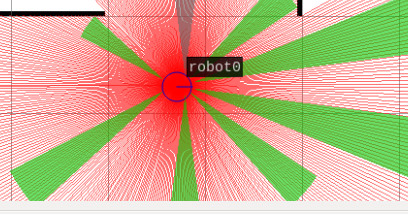
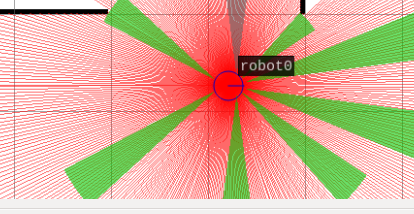
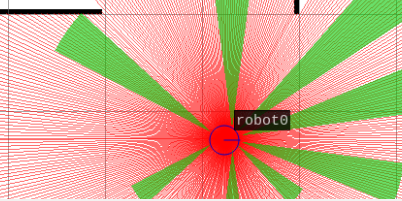
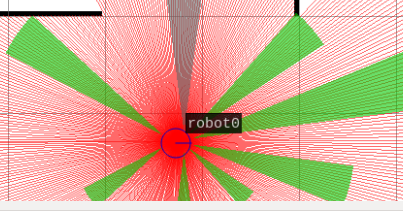
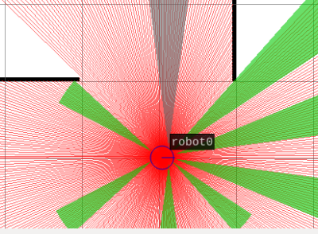
Tiene menos ruido ya que al tener tantos puntos podemos obtener conclusiones del entorno de manera más precisa, a diferencia con los sonares, que obtenemos una distancia de choque y teniendo que tomar en cuenta cualquier punto posible del haz emitido por el sonar.

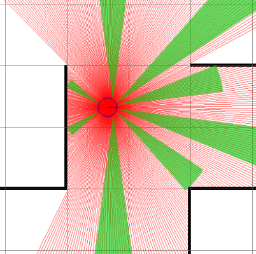
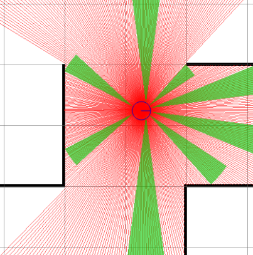
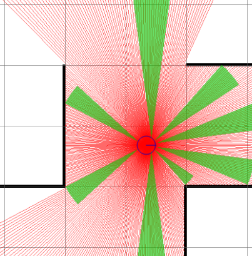
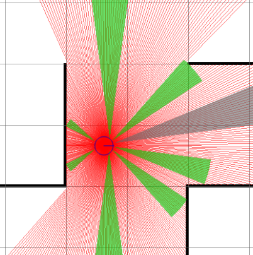
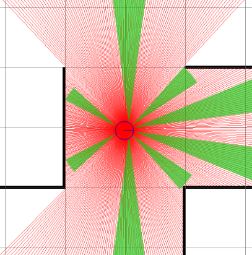
1. ¿Cómo es el grado de confianza empleando este sensor?

