

**UNIVERSIDAD AUTONOMA DE MADRID**

**ESCUELA POLITECNICA SUPERIOR**



## **PROYECTO FIN DE CARRERA**

**DESARROLLO E INTEGRACIÓN DE UN CONJUNTO DE  
SENSORES PARA LA NAVEGACIÓN AUTÓNOMA DE UN  
UGV (VEHÍCULO TERRESTRE NO TRIPULADO)  
PERTENECIENTE A LA PLATAFORMA ROBÓTICA  
ARGOS**

**Jose Ignacio Rejas Hernán**

**Noviembre 2014**



**DESARROLLO E INTEGRACIÓN DE UN CONJUNTO DE  
SENSORES PARA LA NAVEGACIÓN AUTÓNOMA DE UN  
UGV (VEHÍCULO TERRESTRE NO TRIPULADO)  
PERTENECIENTE A LA PLATAFORMA ROBÓTICA  
ARGOS**

**AUTOR: Jose Ignacio Rejas Hernán  
TUTOR: Guillermo González de Rivera**

**HCTLab  
Dpto. Tecnología Electrónica y de Comunicaciones  
Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
Noviembre 2014**

**HCTLab**





## **Resumen**

En este proyecto se ha llevado a cabo el desarrollo e integración de un sistema de sensores para la navegación autónoma de un vehículo terrestre no tripulado. Para lograr dicho propósito es necesaria la realización de un estudio acerca de las tecnologías disponibles para dicho propósito, así como los diferentes sensores existentes. Una vez realizado este estudio, es necesario imponer unos requisitos mínimos que deben ser cumplidos para comenzar a filtrar aquellos sensores que son más apropiados para dicho propósito.

Posteriormente, se cogen aquellos sensores que cumplen con dichos requisitos mínimos y se les evalúa según unos criterios deseables para la elección del sistema más apropiado. Llegados a este punto, se elige un sistema sensorial que, bajo ciertas especificaciones, es capaz de amoldarse perfectamente al objetivo de este proyecto. Dicho sistema está formado por un sensor láser 2D y una plataforma de giro que rotará apropiadamente el sensor para conseguir captar información 3D del entorno que rodea al vehículo.

Elegido ya el sistema sensorial, se pasa a elaborar simulaciones en Matlab acerca de su funcionamiento, para averiguar si el resultado es el esperado. Dichas simulaciones se centran principalmente en la búsqueda de una configuración óptima de velocidad de giro para el sistema. Tras varias situaciones analizadas en profundidad, se llega a la conclusión de que no existe una única velocidad de giro, sino que esta es dependiente del propósito que se tenga, de la velocidad del vehículo y de la densidad de puntos que se desee obtener.

Una vez alcanzado tales conclusiones y con el total convencimiento del teórico buen funcionamiento de la idea desarrollada, se procede a la construcción del sistema sensorial.

Mientras dicho sistema es construido, se procede al desarrollo de algoritmos en lenguaje C++ y ROS para la recepción, procesamiento y análisis de los datos suministrados por el sistema. Todo el procesamiento y exposición de resultados debe lograrse en tiempo real y con la mayor densidad de puntos posible, de modo que permita minimizar el error o la existencia de zonas desconocidas. El correcto funcionamiento de esta programación es verificado con el simulador Gazebo, que ofrece una gran cantidad de posibilidades de análisis.

Una vez logrado que el sistema procese y represente la información obtenida en tiempo real con un coste computacional bajo, se procede a ejecutar dicha algoritmia con el sistema sensorial construido. Tras la realización de pruebas de todo el sistema en entornos controlados y estáticos, en los que se muestran los mapas 3D y 2D generados y analizados, se procede a la instalación del sistema sensorial en el vehículo y al análisis final del sistema completo. Los resultados son esperanzadores y dejan la puerta abierta a la posibilidad de incorporar algoritmos de navegabilidad.

## **Palabras clave**

UGV, densidad de puntos, región, entorno, láser, LIDAR, sensor, sistema sensorial, plataforma de giro, velocidad de giro, RPS, barrido, mapa, tiempo real.



## **Abstract**

This project is based on the development and integration of a sensor set for the autonomous navigation of an unmanned ground vehicle. To achieve this target is necessary making a studio about available technologies oriented to it and being able to find different sensor in each area technological. After this, some minimum requirements are established so every studied sensor must exceed them. Thank to this requirements, less sensors are available to achieve this target.

Later, a new set of desirable additional requirements is created, so the best sensor can be chosen and the sensorial system can be made for this proposal. The system is based on the fabrication of a rotating laser system, as an improvement of an off-the-shelf laser scanner, capable of generating 3D maps from the surrounding.

Before starting to make the sensorial system, some Simulation studies have been conducted to assess the resulting measurement data. Patterns, density and minimum detection studies have been done. The optimal rotational speed of the sensors system will be that one that is higher than the required threshold on each instance, and it must be the immediately higher, because improvements has not been detected at higher rotation speeds.

While the sensorial system is made, the ROS and C++ software is developed. This software must process and analyze system data provided by the system. All this processing has to be in real time, with a high point cloud and low computational cost. The correct operation of this software is verified by Gazebo simulations, which provide a lot of study possibilities.

After testing all system in controlled and static environments, which results and maps can be checked, the complete installation of sensorial system is provided, so a final version is offered. The promising and awesome test are shown so navigability algorithms can be realized.

## **Keywords**

UGV, point cloud, area, environment, laser, LIDAR, sensor, sensorial system, rotating platform, rotation speed, RPS, scan, map, real time, time of flight.



## Agradecimientos

Me gustaría dar las gracias a mi tutor, Guillermo González de Rivera, por darme la oportunidad de realizar este proyecto y al HCTLab, en especial a Alberto Sánchez y Fernando Colino por los pequeños pero muchos momentos divertidos que hemos pasado.

Gracias a Manuel Prieto por todo el apoyo y dedicación que me ha ofrecido. También quiero agradecer a todo ROBOMOTION los grandes momentos que hemos pasado juntos y por su cálida acogida durante toda la duración del proyecto, llegué a sentirme un trabajador más.

A mis amigos, en especial a Xu, Karim y Luis. Hemos pasado muchos buenos momentos y otros bastante agobiados pero todos hemos conseguido superarlos y llegar al final.

A mi familia, por apoyarme en todo momento.

Por último y más importante, quiero agradecer a Miriam la inmensa paciencia y todo el apoyo que me ha dado en todos estos años. Desde que nos conocimos en segundo curso, nos hemos ayudado, divertido y querido. Contigo me he dado cuenta de que esta etapa que hemos vivido ha sido la mejor de mi vida.

Este trabajo ha sido financiado en parte a través del Programa INNPACTO, Proyecto ARGOS, IP-2012-0308-390000 y del Fondo Europeo de Desarrollo Regional (Fondos FEDER)





# ÍNDICE DE CONTENIDOS

<b>1</b>	<b>Introducción</b>	<b>9</b>
1.1	Motivación	9
1.2	Objetivos	10
1.3	Organización de la memoria	10
<b>2</b>	<b>Estado del arte</b>	<b>11</b>
2.1	Introducción	11
2.2	Conocimientos previos	11
2.2.1	Sistemas basados en tiempo de vuelo	11
2.2.2	Sistemas basados en diferencia de fase	12
2.2.3	Sistemas basados en triangulación	13
2.3	Tipos de tecnología	13
2.3.1	Sistema RADAR de onda continua	13
2.3.1.1	Funcionamiento	13
2.3.1.2	Ejemplo de sensor	14
2.3.2	Ultrasonidos	14
2.3.2.1	Funcionamiento	14
2.3.2.2	Ejemplo de sensor	14
2.3.3	Cámaras estereoscópicas	15
2.3.3.1	Funcionamiento	15
2.3.3.2	Ejemplo de sensor	15
2.3.4	Cámara 3D infrarroja	16
2.3.4.1	Funcionamiento	16
2.3.4.2	Ejemplo de sensor	16
2.3.5	LIDAR	17
2.3.5.1	Funcionamiento	17
2.3.5.2	Ejemplo de sensor	18
2.4	Prerrequisitos	19
2.4.1	Introducción	19
2.4.2	Grado de protección	19
2.4.3	Accesibilidad	20
2.4.4	Establecimiento de requisitos	21
2.5	Requisitos deseables	21
2.5.1	Introducción	21
2.5.2	Generación de medidas 3D del entorno	22
2.5.3	Rango de distancias y margen de errores	22
2.5.4	Apertura angular	22
2.5.5	Velocidad de barrido	22
2.5.6	Intensidad de luz	22
2.5.7	IP	23
2.5.8	Resolución angular	23
2.5.9	Coste	23
2.5.10	Tabla de comprobación requisitos deseables	23
<b>3</b>	<b>Propuestas preliminares</b>	<b>26</b>
3.1	Introducción	26
3.2	Propuesta 1: Un sensor LIDAR con inclinación	26
3.3	Propuesta 2: Dos sensores LIDAR. Uno de ellos, con inclinación y el otro horizontal	27
3.4	Propuesta 3: Sensor LIDAR con inclinación y una cámara estereoscópica	28

3.5 Propuesta 4: Sensor LIDAR con inclinación y una plataforma de giro.....	29
3.5.1 Giro con respecto al eje “y” de manera permanente o no permanente.....	30
3.5.1 Giro con respecto al eje “x” de manera permanente .....	31
3.6 Estudio previo a la realización de la propuesta elegida .....	32
3.6.1 Introducción.....	32
3.6.2 Desarrollo del análisis .....	32
3.6.2.1 Estudio de dos barridos laser con diferentes velocidades de giro .....	33
3.6.2.2 Estudio la velocidad óptima de giro .....	35
3.6.2.3 Estudio la velocidad óptima de giro dependiente de la velocidad del	
vehículo .....	39
3.6.2.4 Conclusiones.....	44
3.6.2.5 Estudio resolución de medidas en el centro del escáner.....	45
3.6.3 Elección de aceptación sistema sensorial .....	46
3.7 Simulación en 3D.....	47
<b>4 Composición y elaboración del sistema sensorial .....</b>	<b>51</b>
4.1 Introducción .....	51
4.2 Componentes sistema de giro .....	51
4.2.1 Motor .....	51
4.2.2 Encoder.....	51
4.2.3 Reductora.....	52
4.2.4 Unidad de medición inercial (IMU) .....	53
4.2.5 Controlador EPOS .....	53
4.2.6 Anillo rotatorio (slip ring) .....	53
4.3 Requisitos e implementación .....	54
<b>5 Desarrollo software del sistema sensorial y resultados de la simulación .....</b>	<b>57</b>
5.1 Introducción .....	57
5.1.1 ROS .....	57
5.1.2 Gazebo .....	60
5.1.3 Octovis.....	60
5.2 Desarrollo software del sistema sensorial.....	60
5.2.1 Introducción.....	60
5.2.2 Desarrollo del sistema sensorial real en Gazebo .....	60
5.2.3 Desarrollo software del sistema en ROS .....	66
5.2.3.1 Introducción.....	66
5.2.3.2 Nodo gzhokuyo_node.....	67
5.2.3.3 Nodo urg_node .....	68
5.2.3.4 Nodo gzhokuyo_pose_node .....	69
5.2.3.5 Nodo rls30_controlnode .....	69
5.2.3.6 Nodo P_LECTOR3D.....	70
5.2.3.7 Nodo pcl_filter_ros.....	75
5.2.3.8 Nodo rls30_mapper .....	76
5.2.3.9 Nodo Prueba .....	77
5.2.4 Lanzadores y esquema de conexión .....	77
5.2.4.1 Introducción.....	77
5.2.4.2 Lanzadores .....	77
5.2.4.3 Esquema de conexión.....	79
5.2.4.4 Resultados simulación .....	80
<b>6 Pruebas estáticas y dinámicas del sistema sensorial.....</b>	<b>82</b>
6.1 Prueba estática en habitación .....	82
6.2 Prueba estática largo alcance .....	87

6.3 Prueba dinámica en el pasillo con obstáculos estáticos .....	93
6.4 Prueba dinámica en el garaje con obstáculos estáticos .....	94
6.5 Prueba estática con obstáculos dinámicos .....	96
6.6 Prueba dinámica sistema completo .....	101
<b>7 Conclusiones y desarrollos futuros .....</b>	<b>104</b>
7.1 Conclusiones .....	104
7.2 Trabajos futuros .....	105
<b>Referencias .....</b>	<b>107</b>
<b>Glosario .....</b>	<b>110</b>
<b>Anexos.....</b>	<b>I</b>
A Tabla .....	I
B Tabla .....	III
C Artículo: A Varying Density 3D Laser Scanner for Unmanned Ground Vehicles Mapping and Obstacle Detection.....	IV



## ÍNDICE DE FIGURAS

- Figura 2-1: Esquema del funcionamiento de un sistema basado en tiempo de vuelo  
Figura 2-2: Ejemplo de desfase entre señales  
Figura 2-3: Esquema del funcionamiento de un sistema basado en diferencia de fase  
Figura 2-4: Representación de un sistema basado en triangulación  
Figura 2-5: Funcionamiento sistema LIDAR 2D  
Figura 2-6: Funcionamiento sistema LIDAR 3D
- Figura 3-1: Representación de la configuración propuesta y posibles obstáculos emergentes  
Figura 3-2: Representación de la configuración propuesta y posibles obstáculos emergentes  
Figura 3-3: Fotografía y resultados del procesamiento de datos en un camino  
Figura 3-4: Fotografía y resultados del procesamiento de datos en un camino con un charco  
Figura 3-5: Sistema de giro desarrollado en un artículo  
Figura 3-6: Vehículo Yamaha Rhino  
Figura 3-7: Situación inicial  
Figura 3-8: Una vez alcanzado  $\alpha_{\min}$  se cambia el sentido de giro  
Figura 3-9: Una vez alcanzado  $\alpha_{\max}$  se vuelve a cambiar el sentido de giro  
Figura 3-10: Barrido láser  
Figura 3-11: Barrido láser simulado en Matlab  
Figura 3-12: Representación dos barridos láser con diferentes velocidades de giro  
Figura 3-13: Representación de los barridos láser tomados en un segundo  
Figura 3-14: Representación barrido láser (azul) a 30 metros, 1 RPS y 1segundo y mallado (verde)  
Figura 3-15: Representación barrido láser (azul) a 30 metros, 1 RPS y 1segundo y mallado (verde) en la apertura impuesta  
Figura 3-16: Representación barrido láser (azul) a 8 metros con diferentes velocidades de giro y mallado (verde) de 1 metro  
Figura 3-17: Representación barrido láser (azul) a 10 metros con diferentes velocidades de giro y mallado (verde) de 10 x 10 centímetros  
Figura 3-18: Representación de un barrido láser generado cuando éste ha recorrido 180°
- Figura 4-1: Imagen de una reductora  
Figura 4-2: Imagen de un anillo rotativo  
Figura 4-3: Imagen de un anillo rotativo  
Figura 4-4: Conexión y colocación de los componentes  
Figura 4-5: Diseño del sistema sensorial simulado  
Figura 4-6: Fotografía sistema sensorial real
- Figura 5-1: Sensor simulado en Gazebo. Su barrido se muestra en color azul  
Figura 5-2: Sensor y vehículo simulado en Gazebo.  
Figura 5-3: Barrido realizado por el sensor en un instante de tiempo dado mientras el sistema se encuentra girando respecto al eje x  
Figura 5-4: Barrido realizado por el sensor en un entorno con obstáculos. Se puede ver claramente cómo las patas de la mesa son detectadas  
Figura 5-5: Mensaje de ejemplo de tipo laserScan  
Figura 5-6: Ejemplo de sincronización entre mensajes del láser y mensajes de la posición del motor  
Figura 5-7: Esquema de conexión de temas y nodos con el lanzador gazebo\_hokuyo.launch

Figura 5-8: Esquema de conexión de temas y nodos con el lanzador  
urg30lx\_hokuyo.launch

Figura 5-9: Mapa 3D generado a partir de la simulación en Gazebo sin obstáculos (únicamente el suelo)

Figura 5-10: Ejemplo de rotación laser y una pared en frente que debe ser detectada en tiempo real

Figura 5-11: Mapa 3D generado a partir de la simulación en Gazebo con obstáculo

Figura 6-1: Fotográfica de la parte de la habitación a escanear con el sistema sensorial

Figura 6-2: Escaneado de la habitación sin filtrado (representado con el visualizador Rviz)

Figura 6-3: Escaneado de la habitación sin filtrado (representado con el visualizador Rviz)

Figura 6-4: Cantidad de datos anteriores y posteriores al filtrado realizado

Figura 6-5: Escaneado láser de la habitación con filtrado

Figura 6-6: Mapa 2D global generado de la habitación

Figura 6-7: Mapa 3D de la habitación generado con el visualizador Octovis

Figura 6-8: Fotografía escena a analizar

Figura 6-9: Mapa 3D generado sin filtrado

Figura 6-10: Cantidad de datos anteriores y posteriores al filtrado realizado

Figura 6-11: Mapa 3D generado con filtrado

Figura 6-12: Vista lateral del fondo del garaje

Figura 6-13: Mapa 2D global generado

Figura 6-14: Mapa 2D global generado (vista lateral)

Figura 6-15: Mapa 3D generado a partir del visualizador Octovis

Figura 6-16: Mapa 3D generado a partir del visualizador Octovis (vista lateral)

Figura 6-17: Fotografía de la colocación de las cajas

Figura 6-18: Mapa 3D filtrado del entorno

Figura 6-19: Mapa 3D filtrado del entorno

Figura 6-20: Mapa 2D generado del entorno analizado

Figura 6-21: Fotografía del pasillo

Figura 6-22: Mapa 2D generado del pasillo

Figura 6-23: Mapa 2D que se va completando según la traslación del sistema sensorial

Figura 6-24: Fotografías del recorrido realizado

Figura 6-25: Vehículo Yamaha Rhino equipado con el sistema sensorial

Figura 6-26: Mapa 3D generado, sin obstáculo dinámico

Figura 6-27: Fotografías del movimiento realizado por la persona

Figura 6-28: Detección del obstáculo dinámico

Figura 6-29: Detección del obstáculo dinámico inesperado (con decay time)

Figura 6-30: Mapa 2D generado y actualizado según la posición del sistema sensorial

## ÍNDICE DE TABLAS

- Tabla 2-1: Descripción primer parámetro IP
- Tabla 2-2: Descripción segundo parámetro IP
- Tabla 2-3: Desglose del cumplimiento de requisitos deseables
- Tabla 2-4: Resumen del cumplimiento de requisitos deseables
- Tabla 3-1: Relación entre apertura angular, distancia y apertura en metros
- Tabla 3-2: Mallado mínimo en cada una de las diferentes aperturas angulares y a distintas velocidades de giro
- Tabla 3-3: Velocidad mínima de giro del sistema
- Tabla 3-4: Mallado disponible a diferentes velocidades dada la apertura descrita
- Tabla 3-5: Mallado disponible a diferentes velocidades dada una apertura  $\pm 5$  metros
- Tabla 3-6: Velocidad mínima de giro del sistema
- Tabla 3-7: Mallado disponible a diferentes velocidades dada la apertura descrita
- Tabla 3-8: Mallado disponible a diferentes velocidades dada una apertura  $\pm 5$  metros
- Tabla 3-9: Velocidad mínima de giro del sistema
- Tabla 3-10: Mallado disponible a diferentes velocidades dada la apertura descrita
- Tabla A-1: Desglose de características de sensores
- Tabla B-1: Tablas originales de mallado, sin la imposición de una apertura específica



# 1 Introducción

---

## 1.1 Motivación

A día de hoy, el interés por la navegación autónoma y semi-autónoma de vehículos está en constante crecimiento debido a la gran variedad de tecnologías que se están adaptando a dicho propósito. Las investigaciones se están orientando en eliminar la figura del conductor de los vehículos, y que éstos sean capaces de ejercer la conducción en cualquier entorno. Dicho entorno puede estar previamente definido (autopistas, calles etc...), o ser inicialmente desconocido, como por ejemplo un camino forestal.

Los primeros avances son cada vez más comunes en nuestro día a día. De hecho, muchos de los vehículos ya tienen incorporado un sistema de sensores capaz de generar información sobre el tráfico, o sobre las diversas condiciones periféricas. Por tanto, es posible conocer en tiempo real el entorno alrededor del vehículo, de modo que el conductor pueda tomar distintas decisiones a partir de su apreciación y de los datos proporcionados por el sistema, logrando una conducción más segura y fiable. Incluso hay vehículos capaces de aparcar solos, consiguiendo hacer dicha labor de manera automática.

Gracias a los sistemas que analizan los datos recogidos por el conjunto de sensores, el vehículo podrá tomar medidas críticas, como el frenado de emergencia si se detecta una colisión. Esta puede ser tanto prevista, si el sistema la ha detectado en su trayecto y se aproxima a ella poco a poco, o bien imprevista si el obstáculo se cruza en el camino de manera inesperada. Otras medidas que podrá tomar el sistema será adaptar la velocidad dependiendo de la vía de circulación y de las condiciones atmosféricas.

Se está trabajando en llegar más allá y fabricar un vehículo no tripulado capaz de ejercer la conducción semi-autónoma en diferentes vías reales de circulación. En la actualidad y según avanzan las investigaciones, se espera que pronto estén en el mercado vehículos con estas características.

La motivación de este proyecto es colaborar con el avance de estas tecnologías a través de un sistema sensorial orientado a la navegación semi-autónoma en entornos no controlados, de modo que se pueda conocer en tiempo real los posibles obstáculos y variaciones del terreno en frente del vehículo.

El grupo de investigación HCTLab de la Universidad Autónoma de Madrid está colaborando activamente con la empresa Robomotion en el proyecto ARGOS. Dicho proyecto engloba un conjunto de tecnologías de las que forma parte el sistema sensorial a desarrollar.

## 1.2 Objetivos

El objetivo principal de este proyecto es equipar a un vehículo real (modelo Rhino de la marca Yamaha) con un sistema de sensores que, circulando por cualquier tipo de vía sin información previa de su estructura, sean capaces de detectar obstáculos y variaciones de pendiente en el terreno en tiempo real. Todo ello para lograr que el vehículo sea capaz de ejercer una conducción con asistencia inteligente por los sistemas anticolidión y de seguridad del vehículo.

El desglose de objetivos se puede ver a continuación:

1. Estudio del estado del arte en sensores para conducción autónoma.
2. Estudio y clasificación de los obstáculos que deben ser detectados por el sistema sensorial.
3. Establecer una serie de requisitos y especificaciones para el sistema sensorial.
4. Realizar unas propuestas preliminares de sistema sensorial para el vehículo.
5. Realizar un estudio simulado de las condiciones de operación del sistema para evaluar el rendimiento de las propuestas realizadas para los casos y condiciones de interés.
6. Seleccionar una de las propuestas tras realizar un análisis de las simulaciones realizadas y el cumplimiento de los requisitos.
7. Compra de los componentes necesarios para construir el sistema propuesto.
8. Realizar el desarrollo de software y algoritmia de procesamiento de datos, detección de obstáculos y generación de mapas en tiempo real.
9. Integración del sistema y puesta en marcha en el laboratorio.
10. Probar el sistema integrado en el vehículo.
11. Análisis y discusión de los datos experimentales medidos con el sistema real.

## 1.3 Organización de la memoria

La memoria consta de los siguientes apartados:

- Capítulo 1: Introducción, motivación y objetivos del proyecto.
- Capítulo 2: Estado del arte actual: exposición de las diferentes tecnologías aplicables a la navegación autónoma, requisitos previos, requisitos deseables y exposición de resultados.
- Capítulo 3: Propuestas preliminares y realización de un estudio simulado con las condiciones de operación del sistema.
- Capítulo 4: Composición y elaboración del sistema sensorial elegido.
- Capítulo 5: Desarrollo software del sistema sensorial y resultados de simulación.
- Capítulo 6: Pruebas estáticas y dinámicas del sistema sensorial.
- Capítulo 7: Conclusiones y desarrollos futuros.

## 2 Estado del arte

### 2.1 Introducción

La navegación autónoma de vehículos terrestres no tripulados en entornos no controlados es un área muy activo de investigación en el mundo. El logro de este objetivo está cada vez más cerca debido a la gran variedad y mejoras de la tecnología. Un componente básico para cualquier vehículo no tripulado es el sistema sensorial, con el que se puede extraer y analizar información del entorno alrededor del vehículo.

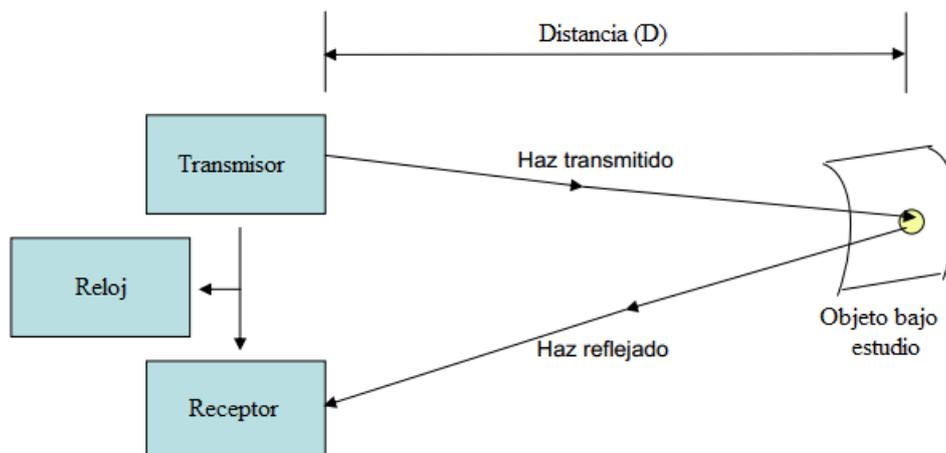
Un sistema sensorial está formado por uno o más sensores capaces de ofrecer datos para su posterior procesamiento. La forma de obtener datos del entorno es muy variada, debido a las diferentes tecnologías que se han orientado a dicho propósito. Esto permite diseñar un sistema sensorial que se ajuste a la situación bajo estudio de la mejor manera posible, abriendo un gran abanico de posibilidades. Por tanto, es posible formar un sistema sensorial con sensores de diferentes tecnologías con el fin de no tener un único tipo de información [1]. No obstante, hay casos en los que con varios sensores de la misma tecnología se puede conseguir el objetivo de la navegación autónoma [2].

En el próximo apartado se analizará algunos de los conocimientos técnicos necesarios para entender el principio de funcionamiento de las diferentes tecnologías. Después se va a realizar un desglose de cada tipo, acompañado de un modelo comercial a modo de ejemplo.

### 2.2 Conocimientos previos

#### 2.2.1 Sistemas basados en tiempo de vuelo

Es un tipo de medición que determina la distancia a un objeto mediante el tiempo que tarda el haz emitido por el sensor en llegar a dicho objeto y ser captado por el receptor del sensor [3]. En función del tiempo de vuelo y conocida la velocidad del haz, se calcula la distancia del objeto. Por tanto, se necesita que el haz sea captado por el receptor para que se pueda analizar la información acerca del objeto. En la siguiente figura se puede ver de manera sencilla esta explicación:



*Figura 2-1: Esquema del funcionamiento de un sistema basado en tiempo de vuelo*

En el caso de emitir un pulso láser se aplican las siguientes fórmulas:

$$D = \frac{c * t}{2} \quad \Delta D = \frac{c * \Delta t}{2}$$

Donde “D” es la distancia (metros) al objeto, “c” es la velocidad de la luz (metros/segundo), “t” es el tiempo de vuelo (segundo), “Δt” es resolución de las medidas de tiempo (segundo) y “ΔD” es resolución en la distancia (metros).

Si se envía más de un haz laser, se siguen las siguientes ecuaciones:

$$D = \frac{c \varphi}{2 2\pi f} \quad \Delta D = \frac{c \Delta\varphi}{2 2\pi f}$$

Donde “f” es la frecuencia (Hz), “φ” fase (radianes) y “Δφ”: resolución de fase (radianes).

### 2.2.2 Sistemas basados en diferencia de fase

Este método permite la máxima precisión. Estos sistemas se basan en observar continuamente la fase de las señales emitida y recibida. La señal recibida habrá recorrido una distancia D, correspondiente a un cierto número entero N de sus longitudes de onda, llamado ambigüedad, más un desfase o cierta parte de la longitud de onda. Si este último valor es igual o superior a 1, significa que se ha recorrido otro número entero de longitud de onda, por lo que se tiene N+1 veces la longitud de onda y una nueva fracción de esta. La siguiente figura presenta un ejemplo en el que N es igual a 0, la longitud de onda viene representada por “λ” y la fracción de la longitud de onda es “φ”

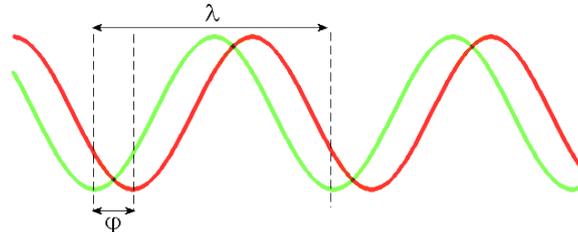


Figura 2-2: Ejemplo de desfase entre señales

Para resolver la distancia a la que se encuentra el objeto, hay que hacer un seguimiento de la fase en todo momento, de manera que se pueda saber cuánta es la ambigüedad y la fracción de las señales. En la siguiente figura se puede ver de manera general este esquema de funcionamiento:

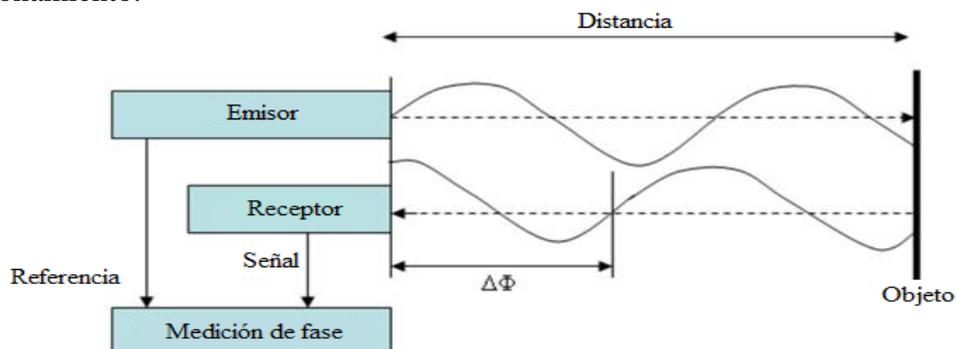


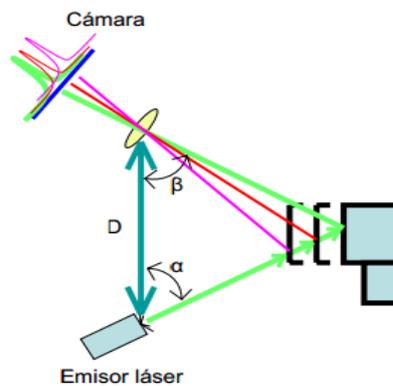
Figura 2-3: Esquema del funcionamiento de un sistema basado en diferencia de fase

Es fundamental que en estos sistemas no se pierda el seguimiento de fase, debido a que es una medida imprescindible para el cálculo de la distancia.

Estos sistemas utilizan señales de onda continua modulada en amplitud (AMCW) y son más rápidos que los sistemas basados en el tiempo de vuelo [4].

### 2.2.3 Sistemas basados en triangulación

Se dirige un haz sobre un objeto y se coloca a una distancia conocida una cámara (D), que trata de captar el haz rebotado por el objeto. De este modo, se consigue un triángulo formado por la cámara, el objeto en cuestión y el emisor del haz, con los ángulos del emisor láser y cámara conocidos ( $\alpha$ ,  $\beta$ ). Por tanto, estos sistemas trabajan con un único punto de medida, es decir un único haz emitido y analizado, pero es posible lanzar varios haces con un determinado ángulo de desviación entre ellos de manera que el receptor pueda captar de manera más rápida un haz rebotado. El único requisito para que esto funcione es que se sepa qué haz de todos los que se han enviado ha sido captado por el receptor.



*Figura 2-4: Representación de un sistema basado en triangulación*

Estos sistemas suelen poseer un corto alcance y no son utilizados en la navegación autónoma de vehículos terrestres [5].

## 2.3 Tipos de tecnología

### 2.3.1 Sistema RADAR de onda continua

#### 2.3.1.1 Funcionamiento

La principal tarea de los sensores RADAR (Radio Detecting And Ranging) es detectar objetos y medir sus velocidades y posiciones relativas. Para ello, este sistema requiere una modulación en la señal radiada, que puede ser en amplitud o en frecuencia. Estos sistemas funcionan entre los 75 y 110 GHz (banda W) y a 24 GHz. [6]

Dichas señales emitidas son reflejadas por los objetos alrededor del vehículo y pasan por el receptor del sistema, que es capaz de calcular la distancia y velocidad de dicho objeto a partir de unos parámetros conocidos, como el tiempo de vuelo y la velocidad de la onda. El diseño de estos robustos sensores hace que sean resistentes a las vibraciones, de modo que se puede incorporar a vehículos.

Se puede implementar un objetivo adicional o una calefacción para que su rendimiento sea completo incluso bajo situaciones adversas.

Este tipo de tecnología ha sido usada para aplicaciones en vehículos terrestres no tripulados en entornos controlados como autopistas o autovías [7] [8] [9]. Si es usado en entornos no controlados, se presentan problemas en cuanto a la detección de algunos tipos de obstáculos.

### 2.3.1.2 Ejemplo de sensor

Uno de los principales fabricantes de esta tecnología es Bosch Automotive Technology [10]. Su modelo LRR3 es un sensor de largo alcance y muy poco consumo cuyas principales características se muestran a continuación [11]:

	Apertura (grados)	Rango de detección (metros)	Precisión (metros)	Velocidad del escáner (milisegundos)
	30 horizontal 5 vertical	0.5-250	±0.1	80

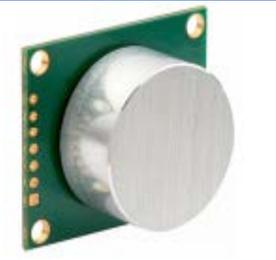
## 2.3.2 Ultrasonidos

### 2.3.2.1 Funcionamiento

Estos sensores están basados en el concepto del tiempo de vuelo. Es decir, la distancia es calculada midiendo el tiempo de retorno de la onda y conociendo la velocidad del sonido. Este tipo de tecnología supone algunos problemas que no los hacen muy recomendables para la navegación en entornos no controlados, como la direccionalidad, la dependencia de la temperatura, la humedad o la suciedad. A pesar de dichos problemas, esta opción se ha utilizado en algunos proyectos. [12]

### 2.3.2.2 Ejemplo de sensor

El MB7260-XL-MaxSonar [13] es un sensor de corto alcance basado en este tipo de tecnología. Algunas de sus características se exponen a continuación:

	Apertura (grados)	Rango de detección (metros)	Precisión (metros)	Velocidad del escáner (milisegundos)
	53	0-7.65	±0.01	100

## 2.3.3 Cámaras estereoscópicas

### 2.3.3.1 Funcionamiento

Estos sistemas copian la idea de la visión humana. Se trata de dos cámaras alineadas en un eje con una conocida separación entre ellas. Con un solo ojo/cámara solo se es capaz de captar imágenes bidimensionales pero con dos ojos/cámaras captan dos imágenes separadas una determinada distancia y con cierto solapamiento entre ellas, de modo que se puedan calcular distancias de profundidad [14]. Las cámaras estereoscópicas presentan requisitos muy estrictos para su conseguir unos resultados coherentes. En primer lugar las dos imágenes captadas deben tener igual brillo, igual intensidad lumínica e igual contraste. Obviamente las dos imágenes deben mostrar la misma escena con un cierto solapamiento entre ellas para lograr el efecto 3D. Por último, el alineamiento de las cámaras es crítico, una ligera desviación puede desencadenar en un sistema de medición inservible.

Existe un compromiso entre solapamiento y apertura. A más solapamiento en las imágenes, la precisión mejora notablemente. Por contrala apertura es menor y viceversa.

Además requieren de transformaciones y cálculos numéricos a partir de las imágenes tomadas, para poder saber a qué distancia se encuentran los objetos captados. Estos cálculos a menudo son tediosos y con alto coste computacional.

Esta tecnología es usada en robots como el Curiosity [15], Spirit y otros proyectos [16][17][18].

Este tipo de tecnología tiene el problema de que un posible reflejo en una imagen pueda dar lugar a un gran número de medidas erróneas. Este efecto es fácilmente entendible en el caso de que en la imagen tomada aparezca el reflejo de un charco. El sistema lo tomará como un obstáculo al ser esta zona diferente a las de su alrededor y esto es un error muy difícil de solucionar [19]. No obstante, es una gran elección de sistema sensorial para la navegación autónoma, sobre todo si sus resultados se fusionan con otros sensores.

### 2.3.3.2 Ejemplo de sensor

Las cámaras Bumblebee BB2-03S2 y BBXB3 [20] son cámaras con dos y tres lentes respectivamente. A continuación se exponen algunas de sus principales características.

	<b>Relación señal-ruido (dB)</b>	<b>Imágenes por segundo</b>	<b>Resolución (píxeles)</b>
	60	48	648x488
	54	16	1280x960

## 2.3.4 Cámara 3D infrarroja

### 2.3.4.1 Funcionamiento

Mientras que la cámara RGB se encarga de obtener la información en color de todo aquello situado en el campo de visión, la cámara infrarroja de corto alcance se encarga de obtener la información de profundidad.

Las cámaras infrarrojas, para la obtención de la profundidad, se basan en el tiempo de vuelo, aunque si emiten de manera constante se utiliza el desfase producido en la longitud de onda entre el rayo emitido y el recibido.

Sin embargo hay otras cámaras que se basan en emitir un patrón o mallado infrarrojo conocido que se captará desde el sensor infrarrojo. En este caso, para calcular la distancia se mide la deformación de los puntos que componen esta malla, por medio de la triangulación de cada punto de la malla infrarroja de partida y su correspondiente punto en la malla recibida. Por medio de este cálculo de todos los puntos del patrón se genera el mapa de profundidad. Estas cámaras suelen ir acompañadas de una cámara CMOS.

Este tipo de tecnología es muy usada en detección de personas, localización de paredes e identificación de objetos. Además estos sensores tienen un precio reducido y consumen por lo general poca potencia.

En contrapartida, hay que mencionar que este tipo de sensores no son eficientes bajo luz solar intensa, debido a que los infrarrojos interferentes pueden provocar medidas erróneas o falta de medición.

### 2.3.4.2 Ejemplo de sensor

Como ejemplo de cámara 3D infrarroja por mallado cabe destacar el fabricante PRIME SENSE y su modelo Carmine [21] o la Kinect de Microsoft [22]. En este caso, se analizará el primer modelo mencionado de manera breve:

	<b>Apertura (grados)</b>	<b>Resolución cámara (píxeles)</b>	<b>Rango de detección (metros)</b>	<b>Precisión en profundidad (metros)</b>
	57.5x45	640x480	0.8-3.5	±0.012

Para exponer un ejemplo de cámara basada en tiempo de vuelo se puede recurrir al fabricante MESA [23], y en concreto al modelo SR4500 [24], el cual es brevemente explicado a continuación:

	<b>Apertura (grados)</b>	<b>Imágenes por segundo</b>	<b>Rango de detección (metros)</b>	<b>Precisión (metros)</b>
	69 horizontal 56 vertical	10-30	0-9	±0.02 hasta 5 metros ±0.04 hasta 9 metros

## 2.3.5 LIDAR

La palabra LIDAR viene de “Light Detection And Ranging” y son ampliamente utilizados en la navegación autónoma gracias a sus fantásticas características, como la alta velocidad de adquisición de datos, independencia a la variación de visibilidad, su precisión y su fiabilidad [1] [2].

La mayoría de los láseres utilizados para este objetivo son de clase 1. Es decir, son seguros en condiciones de utilización previsible, incluyendo el uso de instrumentos ópticos para visión intra-haz.

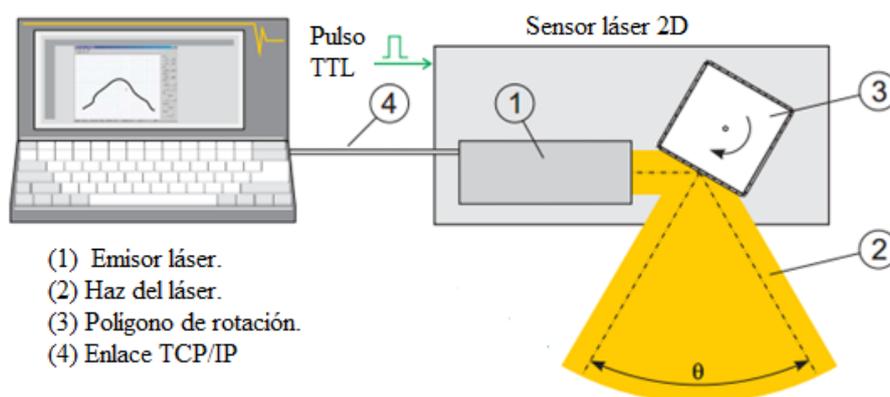
Un sistema basado en LIDAR tiene las siguientes ventajas: Exactitud en las medidas, que son invariantes a la intensidad de luz. El alineamiento del dispositivo no es crítico. Fácil configuración de la apertura angular deseable, ignorando aquellos datos que no se quieran analizar. Rápido procesamiento de datos y por último, bajo ratio de falsos positivos que otros sistemas. [25]

### 2.3.5.1 Funcionamiento

Esta tecnología consta de un emisor y receptor láser y utilizan el tiempo de vuelo para determinar la distancia al objeto.

Hay dos grandes familias de láseres LIDAR: los láseres 2D y los láseres 3D. La adquisición de datos en un sistema 2D está basada en un diodo láser que emite su rayo a través de un espejo giratorio, consiguiendo un conjunto de puntos en el plano bidimensional al ser captados por el receptor. Por tanto el sensor láser hace barridos o escaneos en el espacio bidimensional cada muy poco tiempo.

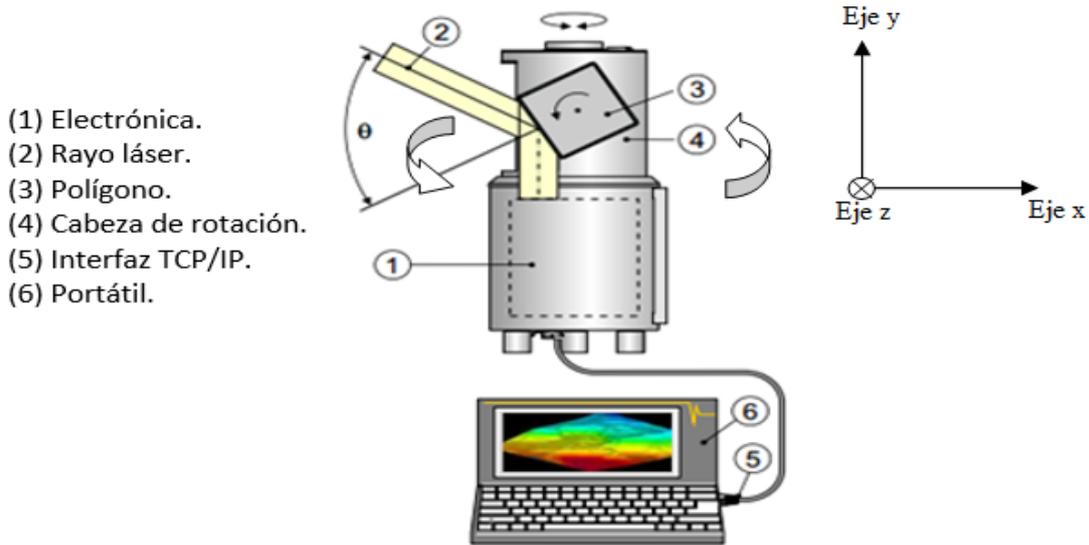
Para entender mejor esto, se incorpora la siguiente imagen en la que estos conocimientos aparecen esquematizados:



**Figura 2-5: Funcionamiento sistema LIDAR 2D**

Para conseguir el efecto 3D, se puede hacer un apilamiento de láseres 2D con diferentes ángulos de orientación o bien elaborando un sistema adicional de rotación en el sensor 2D, de manera que el polígono de rotación obtenga datos bidimensionales y el sistema de giro se encargue de ir moviendo el sensor láser a diferentes alturas. Este último resulta ser el más complejo.

