



UNIVERSIDAD
DE MÁLAGA



TESIS DOCTORAL

Doctorado en Ingeniería Mecatrónica

Escuela de Ingenierías Industriales

Universidad de Málaga

NAVEGACIÓN REACTIVA DEL ROBOT MÓVIL ANDÁBATA EN ENTORNOS NATURALES

Autor: D. Mariano Morán Prados

Directores: Dr. Jorge L. Martínez Rodríguez

Dr. Jesús Morales Rodríguez

Departamento de Ingeniería de Sistemas y Automática

UNIVERSIDAD
DE MÁLAGA




Málaga, Junio de 2021



UNIVERSIDAD
DE MÁLAGA

AUTOR: Mariano Morán Prados

 <https://orcid.org/0000-0002-8186-7671>

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional:

<http://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>

Cualquier parte de esta obra se puede reproducir sin autorización pero con el reconocimiento y atribución de los autores.

No se puede hacer uso comercial de la obra y no se puede alterar, transformar o hacer obras derivadas.

Esta Tesis Doctoral está depositada en el Repositorio Institucional de la Universidad de Málaga (RIUMA): riuma.uma.es



AGRADECIMIENTOS

El trabajo que se presenta en esta tesis doctoral ha sido el fruto de la constancia y el esfuerzo de varios años de dedicación. Sin lugar a dudas, no habría podido compaginar mi actividad profesional con mi trabajo de investigación sin la inestimable ayuda de un buen número de personas a las que quiero expresar mi sincero agradecimiento.

En especial a mis directores de tesis, D. Jorge L. Martínez Rodríguez y D. Jesús Morales Rodríguez, por orientarme e introducirme a la robótica móvil, por la confianza depositada en mí, y por toda la ayuda prestada.

Sin olvidar a los que han colaborado en las publicaciones derivadas de esta tesis a través de sus respectivos proyectos de fin de carrera, D. Manuel Zafra, Dña. Olga Cordero, D. Antonio Pomares, D. Alfredo Robles y D. Manuel Sánchez. A los compañeros del Departamento de Ingeniería de Sistemas y Automática que me ayudaron a avanzar D. Javier Serón, D. Juan Manuel Gandarías, D. Manuel Toscano, D. Antonio Reina y D. Francisco Pastor.

A mi empresa y a mis compañeros de Promálaga, por su ayuda en mi día a día y por permitirme mejorar mi desarrollo profesional y por las responsabilidades que me han asignado.

A mis padres Juan N. y Amelia, por su ayuda incondicional y a mi hermano Manolo, por su apoyo desde la distancia.

Por último, a mi querida familia, mi mujer María José y a mis hijos Marcos y Estrella por tener la paciencia, apoyo y cariño durante estos años, sin su amor y comprensión este trabajo no habría sido posible.



UNIVERSIDAD
DE MÁLAGA

SÍNTESIS

En esta tesis se abordan posibles soluciones para la navegación reactiva del robot móvil terrestre Andábata, cuyo principal sensor exteroceptivo es un escáner láser tridimensional (3D), a baja velocidad en entornos naturales, donde se presentan numerosos retos debido a la complejidad del terreno.

Con este fin, se emplea el *Robot Operating System* (ROS) para organizar el software del robot y procesar la información procedente de los sensores a bordo y llevar a cabo diversas estrategias de navegación.

La primera estrategia consiste en construir mapas de elevación borrosos que describan el entorno cercano al vehículo. Dicha estrategia es bastante novedosa pero conlleva una alta carga computacional que reduce la reactividad del robot.

Para aliviar esta carga, se propone el entrenamiento clasificadores de transitabilidad del terreno mediante aprendizaje supervisado. Para ello es necesario disponer de nubes de puntos 3D etiquetadas sin error que se han generado de forma sintética en el simulador robótico Gazebo.

Finalmente, se analiza una nueva estrategia de navegación local basada en la clasificación de los puntos obtenidos por el telémetro láser 3D. Para ello, se construyen mapas de transitabilidad bidimensionales (2D) que permiten que Andábata evite de forma reactiva obstáculos tanto negativos como positivos.



UNIVERSIDAD
DE MÁLAGA

ABSTRACT

This thesis addresses possible solutions for reactive navigation of the ground mobile robot Andábata, whose main exteroceptive sensor is a three-dimensional (3D) laser scanner, at low-speed in natural environments, where numerous challenges arise due to the complexity of the terrain.

To this end, the *Robot Operating System* (ROS) is used to organize the robot's software and to process the information from the on-board sensors in order to carry out various navigation strategies.

The first strategy consists on building fuzzy elevation maps that describe the environment close to the vehicle. This strategy is quite novel but implies a high computational load that reduces robot's reactivity.

To alleviate this load, training ground traversability classifiers using supervised learning is proposed. For this purpose, it is necessary to have 3D point clouds labeled without errors that have been synthetically generated with the robotic simulator Gazebo.

Finally, a new local navigation strategy based on the classification of the points obtained by the 3D rangefinder is analyzed. To do so, two-dimensional (2D) traversability maps are built that allow Andábata to avoid in a reactive way both negative and positive obstacles.



UNIVERSIDAD
DE MÁLAGA

Acrónimos

1D Unidimensional.

2D Bidimensional.

3D Tridimensional.

ANFIS *Adaptive Network based Fuzzy Inference System.*

BA *Balanced Accuracy.*

BDT *Bagged Decision Tree.*

CC Corriente Continua.

DEM *Digital Elevation Map.*

DT *Decision Tree.*

FEM *Fuzzy Elevation Map.*

FN *False Negative.*

FP *False Positive.*

FRM *Fuzzy Reliability Mask.*

GB Giga Byte.

GBT *Gradient Boosted Tree.*

GNB *Gaussian Naive Bayes.*

GPS *Global Positioning System.*

ICR *Instantaneous Center of Rotation.*

IMU *Inertial Measurement Unit.*

JCR *Journal Citation Reports.*

KNN *K-Nearest Neighbors.*

LIDAR *Laser Imaging Detection And Ranging.*

LSVM *Linear Support Vector Machine.*

MCC *Mathews Correlation Coeficient.*

ODE *Open Dynamics Engine.*

PCA *Principal Components Analysis.*

PR *Precision.*

RAM *Random Access Memory.*

RE *Recall.*

RF *Random Forest.*

RGB *Red, Green and Blue.*

ROS *Robot Operating System.*

RPC *Remote Procedure Call.*

SLAM *Simultaneous Localization And Mapping.*

SVM *Support Vector Machine.*

TN *True Negative.*

TOF *Time of Flight.*

TP *True Positive.*

UAV *Unmanned Aerial Vehicle.*

UGV *Unmanned Ground Vehicle.*

UKF *Unscent Kalman Filter.*

USB *Universal Serial Bus.*

WiFi *Wireless Fidelity.*

Índice general

Síntesis	v
Abstract	vii
Acrónimos	ix
1. Introducción	1
1.1. Robótica de campo	1
1.1.1. Reactividad	4
1.2. Motivaciones de la tesis	4
1.3. Objetivos	5
1.4. Marco de realización	5
1.5. Estructura de la tesis	6
2. El robot móvil Andábata	9
2.1. Introducción	9
2.2. Hardware de Andábata	10
2.2.1. Telémetro láser 3D	13
2.3. Software de Andábata	15
2.3.1. Modelo cinemático aproximado	17
2.3.2. Obtención de nubes de puntos niveladas	19
2.4. Navegación global	22
2.5. Conclusiones	24
3. Navegación reactiva con mapas de elevación borrosos	27
3.1. Introducción	27
3.2. Filtrado de nubes de puntos	28
3.3. Construcción de FEMs y FRMs	30
3.3.1. Cálculo multiproceso de FEM y FRM	35
3.4. Selección de la dirección de movimiento	38
3.5. Resultados experimentales	40
3.6. Conclusiones	42

4. Clasificación de transitabilidad mediante aprendizaje supervisado	45
4.1. Introducción	45
4.2. Obtención de barridos láser 3D sintéticos etiquetados	48
4.3. Entrenamiento de clasificadores de transitabilidad del terreno	54
4.3.1. Cálculo de características espaciales	54
4.3.2. Aprendizaje supervisado	56
4.4. Validación de los clasificadores de transitabilidad	58
4.5. Experimentos de clasificación con barridos láser 3D reales	62
4.6. Conclusiones	65
5. Navegación reactiva mediante clasificación de transitabilidad	67
5.1. Introducción	67
5.2. Simulación de Andábata en Gazebo	68
5.3. Selección de la dirección de movimiento	71
5.4. Experimentos en simulación	75
5.5. Experimentos reales	79
5.5.1. Sendero en una hondonada	79
5.5.2. Parque urbano	81
5.5.3. Discusión de los resultados	82
5.6. Conclusiones	84
6. Conclusiones	87
6.1. Recapitulación	87
6.1.1. Comparativa con otros trabajos	89
6.2. Publicaciones derivadas	89
6.3. Trabajos futuros	90
A. Trabajos relacionados	93
Referencias	99

Capítulo 1

Introducción

1.1. Robótica de campo

Hoy en día, los vehículos terrestres no tripulados o *Unmanned Ground Vehicle* (UGV) pueden emplearse en tareas cada vez más exigentes en aplicaciones robóticas de campo (*field robotics*) tales como minería [1], búsqueda y rescate [2] [3], agricultura [4] [5], operaciones militares [6], navegación fuera de carretera [7] o exploración de otros planetas [8] [9] (ver Figura 1.1). Pueden desenvolverse en estos entornos complejos gracias a la percepción que le proporcionan sus sensores a bordo [10].

Para la navegación en entornos naturales, es necesario establecer un compromiso entre eficiencia en la navegación y la seguridad detectando obstáculos en la dirección de movimiento [11]. El análisis del entorno en exteriores es más complejo que en interiores, como en un camino rural donde se pueden encontrar árboles, piedras o socavones [12]. Además, algunos elementos del entorno pueden ser considerados como obstáculo o transitable, dependiendo de la configuración del robot y de sus capacidades de navegación [13]. Así, por ejemplo, una determinada pendiente puede superarse por algunos robots pero para otros puede suponer que sus ruedas comiencen a perder tracción.

No hace mucho, los robots autónomos debían procesar la información de los sensores durante minutos para tomar la decisión de hacia donde navegar. El aumento de la capacidad de proceso computacional y el abaratamiento de los componentes

electrónicos en los últimos años ha permitido el desarrollo de complejos sistemas de navegación que permiten completar los cálculos durante el desplazamiento.

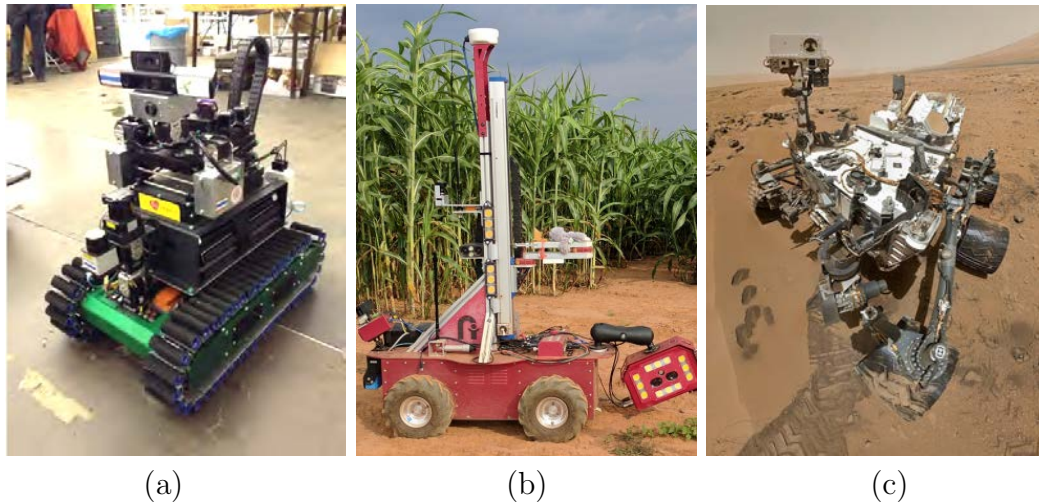


Figura 1.1: (a) Robot de búsqueda y rescate iRap, ganador de la Robocap de 2016 [14]. (b) Robot para agricultura Robotanist [15]. (c) Robot Curiosity, enviado a Marte en 2011 [16].

Para la navegación autónoma es fundamental que el UGV disponga de sensores a bordo que faciliten información actualizada del entorno (sensores exteroceptivos) y del comportamiento del propio robot (sensores propioceptivos) [17]. Los tipos de sensores empleados más comúnmente son: unidades de medida inercial o *Inertial Measurement Unit* (IMU) [18], cámaras RGB [19], cámaras estéreo [20], cámaras de tiempo de vuelo o *Time of Flight* (TOF) [21], telémetros o LIDAR bidimensionales [22] o tridimensionales [23] y GPS [24] entre otros. Resulta habitual utilizar distintos tipos de sensores de forma combinada para obtener información fiable sobre la que planificar los movimientos del robot [25].

Los sensores visuales (cámaras RGB, estéreo y TOF), proporcionan gran cantidad de información pero tienen un rendimiento que varía con la intensidad lumínica [26]. Por el contrario, los LIDAR presentan un comportamiento más confiable, pero tienen una densidad de puntos desigual que decae con la distancia al sensor [27] y también requieren de procesamiento previo [28]. En interiores, los sensores bidimensionales suelen ser suficientes para desenvolverse adecuadamente, pero en aplicacio-

nes de campo son necesarios sensores que proporcionen información tridimensional del entorno.

Los datos de profundidad del entorno se pueden obtener con cámaras estéreo [29] o cámaras de tiempo de vuelo [30]. Se logran rangos más lejanos combinando barridos láser 2D sucesivos mientras el vehículo avanza [31, 32, 33] o con un LIDAR [34]. En el último caso, el sensor puede ser un modelo comercial multihaz [35, 36], que tiene una alta frecuencia de exploración, o un escáner 2D actuado más asequible, que admite una resolución más alta pero requiere más tiempo de adquisición [37, 38, 39].

El procesamiento de escenas al aire libre adquiridas con un escáner láser 3D es muy diferente del de objetos [40], ya que en cualquier dirección y a cualquier distancia se pueden encontrar datos relevantes. Además, puede haber oclusiones, rangos erróneos o numerosos puntos de áreas no conectadas. Sin embargo, estos problemas no han obstaculizado la navegación autónoma procesando nubes de puntos directamente [39, 41].

En áreas urbanas y en interiores, el procesamiento de la información del sensor láser puede emplearse el reconocimiento de características geométricas comunes como planos [42]. La clasificación del terreno natural se puede realizar calculando características destacadas sobre la distribución espacial local de puntos 3D [43] [44]. A partir de las nubes de puntos 3D o de datos visuales se pueden generar tanto mapas de transitabilidad [45] como mapas digitales de elevación o *Digital Elevation Map* (DEM) [46].

En los últimos años, la adquisición de datos 3D de UGV se realizaba generalmente deteniendo el robot [47] [48], pero, hoy en día, se puede realizar en movimiento utilizando algún tipo de localización y mapeo simultáneos (SLAM) combinando datos de odometría [38] o de una unidad inercial [49].

Las nubes de puntos como datos de entrada para la navegación de vehículos terrestres pueden usarse directamente y descartarse inmediatamente [50, 51] o pueden almacenarse de forma incremental usando SLAM y emplearse más tarde [52]. En general, la última opción implica construir y mantener una representación explícita

del entorno a través de un mapa global 3D [39].

1.1.1. Reactividad

La reactividad es un componente necesario en la navegación autónoma para tratar evitar los obstáculos en el entorno [53]. Se pueden encontrar peligros desconocidos en terrenos naturales por encima y por debajo del nivel del suelo del vehículo, lo que comúnmente se conocen como obstáculos positivos y negativos, respectivamente [7, 27, 35].

Los comportamientos reactivos son de uso común para los vehículos terrestres para evitar obstáculos locales en terreno accidentado [54], terreno con vegetación [55], escenarios de desastre [52] o exploración planetaria [47, 56] mientras se intenta alcanzar los puntos objetivo previamente planificados [57]. En este contexto, el riesgo o interés asociado a los movimientos inmediatos del vehículo [58], tales como líneas rectas [31, 51], arcos circulares [7, 59] o ambos [56], deben ser evaluados para producir comandos de velocidad y dirección adecuados [50, 60].

La planificación del movimiento y la evaluación de la capacidad de desplazamiento pueden ocurrir directamente sobre la nube de puntos 3D [36, 41] o en una representación compacta de datos de profundidad 3D como un DEM [37, 54] o una cuadrícula horizontal 2D [61, 62]. La rugosidad y las pendientes del terreno se suelen considerar para evaluar la transitabilidad de DEM [31, 47]. Las celdas de los mapas 2D pueden contener datos de transitabilidad borrosos [60] o valores de ocupación precisos como libre, obstáculo o desconocido [62].

1.2. Motivaciones de la tesis

En esta tesis se han estudiado algunos de los múltiples problemas que tiene todavía abiertos la robótica móvil en exteriores. En concreto, la navegación de robots terrestres sobre entornos naturales plantea numerosos desafíos debido a la presencia de terrenos irregulares con desniveles, vegetación diversa y otros tipos de obstáculos

tanto positivos como negativos. Por esta razón, la integridad del vehículo dependerá de la adecuada localización y ejecución de movimientos durante sus desplazamientos.

Los LIDAR son sensores exteroceptivos que han visto reducido su coste en los últimos años por lo que la comunidad científica ha incrementado su uso en robótica por la gran cantidad de información fiable que proporcionan. Sin embargo, es necesario fusionar dicha información con otros sensores si se desea que el UGV continúe moviéndose durante las capturas de los barridos, y procesarla adecuadamente para determinar por donde podría circular el robot con seguridad.

1.3. Objetivos

Teniendo en cuenta las anteriores motivaciones, los objetivos que se han planteado para esta tesis son los siguientes:

- Diseño y ajuste de un sistema completo de navegación para entornos naturales desarrollado con el *Robot Operating System* (ROS) [63] para que el robot móvil Andábata puede operar sin detenerse.
- Desarrollo e implementación de reactividad mediante mapas locales de navegación construidos a partir de barridos láser 3D adquiridos desde el vehículo en movimiento.
- Aplicación de la lógica borrosa y del aprendizaje supervisado para la determinación de la transitabilidad del terreno cercano al robot móvil.

1.4. Marco de realización

Esta tesis se enmarca dentro del grupo de investigación de Robótica y Mecatrónica del Departamento de Ingeniería de Sistemas y Automática de la Universidad de Málaga. En particular, esta tesis ha sido financiada por los siguientes proyectos de investigación:

- Proyecto de la Junta de Andalucía P10-TEP-6101-R con título: Navegación autónoma de un robot móvil 4x4 en entornos naturales mediante GPS diferencial y telémetro láser tridimensional (2013-2017). Investigador principal: Dr. Jorge L. Martínez Rodríguez.
- Proyecto nacional DPI 2015-65186-R con título: Sistema multi-robot para cooperación con equipos de rescate de primera respuesta humanos y caninos en escenarios de catástrofe *FIRST-ROB* (2016-2019). Investigadores principales: Dr. Alfonso García Cerezo y Dr. Antonio Mandow Andaluz.
- Proyecto de la Junta de Andalucía UMA18-FEDERJA-090 con título: Desarrollo de técnicas de control inteligente con aprendizaje para navegación de vehículos autónomos en entornos no estructurados *DIOMEDES* (2019-2021). Investigadores principales: Dr. Jesús Fernández Lozano y Dr. Jesús Morales Rodríguez.
- Proyecto nacional RTI2018-093421-B-100 con título: Hacia equipos resilientes de manipuladores *UGV* y *UAV* para tareas robóticas de búsqueda y rescate *TRUST-ROB* (2019-2021). Investigadores principales: Dr. Alfonso García Cerezo y Dr. Antonio Mandow Andaluz.

1.5. Estructura de la tesis

La memoria de esta tesis se encuentra dividida en seis Capítulos y las correspondientes referencias bibliográficas. Cada Capítulo, a excepción del primero y del último, comienza con un apartado introductorio y finaliza con otro donde se presentan las conclusiones alcanzadas.

El presente Capítulo “1” denominado “Introducción” se encarga de proporcionar una visión general de esta tesis doctoral, incluyendo sus motivaciones, sus objetivos y la estructura que se ha seguido.

El Capítulo 2, titulado “El robot móvil Andábata”, presenta el robot que ha

sido empleado en los experimentos realizados en esta tesis. Se desarrollan sus características, su hardware, todos sus componentes software y la comunicación entre ellos en ROS.

El Capítulo 3, bajo el epígrafe “Navegación reactiva con mapas de elevación borrosos”, aborda el desarrollo e implementación de la navegación reactiva a baja velocidad en entornos naturales mediante mapas locales de elevación borrosos contruidos a partir de barridos láser 3D adquiridos desde el vehículo en movimiento.

En el Capítulo 4, denominado “Clasificación de transitabilidad mediante aprendizaje supervisado”, analiza la determinación de la transitabilidad de terrenos naturales mediante aprendizaje supervisado. Para ello se generan nubes de puntos sintéticas para entrenar los modelos que se utilizarán en la clasificación de nubes de puntos reales.

El siguiente Capítulo 5, con título “Navegación reactiva mediante clasificación de transitabilidad” explica cómo puede implementarse la navegación reactiva empleando la clasificación de la transitabilidad. Para ello, se han realizado simulaciones con Gazebo para conseguir ajustar los parámetros de la navegación y pruebas de campo reales con el robot.

El Capítulo 6, titulado “Conclusiones”, se dedica a presentar las conclusiones del trabajo realizado, así como a establecer las posibles líneas futuras de investigación derivadas del mismo y detalla las publicaciones realizadas en el marco de esta tesis.



UNIVERSIDAD
DE MÁLAGA

Capítulo 2

El robot móvil Andábata

2.1. Introducción

Andábata es un robot móvil terrestre para exteriores que pesa 41 kg y funciona con baterías (ver Figura 2.1). Fue diseñado y construido en el ámbito del proyecto de investigación *Navegación autónoma de un robot móvil 4x4 en entornos naturales mediante GPS diferencial y telémetro láser tridimensional* [64].