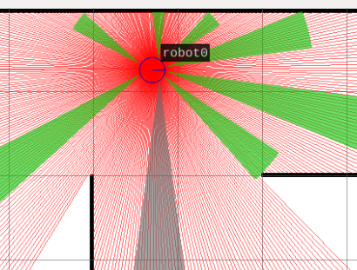
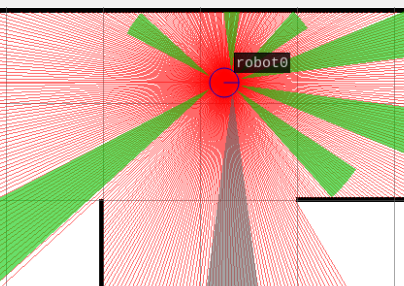
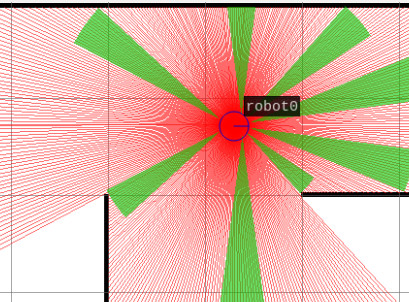
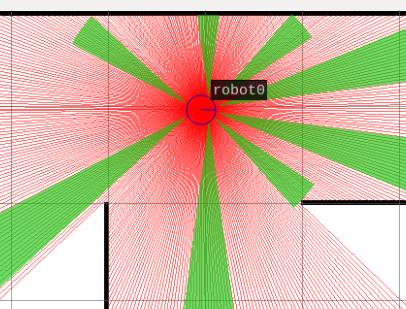
En la simulación el grado de confianza es del 100% debido a que todas las pruebas planteadas en el simulador han tenido un resultado positivo, mientras que en el robot real este grado de confianza se reduce al 65.3% debido a que la mayoría de pruebas el resultado general ha sido positivo, pero también hemos registrado errores.

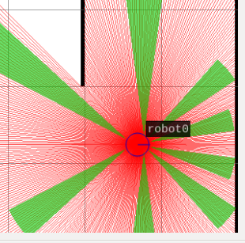
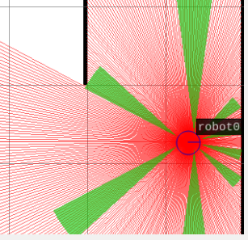
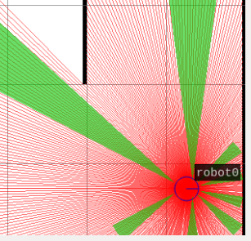
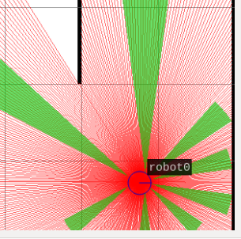
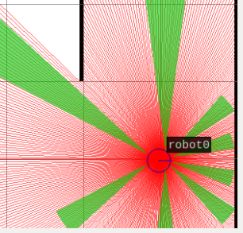
TOMA DE MUESTRAS STDR

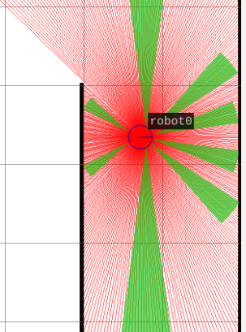
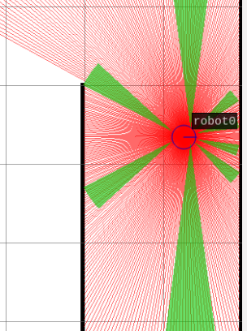
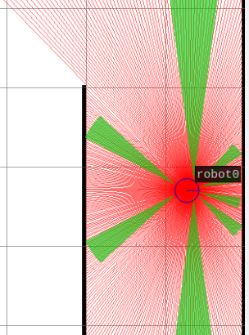
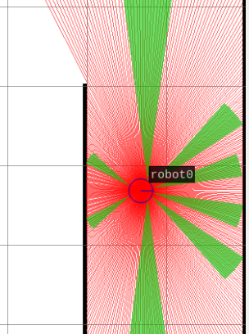
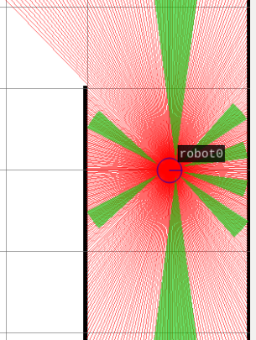
(1)