En la siguiente figura se puede ver un sensor 3D real creado a partir de un sensor 2D más un sistema de rotación adicional.



**Figura 2-6: Funcionamiento sistema LIDAR 3D**

Esta configuración hace que al rotar el polígono se obtenga un barrido a lo largo del eje x e y. Si a esto se le añade la rotación adicional mencionada anteriormente con respecto al eje y se obtendrán puntos con valores no nulos en el eje z, consiguiendo puntos tridimensionales.

### 2.3.5.2 Ejemplo de sensor

Como ejemplo de láser 2D se tomará el modelo 30LX [26] de la marca HOKUYO [27]:

	Apertura (grados)	Resolución angular (grados)	Rango de detección (metros)	Precisión (metros)
	270	0.25	0.1-30	±0.03 hasta 10 metros ±0.06 hasta 30 metros

Un ejemplo de láser 3D con plataforma de giro es el modelo LMS-Z210II-S [28] de la marca RIEGL [29]:

	Apertura (grados)	Resolución angular (grados)	Rango de detección (metros)	Precisión (metros)
	95 en vertical 360 en horizontal	0.01	4-150	±0.015

Por último, un ejemplo de sensor 3D basado en el apilamiento de sensores 2D y a su vez montado en una plataforma de giro es el HDL-64E [30] de VELODYNE [31]. Este modelo posee 64 láseres, apilados en dos filas de 32 o en 4 de 16 cada una. Esta marca es una de las más prestigiosas debido a que sus sensores son, en casi todos los aspectos, los mejores del mercado. Como se puede prever, son los sensores más caros del mercado.

	Apertura (grados)	Resolución angular (grados)	Rango de distancia (metros)	Puntos por segundo (millones)
	360 horizontal 26.8 vertical	0.09	Hasta 120	>1.333

## 2.4 Prerrequisitos

### 2.4.1 Introducción

A la hora de escoger un sistema sensorial, es importante establecer unos requisitos cuyo cumplimiento es imprescindible antes de comenzar a analizar en detalle todos los sensores diferentes que existen. Para ello, a continuación se va a explicar brevemente algunos conceptos que deben ser conocidos para entender dichos requisitos.

No obstante si se quiere ver un desglose más detallada que lo anteriormente expuesto y con varios modelos adicionales se puede consultar la tabla de características del Anexo 1.

### 2.4.2 Grado de protección

El grado de protección (IP) es el nivel de protección proporcionado por la envolvente del dispositivo contra la penetración de cuerpos sólidos, contra la penetración de agua o contra los impactos mecánicos exteriores.

Esta característica es verificada mediante métodos de ensayo normalizados. Posee un sistema de codificación que se encuentran descritos en la norma europea EN 60529.

Este código IP está formado por dos números de una cifra cada uno de ellos, situados después de las letras “IP” y son totalmente independientes entre sí.

El número que va en primer lugar indica la protección del equipo contra la penetración de cuerpos sólidos. Está graduada desde 0 hasta 6, siendo este ultimo valor el que más protección ejerce.

El desglose y explicación de cada valor numérico se describe a continuación:

Grado de protección	
Valor	Descripción
0	No protegida
1	Protegida contra cuerpos sólidos de más de 50 mm
2	Protegida contra cuerpos sólidos de más de 12 mm
3	Protegida contra cuerpos sólidos de más de 2.5 mm
4	Protegida contra cuerpos sólidos de más de 1 mm
5	Protegida contra la penetración de polvo
6	Totalmente protegido de la entrada de polvo

**Tabla 2-1: Descripción primer parámetro IP**

El número en segundo lugar, indica la protección del sistema a la entrada de agua. Esta vez la numeración está comprendida entre 0 y 8, siendo este último número el grado de protección más alto. Los detalles de esta numeración se pueden ver a continuación:

Grado de protección	
Valor	Descripción
0	No protegida
1	Protegida contra la caída vertical de gotas de agua
2	Protegida contra la caída de gotas de agua con una inclinación máxima de 15°
3	Protegida contra la lluvia fina (pulverizada)
4	Protegida contra las proyecciones de agua
5	Protegida contra los chorros de agua
6	Protegida contra fuertes chorros de agua o contra la mar gruesa
7	Protegida contra los efectos de inmersión
8	Protegida contra la inmersión prolongada

**Tabla 2-2: Descripción segundo parámetros IP**

Al formar parte de un estándar, dichos valores pueden ser verificados para asegurar que la protección sea la deseable.

### 2.4.3 Accesibilidad

Es importante conocer cómo acceder a los datos proporcionados por el sensor, y de sus posibles protocolos asociados. La manera de acceso a esta información puede ser por cable Ethernet, USB, RS232, RS442 o CAN BUS. Este enlace debe ser seguro y estable, capaz de poder intercambiar información tan rápido como sea requerido por la cantidad de información disponible.

El USB, en particular la versión 2.0 puede soportar hasta velocidades de 480 Mbps, pero esta velocidad es compartida por todos los dispositivos conectados por este método a un ordenador o sistema.

En cambio, si la conexión se realiza por un cable Ethernet esto no ocurre debido a que dicha velocidad es invariante a cuántos dispositivos haya conectados a un ordenador. La mayoría de los sensores tienen Fast Ethernet, que opera a 100 Mbps.

El estándar RS232 puede operar en distancias de hasta 15 metros, siendo su velocidad máxima dependiente de dicha distancia y de las condiciones de la línea. Su interfaz suele operar a unos 38,4 Kbps sobre distancias muy cortas. [32]

El estándar RS442, disponible en algunos sensores, alcanza velocidades de transmisión de hasta 10 Mbps a 12 metros y puede llegar a una gran distancia con una velocidad de centenares de Kbps. [32]

#### **2.4.4 Establecimiento de requisitos**

Una vez explicado brevemente el grado de protección y la accesibilidad se pueden establecer unos requisitos fundamentales para la elección del sensor. Estos requisitos son:

- El sensor debe tener al menos IP 64: El sensor va a trabajar en entornos exteriores y debe tener una protección frente al polvo o agua. De otra manera, dicho sensor no se podría utilizar en exteriores, o tendría un periodo de vida útil muy corto.
- Se requiere que tenga conexión por cable Ethernet. Es decir, se quiere que el flujo de datos desde el sensor hasta el software sea rápido y no dependa de otras conexiones.
- El alcance máximo de medición del sensor debe ser superior a 10 metros. No se puede olvidar que la elección del sistema sensorial debe ser válido para la navegación autónoma en un vehículo real que irá circulando por una determinada vía. El sistema debe ser capaz de detectar obstáculos a una distancia prudente para poder analizar el entorno y tomar futuras decisiones en cuanto a su velocidad y posición.
- Se requiere que la intensidad lumínica no sea un problema grave para la toma de medidas de distancia en el entorno. El sistema debe ser lo más invariante posible a dichas condiciones.
- Se requiere que la toma de datos del sensor sea lo más simple y eficaz posible. Es decir, que el sensor sea capaz de enviar información de la distancia de los objetos sin que se necesite un alto coste computacional para dicha transformación. Esta norma hace que las cámaras estereoscópicas no deban seguir siendo estudiadas.

Con estos requisitos mínimos, se reduce notablemente el gran abanico de posibilidades que elección del sistema sensorial, quedándose fuera los ultrasonidos estudiados, las cámaras 3D y las cámaras estereoscópicas.

### **2.5 Requisitos deseables**

#### **2.5.1 Introducción**

Una vez definidos los requisitos previos, ahora se exponen aquellas características que se consideran más apropiadas para la elección del sistema sensorial. A continuación se explicarán dichos requisitos deseables y se elaborará una tabla de estudio con los sensores más característicos de cada tecnología válida.

### **2.5.2 Generación de medidas 3D del entorno**

La razón de este requisito es que se debe ser capaz de detectar los dos tipos de obstáculos que se puede encontrar el vehículo: los obstáculos positivos y negativos.

Se considera que los obstáculos positivos son aquellos que poseen una determinada altura sobre el suelo por donde se está circulando.

Los obstáculos negativos son aquellos que tienen una profundidad bajo el suelo por el que se está circulando, por ejemplo un bache en la carretera.

Además se debe poder obtener información sobre pendientes positivas o negativas que puedan surgir alrededor del vehículo, que pueden constituir un obstáculo o en caso contrario, un camino navegable.

Por tanto, el objetivo es que el sistema elegido sea capaz de obtener información 3D sobre estos tipos de obstáculos y dichas pendientes para poder reaccionar a tiempo.

### **2.5.3 Rango de distancias y margen de errores**

Con objetivo de poseer un cierto margen de reacción contra los obstáculos que puedan surgir en la trayectoria del vehículo, se desea que la distancia de detección sea mayor de 20 metros en frente del vehículo.

Además se requiere que el sensor sea fiable y robusto. Por ello se estipula que a 20 metros de distancia se tenga un error menor que  $\pm 5$  centímetros.

### **2.5.4 Apertura angular**

Define en cuantos grados se puede obtener información desde el sensor. A la hora de elegir una apertura, hay que preguntarse cuánta de la posible información del entorno alrededor del vehículo es imprescindible. Se ha estimado que por lo menos se debe captar información de 180 grados en horizontal, es decir 90 grados a cada lado.

### **2.5.5 Velocidad de barrido**

Se debe estimar una velocidad a la cual se vuelva a empezar a tomar medidas desde el punto inicial. Interesa que esta velocidad sea elevada para poder prevenir posibles obstáculos espurios que puedan surgir. Esto conlleva a que se tendrán muchos datos en poco tiempo en la apertura definida, con lo que se pueden detectar obstáculos en tiempo real.

### **2.5.6 Intensidad de luz**

El sistema debe ser capaz de tomar medidas fiables aun cuando esté enfocado directamente al sol. La intensidad de luz se mide en lux y el sol puede llegar a tener 100.000 lux.

### **2.5.7 IP**

Cuanto más protegido esté el sistema mejor. Por tanto, se ha considerado que un IP 67 es una buena característica que debe tener el sistema. Así se puede evitar que el sistema deje de funcionar adecuadamente y garantiza su correcto funcionamiento incluso en situaciones meteorológicas adversas.

### **2.5.8 Resolución angular**

Es deseable que el sensor elegido obtenga información acerca del entorno con muy poco incremento angular entre medidas contiguas, de manera que se pueda obtener mucha exactitud. Se considera que tomar datos con  $0.5^\circ$  en horizontal de diferencia entre dos puntos es más que suficiente para lograr una buena densidad de puntos en el espacio.

### **2.5.9 Coste**

Dicho sensor no debe costar más de 5.000 dólares. Por tanto, sensores como los de Velodyne (75.000 dólares) no están al alcance, aunque cumplan con todas las características previas y deseables con mucho margen. Esto hace que no sea posible elegir el mejor sensor para este ámbito, pero se sigue pudiendo adaptar el sistema a otros sensores que por un precio mucho menor, pueden cumplir con las expectativas. No obstante, se puede aceptar un cierto margen de desviación sobre el precio puesto inicialmente.

### **2.5.10 Tabla de comprobación requisitos deseables**

Una vez explicado cada uno de los requisitos deseables, sólo queda estudiar los sensores y verificar si se cumplen dichas propuestas. Con el fin de simplificar la información, que a veces puede resultar difícil de analizar, se ha decidido que en la tabla siguiente, en lugar de ser como la del Anexo 1 pero quitando los sensores que no cumplan los requisitos previos, sea simplemente de marcación de si se cumplen o no los requisitos, de modo que con un simple vistazo se pueda saber mucha información sin tener que leer los diferentes campos numéricos.

En las hojas de datos de los sensores, a menudo no aparece cierta información requerida para realizar este estudio. En este caso, en el campo del que no se tenga conocimiento sobre su cumplimiento se pondrá “No espec”, es decir, campo no especificado en la hoja de datos.

Muchos de los precios de los sensores son variables o están ocultos para que se le consulte al fabricante. A menudo dicho precio suele venir acompañado de descuentos si se compra un número elevado de sensores o si se van a aplicar para la docencia.

Aunque algunos de los parámetros no sean cumplidos, se buscarán posibles soluciones si se cree que el sensor es el más apropiado para este proyecto.

Para la realización de esta tabla, se han cogido los sensores más característicos y a priori mejores para dicho sistema. Esto es debido a que, incluso con todas las restricciones impuestas anteriormente, hay una gran variedad de sensores que podrían ser apropiados.

Fabricante	Sensor	Medidas 3D del entorno	Rango medida > 20 metros	Error 5 cm a 20 metros	Apertura 180°	V. barrido <50 ms	Intensidad luz 100K lux	IP 67 o superior	Resolución angular <0.5°	Coste < 5K \$
Hokuyo	UTM-30LX/LN	No	Si	Si	Si	Si	No	No	Si	Si
	URG-04LX-UG01	No	No	Si	Si	Si	No	No	Si	Si
	UTM-30LX-EW	No	Si	Si	Si	Si	Si	Si	Si	Si
Sick	TIC 102	No	No	Si	Si	Si	No espec.	No	Si	Si
	LMS511-10100 PRO	No	Si	Si	Si	Si	No espec.	Si	Si	No
	LMS151-10100	No	No	Si	Si	Si	No espec.	Si	Si	No espec.
	MSR400001	No	Si	Si	No	Si	No espec.	Si	Si	No
RIEGL	LMS-Q20	No	No	Si	No	Si	No espec.	No	Si	Si
	VQ-450	No	Si	Si	Si	Si	No espec.	No	Si	No
	LMS-Z210	Si	Si	Si	Si	No	No espec.	No	Si	No
Velodyne	HLD-32E	Si	Si	Si	Si	Si	Si	Si	Si	No
	HLD-64E	Si	Si	Si	Si	Si	Si	Si	Si	No
IBEO	LUX HD	Si	Si	Si	No	Si	No espec.	Si	Si	No
BOSCH	LRR3	No	Si	No espec.	Si	Si	Si	No	Si	Si

*Tabla 2-3: Desglose del cumplimiento de requisitos deseables*

	Total "Sí"	Total "No"	Total "No espec."
<b>UTM-30LX</b>	6	3	0
<b>URG-04LX-UG01</b>	5	4	0
<b>UTM30LX-EW</b>	8	1	0
<b>TIC 102</b>	5	3	1
<b>LMS511-PRO</b>	6	2	1
<b>LMS151</b>	5	2	2
<b>MSR400001</b>	5	3	1
<b>LMS-Q20</b>	4	4	1
<b>VQ-450</b>	5	3	1
<b>LRR3</b>	5	1	3

***Tabla 2-4: Resumen del cumplimiento de requisitos deseables***

A la vista de las tablas anteriores y de las ventajas y desventajas desglosadas anteriormente de cada tipo de tecnología se ha llegado a la conclusión de que el sistema sensorial debe estar formado por un LIDAR. Ahora bien, se debe de buscar la forma de que se cumplan todos los requisitos deseables, de manera que el sistema sensorial elegido sea idóneo para los objetivos marcados por este proyecto. Entre todos los sensores expuestos anteriormente, hay uno, el HOKUYO 30LX-EW que solamente presenta el inconveniente de que dicho sensor por sí solo no logra adquirir información 3D del entorno, pero aún así es el sensor que más cerca está de conseguir dicho objetivo y merece seguir teniendo presente esto a la hora de realizar propuestas de sistemas sensoriales.

## 3 Propuestas preliminares

### 3.1 Introducción

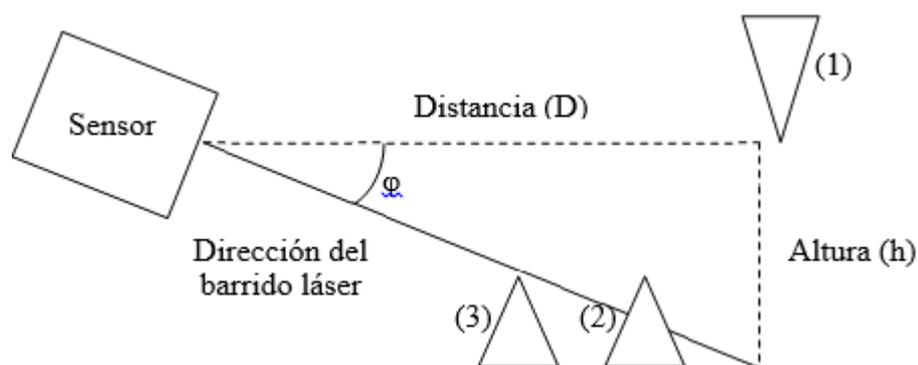
En este capítulo se van a estudiar diferentes procedimientos para lograr que el sistema sensorial resultante se adapte a la perfección a los requerimientos establecidos. A medida que se van explicando las propuestas, se irán evaluando las posibles ventajas y desventajas de cada una de ellas, hasta encontrar una que cumpla con la detección de obstáculos (positivos y negativos) y pendientes en el entorno de  $90^\circ$  a cada lado del vehículo.

Estas propuestas deben ir acompañadas de la posición y la inclinación con las que deben ser instaladas en el vehículo proporcionado.

### 3.2 Propuesta 1: Un sensor LIDAR con inclinación

Esta propuesta consta del sensor láser HOKUYO 30LX-EW [], instalado encima del parachoques del vehículo, de manera que no sobresalga de él y quede protegido frente a impactos. La idea es montarlo con una cierta inclinación hacia abajo, de manera que se consiga un efecto 2.5D.

Un efecto 2.5D consta de datos en dos dimensiones y a medida que la distancia ( $D$ ) al laser varía, se consigue una tercera componente, correspondiente al cálculo de la altura ( $h$ ) sabiendo el ángulo de inclinación ( $\varphi$ ) y la distancia horizontal ( $D$ ) al objeto detectado.



**Figura 3-1: Representación de la configuración propuesta y posibles obstáculos emergentes**

No obstante, este efecto es insuficiente para poder generar medidas 3D, debido a que puede que algunos obstáculos (positivos o negativos) no sean detectados a tiempo para reaccionar en consecuencia, o simplemente nunca sean detectados.

Un ejemplo de este último caso es cualquier obstáculo colgante del techo (1), cuya altura (respecto del suelo) sea igual o superior a la del sensor. Dicho obstáculo en ningún momento sería detectado, por tanto no se actuaría en consecuencia y el vehículo colisionará con él.

Con esta configuración, un obstáculo detectado a una distancia  $D$  puede dejar de ser detectado cuando el vehículo avance, aunque dicho obstáculo sea estático. Esto se puede ver en (2) y (3). Por tanto, no se consigue proporcionar información completa del entorno.

Como consecuencia de esta instalación, se puede llegar a perder alcance de detección al apuntar hacia el suelo. Además, no basta con que el obstáculo sea detectado, sino que hay que saber qué altura tiene y no a qué altura fue detectado, único parámetro que ofrece esta propuesta.

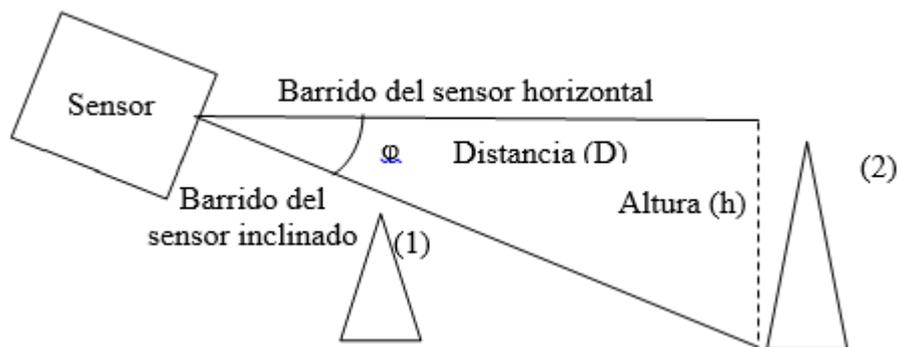
Con este sistema, sí se podrían detectar tanto pendientes negativas como positivas, pero es una de los pocos puntos a favor de esta propuesta.

Por todo esto, dicha propuesta se considera inválida.

### 3.3 Propuesta 2: Dos sensores LIDAR. Uno de ellos, con inclinación y el otro horizontal

Siguiendo con la misma instalación que la propuesta 1, se estima que el mejor lugar de instalación de ambos sensores es encima del parachoques, uno al lado del otro, de manera que ambos estén protegidos frente a impactos.

En este sistema se intenta paliar algunos defectos de la propuesta 1, con un sensor adicional sin inclinación. De manera que el obstáculo (1), si su altura con respecto a la horizontal es igual o menor que “h”, sí sería detectado. Además, se obtendría el doble de información en la periferia del vehículo: el sensor inclinado obtendría datos en 2.5D y el adicional completaría dicha información en el plano bidimensional, sin reducir el alcance del sensor.



**Figura 3-2: Representación de la configuración propuesta y posibles obstáculos emergentes**

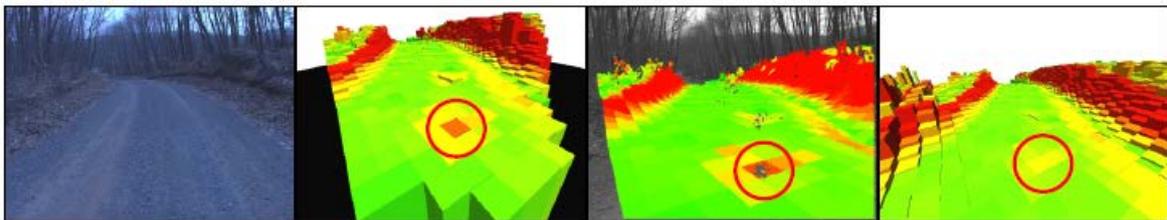
En este caso, el obstáculo (1) sigue sin ser detectado y en el caso del obstáculo (2), en cuanto esté dentro del alcance del sensor, se podrá detectar pero de manera discontinua. Es decir, entre medias de los dos escáneres no se sabe si sigue habiendo obstáculo hasta que el vehículo no avance más en dirección a dicho obstáculo y sea captado en otro punto de altura con el sensor inclinado.

No obstante, tampoco se consigue una generación de medidas 3D tal y como se desea, pero se va acercando más al objetivo. El problema reside en que esta propuesta hace que sea necesario comprar dos sensores LIDAR, por lo que el coste se dispara. Además, hay sensores 3D láser (como los del fabricante RIEGL) que serían más convenientes si se amplía el presupuesto, por tanto esta propuesta va en contra de los requisitos deseables y debe ser rechazada o en su defecto cambiar el coste.

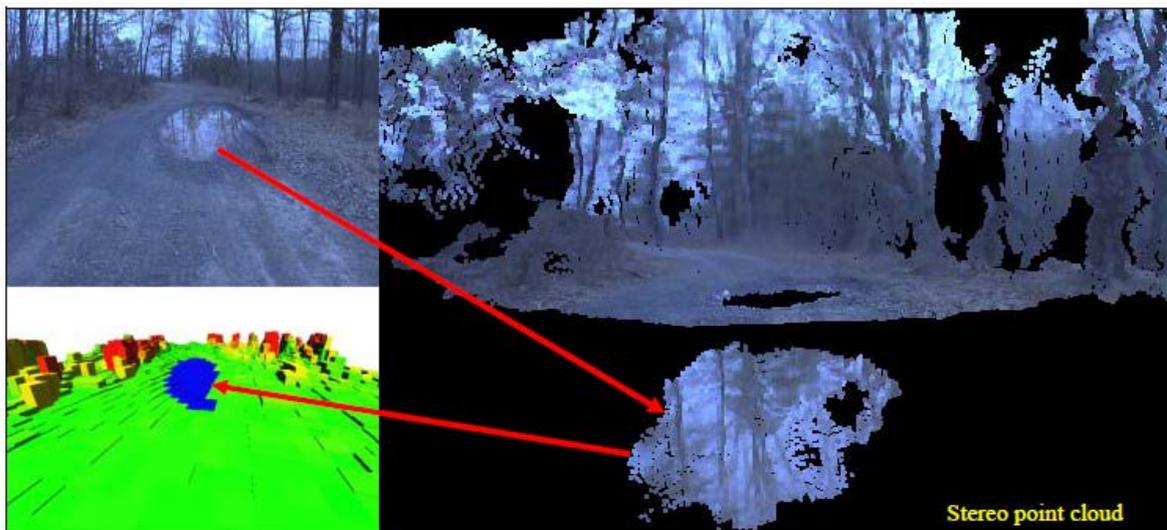
### 3.4 Propuesta 3: Sensor LIDAR con inclinación y una cámara estereoscópica

Esta propuesta basada en la primera y una cámara estereoscópica cumple perfectamente con los requisitos estipulados. La fusión de ambos tipos de información hace que el sistema sea más robusto, los defectos de la cámara estereoscópica son compensados con la complementación de los datos láser. La cámara consigue captar el entorno en 3D y el sensor láser puede “rellenar” o completar dicha información, aunque de manera incompleta como se vio en la propuesta 1.

Por ejemplo, en el artículo [19] utilizan una cámara estereoscópica para la navegación y se pueden ver las siguientes figuras:



*Figura 3-3: Fotografía y resultados del procesamiento de datos en un camino*



*Figura 3-4: Fotografía y resultados del procesamiento de datos en un camino con un charco*

En la figura 3-3, la cámara estereoscópica ha captado una serie de datos erróneos que han generado un obstáculo donde no lo hay (parte redondeada en las imágenes). Esto puede ocurrir de manera más frecuente de lo que gustaría. Con el sistema propuesto, antes de considerar un obstáculo, se podrían comparar los resultados de la cámara y los del sensor, de manera que si los dos obtienen como resultado que es un obstáculo se marque como tal.

Este sistema también solucionaría el problema de la figura 3-4, donde un reflejo en un charco se ha considerado como un obstáculo por la cámara estereoscópica. El LIDAR, al ser invulnerable a este efecto, no sacaría como este resultado por lo que ese obstáculo sería eliminado, al no coincidir con los datos proporcionados con la cámara.

Ahora bien, las cámaras estereoscópicas fueron descartadas anteriormente y aunque se complemente perfectamente con el LIDAR surgen varios problemas adicionales:

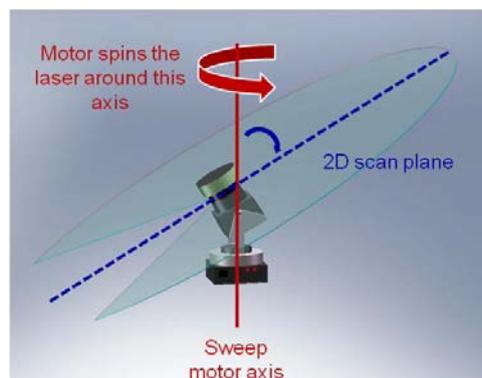
- El coste computacional de procesamiento de datos se dispara notablemente: No sólo se tiene que procesar los datos proporcionados por el sensor láser con los algoritmos necesarios si no que el procesamiento de imágenes necesita una algoritmia aún más pesada y costosa que la anterior.
- Se requiere una sincronización perfecta entre ambos tipos de datos proporcionados, de manera que se puedan complementar de manera adecuada. Un desfase en el tiempo de alguno de los dos sistemas resultaría en un caos en el que las medidas de ambos sistemas pueden ser contradictorias.

Todo esto sin olvidar que la toma de medidas y su procesamiento tiene que ser en tiempo real. Por todo ello, se estima que no es una propuesta viable desde el punto de vista computacional aunque podría ser interesante su utilización junto a un láser 3D más completo, lo que implicaría un coste elevado.

### 3.5 Propuesta 4: Sensor LIDAR con inclinación y una plataforma de giro

La idea es montar el sensor LIDAR 2D sobre una plataforma de giro, de manera que se obtengan puntos 3D de todo el entorno alcanzable por el rango de dicho sensor. Esta propuesta está basada en uno de los métodos de obtención de un láser 3D a partir de un láser 2D. Lo que se plantea en esta propuesta es que el sistema sensorial quede definido como un sensor láser 2D y una plataforma de giro concreta.

En artículos como [33] [34] [35] , se puede ver la ejecución de esta idea, pero en este caso no se optará por el giro mencionado en [33] en concreto (mostrado en la figura ) debido a que si el láser se gira de esa manera, obstáculos en frente del vehículo dejarán de ser detectados por un periodo de tiempo.



*Figura 3-5: Sistema de giro desarrollado en un artículo*

Por esto, en esta propuesta se van a diseñar métodos de giro en los cuales no se dejen de obtener datos delante del vehículo, para poder detectar lo antes posible los obstáculos que presentan una seria amenaza de colisión.

Hay varias posibilidades de girar el sensor láser para conseguir medidas 3D. Para explicarlo, se van a referenciar los diferentes giros respecto a un eje de rotación.

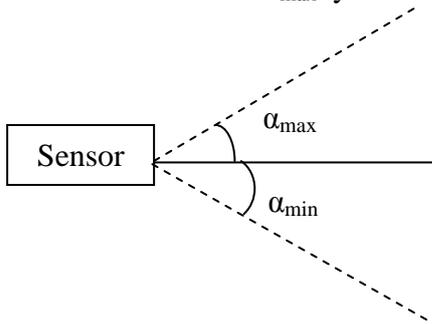


**Figura 3-6: Vehículo Yamaha Rhino**

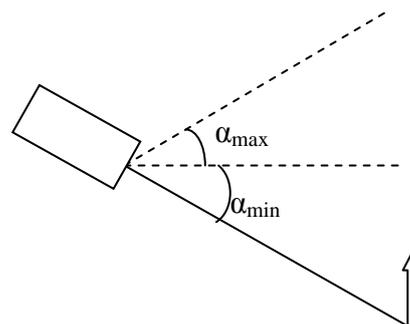
Teniendo en cuenta dicho sistema de coordenadas y asumiendo que el sensor LIDAR toma medidas bidimensionales a lo largo del eje “x” y “y” y que el vehículo si se mueve hacia delante, lo hará en dirección positiva del eje “x”, se procede a elaborar diferentes sistemas de giro para estudiar cual es el más apropiado. Obviamente, cualquier medida que se tome desde el sensor láser debe ir marcado por el ángulo de orientación que tenga en ese momento por el sistema de giro, a fin de poder construir correctamente los puntos 3D del entorno.

### 3.5.1 Giro con respecto al eje “y” de manera permanente o no permanente

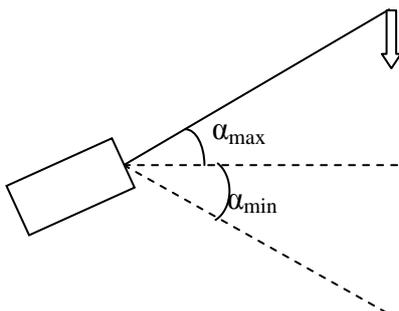
Si se procede a girar el sensor con respecto al eje “y” se conseguirán distintos valores de “z” a medida que el motor rote el sensor. Hacer este giro de manera permanente, es decir, el sistema de giro rota continuamente el sensor respecto al eje “y”, sería un error debido a que cuando se hayan recorrido 180° el sensor apuntaría hacia el mismo vehículo y se perdería información de lo que hay delante de él. Por tanto, con este giro con respecto al eje “y” sólo se puede plantear un giro no permanente. Es decir, una vez alcanzada una inclinación máxima o mínima, desacelerar dicha orientación de giro y comenzar a girar en sentido contrario. En las siguientes figuras se puede ver un ejemplo de esto, con inclinación máxima  $\alpha_{max}$  y mínima  $\alpha_{min}$ :



**Figura 3-7: Situación inicial**



**Figura 3-8: Una vez alcanzado  $\alpha_{min}$  se cambia el sentido de giro**

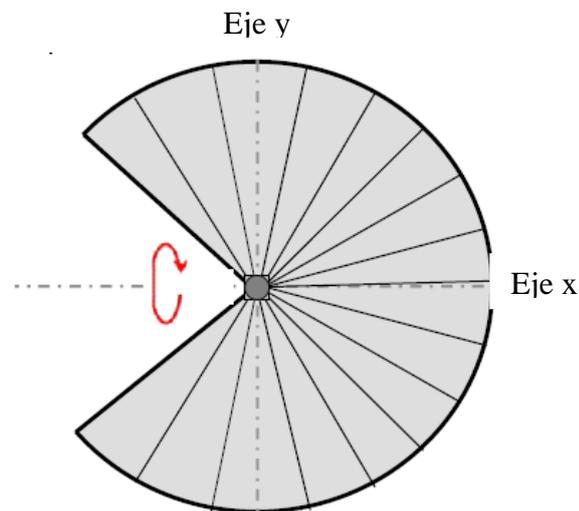


**Figura 3-9: Una vez alcanzado  $\alpha_{max}$  se vuelve a cambiar el sentido de giro**

Este método sigue sin obtener datos de manera continua de lo que hay en frente del vehículo, al tener que pasar por un ángulo máximo o mínimo para invertir el giro. Además, el hecho de estar continuamente desacelerando y acelerando el giro en diferentes sentidos puede hacer que se produzcan desajustes entre los datos del sensor y el ángulo de inclinación. Todo esto desencadena en el estudio de otro sistema de giro para explorar todas las posibilidades.

### 3.5.1 Giro con respecto al eje “x” de manera permanente

Lo que se plantea en este apartado es girar siempre en una dirección, es decir sin continuas desaceleraciones y aceleraciones en sentido contrario. Recordando que el sensor toma medidas bidimensionales en los ejes “x” e “y”, si se girara el conjunto con respecto al eje “x” se conseguiría que siempre haya una serie de puntos que toma distancias en el entorno en frente del vehículo y además dichos con todos los puntos tomados por el sensor y los datos proporcionados por el sistema de giro se puede conseguir generar puntos 3D en toda la región en donde el sensor láser tome información. Con el objetivo de que quede claro dicho concepto, se adjunta la siguiente figura:



**Figura 3-10: Barrido láser**

Ambos tipos de obstáculos y pendientes pueden ser localizadas siempre y cuando se encuentren dentro del rango de medidas del sensor. Si dichos obstáculos se encuentran en frente del vehículo, en ningún momento se dejará de sacar información de dicho obstáculo gracias a este sistema de giro, si no que dicha información se irá completando con aún más datos hasta tener perfectamente definido el obstáculo.

Comparado con el resto de propuestas, esta sin duda es la que más se ajusta: soluciona la generación de puntos 3D sin incrementar notablemente el coste, siempre se obtienen datos relevantes del entorno y permite que la mayor densidad de puntos tomados por el sensor láser esté en frente del vehículo, sin duda la región más importante.

Por ello, esta propuesta con el sistema de giro planteado es la opción elegida. Sólo queda definir qué sensor láser 2D será el usado. Consultando de nuevo la tabla 31313, se puede ver que el 2D LIDAR que más se adapta a los requisitos es el HOKUYO 30LX-EW [36].

## 3.6 Estudio previo a la realización de la propuesta elegida

### 3.6.1 Introducción

Antes de comenzar a desarrollar la propuesta elegida, se va a realizar una serie de simulaciones con las que se podrá comprobar si el funcionamiento del sistema es el deseado y si hay limitaciones de giro que se deban tener en cuenta.

Para realizar dicho estudio, se ha optado por el entorno MATLAB para simular el sensor y el sistema de giro. MATLAB es un lenguaje de alto nivel que ofrece un entorno interactivo para el cálculo número, además de que facilita la representación numérica y la elaboración de algoritmos [37]. En este lenguaje se programará las características de captación de datos del sensor (tales como resolución angular, apertura, velocidad de barrido...), de manera que los resultados obtenidos sean lo más fiables y reales posibles.

Una vez elaborado el sistema en este entorno, se estudiará cual es la densidad de puntos obtenida en cada región del espacio dependiendo de la velocidad de giro impuesta, y qué efectos tiene la velocidad de giro en tales medidas.

### 3.6.2 Desarrollo del análisis

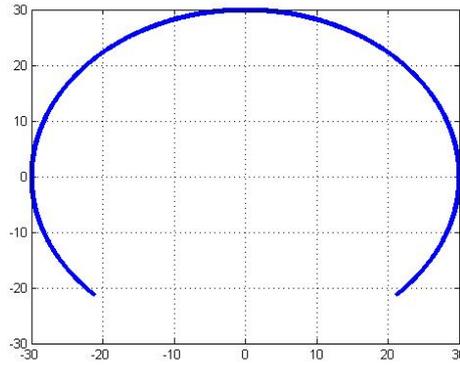
La programación de este sistema tiene que tener en cuenta los reales del sensor HOKUYO 30 LX-EW, especificados en su hoja de datos. A continuación se mencionan aquellos datos del sensor que son usados en el programa para la realización del análisis:

- Número de puntos en un barrido láser: 1080 puntos.
- Tiempo que tarda en láser en hacer un barrido: 25 milisegundos.
- Apertura angular del sensor láser: 270°.
- Resolución angular: 0.25°.
- Rango máximo: 30 metros.

Mediante un cálculo sencillo se puede saber cuál es la exactitud del sensor a 30 metros de distancia. Es decir, cuanta separación se produce entre la toma de dos puntos consecutivos:

$$\text{Exactitud} = 2\pi * (0.25^\circ / 360^\circ) * 30 \text{ metros} = 0.1309 \text{ metros}$$

En la siguiente figura se puede ver un único barrido láser con estas características, con los datos obtenidos en color azul, estando el sensor situado en el centro de coordenadas (posición (0,0)):

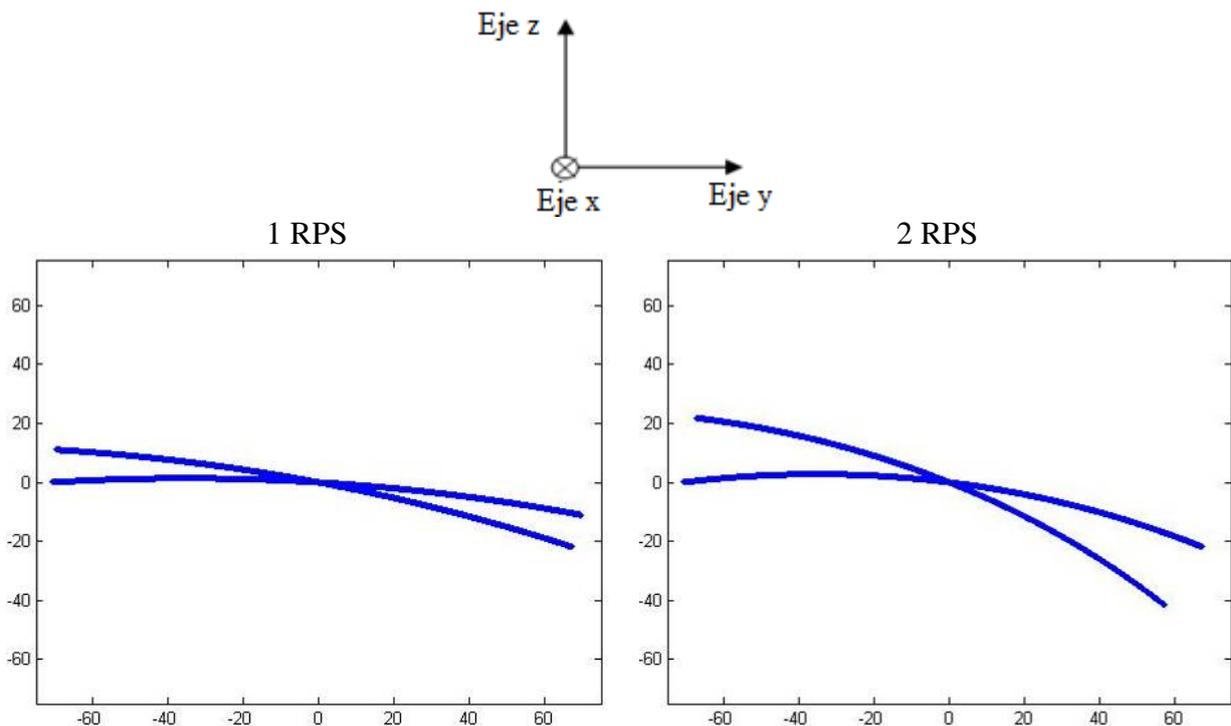


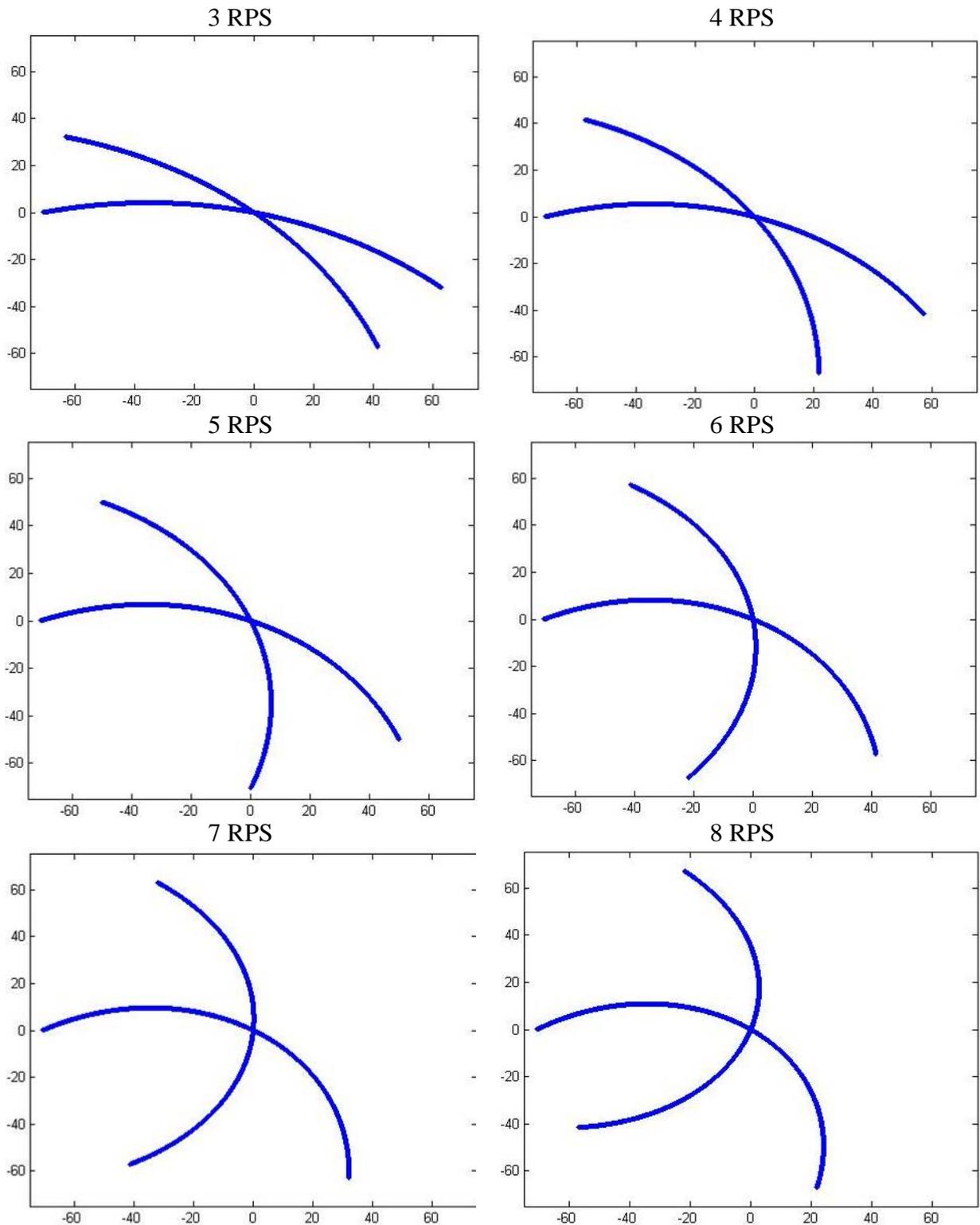
**Figura 3-11: Barrido láser simulado en Matlab**

Simulado ya un barrido bidimensional del sistema, se va a comenzar a rotar el sistema con distintas velocidades y tratar de ver qué efectos ocurren en el entorno. Para poder visualizar de manera adecuada los datos obtenidos, los datos 3D generados se proyectarán en 2D para que algunas características puedan ser vistas de manera más clara. No obstante, no hay que olvidar que el motor hará que el escáner mostrado en la figura anterior rote de manera que en el eje “z” se comiencen a tomar medidas.