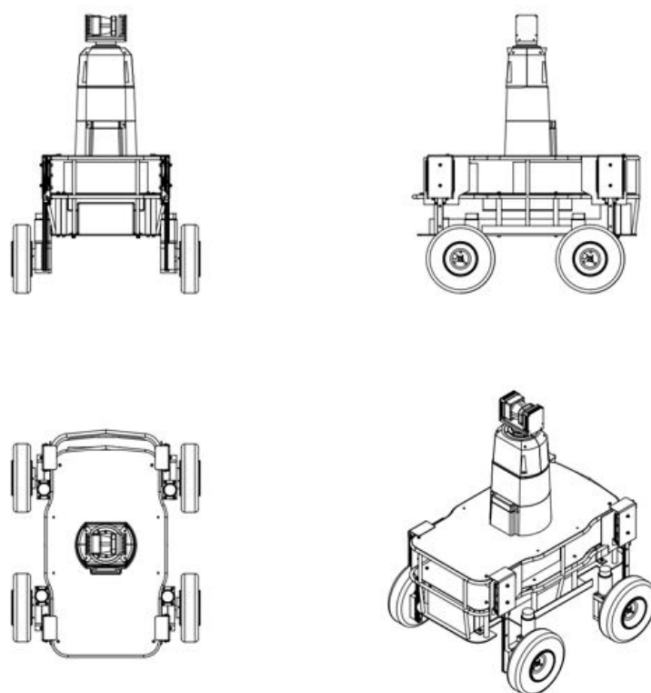


Figura 2.1: Vistas esquemáticas del robot móvil Andábata [65].

El resto del Capítulo está organizado de la siguiente forma. La siguiente sección 2.2 introduce el Hardware del robot y sus elementos principales, incluyendo su telémetro láser 3D en el apartado 2.2.1. Los componentes software del robot Andábata se explican en la sección 2.3, el modelo cinamético aproximado en el apartado 2.3.1 y la obtención de nubes de puntos niveladas en el apartado 2.3.2. Se continua con la navegación global en la sección 2.4. El Capítulo finaliza con las conclusiones en la sección 2.5.

2.2. Hardware de Andábata

Andábata es un vehículo eléctrico rodado para la navegación al aire libre, que tiene 0.67 m de largo, 0.54 m de ancho y 0.81 m de alto [51]. Es un robot con tracción por deslizamiento (*skid-steering*) y tiene como sensor exteroceptivo principal un telémetro láser 3D centrado en la parte superior (ver Figura 2.2), que se basa en la rotación sin restricciones de un telémetro láser bidimensional (2D) comercial alrededor de su centro óptico [66].

Andábata dispone de cuatro ruedas de diámetro 20 cm. Cada rueda tiene un



Figura 2.2: Componentes de Andábata en la parte superior.

sistema de suspensión pasiva (con dos resortes y una guía lineal con una carrera de 6.5 cm) y es accionada por su propio motor de CC con un codificador y una reductora (ver Figura 2.3).

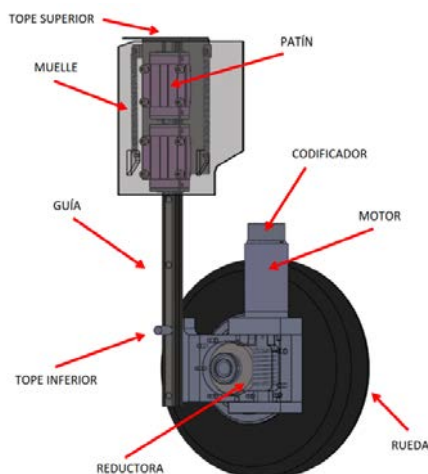


Figura 2.3: Suspensión pasiva acoplada a la rueda [65].

El chasis del robot tiene tres niveles: el inferior para los motores y la batería, el intermedio para la electrónica y la computadora, y el superior para los sensores y un enrutador WiFi (ver Figura 2.2). La computadora tiene un disco duro de estado sólido de 250 GB de capacidad y un procesador Intel Core i7 4771 de 4 núcleos, 3.5 GHz, 8 MB de caché y 16 GB de RAM. Se utiliza una tableta para la supervisión remota y la teleoperación del robot a través del enrutador WiFi (ver Figura 2.4). Además, Andábata porta un teléfono inteligente para obtener datos de su GPS, inclinómetro, acelerómetro, magnetómetro y de su cámara.

Los motores son impulsados por dos etapas de potencia Sabertooth 2x32 que están conectadas a la batería de 29.6 V (ver Figura 2.5) y son controladas por dos tarjetas Kangaroo [65]. Estos controladores, reciben comandos de velocidad para las bandas de rodadura izquierda y derecha, y envían mediciones de los *encoders* de los motores a la computadora a través de un puerto USB (ver Figura 2.4). La velocidad máxima del vehículo es $v_{max} = 0.68 \text{ m s}^{-1}$ para movimiento en línea recta, pero disminuye hasta llegar a cero a medida que se reduce el radio de giro.

El teléfono inteligente incorporado a Andábata es un LG Nexus 4 E960 que

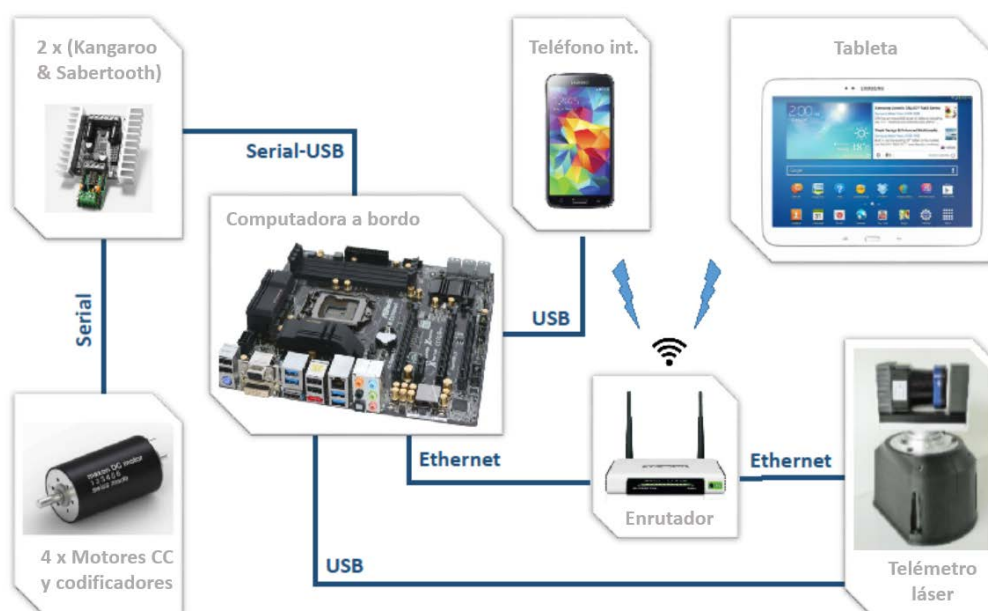


Figura 2.4: Comunicación entre los componentes hardware de Andábata.

presenta las siguientes características: localización por GPS con una precisión de 10 m, cuenta con magnetómetro para ubicar el robot respecto al norte magnético y dispone inclinómetros para poder evaluar en que postura se encuentra el robot.

El robot móvil utiliza los datos de los sensores proporcionados por el teléfono inteligente. Estos datos, junto con las imágenes de vídeo de la cámara, se envían a la tableta Samsung Galaxy Tab 3 mediante la red WiFi establecida por el enru-

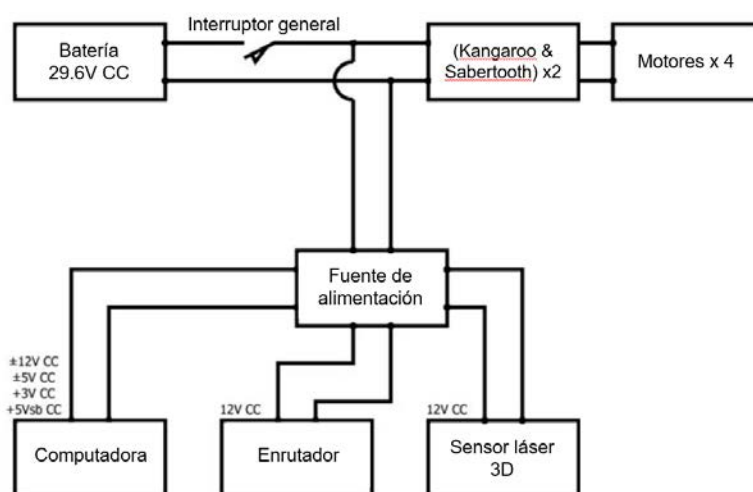


Figura 2.5: Sistema eléctrico de Andábata.

tador de abordo para supervisar y teleoperar de forma remota el movimiento del robot. Además, el teléfono se conecta vía USB a la computadora como interfaz de la aplicación Andábata con la tableta.

2.2.1. Telémetro láser 3D

El telémetro láser 3D de Andábata está construido a partir de un escáner bidimensional (2D) Hokuyo UTM-30LX-EW [66], pesa 1.9 kg y sus dimensiones son 125 mm, 170 mm y 122 mm. Las principales partes de este sensor láser son la cabeza giratoria, donde se ubica el telémetro láser 2D y la base que proporciona el giro adicional (ver Figura 2.6).

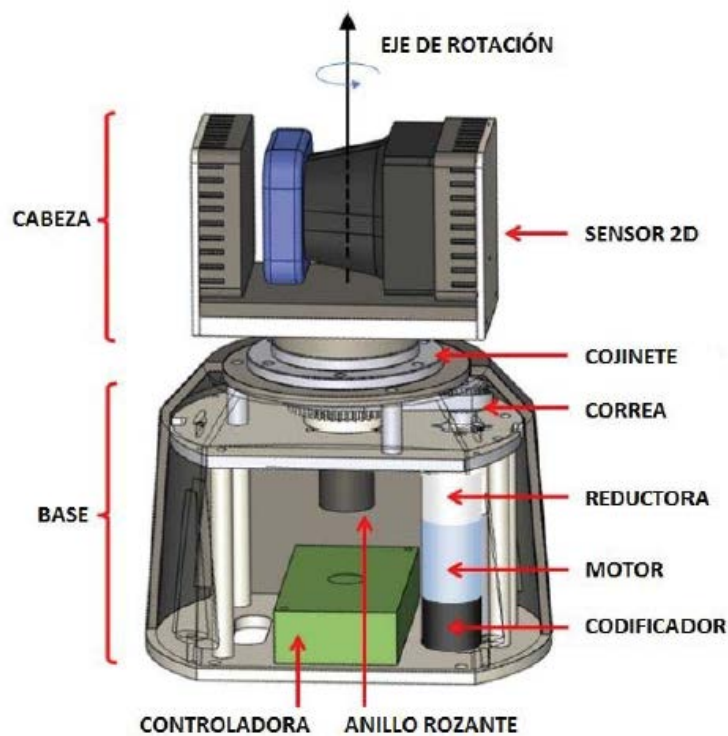


Figura 2.6: Partes del telémetro láser 3D.

El eje de rotación del sensor láser coincide con el centro óptico del sensor 2D. De esa forma, el sensor 3D resultante no requiere de desplazamientos de coordenadas. La transmisión de energía y datos entre la base y la cabeza se realiza mediante un

anillo deslizante que permite una rotación sin restricciones [66].

En la Figura 2.7 se muestra el diagrama funcional del telémetro láser 3D. La computadora a bordo puede acceder al sensor 2D y al controlador de movimiento a través de diferentes puertos de comunicaciones. El enlace Ethernet del sensor láser 2D se utiliza para el envío continuo de barridos y para su configuración. La señal de sincronización del sensor es capturada por el controlador de movimiento, que envía la lecturas del codificador junto con las lecturas temporales a la computadora a través de USB. La computadora ordena la velocidad de giro al controlador de movimiento a través de USB. La rotación de la base se consigue mediante un servomotor dirigido por una controladora cuyo objetivo es mantener constante la velocidad de giro de la cabeza y sincronizar la rotación con la adquisición de datos 2D [66].

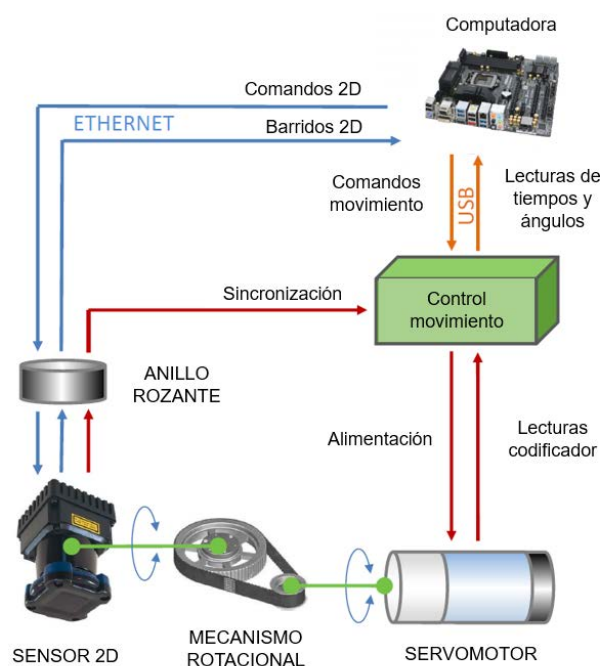


Figura 2.7: Diagrama funcional del telémetro láser 3D. [66]

El sensor 3D ha heredado del escáner 2D su campo de visión de 270°, su resolución vertical de 0.25°, su precisión de ± 3 cm y su rango de mediciones desde 0.1 m a 15 m bajo la luz solar directa.

La resolución horizontal del telémetro 3D depende de la velocidad de giro de la

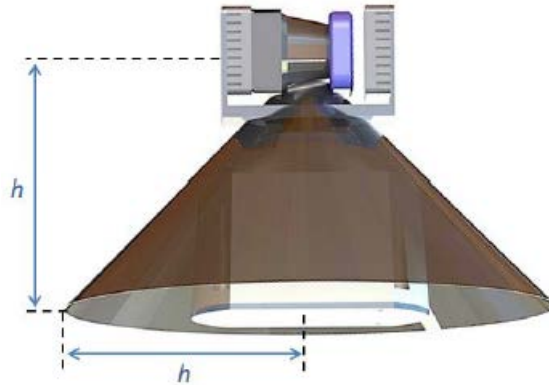


Figura 2.8: Zona ciega del escáner láser 3D.

cabeza. La zona ciega del sensor 2D es 90° , que se sitúa debajo para evitar interferencias con el robot. Así, la zona ciega del telémetro láser 3D es un cono con el radio de la base igual a la altura h de su centro óptico sobre el suelo (ver Figura 2.8) [51]. Cuando el escáner láser 3D está instalado en Andábata la altura del cono equivale $h= 0.723$ m.

2.3. Software de Andábata

El continuo avance de la robótica en el ámbito de la investigación y su aplicación práctica supone, a menudo, un reto debido a su creciente complejidad. El software para los sensores y actuadores de un robot deben ser reutilizable y facilitar su integración en nuevos proyectos. *Robot Operating System* (ROS) [63] ofrece un entorno para el desarrollo libre que ofrece herramientas y librerías que facilitan la puesta en marcha de sistemas robóticos complejos.

ROS no es un sistema operativo, pero provee de los servicios estándar de éste, tales como abstracción de hardware, control de dispositivos de bajo nivel, implementación de funcionalidades usadas habitualmente, paso de mensajes entre procesos y mantenimiento de paquetes. También ofrece la posibilidad de ejecutar código de forma distribuida entre múltiples computadoras.

ROS engloba una red *peer-to-peer* (entre iguales) de nodos, que pueden estar

distribuidos entre varias computadoras y que se comunican libremente usando la infraestructura de comunicaciones de ROS. La comunicación puede realizarse de diferentes formas, incluyendo la forma síncrona (petición/respuesta) cuando se utilizan *services* (servicios) y la transmisión asíncrona de datos mediante *topics* (temas).

El sistema completo de navegación de Andábata está íntegramente implementado con ROS bajo Linux. La Figura 2.9 muestra el esquema general de los nodos ROS empleados por Andábata.

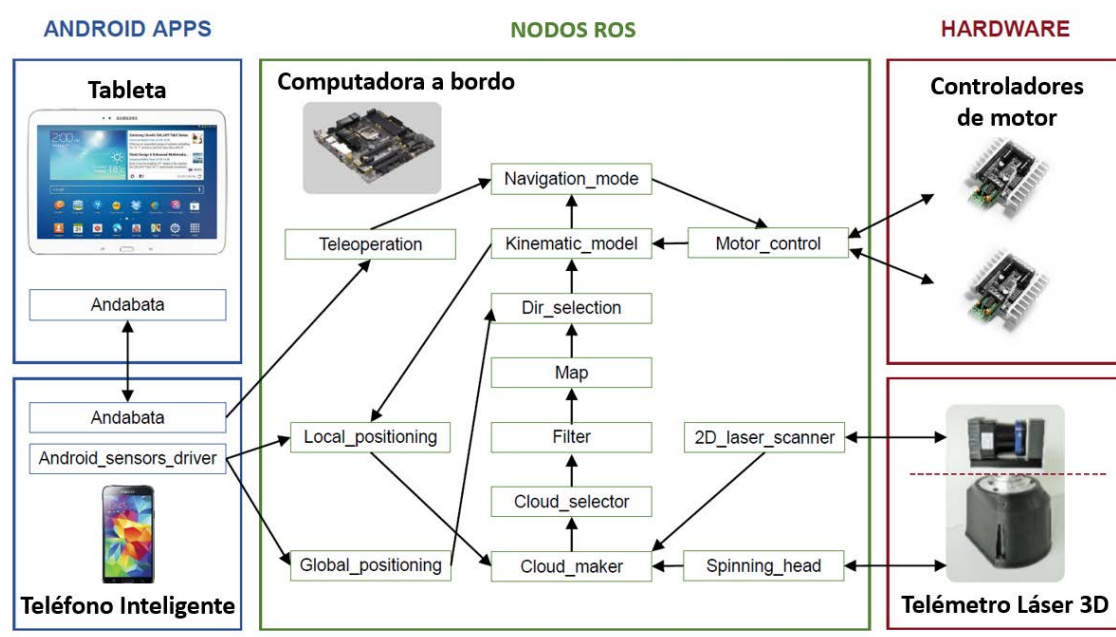


Figura 2.9: Esquema de software de Andábata.

Andábata puede ser teleoperado por una persona a través de la aplicación de Android en la tableta llamada *Andabata*, que es una adaptación de la App *Mover-Bot* [67] que se ejecuta tanto en la tableta como en el teléfono inteligente de a bordo. La interfaz de la tableta tiene dos controles deslizantes para ambas manos para indicar la velocidad de las bandas de rodadura izquierda y derecha, que se envían a través de la red WiFi local a la aplicación *Andabata* en el teléfono inteligente (ver Figura 2.10).

Durante la teleoperación, la aplicación *Andabata* en el teléfono inteligente retransmite los comandos de velocidad lateral desde la tableta al nodo *Teleoperation*

a través de USB (ver Figura 2.10).



Figura 2.10: Interfaz de usuario en la tableta.

El teléfono inteligente proporciona la retroalimentación de video, la posición del GPS y la inclinación del robot a la tableta a través de WiFi. Otra aplicación en el teléfono inteligente, llamada `Android_sensors_driver` [68], publica un *topic* ROS directamente al ordenador con todos los datos sensoriales del teléfono inteligente a través de WiFi.

El nodo `Navigation_mode` selecciona los comandos para controlar la velocidad de las ruedas mediante teleoperación o con navegación autónoma, que se envían a las controladoras de los motores a través del nodo `Motor_control`. El resto de nodos se comentan en las siguientes secciones.

2.3.1. Modelo cinemático aproximado

Andábata utiliza un modelo cinemático aproximado que explota la equivalencia entre la tracción por deslizamiento y la tracción diferencial (ver Figura 2.11). El nodo `Kinematic_model` se encarga de calcular las velocidades longitudinales y angulares reales del vehículo (ver Figura 2.9). Para este propósito, emplea las lecturas de los codificadores de los motores (obtenidas por el nodo `Motor_control` de los controladores del motor) en un modelo cinemático 2D aproximado de *skid-steering* [69].

El modelo cinemático simétrico directo relaciona las velocidades longitudinal y angular del vehículo (v y ω , respectivamente) con las velocidades de las bandas de rodadura izquierda y derecha medidas por los codificadores (v_l y v_r , respectivamente) como:

$$v = \frac{v_l + v_r}{2}, \quad (2.1)$$

$$\omega = \frac{v_r - v_l}{2y_{ICR}}, \quad (2.2)$$

donde $y_{ICR} = 0.386$ m es el valor medio del *Instantaneous Center of Rotation* (ICR) de las bandas de rodadura [51].