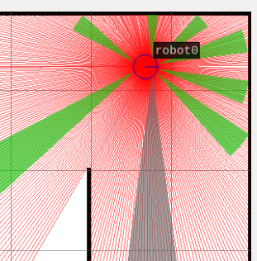
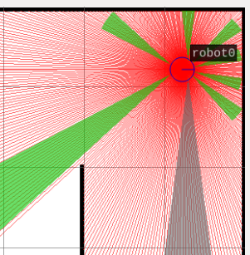
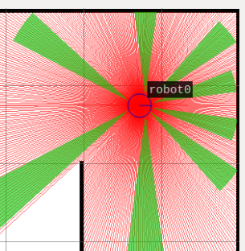
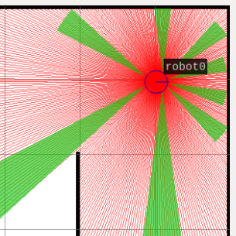
(2)

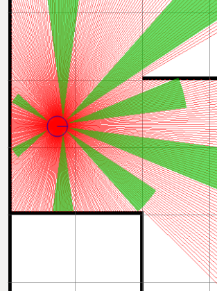
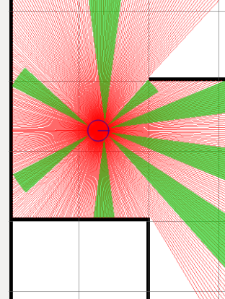
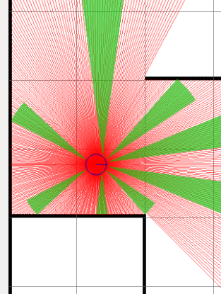
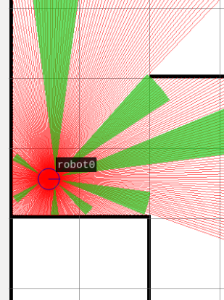
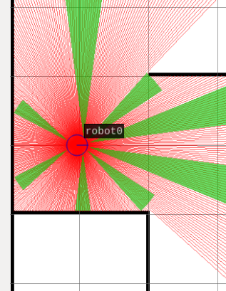
(3)

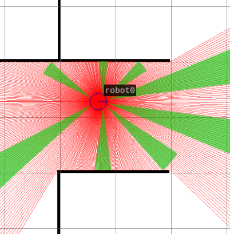
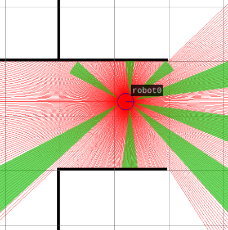
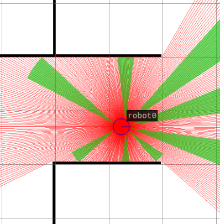
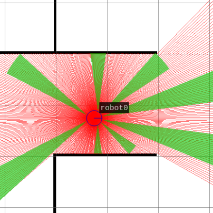
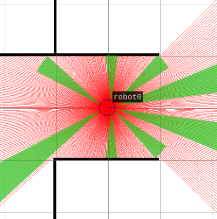
(4)

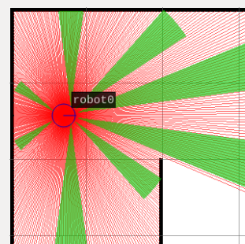
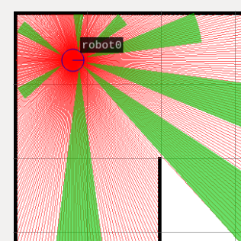
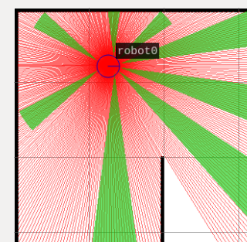
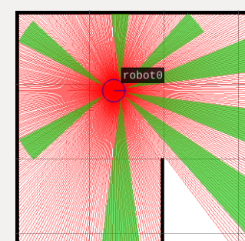
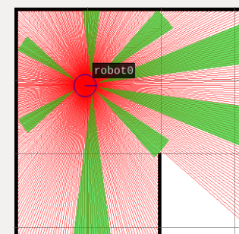
(5)