### 3.6.2.1 Estudio de dos barridos laser con diferentes velocidades de giro

En este estudio, se van a representar dos barridos láser con diferentes velocidades de giro, con el objetivo de estipular una velocidad máxima de giro y qué efectos producen en las medidas las diferentes velocidades. Dichas velocidades se miden en revoluciones por segundo (RPS). En cuanto a la distancia medida por el sensor, se va a suponer el caso en que la distancia sea máxima (30 metros) de manera que si se produce algún tipo de efecto en las medidas, a la máxima distancia debería ser el caso peor de los efectos que puedan surgir. Conviene recordar el sistema de coordenadas en cada caso para que se entienda bien cada una de las figuras, y no olvidar que se está planteando un giro respecto al eje “x”:





**Figura 3-12: Representación dos barridos láser con diferentes velocidades de giro**

Como se puede observar, a medida que la velocidad de giro va aumentando, el barrido láser se va distorsionando de manera significativa. A 5 RPS, en dos barridos el sensor ha girado  $90^\circ$  ( $5 \text{ RPS} * 360^\circ * 0.05 \text{ segundos}$ ), lo que es totalmente innecesario debido a que entre ambos haces se ha dejado un contorno demasiado grande sin escanear. Por tanto sería conveniente ir rellenando el contorno poco a poco y por regiones que ir a una velocidad enorme. Es decir, es mejor que el sistema de una vuelta completa y que haya una buena densidad de medidas a que el sistema sea muy rápido y las regiones que se hayan escaneado sean arbitrarias.

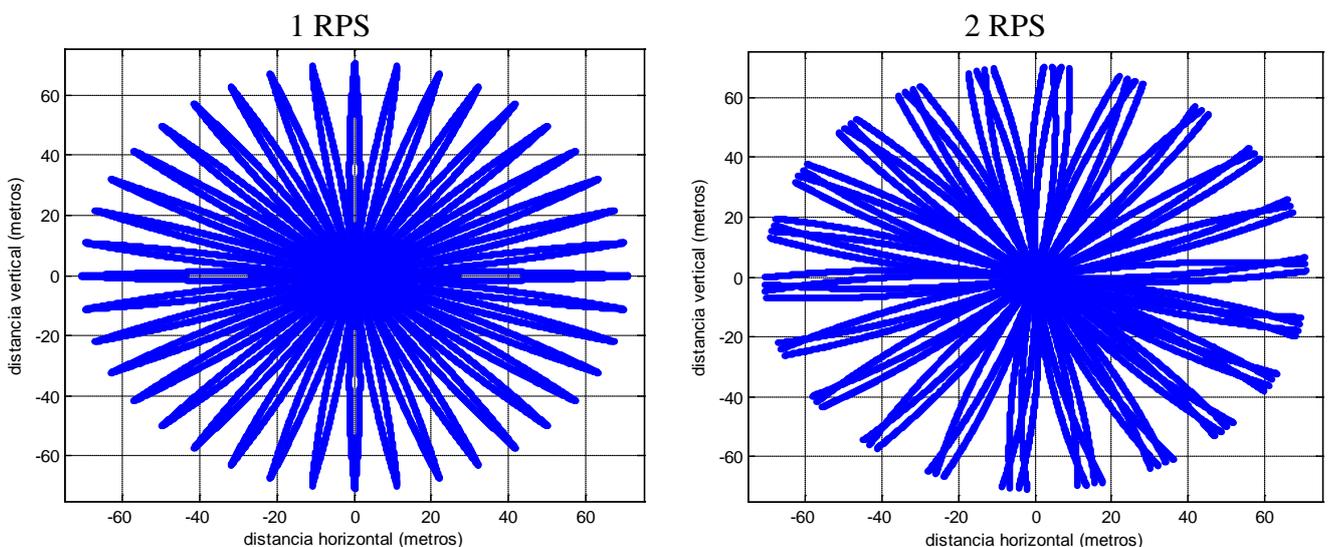
Por tanto, aquí se puede ver un compromiso entre querer girar lo más rápido posible pero teniendo una buena densidad de puntos y sin que el haz se distorsione notablemente. Como velocidad máxima de giro permisible, se ha estimado que 5 RPS es más que suficiente, y como mínima velocidad de giro, aquella que permita que en el eje vertical se pueda conseguir una resolución igual a la horizontal propia del láser, es decir  $0.25^\circ$  de separación entre puntos consecutivos. Por tanto, si 25 milisegundos después del primer haz, el siguiente barrido debe estar a  $0.25^\circ$  como máximo del anterior se obtiene que la velocidad mínima es: 1.62 revoluciones por minuto, es decir 0.027 RPS. No obstante, girar tan lento implicaría que el sistema nunca captase en tiempo real todo el entorno dentro de la apertura del láser que rodea al vehículo, aunque si se requiere mucha resolución en alguna situación estática, dicha velocidad podría ser aceptada.

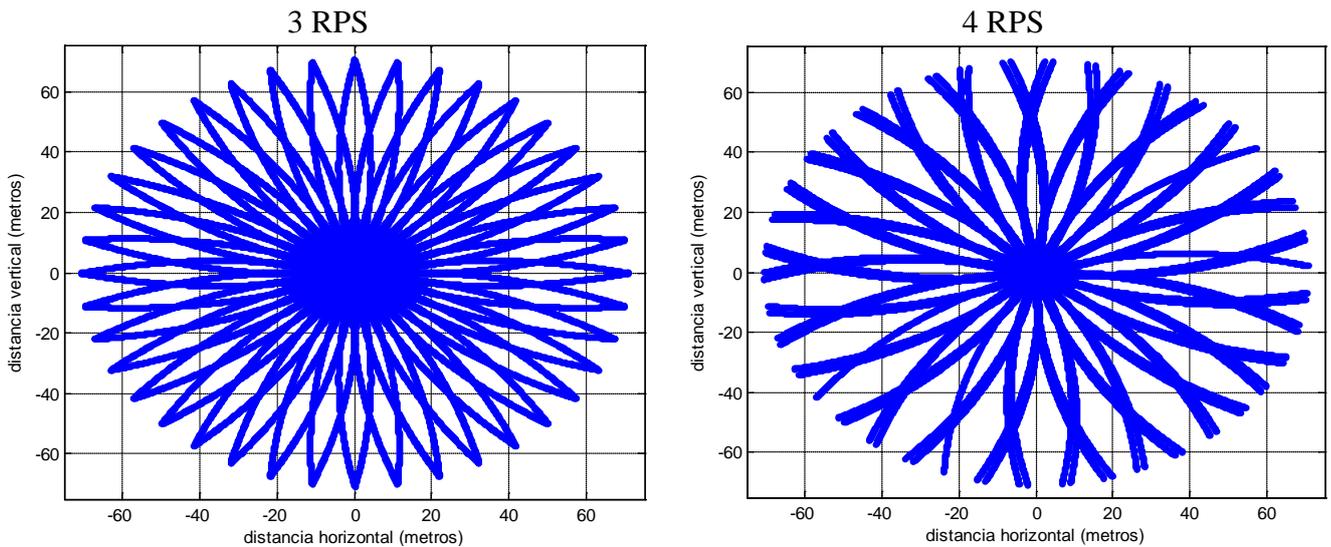
### 3.6.2.2 Estudio la velocidad óptima de giro

Solucionado ya la velocidad mínima y máxima de giro, sólo queda saber cuál es la velocidad de giro óptima del sistema y si lo es para cualquier velocidad del vehículo. Para solucionar estas preguntas, se va a realizar un estudio en el que se va a evaluar la densidad de puntos que se produce a diferentes velocidades.

Para medir la densidad de puntos y poder comparar las velocidades se va a realizar un mallado ajustable, de manera que se impone que debe de haber al menos un punto dentro de ese mallado. Por tanto, cuanto menor sea ese mallado, mejor es la densidad de puntos ofrecido por esa velocidad de giro. Con el fin de realizar un estudio completo, cada una de las velocidades se evaluará con el mismo tiempo transcurrido de toma de medidas y a diferentes alcances, es decir, no sólo a 30 metros, para poder estudiar el efecto de la velocidad en el sistema.

Como primer paso, se va a familiarizar al lector con los barridos realizados por el láser en un segundo y a las diferentes velocidades de giro, de modo que se pueda comprender perfectamente el efecto de la velocidad de giro en las medidas antes de continuar con el mallado posterior:





**Figura 3-13: Representación de los barridos láser tomados en un segundo**

Como se puede observar, se necesitan definir los mallados mínimos tal que haya al menos un punto dentro de él para poder sacar unas conclusiones claras. Además de los criterios de estudio de velocidad, mallado y distancia, se incorpora otro parámetro más: la apertura. Lo que se pretende descubrir es cuán de bueno es la resolución de las medidas en entornos cercanos al frontal del vehículo, región que más importancia tiene para la adquisición de datos. No se requiere un mallado de centímetros en regiones de 30 metros a la derecha o izquierda del vehículo, pero sí en el frontal, es decir a unos pocos metros horizontales. Igual ocurre con el eje “z”: no es necesario tener una precisión de centímetros a 30 metros encima del vehículo pero sí en unos metros de apertura en vertical.

Por estos motivos, se hace necesario establecer una apertura máxima de estudio del mallado mínimo. Dicha apertura será de  $\pm 5$  metros en la escala vertical y  $\pm 20$  metros en la escala horizontal. A las medidas que se tomen fuera de estos no se les aplicará el mallado, al considerar que están en una región prescindible.

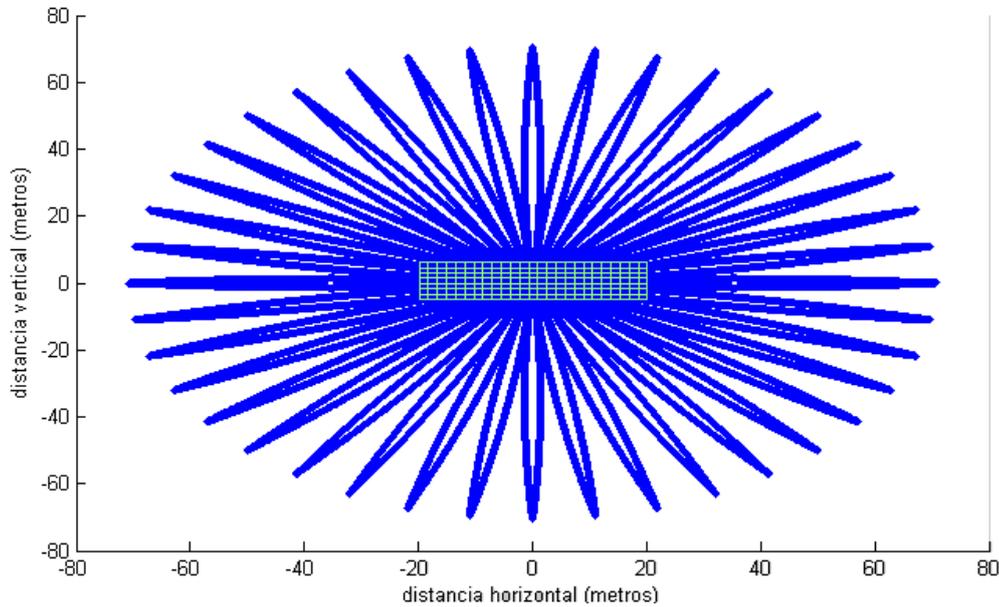
Por tanto, una vez definido el cuadrante de apertura en metros, hay que saber qué cuadrantes con una determinada apertura angular no deben ser estudiados. Para ello en la siguiente tabla se expone una relación de apertura angular en grados y distancia en metros en frente del vehículo, que da como resultado la apertura en metros equivalente en horizontal y vertical:

Grados	Distancia en frente del vehículo (metros)					APERTURA IMPUESTA
	30	20	10	5	1	
180°	94	62,8	31,415	15,7	3,14	Vertical $\pm 5$ metros Horizontal $\pm 10$ metros
90°	47	31,41	15,7	7,854	1,57	
45°	23	15,7	7,854	3,927	0,785	
30°	15	10,4	5,236	2,61	0,5236	

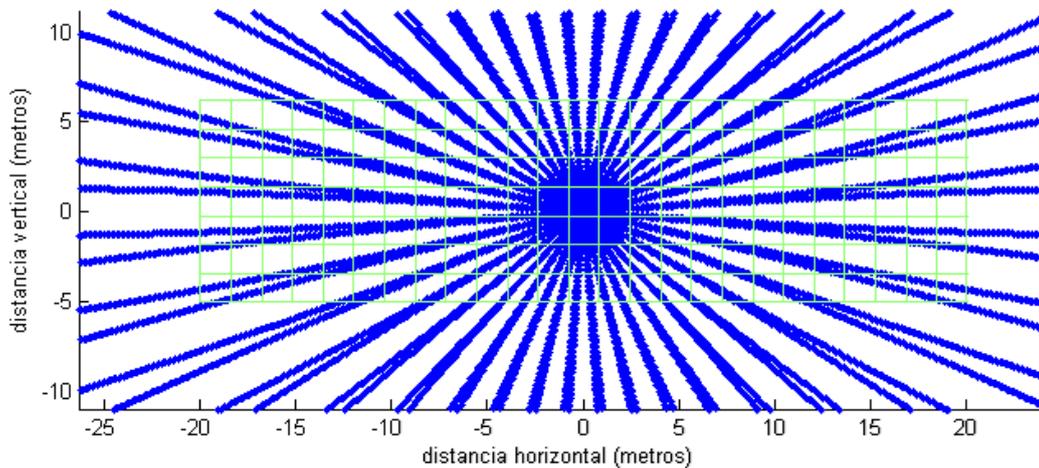
**Tabla 3-1: Relación entre apertura angular, distancia y apertura en metros**

Lo que se puede deducir de esta tabla es que, en caso de que la apertura en metros que corresponde a la relación entre distancia y apertura angular exceda la apertura impuesta, sólo se evaluará dicha apertura impuesta.

En caso de que la apertura bajo estudio (por ejemplo 5 metros y 45°) sea menor que la apertura impuesta, significa que la apertura bajo estudio se evaluará en su totalidad. A modo de ejemplo, en las siguientes figuras se expone el mallado (en color verde) mínimo en las regiones bajo estudio:



**Figura 3-14:** Representación barrido láser (azul) a 30 metros, 1 RPS y 1segundo y mallado (verde)



**Figura 3-15:** Representación barrido láser (azul) a 30 metros, 1 RPS y 1segundo y mallado (verde) en la apertura impuesta

A continuación se exponen tablas en las que aparece el mallado mínimo en metros para cada velocidad, distancia de evaluación (en metros) en frente del vehículo y apertura del láser (en grados), sin olvidar que como máximo se evalúa la apertura arriba impuesta. El formato en que aparece el mallado es X\*Y donde X es distancia en horizontal en metros e Y es distancia en vertical en metros.

<b>1 RPS</b>				
<b>Distancia\ Grados</b>	180 °	90 °	45 °	30 °
<b>30 m</b>	1,6*1,6	1,6*1,6	1*1,5	1*1
<b>20 m</b>	2*1,5	1,5*1,5	1*1	0,8*0,8
<b>10 m</b>	2*2	0,9*0,9	0,5*0,5	0,3*0,3
<b>5 m</b>	0,8*0,8	0,3*0,3	0,2*0,2	0,1*0,2
<b>1 m</b>	0,2*0,2	0,1*0,1	0,1*0,1	0,1*0,1

<b>2 RPS (1,8)</b>				
<b>Distancia\ Grados</b>	180 °	90 °	45 °	30 °
<b>30 m</b>	2*2	2*2	2*1,5	1,5*1
<b>20 m</b>	2*2	2*2,5	1*1,5	1*1
<b>10 m</b>	2*3	1*1	0,5*0,5	0,4*0,4
<b>5 m</b>	0,8*0,8	0,4*0,4	0,3*0,3	0,2*0,2
<b>1 m</b>	0,2*0,2	0,1*0,1	0,1*0,1	0,1*0,1

<b>3 RPS</b>				
<b>Distancia\ Grados</b>	180 °	90 °	45 °	30 °
<b>30 m</b>	1,6*1,4	1,6*1,4	1*1,5	0,8*0,8
<b>20 m</b>	2*1,7	1,6*1,6	0,7*0,8	0,4*0,5
<b>10 m</b>	1,7*1,7	0,9*0,9	0,4*0,3	0,2*0,2
<b>5 m</b>	0,8*0,8	0,3*0,4	0,2*0,1	0,1*0,1
<b>1 m</b>	0,2*0,1	0,1*0,1	0,04*0,04	0,03*0,03

<b>4 RPS (3,9)</b>				
<b>Distancia\ Grados</b>	180 °	90 °	45 °	30 °
<b>30 m</b>	1,6*2	1,6*2	1,1*1,5	0,8*1
<b>20 m</b>	2*1,5	1,7*1,7	0,7*0,9	0,5*0,7
<b>10 m</b>	2,5*2	0,9*0,9	0,4*0,4	0,3*0,3
<b>5 m</b>	1*1	0,3*0,5	0,2*0,2	0,15*0,15
<b>1 m</b>	0,2*0,3	0,1*0,1	0,04*0,04	0,04*0,03

<b>5 RPS (4,9)</b>				
<b>Distancia\ Grados</b>	180 °	90 °	45 °	30 °
<b>30 m</b>	1,6*2	1,6*2	1,6*2	1,3*1,3
<b>20 m</b>	2,3*1,8	1,5*1,6	1,1*1,3	0,9*0,1
<b>10 m</b>	2,5*2,5	0,8*0,9	0,6*0,6	0,5*0,4
<b>5 m</b>	1,5*1,5	0,3*0,5	0,27*0,4	0,2*0,31
<b>1 m</b>	0,3*0,4	0,1*0,1	0,06*0,06	0,05*0,04

**Tabla 3-2: Mallado mínimo en cada una de las diferentes aperturas angulares y a distintas velocidades de giro**

Como se puede observar en cada una de las tablas a 30 metros y 180° la apertura es tan grande, que al evaluar solamente la apertura previamente establecida, es equivalente a la apertura de 90°. En el Anexo 2, se pueden ver estas mismas tablas pero sin aplicar la restricción de apertura impuesta de  $\pm 20$  metros en horizontal y  $\pm 5$  metros en vertical.

Como conclusión de estas tablas se puede decir que una mayor velocidad de giro no implica una mayor densidad de puntos dentro de la apertura establecida, si no que en algunos casos se obtiene incluso peor mallado. Un ejemplo claro lo podemos ver comparando la tabla de 4 RPS y 5 RPS. Con 4 RPS se obtiene mejor densidad de puntos que con 5 RPS a todas las distancias y aperturas, por lo que una mayor velocidad en este caso ha empeorado. Igual ocurre si se compara 3 RPS y 4 RPS, con lo que parece ser que a partir de una velocidad, la densidad de puntos empeora en vez de mejorar.

### **3.6.2.3 Estudio la velocidad óptima de giro dependiente de la velocidad del vehículo**

Así pues parece que no hay una única velocidad óptima, si no que dependiendo del caso, será una u otra. Por tanto, se incluye otro factor más, la velocidad del vehículo. Con esto se puede hacer balance entre obtener puntos en todas las regiones con poca densidad u obtener puntos de algunas regiones con mucha densidad y en otras no obtener puntos.

Es orientativo el hecho de que a mayor velocidad del vehículo, es necesaria mayor velocidad de giro del sistema, debido a que debe de haber información de cada región y no puntos ciegos en los que se desconoce si hay o no obstáculo. Si no se ha detectado suficiente información sobre una región, el vehículo debe aminorar la marcha hasta que se obtengan datos de dicha región.

Por esto, este estudio se centra en evaluar una velocidad de giro mínima para una velocidad del vehículo dada. Dicha velocidad de giro debe hacer que se obtengan datos en todas las regiones en frente del vehículo, aunque sea con poca densidad, de manera que la conducción pueda ser segura y fiable.

Para ello se va a estimar en primer lugar que el vehículo circula a 40 km/h, con un tiempo de detención de 5 metros y sin información previa del entorno que le rodea. Esto es una situación imaginaria, porque si el vehículo va a 40km/h es porque ha obtenido datos suficientes del entorno en tiempo real como para garantizar una conducción segura, por lo que ya se conocían datos del entorno. Aún así se supone que no para poder estudiar el caso peor, es decir sin datos previos, de modo que lo que se puede encontrar en la realidad sea sin duda mejor que este estudio.

En resumen, es como si el vehículo fuese a 40 km/h y de repente se enciende el sistema sensorial.

A continuación se va a presentar unas tablas en las que se evalúa la velocidad mínima de giro del sistema para que haya al menos un barrido láser en cada región. Se mostrará a diferentes distancias (en el rango del láser) el tiempo que ha transcurrido en recorrer esta, descontando los 5 metros que el vehículo tarda en detenerse a 40 km/h. Es decir, si la distancia que se quiere evaluar en frente del vehículo es 10 metros, se busca cuánto tiempo tarda el vehículo en recorrer 5 metros (10 metros - 5 metros en detenerse) y dicho tiempo es el que se utiliza para evaluar las revoluciones por segundo mínimas para que haya una pasada láser en cada región.

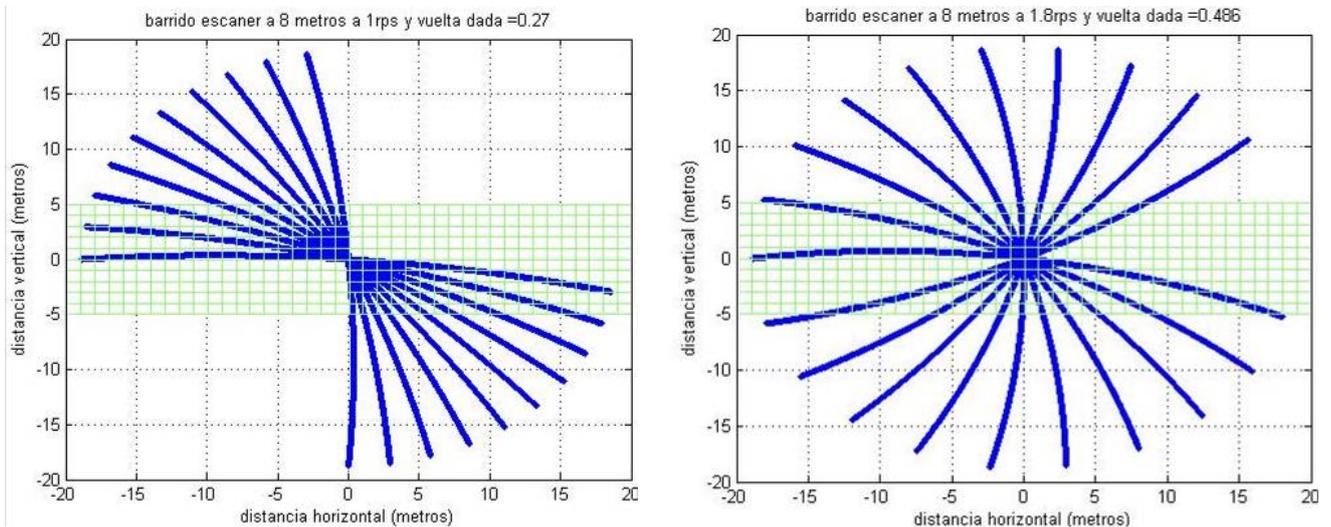
Metros	Metros antes de inicio de detención	Tiempo transcurrido hasta inicio de detención (s)	Velocidad mínima recomendada cubriendo 360°	Velocidad mínima recomendada cubriendo 180°
6	1	0,09	11,11	<b>5,555</b>
7	2	0,18	5,5	<b>2,75</b>
8	3	0,27	3,7	<b>1,85</b>
10	5	0,45	2,22	<b>1,11</b>
15	10	0,9	1,11	-
20	15	1,35	< 1	-
30	25	2,25	< 1	-

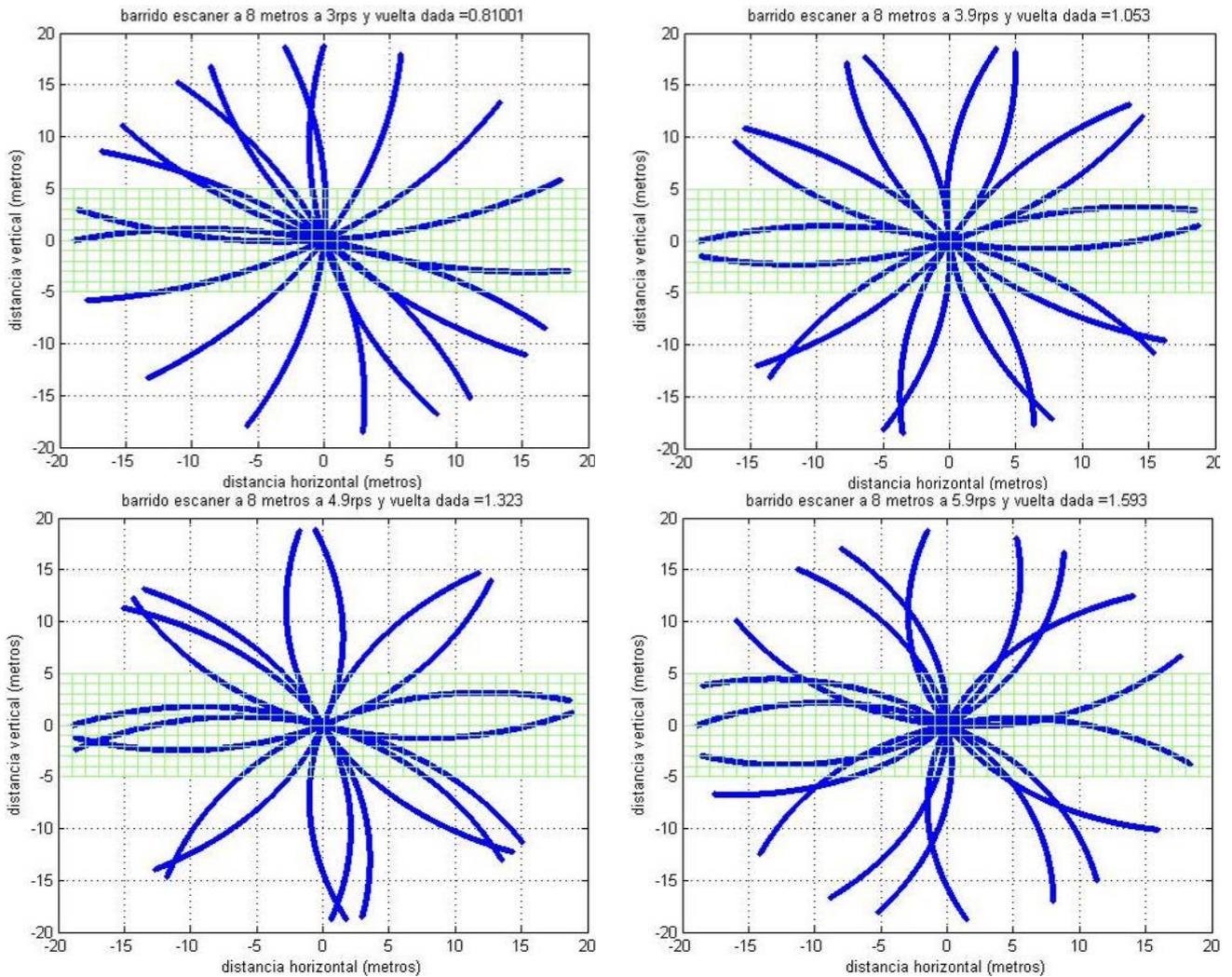
**Tabla 3-3: Velocidad mínima de giro del sistema**

La columna en negrita son las RPS mínimas necesarias en cada caso, recordar que el sensor gira con respecto a su propio eje, y la apertura que posee es de 135° a cada lado, por lo que no es necesario que el sensor haga una vuelta entera para obtener datos de todas las regiones, sino 180°.

Como se puede observar, no se requieren unas velocidades de giro por encima de los 3 RPS para conseguir captar puntos en este caso extremo, lo que ya empieza a dar una idea de que por encima de 3 RPS no se debe de girar en ningún caso.

Ahora se va a representar gráficamente estos efectos, en concreto se ha escogido la fila verde para dicha representación. Mencionar que en las figuras no se ha puesto el mallado mínimo en cada caso (color verde), sino que se ha puesto un mallado de 1 metro x 1 metro.





**Figura 3-16: Representación barrido láser (azul) a 8 metros con diferentes velocidades de giro y mallado (verde) de 1 metro**

Aquí se puede ver de nuevo dos propiedades enfrentadas: obtención de puntos en toda la región y mallado mínimo. A una RPS alta, se consiguen puntos en todas las regiones pero el mallado mínimo empieza a incrementarse por culpa de la no obtención de datos en los bordes más alejados. En este caso es fácil decantarse por una de las dos características: obtención de puntos en todas las regiones, debido a que el mallado en determinadas zonas no importa si es muy grande.

Por ejemplo, el mallado que se puede obtener en la región desde -20 a -15 metros en horizontal y vertical no interesa en absoluto.

Como se puede observar, a partir de las figuras anteriores se verifican los datos obtenidos en la tabla de cálculo de velocidad mínima recomendada (tabla 1321312). A continuación se va a exponer una tabla con el mallado mínimo en cada caso mencionado anteriormente. Dicho mallado se realiza para la apertura impuesta y una nueva. Esta última es de  $\pm 5$  metros en horizontal y vertical, es decir, la zona más crítica debido a que el impacto contra un obstáculo es más probable al estar dicha región definida muy cerca del vehículo.

Mallado (en metros) en apertura $\pm 20$ horizontal $\pm 5$ vertical		
RPS	Horizontal	Vertical
1	>7	> 7
1,8	6	6
3	5	5
3,9	4,5	5
4,9	4,5	4,5
5,9	4,5	4,5

*Tabla 3-4: Mallado disponible a diferentes velocidades dada la apertura descrita*

Mallado (en metros) en apertura $\pm 5$ metros		
RPS	Horizontal	Vertical
1	>3	>3
1,8	1,7	2
3	1,4	1,4
3,9	1,5	1,5
4,9	1,9	1,6
5,9	1,5	1,6

*Tabla 3-5: Mallado disponible a diferentes velocidades dada una apertura  $\pm 5$  metros*

Con estas tablas, se puede ver que el mallado mínimo en frente del vehículo no mejora a partir de una velocidad de giro determinada, muy parecido a lo que se obtuvo en el estudio anterior. No obstante, se va a proseguir estudiando otras dos velocidades del vehículo posibles y después se sacarán conclusiones.

Las tablas siguientes corresponden a una velocidad del vehículo de 30 Km/h con una distancia de frenado del vehículo de 2,8 metros.

Metros	Metros antes de inicio de detención	Tiempo transcurrido hasta inicio de detención (s)	Velocidad mínima recomendada cubriendo 360°	Velocidad mínima recomendada cubriendo 180°
4	1,19	0,1423	7	3,5
5	2,19	0,2623	3,81	1,905
6	3,19	0,3823	2,61	1,305
7	4,19	0,5023	2	1
8	5,19	0,6223	1,6	0,8
10	7,19	0,8623	1,15	0,575

*Tabla 3-6: Velocidad mínima de giro del sistema*

De nuevo se expone el mallado mínimo para el caso de los 8 metros de distancia (fila de color verde en la tabla anterior).

Mallado (en metros) en apertura $\pm 20$ horizontal $\pm 5$ vertical		
RPS	Horizontal	Vertical
1	2	2
1,8	2,3	2,3
3	2,5	2,5
3,9	2,5	3,5
4,9	> 4	> 4
5,9	>> 4	>> 4

**Tabla 3-7: Mallado disponible a diferentes velocidades dada la apertura descrita**

Mallado (en metros) en apertura $\pm 5$ metros		
RPS	Horizontal	Vertical
1	0,7	0,7
1,8	0,9	0,9
3	1	1,5
3,9	1	1,5
4,9	1	1,5
5,9	1	1,5

**Tabla 3-8: Mallado disponible a diferentes velocidades dada una apertura  $\pm 5$  metros**

Por último se estudia este efecto cuando el vehículo circula a 20 Km/h. Esta vez, la distancia que necesita el vehículo para detenerse es de 1,25 metros.

Metros	Metros antes de inicio de detención	Tiempo transcurrido hasta inicio de detención (s)	Velocidad mínima recomendada cubriendo 360°	Velocidad mínima recomendada cubriendo 180°
2	0,75	0,1349	7,41	3,71
3	1,75	0,3149	3,17	1,59
4	2,75	0,4949	2,02	1,01
5	3,75	0,6749	1,4817	0,74
6	4,75	0,8549	1,16	0,58
8	6,75	1,2149	0,82	0,41

**Tabla 3-9: Velocidad mínima de giro del sistema**

Mallado (en metros) en apertura $\pm 20$ horizontal $\pm 5$ vertical		
RPS	Horizontal	Vertical
1	1,3	1,3
1,8	1,7	1,7
3	1,6	1,6
3,9	1,5	1,5
4,9	1,5	1,5
5,9	1,6	1,6

**Tabla 3-10: Mallado disponible a diferentes velocidades dada la apertura descrita**

### 3.6.2.4 Conclusiones

A la vista de todos los resultados expuestos anteriormente, es imperativo sacar unas conclusiones en las que se pueda estimar la velocidad de giro óptima del sistema sensorial a diferentes velocidades del vehículo. Dichas conclusiones son basadas en que lo más importante es que el sistema sea capaz de captar datos en cada región, y que el mallado sea el mínimo posible en las regiones que se encuentren enfrente del vehículo. Con esto, se procede a sacar las siguientes conclusiones:

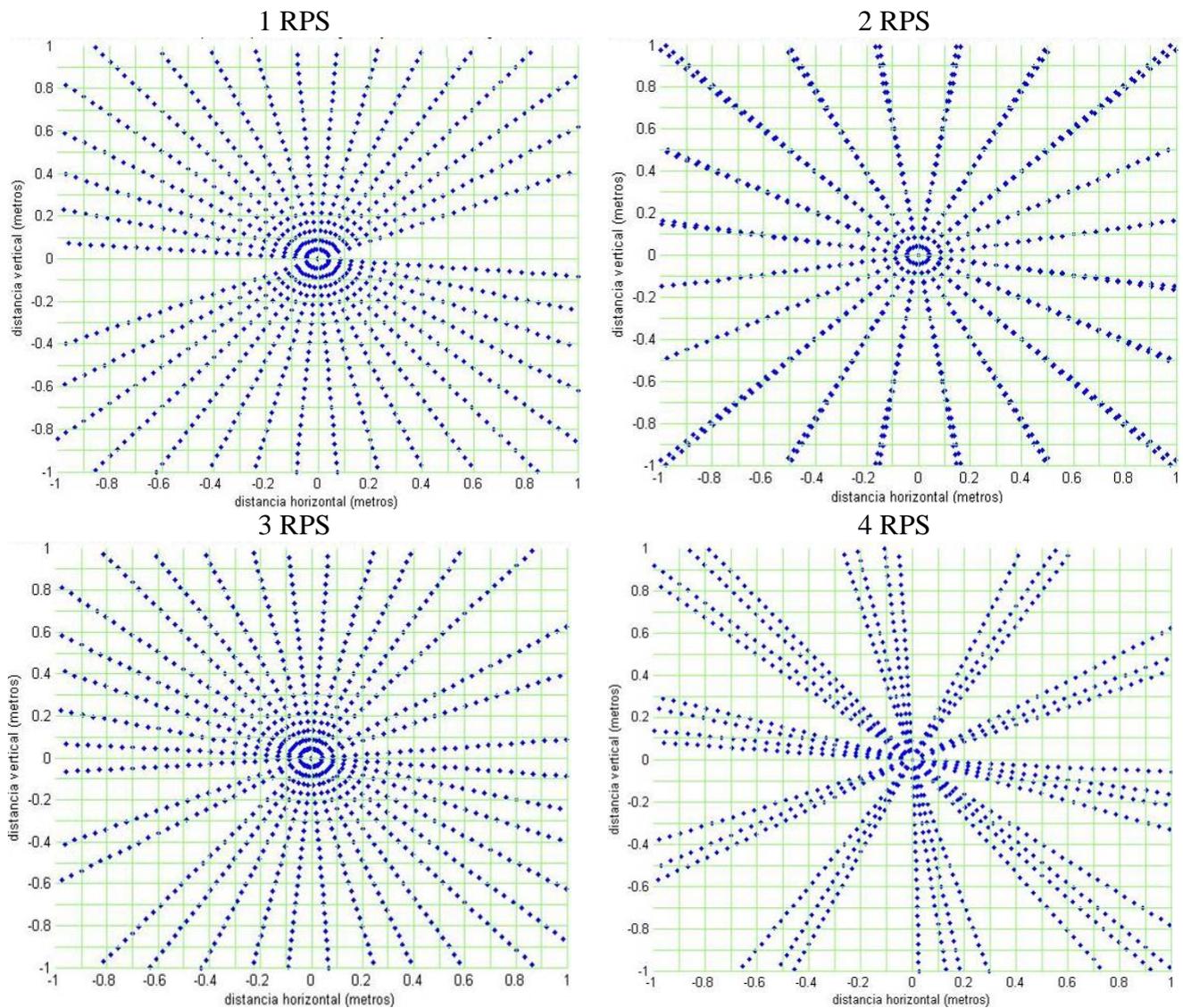
- Para la toma de medidas en frente del vehículo y éste en estático, un RPS bajo (es decir, menor o igual que 1) es la mejor solución debido a que ofrece gran densidad de puntos con poca separación entre haces. Puesto que el vehículo está estático, la captación de medidas en todo el entorno no tiene por qué ser rápida. Por tanto, es la mejor opción gracias a la gran definición que ofrece.
- Si el sistema sensorial gira a un RPS elevado, se puede tener información en todas las regiones, pero no se garantiza que dicha información ofrezca una densidad de puntos aceptable. En aperturas alejadas del centro de rotación, la densidad de puntos será francamente mala.
- En frente del vehículo, con una velocidad de giro mínima tal que el escáner pueda completar 180°, se obtiene la mejor densidad de puntos. Este efecto se puede ver en las tablas de velocidad de giro dependiendo de la velocidad del vehículo. La velocidad de giro mínima tal que garantiza que se obtienen datos en todas las regiones es la que presenta a su vez mejor densidad.
- Alcanzada la velocidad mínima de giro necesaria para que una determinada velocidad del vehículo, no se debe incrementar la velocidad de giro si la velocidad del vehículo no se incrementa también. La densidad de puntos no mejora y por tanto no se gana absolutamente nada girando más rápido de lo necesario.
- La velocidad de giro óptima dada una velocidad del vehículo y un alcance al cual se quiera optimizar, coincide con la mínima necesaria para captar puntos en todas las regiones. Así pues, para el caso de 40 Km/h y 8 metros en frente del vehículo, la velocidad óptima de giro es 3 revoluciones por segundo. Si esos 8 metros son incrementados, la velocidad óptima descenderá según las tablas expuestas en este apartado.
- No olvidar que este estudio se ha realizado asumiendo un entorno en el cual no hay escaneado previo y el vehículo se mueve a velocidades de hasta 40 Km/h. Lo que hay que destacar es que antes de iniciar cualquier movimiento en el vehículo se debe captar totalmente y con mucha densidad el entorno y, una vez analizado, se procederá al movimiento y a la toma de decisiones en tiempo real. Por ello, realmente no se necesita una velocidad tan elevada como 4 revoluciones por segundo. Si se tuviera que especificar una contorno de velocidad de giro mínima y máxima sería desde 0.027 RPS (velocidad a la que se conseguía 0.25° de resolución angular en vertical) hasta 3 RPS.
- A partir de 1 RPS, el haz láser se empieza a distorsionar de manera notable. Por debajo de esta velocidad, dicho efecto no es detectable.

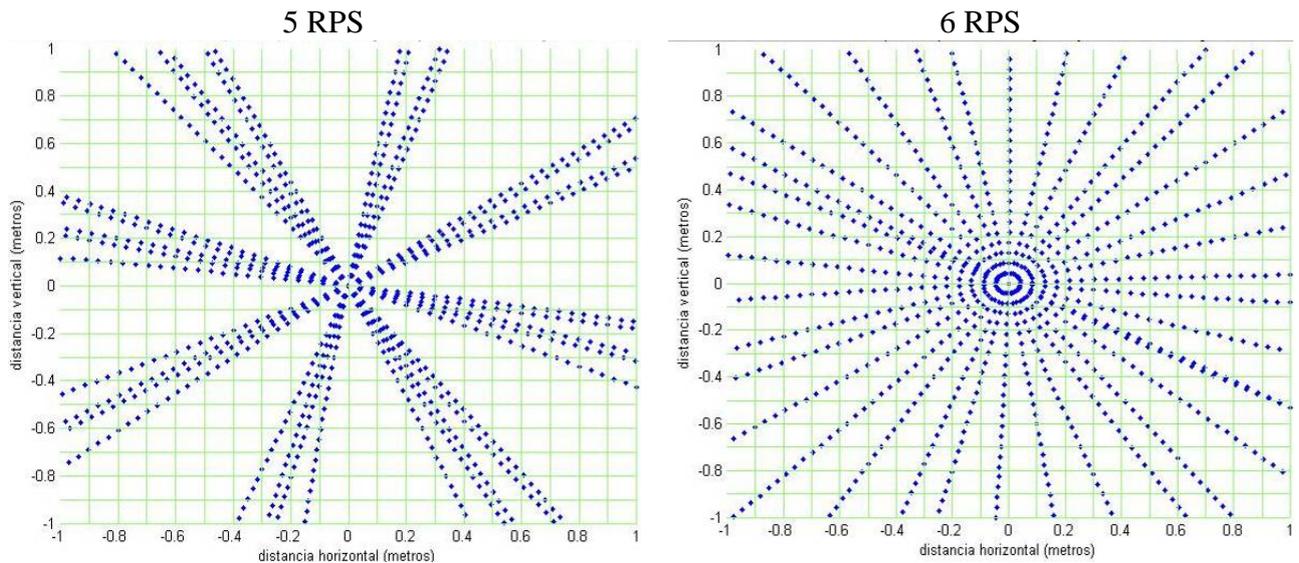
### 3.6.2.5 Estudio resolución de medidas en el centro del escáner

Ya se ha visto el mallado mínimo en contornos definidos, pero en este estudio se va a enfocar en qué ocurre en una apertura mucho más pequeña. Dicha nueva apertura queda definida como  $\pm 1$  metro en horizontal y vertical. Dicho contorno debe ser el que más densidad de puntos ofrezca, de modo que con este estudio se pretende ver cuánto de bueno es dicha región. En este caso, no se va a buscar el mallado mínimo sino que se va a imponer un mallado de  $0.1 \times 0.1$  metros para comprobar si en todo el contorno hay al menos un punto por cada malla. Las condiciones elegidas para este estudio son:

- Distancia en frente del vehículo: 10 metros.
- Segundos de simulación: 0.45 segundos.
- Mallado:  $0.1 \times 0.1$  metros.

A continuación se exponen las figuras correspondientes a cada velocidad de giro estudiada.





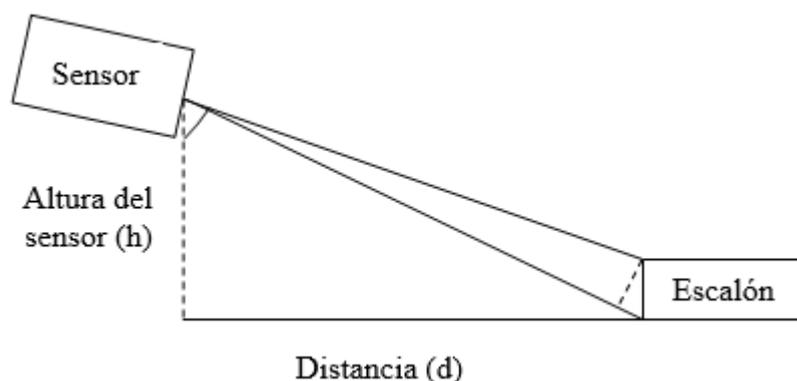
**Figura 3-17: Representación barrido láser (azul) a 10 metros con diferentes velocidades de giro y mallado (verde) de 10 x 10 centímetros**

A 2 y 3 RPS se puede ver como esta región es bastante homogénea y el mallado es cubierto de manera aceptable. Es decir, a 10 metros de distancia, el sistema sería capaz de detectar un obstáculo de 10 x 10 centímetros si éste se encuentra justo enfrente del láser. No se puede olvidar que todos estos datos han sido tomados en tal sólo 0.45 segundos, lo que hace que este sistema resulte muy prometedor.