Por el contrario, las entradas de control v_l^{sp} y v_r^{sp} se pueden obtener a partir de las velocidades deseadas v^{sp} y ω^{sp} a partir del modelo cinemático simétrico inverso como:

$$v_l^{sp} = v^{sp} - y_{ICR} \omega^{sp}, \quad (2.3)$$

$$v_r^{sp} = v^{sp} + y_{ICR} \omega^{sp}. \quad (2.4)$$

Si alguna de las entradas de control excede los límites de $v_{max} = \pm 0.68$ m s⁻¹,

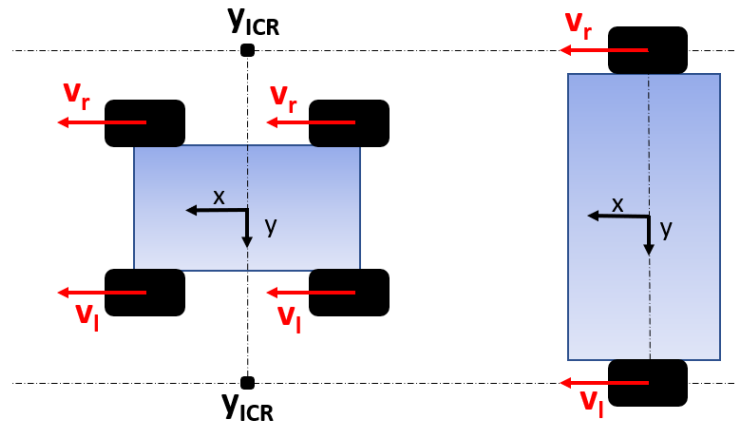


Figura 2.11: Equivalencia entre un vehículo con tracción por deslizamiento (izquierda) y diferencial (derecha).

las velocidades deseadas se dividen por el factor positivo:

$$e = \frac{|v^{sp}| + y_{ICR} |\omega^{sp}|}{|v_{max}|}, \quad (2.5)$$

para mantener el radio de giro deseado r^{sp} para el vehículo:

$$r^{sp} = \frac{v^{sp}}{\omega^{sp}} = \frac{v^{sp}/e}{\omega^{sp}/e}. \quad (2.6)$$

La curvatura deseada γ^{sp} del vehículo también se mantiene inalterada:

$$\gamma^{sp} = \frac{\omega^{sp}/e}{v^{sp}/e} = \frac{\omega^{sp}}{v^{sp}}. \quad (2.7)$$

De esta manera, la velocidad lineal máxima del vehículo v_{max} solo se puede lograr durante el movimiento en línea recta, y se aproxima a cero cuando se reduce r^{sp} [51].

Para realizar un movimiento, el nodo `Dir_selection` envía las velocidades longitudinales v^{sp} y angular ω^{sp} deseadas al nodo `Kinematic_model`, que se convierten en velocidades de los motores izquierdo y derecho utilizando el modelo cinemático inverso.

2.3.2. Obtención de nubes de puntos niveladas

Esta sección describe la adquisición de nubes de puntos 3D niveladas en Andábata mientras se mueve sobre terreno irregular. Para este propósito, es necesario sincronizar el movimiento del vehículo con la adquisición de los barridos 2D. Con todo, representa un esquema de SLAM local implementado a bajas velocidades sin cierre de bucle [70]

Sea XYZ el sistema de referencia asociado con el robot, que se encuentra en el centro geométrico del rectángulo definido por los puntos de contacto de las ruedas con el suelo. El eje X coincide con el eje longitudinal del vehículo apuntando hacia adelante, el eje Z es normal al plano de soporte y apunta hacia arriba y el eje Y completa este sistema de coordenadas ortogonales apuntando hacia la izquierda (ver

Figura 2.12).

Las coordenadas Cartesianas para una nube de puntos 3D nivelada se expresan en un sistema de referencia $X_p Y_p Z_p$ cuyo origen coincide con el del vehículo al comienzo de cada barrido 3D. El eje X_p equivale a la proyección de X en el plano horizontal, $Y_p = Y$ y Z_p se define con la misma dirección que la gravedad pero en sentido opuesto (ver Figura 2.12).

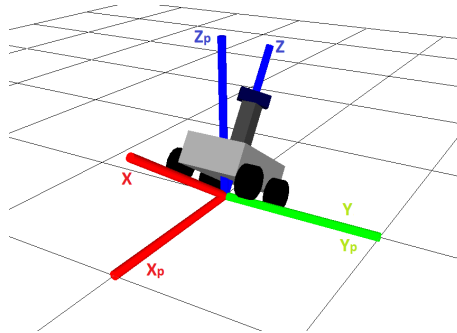


Figura 2.12: Sistemas de referencia para el robot (XYZ) y para un barrido 3D nivelado ($X_p Y_p Z_p$).

El nodo `Local_positioning` (ver Figura 2.9) emplea un *Unscent Kalman Filter* (UKF) [71] para ubicar el vehículo respecto a $X_p Y_p Z_p$ durante la adquisición de todo el barrido 3D. Con este objetivo, el nodo fusiona los ángulos de alabeo y cabeceo, la velocidad de guiñada del vehículo (proporcionada por el teléfono inteligente y publicada por la aplicación `Android_sensors_driver`), y las velocidades lineal y angular del vehículo proporcionadas por el nodo `Kinematic_model`.

Para la adquisición de un barrido láser 3D, el nodo `Spinning_head` ordena una velocidad de rotación constante para la cabeza y recibe periódicamente (cada 25 ms) el ángulo relativo entre la cabeza y la base al comienzo de cada barrido 2D a través de la conexión USB. De esta manera, este nodo puede publicar la transformación entre el sistema de referencia del sensor láser 2D y XYZ . Además, el nodo `2D_laser_scanner` permite configurar el sensor Hokuyo y publica los barridos 2D que la computadora recibe a través de la conexión Ethernet.

Finalmente, el nodo `Cloud_maker` publica la nube de puntos nivelada 3D combinando los sucesivos barridos verticales 2D (publicados por el nodo `2D_laser_scanner`)

con las transformaciones relativas entre el sistema de referencia del escáner láser 2D y XYZ (provisto por el nodo `Spinning_head`) y entre XYZ y $X_pY_pZ_p$ (publicado por el nodo `Local_positioning`).

Con Andábata detenido, un giro horizontal de 180° de la cabeza del escáner 3D es suficiente para capturar todo el entorno del robot, pero con el vehículo en movimiento, es necesario realizar un giro completo de 360° . En la Figura 2.13 pueden observarse dos ejemplos de barridos de giros de 180° y 360° respectivamente, siendo en esta última donde se evitan las áreas no muestreadas alrededor del robot y se reduce la zona ciega del sensor 3D.

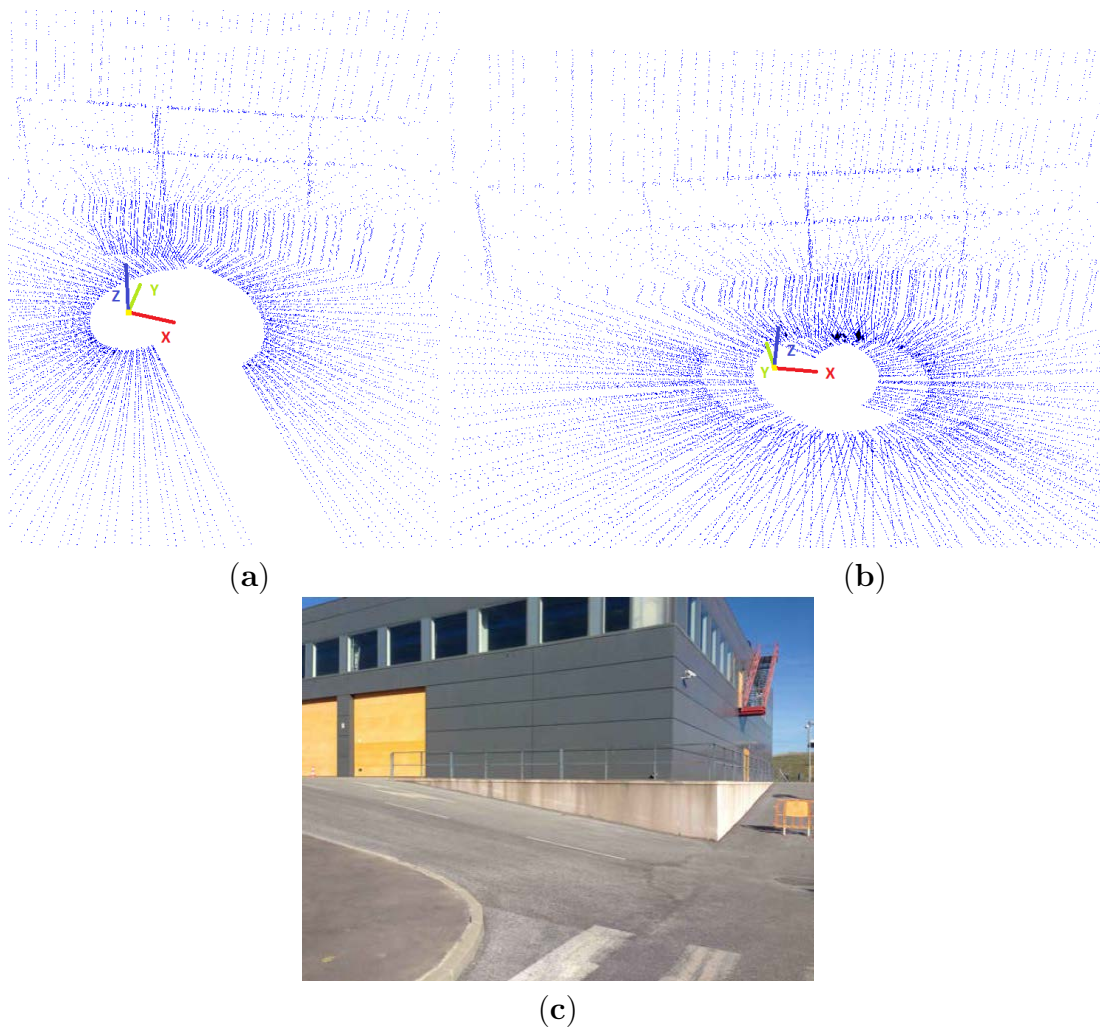


Figura 2.13: Barridos láser 3D nivelados obtenidos con una rotación de 180° (a) y 360° (b) mientras Andábata se movía a lo largo de una carretera en pendiente (c).

2.4. Navegación global

La navegación en terrenos naturales se puede realizar siguiendo puntos de objetivos distantes dados por sus coordenadas geográficas [72] [73]. Estos puntos de referencia se pueden planificar en base a mapas terrestres 2D o a imágenes aéreas [74], y representan la trayectoria que, en términos generales, debe seguir el vehículo. Pero para navegar entre puntos de objetivo distantes, el robot debe evitar obstáculos locales [60].

Una misión de navegación para Andábata se define mediante una lista ordenada de puntos objetivo distantes, cada uno determinado por sus coordenadas GPS de latitud y longitud. La estrategia de navegación consiste en acercarse a los puntos objetivo mientras se evitan los obstáculos locales de manera reactiva. La misión finalizará cuando se alcance el último punto objetivo (finalización de la misión) o cuando los ángulos de alabeo o cabeceo del vehículo superen 20° (abortar misión debido a una inclinación excesiva). La velocidad longitudinal durante la navegación v^{sp} es un valor constante que depende de la dificultad del terreno.

Se ha utilizado el paquete de ROS `actionlib` [75] para implementar esta estrategia de navegación, ya que la misión debe contar con varios objetivos y tarda cierto tiempo en completarse y debe poder cancelarse o interrumpirse en cualquier momento.

Una vez determinada la dirección de movimiento hacia el punto objetivo d_j , se obtiene el error de orientación δ_j (ver Figura 2.14). La velocidad angular deseada ω^{sp} se calcula empleando la ley de control de movimiento que corrige δ_j , actuando sobre la velocidad angular deseada :

$$\omega^{sp} = G \delta_j, \quad (2.8)$$

donde $G = 1$ es una ganancia proporcional que regula los cambios de orientación del vehículo [73].

La posición relativa y la orientación del robot móvil con respecto al punto obje-

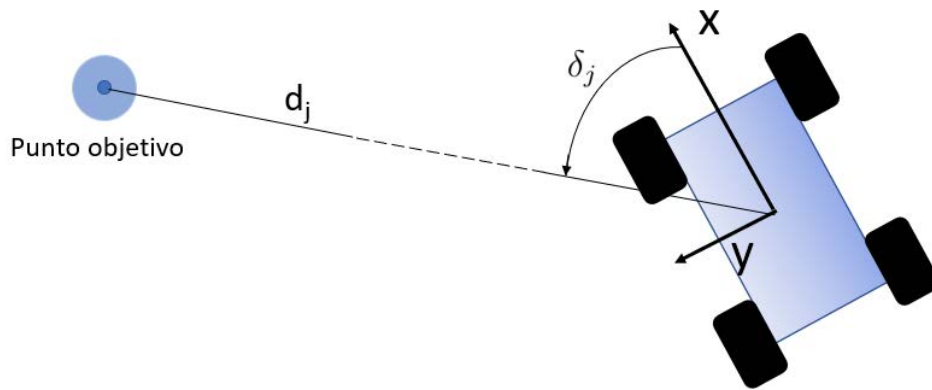


Figura 2.14: Control de la orientación respecto de un punto objetivo.

tivo actual se obtienen mediante el nodo `Global_positioning`. Este nodo procesa los datos del GPS y el magnetómetro del teléfono inteligente publicados por la aplicación `Android_sensors_driver` (ver Figura 2.9).

Antes de comenzar la navegación, es necesario calibrar el magnetómetro, ya que sus mediciones pueden verse afectadas por el entorno y por el propio robot móvil. La calibración consiste en girar el robot sobre si mismo durante una vuelta completa sobre un plano horizontal. Luego, las medidas magnéticas en los ejes X e Y , B_x y B_y , respectivamente, se ajustan para parecerse a una circunferencia (ver Figura 2.15).

Posteriormente, la orientación del vehículo con respecto al norte, es decir, el ángulo de guiñada *yaw*, se puede obtener durante la navegación como:

$$yaw = \arctan\left(\frac{B_x}{B_y}\right), \quad (2.9)$$

con lo que se deduce la orientación relativa del vehículo con respecto al objetivo actual.

Cuando la distancia entre la posición actual de Andábata, dada por sus coordenadas GPS, y el punto objetivo es menor que un cierto umbral, el punto objetivo se considera ya visitado y el siguiente punto objetivo de la lista se convierte en el punto objetivo actual. Ésto puede provocar un giro brusco para compensar el nuevo

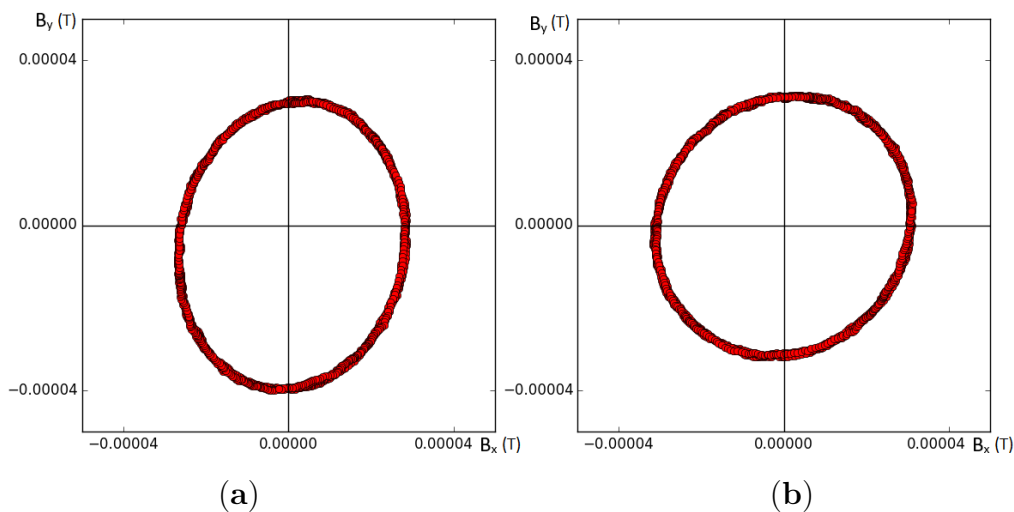


Figura 2.15: Mediciones de campo magnético antes (a) y después de una calibración (b).

error de orientación.

Los nodos `Filter`, `Map` y `Dir_selection` de la Figura 2.9 se explican en los siguientes capítulos, porque dependen de la estrategia reactiva que se elija a la hora de analizar la información proporcionada por los barridos láser y el resto de sensores.

2.5. Conclusiones

En este capítulo se ha presentado al robot móvil Andábata para navegar en exteriores. En los diferentes apartados se han descrito de forma pormenorizada los distintos componentes hardware del robot, haciendo hincapié en los más relevantes, como su sistema de suspensión pasiva y su telémetro láser 3D.

Posteriormente se ha introducido los componentes software de Andábata, cuyo código está enteramente desarrollado bajo el entorno ROS. El esquema software comprende la implementación de un modelo cinemático aproximado y la obtención de nubes de puntos 3D niveladas en movimiento.

Finalmente, se presenta la navegación global punto a punto, que se realiza mediante los sensores GPS y el magnetómetro que porta Andábata para desplazarse de

forma autónoma y alcanzar una lista de objetivos indicados previamente mediante sus coordenadas GPS.

Sin embargo, los puntos objetivos están muy alejados unos de otros, por lo que el robot debe reaccionar ante posibles obstáculos que encuentre a su paso. Así, la parte a desarrollar en los siguientes capítulos comprende la forma de elaborar algún tipo de mapa local del entorno y determinar la dirección a seguir a partir de las información proporcionada por las nubes de puntos.

El autor de esta tesis ha participado en el desarrollo de los componentes software presentados en este Capítulo, empleando librerías en ROS y realizando las pruebas necesarias para comprobar el correcto funcionamiento del sistema completo de Andábata en entornos naturales.



UNIVERSIDAD
DE MÁLAGA

Capítulo 3

Navegación reactiva con mapas de elevación borrosos

3.1. Introducción

La lógica borrosa se ha utilizado desde hace tiempo en robótica móvil [76] [77]. El objetivo que se persigue con su uso es manejar la incertidumbre inherente a los entornos y a los sensores. Así, el modelado borroso se ha utilizado en el diseño de comportamientos [78] [79], en la planificación de caminos [80] y en la clasificación de información proporcionada por los sensores.

Una nube de puntos 3D nivelada es una representación del entorno que se puede utilizar para la navegación local [81]. Los DEM son mapas divididos en celdas 2D que contienen la altura media del terreno que encierran [82]. Los mapas de elevación ofrecen una forma compacta de representar las nubes de puntos 3D con una estructura de datos 2,5-dimensional [83]. Sin embargo, estos mapas no permiten representar, por ejemplo, obstáculos colgantes como puentes o copas de árboles.

Los mapas de elevación borrosos o *Fuzzy Elevation Map* (FEM) emplean la lógica borrosa para evitar las discontinuidades inherentes a la discretización del suelo y para funcionar con incertidumbres en la medida de los sensores [84] [85]. Sin embargo, la interpolación difusa puede producir soluciones poco confiables en áreas donde las medidas son dispersas, y es necesario usar una máscara de confiabilidad o *Fuzzy Reliability Mask* (FRM) para cada FEM [37] [86]. Este Capítulo describe

la navegación reactiva en entornos naturales utilizando mapas de elevación borrosos con el robot móvil Andábata.

La Figura 3.1 ilustra el proceso que se va a seguir para realizar la navegación autónoma de Andábata. Tras la obtención de la barridos 3D, se realizará un filtrado. Los puntos resultantes del filtrado se emplean para la construcción del FEM y de la FRM, sobre los que se analiza la transitabilidad y se elige la dirección para realizar la navegación local.



Figura 3.1: Pasos para la navegación mediante mapas de elevación borrosos.

El Capítulo está organizado de la siguiente manera. La siguiente sección 3.2 aborda el filtrado de la nubes de puntos. La construcción de FEM y FRM a partir de la nube de puntos 3D filtrada se describe en la sección 3.3. La computación multiproceso de FEM y FRM en ROS se analiza en la subsección 3.3.1. La sección 3.4 presenta la estrategia de navegación reactiva y en la sección 3.5 se presentan algunos resultados experimentales. El Capítulo termina con las conclusiones en la sección 3.6.

3.2. Filtrado de nubes de puntos

Antes de comenzar el procesamiento, es necesario filtrar las nubes de puntos 3D niveladas. Estos pasos son realizados por el nodo `Filter` (ver Figura 2.9), y su salida es procesada por el nodo `Map` para construir mapas de navegación locales. Finalmente, el nodo `Dir_selector` escoge la dirección de movimiento más adecuada. El filtrado se compone de tres pasos consecutivos: eliminación de puntos lejanos,

eliminación de voladizos y normalización de la altura máxima.

El primer paso es eliminar los puntos lejanos, concretamente aquellos cuya proyección en el plano horizontal está más lejos de $d_{max} = 10$ m desde el origen del barrido 3D. Este paso es necesario por la baja visibilidad del terreno desde el punto de vista del escáner láser 3D, que está limitado a h en Andábata (ver sección 2.2.1) y porque las zonas distantes se muestrean con un menor densidad que las más cercanas. Además, la cantidad de puntos a procesar se reduce y la nube de puntos 3D resultante se enfoca en el área donde tendrán lugar los próximos movimientos del vehículo.

Los voladizos que no constituyen un obstáculo para Andábata, como el techo de un túnel o la copa de un árbol, deben eliminarse. Para este propósito, el segundo paso del filtro, emplea el método de cubos abatibles con cubos binarios de arista de $E=0.5$ m [87]. Básicamente, se descartan los cubos 3D ocupados por encima de un espacio de paso vertical mínimo. Por lo tanto, los puntos dentro de cubos abatidos se eliminan de la nube de puntos. Dos cubos vacíos son suficientes para que Andábata navegue de manera segura por debajo de los voladizos.

En el tercer paso, la altura máxima de los obstáculos se limita a 3 m sobre el suelo. El objetivo de este paso es que la altura de los obstáculos altos y pequeños pueda ser comparable. Para este propósito, se reducen las coordenadas z de los puntos 3D dentro de cubos gruesos con cinco o más cubos por encima del cubo suelo.

La Figura 3.2 muestra un barrido 3D nivelado obtenido mientras Andábata se movía en un parque urbano (ver Figura 3.9). Se puede observar en la Figura 3.2 que los puntos lejanos (azul) y los voladizos del dosel de los árboles (verde) se descartan del barrido 3D filtrado (rojo). También es visible que la altura máxima de los árboles ha sido limitada con la consiguiente reducción de los puntos a procesar posteriormente.