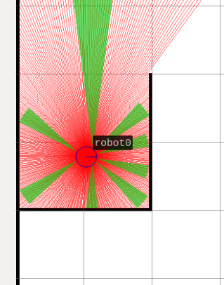
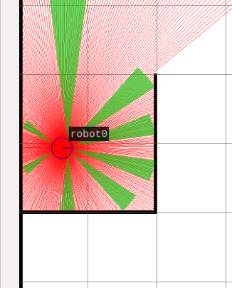
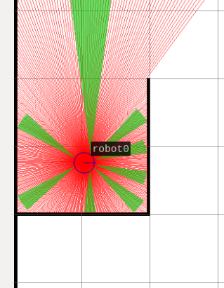
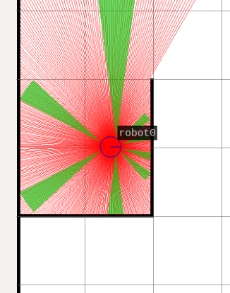
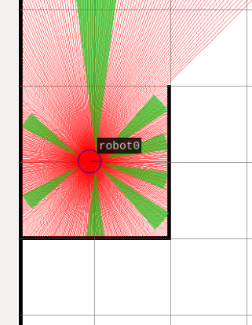
(6)

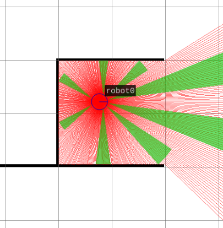
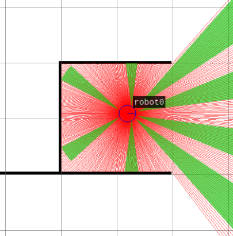
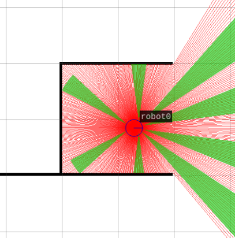
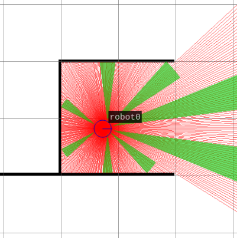
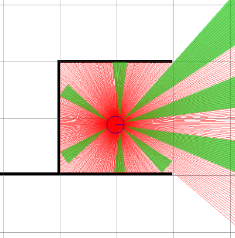
(7)

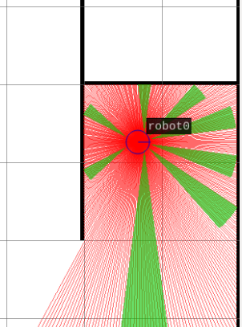
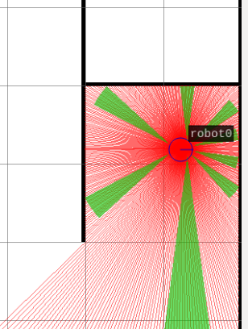
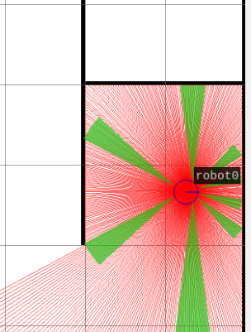
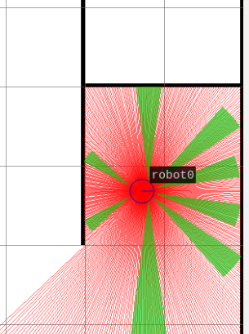
(8)

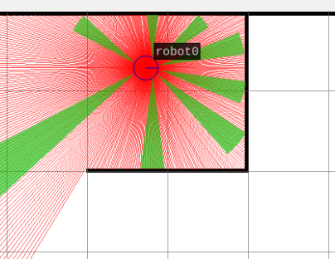
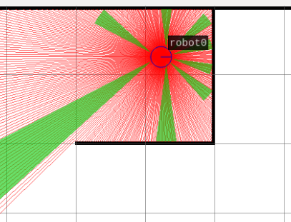
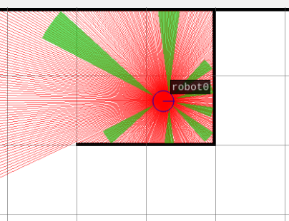
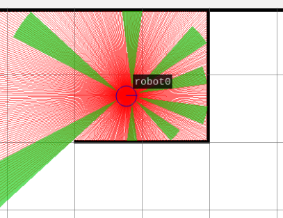
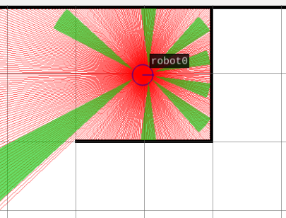
(9)