### 3.6.3 Aceptación de la configuración del sistema sensorial

Después de los estudios realizados y de sacar las conclusiones pertinentes, está claro que este sistema presenta unas características verdaderamente prometedoras para la navegación autónoma. Se ha conseguido a partir de un laser 2D generar puntos 3D y con unas densidades muy apropiadas para el objetivo de este proyecto. Todo ello sin aumentar el coste inicial planteado.

Con el último estudio se ha observado que un obstáculo de 10 centímetros puede ser detectado si se encuentra justo en frente del sensor. Por ello, se cree que si al sensor láser se le impusiera una cierta inclinación hacia abajo, podría incluso detectar pequeños escalones. Con esto en mente, para que a 10 metros la zona de máxima densidad de puntos se encuentre apuntando a 10 centímetros sobre el suelo se tiene que saber a qué altura debe colocarse el laser. Supuesta ésta última a 1 metro, se procede al cálculo del ángulo de inclinación.

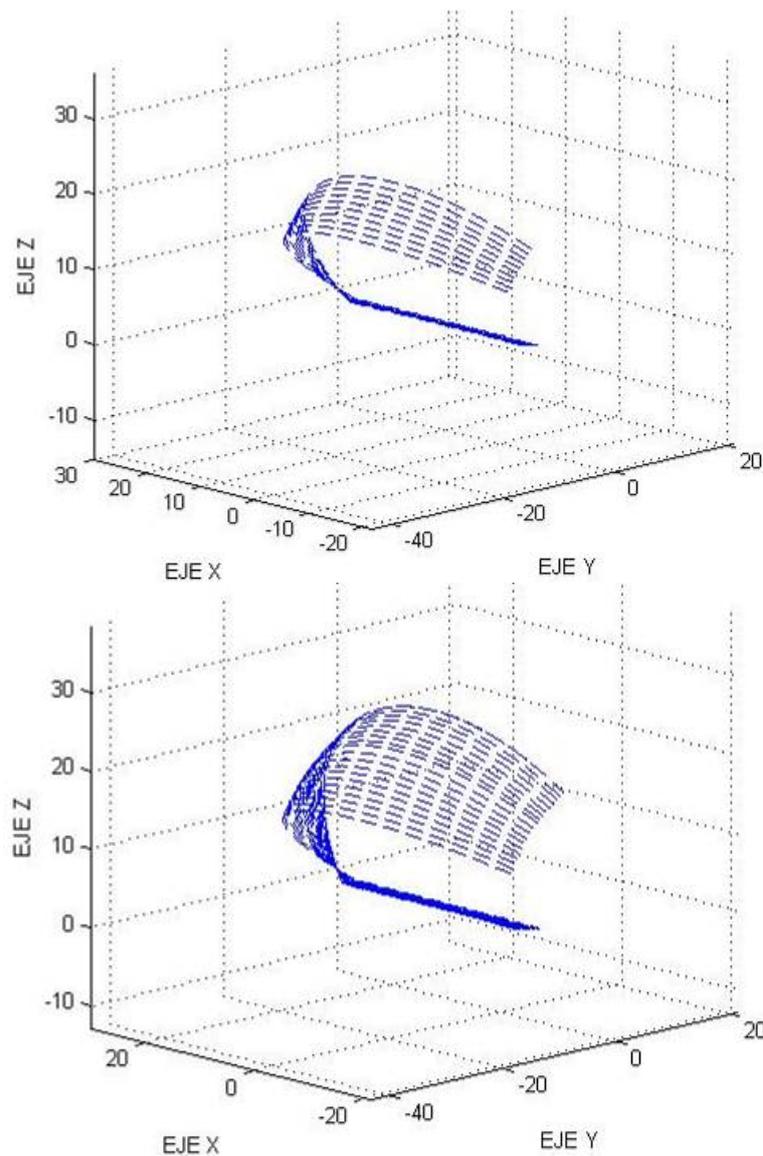


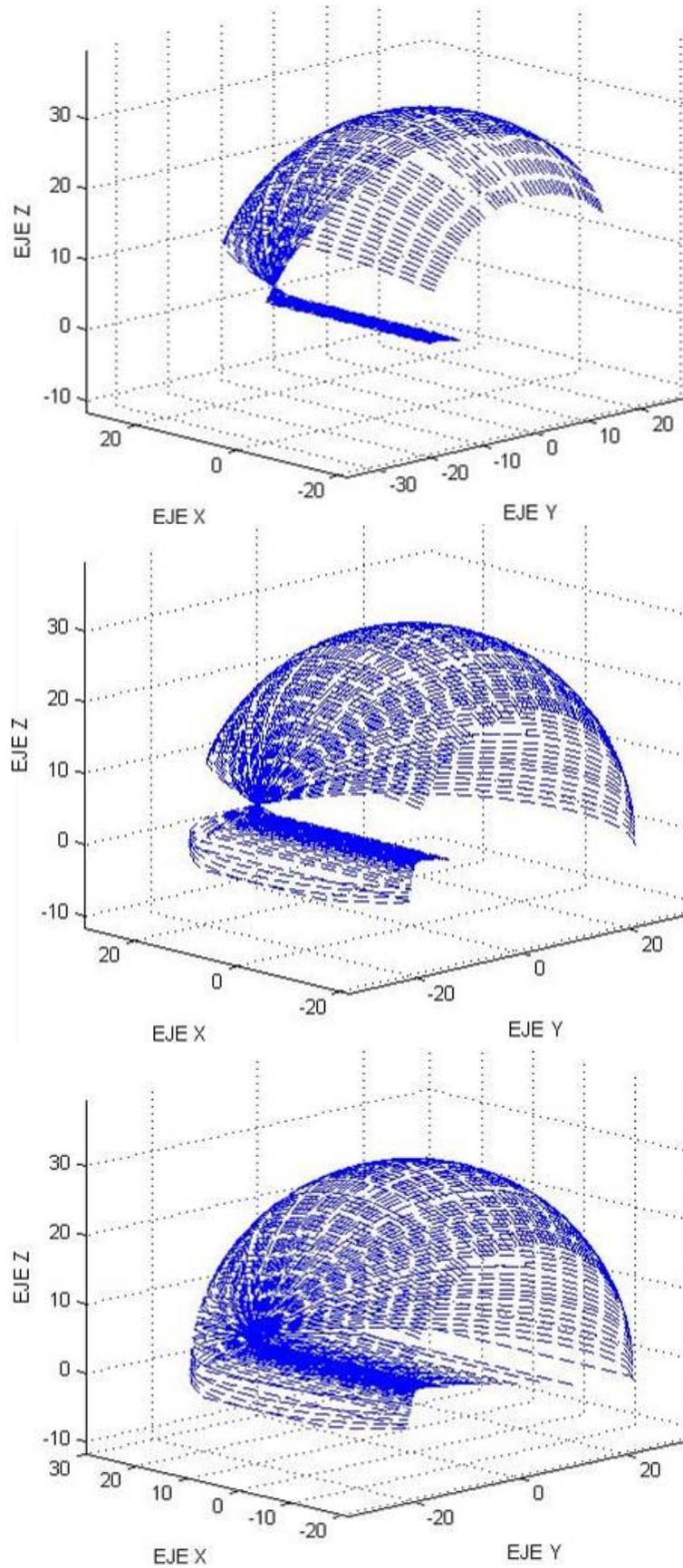
**Figura 3-17: Representación ángulo de inclinación del sensor**

Aplicando el teorema de Pitágoras y de las definiciones de seno y coseno se ha obtenido que el ángulo de inclinación hacia abajo con respecto a la horizontal es de unos 5 grados. Por tanto, el sistema sensorial elegido es el sensor 2D HOKUYO 30LX-EW con una inclinación hacia debajo de 5 grados y una plataforma de giro que debe cumplir con las especificaciones expuestas anteriormente de velocidad máxima permisible.

### 3.7 Simulación en 3D

A continuación se va a exponer las simulaciones realizadas anteriormente pero en tres dimensiones, cuando el sistema sensorial ha girado 180°:





**Figura 3-18:** Representación de un barrido láser generado cuando éste ha recorrido  $180^\circ$





# 4 Composición y elaboración del sistema sensorial

---

## 4.1 Introducción

Teniendo claro el sistema sensorial elegido en el capítulo anterior, hay que definir una serie de especificaciones para fabricar dicho sistema de la mejor manera posible. Por un lado está el sensor láser que no requiere de ninguna elaboración puesto que es comprado. Por otro, el sistema de giro sí debe de tener una serie de especificaciones para asegurar su correcto funcionamiento. Por último, la inclinación de unos 5 grados hacia abajo de todo el sistema se realizará cuando dicha implementación se adjunte al vehículo Yamaha Rhino.

En los próximos apartados se va a desglosar un conjunto de componentes del sistema de giro necesarios para que se generen de manera correcta puntos 3D del entorno. Cada uno de estos componentes será explicado de modo que se pueda ver la utilidad de cada pieza incorporada.

## 4.2 Componentes sistema de giro

El sistema de giro está formado por un motor, un encoder, una reductora planetaria, una unidad de medición inercial, un controlador EPOS (controlador de posición miniaturizado) y un anillo rotatorio. A continuación se desglosa cada uno de estos componentes:

### 4.2.1 Motor

El motor escogido es de un motor plano de tipo EC. Un motor EC es un motor síncrono sin escobillas que combina tensión alterna y continua, de modo que dicho motor posee lo mejor de ambos tipos: poco consumo, alta eficiencia y flexibilidad de utilización. Este tipo de motores son más compactos, lo que conlleva a tener ventajas aerodinámicas y de funcionamiento. Por último, la velocidad del motor depende del voltaje suministrado a la entrada de éste [38].

### 4.2.2 Encoder

El encoder es un transductor rotativo, que permite conocer el ángulo girado por el motor. Hay dos tipos de encoder según el tipo de información que generan:

- Encoder incremental: La señal de salida es transmitida por un hilo por el cual se transmite un pulso por cada ángulo girado. Por ejemplo, si un determinado encoder tiene 2000 pulsos por rotación, se tendrá un pulso cada  $360^\circ/2000 = 0.18^\circ$ . El inconveniente de este tipo de encoder es que no se dispone de una referencia absoluta de la precisión en la que se encuentra el eje. Por ello, no se guarda la posición inicial con el que el motor comenzó a girar.

- Encoder absoluto: La posición se da mediante un bus paralelo, de manera que si se tuviera un encoder de 256 ( $2^8$ ) posiciones, el bus sería de 8 líneas que indicará en binario la posición en que se encuentra el motor. Dicha posición es codificada en código Gray [39] para lograr mayor robustez a errores.

El inconveniente de estos encoder es que se requiere un bus de datos para obtener la información y que la resolución no es elevada.

En este caso, el encoder elegido es incremental óptico que sea compatible con el motor elegido EC flat, ya que se necesita la máxima precisión posible en la medición de ángulos del motor.

### 4.2.3 Reductora

Dispositivo conectado al eje de giro del motor con el objetivo de aumentar el par de fuerza que el motor es capaz de girar. Un motor posee unas velocidades de giro muy elevadas pero el peso que pueden manejar no lo es tanto, por ello se crearon estos componentes, capaces de aumentar la fuerza del motor a costa de disminuir la velocidad máxima de este. De esta forma, una reductora puede hacer que se limite la velocidad máxima de giro del sistema y además que sea capaz de manejar más peso. De esta manera y orientándolo al propósito del sistema de giro que se quiere elaborar, se necesita que el motor sea capaz de girar el conjunto formado por el sensor HOKUYO 30LX-EW y una pequeña plataforma que una éste con el motor.

Hay varios tipos de reductoras entre las que se ha elegido la reductora por engranajes, que son aquellos en los que el eje del motor hace girar un conjunto de engranajes. Sus ventajas son el mayor rendimiento energético, menor mantenimiento y menor tamaño.

Por último, el término planetario significa que el eje de giro del motor, al pasar por la reductora, no es desviado a otra posición, si no que sigue estando en el mismo eje.

A continuación se puede ver un ejemplo de reductor planetario con engranajes:



*Figura 4-1: Imagen de una reductora*

#### **4.2.4 Unidad de medición inercial (IMU)**

Es un dispositivo electrónico capaz de medir la orientación y las fuerzas gravitacionales que se le aplican, usando una combinación de acelerómetros y giroscopios. Por tanto, es un aparato que te permite saber si el sistema está inclinado respecto a un eje, muy útil para la navegación autónoma.

Dicho sistema es necesario debido a que ofrece información de en qué orientación e inclinación se encuentra el vehículo en cualquier instante de tiempo, de modo que se puede tener un control de las condiciones en las que se encuentra.

El término IMU se utiliza para referirse a una caja que contiene tres giroscopios, tres acelerómetros y de manera opcional, tres magnetómetros. En cada caso, se colocan de manera que sus ejes de medición sean perpendiculares entre sí. Por desgracia, la IMU sufren un error acumulativo debido a que cualquier error en una posición, por pequeño que este sea, es tenido en cuenta en el cálculo de la siguiente posición.

#### **4.2.5 Controlador EPOS**

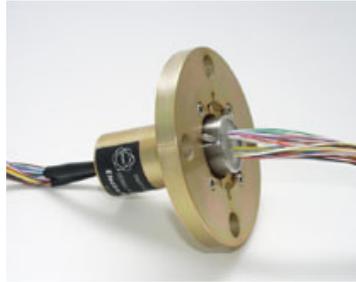
Es el encargado de controlar y modificar los parámetros de funcionamiento del motor y de traducir la información suministrada por el encoder. Por tanto, este dispositivo incorpora el software necesario para poder modificar de manera sencilla la velocidad del motor o el sentido del giro, siempre acompañado del cálculo del ángulo en que se encuentra gracias a los datos proporcionados por el encoder. Además ofrece un doble lazo de control, de manera que si surge algún tipo de desviación a la hora de girar a una determinada velocidad, el controlador lo sabrá y actuará en consecuencia para reducir dicho error. Por si fuera poco, permite eliminar las vibraciones producidas por una velocidad dada, de manera que se puedan obtener giros continuos y con fiabilidad.

El controlador EPOS entregará la posición del sensor con una frecuencia elevada de manera que se pueda tener gran exactitud y fiabilidad.

#### **4.2.6 Anillo rotatorio (slip ring)**

Como se pretende que el giro del sensor láser sea continuo y éste posee cables de alimentación, sincronización y datos surge el problema de que dichos cables acabarán enrollados, no permitiendo el giro continuo y pudiendo romper todo el sistema. Por ello es necesaria la incorporación del anillo rotativo.

El anillo rotativo es un dispositivo electromecánico que permite la transmisión de potencia y de señales eléctricas desde una posición estática a una estructura rotacional continua, sin que ello suponga un peligro para el cableado ni para sus señales asociadas. Por dentro poseen un conjunto de vías circulares conductoras que hacen que aunque se gire uno de los dos extremos de conexión, la señal se mueva por toda la vía circular y por ello siga conectado al cable estático.



*Figura 4-2: Imagen de un anillo rotativo*

Las vías circulares que posee el anillo rotativo en su interior pueden ser vistas en la siguiente figura:

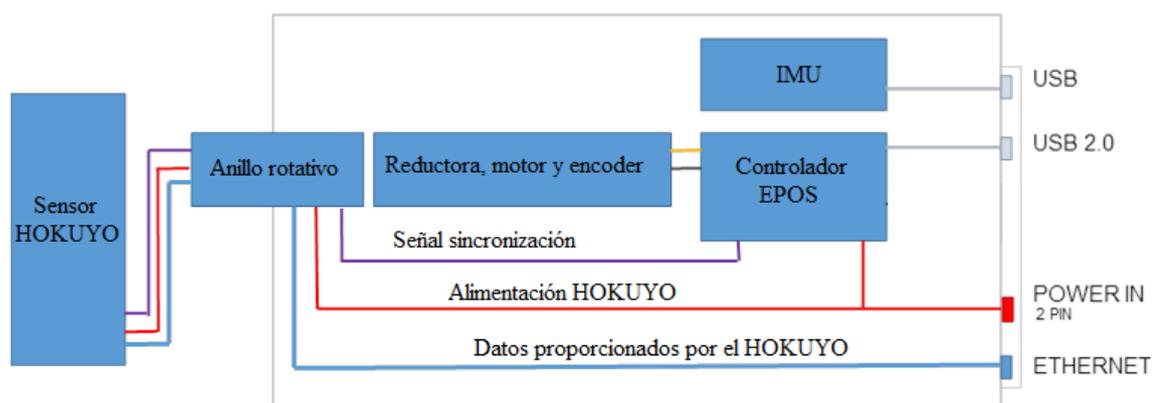


*Figura 4-3: Imagen de un anillo rotativo*

### 4.3 Requisitos e implementación

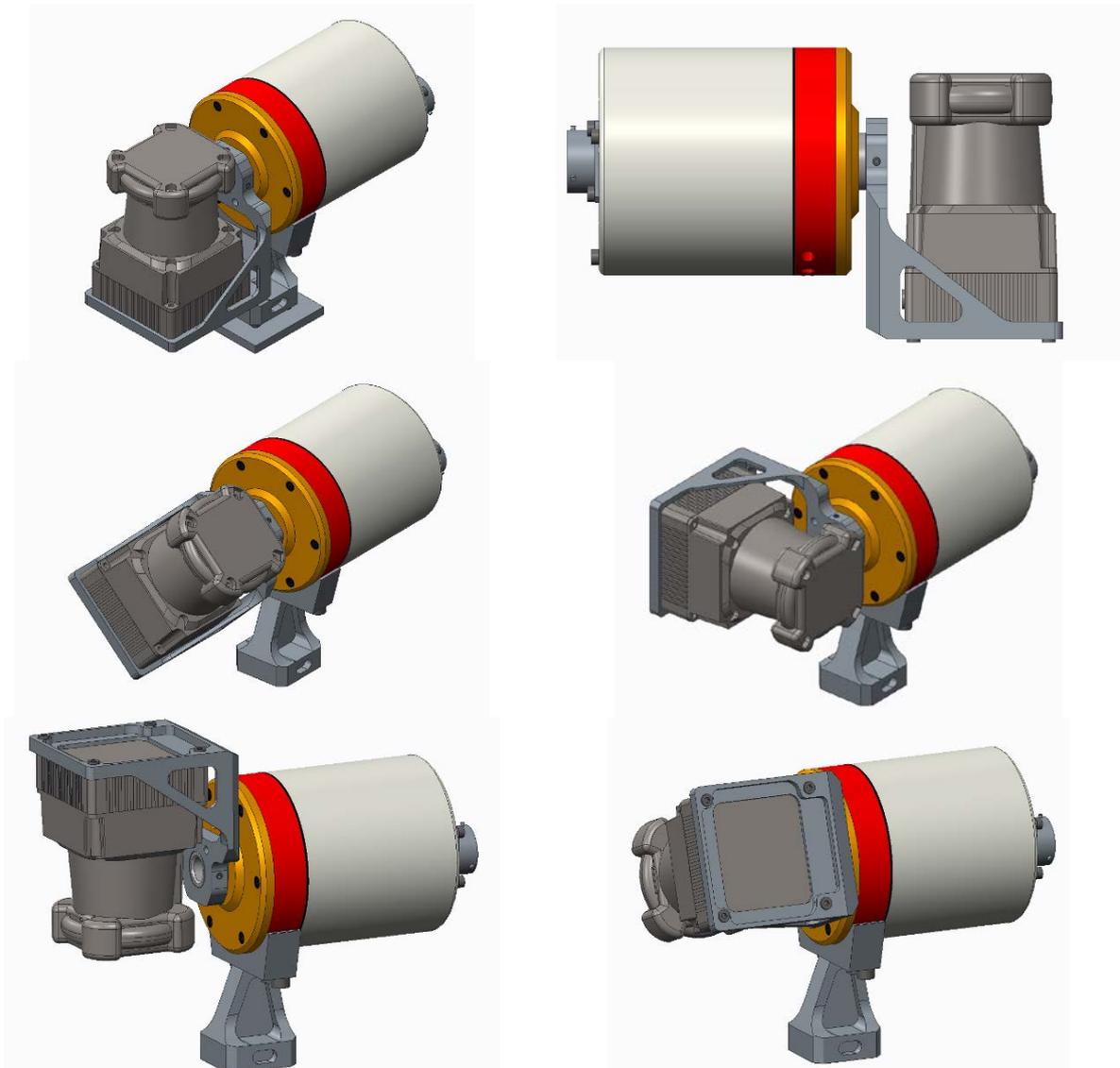
El sistema de rotación tiene que ser capaz de girar el peso del sensor HOKUYO y una pequeña plataforma de sujeción en el rango de velocidad impuesta anteriormente. Además, dicho sistema de giro no debe hacer que la apertura angular del sensor se reduzca. Es decir, como mucho el sistema de giro debe estar comprendido en la región en la que el sensor no puede obtener datos. Además, el sistema de giro debe de tener protección frente a efectos meteorológicos adversos o frente a la entrada de partículas de polvo que puedan dañar el sistema.

Con estas características el sistema de giro es incorporado al sensor HOKUYO de la siguiente manera:



*Figura 4-4: Conexión y colocación de los componentes*

La caja en donde se encuentran todos los componentes excepto el sensor láser y parte del anillo rotativo es un encapsulado de protección diseñado específicamente para la debida protección de los componentes. Así pues, a continuación se va a exponer el sistema simulado basado en estos requerimientos y dispositivos, en los que se puede ver el sistema sensorial completo:



***Figura 4-5: Diseño del sistema sensorial simulado***

En el Anexo D se pueden consultar las características finales del sistema sensorial realizado.



***Figura 4-6: Fotografía sistema sensorial real***



# 5 Desarrollo software del sistema sensorial y resultados de la simulación

---

## 5.1 Introducción

Para la generación de medidas 3D del entorno, es necesario procesar la información obtenida desde el sensor láser y el encoder del motor. Posteriormente, es necesario que ambos tipos de información se fusionen en uno solo, haciendo que se complementen de manera constructiva. Para ello es necesario un software específico que haga que dicha fusión de información sea coherente y pueda generar lo mejor posible el entorno del vehículo.

Antes de probar con el sistema sensorial real, expuesto anteriormente, se ha simulado dicho sistema en una plataforma llamada GAZEBO, en el que se es capaz de simular cada detalle. El procesamiento de la información se llevará a cabo en ROS (Robot Operating System) y los resultados se pueden ver gráficamente mediante el programa RVIZ. De todos ellos se realizará a continuación una breve introducción.

Todas estas plataformas y programas han sido instaladas bajo el sistema operativo Ubuntu 14.04.

### 5.1.1 ROS

Es una infraestructura específica para el desarrollo software. Brinda servicios similares al de un sistema operativo pero orientados a sistemas robóticos extensibles y multitarea. Dispone de potentes mecanismos de abstracción de hardware, comunicación de interprocesos, etc... Proporciona una inmensa cantidad de herramientas y librerías que ayudan a los programadores a realizar sus objetivos en la robótica. Su estructura está compuesta de una red de procesos (nodos) que se ejecutan independientemente pero que pueden estar conectados entre sí por medio de mensajes o servicios, de modo que ROS proporciona una sencilla manera de crear interconexiones entre programas, haciendo que todo esté montado sobre una red de nodos. Gracias a las facilidades de sus librerías, ROS hace que trabajar con nodos independientes resulte sencillo, de manera que apenas hay que programar sino que solamente se deben hacer llamadas a estructuras ya definidas.

Además, ROS es compatible con programación en C++, de manera que una vez creados los nodos y los mensajes de interconexión, la información puede ser tratada fácilmente.

Los conceptos básicos de procesamiento de datos en ROS son los siguientes:

- **Master:** Es el nodo principal de ROS, encargado del control del resto de nodos y del cumplimiento de las interconexiones. Siempre debe de estar activo. Guarda una serie de parámetros de control de manera que si ha sucedido algo erróneo, permite rastrearlo.
- **Nodos:** son las unidades que llevan a cabo el procesamiento de los datos. Como ya se ha mencionado anteriormente, dichos nodos pueden contener código C++, compatible con ROS, lo que facilita la tarea del desarrollo software.

- Mensajes o servicios: Estructuras de datos destinadas al intercambio de información entre nodos. Pueden contener una gran variedad de información y está disponible para cualquier nodo que los solicite. Cada uno de los mensajes y servicios disponibles posee una estructura de datos definidos por ROS. No obstante, el usuario puede crear un nuevo mensaje o servicio a su gusto que se amolde a sus necesidades. Cualquier mensaje es enviado con un nombre determinado, por lo que si el receptor del mensaje conoce dicho nombre de emisión, se suscribirá a dicho mensaje. La diferencia entre servicio y mensaje es que con los mensajes un nodo envía información y uno o varios la reciben y con los servicios, un nodo hace una llamada para que se le atienda la petición por otro nodo, y éste, una vez procesados los cálculos, contesta al nodo que hizo la petición.
- Tema: Los mensajes se enrutan a través de un sistema de transporte de ubicación semántica. Un nodo envía un mensaje con un determinado tema. Un nodo que está interesado en el tema lo escuchará y los demás no. Simplemente un nodo debe comunicar al Master que desea publicar en él para enviar información o suscribirse en él para recibir su información. Pueden existir varios nodos publicando y varios nodos suscritos a la vez en un mismo tópico.

Además de todo esto, ROS incorpora algunas herramientas que hacen que permiten obtener gran cantidad de información del sistema de manera muy sencilla. En especial cabe destacar `ros_console`, `rqt_graph` y `RVIZ`. La primera herramienta mencionada hace posible un seguimiento de cada uno de los nodos en ejecución, permitiendo ver si hay algún tipo de error. En cuanto a la segunda, es una herramienta que permite ver de manera visual qué nodos se están ejecutando, qué mensajes o servicios publican y a qué mensajes están suscritos. `RVIZ` es una herramienta de visualización que permite observar de manera gráfica el escaneo de un laser y generar un mapa a partir de este. En realidad es un nodo ROS ya programado en el que solamente hay que incorporar el nombre del mensaje que se quiera representar en él. Incluso es capaz de hacer un estudio de navegabilidad en el que se puede ver por dónde se ha movido el robot y qué datos ha tomado en cada zona.

En [40] se puede ampliar información acerca de ROS y sus diferentes aplicaciones.

Hay diferentes versiones de ROS, entre las cuales se ha optado por ROS Indigo, compatible con el sistema operativo instalado en el ordenador del robot, Ubuntu 14.04. Para el correcto funcionamiento del enlace entre ROS y Gazebo, se recomienda la creación de un entorno de trabajo ROS basado en “catkin”. A continuación se expone el procedimiento para la creación y caracterización de este entorno de trabajo ROS desde la consola:

Se crea una carpeta llamada “src” dentro del directorio “catkin_ws”
<pre>mkdir -p ~/catkin_ws/src</pre>
Dentro de ese directorio, para inicializar el entorno se ejecuta la siguiente línea:
<pre>catkin_init_workspace</pre>
Una vez creado, en el directorio “catkin_ws” se ejecuta la siguiente línea:
<pre>catkin_make</pre>

Éste comando es el que compila y construye el software desarrollado en su interior. Para que ROS “sepa” que este directorio es un entorno de trabajo, siempre hay que referenciarlo mediante:

```
source devel/setup.bash
```

Una vez que ROS sabe que en `catkin_ws/src` hay un entorno de trabajo, sólo queda crearlo mediante la siguiente línea:

```
catkin_create_pkg <NOMBRE_ENTORNO> std_msgs rospy roscpp
```

En `<NOMBRE_ENTORNO>` se puede poner el nombre que se quiera, en el caso de este proyecto se llamará “`rls30`”. Las otras tres etiquetas son librerías que ROS incluirá dentro del entorno de trabajo. Según se vaya incorporando nuevas funciones software, dichas librerías se irán incrementando.

Catkin\_make/

src/

CMakeLists.txt --Fichero creado automáticamente. No se modifica

Rls30/

CMakeLists.txt -- Fichero que contiene el nombre los ficheros C++ que deben ejecutarse, además del enlace a GAZEBO u otras externos necesarios.

package.xml -- Contiene todas las librerías de las que se hace mención en el software. Al crear el entorno con `std_msgs rospy` y `roscpp`, en el interior de este fichero se encontrarán referencias a dichas librerías

src/ -- Directorio en el que se encuentran cada uno de los programas C++ desarrollados en este proyecto. Cada uno de ellos será compilado si se han incluido en `CMakeList.txt`

launch/ -- Directorio en el que se encuentran los lanzadores de nodos

Si el número de programas en `src/` es grande, es conveniente hacer un lanzador de nodos. Estos ficheros se utilizan para que sea cómodo ejecutar un gran número de nodos a la vez, sin necesidad de que el usuario tenga que ejecutarlos uno a uno.

### **5.1.2 Gazebo**

Gazebo [41] es un simulador 3D utilizado ampliamente en la robótica, ya que permite testear el funcionamiento y las características de un robot antes de que éste sea montado físicamente. Además, permite evaluar el comportamiento de un robot bajo diferentes condiciones. En este proyecto Gazebo es utilizado para simular el sistema sensorial del vehículo de manera sea sencillo configurar diferentes escenas y obstáculos. En este entorno, es posible configurar el sensor laser elegido en este proyecto, de manera que el simulador será capaz de ofrecer datos totalmente acordes con lo que se va a encontrar cuando se monte el sistema real.

Por medio de unas librerías adicionales [42], Gazebo es capaz de exportar información entendible por ROS, de manera que se tiene todo lo necesario para que se pueda probar el software desarrollado antes de que el sistema real este construido.

### **5.1.3 Octovis**

Es un programa de visualización 3D, ligeramente distinto de RVIZ en que Octovis recibe los datos y los procesa con un modelo de “cubos” de dimensión configurable mientras que RVIZ únicamente representa el dato que obtiene, sin procesamiento adicional.

## **5.2 Desarrollo software del sistema sensorial**

### **5.2.1 Introducción**

Una vez explicadas anteriormente cada una de las partes en que se divide la programación de este proyecto, se procede a elaborar el software necesario para obtener datos 3D alrededor del vehículo. El objetivo final es que dicho software sea capaz de realizar un mapa 3D en tiempo real del entorno y un mapa 2D a partir de éste, de manera que se pueda ver en cada instante de tiempo dónde están los obstáculos, qué forma tienen etc... Como se va a configurar el sistema sensorial real en Gazebo, el software desarrollado puede ser testado de manera que los resultados se ajustan perfectamente a la realidad, además de que el sistema final funcionará tanto con el sistema simulado como con el sistema real.

Las pautas que se van a seguir son las siguientes:

- Desarrollo en Gazebo del sistema sensorial real (salvo IMU).
- Desarrollo software en ROS de procesamiento de información suministrada por el punto anterior.
- Estudio de resultados con el visualizador Rviz y comparación con el entorno de trabajo inicial de Gazebo como medida de corrección de errores.

### **5.2.2 Desarrollo del sistema sensorial real en Gazebo**

En primer lugar, se debe crear un fichero .sdf para configurar dicho sensor láser y posteriormente dotarlo del movimiento circular del sistema de giro y acoplarlo al vehículo Yamaha Rhino.

Para este propósito se crea el un directorio nuevo llamado HOKUYO30LX en el directorio .gazebo/models, el cual debe contener dos fichero: model.sdf y model.config.

En el interior de model.sdf se ha programado cada una de las características de este sensor láser tal como rango de distancias, error, apertura angular, tiempo entre escáneres etc... El código utilizado se puede ver a continuación:

```

<?xml version="1.0" ?>
<sdf version="1.4">
  <model name="HOKUYO30LX">

    <pose>1.3 0 1.2 0 0 0</pose>
    <link name="link">
      <gravity>>false</gravity>
      <must_be_base_link>>true</must_be_base_link>
      <inertial>
        <mass>0.37</mass>
      </inertial>
      <visual name="visual">
        <geometry>
          <mesh>
            <uri>model://hokuyo/meshes/hokuyo.dae</uri>
          </mesh>
        </geometry>
      </visual>

      <collision name="collision-base">
        <pose>0 0 -0.0145 0 0 0</pose>
        <geometry>
          <box>
            <size>0.06 0.06 0.087</size>
          </box>
        </geometry>
      </collision>

      <collision name="collision-top">
        <pose>0 0 0.0205 0 0 0</pose>
        <geometry>
          <cylinder>
            <radius>0.021</radius>
            <length>0.029</length>
          </cylinder>
        </geometry>
      </collision>

      <sensor name="laser" type="ray">
        <pose>0.01 0 0.0175 0 -0 0</pose>
        <ray>
          <scan>
            <horizontal>
              <samples>1080</samples>
              <resolution>1</resolution>
              <min_angle>-2.356194</min_angle>
              <max_angle>2.356194</max_angle>
            </horizontal>
          </scan>
          <range>
            <min>0.1</min>
            <max>30</max>
            <resolution>0.05</resolution>
          </range>
        </ray>
      </sensor>
    </link>
  </model>
</sdf>

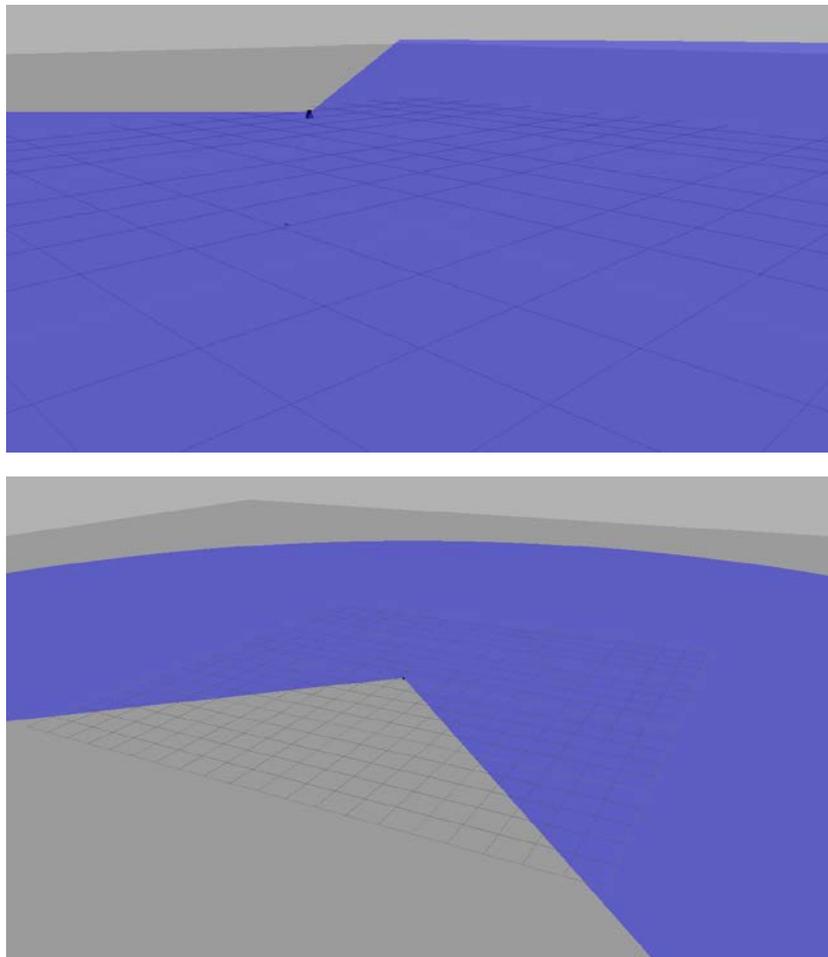
```

```
<type>gaussian</type>
<mean>0.0</mean>
<stddev>0.01</stddev>
</noise>
</ray>

<plugin name="laser" filename="libRayPlugin.so" />
<always_on>1</always_on>
<update_rate>40</update_rate>
<visualize>true</visualize>
</sensor>
</link>
</model>
</sdf>
```

El segundo fichero sirve únicamente para dotar al sistema creado de un nombre, en este caso también se llama HOKUYO30LX.

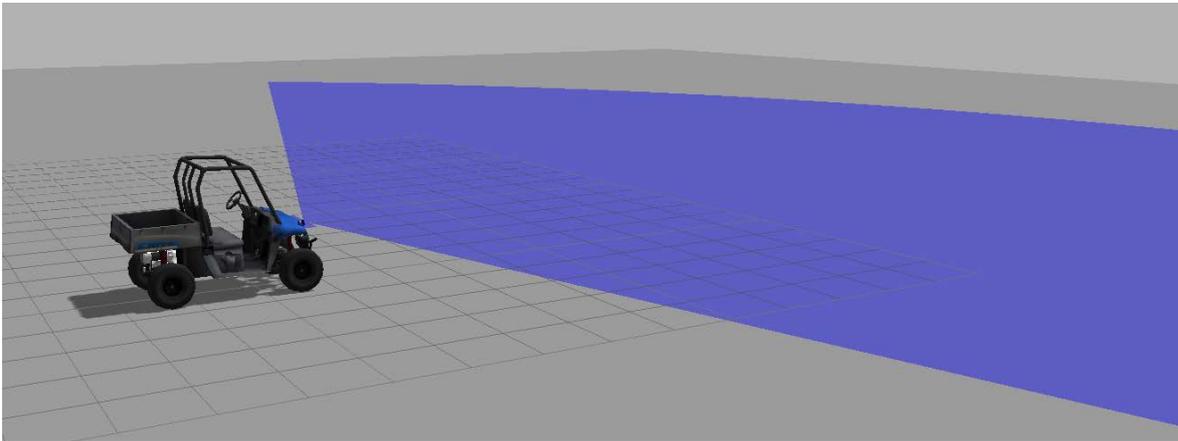
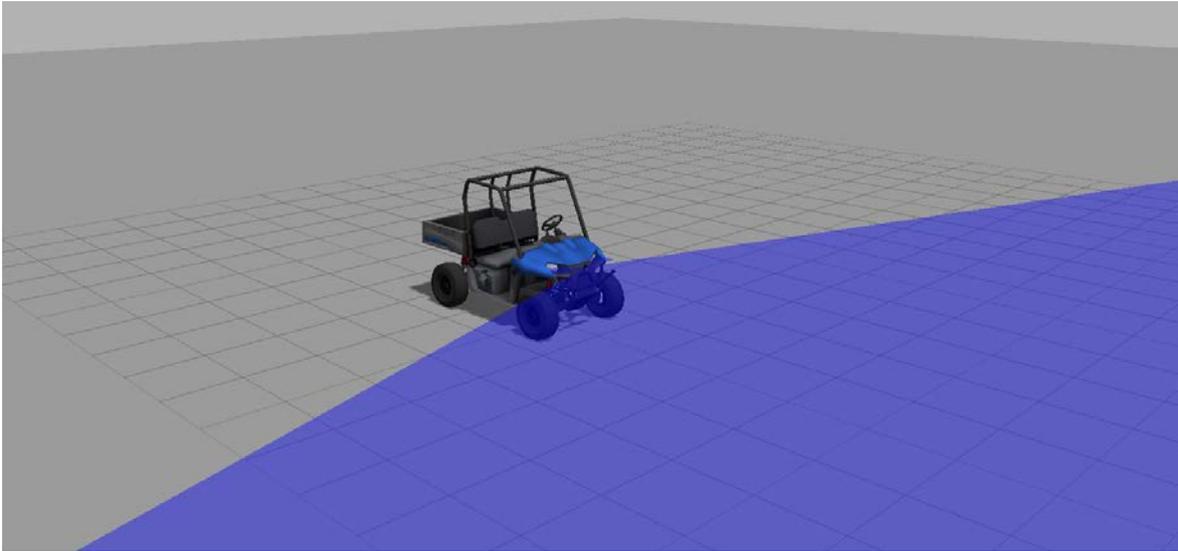
A continuación se muestra el resultado del láser generado por el código anterior:



**Figura 5-1: Sensor simulado en Gazebo. Su barrido se muestra en color azul**

Como siguiente paso, se le va a acoplar a este sensor creado, el vehículo. Para ello se coge del Gazebo un modelo ya creado llamado Polaris Ranger EV, muy similar al Yamaha Rhino.

Modificando algunas funciones del fichero model.sdf del Polaris Ranger se le acopla el sensor láser. El resultado se puede ver a continuación:



**Figura 5-2: Sensor y vehículo simulado en Gazebo.**

Se puede ver que el vehículo hace que la apertura angular del sensor se reduzca, debido a que se capta parte del coche y por tanto la información captada por el sensor no es de utilidad.