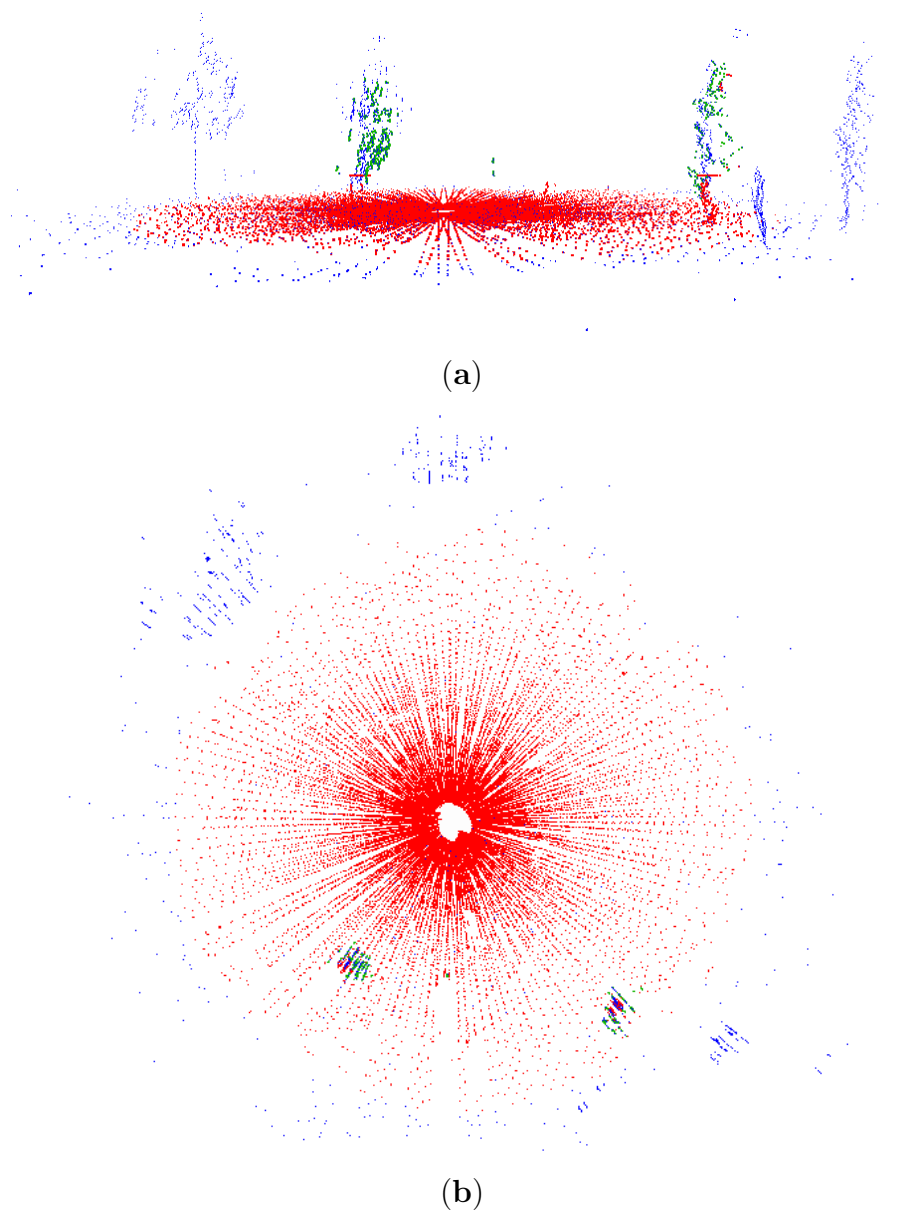


Figura 3.2: Vistas lateral (a) y aérea (b) de una nube de puntos 3D filtrada (rojo) obtenida de una nube de puntos 3D nivelada. Los colores azul y verde representan puntos lejanos y voladizos, respectivamente.

3.3. Construcción de FEMs y FRMs

Esta sección describe el cálculo de un FEM y su FRM a partir de una nube de puntos 3D filtrada. En lugar de utilizar ANFIS para la identificación [84][37][86], se ha empleado el modelado borroso de mínimos cuadrados [88] para mejorar el tiempo

de cálculo. Además, se han empleado coordenadas cilíndricas (α, r, z) en lugar de coordenadas Cartesianas (x, y, z) .

Se supone que la superficie del suelo alrededor del robot puede ser representada como una función continua:

$$z = H(\alpha, r), \quad (3.1)$$

donde la proyección de un punto 3D de coordenadas (x, y, z) en el plano $X_p Y_p$ (ver Figura 2.12) se describe mediante el ángulo α y la distancia r :

$$\alpha = \arctan(y/x), \quad (3.2)$$

$$r = \sqrt{x^2 + y^2}. \quad (3.3)$$

De manera similar, una FRM se define como una función continua $v = V(\alpha, r)$, donde los valores de v cercanos a uno y cero, representan los valores de confiabilidad máximos y mínimos para el FEM, respectivamente. FEM y FRM se definen ambos en el universo $U_\alpha = [0^\circ, 360^\circ]$ para la variable α y $U_r = [0 \text{ m}, d_{max} = 10 \text{ m}]$ para r , que corresponde a un círculo centrado en el origen de $X_p Y_p$.

Un total de dieciséis conjuntos difusos han sido definidos uniformemente sobre U_α (ver Figura 3.3 a). En este caso, las funciones triangulares de pertenencia μ_i para $i = 1, 2, \dots, 16$ están definidas como:

$$\mu_i(\alpha) = \begin{cases} \frac{\alpha - \alpha^{i-1}}{\alpha^i - \alpha^{i-1}} & \text{si } \alpha^{i-1} \leq \alpha < \alpha^i, \\ \frac{\alpha^{i+1} - \alpha}{\alpha^{i+1} - \alpha^i} & \text{si } \alpha^i \leq \alpha < \alpha^{i+1}, \\ 0 & \text{en otro caso,} \end{cases} \quad (3.4)$$

donde $\alpha^i = (i - 1)22.5^\circ$ representa los valores máximos, es decir, $\mu_i(\alpha^i) = 1$.

Para la distancia r , se han definido diez conjuntos difusos distribuidos desigualmente sobre U_r para proporcionar un mayor detalle a las regiones más cercanas al robot (ver Figura 3.3 b). Concretamente, el parámetro de pico r^j para $j = 1, 2, \dots, 10$

se calcula como:

$$r^j = \left(\frac{\tau^j - 1}{\tau^{10} - 1} \right) \left(d_{max} - \frac{h}{2} \right) - \frac{h}{2}, \quad (3.5)$$

donde $\tau = 1,0624$ es una relación de expansión que cumple que $r^1 = h/2$. Las funciones de pertenencia μ_j para r se definen como:

$$\mu_j(r) = \begin{cases} 1 & \text{si } r \leq r^1 \text{ o } r \geq r^{10}, \\ \frac{r-r^{j-1}}{r^j-r^{j-1}} & \text{si } r^{j-1} \leq r < r^j, \\ \frac{r^{j+1}-r}{r^{j+1}-r^j} & \text{si } r^j \leq r < r^{j+1}, \\ 0 & \text{en otro caso.} \end{cases} \quad (3.6)$$

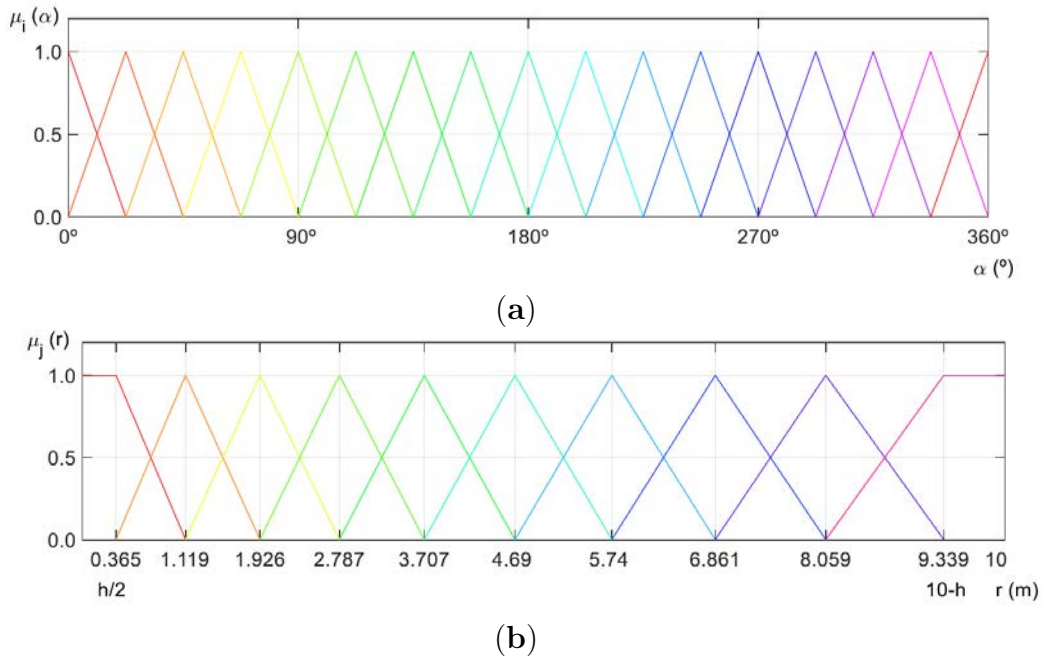


Figura 3.3: Funciones de pertenencia para las variables α (a) y r (b).

La pertenencia combinada para ambas variables $\mu_{i,j}$ se calcula con el operador producto para un punto 3D dado:

$$\mu_{i,j}(\alpha, r) = \mu_i(\alpha) \mu_j(r), \quad (3.7)$$

donde se cumple la siguiente condición:

$$\sum_{\forall i,j} \mu_{i,j}(\alpha, r) = 1. \quad (3.8)$$

Sea m un vector de fila que contenga todas las pertenencias posibles:

$$m(\alpha, r) = (\mu_{1,1}(\alpha, r), \mu_{1,2}(\alpha, r), \dots, \mu_{16,10}(\alpha, r)), \quad (3.9)$$

donde la mayoría de sus 160 elementos son nulos. Los elementos no nulos corresponden a las celdas donde la proyección de un punto 3D cae en el plano $X_p Y_p$ y en sus celdas adyacentes (ver Figura 3.4).

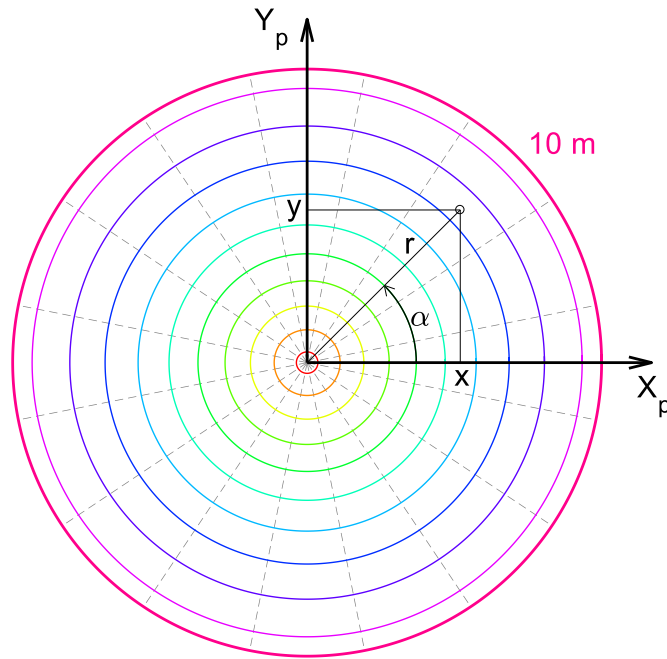


Figura 3.4: Celdas definidas por las funciones de pertenencia borrosas.

El uso de consecuentes de Sugeno de orden cero para la regla que relaciona el conjunto borroso i para α y el conjunto borroso j para r requiere de solo un parámetro $z_{i,j}$ para el FEM, y otro $v_{i,j}$ para la FRM. Sean θ_z y θ_v los vectores columna que contienen todos los parámetros desconocidos:

$$\theta_z = \begin{pmatrix} z_{1,1} \\ z_{1,2} \\ \vdots \\ z_{16,10} \end{pmatrix}, \quad \theta_v = \begin{pmatrix} v_{1,1} \\ v_{1,2} \\ \vdots \\ v_{16,10} \end{pmatrix}. \quad (3.10)$$

Dadas las ecuaciones algebraicas: $z = m(\alpha, r) \theta_z$ y $1 = m(\alpha, r) \theta_v$ para el FEM y la FRM, respectivamente, el método de mínimos cuadrados encuentra los parámetros θ_z y θ_v , respectivamente, que minimizan la suma de los cuadrados de los residuos: $z - m(\alpha, r) \theta_z$ y $1 - m(\alpha, r) \theta_v$, respectivamente, realizado sobre cada punto 3D en coordenadas cilíndricas (α, r, z) .

Al principio, θ_z y θ_v se inicializan con ceros, con la excepción de los parámetros asociados con el primer conjunto difuso para r , es decir, aquellos numerados $j = 1$. Estas zonas pueden no contener ningún punto 3D debido a la zona ciega del escáner láser 3D. Por lo tanto, los parámetros $z_{i,1}$ se completan con las alturas deducidas de los ángulos de cabeceo y alabeo del robot móvil en $X_p Y_p Z_p$. Además, para todo i , $v_{i,1} = 1$ para indicar que estas zonas en el FEM son confiables porque Andábata ya ha pasado sobre en ellas.

Para obtener θ_z , las ecuaciones recursivas de mínimos cuadrados son:

$$L_1 = \frac{P_1 m^T}{1 + m^T P_1 m}, \quad \theta_z = \theta_z + L_1 (z - m \theta_z), \quad P_1 = P_1 - L_1 m P_1, \quad (3.11)$$

donde L_1 es un vector de ganancia y P_1 es la matriz de covarianza, que se elige inicialmente como la matriz de identidad multiplicada por un gran número positivo para indicar que no hay confianza en los parámetros iniciales de $z_{i,j}$. Del mismo modo, para calcular θ_v , las ecuaciones recursivas son:

$$L_2 = \frac{P_2 m^T}{1 + m^T P_2 m}, \quad \theta_v = \theta_v + L_2 (1 - m \theta_v), \quad P_2 = P_2 - L_2 m P_2, \quad (3.12)$$

donde L_2 y P_2 son el vector de ganancia y la matriz de covarianza, respectivamente. Inicialmente, P_2 se elige como la matriz de identidad multiplicada por un pequeño

número positivo para indicar que existe confianza en los parámetros iniciales $v_{i,j}$ (lo que significa que no hay confiabilidad en el FEM inicial).

Una vez identificados los parámetros difusos en θ_z y θ_v , es posible estimar por cada (α, r) la altura en el FEM: $\hat{z} = m(\alpha, r)\theta_z$ y el valor de confiabilidad en la FRM: $\hat{v} = m(\alpha, r)\theta_v$.

La Figura 3.5 muestra el FEM y la FRM obtenidos de la nube de puntos 3D filtrada construida a partir de la Figura 3.2. Puede observarse cómo los árboles aparecen como dos salientes del FEM destacados como zonas moradas y como el entorno cercano al centro del barrido tiene una mayor confiabilidad debido la mayor densidad de puntos en la zona alrededor al robot.

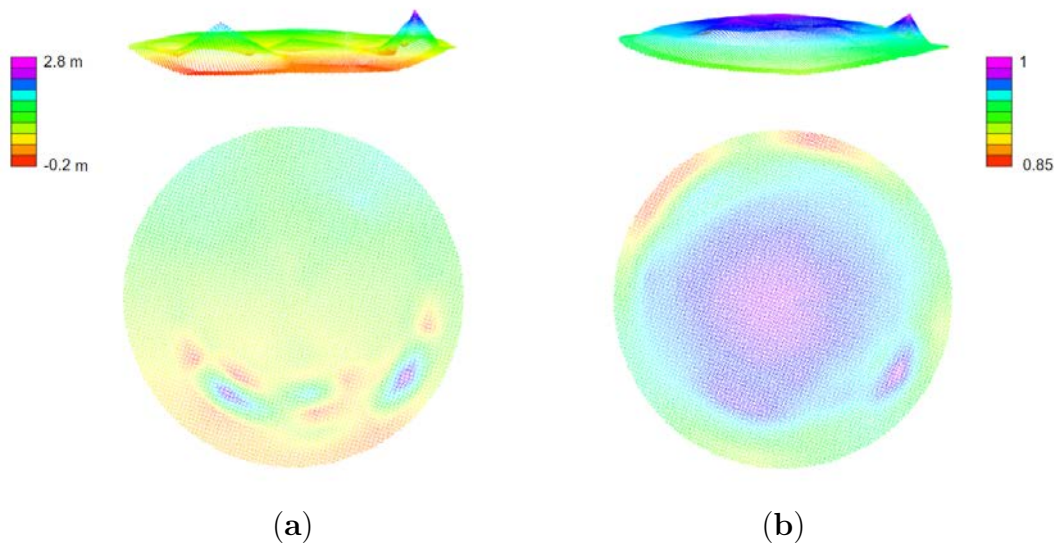


Figura 3.5: Vistas laterales (arriba) y aéreas (abajo) del FEM (a) y la FRM (b) asociadas con la nube de puntos 3D filtrada de la Figura 3.2.

3.3.1. Cálculo multiproceso de FEM y FRM

La computadora a bordo de Andábata tiene un procesador con cuatro núcleos físicos, pero emplea procesos múltiples simultáneos (*multithreading*) para dividir cada núcleo en dos unidades virtuales, lo que resulta en ocho núcleos virtuales independientes. A continuación, se discute cómo emplear esta capacidad de procesamiento

múltiple con ROS para mejorar la tiempo de cálculo y la tasa de creación de los FEM y FRM.

Con una resolución horizontal de 2.4° para el escáner láser 3D, se obtiene una nube de puntos 3D filtrada cada 3.75 s, con 60,000 puntos láser aproximadamente (este número depende en gran medida de visibilidad del cielo, que reduce el número de puntos válidos). Pero el cálculo del FEM y su FRM (realizado por el nodo `Map`) lleva mucho más tiempo que el de adquisición de una nube de puntos 3D. Este tiempo depende del número total de puntos 3D y de la carga de trabajo que tenga el procesador.

Para aumentar la tasa de salida, el cálculo del FEM y el FRM se ha paralelizado en dos hilos de ejecución para el nodo `Map` separando el cálculo de (3.11) al de (3.12). Para mejorar aún más, el nodo `Map` se puede iniciar dos o más veces trabajando en diferentes nubes de puntos 3D.

Para decidir qué nubes de puntos 3D se procesarán y cuáles se descartarán, se ha desarrollado el nodo `Cloud_selector` (ver Figura 2.9). En el instante que está disponible un nodo `Map`, es necesario decidir si procesar la nube de puntos 3D recibida anteriormente o esperar a la recepción de la nube de puntos 3D actual. Además, en caso de usar múltiples nodos `Map`, también puede ser interesante esperar futuros barridos 3D para evitar el procesamiento de barridos consecutivos (introduciendo uno o más huecos).

Las Figuras 3.6 y 3.7 muestran la temporización con las tasas de salida de las nubes de puntos 3D, FEM y FRM usando los barridos 3D anteriores y actuales, respectivamente. En estas Figuras, el nodo `cloud_selector` ha empleado diferentes estrategias de selección. Las nubes de puntos 3D procesadas se colorean y se emplea el mismo color para indicar cuándo se han obtenido sus FEM y FRM (marca superior) de una nube de puntos 3D determinada (marca inferior) correspondientes. Las nubes de puntos 3D descartadas no están coloreadas.

La Tabla 3.1 compara la edad del mapa (es decir, el tiempo empleado para producir un FEM y su FRM a partir de una nube de puntos 3D filtrada), el inter-

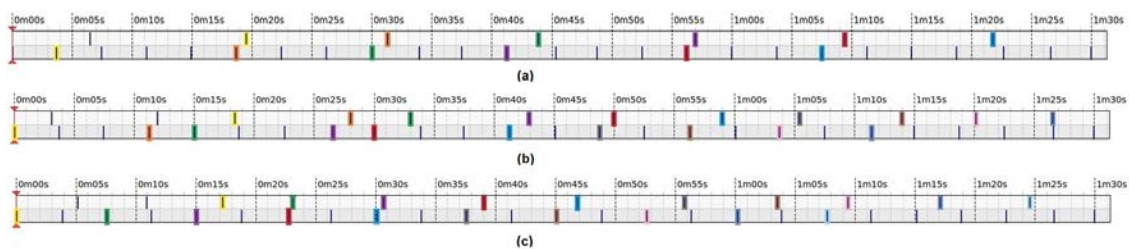


Figura 3.6: Tasa de salida para nubes de puntos 3D, FEM y FRM usando la última nube de puntos 3D recibida con 1 nodo Map (a), 2 nodos Map sin espaciado forzado (b) y 2 nodos Map con un espaciado forzado (c).

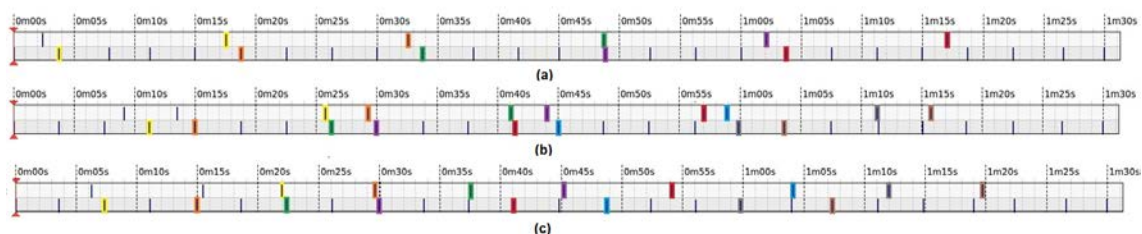


Figura 3.7: Tasa de salida para nubes de puntos 3D, FEM y FRM usando la nube de puntos 3D actual con 1 nodo Map (a), 2 nodos Map sin espaciado forzado (b) y 2 nodos Map con un espaciado forzado (c)

valo entre mapas de elevación borrosos consecutivos y sus respectivas desviaciones estándar para diferentes estrategias de selección. Cuando se usa la nube de puntos 3D actual, la antigüedad del mapa es menor, pero el intervalo entre mapas es mayor que cuando se usa la nube de puntos 3D anterior. La mejor edad del mapa se puede obtener procesando la nube de puntos 3D actual con un solo nodo Map. Por otro lado, el intervalo entre mapas se puede mejorar mediante el uso de más nodos Map. La estrategia ideal sería la que minimiza la edad del mapa y el intervalo entre mapas. Se puede lograr un compromiso entre ambos objetivos usando dos nodos Map y un espaciado forzado con la nube de puntos anterior. En consecuencia, esta última es la estrategia de selección adoptada para la navegación autónoma de Andábata.