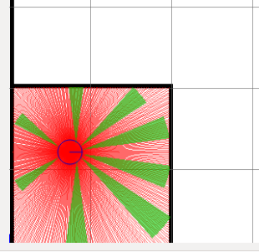
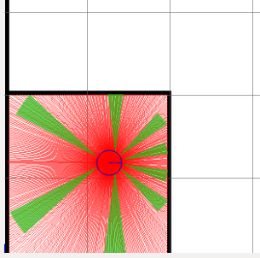
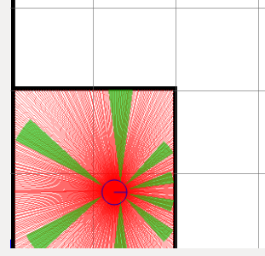
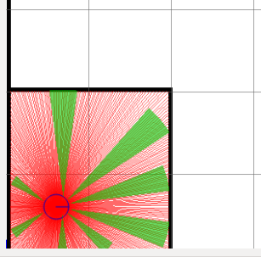
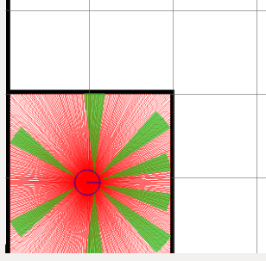
(10)

(11)

(12)

(13)

(14)

(15)

# 4.Actuadores en ROS

1. Diseñe una función avanzar que reciba como parámetro de entrada la distancia a avanzar (2m, 4m…).

La función desarrollada hay que pasarla por parámetros la velocidad y la distancia que ha de recorrer el robot.

Conociendo los datos creamos una instancia al subscriber de la odometría y otra al Publisher de la velocidad y del apartado Twist de la odometría.

Formamos un mensaje añadiendo la velocidad lineal declarada en los parámetros de entrada y posteriormente entramos en un bucle, el cual compara si la distancia recorrida respecto al inicio es la facilitada por parámetros. En caso de que aún no se haya recorrido la distancia especificada no se alterará el mensaje conformado al principio, manteniendo el movimiento del robot a la misma velocidad; en caso de haber recorrido la distancia prefijada, se edita el mensaje declarando la velocidad lineal a 0, parando el robot en el momento.

1. Diseñe una función girar que se reciba como parámetro de entrada el ángulo a girar (45º, 90º…).

La función desarrollada hay que pasarla por parámetros la velocidad y el ángulo que ha de girar el robot.

Conociendo los datos creamos una instancia al subscriber de la odometría y otra al Publisher de la velocidad y del apartado Twist de la odometría.

Formamos un mensaje añadiendo la velocidad angular declarada en los parámetros de entrada, transformamos el ángulo facilitado por parámetros de grados a radianes y calculamos cuanto tiempo tardará el robot con la velocidad fijada en posicionarse. Tras esto entramos en un bucle que se ejecutará durante el tiempo estimado de giro, enviando al Publisher el mensaje manteniendo la velocidad angular. Cuando finalice el bucle, significara que ya esta orientado correctamente, por lo que enviamos al robot un mensaje indicándole que ponga la velocidad angular a 0 y por tanto frenando el robot.

1. Empleando el simulador STDR, y utilizando las funciones avanzar y girar, navegue con el robot desde la esquina inferior derecha(9,1,0) hasta la salida utilizando los puntos centrales de las casillas como coordenadas de destino.

MIAU

1. Realice un recorrido con el robot real concatenando los siguientes tramos:
   * 1 tramo recto de 2m
   * 1 giro de 90º
   * 1 tramo recto de 1m
   * 1 giro de -90º
   * 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?

MIAU