Como último paso, el sensor debe de girar con respecto al eje “x” tal y como se ha explicado anteriormente. Para lograr esto, se utiliza un nodo ROS que publica un mensaje hacia Gazebo de modo que este lo entienda y lo ejecuta. Este nodo llama “gzhokuyo\_rotator\_node.cpp” en el que la velocidad de giro es plenamente configurable con sólo cambiar un parámetro. El código de este nodo es bastante sencillo, de modo que se expone a continuación junto con comentarios a modo de comprender mejor lo que hace cada línea del código:

```
#include "ros/ros.h"
#include "geometry_msgs/Twist.h"
#include "gazebo_msgs/SetModelState.h"
#include "gazebo_msgs/ModelState.h"
#include <sstream>
#define pi 3.14159265

int main (int argc, char **argv)
```

```

{
    int count=0;
    float velocidad=0.0;

    ros::init (argc,argv,"gzhokuyo_rotator"); //inicialización ROS
    ros::NodeHandle n;
    ros::Duration(2.0).sleep();

    if (argc!=2) //comprobación de errores
    {
        ROS_ERROR("Ha olvidado meter una velocidad");
        exit(1);
    }

    velocidad=(atof(argv[1]))*2*pi/60; //conversion rpm a rad/s

    geometry_msgs::Pose start_pose; //inicialización de la posición
    start_pose.position.x=0.0; //en este caso, el sensor está a
    start_pose.position.y=0.0; //1 metro del suelo.
    start_pose.position.z=1.0;
    start_pose.orientation.x=0.0; //inicialización orientación
    start_pose.orientation.y=0.0;
    start_pose.orientation.z=0.0;
    start_pose.orientation.w=1.0;

    geometry_msgs::Twist start_twist; //se incorpora la velocidad
    start_twist.linear.x=0.0; //angular que se desea.
    start_twist.linear.y=0.0;
    start_twist.linear.z=0.0;
    start_twist.angular.x=velocidad;
    start_twist.angular.y=0.0;
    start_twist.angular.z=0.0;

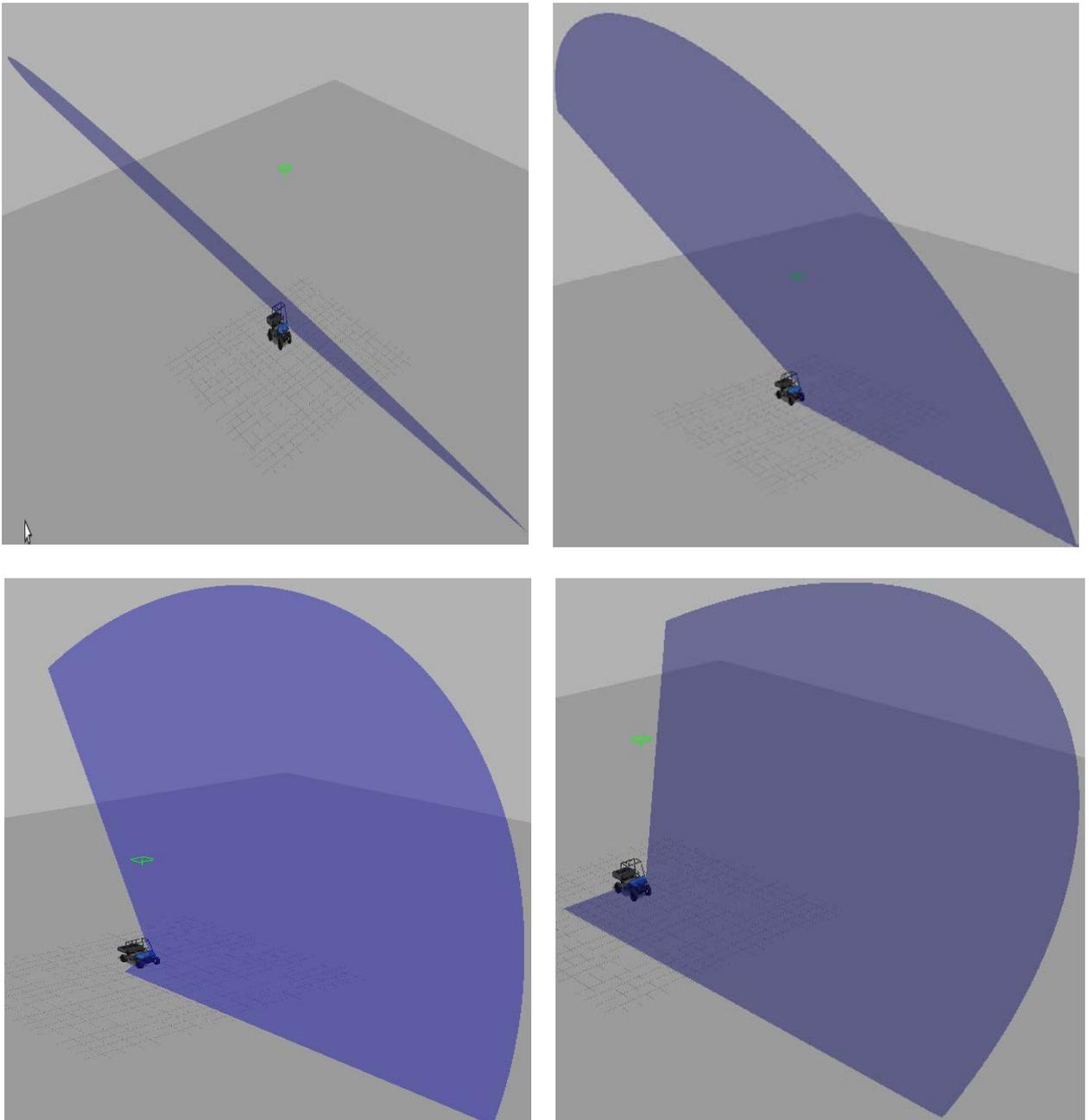
    gazebo_msgs::ModelState modelstate; // configuración de la petición
    modelstate.model_name=(std::string)"hokuyo30lx";
    modelstate.reference_frame=(std::string)"world";
    modelstate.pose=start_pose; // incorporación de posición
    modelstate.twist=start_twist; // incorporación de orientación

    gazebo_msgs::SetModelState setmodelstate; // definicion del mensaje
    ros::ServiceClient principal =
n.serviceClient<gazebo_msgs::SetModelState>("/gazebo/set_model_state");

    setmodelstate.request.model_state = modelstate;
    principal.call(setmodelstate); // petición de servicio para que
// se comience a girar el sensor
}

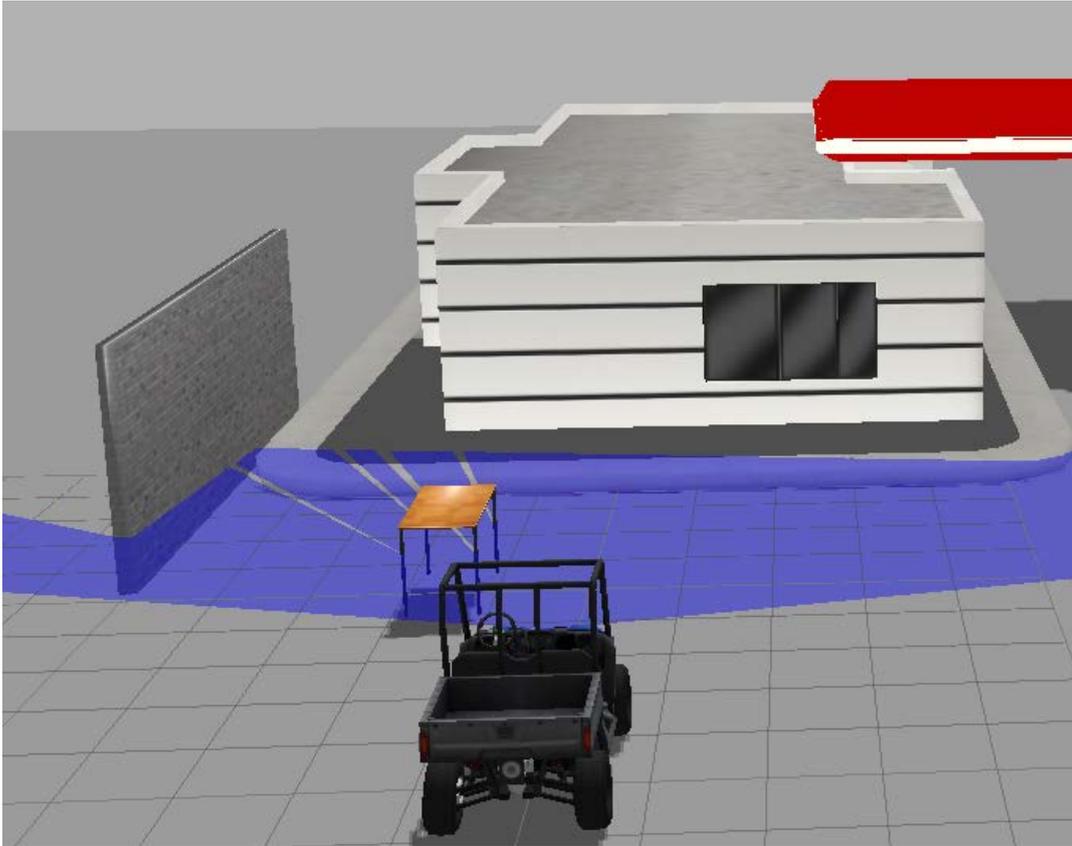
```

El resultado de ejecutar este código se puede ver a continuación:



***Figura 5-3: Barrido realizado por el sensor en un instante de tiempo dado mientras el sistema se encuentra girando respecto al eje x***

Como se ha mencionado anteriormente, este entorno de simulación permite una gran variedad de alternativas para estudiar el comportamiento del sistema sensorial. Un ejemplo de la inmensa cantidad de pruebas que se pueden realizar se puede ver a continuación, en la que se puede ver si se detectarían las patas de una mesa:



*Figura 5-4: Barrido realizado por el sensor en un entorno con obstáculos. Se puede ver claramente cómo las patas de la mesa son detectadas*

Ahora bien, para disponer de los datos recogidos por el sensor, lo que se tiene que hacer es suscribirse al tema que está exportando este sensor en el Gazebo. Poniendo en consola “rostopic list” aparecerá un listado de todos los temas activos por los que algún programa está publicando y con “rostopic echo EJEMPLO” se suscribirá al tema “EJEMPLO” y se comenzarán a recibir los mensajes. En este caso particular del sensor. El tema al que se debe de suscribir es “scan”.

Llegados a este punto, ya se dispone del sistema sensorial simulado en un escenario que se puede modificar fácilmente y cuyo barrido láser se puede ver de manera sencilla y clara. Por tanto, ya se ha construido la base del desarrollo software. Cabe mencionar que tanto el sistema real como el creado en Gazebo tienen exactamente las mismas características, de manera que todo el software desarrollado es válido tanto para simular en Gazebo como para el sistema real.

## **5.2.3 Desarrollo software del sistema en ROS**

### **5.2.3.1 Introducción**

A continuación se van a explicar cada uno de los nodos que se han programado, de manera que se pueda entender el por qué de estas estructuras y conexiones.

En cada uno de estos nodos, hay una estructura que permanece invariante en todos ellos. A continuación se mostrará un pequeño ejemplo de esta estructura invariante.

```

int main(int argc, char *argv [])
{
    //Inicialización y nombramiento del nodo
    ros::init(argc, argv, "NombreDelNodo");

    //Constructor acceso a las comunicaciones de ROS
    ros::NodeHandle n;

    // Definición del tipo de mensaje a enviar. En este caso el mensaje se
    llama “/cadena” y tiene una cola de 1000 mensajes.
    publi=z.advertise<std_msgs::String>("/cadena", 1000);

    // Definición frecuencia de ejecución del nodo. En este caso 10 Hz
    ros::Rate loop_rate(10);

    while (1)
    {
        // Suscripción al tópico “/msn”, si se recibe mensaje, el programa
        realiza la función “recibido”
        ros::Subscriber Subscriber = n.subscribe("/msn", 1000, recibido);
        ros::spinOnce();

        // Se envía el string contenido en msg por el tópico “/cadena”
        publi.publish(msg);

        // Efectua la frecuencia de ejecución puesta anteriormente
        loop_rate.sleep();
    }
    return 0;
}

```

Como se puede ver, Se publican mensajes con el contenido de “msg” de tipo String bajo el tópico “/cadena” y se recibe los mensajes publicados en “/msn”. Dicha estructura es básica para cada uno de los programas desarrollados, teniendo todos ellos un publicador y varias suscripciones.

Vista ya la estructura básica de todos los nodos desarrollados se procede a explicar cada uno de ellos.

### 5.2.3.2 Nodo gzhokuyo\_node

Este nodo tiene la tarea de publicar mensajes con los datos recogidos por el sensor en Gazebo. Para conseguir disponer de los datos recogidos por el sensor en Gazebo se recurre a unas librerías de transporte de éste y a su unión con ROS. Por tanto, este nodo lo único que hace es publicar mensajes con información del sensor y los datos escaneados por este.

Estos mensajes poseen la siguiente estructura de datos:

```

std_msgs/Header header // Estructura de datos de cabecera, posee un identificador
                        // y el tiempo de generación de los datos con precisión de
                        // nanosegundos

float32 angle_min      // Ángulo de inicio del sensor laser [rad]
float32 angle_max      // Ángulo final del barrido laser [rad]
float32 angle_increment // Increment de ángulo entre cada punto [rad]
float32 time_increment // Tiempo transcurrido entre medidas [segundos]

```

float32 scan_time	// Tiempo entre barridos láser consecutivos [segundos]
float32 range_min	// Rango mínimo del sensor [metros]
float32 range_max	// Rango máximo del sensor [metros]
float32[] ranges	// Medidas tomadas por el sensor [metros]
float32[] intensities	// Datos de la intensidad de las medidas tomadas

En el interior de “ranges” se encuentran los datos recogidos por el sensor, en forma de distancia. Juntando este dato con el ángulo en el que dicho dato ha sido tomado, se obtienen coordenadas polares.

La dimensión de la tabla “ranges” es de 1080 puntos, de acuerdo con la hoja de datos del sensor utilizado. En cambio, el campo “intensities” se ha desactivado debido a que no interesa la intensidad de las medidas tomadas, destinadas a la clasificación de los objetos según su composición material. Al desactivarlo, se están ahorrando 1080 medidas por mensaje.

El resto de los campos, salvo la cabecera, están basados en la hoja de datos del sensor. La cabecera se rellena a partir de un identificador que se le quiera poner y una marca de tiempo, totalmente necesaria para poder saber en qué instante de tiempo se han creado las medidas.

El tópico con el que se transmite este mensaje se llama “/scan”. A continuación se procede a visualizar un mensaje publicado por este nodo, mediante el comando rostopic echo “/scan”:

```

---
header:
  seq: 2664
  stamp:
    secs: 1415191589
    nsecs: 816814130
  frame_id: laser
angle_min: -2.26892805099
angle_max: 2.26892805099
angle_increment: 0.00436332309619
time_increment: 1.73611151695e-05
scan_time: 0.0250000003725
range_min: 0.0230000000447
range_max: 60.0
ranges: [0.6549999713897705, 0.6549999713897705, 0.6639999747276306, 0.6639999747276306, 0.6639999747276306, 0.6610000133514404, 0.6650000214576721, 0.6679999828338623, 0.6769999861717224, 0.6809999942779541, 0.6840000152587891, 0.6840000152587891, 0.6850000023841858, 0.6840000152587891, 0.6850000023841858, 0.7070000171661377, 0.7070000171661377, 0.6990000009536743, 0.6990000009536743, 0.7069999752044678, 0.7120000123977661, 0.7239999771118164, 0.7279999852180481, 0.7289999723434448, 0.7289999723434448, 0.7519999742507935, 0.7929999828338623, 0.8040000200271606, 0.8109999895095825, 0.8109999895095825, 0.8109999895095825, 0.8029999732971191, 0.6869999766349792, 0.6610000133514404, 0.6600000262260437, 0.6539999842643738, 0.6520000100135803, 0.6520000100135803, 0.6510000228881836, 0.6520000100135803, 0.6610000133514404, 0.6629999876022339, 0.6629999876022339, 0.6600000262260437, 0.646000027655553, 0.6440000000000000, 0.6440000000000000, 0.6440000000000000, 0.6570000000000000]

```

*Figura 5-5: Mensaje de ejemplo de tipo laserScan*

### 5.2.3.3 Nodo urg\_node [43]

Este nodo ha sido facilitado por HOKUYO, de manera que el sensor exporte los datos directamente a ROS sin que sea necesaria una programación adicional. El tipo de mensajes que publica es LaserScan y el nombre es “/scan”. Como se puede observar, este nodo y el gzhokuyo\_node publican en el mismo tópico, pero esto no supone un problema debido a que o se realiza una simulación con Gazebo o con el sensor real, de modo que dependiendo de esto se ejecutarán unos nodos u otros.

#### 5.2.3.4 Nodo gzhokuyo\_pose\_node

Nodo encargado de exportar desde Gazebo la orientación y posición del sensor. Por tanto, hace las funciones de encoder dentro de Gazebo. Dicha información es obtenida suscribiéndose al tópico “~/hokuyo30lx/link/laser/scan”. En este caso, el mensaje publicado es de tipo QuaternionStamped, y se le llama “/quaternionENCODER”.

Está compuesto de una estructura de datos cabecera como el laser\_scan y otra estructura denominada Quaternion, que posee cuatro variables de tipo float64 llamadas “w”, “x”, “y” y “z”.

La representación de una rotación a partir del cuaternión es un método sencillo debido a que simplifica toda la complejidad asociada a las rotaciones a una multiplicación de variables [44]. Es necesaria la cabecera para poder saber en qué instante de tiempo se ha dado ese cuaternión específico del mensaje.

A continuación se puede ver un mensaje de tipo QuaternionStamped creado por este nodo cuando el sistema sensorial se encuentra girando:

```
---
header:
  seq: 1549
  stamp:
    secs: 1415192812
    nsecs: 451282252
  frame_id: angleENCODER
quaternion:
  x: -0.862641704938
  y: -0.0
  z: -0.0
  w: -0.50581546922
---
```

*Figura 5-5: Mensaje de ejemplo de tipo QuaternionStamped*

#### 5.2.3.5 Nodo rls30\_controlnode

Este nodo es el encargado de publicar por ROS los datos obtenidos por el controlador EPOS (es decir los datos del encoder) y los datos de la IMU. De manera que toda la información obtenida pueda ser analizada y procesada en el entorno ROS. De esta manera, este nodo publica los datos obtenidos del controlador a partir del encoder mensajes llamados “/quaternionENCODER” de tipo QuaternionStamped, idéntico a como se hacía en el nodo gzhokuyo\_pose\_node. Al igual que con el nodo urg\_node y gzhokuyo\_node, sólo puede estar activo uno de ellos, dependiendo de qué se esté analizando (entorno real o entorno simulado de Gazebo).

Adicionalmente, este nodo publica en el tópico “/quaternionIMU” mensajes de tipo QuaternionStamped con las lecturas ofrecidas por la IMU. La marca de tiempo de la cabecera, igual que en casos anteriores, es necesaria para saber en qué instante se ha producido dicha rotación del mensaje en cuestión.

### 5.2.3.6 Nodo P\_LECTOR3D

Es el nodo encargado del procesamiento de la información. En primer lugar este nodo se suscribe al tópico “/scan”, que posee los datos recogidos por el sensor láser y los transformar en coordenadas cartesianas en el espacio bidimensional. Para ello es necesario procesar los diferentes campos de este tópico y realizar la transformación apropiada. Dicha transformación se detalla en pseudocódigo a continuación a partir de la figura anterior:

```
Angle_min:=-2.26892 // Radianes
Angle_max:=2.26892 // Radianes
Angle_increment:=0.00436332 // Radianes

Num_puntos:= dimension(Ranges) // 1080 puntos

For (i=0; i<Num_puntos; i++)
{
    Coordenada_y=Ranges[i]*sin(Angle_min+i*Angle_increment);
    Coordenada_x=Ranges[i]*cos(Angle_min+i*Angle_increment);
}
```

Una vez hecho esto, el nodo también se suscribe al “quaternionENCODER”, para dotar a las medidas de una tercera componente cartesiana. Para lograr esto, es de vital importancia que el nodo sea capaz de encolar ambos mensajes en una cola FIFO (First Input First Output), sincronizar de manera estricta las medidas del sensor con las medidas del encoder y realizar la operación adecuada. El encolado se hace necesario debido a que los datos que se están analizando pueden ser reemplazados por un nuevo mensaje recibido, de manera que es importante realizar un control en el ordenamiento y protección de los datos.

A la hora de sincronizar los datos del encoder y del sensor, es necesario analizar previamente cual es la frecuencia de ambos tipos de información, qué marcas de tiempo llevan asociadas y cómo se debería de hacer dicha sincronización. Todo esto se va a resolver de manera detallada a continuación:

- Estudio datos ofrecidos por el láser

Con el mensaje de tipo laserScan expuesto en la figura anterior, se procede analizar dicha información. Se tiene que el mensaje ha sido generado (no rellenado, que es distinto) en el tiempo disponible en la cabecera. Teniendo esto claro, ahora se dispone de 1080 puntos, cada uno de ellos obtenido en un tiempo distinto, que se deben sincronizar con los datos del encoder. Dichos tiempos se calculan de la siguiente manera:

```
// Se guarda en una variable el mensaje recibido
_data=guardar_mensaje_laserScan

// Obtención del tiempo de cabecera (tiempo de inicio)
t_data = _data.header.stamp.sec +_data.header.stamp.nsec*(1E-9);

//Incremento de tiempo entre puntos
increment_scan=_data.time_increment;
```

```

// Tiempo entre escáneres
total_scan=_data.scan_time;

//Tiempo de inicio y fin del escaner
scan_ini=t_data+180*increment_scan; //segundos
scan_fin=t_data+total_scan-180*increment_scan; //segundos

for (contador=0; contador<Num_puntos; contador++)
{
    time_scan=scan_ini+(contador)*increment_scan;
}

```

El tiempo que posee el mensaje en cabecera es el tiempo base para calcular los demás parámetros. Por tanto, el tiempo de inicio de los datos será el tiempo que aparece en la cabecera más un incremento de tiempo. Definido el tiempo de inicio y el tiempo final, solo queda realizar un bucle de manera que dichos tiempos estén separados `increment_scan` entre muestras consecutivas. Esto se demuestra a continuación:

Número de puntos obtenidos:=1080.

Número de puntos reales:=1440.

Tiempo escáner:=25 ms.

Incremento de tiempo entre puntos consecutivos:= $1.73611151 \cdot 10^{-5}$  segundos.

El número de puntos reales que se obtiene del sensor, son al recorrer el láser  $360^\circ$  pero como se ha expuesto anteriormente, la apertura máxima del sensor es de  $270^\circ$  y por tanto dicho número de puntos se reduce a 1080, pero hay que tener esto en cuenta para calcular el tiempo de cada punto, debido a que el tiempo de inicio del escáner no se corresponde con el primer punto de los datos obtenidos, sino con el primer punto real. Por tanto se tiene un contorno de puntos reales entre 0 y 1439 entre los cuales están los tiempos de los puntos de los que se entrega información. Por ello, hay que recortar una cierta cantidad de puntos del inicio y del final del contorno para que dichos tiempos se correspondan con las medidas tomadas y adquiridas por el sensor. Por tanto, el contorno el que se encuentran las 1080 medidas que se necesitan es desde 180 hasta 1259. Con lo que el tiempo de escaneo del primer punto es el tiempo de inicio del escáner más 180 veces el tiempo transcurrido entre la adquisición de puntos consecutivos.

El funcionamiento del sensor no está basado en un giro con desaceleraciones y aceleraciones en sentido contrario cuando se alcanza el mínimo o máximo ángulo de apertura, sino que es un movimiento continuo de  $360^\circ$ . Por ello se tienen las características explicadas anteriormente.

- Estudio datos ofrecidos por el encoder

Los datos del encoder (o del nodo Gazebo que hace las mismas funciones) viene dado por un mensaje tipo `QuaternionStamped`, en el que se tiene la orientación y el tiempo que el que se ha obtenido este. Solamente queda mencionar que la adquisición de estos datos debe ser a una frecuencia mayor que la del sensor láser, de manera que por cada escáner que realice el láser, se tengan al menos 3 medidas del encoder para poder sincronizar con poco error. Obviamente, cuanto mayor sea la velocidad de giro, menor precisión se va a conseguir a la hora de asignar a cada punto obtenido el ángulo de orientación con la que se ha tomado dicho punto.

En este proyecto, la velocidad del encoder (tanto el simulado como el real) es tal que se obtiene datos de la orientación del sensor en un rango de 4 a 8 milisegundos.

- Sincronización

Una vez que se tiene claro los tiempos en los que se ha tomado cada punto del sensor láser y los datos y frecuencia del encoder, se procede a realizar la sincronización. Para realizar esto, la estrategia llevada a cabo consiste en coger la cabecera de los mensajes del encoder y compararlo con los valores `scan_ini` y `scan_fin`, para saber si dicha información se corresponde con ese barrido. Si coincide, el mensaje del encoder se guarda en una tabla y se coge otro mensaje encoder. Se repite el proceso hasta que haya un valor de cabecera en el encoder que exceda el `scan_fin`.

A continuación se puede ver un ejemplo de esta tabla:

```
scan_ini 1415193246.216906,scan_fin 1415193246.235656,time_angle 1415193246.215144
scan_ini 1415193246.216906,scan_fin 1415193246.235656,time_angle 1415193246.220811
scan_ini 1415193246.216906,scan_fin 1415193246.235656,time_angle 1415193246.226763
scan_ini 1415193246.216906,scan_fin 1415193246.235656,time_angle 1415193246.233102
scan_ini 1415193246.216906,scan_fin 1415193246.235656,time_angle 1415193246.238236

TABLA[0] angle:2.513850 t:215144429 ns
TABLA[1] angle:2.518283 t:220810695 ns
TABLA[2] angle:2.522769 t:226763110 ns
TABLA[3] angle:2.527760 t:233101845 ns
TABLA[4] angle:2.532979 t:238236075 ns

scan_ini 1415193246.242458,scan_fin 1415193246.261208,time_angle 1415193246.238236
scan_ini 1415193246.242458,scan_fin 1415193246.261208,time_angle 1415193246.250048
scan_ini 1415193246.242458,scan_fin 1415193246.261208,time_angle 1415193246.253678
scan_ini 1415193246.242458,scan_fin 1415193246.261208,time_angle 1415193246.258154
scan_ini 1415193246.242458,scan_fin 1415193246.261208,time_angle 1415193246.264347

TABLA[0] angle:2.532979 t:238236075 ns
TABLA[1] angle:2.541391 t:250048114 ns
TABLA[2] angle:2.544812 t:253678063 ns
TABLA[3] angle:2.547622 t:258153981 ns
TABLA[4] angle:2.552561 t:264346688 ns
```

**Figura 5-6: Ejemplo de sincronización entre mensajes del láser y mensajes de la posición del motor**

Con los mensajes del encoder organizados en una tabla, se coge cada uno de los puntos del láser y se compara con la tabla, hasta encontrar que dicho punto está entre dos valores de tiempo del encoder. Por tanto, sabiendo que el dato láser se encuentra en una determinada región delimitada por dos puntos (ángulos), y sabiendo el tiempo que ha transcurrido entre uno y otro, se calcula la ecuación de la recta que pasa por dos puntos, de manera que se obtiene una gran aproximación.

Para terminar, se muestra en pseudocódigo todos estos procedimientos:

Si se recibe mensaje de encoder → encolar cola encoder  
Si se recibe mensaje de datos → encolar cola datos

1. Desencolar un mensaje de cola datos
2. Obtener tiempos del barrido del escáner realizado en dicho mensaje
3. Desencolar un mensaje de cola encoder
4. Si tiempo de encoder < scan\_fin, guardar valor encoder en tabla y volver a 2. En caso contrario, se sigue.
5. Comparación de cada punto del escáner con puntos del encoder. Una vez encontrado que dicho punto es mayor que un mensaje encoder y menor que uno de ellos, se calcula ecuación de la recta que pasa por dos puntos.
6. Otras instrucciones....
7. Se publica el mensaje tipo pointCloud y vuelta al punto 1.

Por último, el nodo se suscribe al “quaternionIMU”, con el objetivo de transformar las medidas obtenidas por el sensor a un sistema de coordenadas invariante a las rotaciones u orientación del vehículo, de modo que sea sencillo generar un mapa 3D a partir de las transformaciones a sistema de coordenadas fijo. Para lograr esto, se necesita sincronizar de nuevo los datos, pero sin la necesidad de que esta sea estricta y sin que sea necesario encolar los mensajes para su posterior procesamiento, debido a que los datos facilitados por la IMU son mucho más lentos y no es importante que se derroche información ya que no va a haber cambios bruscos en el vehículo, y si se producen no van a ser de un periodo menor al de captación de la IMU y la capacidad de análisis de este nodo. Además de esta nueva sincronización, se ha programado la IMU de tal manera, que cuando el software del sistema sensorial sea lanzado, se tome la situación en ese momento de la IMU como inicial.

Al necesitar tantas suscripciones, es necesario recurrir a la instrucción `ros::MultiThreadSpinner`, que crea un número configurable de hilos todos ellos pendientes de si se publica información en cada uno de los temas suscritos. Si un hilo capta información por un tópico, se ejecutará la correspondiente función asociada y volverá a estar pendiente de si se recibe más información. Si no se opta por el sistema multihilo ocurrirá que mientras se está ejecutando la información asociada a una suscripción recibida, llegue información por otro tópico y dichos datos no sean recogidos ni actualizados, provocando lentitud, pérdida de datos y aleatoriedad en el proceso de información, lo que haría que la generación de puntos 3D sea un fracaso al no disponer de la información correcta en cada instante de tiempo.

Los resultados obtenidos a partir de todas estas transformaciones son publicados en un t3pico llamado “/cloud” de tipo PointCloud. Este tipo de datos es analizado a continuaci3n:

```
std_msgs/Header header // Datos de cabecera
uint32 seq // N3mero identificador
time stamp // Marca de tiempo del mensaje
string frame_id // Identificador del mensaje

geometry_msgs/Point32[] points
float32 x // coordenada cartesiana x en metros
float32 y // coordenada cartesiana y en metros
float32 z // coordenada cartesiana z en metros

sensor_msgs/ChannelFloat32[] channels
string name // Sin uso en este nodo
float32[] values // Sin uso en este nodo
```

Por todo esto, la complejidad de este nodo es grande, no obstante el coste computacional no es muy elevado. Es necesario recordar que el sistema debe procesar los datos en tiempo real, con lo que se dispone de menos de 25 milisegundos para procesar cada mensaje, para que no se produzcan retardos que hagan que el sistema no funcione correctamente.

A continuaci3n se puede ver la parte representativa de un mensaje publicado por este nodo:

```
--
header:
  seq: 4890
  stamp:
    secs: 1415191983
    nsecs: 852326631
  frame_id: laser
points:
-
  x: 0.0
  y: 0.0
  z: 0.0
-
  x: 0.0
  y: 0.0
  z: 0.0
```

*Figura 5-7: Ejemplo de mensaje de tipo PointCloud*

Por supuesto, “points” contiene 1080 puntos, cada uno de ellos con sus coordenadas tridimensionales de modo que solo se muestra una peque1a porci3n de dicho mensaje.

### 5.2.3.7 Nodo `pcl_filter_ros`

Este nodo se suscribe al tópico `"/cloud"` publicado por el `P_LLECTOR3D`, procesa los datos cartesianos adquiridos y publica mensajes de tipo `PointCloud2` bajo el tópico `"/assembler_cloud"`. El objetivo de este nodo es el de filtrar todos aquellos datos cartesianos recibidos, de manera que sin apenas perder información del entorno se reduzca el número de puntos notablemente. Para poder realizar este filtrado correctamente, se necesita juntar gran cantidad de información 3D, ya que cuantos más datos haya, más densidad de puntos habrá en el entorno y por tanto se realizará mejor filtrado, al haber muchos puntos redundantes. Además, con esta estrategia de juntar varios escáneres en uno sólo, se puede construir fácilmente un mapa 3D local del entorno alrededor del vehículo. No obstante, dicho almacenamiento de información no puede hacer que el sistema deje de ser en tiempo real, por lo que hay que buscar un equilibrio. Para solucionar este problema se ha realizado la siguiente estrategia:

1. Se recibe mensaje desde tópico `"/cloud"`.
2. Si `contador <= Dimensión_almacenamiento`: se almacena, se filtra y se incrementa contador. En caso contrario, se tira el mensaje más antiguo y se pone el mensaje recibido en dicha posición.
3. Se publica todo el conjunto.
4. Se vuelve al punto 1.

Con este esquema no se pierde el tiempo real del sistema, pero se obtiene una gran redundancia de información. Por tanto se ha estimado que en vez de refrescar y enviar todo el bloque de datos de dimensión configurable por cada mensaje recibido, se haga cada un cuarto de la capacidad máxima del bloque. Así pues, si el bloque tiene se ha configurado para agrupar 40 mensajes, el refresco y la publicación de toda la información se haga cada 10 mensajes, es decir 250 milisegundos. Como ya se ha dicho, la dimensión total de almacenamiento es configurable y se puede adaptar a la velocidad de giro del sensor o la velocidad a la que esté circulando el vehículo, como se prefiera. Lo único a tener en cuenta es que cuanto más información en el bloque y cuanto más cerca esté ésta, mejor filtrado se podrá realizar.

Una vez creado el bloque de datos, se procede a filtrar la información que contiene para reducir la gran cantidad de puntos que han sido obtenidos. Para realizar este filtrado se recurre a unas librerías adicionales, denominadas `"point cloud library"` [45], que ofrecen métodos de filtrado de manera sencilla y directa. Únicamente se debe de hacer una pequeña transformación en la información contenida en el bloque, ya que en vez de realizarse el filtrado con `sensor_msgs::PointCloud`, se realiza con el tipo `pcl::PointCloud2`. El filtro elegido consta en convertir cada punto en una caja de dimensión configurable en los parámetros del filtro, y en estudiar la vecindad que tiene dicho punto. Si ésta vecindad es elevada y lo suficientemente junta como para solaparse con este modelo de cajas, los puntos vecinos son eliminados ya que no incorporan información adicional al sistema. Una vez realizado el filtrado, se vuelven a transformar los datos filtrados, pero esta vez al tipo `PointCloud2` [56].

Esta estructura de datos es mucho más compacta que `PointCloud`, de manera que el coste computacional se reduce considerablemente. Aun así, este nodo es sin duda el que más coste computacional tiene debido a la creación del bloque de información y su posterior filtrado.

### 5.2.3.8 Nodo rls30\_mapper

Este nodo se suscribe al tema “/assembler\_cloud” publicado por el nodo anterior y publica dos temas, uno de tipo pointCloud cuyo t3pico es “/2Dmap\_local” y el otro es tipo OccupancyGrid con t3pico “/2Dmap\_global”.

El objetivo es que a partir del bloque de datos filtrado, este nodo sea capaz de utilizar dicha informaci3n para realizar un mapa 2D del entorno, a partir del cual se puedan aplicar algoritmos de navegabilidad. Por tanto, se parte de datos 3D locales que deben ser convertidos en 2D globales.

La diferencia entre un mapa global y local es el punto de referencia escogido. En el local, se tiene como referencia el propio veh3culo, de manera que estos mapas 3nicamente representan el entorno actual del veh3culo. Los globales tambi3n representan los datos del sistema sensorial pero desde un sistema externo, fijo e invariante, de manera que si el veh3culo se mueve, el mapa global ir3 completando un mapa con su respectiva trayectoria y las medidas tomadas en cada instante de tiempo. Los mapas globales son ampliamente utilizados en la rob3tica m3vil ya que permite hacer un seguimiento de todo lo que ha escaneado el sensor y de la trayectoria seguida.

Para la correcta realizaci3n de ambos mapas, a partir de las medidas, se debe clasificar lo que es un obst3culo y lo que no lo es. El problema reside en que el suelo o el techo pueden ser considerados como obst3culo, lo que provocar3a que toda la regi3n se considere obst3culo, cuando en realidad no es as3. Para solucionar el problema del techo detectado como obst3culo, se ha optado por filtrar aquellos puntos con componente z superior a la altura del veh3culo, puesto que si supera 3sta no se colisionar3 en ning3n caso con dicho punto detectado. Para solucionar el segundo problema se ha optado por realizar el siguiente procedimiento:

1. Se le suma a la componente z de cada punto la altura a la que se encuentra el sensor del suelo.
2. Si el valor resultante es mayor o menor que un determinado umbral, se extrae su correspondiente coordenada x e y, y se consideran como obst3culo. En caso contrario, dichas medidas no se corresponden con un obst3culo.

Por tanto, el mapa local 2D es generado a partir de la publicaci3n de los puntos tratados como obst3culos. Para la realizaci3n del mapa global 2D se registran todos los datos (sean o no obst3culos) pero con un c3digo de colores. Si en una determinada regi3n se ha detectado un obst3culo, dicha regi3n se pone de color negro. En caso de que se obtengan datos pero que estos no constituyan una amenaza para el veh3culo, dicha regi3n se pone de color blanco. En el caso del mapa global, lo que se hace es publicar los mensajes pero a su vez se “recuerda” lo que se ha enviado anteriormente. Es decir, si alguno de los esc3neres del laser ha detectado un obst3culo en una determinada regi3n, dicho obst3culo aparecer3 siempre en el mapa y nunca desaparecer3. Por tanto, las publicaciones ofrecen un “refresco” al mapa ya generado, por lo que dicho mapa actualiza su almacenaje con estos nuevos datos pero seguir3 conteniendo los valores escaneados antiguos.

El m3todo para poner los puntos en un mapa 2D global es a partir de la realizaci3n de un mallado en el espacio. En este nodo se ha considerado un mallado de 10 cent3metros por 10 cent3metros. Por tanto, se debe realizar la correspondencia entre coordenadas cartesianas x e y, y la celda correspondiente a dicho punto.

### 5.2.3.9 Nodo Prueba

Como su propio nombre indica, este nodo ha sido creado como test para las posibles traslaciones del vehículo. En él se utiliza una clase llamada `tf`, que publica información acerca del estado del vehículo. Parte de esa información viene dada por un cuaternión para la orientación y por un vector de tres posiciones para las traslaciones. Este nodo capta estas características del vehículo y las publica de manera que otros nodos pueden hacer cálculos con dichos datos.

## 5.2.4 Lanzadores y esquema de conexión

### 5.2.4.1 Introducción

Una vez explicados todos los nodos realizados, se va a explicar los ficheros encargados de ejecutar todos estos nodos, de manera que no haya que ejecutarlos de uno en uno. La mayoría de estos nodos necesita de argumentos externos, los cuales se analizarán posteriormente para su total comprensión. Una vez se expliquen los lanzadores creados, se ejecutarán ambos sistemas (sistema simulado Gazebo y el real) y se expondrá un esquema en el que se podrá ver cada uno de los nodos activos del sistema, a qué temas se suscriben y qué temas publican.

### 5.2.4.2 Lanzadores

Por un lado se ha creado el lanzador “`gazebo_hokuyo.launch`” que contiene la siguiente información:

```
<launch>

<!--Se lanza el simulador Gazebo con el sistema sensorial creado -->
<include file="$(find gazebo_ros)/launch/empty_world.launch">
  <arg name="world_name" value="worlds/hokuyo30lx.world"/>
  <arg name="paused" value="false"/>
  <arg name="use_sim_time" value="true"/>
  <arg name="gui" value="true"/>
  <arg name="headless" value="false"/>
  <arg name="debug" value="false"/>
</include>

<!--Se lanza el nodo gzhokuyo_node bajo el nombre gzhokuyo_reader-->
<node name="gzhokuyo_reader" pkg="rls30" type="gzhokuyo_node" />

<!--Se lanza el nodo gzhokuyo_pose_node bajo el nombre gzhokuyo_pose-->
<node name="gzhokuyo_pose" pkg="rls30" type="gzhokuyo_pose_node" />

<!--Se lanza el nodo gzhokuyo_rotator_node bajo el nombre gzhokuyo_rotator-->
<!--Posee como argumento las revoluciones por minuto a las cuales se desea girar-->
<node name="gzhokuyo_rotator" pkg="rls30" type="gzhokuyo_rotator_node"
args="20"/>

<!--Se lanza el nodo P_LECTOR3D. Los argumentos son:-->
  <!-- args1: Puede tener dos valores; 1: Simulación GAZEBO -1: Sistema real.-->
```

```

    <!-- args2-6: Son los limitadores de rango del sensor. Se corresponde con: xmax,
        Ymin, Ymax, Zmin Y Zmax -->
    <!--args7: 1: Volcar a fichero scan.log los datos publicados, 0: No volcar-->
<node name="P_LECTOR3D" pkg="rls30" type="P_LECTOR3D" args="1 29 -10 10 -
10 1.5 0"/>

<!--Se lanza el nodo prueba -->
<node name="prueba" pkg="rls30" type="prueba"/>

<!--Se lanza el nodo pcl_filter_ros. Los argumentos son:-->
    <!--args1: Dimensión agrupación pointcloud-->
    <!--args2: 1: Volcar a scan_after_filter.log los datos publicados, 0: No volcar-->
<node name="pcl_filter_ros" pkg="rls30" type="pcl_filter_ros" args="20 0"/>

</launch>

```

Como se puede observar, falta el nodo rls\_mapper. Esto es debido a que se ha considerado que las diferentes pruebas que se realicen con el simulador Gazebo se realizarán con el vehículo estático de modo que está más enfocado a la construcción de un mapa 3D.

Por otro lado, se muestra la parte más representativa del lanzador del sistema real, llamado urg30lx\_hokuyo.launch:

```

<launch>

<!--Se lanza el nodo urg_node bajo el nombre de urg30lx_hokuyo_node-->
<node name="urg30lx_hokuyo_node" pkg="urg_node" type="urg_node" output="screen"
>
    <-- Se configura la dirección IP del sensor láser-->
    <param name="ip_address" value="192.168.13.110" type="string" />
    <!-- Se define si se quiere evaluar más de un barrido láser antes de entregar -->
    <!-- la información con el objetivo de corregir errores en las mediciones-->
    <param name="publish_multiecho" value="false" type="bool" />
    <!-- Se define si se quieren recoger datos sobre la intensidad de recepción -->
    <param name="publish_intensity" value="false" type="bool" />
    <!--Definición de ángulo de empuje y ángulo de finalización del barrido -->
    <param name="angle_min" value="-2.26892805099" type="double" />
    <param name="angle_max" value="2.26892805099" type="double" />
    <!-- Se define si se quiere ignorar un barrido laser. Si es 0, no se ignora ninguno -->
    <!-- Si por ejemplo es 1 se eliminará un escáner por cada uno que se transmita-->
    <param name="skip" value="0" type="int" />
</node>

<!--Se lanza el nodo rls30_control_node-->
<node name="rls30_control_node" pkg="rls30" type="rls30_control_node">
    <!-- Se define la velocidad de giro del sistema sensorial, en radianes por segundo -->
    <param name="start_speed" value="3.14159" type="double" />
</node>

```

```

<!--Se lanza el nodo prueba -->
<node name="prueba" pkg="rls30" type="prueba"/>

<!--Se lanza el nodo P_LECTOR3D. Los argumentos son:-- >
  <!-- args1: Puede tener dos valores; 1: Simulación GAZEBO -1: Sistema real.-->
  <!-- args2-6: Son los limitadores de rango del sensor. Se corresponde con: x_max,
    y_min, y_max, z_min y z_max -->
  <!--args7: 1: Volcar a fichero scan.log los datos publicados, 0: No volcar-->
  <node name="P_LECTOR3D" pkg="rls30" type="P_LECTOR3D" args="-1 30 -10 10 -
10 1.5 0"/>

<!--Se lanza el nodo pcl_filter_ros. Los argumentos son:-->
  <!--args1: Dimensión agrupación pointcloud-->
  <!--args2: 1: Volcar a scan_after_filter.log los datos publicados, 0: No volcar-->
  <node name="pcl_filter_ros" pkg="rls30" type="pcl_filter_ros" args="20 0"/>

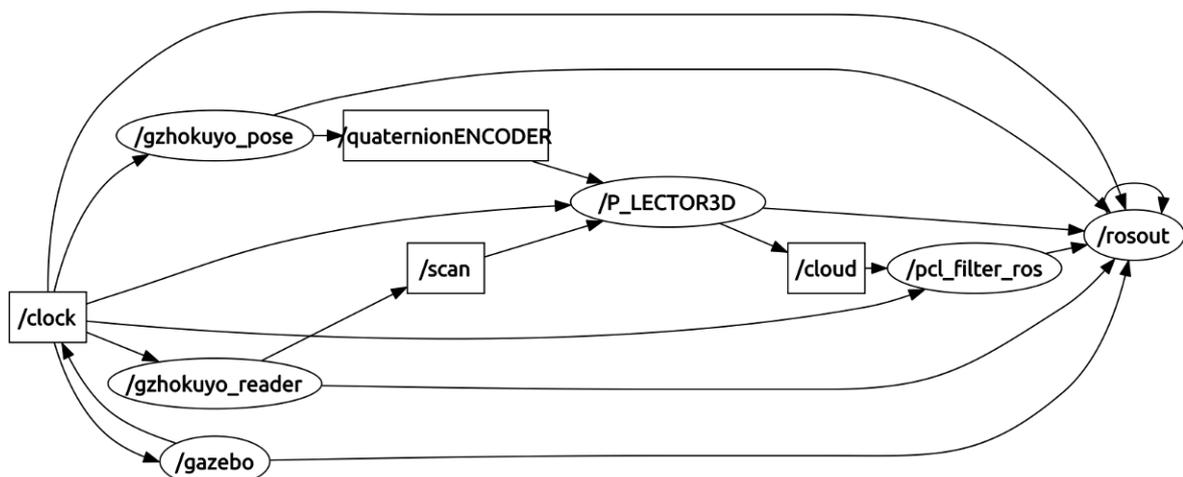
<!--Se lanza el nodo rls_mapper. Los argumentos son:-->
  <!--args1: Altura (en metros) del sistema sensorial con respecto al suelo-->
  <!--args2: 1: Volcar a fichero map.log los datos publicados, 0: No volcar-->
  <node name="rls_mapper" pkg="rls30" type="rls_mapper" args="0.95 1"/>

</launch>

```

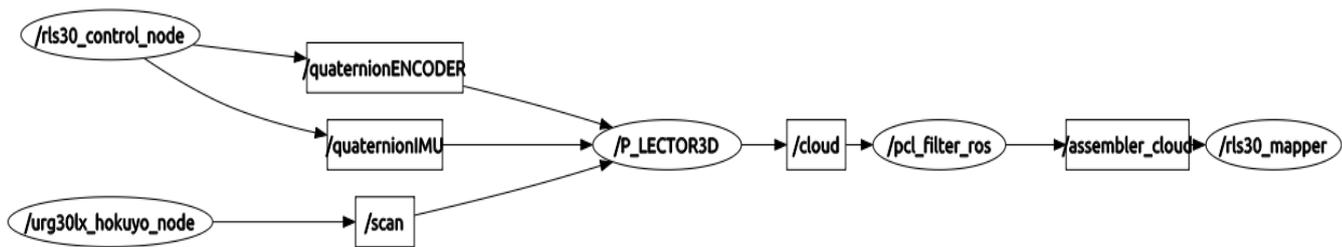
### 5.2.4.3 Esquema de conexión

A continuación se pueden ver los nodos y temas explicados anteriormente eligiendo la simulación con Gazebo. En estos esquemas, los rectángulos son los temas y las elipses los nodos.