Nube de puntos 3D	Nodos Map	Espaciado forzado	Edad del mapa (s)	Desviación estándar (s)	Intervalo (s)	Desviación estándar (s)	
previo	1	0	14.20	1.31	12.56	0.48	
	2	0	18.21	1.90	8.27	5.76	
		1	15.30	1.08	7.45	1.37	
		2	15.73	1.62	11.37	0.50	
	3	0	27.34	3.41	6.60	6.09	
		1	20.21	0.45	7.55	0.77	
		2	23.95	2.11	11.13	2.30	
	actual	1	0	13.81	0.68	14.91	0.98
		2	0	14.83	1.79	8.36	4.50
1			15.21	1.33	8.13	2.20	
2			12.06	1.58	11.70	2.40	
3		0	28.44	4.16	6.88	2.65	
		1	20.89	1.24	7.70	0.98	
		2	23.18	0.72	11.35	0.59	

Tabla 3.1: Comparación entre diferentes estrategias multiproceso.

3.4. Selección de la dirección de movimiento

Cada vez que un FEM y su FRM son entregados por un nodo Map, el nodo Dir_selection calcula una nueva dirección de movimiento (ver Figura 2.9). Para ello, se analizan un total de 32 direcciones posibles d_i desde la proyección del actual sistema de referencia del robot XYZ sobre el plano X_pY_p (ver Figura 3.8). La dirección elegida d_j será la que minimice la siguiente función de coste:

$$J(d_i) = \frac{|\Delta_i| + g |\delta_i|}{D_i}, \quad (3.13)$$

donde Δ_i es la diferencia entre d_i y la dirección hacia el punto objetivo, δ_i es la diferencia angular entre d_i y la proyección del eje X , $g = 0,02$ es un factor que penaliza cambios de dirección, y D_i es la distancia máxima que el robot podría recorrer sobre el FEM siguiendo d_i (ver Figura 3.8).

Para calcular D_i , inicialmente se fija su valor a cero, y se evalúan varios puntos (x, y) en el FEM y FRM a lo largo de d_i con pasos de 0.1m comenzando desde la posición actual del robot. Siempre que el punto analizado tenga un valor de

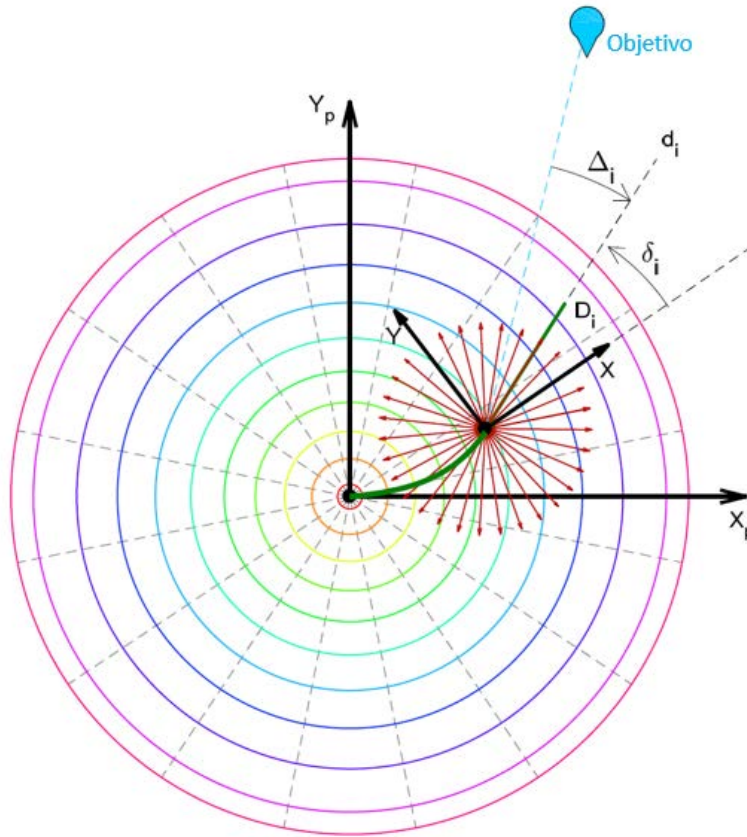


Figura 3.8: Ilustración de la selección de dirección sobre el FEM.

confiabilidad en el FRM menor que 0,75, o una magnitud de gradiente en el FEM mayor a 0,35, o se alcancen los bordes del FEM, la distancia D_i no se incrementará más.

El gradiente del FEM ∇H se utiliza para detectar áreas no transitables [37]. La magnitud del gradiente en un punto con coordenadas cilíndricas (α, r) en el FEM se puede calcular como:

$$|\nabla H(\alpha, r)| = \sqrt{\left(\frac{\partial H(\alpha, r)}{\partial x}\right)^2 + \left(\frac{\partial H(\alpha, r)}{\partial y}\right)^2}, \quad (3.14)$$

donde:

$$\frac{\partial H(\alpha, r)}{\partial x} = \sum_{\forall i, j} \left[z_{i, j} \left(\mu_i(\alpha) \frac{\partial \mu_j(r)}{\partial x} + \mu_j(r) \frac{\partial \mu_i(\alpha)}{\partial x} \right) \right], \quad (3.15)$$

$$\frac{\partial H(\alpha, r)}{\partial y} = \sum_{\forall i, j} \left[z_{i, j} \left(\mu_i(\alpha) \frac{\partial \mu_j(r)}{\partial y} + \mu_j(r) \frac{\partial \mu_i(\alpha)}{\partial y} \right) \right], \quad (3.16)$$

y:

$$\frac{\partial \mu_i(\alpha)}{\partial x} = \begin{cases} \frac{y}{r^2} \frac{1}{\alpha^{i-1} - \alpha^i} & \text{si } \alpha^{i-1} \leq \alpha < \alpha^i, \\ \frac{y}{r^2} \frac{1}{\alpha^{i+1} - \alpha^i} & \text{si } \alpha^i \leq \alpha < \alpha^{i+1}, \\ 0 & \text{en otro caso,} \end{cases} \quad \frac{\partial \mu_j(r)}{\partial x} = \begin{cases} \frac{x}{r} \frac{1}{r^j - r^{j-1}} & \text{si } r^{j-1} \leq r < r^j, \\ \frac{x}{r} \frac{1}{r^j - r^{j+1}} & \text{si } r^j \leq r < r^{j+1}, \\ 0 & \text{en otro caso,} \end{cases} \quad (3.17)$$

$$\frac{\partial \mu_i(\alpha)}{\partial y} = \begin{cases} \frac{x}{r^2} \frac{1}{\alpha^i - \alpha^{i-1}} & \text{si } \alpha^{i-1} \leq \alpha < \alpha^i, \\ \frac{x}{r^2} \frac{1}{\alpha^i - \alpha^{i+1}} & \text{si } \alpha^i \leq \alpha < \alpha^{i+1}, \\ 0 & \text{en otro caso,} \end{cases} \quad \frac{\partial \mu_j(r)}{\partial y} = \begin{cases} \frac{y}{r} \frac{1}{r^j - r^{j-1}} & \text{si } r^{j-1} \leq r < r^j, \\ \frac{y}{r} \frac{1}{r^j - r^{j+1}} & \text{si } r^j \leq r < r^{j+1}, \\ 0 & \text{en otro caso.} \end{cases} \quad (3.18)$$

3.5. Resultados experimentales

El rendimiento del sistema de navegación anteriormente propuesto se ha probado en un parque urbano con farolas, bancos de piedra y diversos árboles (ver Figura 3.9). El parque contiene pendientes suaves y diferentes terrenos como césped y caminos de arena.

El experimento que se muestra en la Figura 3.10 consiste en un misión desde un punto de inicio (cruz blanca) para alcanzar dos objetivos GPS (puntos azules), que se completan cuando el robot está a una distancia menor de 10 m (círculos rojos). El camino seguido por Andábata se muestra con puntos amarillos que representan las posiciones GPS obtenidas por el teléfono inteligente de a bordo. La velocidad longitudinal del robot se estableció en $v^{sp} = 0.34 \text{ m s}^{-1}$ y el viaje duró 540 s, aproximadamente. La Figura 3.10 se ha generado con la ayuda de *Google Earth* [89].

Los cambios en el ángulo de guiñada, obtenido con el magnetómetro del teléfono



Figura 3.9: Andábata navegando en un parque urbano.

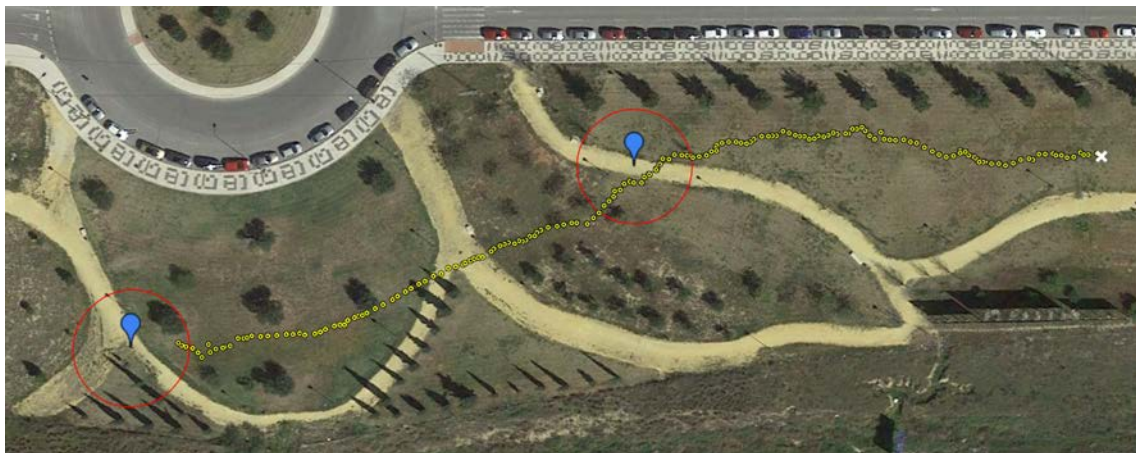


Figura 3.10: Posiciones GPS de Andábata durante la navegación (vista superior).

a bordo, se pueden observar en la Figura 3.11. Se producen cambios notables cuando el robot cambia entre objetivos. Este efecto puede observarse cuando comienza la misión y entre el primer y el segundo punto objetivo al principio y a mitad de la trayectoria, respectivamente. Una vez que se obtiene la dirección del objetivo, se observan algunas correcciones para cruzar por áreas transitables.

La Figura 3.12 muestra un FEM local y su FRM generados durante la navegación autónoma. Se han superpuesto pequeñas flechas negras para representar las posiciones sucesivas del robot móvil en coordenadas locales (proporcionadas por el

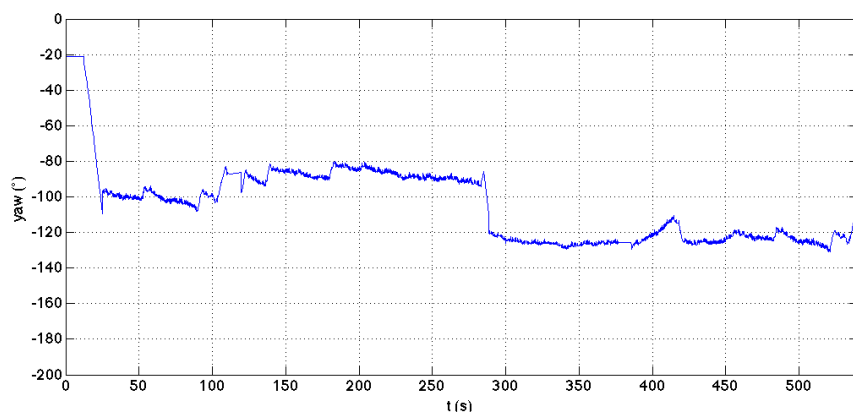


Figura 3.11: Variaciones de guiñada (*yaw*) durante el experimento.

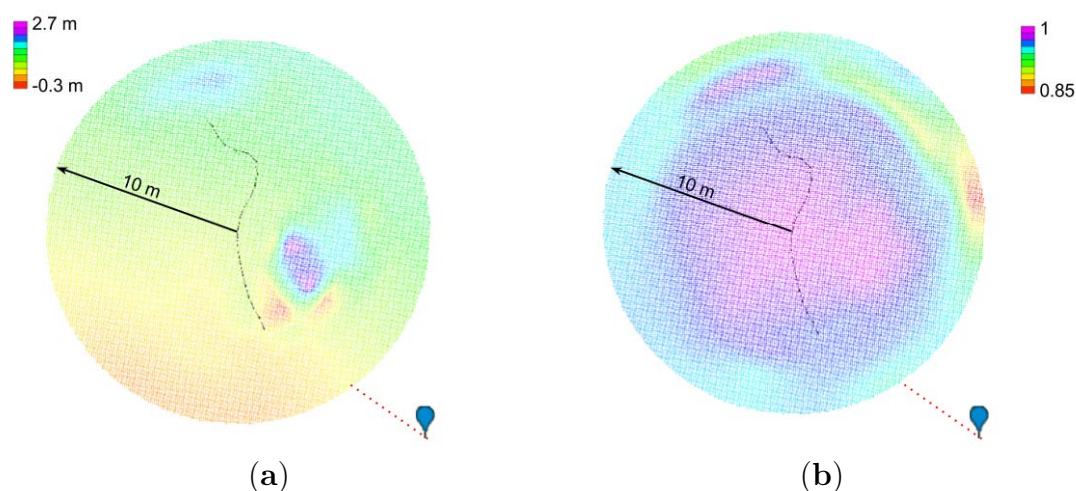


Figura 3.12: Vista superior de la ruta local seguida por Andábata (flechas negras) sobre un FEM (a) y su correspondiente FRM (b).

nodo `Local_positioning`). Se puede observar cómo se evita un obstáculo cercano mientras se mueve hacia el punto objetivo.

3.6. Conclusiones

En este Capítulo se ha descrito el sistema de navegación en exteriores del robot móvil Andábata a bajas velocidades con mapas de elevación borrosos (FEM). Algunos barridos 3D nivelados obtenidos en movimiento se emplean para producir un FEM y su máscara de confiabilidad borrosa (FRM) de los alrededores, que se proce-

san para elegir la mejor dirección de movimiento para alcanzar un punto de objetivo distante mientras se cruzan áreas transitables. Todo ello se ha implementado con diferentes nodos ROS.

Es la primera vez que un robot móvil navega con FEM obtenidos de barridos láser 3D en exteriores. A diferencia de los DEM convencionales, los FEM proporcionan una representación continua del terreno cercano al robot, lo que permite una mayor resolución para escoger los próximos movimientos.

Sin embargo, la navegación con FEM y FRM exige una gran carga computacional que impide que se procesen todos los barridos 3D disponibles, lo que va en detrimento de la reactividad del robot, especialmente en entorno naturales más complejos. Por ello, en los siguientes capítulos se van a desarrollar clasificadores de transitabilidad del terreno que reduzcan en buena medida el esfuerzo computacional, pudiendo así procesar todos los barridos láser 3D para mejorar la reactividad.

El autor de esta tesis ha participado en el desarrollo del software necesario para la implementación de las distintas técnicas presentadas en Andábata, en la planificación, ejecución y análisis de los experimentos necesarios para probar su correcto funcionamiento y en la redacción del artículo de revista [51].



UNIVERSIDAD
DE MÁLAGA

Capítulo 4

Clasificación de transitabilidad mediante aprendizaje supervisado

4.1. Introducción

La detección del suelo navegable resulta fundamental para la navegación en entornos naturales [90]. Para desarrollar un clasificador que cumpla esta misión es necesario entrenar un modelo predictivo que etiquete cada punto 3D en función de sus características espaciales.

La extracción de qué parte del terreno es navegable es un proceso relacionado con la evaluación de la capacidad de desplazamiento del vehículo, que se realiza comúnmente justo antes de la segmentación de escena [91]. Los procedimientos para evaluar la transitabilidad del terreno se pueden diseñar específicamente [92, 93], pero también se pueden entrenar con datos reales [94, 95] o por medio de datos sintéticos [96, 97]. Este relevante análisis generalmente se realiza con nubes de puntos 3D de los alrededores adquiridas por un sensor a bordo del robot móvil [28, 41]. En lugar de diseñar procedimientos de segmentación específicos para la detección de suelo [98], se pueden entrenar diferentes técnicas de aprendizaje automático con características espaciales calculadas directamente desde la nube de puntos 3D [43] [99].

Algunos de los métodos empleados para determinar la transitabilidad son cadenas de Markov [100], filtros Bayesianos [101], máquinas de vectores soporte o *Support Vector Machine* (SVM) [102], modelado borroso [10], redes neuronales [103] o

aprendizaje supervisado [104]. Para la construcción de modelos predictivos mediante técnicas de aprendizaje supervisado es necesario disponer de conjuntos de datos perfectamente clasificados y etiquetados libres de errores para entrenarlos [105] [106].

El aprendizaje supervisado generalmente emplea puntos etiquetados a mano para obtener modelos predictivos que se pueden aplicar a nuevos datos [107]. De esta manera, la aplicación *Classification Learner* de Matlab se ha empleado para extraer el suelo de nubes de puntos 3D de un conjunto de datos urbano [90]. Asimismo, se ha utilizado el algoritmo SVM para detectar carreteras urbanas y rurales con visión estéreo [95].

Sin embargo, el etiquetado de datos 3D reales punto a punto de vehículos terrestres en entornos naturales requiere un gran esfuerzo y es propenso a errores [108]. Además, no hay repositorios etiquetados con este tipo de datos que reúnan escenas de diferentes entornos naturales. Como alternativa a los datos etiquetados manualmente, el aprendizaje por demostración con nubes de puntos 3D adquiridas por un vehículo teleoperado en zonas transitables puede emplearse mediante un clasificador *Positive Naive Bayes* [28], un proceso Gaussiano [30] o con SVM [94].

Los datos de profundidad sintéticos son interesantes para entrenar la detección de transitabilidad [109]. En este sentido, datos virtuales de LIDAR generados con Matlab se han empleado para construir una red neuronal que clasifica el terreno transitable de superficies planetarias [110]. De manera similar, en [96] se ha entrenado una red neuronal convolucional para distinguir parches transitables de mapas de elevación obtenidos con el simulador robótico Gazebo [111].

La Figura 4.1 explica las fases necesarias para la generación de modelos mediante aprendizaje supervisado. Una vez se cuenta con datos para el entrenamiento etiquetados, es necesario realizarles un preprocesado, que puede comprender algunas de las fases que aparecen representadas. Posteriormente, se aplican algoritmos de regresión o clasificación para entrenar los modelos.

En la Figura 4.2 se explica el proceso para realizar una predicción de datos a partir de un modelo previamente entrenado. De nuevo, el primer paso es realizar un

preprocesamiento previo. Con los datos preprocesados, se aplica el modelo entrenado para obtener la predicción.

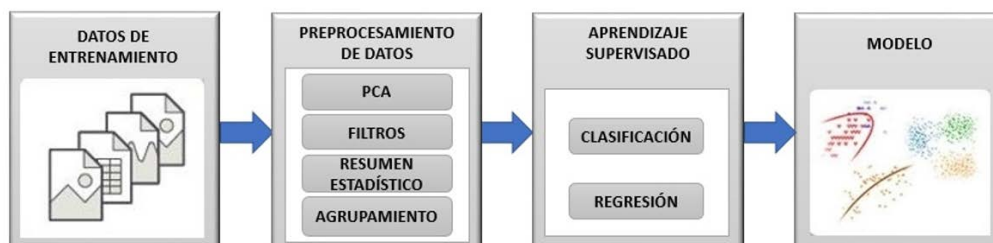


Figura 4.1: Fases para la generación de modelos entrenados mediante aprendizaje supervisado [112].

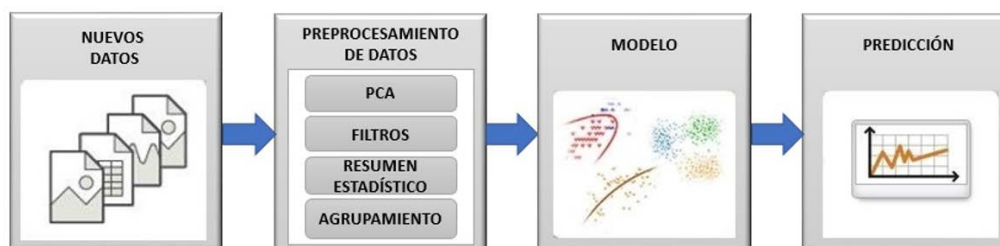


Figura 4.2: Fases para la clasificación de datos a partir de modelos entrenados [112].

En este capítulo, las nubes de puntos 3D sintéticas generadas en el entorno Gazebo son etiquetadas automáticamente sin errores [108] y se emplean para el entrenamiento y la validación de clasificadores de transitabilidad.

El capítulo está organizado de la siguiente manera. La siguiente sección resume el procedimiento para etiquetar individualmente los puntos procedentes de barridos láser 3D sintéticos según su transitabilidad. La sección 4.3 presenta el entrenamiento de varios clasificadores con estas nubes de puntos etiquetadas. Luego, las secciones 4.4 y 4.5 contienen resultados de validación para datos simulados y reales, respectivamente. El capítulo termina con la sección de conclusiones.

4.2. Obtención de barridos láser 3D sintéticos etiquetados

En esta sección se emplea el simulador robótico Gazebo [111] para obtener nubes de puntos realistas de terreno irregular generadas por LIDAR virtuales [36, 109].

La Figura 4.3 muestra el esquema general de la aplicación desarrollada, la cual involucra al simulador robótico Gazebo, al entorno Matlab y a ROS [108]. Gazebo se emplea para construir un modelo de entorno natural y para simular las medidas del telémetro láser 2D. Matlab se utiliza para proporcionar una rotación al sensor 2D y para componer la nube de puntos 3D etiquetada. ROS sirve como interfaz entre Gazebo y Matlab.

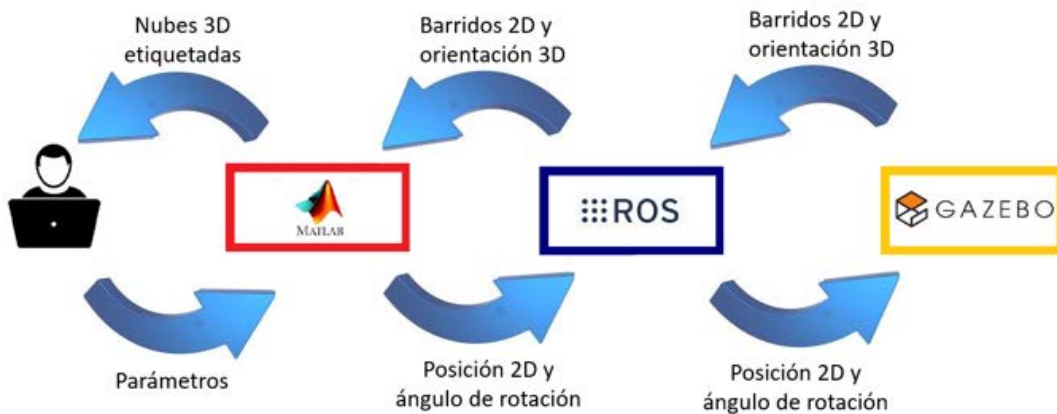


Figura 4.3: Diagrama de la aplicación desarrollada [108].

La Figura 4.4 muestra una vista general del entorno natural generado con Gazebo, cuyas dimensiones máximas son 150 m de largo, 150 m de ancho y 20 m de alto [108]. Contiene elementos naturales tales como terreno irregular, hierba, arbustos, rocas, árboles y agua. Sin embargo, también tiene elementos artificiales como mesas, bancos, cercas, líneas eléctricas y pavimento. Se pueden distinguir cinco zonas diferentes: colinas (A), una cueva (B), un bosque (C), un lago (D) y un parque (E).

En la Figura 4.5 puede verse el interfaz empleado en Matlab. El usuario indica la posición desde la cual se adquirirá el barrido 3D en el suelo sobre una interfaz que incluye una vista aérea del entorno sintético.

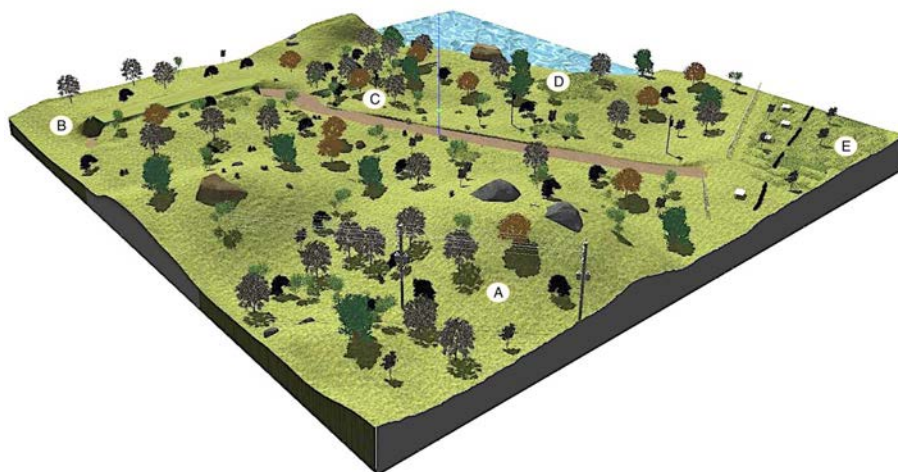


Figura 4.4: Vista general del entorno natural construido con Gazebo. Cinco zonas diferentes están marcadas con letras mayúsculas.

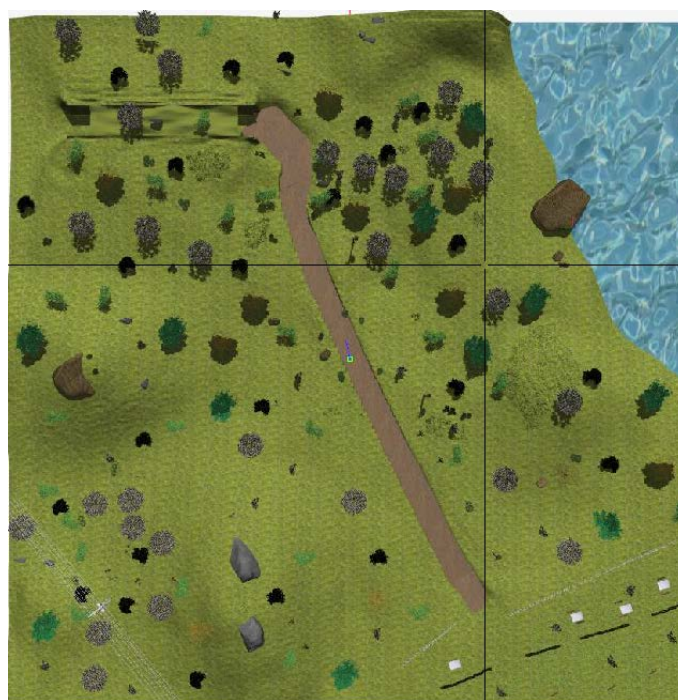


Figura 4.5: Vista aérea del entorno natural generado en Gazebo. La intersección entre las dos líneas negras perpendiculares indica la posición desde la que se adquirirá el barrido 3D.

Además de modelar el entorno natural, Gazebo simula las mediciones de rango e intensidad del sensor láser 2D Hokuyo [111], con el que está construido el sensor 3D de Andábata. Esta simulación incluye medidas con ruido y el escáner láser 2D

obtiene estimaciones de intensidad que dependen de los valores asignados a cada elemento del entorno virtual. Precisamente, estos datos permiten etiquetar cada punto 3D dentro de su correspondiente clase.

La Tabla 4.1 indica las etiquetas de color y los valores de reflectividad asignados a cada elemento del entorno virtual. Para emular un rayo láser que es desviado por la superficie del agua, los rangos con reflectividad del agua se eliminarán de la nube de puntos 3D.

Material	Etiqueta de color	Reflectividad
Agua		120
Terreno o suelo	Marrón 	100
Tronco o ramas	Magenta 	90
Copa de árbol	Oliva 	80
Mesa	Amarillo 	75
Arbusto	Verde 	70
Valla	Negro 	60
Banco	Rosa 	55
Hierba baja	Naranja 	50
Roca	Gris 	40
Hierba altas	Rojo 	30
Cable	Azul 	20
Poste	Cián 	15

Tabla 4.1: Etiquetas de color y reflectividad asignadas a cada elemento.

En lugar de emplear un modelo completo para Andábata, se ha desarrollado una estructura más simple para llevar su escáner láser 3D. La estructura contiene dos brazos para los ejes X e Y locales y una columna para el eje Z que apunta hacia arriba (ver Figura 4.6 a). El escáner láser 2D se une al extremo superior de la columna con su centro óptico alineado con el eje Z. Al aplicar ángulos de rotación de 180° a la columna se obtiene un barrido 3D completo [108]. Luego, Matlab une los sucesivos barridos 2D a partir de la simulación Gazebo del escáner láser 2D con diferentes ángulos de rotación y los combina en una nube de puntos 3D completa.

En los cuatro extremos de los brazos hay telémetros láser unidimensionales que apuntan hacia abajo. Estos sensores emplean el mismo modelo de Hokuyo, pero

con el número de rayos láser restringido a uno. Las distancias proporcionadas por los cuatro escáneres unidimensionales se emplean para colocar la estructura con ángulos de alabeo y cabeceo similares a los que tendría Andábata y para ajustar la altura h sobre el suelo del sensor 3D (ver Figura 4.6 b). Las cuatro medidas de los láser unidimensionales se utilizan para comprobar que la estructura está ubicada íntegramente en el suelo.

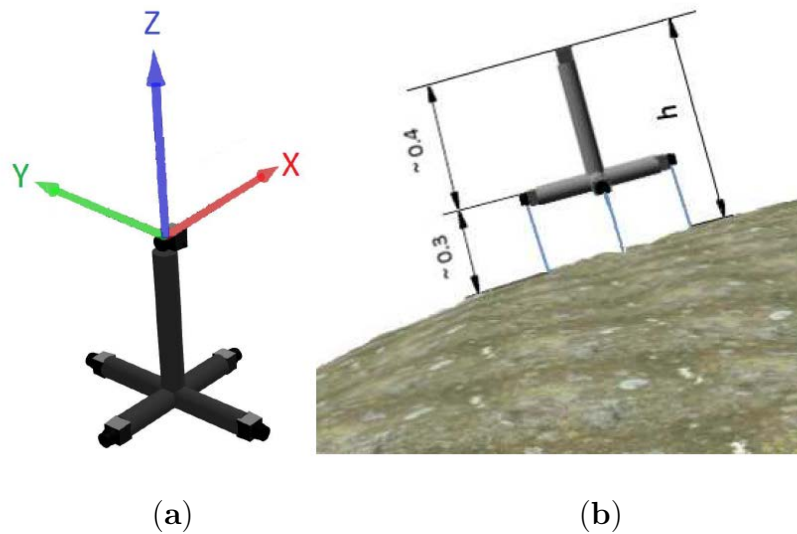


Figura 4.6: La estructura para situar el telémetro láser 3D (a) y su colocación sobre el terreno (b).

En la Figura 4.7 se puede observar la representación de un barrido 2D simulado, una vez que la estructura se ha colocado correctamente en el suelo del entorno virtual.

La Figura 4.8 muestra el grafo computacional ROS, que incluye dos nodos y siete *topics*. El nodo Gazebo se generó utilizando los paquetes *Gazebo_ros* [113], mientras que el nodo Matlab se creó con la librería *Robotics System Toolbox* [114].

Los *topics* del `/laser1D1` al `/laser1D4` contienen los rangos e intensidades de cada barrido láser unidimensional de cada sensor de los situados en los brazos de la estructura. De manera similar, el *topic* `/laser2D` contiene barridos 2D completos, cada uno compuesto por 1,080 mediciones de rango y de reflectividad.

Gazebo envía mensajes con la orientación global 3D actual del sensor láser 2D

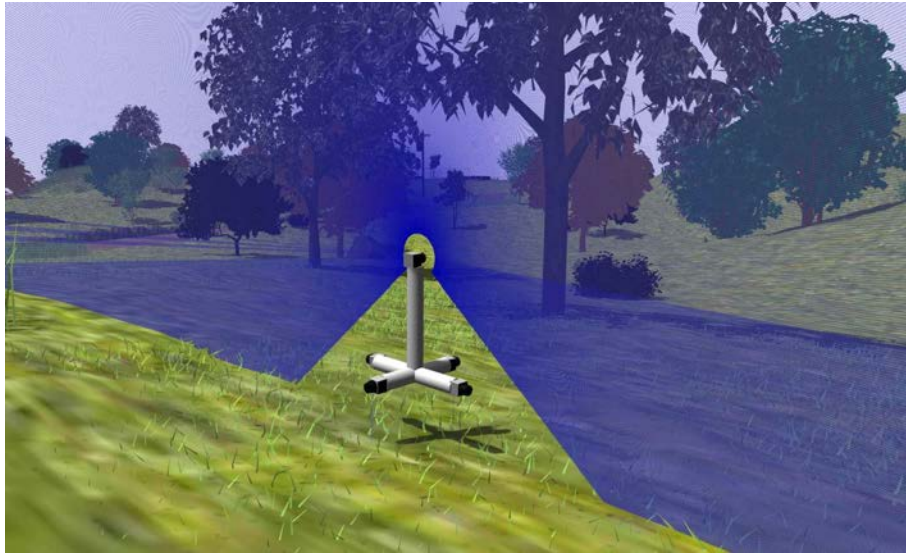


Figura 4.7: Simulación de un barrido 2D en el entorno virtual.

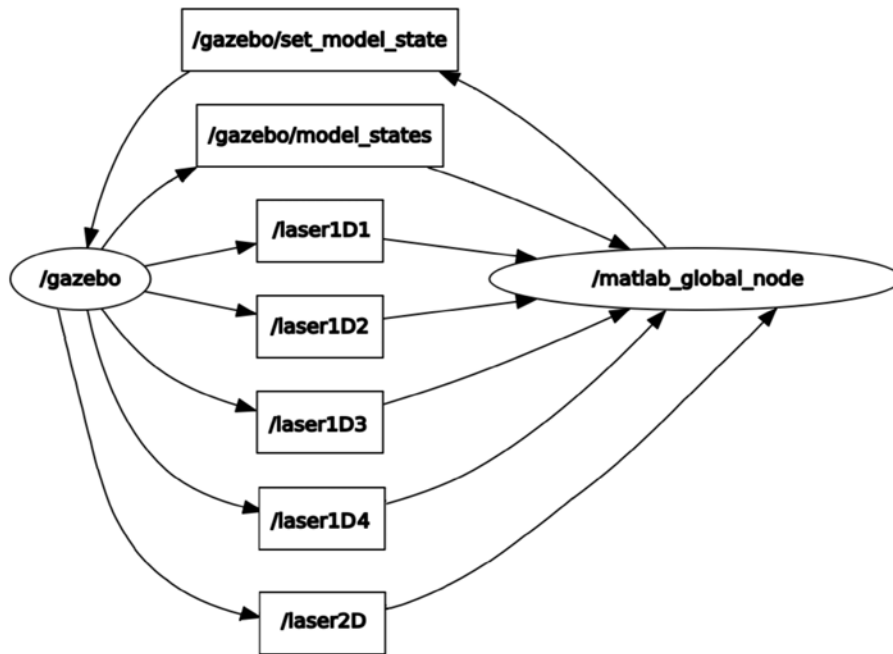


Figura 4.8: Nodos ROS (elipses) y *topics* (rectángulos) empleados en la simulación

a través del *topic* `/gazebo/model_states`. Por otro lado, Matlab emplea el *topic* `/gazebo/set_model_states` para colocar la estructura del sensor 3D en el suelo y para proporcionar los ángulos de rotación sucesivos para el sensor 2D.

Teniendo en cuenta los ángulos de alabeo y cabeceo en el terreno [108], toda

la nube de puntos se nivela para operar tal y como Andábata las construye [51]. Las mediciones de intensidad se utilizan para etiquetar cada punto 3D de forma independiente asignando diferentes valores de reflectividad a cada elemento natural o artificial. Se han obtenido tres barridos láser sintéticos con una resolución horizontal de un grado en cada zona del entorno natural colocando Andábata en diferentes puntos del suelo. En total, quince barridos láser 3D etiquetados sin errores.

Posteriormente, estos barridos sintéticos se binarizan de la siguiente manera; aquellos puntos que pertenecen al suelo, al pavimento o a la vegetación baja (con una altura máxima de 5 cm) se etiquetan como transitables y el resto como no transitables. Además, la inclinación de cada punto transitable se obtiene calculando la normal al plano local formado por los veinte vecinos transitables más cercanos. Finalmente, todos los puntos transitables con una pendiente mayor que 20° (inclinación máxima que Andábata puede navegar) se reetiquetan como no transitables.

La Figura 4.9 resume las diferentes etapas requeridas en el etiquetado de los puntos:

- (a) Representa una vista de la zona de colinas (A) construida con Gazebo [111].
- (b) Muestra un barrido 3D simulado. El círculo vacío en el suelo en el centro del barrido láser corresponde al área ciega del sensor 3D.
- (c) Representa el barrido anterior una vez que ha sido nivelado y sus puntos 3D etiquetados con diferentes colores según sus valores de intensidad [108].
- (d) Muestra los puntos transitables del barrido láser en color verde y el resto en rojo. En este caso, los puntos no transitables se corresponden con árboles, arbustos, una línea eléctrica y terreno muy inclinado.

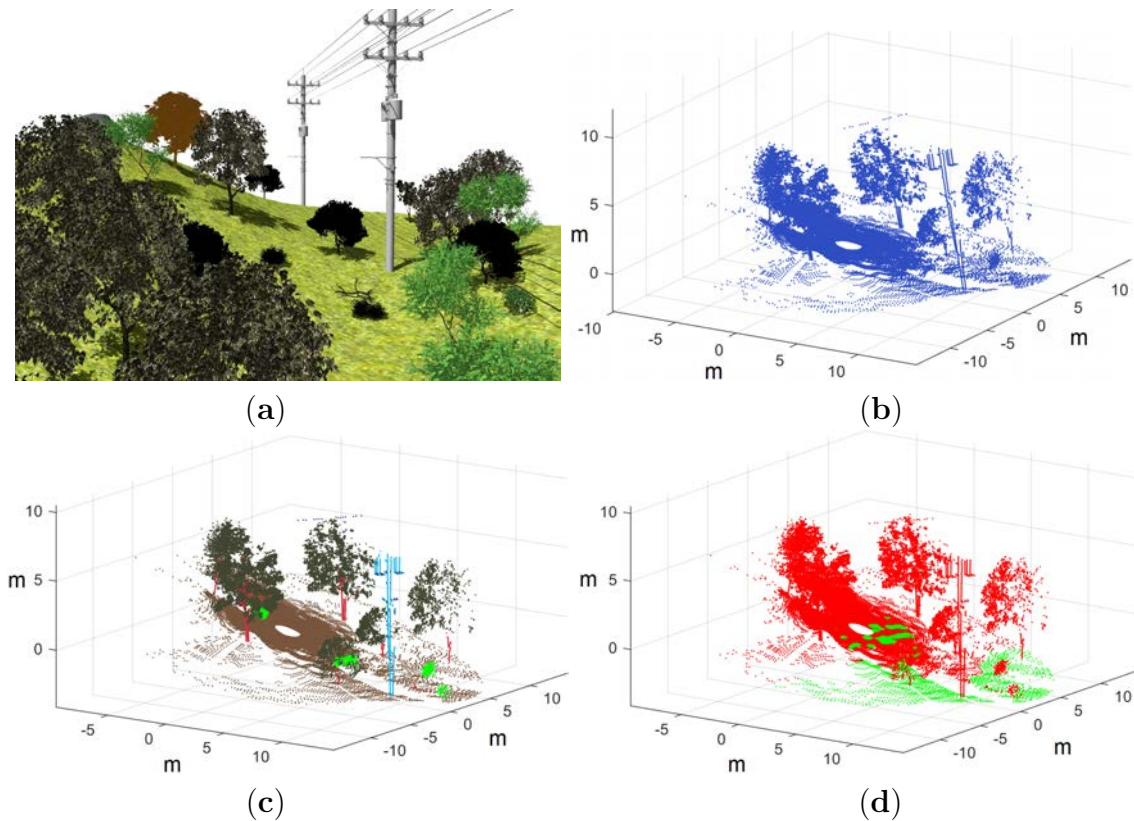


Figura 4.9: Una vista de las colinas generadas con Gazebo (a), un barrido 3D sintético tomado de esta zona (b), su nube de puntos nivelada y etiquetada con intensidad (c) y los datos 3D etiquetados según su transitabilidad (d).

4.3. Entrenamiento de clasificadores de transitabilidad del terreno

El primer paso es extraer las características espaciales o *features* apropiadas para la clasificación de transitabilidad. Luego, se pueden entrenar diferentes técnicas de aprendizaje supervisado.

4.3.1. Cálculo de características espaciales

Las características espaciales se extraen para cada punto 3D a partir de su vecindario, que comprende todos los puntos contenidos en un radio de proximidad fijo de 0.3 m. Los puntos Cartesianos con menos de cinco vecinos se descartan del cálculo de

características para garantizar un mínimo de información espacial. La búsqueda de vecinos más cercanos para cada nube de puntos nivelada se acelera mediante el uso de una estructura de datos en árbol 3D, que se construye con la función de Python *KDTree* de la librería *Scikit-learn* [115].

La distribución espacial se puede estudiar a través del análisis de componentes principales o *Principal Components Analysis* (PCA) [116]. PCA emplea la matriz $n \times 3$ denominada P_i que contiene las coordenadas Cartesianas en un sistema de referencia *XYZ* de un punto p_i y de todos sus vecinos. Entonces, la matriz S_i de covarianza simétrica 3×3 asociada a p_i se calcula como:

$$S_i = \frac{(P_i - \overline{P}_i)^T (P_i - \overline{P}_i)}{n}, \quad (4.1)$$

donde T representa la operación de transposición y \overline{P}_i es la matriz $n \times 3$ que contiene el valor medio de P_i : $(\bar{x}_i, \bar{y}_i, \bar{z}_i)$ en cada una de sus n filas. Los autovalores λ asociados a S_i pueden calcularse con la ecuación característica:

$$|S_i - \lambda I| = 0, \quad (4.2)$$

donde I es la matriz identidad 3×3 . Los autovalores son números reales positivos que pueden ser ordenados:

$$0 < \lambda_1 \leq \lambda_2 \leq \lambda_3. \quad (4.3)$$

Los autovectores v asociados con cada autovalor λ son calculados mediante la siguiente ecuación:

$$S_i v = \lambda v. \quad (4.4)$$

Este análisis puede realizarse con las funciones Matlab `cov` y `eig`.

Se emplean tres características $\{f_1, f_2, f_3\}$ que son comúnmente empleadas en la detección de suelo [108]. Para nubes de puntos niveladas, la mínima altura (f_1) es la característica geométrica más relevante para la clasificación del terreno [99]. Además, la orientación vertical (f_2) y la rugosidad (f_3) pueden emplearse para detectar el

suelo transitable [30].