**Figura 5-7: Esquema de conexión de temas y nodos con el lanzador gazebo\_hokuyo.launch**

Ahora se lanza el sistema real y se expone su esquema de conexiones, siguiendo la misma nomenclatura que en el caso anterior:



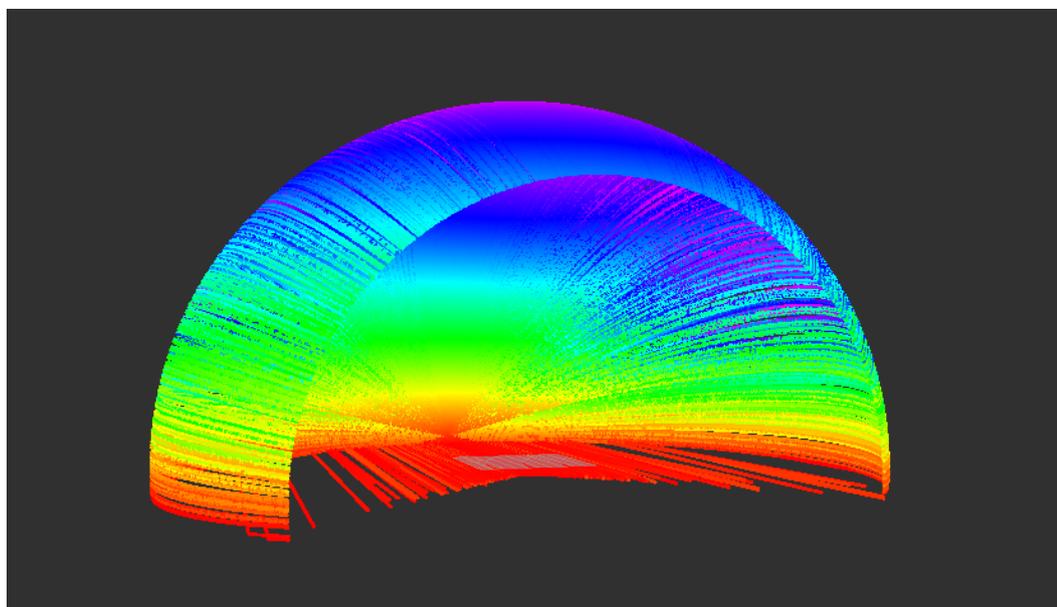
**Figura 5-8:** Esquema de conexión de temas y nodos con el lanzador `urg30lx_hokuyo.launch`

Dichos esquemas han sido construidos gracias al comando de `ros rqt_graph`, que como ya se ha explicado, permite ver de manera gráfica todas las conexiones del sistema.

#### 5.2.4.4 Resultados simulación

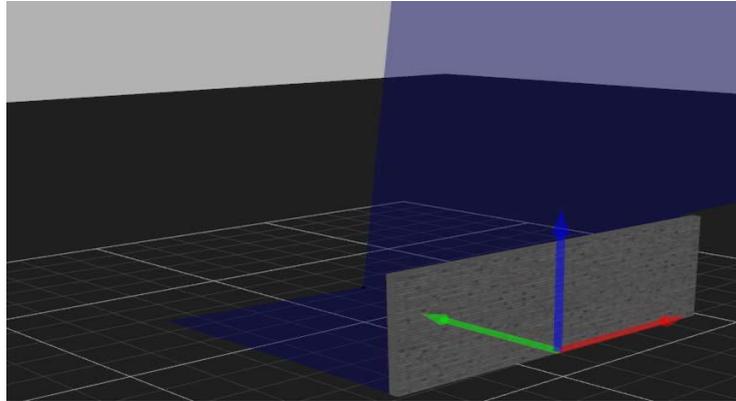
La simulación en Gazebo ha ayudado a la corrección de errores y al estudio del comportamiento del sistema sensorial debido a la gran variedad de caracterizaciones y a su fácil configuración. En este apartado se realizarán dos simulaciones en Gazebo, una sin obstáculo y otro con él. La salida del sistema (mapa 3D) se va a visualizar por medio del Rviz de manera que sea sencillo comprobar si los datos recogidos por el sensor se han tratado adecuadamente. Recordar que ambos sistemas (Gazebo y el sistema real) funcionan en tiempo real, de manera que si surgiera un elemento espurio que se cruzase en el escáner del láser, sería rápidamente detectado.

A continuación se muestra el mapa 3D de la simulación Gazebo del sistema sensorial sin obstáculos de manera que se espera ver el rango máximo del sensor únicamente limitado por el suelo de la simulación.



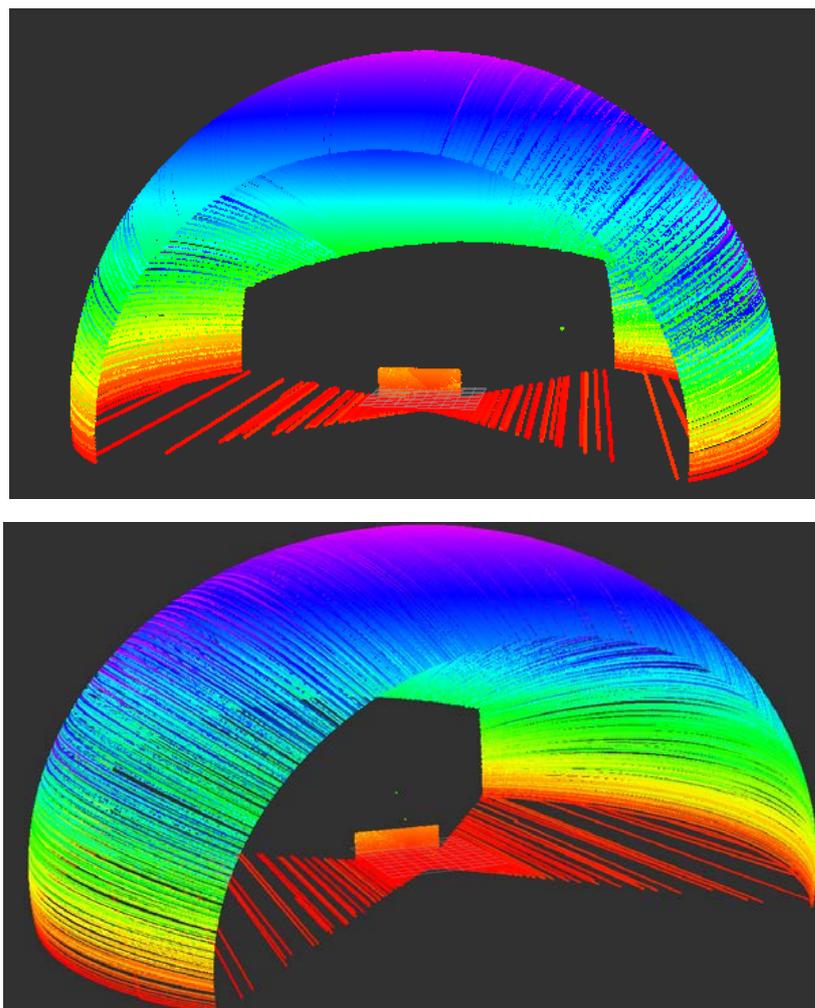
**Figura 5-9:** Mapa 3D generado a partir de la simulación en Gazebo sin obstáculos (únicamente el suelo)

Efectivamente se verifica el razonamiento anterior. Para verificar que el sistema funciona de manera correcta, se le incorpora a la simulación una pared y se comprueba si acto seguido de poner dicho objeto el sistema sensorial lo detecta. La pared ha sido colocada de la siguiente manera mientras el sensor se encontraba girando:



*Figura 5-10: Ejemplo de rotación laser y una pared en frente que debe ser detectada en tiempo real*

Los resultados se pueden ver a continuación, en los que aparece claramente el obstáculo establecido, el suelo y el rango máximo del láser.



*Figura 5-11: Mapa 3D generado a partir de la simulación en Gazebo con obstáculo*

# 6 Pruebas estáticas y dinámicas del sistema sensorial

---

Una vez verificado el correcto funcionamiento del software y que dicho funcionamiento es en tiempo real, se procede a realizar pruebas con el sistema real en diferentes condiciones, algunas de ellas estáticas en un ambiente controlado. En cada apartado siguiente, se describirá cómo se ha realizado la prueba, qué parámetros de configuración del sistema se han utilizado y se tomarán fotografías del entorno que se va a escanear. Por último se procederá a exponer el mapa 3D generado antes del filtrado y posterior a este y el mapa 2D global de cada caso. Además se ofrecerán datos de cuántos puntos ha sido capaz de eliminar el filtro diseñado, y se verificará que no se pierde apenas información.

Para la visualización del mapa 3D, se va a utilizar tanto el Rviz como el Octovis, para poder ver las diferencias notables entre ambos. En principio, el Octovis puede parecer mejor constructor del entorno 3D pero para representar las medidas con esta herramienta, se necesita volcar los datos cartesianos a un fichero de datos, y posteriormente transformar dicho fichero a una extensión entendible por Octovis. Por lo que el Octovis no ofrece mapas 3D en tiempo real, sino que los datos necesitan ser almacenados y posteriormente analizados. Por tanto, este sistema de visualización no será válido en entornos no controlados con obstáculos dinámicos. En cambio, con el Rviz sólo hay que suscribirse al tópic que se quiera y se puede visualizar en tiempo real, por lo que no requiere almacenar datos que se pueden considerar innecesarios.

En cada una de las pruebas se ha optado por limitar la velocidad máxima a 1.5 RPS. No obstante, esto no supone un problema ya que en el caso de las pruebas dinámicas con el vehículo no se han alcanzado velocidades superiores a 20 Km/h.

En las pruebas estáticas, no se requiere una velocidad de giro elevada, sino que es recomendable unas velocidades bajas para poder tener más densidad de puntos en cada región. Al no moverse el sistema sensorial del lugar donde está instalado, los obstáculos no constituyen ningún riesgo para el sistema y por tanto, no importa que dichos obstáculos no sean detectados inmediatamente. Independientemente de la velocidad de giro, el sistema siempre representará el entorno en tiempo real.

## 6.1 Prueba estática en habitación

La primera prueba se ha realizado con el sistema sensorial instalado en una habitación en la que todos los obstáculos son estáticos.

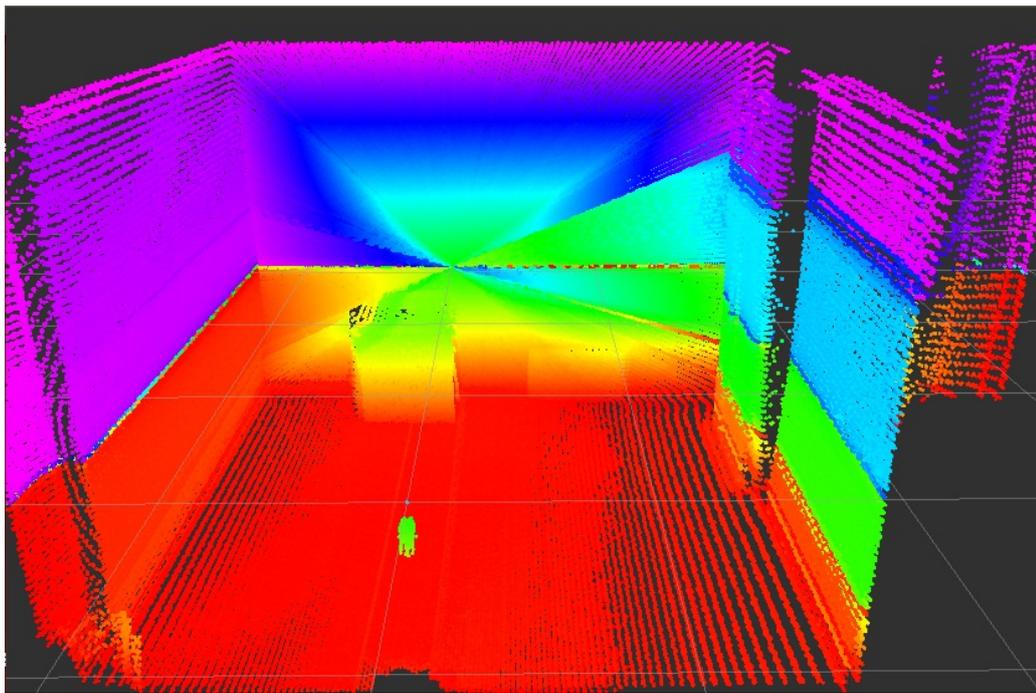
En el lanzador únicamente se ha cambiado la limitación de altura máxima (para aislar el techo), la altura a la que se encuentra el sensor (con el objetivo de aislar el suelo en los mapas 2D) y se ha limitado la apertura angular del sensor a 180°, de manera que no escanee lo que se encuentre detrás de él.

Se ha preparado especialmente la instalación para que en los mapas aparezcan paredes, obstáculos de tamaño diferente e incluso una zona acristalada para ver qué efectos provoca. Se tiene por tanto el siguiente entorno:

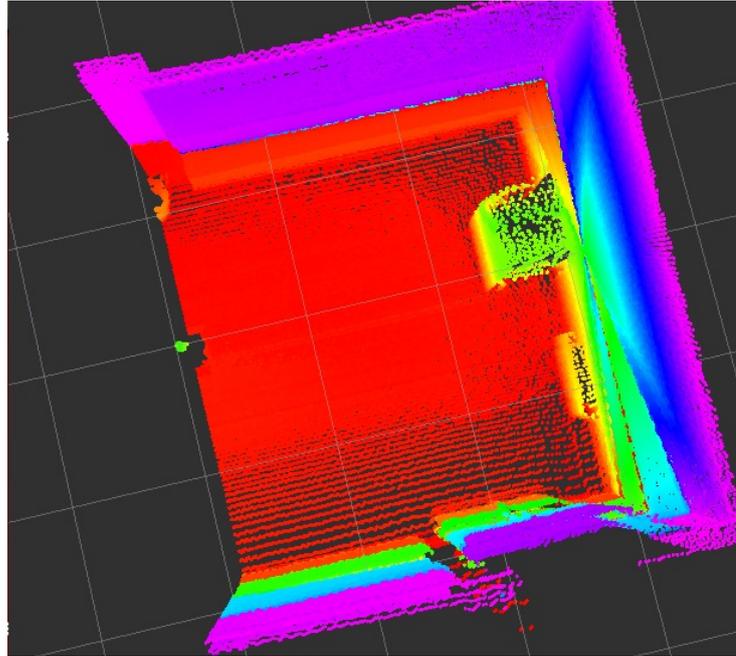


*Figura 6-1: Fotográfica de la parte de la habitación a escanear con el sistema sensorial*

Con los cambios en el lanzador descritos anteriormente, se ejecuta el sistema y se obtiene la siguiente información.



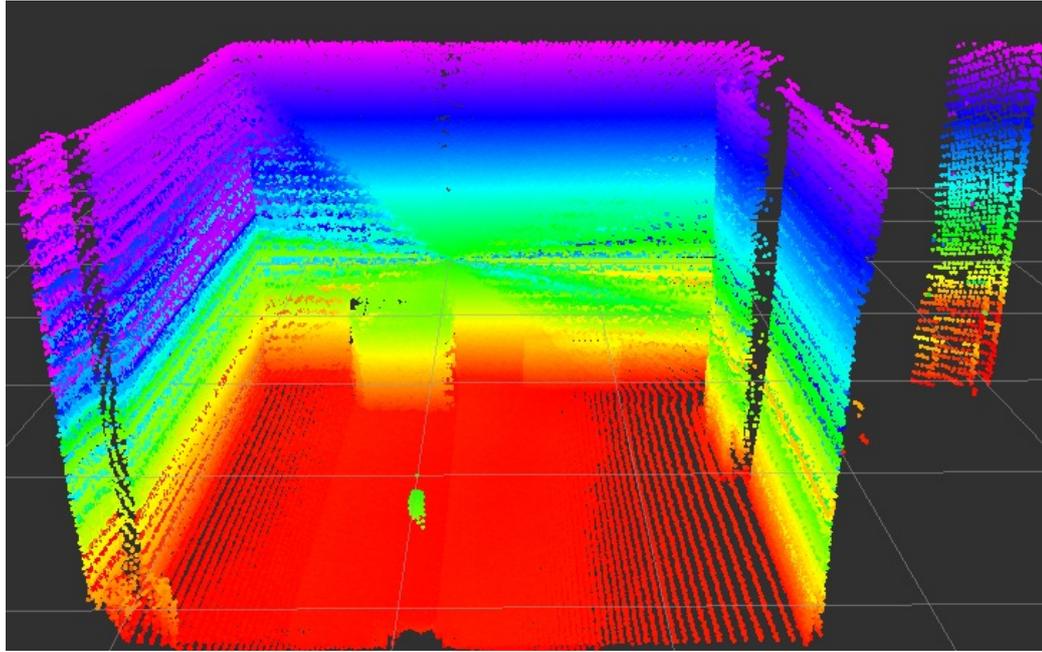
*Figura 6-2: Escaneado de la habitación sin filtrado (representado con el visualizador Rviz)*



**Figura 6-3: Escaneado de la habitación sin filtrado (representado con el visualizador Rviz)**

[1412000562.920690693]:	points before 14420	, points after 5757
[1412000562.964279187]:	points before 14420	, points after 5539
[1412000563.000932623]:	points before 14420	, points after 5295
[1412000563.040848166]:	points before 14420	, points after 5261
[1412000563.080296973]:	points before 14420	, points after 5310
[1412000563.083822737]:	points before 14420	, points after 5335
[1412000563.120220303]:	points before 14420	, points after 5373
[1412000563.160727008]:	points before 14420	, points after 5402
[1412000563.200301016]:	points before 14420	, points after 5431
[1412000563.207830905]:	points before 14420	, points after 5428
[1412000563.245193643]:	points before 14420	, points after 5433
[1412000563.285856669]:	points before 14420	, points after 5437
[1412000563.325354856]:	points before 14420	, points after 5448
[1412000563.332731101]:	points before 14420	, points after 5440
[1412000563.372671788]:	points before 14420	, points after 5422
[1412000563.413185888]:	points before 14420	, points after 5375
[1412000563.457389181]:	points before 14420	, points after 5360
[1412000563.461925928]:	points before 14420	, points after 5342
[1412000563.500968499]:	points before 14420	, points after 5323
[1412000563.542899179]:	points before 14420	, points after 5263
[1412000563.581420296]:	points before 14420	, points after 5215
[1412000563.584801936]:	points before 14420	, points after 5181
[1412000563.623830263]:	points before 14420	, points after 5119
[1412000563.664284814]:	points before 14420	, points after 5033
[1412000563.704244601]:	points before 14420	, points after 4954
[1412000563.743942298]:	points before 14420	, points after 4949
[1412000563.784317809]:	points before 14420	, points after 4927
[1412000563.823849393]:	points before 14420	, points after 4909
[1412000563.828905977]:	points before 14420	, points after 4888
[1412000563.868762199]:	points before 14420	, points after 4870

**Figura 6-4: Cantidad de datos anteriores y posteriores al filtrado realizado**



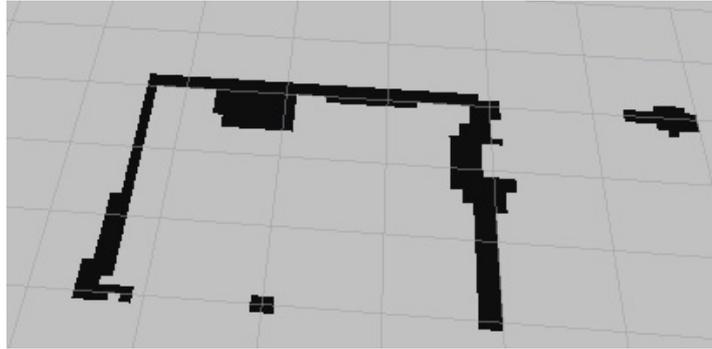
***Figura 6-5: Escaneado láser de la habitación con filtrado***

Tal y como se puede ver, se ha reducido el número de puntos a menos de la mitad y aún así se obtiene suficiente información para construir un mapa 3D con todo detalle. Por tanto, este filtrado resulta ser muy efectivo y útil para reducir la carga de datos que debe manejar el sistema.

A continuación se va a analizar todos los datos adquiridos en las gráficas anteriores. Se pueden ver perfectamente las dos cajas en frente del sensor, una esquina en la parte izquierda y el panel que aparece en la fotografía anterior en la parte derecha. Los datos que se ven más allá de la habitación son debidos a que esta tiene una parte acristalada parcialmente cubierta por el panel. Se puede ver como el cristal no es detectado sino que el haz láser lo atraviesa y coge datos de parte del pasillo.

Por último, se puede ver que se ha detectado algo más en la zona en la que se encuentra el sensor. Esto es debido a dos razones que son aplicables a cada uno de los mapas 3D que se expongan en este proyecto. La primera que cualquier objeto sobre el que esté el sistema sensorial va a ser captado por éste, y por tanto detectado. En este caso, el sistema sensorial está detectando la propia mesa sobre la que está instalado. En segundo lugar, se ha impuesto que el láser aparezca en cada mapa 3D representado como un punto, de manera que sea muy sencillo a partir de los mapas saber dónde estaba el sensor colocado. Dicho punto se puede ver justo encima de la porción de mesa detectada.

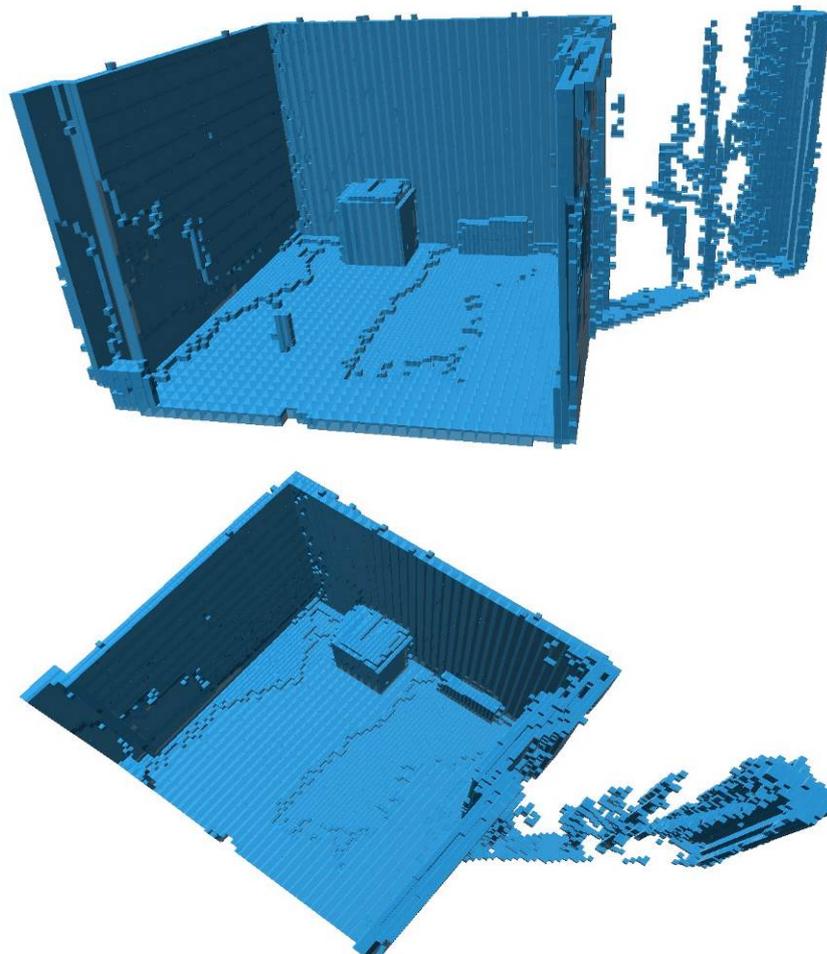
Una vez explicada cada región del mapa 3D se pasa a exponer el mapa 2D global creado, en el que debe de ignorarse el suelo de la habitación y deben aparecer las paredes, las cajas, el panel y parte del pasillo como obstáculo en color negro.



*Figura 6-6: Mapa 2D global generado de la habitación*

Efectivamente, se pueden ver cada uno de los obstáculos mencionados anteriormente de color negro.

A continuación se muestra el mapa 3D generado por el Octovis. Se recuerda este programa transforma los puntos en un modelo de cajas de dimensión 5 x 5 centímetros cada una de ellas. Dichos puntos son proporcionados al octovis gracias al guardado de los datos procesados en un fichero (parámetro configurable en algunos nodos ROS). En este caso, se parte de los datos ya filtrados por el nodo `pcl_filter_ros`.



*Figura 6-7: Mapa 3D de la habitación generado con el visualizador Octovis*

Sabiendo las medidas reales de la parte de la habitación escaneada, se procede a comparar las medidas obtenidas por el sistema sensorial con las reales, con el fin de poder saber qué error se está cometiendo y si éste es superior al error que posee el propio láser.

La longitud de la pared izquierda (en la figura 6-5) es de 3,2 metros. Analizando los datos ofrecidos por el sistema se comprueba que dicha distancia es de 3,3 metros.

En anchura, la habitación mide 3,4 metros. Los datos obtenidos muestran que la distancia medida es de 3,47 metros.

La distancia entre la pared de la izquierda y la esquina más alejada a esta de la caja cuadrada es de unos 1,405 metros, mientras que los datos obtenidos miden 1,5.

Se puede decir que el error en las medias es de 10 centímetros aproximadamente.

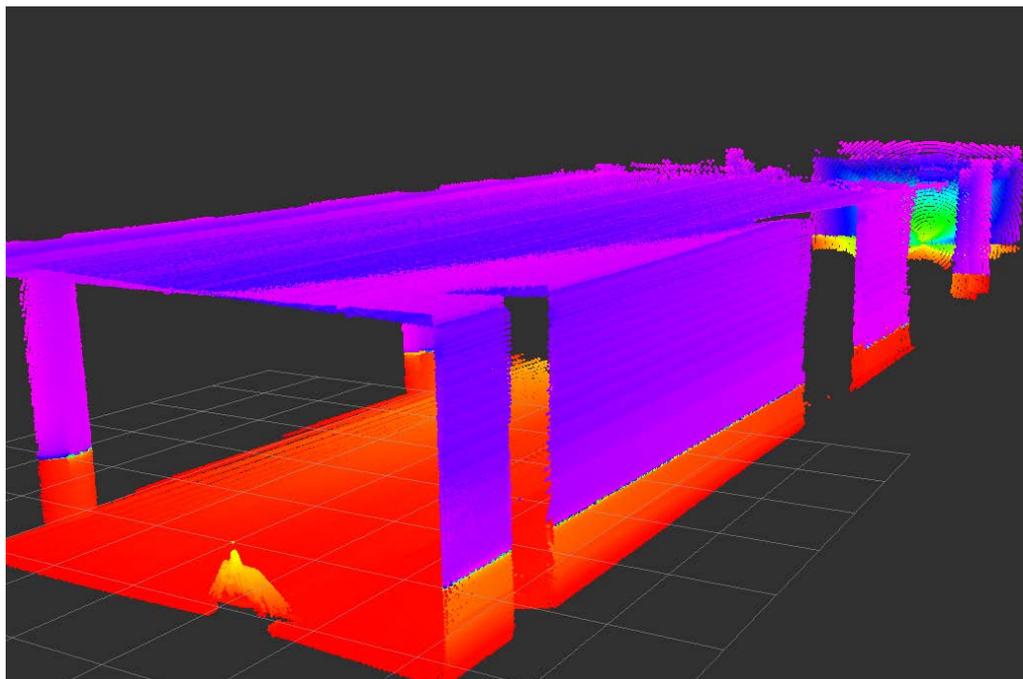
## 6.2 Prueba estática largo alcance

Esta prueba se ha realizado en un garaje en el que de nuevo todos los obstáculos y el sistema sensorial son estáticos. La diferencia con la anterior prueba es que este entorno posee mayor complejidad y mucho mayor alcance. De nuevo se situarán las dos cajas en el escenario y se comprobará si son detectadas. Esta vez, la distancia de separación es mucho mayor, de manera que se desea verificar si el sistema es muy vulnerable al alcance o si por el contrario, detecta ambas cajas pegadas a una pared sin problemas.



*Figura 6-8: Fotografía escena a analizar*

En el lanzador únicamente se ha modificado el filtrado de rango, de modo que el sensor no ofrezca información más allá de las columnas de la izquierda, debido a que dicha zona no constituye un entorno controlado. A parte de esto, de nuevo se ha configurado la altura del sensor, midiéndola con respecto al suelo.



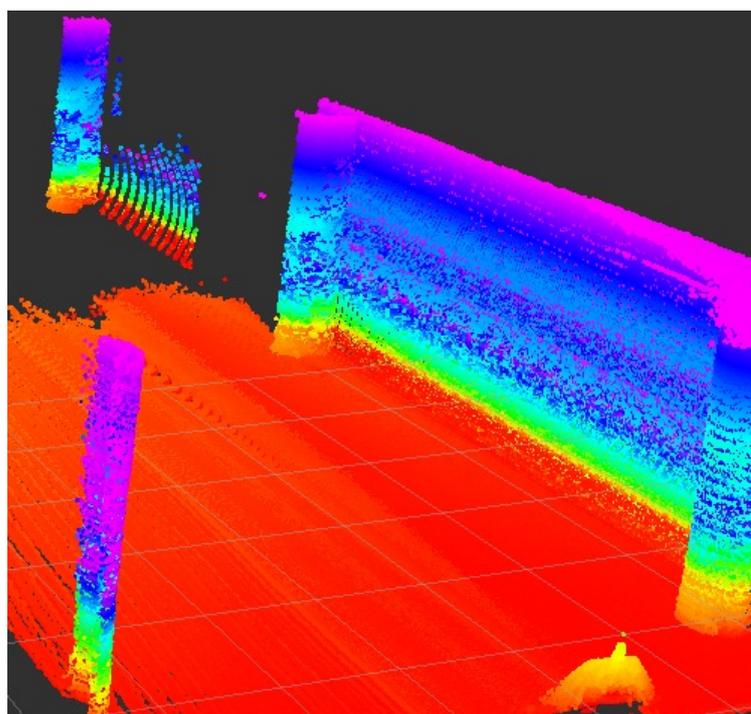
*Figura 6-9: Mapa 3D generado sin filtrado*

```

[1412176744.977162204]: points before 14420 , points after 11074
[1412176744.982098194]: points before 14420 , points after 10831
[1412176745.011777240]: points before 14420 , points after 10653
[1412176745.052062247]: points before 14420 , points after 10203
[1412176745.095700259]: points before 14420 , points after 10013
[1412176745.100025543]: points before 14420 , points after 9765
[1412176745.139431626]: points before 14420 , points after 9546

```

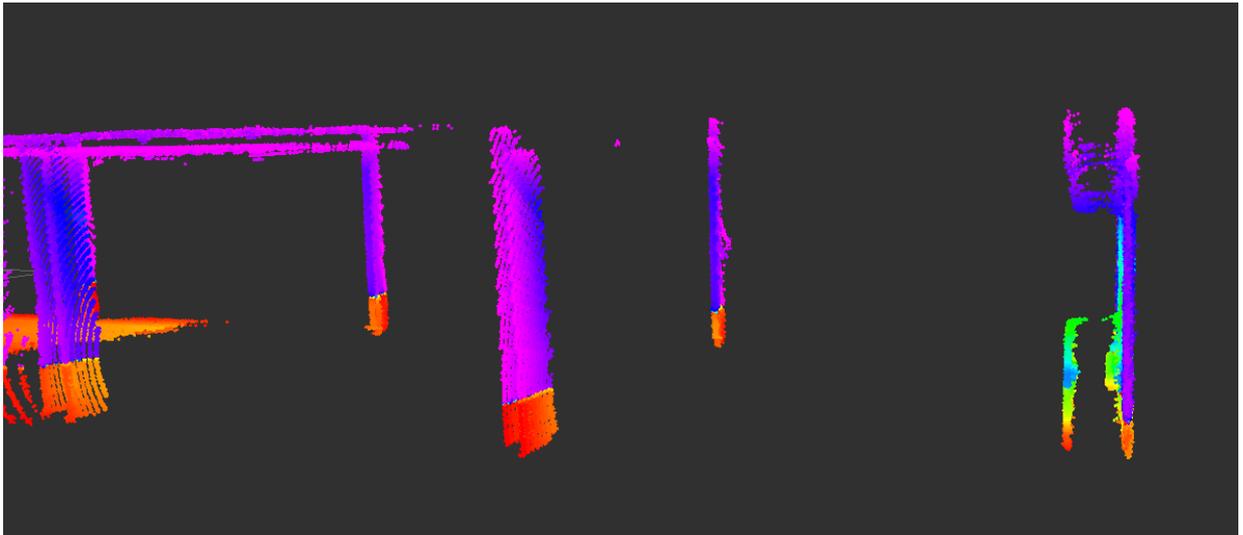
*Figura 6-10: Cantidad de datos anteriores y posteriores al filtrado realizado*



*Figura 6-11: Mapa 3D generado con filtrado*

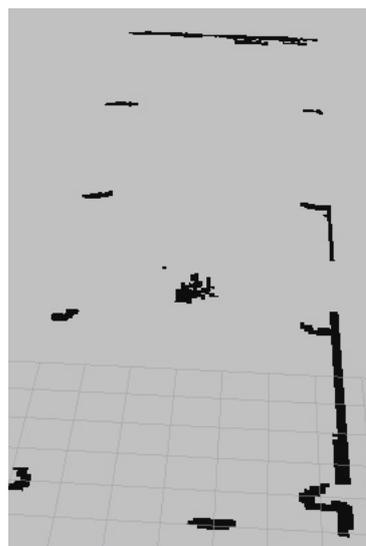
De nuevo se puede observar que el filtrado reduce notablemente la cantidad de datos del sistema, aunque en mucha menor medida que la prueba anterior. Esto es debido a que al ser la región mucho más grande que la prueba anterior, las medidas están más dispersas y el filtrado no es tan efectivo. Aún así reduce considerablemente las medidas a analizar sin perder información trascendental.

Se pueden ver perfectamente cada una de las columnas y el lugar donde se encuentra el láser en cada imagen anterior, debido a que, como ya se ha mencionado, se capta la plataforma en la que está instalada y además se marca con un punto la posición de dicho sistema sensorial.



*Figura 6-12: Vista lateral del fondo del garaje*

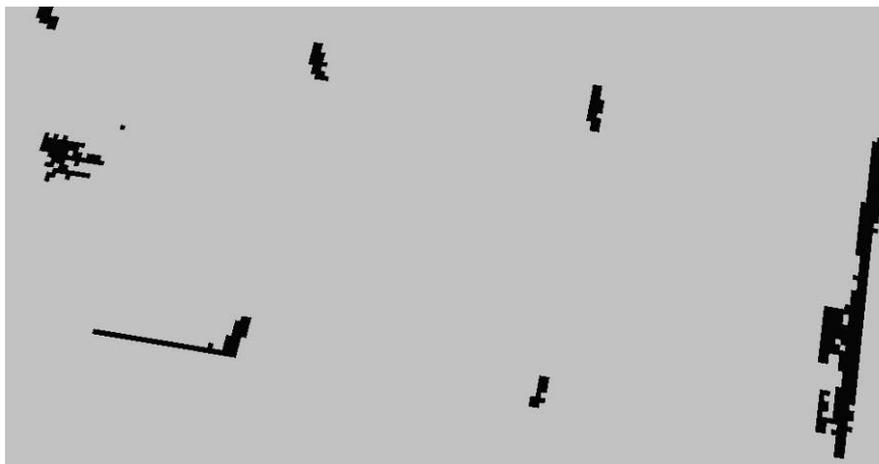
En esta figura se pueden ver las cajas de perfil. La caja grande se distingue del fondo sin ningún problema. En cambio, la caja pequeña apenas sobresale de la pared, sino que más bien parece una pequeña ondulación de la pared del fondo. En la parte superior se puede ver otro saliente de la pared. Dichas medidas se corresponden con las típicas cañerías ancladas al techo, muy comunes en los garajes.



*Figura 6-13: Mapa 2D global generado*

En la figura del mapa 2D global se pueden ver las columnas, las paredes laterales, el lugar donde está colocada la mesa que sustenta el sistema sensorial, la pared frontal en donde se encuentran los obstáculos y una pequeña región situada en medio de la imagen que se ha detectado como obstáculo donde en realidad no lo hay. Esto es un error que puede ser debido a una pequeña ondulación del garaje que el software no ha sido capaz de aislar.

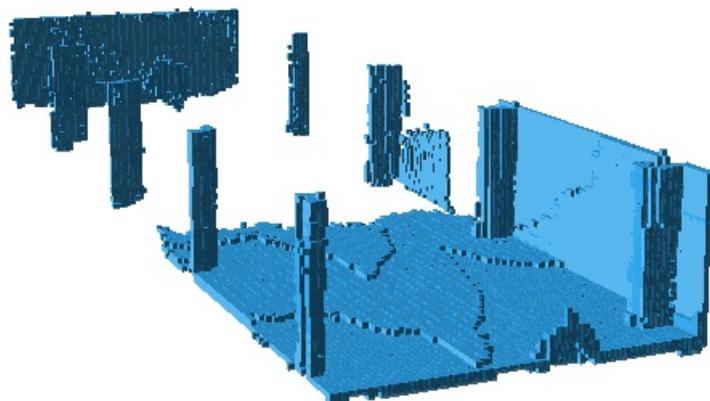
A continuación se va a ampliar la región del mapa 2D global con más interés, la pared frontal más alejada.



*Figura 6-14: Mapa 2D global generado (vista lateral)*

Se puede ver que la caja grande situada en dicha pared es detectada casi completamente, en cambio, la caja pequeña casi se ha fusionado con la pared y apenas es apreciable. Esto puede ser debido a que el grosor de la caja pequeña es de 15 centímetros aproximadamente mientras que la distancia a la que se encuentra dicha caja del sistema sensorial es de unos 25 metros aproximadamente.

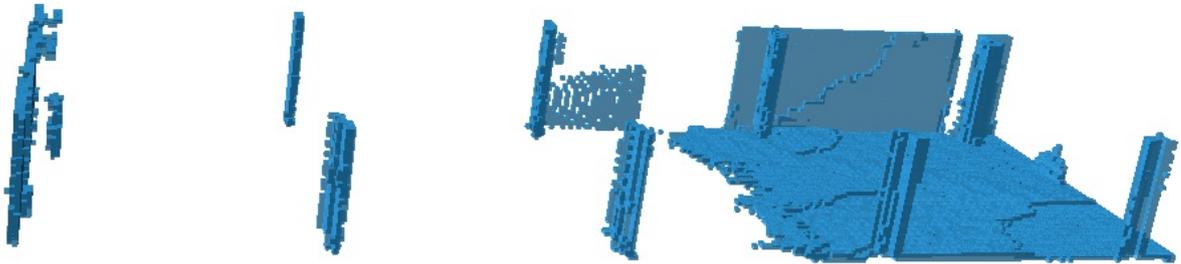
De nuevo se vuelve a representar el mapa 3D generado por el Octovis gracias al fichero “scan\_after\_filter.log” creado por el nodo pcl\_filter\_ros.



*Figura 6-15: Mapa 3D generado a partir del visualizador Octovis*

En esta figura se puede ver que en la región donde se produce el error en el mapa 2D está ligeramente elevado respecto al nivel normal del suelo.

Concretamente, como cada cuadrado tiene 5 centímetros de lado, la región donde se produce el error está levantada 10 centímetros con respecto al nivel del suelo.



*Figura 6-16: Mapa 3D generado a partir del visualizador Octovis (vista lateral)*

Las cajas siguen siendo detectadas con esta visualización, aunque la caja pequeña apenas se distingue del fondo, como en los casos anteriores.

De nuevo, se han obtenido medidas reales para poder compararlas con las del sistema y conocer cuánto error se está cometiendo. De nuevo se van a exponer varios ejemplos de medidas de las que se sacará una conclusión con el error.

La distancia entre el sensor y la pared frontal es, exactamente, de 24,86 metros, mientras que el sistema establece dicha distancia en 25 metros. Es decir, 14 cm de error, casi igual al grosor de la caja pequeña pero que aun así es detectada.

La anchura de la escena, desde la pared derecha hasta las columnas de la izquierda es de 6,72 metros, mientras que los datos del sistema ofrecen una distancia de 6,7 metros.

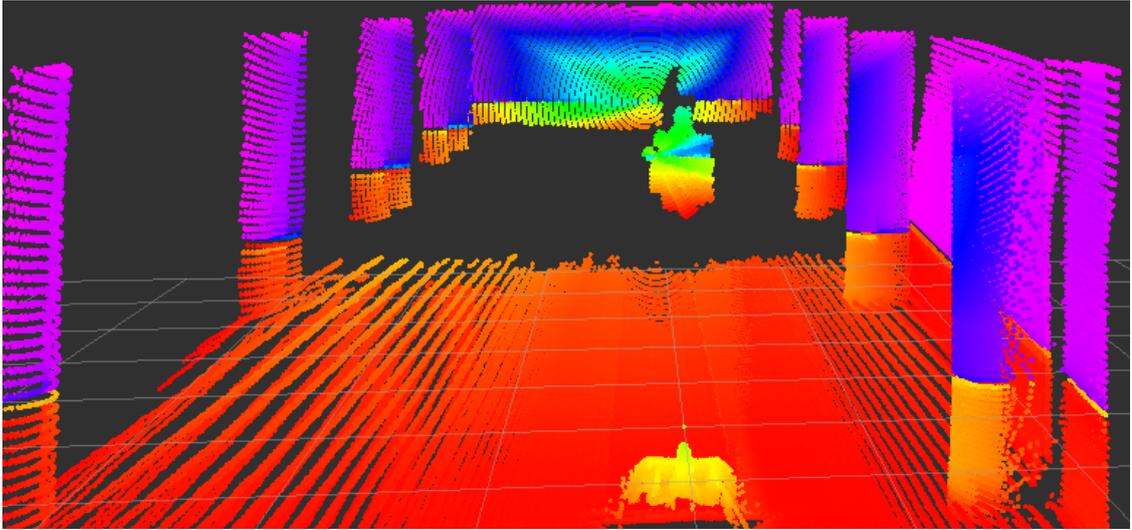
Por tanto, en esta ocasión se obtiene un error de 14 centímetros a 25 metros de distancia. Como se puede ver, el error apenas se ha incrementado con respecto a la prueba de la habitación mientras que las distancias de medición se han multiplicado.