Estas tres características pueden calcularse como:

1. *Mínima altura*: La mínima coordenada Z entre todos los puntos p_j de P_i :

$$f_1(P_i) = \min_{\forall p_j} (z_j). \quad (4.5)$$

2. *Orientación vertical*: La pendiente que se obtiene del autovector v_1 con el menor autovalor λ_1 :

$$f_2(P_i) = \arccos((0, 0, 1) \cdot v_1). \quad (4.6)$$

3. *Rugosidad*: El menor de los autovalores puede emplearse directamente como índice de dispersión espacial:

$$f_3(P_i) = \lambda_1. \quad (4.7)$$

El cálculo del PCA de cada vecindario de puntos se acelera con el compilador Python y la librería *Numba* [117]. Aun así, el tiempo de procesamiento del PCA sigue dependiendo del número de puntos del barrido láser 3D. De esta forma, para un barrido sintético promedio de 76,000 puntos, donde aproximadamente el 3% de los puntos no tiene suficientes vecinos, este preprocesamiento de datos se realiza en 3.16s. Para mejorar este tiempo, aprovechando los cuatro núcleos del procesador de Andábata, se ha probado la librería de Python *multithreading* [118], pero con resultados decepcionantes, por lo cual ha sido descartada.

4.3.2. Aprendizaje supervisado

Se han elegido diferentes algoritmos de clasificación de la librería de Python *Scikit-learn* [115] para predecir la transitabilidad de los puntos 3D utilizando el conjunto de características espaciales anteriormente introducido. Esta librería de apren-

dizaje automático fue diseñada para operar con las librerías numéricas y científicas de Python *NumPy* y *SciPy*, respectivamente [115].

Teniendo en cuenta que no es necesario emplear métodos de clasificación complejos para extraer el suelo con precisión de los barridos 3D [90], se han seleccionado siete técnicas de aprendizaje supervisado relevantes para el entrenamiento: *Decision Tree* (DT), *Gaussian Naive Bayes* (GNB), *K-Nearest Neighbors* (KNN), *Linear Support Vector Machine* (LSVM), *Bagged Decision Tree* (BDT), *Random Forest* (RF) y *Gradient Boosted Tree* (GBT). Los tres últimos son métodos combinados que emplean varios estimadores básicos.

Diez de las quince nubes de puntos 3D generadas con Gazebo están dedicadas exclusivamente al entrenamiento. Estos datos sintéticos libres de errores contienen un total de 743,346 puntos, donde 721,616 cumplen con la restricción mínima de vecindad. Los datos de entrenamiento no están equilibrados porque aproximadamente el 70 % de los puntos pertenecen a la clase transitable. Esto sucede principalmente porque la mayoría de los puntos láser se obtienen del suelo cerca de Andábata.

La Tabla 4.2 muestra los tiempos de entrenamiento requeridos por cada estimador ajustado con sus opciones predeterminadas. Los métodos que requieren más tiempo son LSVM y GBT, mientras que los métodos menos exigentes son GNB y KNN. Es notable la gran brecha de 136 s entre los mejores y los peores tiempos. Sin embargo, estos tiempos no son críticos para la navegación porque el entrenamiento sólo se realiza una vez fuera de línea.

Estimador	Acrónimo	Tiempo (s)
<i>Decision Tree</i>	DT	3.3
<i>Gaussian Naive Bayes</i>	GNB	0.1
<i>K-Nearest Neighbors</i>	KNN	1.1
<i>Linear Support Vector Machine</i>	LSVM	136.3
<i>Bagged Decision Tree</i>	BDT	21.0
<i>Random Forest</i>	RF	8.1
<i>Gradient Boosted Tree</i>	GBT	41.3

Tabla 4.2: Tiempos de entrenamiento para la clasificación de transitabilidad.

4.4. Validación de los clasificadores de transitabilidad

Cinco nubes de puntos 3D sintéticas, una por cada zona del entorno virtual, que no han sido utilizadas durante el entrenamiento, se emplean exclusivamente para fines de validación. Estos datos contienen un total de 397,426 puntos donde 385,959 tienen al menos cinco vecinos y están desequilibrados con aproximadamente el 68 % de puntos en la clase transitable.

Para todos los clasificadores, el tiempo de predicción de un barrido sintético completo es, en promedio, insignificante respecto a su tiempo de preprocesamiento con la excepción del estimador KNN que requiere de 0.2 s.

La Tabla 4.3 contiene los componentes de la matriz de confusión de cada clasificador entrenado, donde TP, FP, TN y FN representan el número de verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos, respectivamente. Verdadero se refiere a puntos clasificados correctamente y falso al opuesto, mientras que positivo se refiere a la clase no transitable y negativo a la clase transitable.

Estimador	TP	TN	FP	FN
<i>Decision Tree</i>	111,912	252,446	10,007	11,594
<i>Gaussian Naive Bayes</i>	89,890	259,351	3102	33,616
<i>K-Nearest Neighbors</i>	111,538	256,915	5538	11,968
<i>Linear Support Vector Machine</i>	97,559	21,8384	44,069	25,947
<i>Bagged Decision Tree</i>	113,111	254,622	7831	10,395
<i>Random Forest</i>	112,547	257,396	5057	10,959
<i>Gradient Boosted Tree</i>	111,335	258,922	3531	12,171

Tabla 4.3: Componentes de las matrices de confusión para los datos sintéticos.

Para comparar el rendimiento de los clasificadores binarios, se consideran cinco índices para datos desequilibrados, calculados por las funciones de la librería *Scikit-learn* [115]: *Precision* (PR), *Recall* (RE) y *F1 Score* son el primer, segundo y tercer índice, respectivamente:

$$PR = \frac{TP}{TP + FP}, \quad RE = \frac{TP}{TP + FN}, \quad F1 = 2 \frac{PR \cdot RE}{PR + RE}. \quad (4.8)$$

El cuarto índice es *Balanced Accuracy* (BA):

$$BA = \frac{1}{2} \left(RE + \frac{TN}{TN + FP} \right). \quad (4.9)$$

Todos los índices anteriores varían entre 0 y 1 para el peor y mejor resultado de clasificación, respectivamente. El último índice es el coeficiente de correlación Matthews o *Mathews Correlation Coeficient* (MCC):

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \quad (4.10)$$

cuyo rango oscila entre -1 a 1 , donde -1 indica una clasificación inversa, 0 una predicción aleatoria y 1 una predicción perfecta.

La Tabla 4.4 incluye los cinco índices de precisión para cada estimador. En general se logra una alta precisión, pero el mejor rendimiento proviene de los clasificadores RF y GBT y el peor resultado de los estimadores GNB y LSVM.

Estimador	PR	RE	F1	BA	MCC
<i>Decision Tree</i>	0.918	0.906	0.912	0.934	0.871
<i>Gaussian Naive Bayes</i>	0.967	0.728	0.830	0.858	0.781
<i>K-Nearest Neighbors</i>	0.953	0.903	0.927	0.941	0.895
<i>Linear Support Vector Machine</i>	0.689	0.790	0.736	0.811	0.602
<i>Bagged Decision Tree</i>	0.935	0.916	0.925	0.943	0.891
<i>Random Forest</i>	0.957	0.911	0.934	0.946	0.904
<i>Gradient Boosted Tree</i>	0.969	0.901	0.934	0.944	0.906

Tabla 4.4: Índices de precisión para los datos sintéticos.

Las Figuras 4.10–4.14 ilustran los resultados de aplicar el clasificador RF a los cinco barridos de validación. Se pueden observar excelentes estimaciones de ambas clases en colores rojo y verde. El color azul representa puntos 3D que no se han clasificado debido a la falta de vecinos. Estos puntos no clasificados generalmente se

encuentran lejos de Andábata, donde disminuye la densidad de puntos y el número de vecinos de cada punto es menor que el umbral definido.

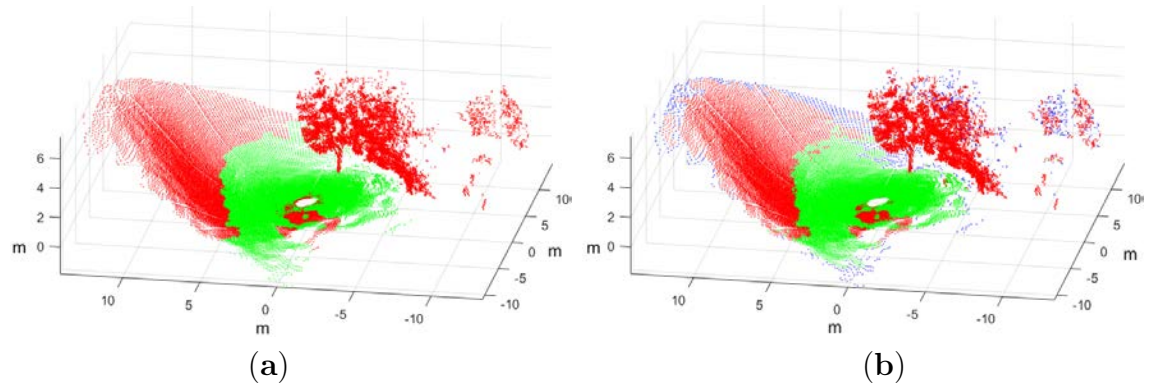


Figura 4.10: Barrido de validación en las colinas (a) y los resultados de la predicción con RF (b).

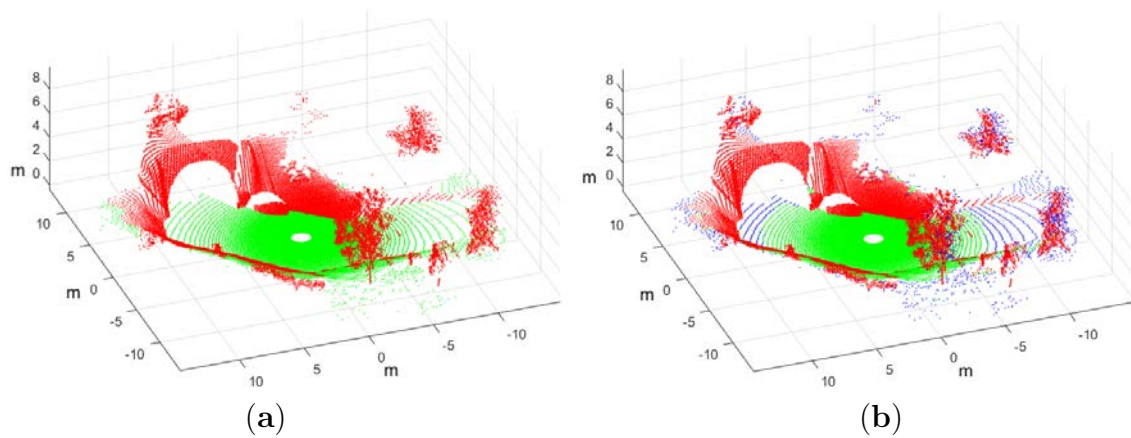


Figura 4.11: Barrido de validación en la entrada de la cueva (a) y los resultados de la predicción con RF (b).

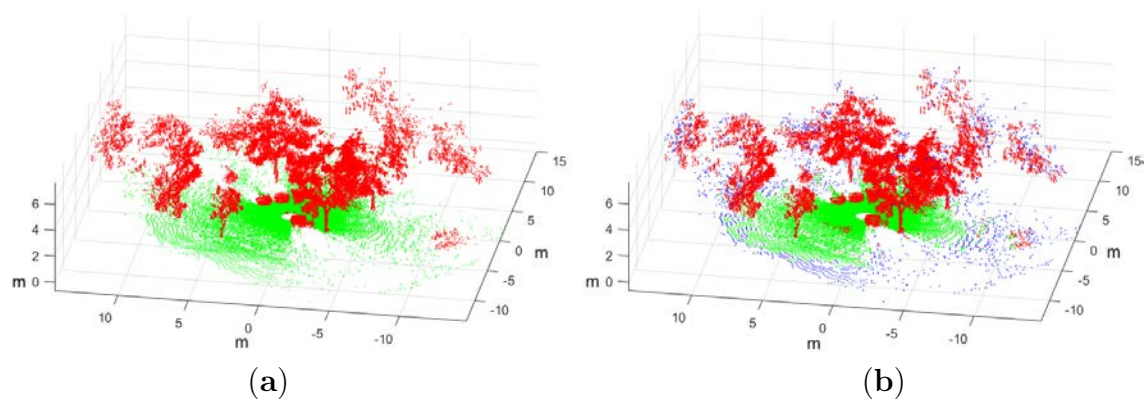


Figura 4.12: Barrido de validación dentro del bosque (a) y los resultados de la predicción con RF (b).

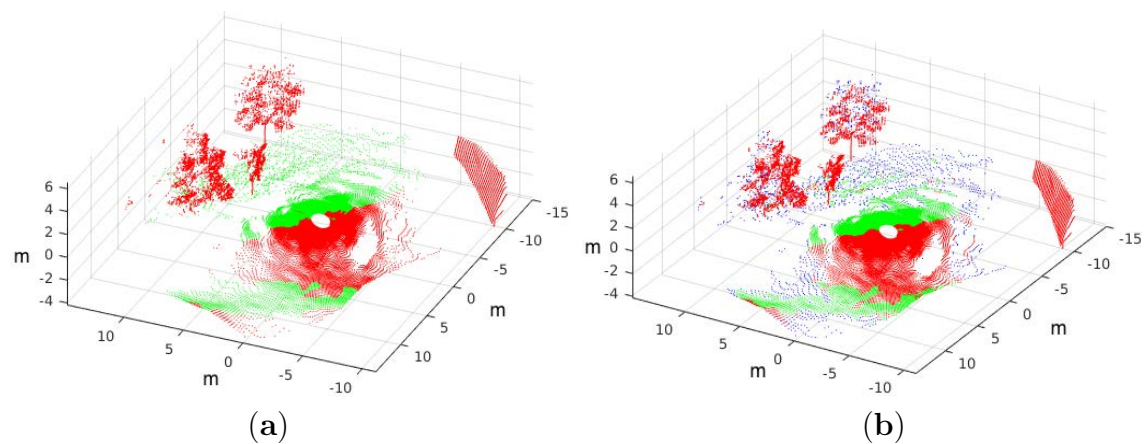


Figura 4.13: Barrido de validación cerca de la orilla del lago (a) y los resultados de la predicción con RF (b).

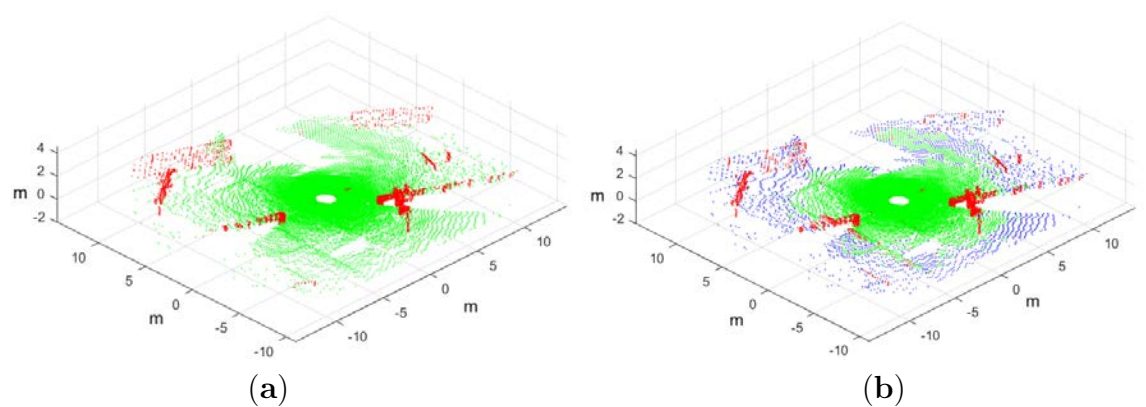


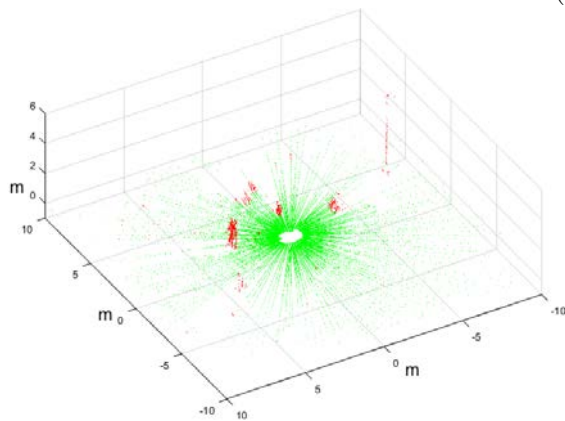
Figura 4.14: Barrido de validación en el parque (a) y los resultados de la predicción con RF (b).

4.5. Experimentos de clasificación con barridos láser 3D reales

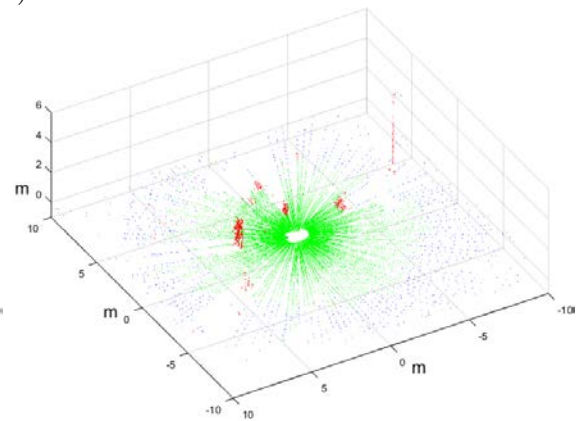
Las Figuras 4.15 a–4.17 a muestran un parque, un camino rural y un paso subterráneo donde Andábata ha sido teleoperado. Para cada escena, se ha obtenido una nube de puntos 3D nivelada con una resolución horizontal de 1.2° sin datos de intensidad en 3.75 s [51]. La visibilidad del cielo determina el número de puntos para cada barrido láser, que van desde 32,795 para el parque hasta 83,183 para el paso subterráneo.



(a)



(b)



(c)

Figura 4.15: Imagen de un parque urbano (a), una nube de puntos 3D real etiquetada manualmente (b) y clasificada con RF (c).

Todos estos datos reales han sido etiquetados manualmente para servir como *ground truth* (ver Figuras 4.15 b–4.17 b). Se puede observar que el área ciega en

el suelo se reduce porque Andábata se estaba moviendo durante la adquisición del barrido. Los datos etiquetados a mano contienen un total de 162,999 puntos, donde 90,074 pertenecen a la clase transitable.

La extracción de características y la predicción de transitabilidad para cada barrido se pueden obtener en 2.5s para todos los estimadores con la excepción de KNN que requiere de 2.7s. En cualquier caso, estos tiempos de clasificación hacen posible procesar cada barrido láser 3D por separado para la navegación autónoma. Solo 3,215 puntos no se han clasificado debido a la falta de cinco vecinos.

La Tabla 4.5 contiene las componentes de las matrices de confusión de cada clasificador. Los índices de precisión equilibrados correspondientes a esta tabla se pueden encontrar en la Tabla 4.6. Con datos reales, se logra una precisión ligeramente peor que con datos sintéticos, pero aún es muy alta. Los estimadores peor calificados — GNB y LSVM — coinciden con los señalados en la sección anterior. También se repite como mejor clasificado RF, esta vez acompañado por KNN.

Clasificador	TP	TN	FP	FN
<i>Decision Tree</i>	66,060	80,747	6918	6059
<i>Gaussian Naive Bayes</i>	51,110	83,541	4123	21,010
<i>K-Nearest Neighbors</i>	64,810	84,887	6679	3408
<i>Linear Support Vector Machine</i>	69,630	74,623	13,192	2339
<i>Bagged Decision Tree</i>	66,423	81,624	6027	5710
<i>Random Forest</i>	67,390	81,175	6475	4744
<i>Gradient Boosted Tree</i>	65,025	80,657	7893	6209

Tabla 4.5: Componentes de las matrices de confusión para los datos reales.

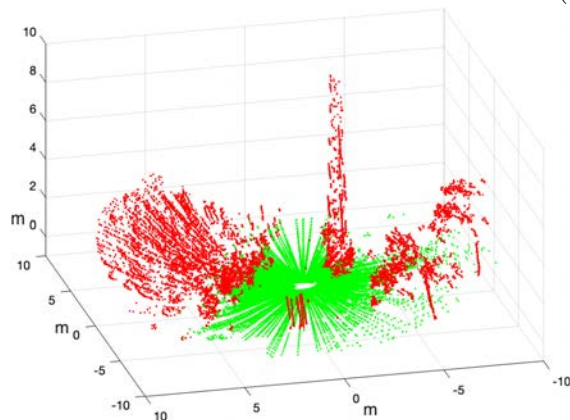
Las Figuras 4.15 c–4.17 c ilustran los resultados de aplicar el clasificador RF a las tres nubes de puntos reales. Se pueden observar buenos resultados de clasificación para todas estas escenas. Sin embargo, contienen errores como algunos puntos verdes aislados en la pendiente cerca del camino rural y en las paredes verticales del túnel.

Clasificador	PR	RE	F1	BA	MCC
<i>Decision Tree</i>	0.905	0.916	0.911	0.919	0.836
<i>Gaussian Naive Bayes</i>	0.925	0.709	0.803	0.831	0.692
<i>K-Nearest Neighbors</i>	0.907	0.950	0.928	0.939	0.873
<i>Linear Support Vector Machine</i>	0.841	0.968	0.900	0.909	0.814
<i>Bagged Decision Tree</i>	0.917	0.921	0.920	0.926	0.852
<i>Random Forest</i>	0.912	0.934	0.923	0.930	0.859
<i>Gradient Boosted Tree</i>	0.892	0.913	0.902	0.912	0.822

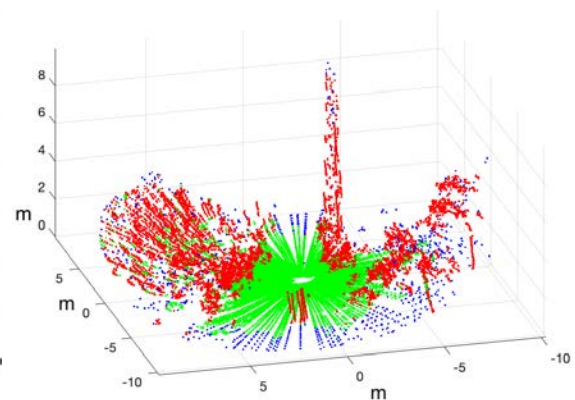
Tabla 4.6: Índices de precisión equilibrados para los datos reales.