A continuación se ha cambiado la colocación de las cajas a la que se muestra en la siguiente fotografía, de modo que se pueda ver si el canto de la caja pequeña es detectado:

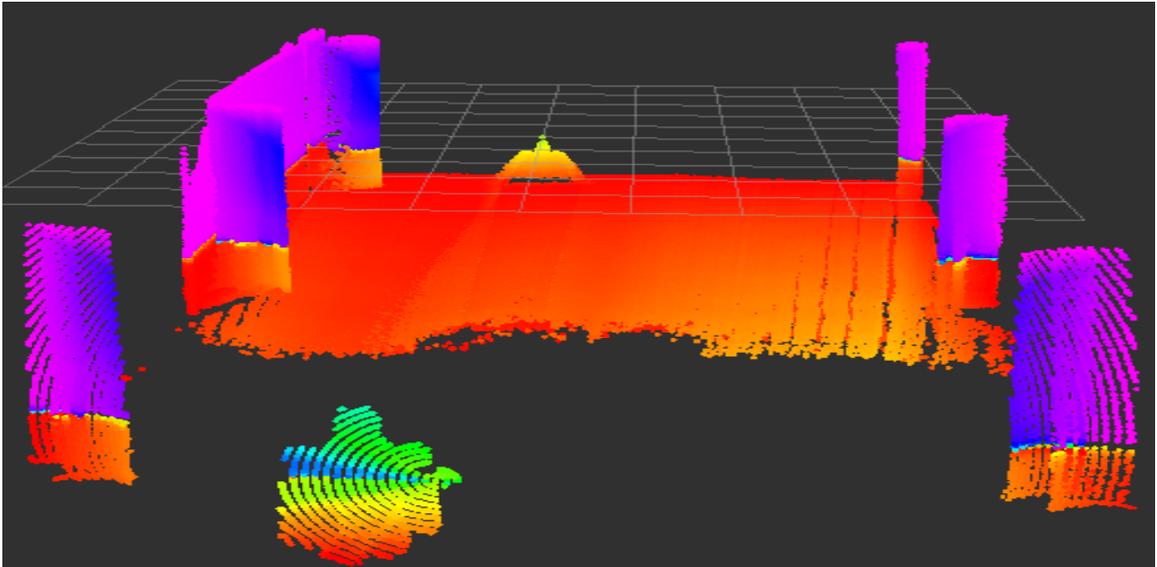


*Figura 6-17: Fotografía de la colocación de las cajas*

Bajo esta nueva configuración se muestra el mapa 3D generado por el sistema sensorial:



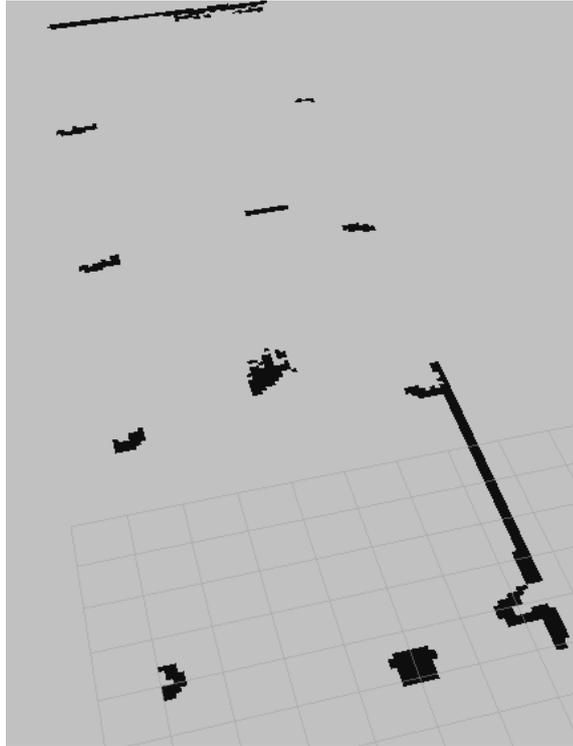
*Figura 6-18: Mapa 3D filtrado del entorno*



*Figura 6-19: Mapa 3D filtrado del entorno*

En esta prueba se puede ver que el canto de la caja pequeña es casi perfectamente detectado, gracias a que se encuentra situado a unos 15 metros en frente del sensor.

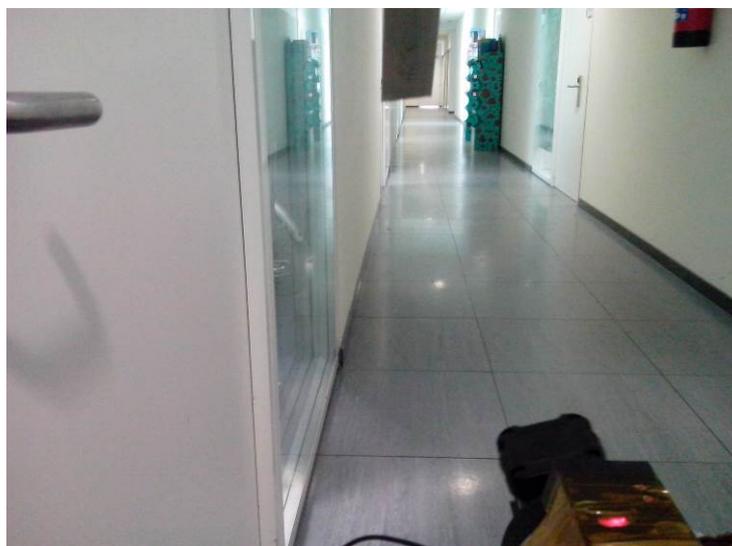
Por último, el mapa 2D global generado se presenta a continuación, en el que se pueden ver todos los obstáculos anteriores que han permanecido inmóviles, la región errónea detectada como obstáculo y una línea que representa el obstáculo de las cajas.



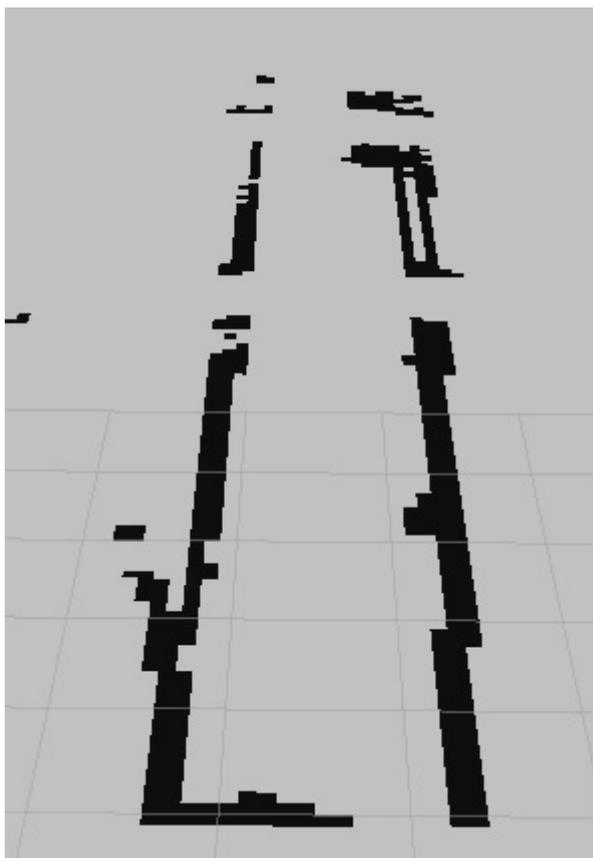
*Figura 6-20: Mapa 2D generado del entorno analizado*

### **6.3 Prueba dinámica en el pasillo con obstáculos estáticos**

Esta prueba se ha realizado en un determinado pasillo del Parque Científico. El objetivo de esta prueba es mover el sistema sensorial a lo largo del pasillo, de manera que el mapa 2D global generado se vaya rellenando con los datos suministrados por el sistema sensorial, aplicándoles a todos ellos una determinada traslación del sistema, suministrada desde el nodo prueba. Por tanto, se ha tenido que estimar la velocidad de traslación del conjunto debido a la imposibilidad de conocer la velocidad instantánea por el hardware y el software desarrollado. Como el mapa 3D generado es local, no se va a mostrar en dicha prueba debido a que, al ser local, sólo muestra lo que haya en frente del sistema sensorial en cada instante de tiempo. Dicho esto, a continuación se presenta una fotografía y el mapa 2D global obtenido.



*Figura 6-21: Fotografía del pasillo*



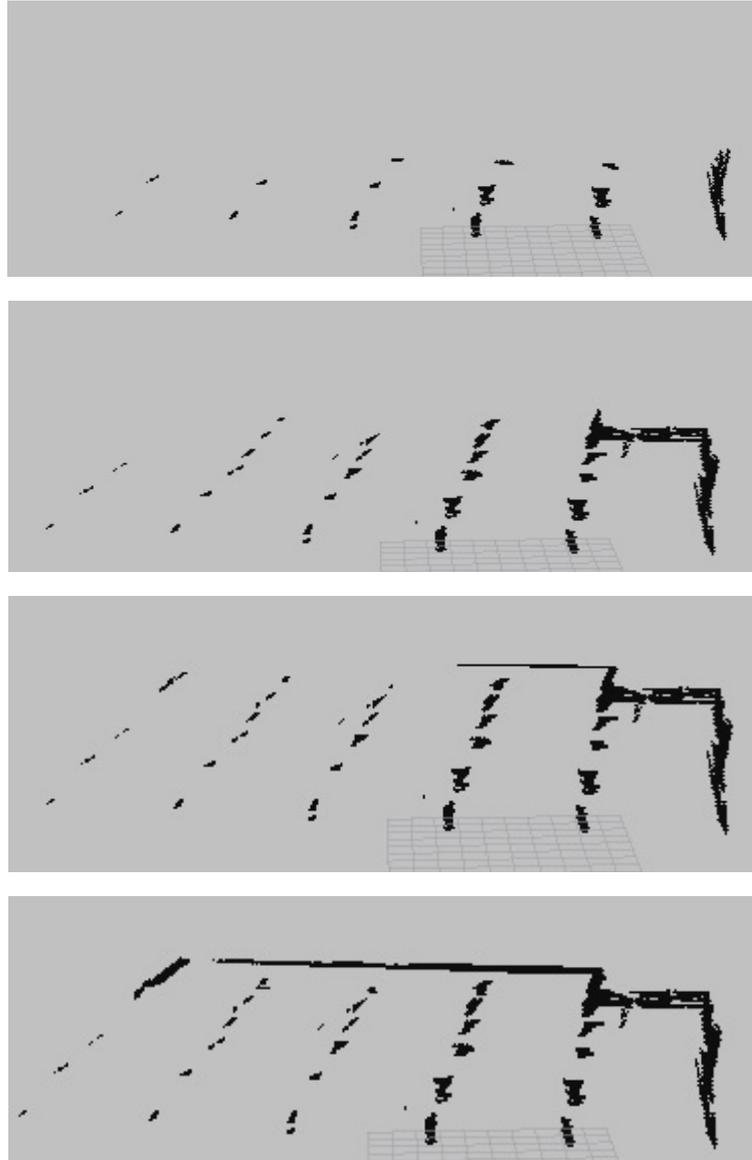
*Figura 6-22: Mapa 2D generado del pasillo*

Al estar en movimiento, se ha eliminado el punto en el que estaba localizado el láser y se ha filtrado el obstáculo producido por la mesa que sustenta dicho sistema sensorial. Además se ha reducido la apertura angular de modo que quedase una pequeña región detrás del sensor en la que no se adquiere datos. Esto es debido a que no se quiere captar a la persona que está empujando la mesa por el pasillo, ya que constituiría un obstáculo en todo el recorrido. La caja verde de la fotografía es fácilmente detectable en el mapa 2D expuesto, pero se puede observar que la pared va sufriendo ondulaciones, debidas a que es difícil empujar el sistema por el pasillo sin que la distancia de separación del sistema sensorial con las paredes laterales varíe, ya que existe inestabilidad del sistema motor que desplaza dicho sistema.

Además, se puede ver que las paredes laterales no son continuas. Esto es debido a que de nuevo hay zonas acristaladas en las que el sistema sensorial no consigue detectarlas.

#### **6.4 Prueba dinámica en el garaje con obstáculos estáticos**

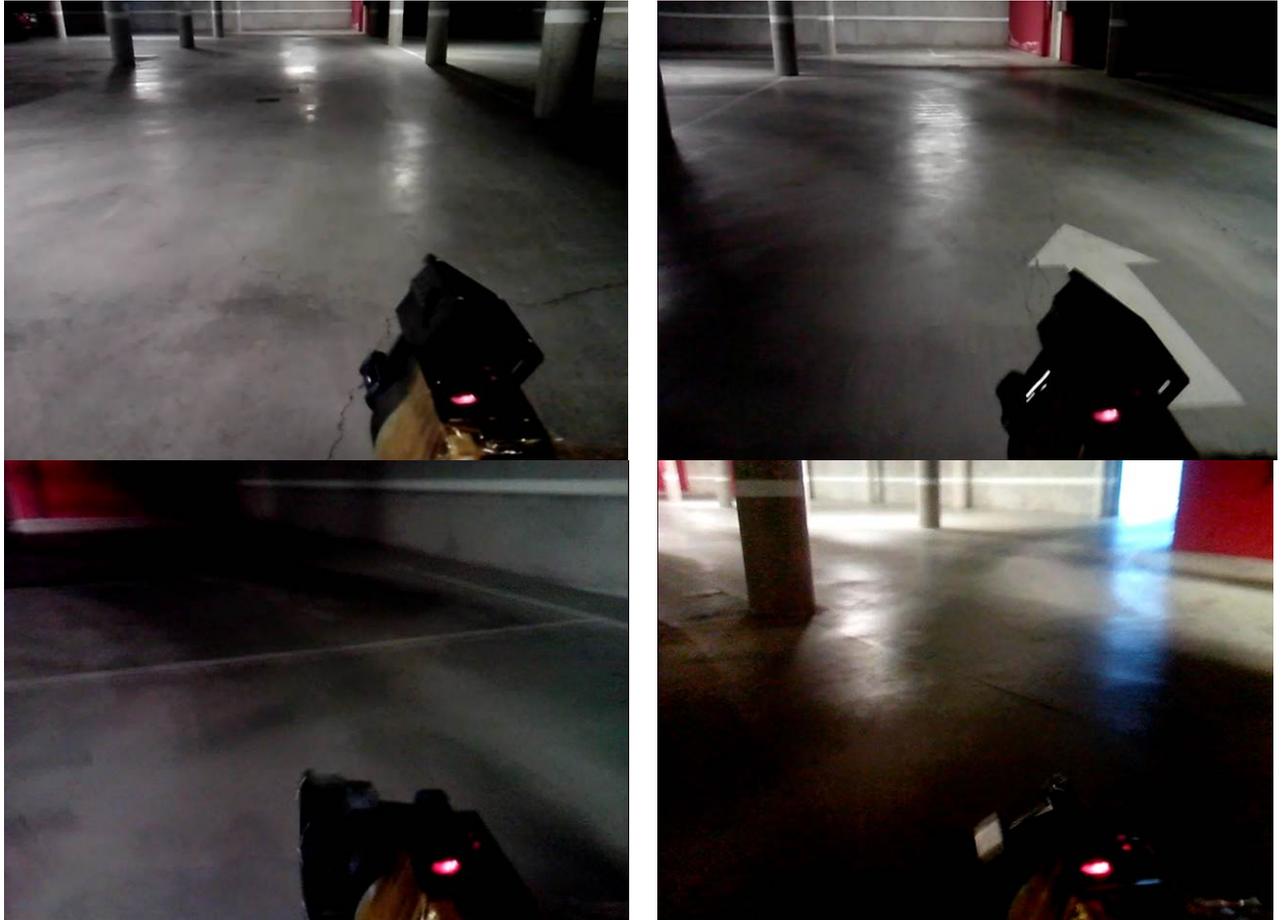
A continuación se aumenta la complejidad del entorno y se plantea de nuevo el entorno del garaje pero esta vez, con el sistema sensorial en movimiento (de traslación). De nuevo, es difícil estimar la velocidad de movimiento, por lo que las medidas aparecerán duplicadas. Aún así, se quiere ver cómo se debe ir completando el mapa 2D global. A continuación se puede ver las continuas actualizaciones que ha sufrido dicho mapa.



*Figura 6-23: Mapa 2D que se va completando según la traslación del sistema sensorial*

Tanto las columnas como las paredes sufren distorsión debido de nuevo al desconocimiento de la velocidad de traslación con la que se está moviendo el sistema sensorial, de modo que es imposible que ambos obstáculos queden perfectamente definidos. Aún así se puede ver como poco a poco se va completando el mapa 2D, objetivo de esta prueba, por lo que el sistema sensorial es capaz de generar un mapa 2D tan grande y completo como se quiera, dentro de los límites de capacidad de memoria disponible.

A partir de cómo se va rellenando el mapa 2D global, se puede predecir fácilmente la trayectoria que se ha seguido de manera razonable. No obstante, a continuación se adjuntan algunas fotografías del recorrido realizado.



*Figura 6-24: Fotografías del recorrido realizado*

### **6.5 Prueba estática con obstáculos dinámicos**

A partir de esta prueba se procede a la completa instalación del sistema, es decir, se monta el sistema sensorial en el vehículo, como se puede ver a continuación.

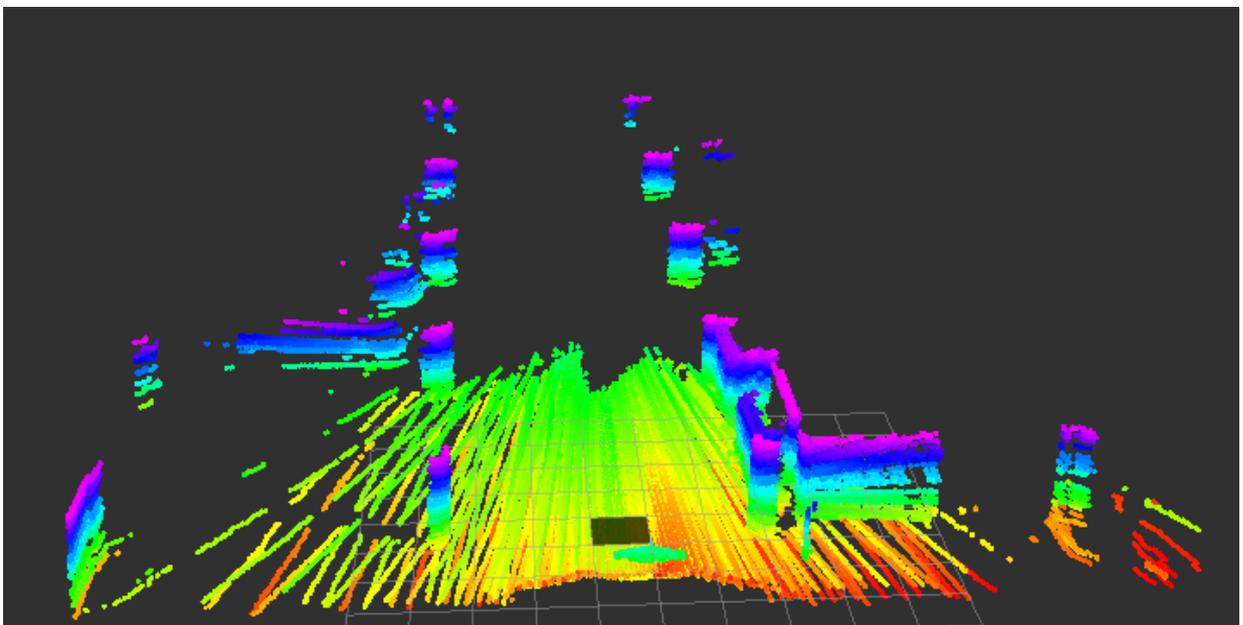


*Figura 6-25: Vehículo Yamaha Rhino equipado con el sistema sensorial*

En esta prueba en la que el vehículo permanece estático en todo momento, se quiere observar la rapidez del sistema para detectar obstáculos dinámicos. Como ya se ha mencionado en previas explicaciones, si lo que se requiere es conocer todas las regiones alrededor del vehículo se procederá a girar el sistema sensorial a una revolución por segundo elevada, pero siempre dentro del límite impuesto anteriormente. Por ello, en esta prueba no es necesario conocer con mucho detalle cada zona de la periferia del vehículo, sino captar lo antes posible los obstáculos dinámicos. Para ello, esta prueba plantea un garaje ligeramente distinto al anterior en el que una persona cruza de lado a lado a una determinada distancia del vehículo. El objetivo es que dicha persona sea detectada cuanto antes, y ver con qué precisión es detectada.

Se han configurado los parámetros del lanzador para esta prueba concreta, tales como la imposición de una velocidad de giro a 1.5 RPS, la altura del sensor respecto del suelo, y la limitación en altura para que no aparezca el techo.

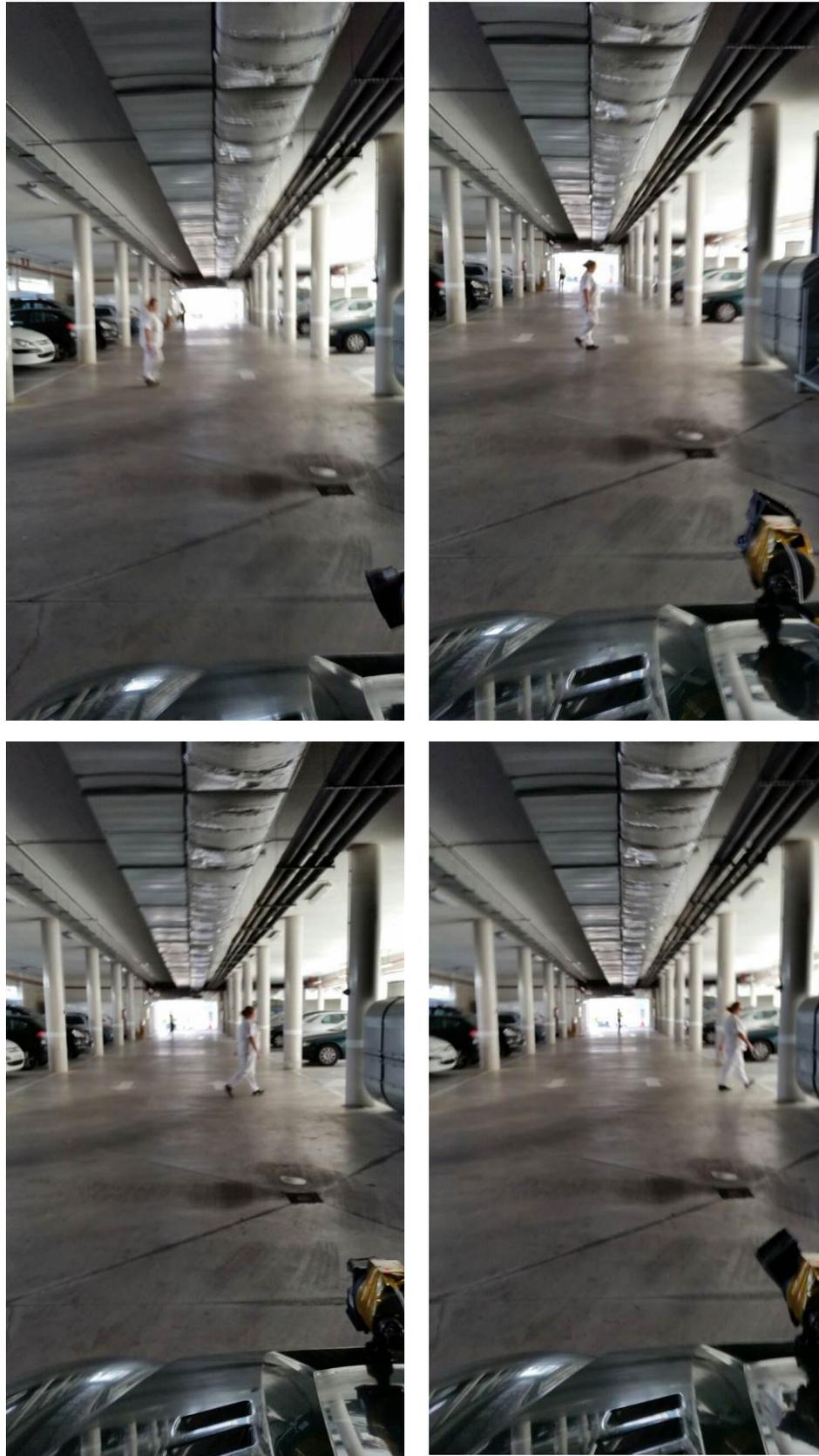
Antes de mostrar el resultado de la prueba, es necesario explicar la manera en la que se van a representar dichos resultados. Como es un obstáculo dinámico, el mapa 3D local detectará a la persona en una región distinta en cada instante de tiempo. Por tanto, la cantidad de figuras de representación del mapa 3D que habría que poner en este caso son elevadas y se ha optado por una solución más breve. El visualizador Rviz posee una herramienta llamada “decay time” que permite que las medidas se almacenen por un periodo de tiempo configurable, de modo que hace posible ver muchos barridos escáneres a la vez correspondientes a tiempos diferentes. Este parámetro no implica perder el tiempo real del sistema, pero sí la tardía desaparición de los datos adquiridos del visualizador. Teniendo en cuenta esto y sabiendo que una persona va a cruzar delante del vehículo, uno espera ver el recorrido que ha seguido dicha persona, como si de muchas de ellas se tratasen.



*Figura 6-26: Mapa 3D generado, sin obstáculo dinámico*

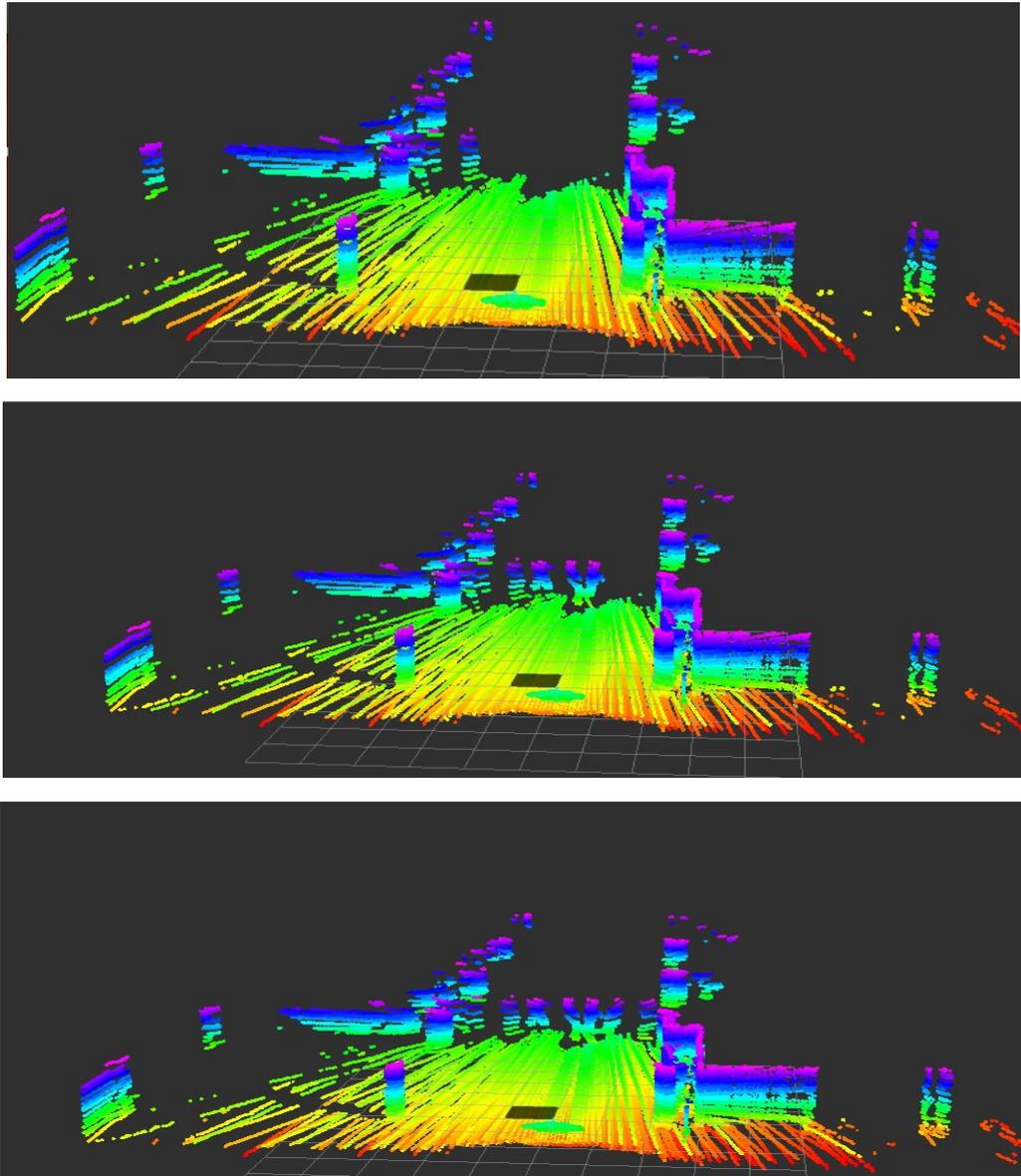
Como hasta ahora, se presenta el escenario a partir del mapa 3D generado. En esta figura se pueden ver columnas, paredes, uno o varios vehículos estacionados a la izquierda y parte del frontal del vehículo en el que está instalado en el sistema sensorial.

A continuación, dicha persona comienza a moverse de izquierda a derecha, tal y como se puede ver en la siguiente figura:



*Figura 6-27: Fotografías del movimiento realizado por la persona*

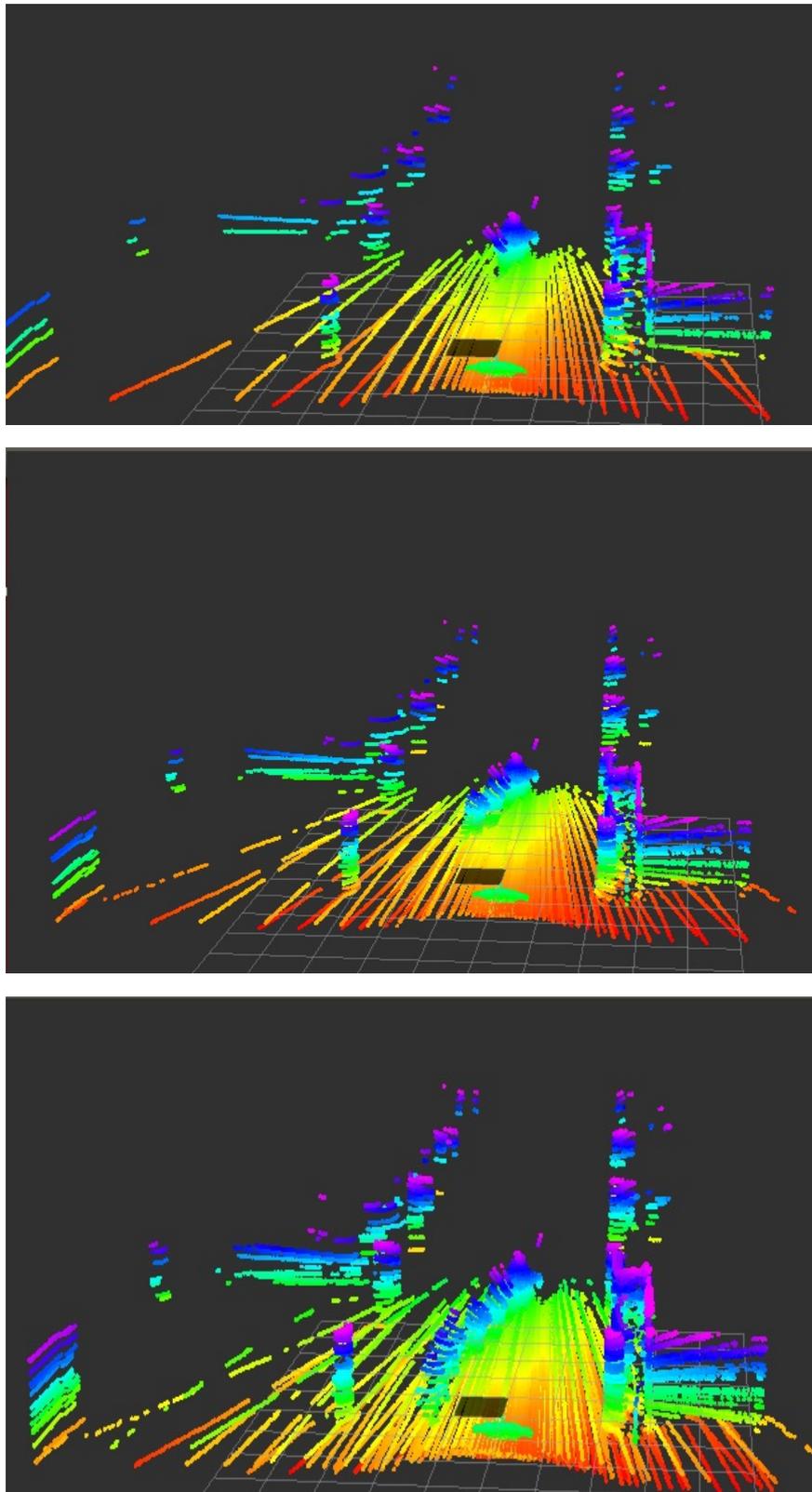
A continuación se procede a la exposición del mapa 3D generado. Para facilitar la visualización, se activa la herramienta “decay time” del simulador.



*Figura 6-28: Detección del obstáculo dinámico*

Gracias a dicha herramienta, se puede ver el recorrido que ha seguido dicha persona, y la diferente toma de medidas captadas por el sensor. Como se puede ver, se ha captado de manera aceptable la persona en cada instante de tiempo, quedando más definida cuanto más en frente está situada del sistema sensorial. Si se llega a girar más rápido el sistema sensorial, se tendría menos detalle del entorno pero en cambio este recorrido de la persona aparecería casi continuo, gracias a la rapidez de la toma de datos en las diferentes regiones.

Como representación adicional, se presenta otro caso de obstáculo dinámico generado por una persona en las cercanías del vehículo.

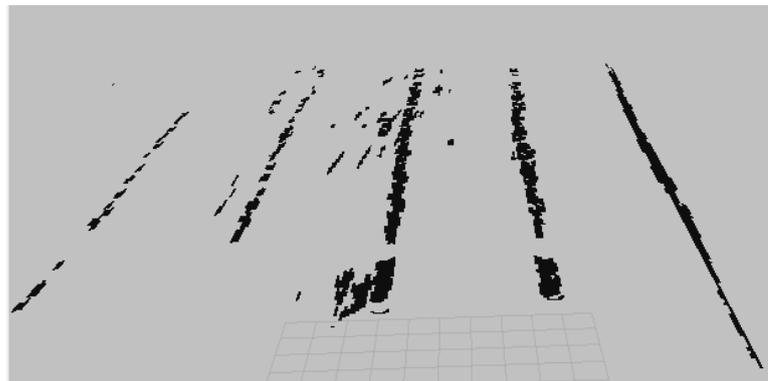
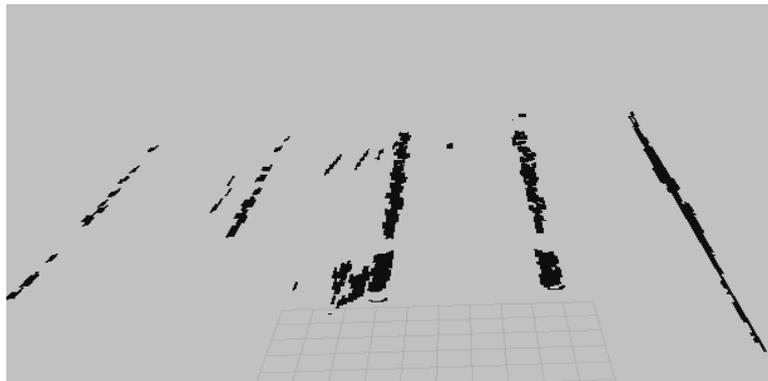
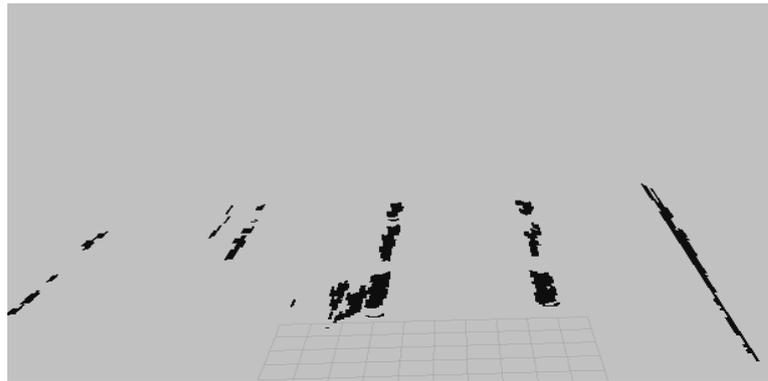


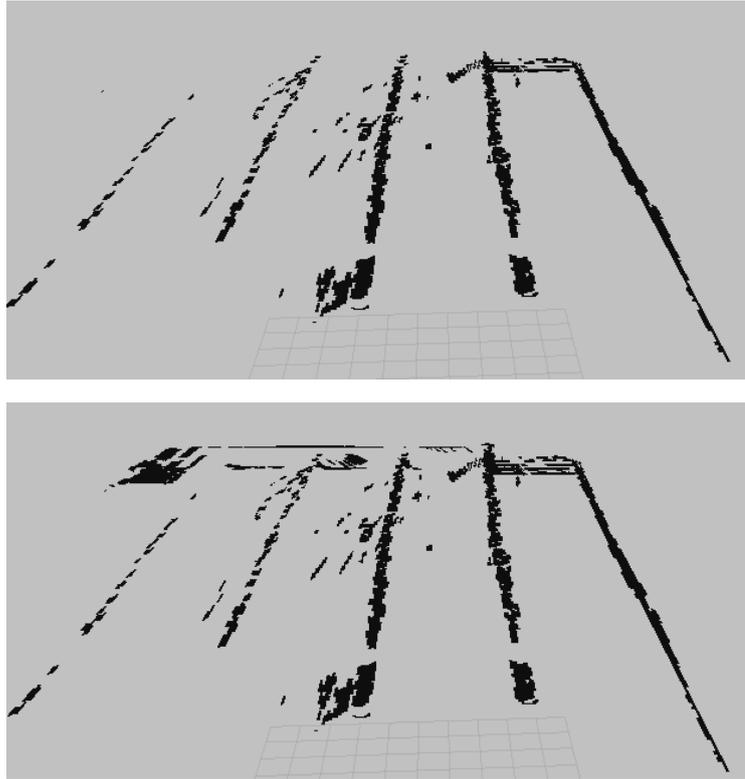
*Figura 6-29: Detección del obstáculo dinámico inesperado (con decay time)*

De nuevo se puede ver la trayectoria seguida por la persona, pero esta vez con menos definición y de manera más continua que en el caso anterior.

## 6.6 Prueba dinámica sistema completo

Una vez montado el sistema sensorial en el vehículo, se procede a realizar esta prueba en la que el vehículo se recorre parte del garaje con el objetivo de que se vaya rellenando el mapa 2D global que se genera. De nuevo hay que mencionar que los nodos ROS desarrollados no disponen de la velocidad a la que el vehículo se mueve, por tanto es necesario estimarla. Como ya se ha mencionado, adaptar la velocidad del vehículo a la estimación realizada (o viceversa) es realmente complejo. Por ello, el efecto que se obtiene al no poder tener la velocidad real del vehículo es un estiramiento de los datos obtenidos. La orientación del vehículo no supone ningún problema gracias a los datos recogidos por la IMU. Sin embargo, la vibración del vehículo hace que las medidas sufran una pequeña distorsión.





*Figura 6-30: Mapa 2D generado y actualizado según la posición del sistema sensorial*

Comparando estos resultados con los de la prueba 6.4 se pueden observar diferencias notables. En primer lugar, en esta prueba las paredes aparecen más rectas que en dicha prueba anterior, ajustándose más a la realidad. No obstante, las columnas aparecen muchas más veces de las que deberían al no ajustarse la velocidad de traslación estimada con la real. Además se puede observar que en esta prueba aparecen obstáculos en forma de líneas sin ningún sentido, evidentemente se trata de errores producidos bien por algún tipo de bache o bien por otros motivos. No obstante, queda demostrado que el sistema es capaz de adquirir datos del entorno, analizarlos y generar mapas 3D y 2D a partir de ellos en tiempo real.



# 7 Conclusiones y desarrollos futuros

---

## 7.1 Conclusiones

En este proyecto se ha propuesto un sistema sensorial basado en un láser 2D y una plataforma de giro, especialmente diseñado para la navegación autónoma de vehículos terrestres no tripulados. El sistema exhibe capacidades interesantes tales como la continua generación de grandes densidades de puntos en tiempo real y a alta velocidad.

Para elección de este sistema sensorial, se ha realizado un estudio del estado del arte. De toda la gama de sensores capaces de adquirir datos 3D del entorno, se ha optado por el sistema sensorial basado en láser como la mejor opción para lograr el objetivo de la navegación autónoma. A dicha conclusión se ha llegado gracias a un minucioso estudio sobre las características, ventajas y desventajas de cada tecnología orientada a este propósito.

El sistema sensorial está basado en un sistema de giro de un sensor láser 2D, capaz de generar mapas 3D y 2D del entorno en tiempo real, de manera que a partir de ellos se puedan aplicar algoritmos de navegabilidad. Antes de la construcción de dicho sistema, se ha realizado un estudio para evaluar los datos de medición resultantes, tales como estudio de la forma de giro, mínimo obstáculo detectable o densidad de puntos adquirida dependiendo de la velocidad de giro.

La velocidad de giro del sistema sensorial es adaptativa a la velocidad del vehículo, ya que no existe una velocidad de giro óptima independiente de la traslación del vehículo. Dependiendo de si se requiere una alta densidad de puntos en cada región o una rápida adquisición de datos en cada región, se elegirá una velocidad consecuente a cada objetivo. Además, se ha acotado la velocidad máxima y la velocidad mínima de giro del sistema sensorial de manera teórica, y se ha impuesto de manera complementaria una velocidad máxima práctica a partir de la cual los datos presentan más problemas de fiabilidad, ya que podrían aparecer de manera frecuente valores espurios.

De hecho, es posible detectar un obstáculo de 10 cm a 10 metros en frente del vehículo, haciendo que la máxima densidad de puntos adquirida coincida con la colocación de dicho obstáculo.

Se han desarrollado algoritmos de procesamiento de los datos crudos a partir de lecturas laser sincronizadas con un sistema motorizado de giro. Se ha realizado el filtrado y la exportación de estos datos a herramientas de visualización 3D. Además de la generación de nubes de puntos 3D en tiempo real, se ha desarrollado un sistema que genera mapas de obstáculos en 2D para su utilización en tiempo real en el vehículo.

Para la correcta evaluación del sistema sensorial, se ha elaborado un software que permite tanto simular el sistema sensorial en un entorno de simulación completo y fiable, como trabajar con el sistema real. Dicho software posee parámetros fácilmente configurables por el usuario para que el sistema de amolde perfectamente al objetivo que se quiera, tales como altura del sensor, velocidad de giro, apertura angular etc...

Una vez probado el correcto funcionamiento del sistema y del software elaborado, se han realizado diversas pruebas estáticas y dinámicas y se han mostrado los resultados obtenidos en cada caso, quedando claro que el sistema propuesto es realmente eficiente y efectivo a un coste relativamente bajo.

El sistema sensorial desarrollado tiene un alto potencial para brindar datos de calidad para la medición sin contacto de objetos tridimensionales y escenas. Si se utiliza a bordo de robots móviles o vehículos no tripulados puede brindar datos relevantes para desarrollar sistemas inteligentes de exploración autónoma en entornos desconocidos.

## **7.2 Trabajos futuros**

Una posible línea de futura es realizar un mapa 3D global, de manera que mientras el vehículo se mueve, se vaya completando el mapa con más y más información, de manera que se podría ver en 3D todo lo escaneado por el sistema sensorial y no sólo lo que se encuentra escaneando en ese instante. No obstante, dicho mapa constituye un coste computacional enorme, lo que supone la creación de una algoritmia que permita reducir aún más la cantidad de datos guardados para dicho propósito sin perder información significativa.

Otra posible línea futura es la realización de algoritmos de navegabilidad a partir del mapa 2D global generado. Pero para que dicho mapa sea fiable, es necesario tener información sobre la traslación del vehículo, de manera que se tenga en cuenta esta en la colocación de los datos escaneados por el sistema sensorial.

Además, se podría mejorar el mapa 2D global por medio de un establecimiento de un margen de seguridad en los obstáculos detectados, de modo que se tenga aún más fiabilidad y confianza en el sistema. Dicho margen podría ser de un color distinto al negro, para que se pueda ver que, aunque no constituye dicho margen un obstáculo, es recomendable que el sistema no navegue cerca de este. Adicionalmente, los datos espurios deben ser sometidos a otro tipo de filtrado adicional al elaborado en los nodos ROS.

Los datos generados por el sistema sensorial podrían ser enviados a través de internet a un equipo externo, de manera que sea en ese ordenador en el que se analicen y procesen los datos adquiridos para no saturar el ordenador de a bordo del vehículo.

Por último, se podría equipar al sistema sensorial con una cámara estereoscópica, de manera que se obtenga mayor riqueza de información. Incluso se podrían usar las imágenes captadas en cada momento por las cámaras estereoscópicas como base para representar los puntos obtenidos por el sensor láser, de manera que el usuario que controle el sistema pueda ver fácilmente a partir de las fotografías el entorno alrededor del vehículo, completado con los datos obtenidos por el sensor láser. De esta manera el sistema sería más robusto y resistente a posibles errores y fiable, ya que en caso de que uno de los sensores falle en un determinado momento, se tendrá el otro sistema como posible solución gracias a que con ambos sistemas por separado se puede lograr la navegación autónoma.



# Referencias

---

- [1] M. Montemerlo et al., “Junior: The Stanford entry in the Urban Challenge,” Journal of Field Robotics, vol. 25, no. 9, September 2008, Special Issue on DARPA Urban Challenge (Part II).
- [2] M. Montemerlo et al, “Stanley: The robot that won the DARPA Grand Challenge,” J. Field Robot., vol. 23, no. 9, pp. 661– 692, Sep. 2009
- [3] Más información acerca del tiempo de vuelo disponible en:  
<http://web.pdx.edu/~jduh/courses/geog493f12/Week04.pdf>  
[http://www.ct3.es/web/publicaciones/SNE\\_2012\\_Evolucion\\_Tecnologia\\_Laser.pdf](http://www.ct3.es/web/publicaciones/SNE_2012_Evolucion_Tecnologia_Laser.pdf)
- [4] Más información acerca de la diferencia de fase disponible en:  
<http://www.upv.es/satelite/trabajos/pracGrupo4/sistmed.htm>
- [5] Más información acerca de la triangulación disponible en:  
<http://www.infaimon.com/es/camaras-3d-basadas-en-triangulacion-laser>
- [6] Más información acerca del funcionamiento de sistemas RADAR disponible en:  
[http://www.radartutorial.eu/01\\_basics/Determinaci%C3%B3n%20de%20Distancia.es.html](http://www.radartutorial.eu/01_basics/Determinaci%C3%B3n%20de%20Distancia.es.html)  
<http://agamenon.tsc.uah.es/Asignaturas/ittst/rdet/apuntes/handout1.pdf>
- [7] Phil Corya, H. R. Everettb, Tracy Heath Pastoreb, “Radar-Based Intruder Detection for a Robotic Security System”, Robotic Systems Technology, 1234 Tech Court, Westminster, MD 21157.Space and Naval Warfare Systems Center, San Diego, CA 92152.
- [8] Daniel Vargas González, Antonio Broquetas Ibars, “Detección de movimiento mediante técnicas radar CW-FM en banda W”, Marzo 2014. Escuela politécnica de Cataluña.
- [9] Davide Brizzolara, —Future trends for automotive radars: Towards the 79 GHz bandll, N°4 2013, Road safety.
- [10] Página web oficial “Bosch Automotive Technology”:  
[http://www.bosch-mobility-solutions.com/en/de/homepage/homepage\\_1.html](http://www.bosch-mobility-solutions.com/en/de/homepage/homepage_1.html)
- [11] Hoja de datos del sensor LRR3 disponible en:  
[http://www.bosch-automotivetechnology.com/media/db\\_application/downloads/pdf/safety\\_1/en\\_4/lrr3\\_datenblatt\\_de\\_2009.pdf](http://www.bosch-automotivetechnology.com/media/db_application/downloads/pdf/safety_1/en_4/lrr3_datenblatt_de_2009.pdf)
- [12] S. Elvira, A. de Castro, J. Garrido, —ALO: An ultrasound system for localization and orientation based on anglesl, Microelectronic Journal, 44, 959-967, 2012.
- [13] Hoja de datos sensor ultrasonido disponible en:  
[http://www.maxbotix.com/documents/UCXL-MaxSonar-WR\\_Datasheet.pdf](http://www.maxbotix.com/documents/UCXL-MaxSonar-WR_Datasheet.pdf)
- [14] Más información acerca de las cámaras estereoscópicas disponible en:  
<http://ingeniatic.euitt.upm.es/index.php/tecnologias/item/394-c%C3%A1maras-estereosc%C3%B3picas>
- [15] NASA, Cámaras equipadas en el Curiosity, disponible en:  
[http://www.nasa.gov/mission\\_pages/msl/multimedia/malin-4.html#.U38jzvl\\_tig](http://www.nasa.gov/mission_pages/msl/multimedia/malin-4.html#.U38jzvl_tig)

- [16] Trincado Alonso, Juan Luis y Torres Salcedo, Beatriz y Pérez Alonso, Ana. “Sistema de Visión Estereoscópica para Navegación Autónoma de vehículos no tripulados”.2011.
- [17] Magister 3D. “Diseño de cámara estereoscópica real y virtual”. Disponible en: <http://www.cedav.net/rs/1162/d112d6ad-54ec-438b-9358-4483f9e98868/214/filename/disen0-camara-estereoscopica-y-virtual-f-1-y-2.pdf>
- [18] D. Martín Carabias, R. Requero García and J. A. Rodríguez Salor. “Sistema de visión Estereoscópica para navegación Autónoma de vehículos no tripulados”. 2010. Disponible en: <http://eprints.ucm.es/11278/>
- [19] Rankin, A.L.; Huertas, A.; Matthies, L.H. “Stereo vision based terrain mapping for off-road autonomous navigation”. Proc. SPIE 2009, 733210-17.
- [20] Hojas de datos sensors Blumleblee disponible en: [http://www.ptgrey.com/products/bbxb3/bumblebee2\\_xb3\\_datasheet.pdf](http://www.ptgrey.com/products/bbxb3/bumblebee2_xb3_datasheet.pdf)
- [21] Hoja de datos del sensor Carmine de Primesense disponible en: <http://www.i3du.gr/pdf/primesense.pdf>
- [22] Hoja de datos del sensor Kinect disponible en: [https://readytogo.microsoft.com/en-us/\\_layouts/RTG/AssetViewer.aspx?AssetUrl=https%3A%2F%2Freadytogo.microsoft.com%2Fen-us%2FAsset%2FPages%2F08%20K4W%20Kinect%20for%20Windows\\_Technical%20atasheet.aspx](https://readytogo.microsoft.com/en-us/_layouts/RTG/AssetViewer.aspx?AssetUrl=https%3A%2F%2Freadytogo.microsoft.com%2Fen-us%2FAsset%2FPages%2F08%20K4W%20Kinect%20for%20Windows_Technical%20atasheet.aspx)
- [23] Página web MESA: <http://www.mesa-imaging.ch/home/>
- [24] Hoja de datos del sensor de MESA: <http://www.mesa-imaging.ch/products/sr4000/>
- [25] T. Stoyanov, A. Louloudi, H. Andreasson and A.J. Lilienthal. “Comparative Evaluation of Range Sensor Accuracy in Indoor Environments”, in Proc. Eur. Conf. Mobile Robots, 2011, Örebro University, Sweden.
- [26] Hoja de datos del sensor HOKUYO-30LX disponible en: [http://www.hokuyo-aut.jp/02sensor/07scanner/download/pdf/UTM-30LX\\_spec\\_en.pdf](http://www.hokuyo-aut.jp/02sensor/07scanner/download/pdf/UTM-30LX_spec_en.pdf)
- [27] Página web oficial fabricante HOKUYO: <http://www.hokuyo-aut.jp/>
- [28] Hoja de datos del sensor disponible en: [http://www.riegl.com/uploads/tx\\_pxriegldownloads/10\\_DataSheet\\_LMS-Z210ii-S\\_07-06-2010.pdf](http://www.riegl.com/uploads/tx_pxriegldownloads/10_DataSheet_LMS-Z210ii-S_07-06-2010.pdf)
- [29] Página web oficial fabricante RIEGL: <http://www.riegl.com/>
- [30] Hoja de datos del sensor disponible en <http://www.velodynelidar.com/lidar/products/manual/HDL-64E%20Manual.pdf>
- [31] Página web oficial fabricante VELODYNE: <http://www.velodynelidar.com/lidar/lidar.aspx>
- [32] Más información disponible en: [http://www.cimco.com/docs/cimco\\_dnc-max/v7/es/#SerialComStandards](http://www.cimco.com/docs/cimco_dnc-max/v7/es/#SerialComStandards)
- [33] A. Chambers, S. Achar, S. Nuske, J. Rehder, B. Kitt, L. Chamberlain, J. Haines, S. Scherer, and S. Singh, "Perception for a river mapping robot", in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011, September 2011.

- [34] Tomoaki Yoshida, Kiyoshi Irie, Eiji Koyanagi and Masahiro Tomono, "A Sensor Platform for Outdoor Navigation Using Gyro-assisted Odometry and Roundly-swinging 3D Laser Scanner", October 2010, Taipei, Taiwan.
- [35] Matsumoto, M., & Yuta, S. (2010). 3D laser range sensor module with roundly swinging mechanism for fast and wide view range image. In IEEE conference on multisensor fusion and integration for intelligent systems, MFI, 2010 (pp. 156–161)
- [36] Hoja de datos HOKUYO 30LX-EW disponible en: <http://www.hokuyo-aut.jp/02sensor/07scanner/download/products/utm-30lx-ew/>
- [37] Más información sobre el entorno MATLAB disponible en: <http://es.mathworks.com/products/datasheets/pdf/matlab.pdf>
- [38] Más información sobre motores EC disponible en: <http://www.airtecnics.com/es/Technology/69/ventiladores-ec-generalidades>
- [39] Información sobre el código gray disponible en: [http://www.ecured.cu/index.php/C%C3%B3digo\\_Gray](http://www.ecured.cu/index.php/C%C3%B3digo_Gray)
- [40] Información sistema ROS disponible en: <http://www.ros.org/>
- [41] Página oficial Gazebo: <http://www.gazebo.org/>
- [42] Librerías utilizadas disponibles en: [http://gazebo.org/tutorials?cat=connect\\_ros](http://gazebo.org/tutorials?cat=connect_ros)
- [43] Nodo suministrado por el fabricante, disponible en: [http://wiki.ros.org/urg\\_node](http://wiki.ros.org/urg_node)
- [44] Lic Adriana Favieri, "Introducción a los Cuaterniones", Facultad Regional Haedo, Universidad Tecnológica Nacional, Argentina. 2008
- [45] Librerías e información sobre Pcl disponible en: <http://pointclouds.org/>
- [46] Estructura de datos pointCloud2 disponible en: [http://docs.ros.org/api/sensor\\_msgs/html/msg/PointCloud2.html](http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html)

## **Glosario**

---

LIDAR: Light Detection and Ranging  
UGV: Unmanned Ground Vehicle  
RADAR: Radio Detection and Ranging  
RPS: Revoluciones Por Segundo  
IP: Ingress Protection  
ROS: Robot Operating System

## Anexos

### A. Tabla sensores

TECNOLOGÍA	FABRICANTE	MODELO	APERTURA(º)	RESOLUCIÓN (º)	RANGO DETEC (m)	PRECISIÓN (cm)	VELOCIDAD ESCÁNER	CONECTIVIDAD	PROTECCIÓN
LÁSER 2-D	HOKUYO	UTM-30LX/LN	270	0,25	0.1-30	5	25 ms	USB	IP 64
		UTM-30LX-EW	270	0.25	0.1-30		25 ms	Ethernet	IP 67
	SICK	TIC 102	180	0.2	<10		20 ms	Ethernet	IP 66
		LMS511-101 PRO	190	0.167/0.25	0-40	5	10 ms	Ethernet/ USB	IP 67
		LMS511-111 Lite	190	0.25/0.5/1	0-40	5	13 ms	Ethernet/ USB	
		LMS151-10100	270	0.25/0.5	0.5-18	3	20 ms	Ethernet	
		LMS111-10100	270	0.25/0.5	0.5-18	3	20 ms	Ethernet/ USB	
		MSR400001	110	0.125/0.25	0.5-50		20 ms	Ethernet	IP 69 K
	RIEGL	LMS-Q120	80	0.01	hasta 150	2.5	20 ms	Ethernet	IP 64
		LMS-Q20	90	0.01	0-20	2.4	10ms		
VQ-250		360	0.01	1.5-75	1.5	10ms			
LÁSER 3-D	IBEO	LUX HD	de 85 a 110 (H)	0.125 (H)	<120	10	20 ms	Ethernet	IP 69K IP 68
			3.2 (V)	0.8 (V)					
		LUX 8L	110 (H) 6.4 (V)	0.125 (H) 0.8 (V)					
	RIEGL	LMS-Z210II-S	360 (H) 95 (V)	0.005	4-100	40 a 100m	50ms	Ethernet	IP 64
	VELODYNE	HLD-32E	360 (H)	0.09 (H)	1-80	2		Ethernet	IP 67
			40 (V)	1.33 (V)					
HLD-64E	360 (H)	0.09 (H)	1-50	< 2					
	26.8 (V)	1.33 (V)							
CÁMARA INFRARROJO	PRIME SENSE	CARMINE 1.08	57.5 (H) 45 (V)	640x480 (pixels)	0.8 - 3.5	3.4 (H,V) 1.2 (P)		USB	
		CARMINE 1.09	69 (P)		0.35 - 1.4	0.9 (H,V) 1 (P)		USB	
CÁMARA TIEMPO DE VUELO	MESA SR 4000	00400001/14	43 (H) 34 (V)	0.24 (H) 0.39 (V)	0.1-5	1	20 ms	USB	IP 64
		00400002/11						Ethernet	
	MESA SR4500	00400020/1	69 (H) 56 (V)		0-9	5	33 ms	Ethernet	IP 67
RADAR	BOSCH	LRR3	30 (H) 5 (V)		0.5-250	10	80 ms		

Tabla A-1: Desglose de características de sensores



## B. Tablas de mallado

1 RPS				
	180 °	90 °	45 °	30 °
30 m	4,5*5	2,5*2	1*1,5	1*1
20 m	3,5*3	1,5*1,5	1*1	0,8*0,8
10 m	2*2	0,9*0,9	0,5*0,5	0,3*0,3
5 m	0,8*0,8	0,3*0,3	0,2*0,2	0,1*0,2
1 m	0,2*0,2	0,1*0,1	0,1*0,1	0,1*0,1

2 RPS (1,8)				
	180 °	90 °	45 °	30 °
30 m	7*6	2*3	2*1,5	1,5*1
20 m	5,5*5	2*2,5	1*1,5	1*1
10 m	2*3	1*1	0,5*0,5	0,4*0,4
5 m	0,8*0,8	0,4*0,4	0,3*0,3	0,2*0,2
1 m	0,2*0,2	0,1*0,1	0,1*0,1	0,1*0,1

3 RPS				
	180 °	90 °	45 °	30 °
30 m	4,5*5	2,5*3	1*1,5	0,8*0,8
20 m	3,2*3	1,6*1,6	0,7*0,8	0,4*0,5
10 m	1,7*1,7	0,9*0,9	0,4*0,3	0,2*0,2
5 m	0,8*0,8	0,3*0,4	0,2*0,1	0,1*0,1
1 m	0,2*0,1	0,1*0,1	0,04*0,04	0,03*0,03

4 RPS (3,9)				
	180 °	90 °	45 °	30 °
30 m	5*5,5	2*3	1,1*1,5	0,8*1
20 m	3,5*4	1,7*1,7	0,7*0,9	0,5*0,7
10 m	2,5*2	0,9*0,9	0,4*0,4	0,3*0,3
5 m	1*1	0,3*0,5	0,2*0,2	0,15*0,15
1 m	0,2*0,3	0,1*0,1	0,04*0,04	0,04*0,03

5 RPS (4,9)				
	180 °	90 °	45 °	30 °
30 m	6*7	2*3	1,6*2	1,3*1,3
20 m	5,5*5	1,5*1,6	1,1*1,3	0,9*0,1
10 m	2,5*2,5	0,8*0,9	0,6*0,6	0,5*0,4
5 m	1,5*1,5	0,3*0,5	0,27*0,4	0,2*0,31
1 m	0,3*0,4	0,1*0,1	0,06*0,06	0,05*0,04

**Tabla B-1: Tablas originales de mallado, sin la imposición de una apertura específica**

## C. *Artículo*: A Varying Density 3D Laser Scanner for Unmanned Ground Vehicles Mapping and Obstacle Detection.

**Abstract—** In this paper a 3D laser scanning sensor design specifically tailored for mobile UGVs (Unmanned Ground Vehicles) has been proposed. The system exhibits interesting capabilities such as high speed continuous real-time point cloud generation in a cost effective way. The system is based on the fabrication of a rotating laser system, as an improvement of an off-the-shelf laser scanner, capable of generating 3D maps from the surrounding. The system is able to detect an obstacle of 10 centimeters at 10 meters in front of vehicle, making the highest density of points acquired by the sensory system point to the obstacle, so defining an angle of inclination.

### I. INTRODUCTION

In this article we describe the design of a 3D sensor system for autonomous and semi-autonomous navigation of an unmanned vehicle for outdoor operation. In particular the system will operate in field environments such as off-road tracks.

Autonomous and semiautonomous navigation for unmanned vehicles in non-pre-established off-road surrounds is an area of active research by many groups around the world [1] [2]. The achievement of this target is becoming closer and closer due to the high variety and improvements of the technology. A basic component of any unmanned robotic vehicle is the acquisition of exteroceptive models of the environment. The main goal of this paper is to describe the design of a 3D sensing system for an off-road unmanned vehicle.

There are many sensors based on different technologies that have been used for semiautomatic navigation of unmanned vehicles. These technologies are based on different data acquisition models, which will be briefly explained here:

- *Ultrasonic based systems*

These sensors are based on the time of flight concept: The ultrasonic waves are emitted, bounced against the objects, and returned to the system. The distance is calculated measuring the returning time of the wave, and knowing the speed of sound. On the other hand this type of technology suffers some problems that make it less suitable for off-road applications, like the directionality, the temperature reliance, the humidity or the dust. Even so, this option was used in some projects. [3]

- *RADAR Systems*

The main task of these sensors is to detect objects and measure the speed and the relative position of those. To get this, the sensor has four antennas transmitting simultaneously radar waves. These waves are reflected by the objects situated in front of the vehicle, allowing the acquisition of data on bidimensional space. This type of technology has been used for applications in semiautomatic navigation for unmanned vehicles in pre-established surroundings (highways, roads, etc...). If it is used in unstructured environments, it comes with many problems referring to the detection of some obstacles [4].

- *Systems based on stereoscopic cameras*

These systems copy the natural vision of the eyes of the human-being. The camera takes two images that are partially overlapping to each other, so that depth distances can be calculated, generating the result in a 3D effect. This type of technology is very sensitive to the oscillation in light intensity. This is an interesting option if some different sensors can be complemented with a stereoscopic camera. The system is very useful in semiautomatic navigation for unmanned vehicles, although it requires heavy data processing. Also, this system has an important problem regarding the sensibility to alignment of the measuring cameras. A small deviation in the calibration can make all subsequent distance calculation deviate. When the overlaps of the images become bigger, the system accuracy improves. Nevertheless, the

aperture of the system will be smaller. This technology is used in robots like Curiosity, Spirit [5] and other projects [6].

- *LIDAR systems (Light Detection And Ranging)*

These systems are chosen in a lot of projects dealing with semiautomatic navigation for unmanned vehicles thanks to its great features, such as high speed, independence from visibility variations, its accuracy and reliability of the provided data.

Many manufacturers and existing models make use of this technology. The Velodyne lasers 3D [7] and Sick lasers 2D [8] among many others are examples of laser measuring devices.

The acquisition of data in a 2D system is based on a diode laser that emits its beam toward a rotating mirror, getting a set of points in a bidimensional plane.

To get the 3D effect, a —stackl of lasers with different orientation angles is used or a system of internal rotation is done. This last increases complexity in a significant way.

Taking into account the different technologies previously explained, a system based on LIDAR has the following advantages over the other technologies:

- Accuracy in measurements.
- Measures invariant with light intensity. This is a very important parameter for the outdoor case.
- The alignment of the sensor device is not critical.
- Easy configuration of the desired angular aperture, dismissing the kind of data that is not desirable to process.
- Fast data processing.
- Lower rate of false positives than other systems. [9]

Because of these advantages, a system based on LIDAR has been chosen.

Then there is the decision of whether to choose a 3D or 2D laser system with an additional external rotation. The problem with a multilaser 3D system is that it is too expensive. Therefore, a 2D laser system has been chosen with an additional external rotation. Sensorial system is composed by these elements, together a processor to process the information and build point clouds of 3D ranging data.

Many studies have been done about obstacle detection and avoidance in outdoor environments, for example [1] and [2], in which many LIDAR sensors can be seen in a vehicle.

The requirements of sensorial system are:

- IP (Ingress protection)

It is imperative that the developed sensorial system must be protected against dust and water, very common in outdoor environments. This information is given by the manufacturer, in IP tables [10].

The sensorial system must have at least IP 66: Dust tight and protects against heavy water showers.

- Range and aperture

The LIDAR chosen must have a forward measurement range of 30 meters and have a customizable aperture of at least 180 degrees. In spite of gyro system, data in front of vehicle must be provided in real time.

- Accuracy and angular resolution

The sensorial system must have an accuracy that allows obstacle detection and avoidance with its data near of vehicle. The laser chosen must have at least 0.5 degrees of angular resolution so that can offer highly dense measurements.

At most 15 centimeters of ranging error at 30 meters has been chosen as accuracy benchmark.

- Light intensity

The system must support a light intensity of 100Klux in order to support operation under direct light of the sun.

- Scanning speed

The sensorial system should provide fast data acquisition so that it can process and detect obstacles in real time. This scan period should be at most 50 ms, delivering at least 1000 points per scan. So data will become available at least at 200000 points per second.

- Price

It is not always possible to obtain the best technology for the purpose stipulated at a reasonable cost. Sometimes it is necessary to look for other alternatives or integrate different technologies to achieve an effective price point.

- Data transfer

A data throughput channel should be provided to provide proper refresh rate of high frequency data. This is required to measure the environments in a very dynamic driving condition.

- 3D maps generation

The chosen system must be able to acquire and export three-dimensional measurements of the environment in real time in order to generate maps for processing the information provided. After considering all engineering requirements, a system based on a 2D laser sensor on a rotating platform has been proposed. A sensor model can be seen in the Fig 1.



Fig 1: 2D line scanning laser sensor

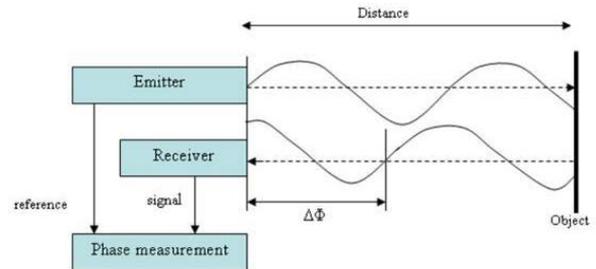


Fig 2: Time-of-Flight (TOF) laser scanning.

In the Fig. 3 can be seen a specific laser scanning chosen.

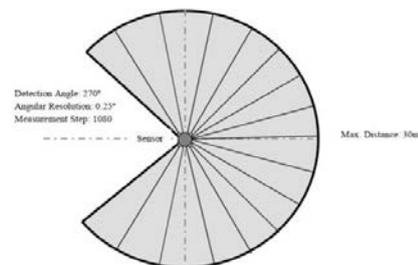


Fig 3: Diagram of Scanned Area

Each pass of the laser has 1080 points, obtained in 25 ms. It has a separation of 0.25 degrees in each point of beam.

Doing a simple calculation, the required accuracy can be verified :

$$\text{Accuracy} = 2\pi * (0.25^\circ / 360^\circ) * 30 \text{ metres} = 0.1309 \text{ meters.}$$

Rotation properties

The designed system will rotate continuously around the laser longitudinal axis, like it is shown in Fig. 4.

This configuration has been chosen due to the high data redundancy located in front of the laser, namely, more data and more continuously on the obstacles that could be in front of the vehicle. This place is where the detection time is most important. Continuous rotation avoids having to stop and start laser motion thus

## II. DESCRIPTION OF THE PROPOSED SYSTEM

It's worth to mention that the strategy used in this sensory system has been previously used in other applications [11]. The difference in this case is the laser directionality configuration, the rotation data treatment and its incorporation into a vehicle in order to get a semiautonomous navigation. This sensory system makes timing as a critical parameter in the rotation of the laser and the data collected from this scanner.

When the laser emits a beam and it bounces in any object, the internal rotating mirror's laser redirect the beam to an electronic circuit, which is able to calculate the time of the flight of the beam [12]. In Fig 2, this method can be seen.

maintaining high data rates in a uniform way. The system can slow down rotation to acquire more densely populated point clouds when necessary.

This system of rotation, while achieves high data redundancy in front of the vehicle, it makes enough information to detect multiple types of obstacles, negatives as well as positives, that are very important to provide a semiautonomous navigation. When the sensory system is rotated at least  $180^\circ$ , redundancy of data start to be obtained having more data density in rotation axis contour. With this, and considering the  $270^\circ$  of laser sweep, it is easy to generate a point cloud around vehicle as well.

Now if this rotation method is compared with others articles using laser yaw rotation as for example in [13], it is easy to point some disadvantages of previous systems when compared to the system proposed in this article.



Fig 4: Laser scanner with gyro system

In the first place, the system [13] ceases to obtain continuous data in front of the vehicle. For the semiautonomous navigation, the continuous data acquisition in front of vehicle is necessary, because an accidental collision can happen while the vehicle still moves. This requirement is accomplished in the presented system in this article. Furthermore, a complete 3D map is essential.

Others systems do not turn in a continuous shape, but they do beams from side to side. This option is not acceptable because deceleration produced by laser will make the important data for a safe and reliable navigation impossible to

be obtained. All of this, while the vehicle goes on moving, that will do that in the generated 3D map blind regions appear.

Another advantage of our system is that laser scanning can be concentrated in a specified region in order to measure with high accuracy and redundancy.

The continuous gyro system is composed of the followings components:

- Servo controlled precision motor.
- Encoder: Allows knowing exactly the speed and the positioning of the motor.
- Inertial measurement system: It is necessary to measure the inclination of the UGV and properly correct mapping data according to vehicle pose.

It is important to point out that these components will be accommodated inside a field body and have to be protected from water and dust particles from the environment.

### III. 3D MAPPING COVERAGE

It has been different behaviors of a laser scanning regarding density coverage as a function of rotational speed imposed by the system designed as above. With this, it can be studied if there is an optimal rotation speed depending on the distance to the obstacles, its shape, and the speed of the vehicle.

For the data analysis, meshing techniques using simulated data have been used to observe the mesh point density generated by the laser at different rotation speeds.

For this objective, simulated rotation of the laser beam has been done and the laser measurement pattern has been plotted in a simulation.

In Fig. 5 the obtained laser geometry at different rotation speeds is shown, where the axis of rotation of the sensory system is centered on the axis center of coordinates. The horizontal axis

represents width distance and vertical axis represents height distance both in meters.

Analysis of the curvature of the laser beam 30 meters away

In Fig. 5 is observed successive full scans of the laser, corresponding to one second of simulation. Also can be see the distortion suffered by the rotation of the laser at different speeds. The objective is to analyze the coverage patterns of laser beams depending on the revolutions per second.

In these figure can be seen that for a maximum longitudinal distance of 30 meters (in front of the laser) the lateral range of systems covers from -70 to 70 meters.

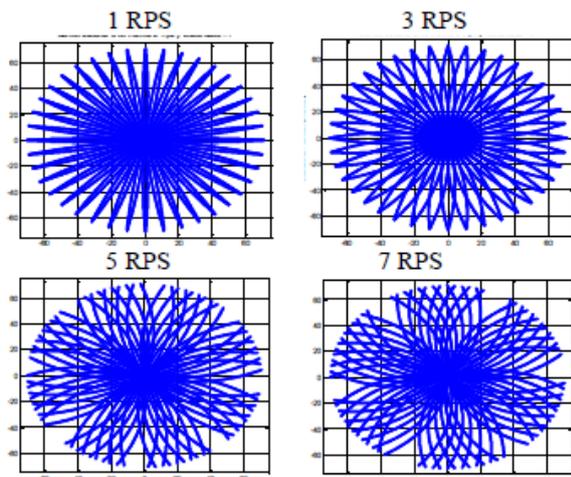


Fig. 5: laser scanner

In Fig 5, it can be seen that, as the rotation of the system gets to be increased, the laser scanning will be increasingly curved. An interesting fact is that homogeneity of the data is lost at speeds above 5 RPS, because each scanning suffers too many distortions to obtain acceptable density points.

From these results, it is observed that as the rotational speed of the system is increased, the laser scanner will be increasingly curved and so dead zones appear. Therefore, the homogeneity of the data is lost above 5 revolutions per second, in which each laser beam is too distorted to get an acceptable density points.

Revolutions/point density produced in 1 second simulation

To observe the direct relationship between the density of points and the speed of rotation of the sensory system it has been chosen a maximum aperture. The aperture is  $\pm 20$  meters in horizontal axis and  $\pm 5$  meters in vertical axis. This aperture is the most important region for analyzing in detail for the semiautonomous navigation in a mobile vehicle system with non-holonomic constrains. Therefore the angular apertures out of this region in front of the vehicle will be dismissed.

With this configuration, a system of minimum mesh based on variable cells is developed. This mesh is defined to have at least one point in each cell. In Table I, it is represented the experimental meshed values, in meters, for 1 second simulation with different apertures and for three rotation speeds.

TABLE I: Meshes with different apertures

	1 RPS				
	180°	90°	45°	30°	
30 m	1,6*1,6	1,6*1,6	1*1,5	1*1	
20 m	2*1,5	1,5*1,5	1*1	0,8*0,8	
10 m	2*2	0,9*0,9	0,5*0,5	0,3*0,3	
5 m	0,8*0,8	0,3*0,3	0,2*0,2	0,1*0,2	
	3 RPS				
	30 m	1,6*1,4	1,6*1,4	1*1,5	0,8*0,8
	20 m	2*1,7	1,6*1,6	0,7*0,8	0,4*0,5
	10 m	1,7*1,7	0,9*0,9	0,4*0,3	0,2*0,2
	5 m	0,8*0,8	0,3*0,4	0,2*0,1	0,1*0,1
	5 RPS (4,9)				
	30 m	1,6*2	1,6*2	1,6*2	1,3*1,3
	20 m	2,3*1,8	1,5*1,6	1,1*1,3	0,9*0,1
	10 m	2,5*2,5	0,8*0,9	0,6*0,6	0,5*0,4
	5 m	1,5*1,5	0,3*0,5	0,27*0,4	0,2*0,31

As an example of these result, for 5 meters away and 1 RPS, the information of each mesh is obtained with 80 centimeters horizontally and 80 centimeters vertically.

The meshed at 5 RPS has been improved with 1 and 3 RPS.

With these results, it can be seen that a high rotational speed does not involve better meshed. Therefore, a complementary study is required to

select the optimum rotational speed in each case, depending on the vehicle speed or density of points required.

In addition to the previous study of static densities coverage is necessary to consider the behavior under dynamic conditions. Therefore, the speed of the vehicle has been taken into account.

Knowing the speed of the vehicle and the time that it gets to stop, it can be estimated the maximum detection time to avoid the obstacle and consider the minimum rotation speed necessary to detect it. The minimum revolution per second has to permit to obtain points in all regions defined in the time established. With 180 degrees rotation, information of the complete environment can be obtained. The result can be seen in Table II.

In Fig 6 is presented an example of verification of these minimum values of rotation (Table II), in which it can be seen all laser scans performed up to that time and meshes generated at each speed of rotation.

The selected meshes appear in green, that later will be adjusted in the way which has been discussed. This adjustment can be seen in the Tables III and V.

Table II: Time until detection at 11 m/s

Distance to obstacle in meters	Real meters to detect the obstacle	Time until detection(s)	RPS minimum
6	1	0,09	5,555
7	2	0.18	2,75
8	3	0,27	1,85
10	5	0,45	1,11

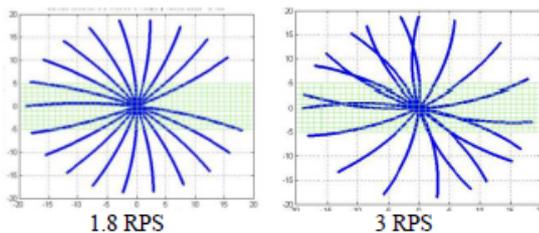


Fig.6: Scan at different rotations with time until detection

It has been determined that at 3 RPS the system was able to obtain data from all regions while at

1,8 RPS it has not turned round and therefore it have not been obtained data from all regions.

Now in Table III, the adjustment of the mesh is performed. This case of study is the worst possible because vehicle is considered to be moving at 11 m/s without any previous data of the sensory system. This situation will not be possible in real situations because before start moving the vehicle, the sensory system has to have created point clouds around the vehicle, and will have some knowledge of possible obstacles around. But that theoretical assumption is valid for the case that a dynamic spurious obstacle may arise in a given time.

The Table III shows that the minimum mesh at one revolution per second less than minimum RPS in Table II, involves a catastrophic meshed. Instead, as the revolution per second threshold is exceeded, the meshed improves considerably.

In a new case of study, there is more time to analyze the environment, because the vehicle speed is considerably smaller. This will get more data to work and therefore the minimum density mesh can be considerably improved. The results can be seen in Table IV and V.

The difference with the previous case is that as the RPS threshold is increased, again, it can be seen that more speed does not implies meshes improvement.

After several additional simulations, it can be deduced that the best mesh is obtained with the rotational speed at which minimally meets the detection condition.

That conclusion may conflict with the idea that as faster turn the laser, better redundancy and data acquisition environment obtained. These results have been used from an engineering point of view to design the rotating system to operate at the necessary maximum speeds and no more.

TABLE III: Meshing point

	MESHED (meters) ( $\pm 20, \pm 5$ )	
RPS	Horizontal axis	Vertical axis
1	>7	>7
1,8	6	6
3	5	5
3,9	4,5	5
4,9	4,5	4,5
5,9	4,5	4,5

TABLE IV: Time until detection at 8,33 m/s

Distance to obstacle in meters	Real meters to detect the obstacle	Time until detection(s)	RPS minimum
6	3,19	0,3823	1,3
7	4,19	0,5023	1
8	5,19	0,6223	<1

TABLE V: Meshed point

	Mesh (meters) ( $\pm 20, \pm 5$ )	
RPS	horizontal	vertical
1	2	2
1,8	2,3	2,3
3	2,5	2,5
3,9	2,5	3,5
4,9	>4	>4
5,9	>>4	>>4

### Minimum detectable obstacle and position laser study

The place chosen for installing the sensory system is near the bumper of the vehicle, it is has been set at 1 meter above the ground and with a slight down tilt to get maximum density surface data at the road level in front of the vehicle.

Angle of inclination has been chosen so as to allow detecting an obstacle of 10 cm high and 10 meters away from the vehicle, namely, that the higher density of points is focused at 10 cm from the ground to the distance. The tilt angle was calculated to achieve that objective, so it has to detect a conventional curb or step 10 cm high.

The graphs in Fig 7 show the laser scanning in 0.45 seconds, with different revolutions per second and with a mesh of 10 cm in order to verify that it is possible to detect an obstacle of that height at 10 meters from the vehicle.

Results show that the obstacle will always be detected in front of the vehicle even when the coverage pattern is dependent on rotation speed.

The system exhibits properties similar to the human vision where more resolution and detail

is obtained in the center of the measurement volume while at the periphery, data is processed at the same rate but with less detail. In case of detection of an interesting pattern at the periphery, the system can decide to “focus” more closely by reducing rotation speed when passing along the area.

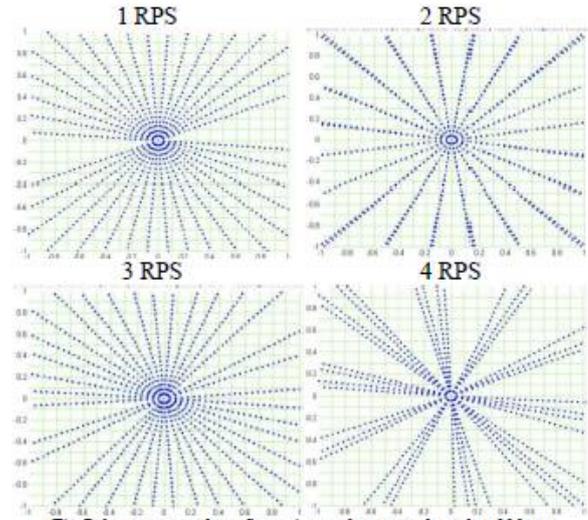


Fig 7: laser scan and configuration to detect an obstacle of 10 cm

### IV. FUTURE LINES

The sensory system is under construction. In a future work the experimental results of the system will be shown.

The system has great potential by using the output point cloud in further processing and integration with other sensory information for advanced representations of the environment as well as automatic object detection. In particular, integration of the laser point cloud data with visual information is highly promising for semi-autonomous UGV navigation.

The objective of this project is to equip a real vehicle with front sensors, that will detect and avoid any obstacles when this one is on any road type, also it will detect forbidden regions, that is, basing on the information provided by the sensors, the vehicle will be able to recognize the front parts of the road that it can't get into. All this will provide the capability to offer an assisted teleoperated driving.

## V. CONCLUSIONS

A 3D laser scanning sensor design specifically tailored for mobile UGVs has been proposed. The system exhibits interesting capabilities such as high speed continuous real-time point cloud generation in a cost effective way.

A state of the art study has been conducted. From all the range of sensors capable to acquire 3D data of the surrounding, the laser based sensor has been chosen as the best choice to achieve unmanned vehicle navigation.

The system is based on the fabrication of a rotating laser system, as an improvement of an off-the-shelf laser scanner, capable of generating 3D maps from the surrounding.

Simulation studies have been conducted to assess the resulting measurement data. Patterns, density and minimum detection studies have been done.

The rotational speed is adaptive to the speed of the vehicle. To calculate that speed, a density study was made, bringing up with important conclusions that will be used to get a big redundancy at data provided by the frontal sensors, so the unmanned driving will be safe and reliable.

The optimal rotational speed of the sensors system will be that one that is higher than the required threshold on each instance, and it must be the immediately higher, because improvements has not been detected at higher rotation speeds.

It is possible to detect an obstacle of 10 cm at 10 meters ahead of the vehicle, making the highest density of points acquired by the sensory system point to the obstacle, so defining an angle of inclination.

According to the simulations, both static and dynamic, the proposed 3D sensors system shows high performance in a cost effective way. The 3D sensor accomplishes well the objective of providing relevant real time measurements of the surrounding for unmanned vehicles.

## ACKNOWLEDGMENTS

This work has been done with support of the INNFACTO program, in the frame of project ARGOS, published in bulletin ECC/1345/2012. It is a joint project between Robomotion and the HCTLab, Human Computer Technology Laboratory, of Universidad Autónoma de Madrid, Spain.

## REFERENCES

- [1] M. Montemerlo et al., —Junior: The Stanford entry in the Urban Challenge, Journal of Field Robotics, vol. 25, no. 9, September 2008, Special Issue on DARPA Urban Challenge (Part II).
- [2] M. Montemerlo et al, —Stanley: The robot that won the DARPA Grand Challenge, J. Field Robot., vol. 23, no. 9, pp. 661– 692, Sep. 2009.
- [3] S. Elvira, A. de Castro, J. Garrido, —ALO: An ultrasound system for localization and orientation based on angles, Microelectronic Journal, 44, 959-967, 2012.
- [4] Davide Brizzolara, —Future trends for automotive radars: Towards the 79 GHz band, N°4 2013, Road safety.
- [5] NASA, —Seventeen Cameras on Curiosity. Available: [http://www.nasa.gov/mission\\_pages/msl/multimedia/malin-4.html#.U38jzvl\\_tig](http://www.nasa.gov/mission_pages/msl/multimedia/malin-4.html#.U38jzvl_tig)
- [6] D. Martín Carabias, R. Requero García and J. A. Rodríguez Salor, —Sistema de visión Estereoscópica para navegación Autónoma de vehículos no tripulados, 2010, available: <http://eprints.ucm.es/11278/>
- [7] Velodyne products available in: <http://velodynelidar.com/lidar/lidar.aspx>
- [8] Sick products available in: [http://www.sick.com/us/en-us/home/products/product\\_portfolio/Pages/product\\_portfolio.aspx](http://www.sick.com/us/en-us/home/products/product_portfolio/Pages/product_portfolio.aspx)
- [9] T. Stoyanov, A. Louloudi, H. Andreasson and A.J. Lilienthal, —Comparative Evaluation of Range Sensor Accuracy in Indoor Environments, in Proc. Eur. Conf. Mobile Robots, 2011, Örebro University, Sweden
- [10] IP (Ingress Protection), available in: <http://www.mpl.ch/info/IPratings.html>

- [11] C. Cano Olivera, —Telémetro laser 3D con sensor Hokuyo UTM-30LX, 2010, available in: <http://e-archivo.uc3m.es/handle/10016/9947?locale-attribute=en>
- [12] More info and application available in: [http://www.tankonyvtar.hu/en/tartalom/tamop425/0032\\_precizios\\_mezogazdasag/ch02s04.html](http://www.tankonyvtar.hu/en/tartalom/tamop425/0032_precizios_mezogazdasag/ch02s04.html)
- [13]: A. Chambers, S. Achar, S. Nuske, J. Rehder, B. Kitt, L. Chamberlain, J. Haines, S. Scherer, and S. Singh, —Perception for a river mapping robot, in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011, September 2011.

## D. Especificaciones sistema sensorial final

Operation	
Measurement Range	From 0.1 m to 30 m
Horizontal and Vertical Field Of View	270° Spherical
Angular Resolution	0.25 deg
Single Scan Period	25 msec
Accuracy	0.1m to 1m: +/- 10mm, 1m to 10m: +/- 30mm, 10m to 30m: +/- 50mm
Scanning Speed	43200 points per second
Dimensions	130 mm (height) x 195 mm (length) x 80 mm (width)
Weight	1kg (without cable)
Power Consumption	12 Watts max, 9.5 Watts mean (0.8A @ 12V)
Operating Temperature	-10°C to +50°C
Orientation correction	Yes. Internal IMU for real-time orientation correction.
Rotation Speed	User selectable from 0.1 deg/sec to 1080 deg/sec
Positioning Capability	User selectable angular control with 0.01deg accuracy
Laser	Pulsed Laser 905 nm
Safety	Laser Safety Class 1 (eye safe)
Environmental Protection	IP67 Ingress Protection
External Light	Up to 100000 lux
Interfaces	
Power Supply	12V DC +/-10%
Data	1x Ethernet 100BASE-TX (100Mbps) 2x USB 2.0 ports
Connector	1x AmphenolMIL
Software	Linux drivers, library and test application. ROS modules.



## PRESUPUESTO

### 1) Ejecución Material

- Compra de ordenador personal (Software incluido)..... 1.000 €
- Alquiler de impresora láser durante 6 meses..... 50 €
- Material de oficina..... 150 €
- Compra sensor HOKUYO30LX-EW ..... 3555 €
- Compra componentes para el sistema de giro
  - Compra anillo rotativo .....39,5 €
  - Compra encoder, reductora, motor y controlador.....181,7 €
  - Compra IMU.....79 €
  - Compra soporte mecánico de giro y anclaje al vehículo.....79 €
- Compra vehículo Yamaha Rhino.....3950 €
- Total de ejecución material..... 9084,2 €

### 2) Gastos generales

- 16 % sobre Ejecución Material..... 1.453,47 €

### 3) Beneficio Industrial

- 6 % sobre Ejecución Material..... 545,04 €

### 4) Honorarios Proyecto

- 1120 horas a 15 €/ hora..... 16.800 €

### 5) Material fungible

- Gastos de impresión..... 100 €
- Encuadernación..... 20 €

### 6) Subtotal del presupuesto

- Subtotal Presupuesto..... 28.002,71 €

### 7) I.V.A. aplicable

- 21% Subtotal Presupuesto ..... 5.880,56 €

### 8) Total presupuesto

- Total Presupuesto..... 33883,28 €

Madrid, Noviembre de 2014

El Ingeniero Jefe de Proyecto

Fdo.: Jose Ignacio Rejas Hernán  
Ingeniero de Telecomunicación





## **PLIEGO DE CONDICIONES**

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un desarrollo e integración de un conjunto de sensores para la navegación autónoma de un vehículo terrestre no tripulado. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

### **Condiciones generales**

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partidaalzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

### **Condiciones particulares**

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.