(a)



(b)

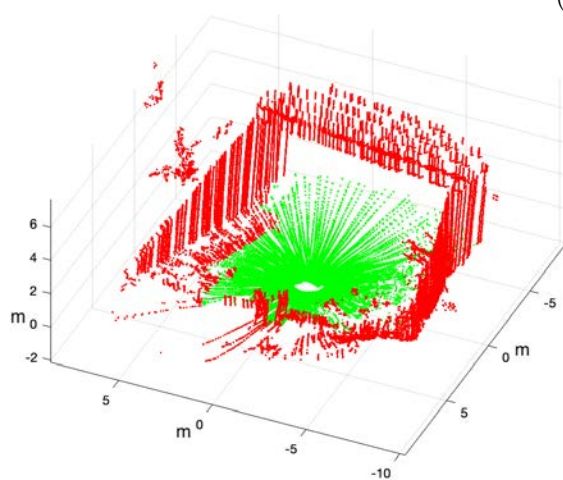


(c)

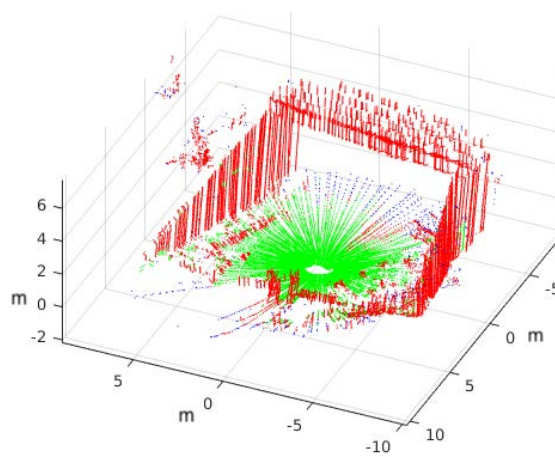
Figura 4.16: Imagen de un camino rural (a), una nube de puntos 3D real etiquetada manualmente (b) y clasificada con RF (c).



(a)



(b)



(c)

Figura 4.17: Imagen de un paso subterráneo (a), una nube de puntos 3D real etiquetada manualmente (b) y clasificada con RF (c).

4.6. Conclusiones

En este capítulo se ha descrito la obtención de clasificadores de transitabilidad del terreno para los barridos láser 3D adquiridos por el robot móvil Andábata en entornos naturales. Para este propósito, se han empleado siete potentes métodos de aprendizaje supervisado de la librería Python *Scikit-learn*. Además de ser un software libre muy completo, esta librería también ha facilitado el flujo de trabajo en gran medida.

Para el entrenamiento y validación, se han empleado nubes de puntos 3D con etiquetas binarias obtenidas automáticamente mediante el simulador robótico Gazebo. Estos datos nivelados se parecen mucho a los obtenidos por el sensor 3D de

Andábata en terreno sólido no resbaladizo. Sin embargo, cada punto sintético tiene asociado una etiqueta de transitabilidad que depende del valor de intensidad medida y que puede emplearse como modelo de datos para el entrenamiento.

Para la evaluación de la transitabilidad, se han calculado tres características espaciales simples para cada punto Cartesiano. Sin embargo, la extracción de características es un proceso que requiere mucho tiempo en Python y que ha sido necesario acelerar mediante compilación. Por el contrario, los tiempos de predicción, una vez obtenidas las características, son generalmente insignificantes. Con todo, Andábata podría determinar la transitabilidad de cada punto de un barrido láser 3D completo mucho antes de que esté disponible el siguiente barrido 3D.

Se ha obtenido una alta precisión en los índices de validación de datos desequilibrados con la mayoría de los estimadores, destacando el método *Random Forest* para nubes de puntos 3D sintéticas y reales. También se ha confirmado que los clasificadores de transitabilidad, entrenados solo con datos simulados, pueden funcionar bastante bien con los datos reales adquiridos mientras Andábata se teleoperaba en entornos naturales.

El autor de esta tesis ha contribuido elaborando material para la publicación en revista [97] y analizando los resultados de los clasificadores y evaluando sus prestaciones.

Capítulo 5

Navegación reactiva mediante clasificación de transitabilidad

5.1. Introducción

La transitabilidad del terreno debe evaluarse continuamente en los robots móviles para planificar movimientos de forma segura [13] con recursos computacionales limitados [56, 119]. Si los resultados de la transitabilidad son muy limitados, los movimientos del robot se restringirán innecesariamente, pero, por otro lado, si son muy permisivos, la seguridad del vehículo estará en peligro [28].

En el Capítulo anterior se han entrenado varios clasificadores para obtener la transitabilidad de nubes de puntos 3D niveladas. Esta evaluación confiable del terreno puede emplearse para construir mapas de transitabilidad 2D locales [97]. Un mapa de transitabilidad en coordenadas polares se construye proyectando sobre un mismo plano los puntos clasificados del barrido 3D y asignando valores de transitabilidad a cada región del mapa. Después es necesario seleccionar la dirección hacia donde dirigirse en función del terreno y de la ubicación del objetivo, evitando zonas no navegables. Se emplean simulaciones robóticas con Gazebo [111] para ajustar la reactividad antes de las pruebas reales.

La Figura 5.1 ilustra el proceso que se va a seguir para realizar la navegación autónoma del robot Andábata empleando los clasificadores previamente entrenados. Tras la obtención de la barridos 3D, se realizará un filtrado de las nubes de puntos

3D. Los puntos resultantes del filtrado se clasifican, para construir el mapas local de transitabilidad del entorno cercano. A partir de este mapa, se realiza la selección de dirección evitando zonas no navegables intentando alcanzar el objetivo.



Figura 5.1: Pasos para realizar navegación reactiva mediante clasificadores de transitabilidad.

El resto del Capítulo está organizado de la siguiente forma. En la sección 5.2 se describe la simulación de Andábata en Gazebo [111] que se utiliza para ajustar la navegación reactiva, cuyo esquema se propone en la sección 5.3. Los experimentos simulados y reales se comentan en las secciones 5.4 y 5.5, respectivamente. Finalmente, las conclusiones completan el Capítulo.

5.2. Simulación de Andábata en Gazebo

Las plataformas de simulación robótica que incluyen un motor de física como V-REP [120] o Gazebo [111] permiten obtener información realista de un vehículo terrestre moviéndose en su entorno. Por lo tanto, se pueden emplear para evaluar movimientos elementales [54], para evaluar la transitabilidad [36, 96] o para ajustar manualmente los parámetros de navegación [52].

Todo el software de Andábata se puede simular en Gazebo a través de un conjunto de paquetes ROS llamado *gazebo_ros_pkgs* [121], que proporcionan las interfaces necesarias mediante el uso de servicios y mensajes ROS. Con ellos se pueden construir *plugins* para la salidas de los sensores y las entradas de los motores tal y como funcionan en el mundo real. De esta manera, es posible probar los mismos nodos ROS en el robot o en el simulador de manera intercambiable.

En el Capítulo 4 sólo se simulaba en Gazebo el sensor láser 3D pero ahora es necesario un modelado completo para evaluar su comportamiento en experimentos en movimiento. En la Figura 5.2 puede observarse el chasis de Andábata desarrollado en detalle en Gazebo [122].

El modelo de suspensión de Andábata en Gazebo supone ruedas rígidas, de constante de elasticidad 3976.6 N m^{-1} y un coeficiente de amortiguación de 75.76 kg s^{-1} . La respuesta de las velocidades de las bandas de rodadura (v_l y v_r) a los comandos de velocidad (v_l^{sp} y v_r^{sp}) desde la computadora no es instantánea y se modela en Gazebo como un sistema de primer orden con una constante de tiempo de 35 ms.

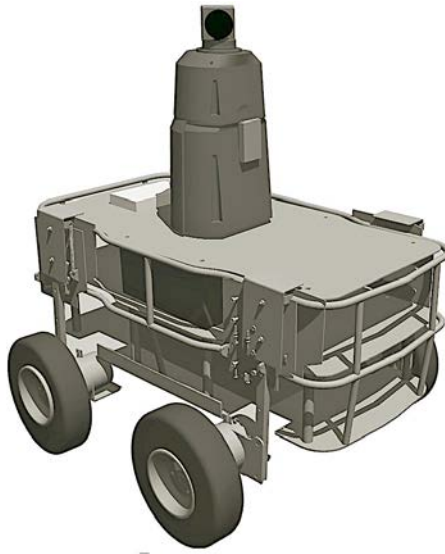


Figura 5.2: Modelo Andábata en Gazebo [122].

Se ha adjuntado una columna al chasis principal en la parte superior y centrada (ver Figura 5.2). En el lado frontal de la columna, se fija un cuboide rectangular para representar el teléfono inteligente a bordo de Andábata. Los datos de giroscopios e inclinómetros se pueden simular directamente del motor de física predeterminado de Gazebo *Open Dynamics Engine* (ODE). Los datos del GPS y del magnetómetro se simulan agregando ruido Gaussiano a la posición y rumbo exactos del robot móvil en el entorno virtual, respectivamente. Los *pluggings hector_gazebo* [123] se han empleado para incorporar todos estos sensores a Gazebo con sus correspondientes

tasas de adquisición.

La Figura 5.3 muestra una vista general del entorno natural generado con Gazebo [122], que es un cuadrado de 50 m de lado donde navega Andábata. Contiene muchos obstáculos positivos como hierba alta, rocas, árboles, una cerca y una barrera. Pero también contiene varias zanjas que actúan como obstáculos negativos.

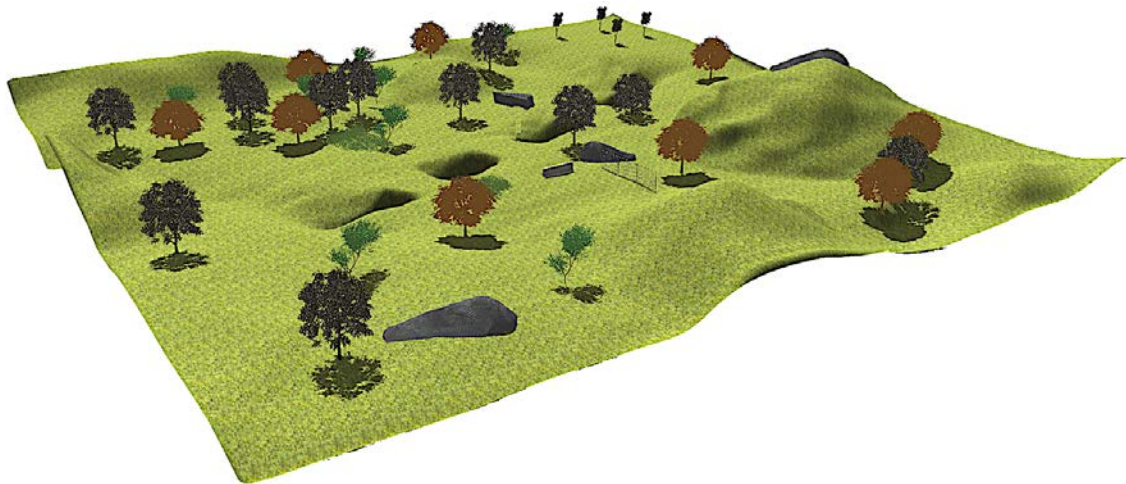


Figura 5.3: Vista general del entorno natural construido en Gazebo.

La Figura 5.4 muestra la simulación de Andábata moviéndose sobre el entorno virtual. En ella se representa la adquisición de uno de los barridos 2D verticales que componen un barrido 3D completo mediante líneas azules. Las líneas gruesas representan rangos detectados, mientras que las líneas delgadas indican falta de medición.

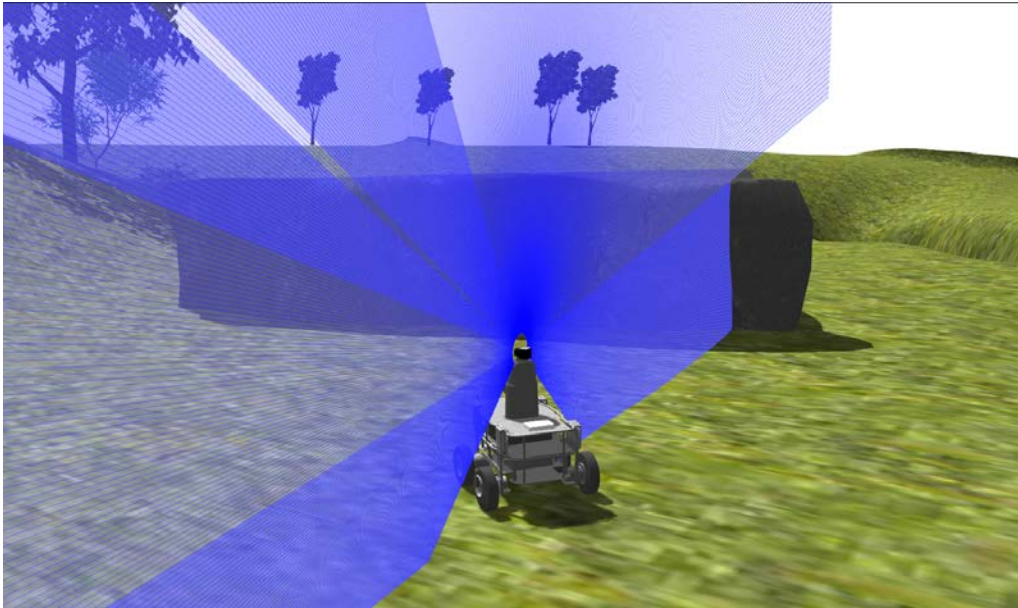


Figura 5.4: Simulación en Gazebo de Andabata moviéndose sobre el entorno virtual. Las líneas azules representan la adquisición de un solo barrido 2D vertical.

5.3. Selección de la dirección de movimiento

En el Capítulo 2 se presentó el esquema software de Andábata empleando nodos en ROS (ver Figura 2.9). El empleo de nodos hace que el software sea reutilizable y que se pueda modificar el comportamiento de un nodo sin afectar al resto, gracias a la propiedad de encapsulación de los nodos.

En esta sección se presentan las modificaciones llevadas a cabo en algunos nodos para mejorar la reactividad de Andábata. El nodo `map`, que en el Capítulo 3 generaba los FEM y FRM, ahora realiza la construcción de mapas 2D de transtibilidad empleando los modelos entrenados presentados en el Capítulo 4. Del nodo `filter` sólo se usa el primer paso, que se ha modificado aumentando el rango máximo de la proyección al plano horizontal de los puntos del barrido hasta 12m en lugar de 10m. Esto se ha realizado porque aunque el mapa de transitabilidad sigue teniendo un radio de 10m ahora esta centrado en la posición actual del robot en lugar del inicio del barrido como en el Capítulo 3.

El nodo `cloud_selector` deja de ser necesario porque el tiempo de cómputo

de los nodos `filter`, `map` y `dir_selection` serán menores que la generación de un barrido nivelado y no será necesario elegir cuál se procesa dado que se utilizarán todos los barridos disponibles.

Para la navegación local, el telémetro láser 3D se ha configurado para proporcionar un barrido 3D nivelado de los alrededores cada $t_s = 3.3$ s con 32,000 puntos aproximadamente, mientras Andábata se mueve. El radio de proximidad alrededor del punto objetivo utilizado para cambiar al siguiente se ha reducido de 10 m a 3 m debido a la incorporación a bordo del vehículo de un módulo GPS de bajo coste modelo Ublox ZED F9P [124] con una precisión de 1m.

La transitabilidad se evalúa para puntos individuales con el clasificador *Random Forest* previamente entrenado [97]. De esta manera, cada punto a menor distancia de 12 m desde el centro del barrido 3D se clasifica individualmente como transitable, no transitable o indefinido. Este procesamiento toma $t_c = 1.23$ s aproximadamente por cada barrido 3D, siendo la mayoría de este tiempo dedicado a la extracción de características.

Una vez clasificado cada barrido 3D, se construye un mapa de transitabilidad 2D proyectando cada punto 3D en un plano horizontal centrado en la posición actual del robot, la cual es diferente del centro del barrido 3D debido al movimiento del robot durante la adquisición del barrido 3D. El mapa de navegación consta de una rejilla polar dividida en 32 sectores de 11.25° y nueve coronas circulares formadas por diez radios desiguales sucesivos r :

$$r^j = 10 \frac{\tau^j - 1}{\tau^{10} - 1}, \quad j = 1 \dots 10, \quad (5.1)$$

donde la relación de expansión $\tau = 1,0682$ permite un radio de creciente de h a 10 m (ver Figura 5.5). Además, estos mapas locales están alineados con el Oeste y con el Sur a 180° y 270° , respectivamente.

Luego, cada celda de la rejilla 2D, con la excepción del círculo central, se etiqueta según los puntos proyectados que le cayeron dentro de la siguiente manera:

- Si la celda no contiene ningún punto, se etiqueta como vacía en color blanco.

- Con al menos un 15 % de puntos no transitables, la celda se clasifica como no transitable en color rojo.
- Con al menos un 85 % de puntos transitables, la celda se etiqueta como transitable en color verde.
- En cualquier otro caso, la celda se clasifica como indefinida en color gris.

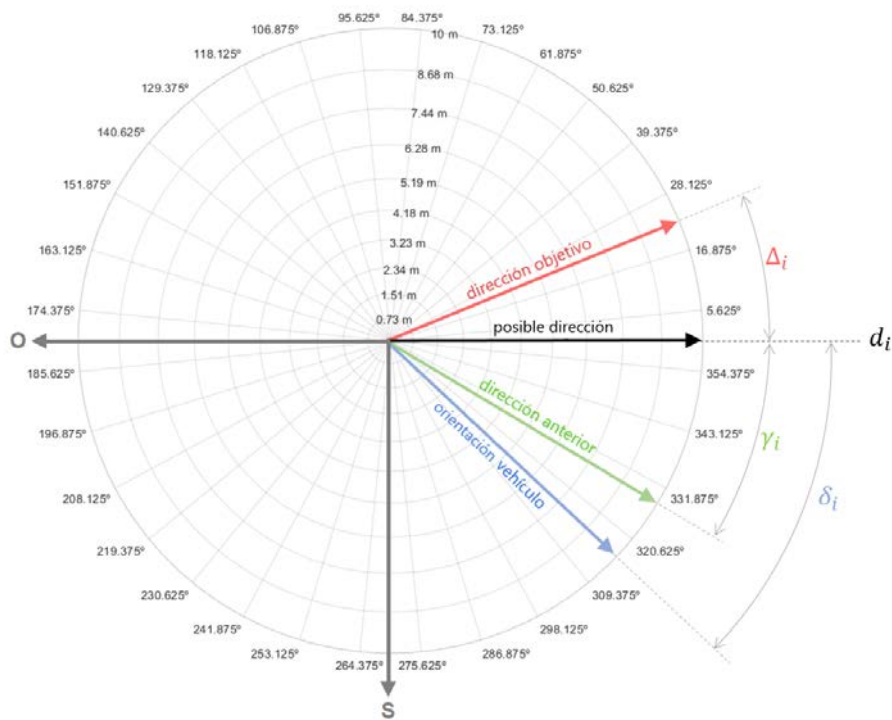


Figura 5.5: Direcciones relevantes sobre un mapa de transitabilidad 2D.

Todas las líneas d_i desde el centro de cada celda i son posibles direcciones de movimiento para Andábata (ver Figura 5.5). La dirección seleccionada d_j será la que minimice la función de coste:

$$J(d_i, i) = \frac{G(d_i)}{T(i)}, \quad i = 1 \dots 32, \quad (5.2)$$

que tiene en cuenta tanto la coincidencia con la dirección al objetivo G como la transitabilidad T de la celda.

La coincidencia con la dirección objetivo G para cada d_i se calcula como:

$$G(d_i) = |\Delta_i| + k_1 |\delta_i| + k_2 |\gamma_i|, \quad (5.3)$$

donde k_1, k_2 son ganancias ajustables, y Δ_i, δ_i y γ_i son las diferencias angulares de d_i respecto a la dirección del punto objetivo, la orientación actual del vehículo y la dirección anteriormente escogida, respectivamente (ver Figura 5.5).

La transitabilidad T para cada sector i se calcula como:

$$T(i) = k_3 (1 + n(i)) + k_4 (n(i+1) + n(i-1) - |n(i+1) - n(i-1)|), \quad (5.4)$$

donde $n(i)$ es el número de celdas transitables en el sector i desde el centro hasta que se alcanza una celda no transitable o la celda más externa (ver Figura 5.6), k_3 y k_4 son ganancias ajustables que recompensan las direcciones sin obstáculos en el sector y en los dos sectores adyacentes, respectivamente.

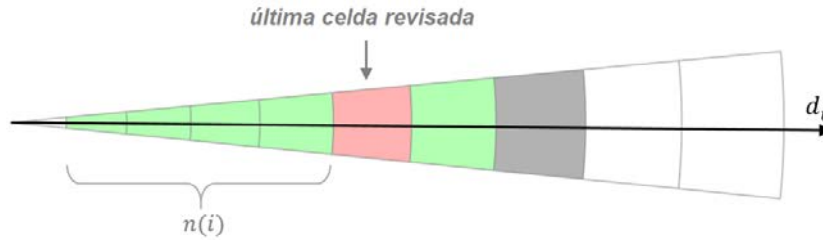


Figura 5.6: Evaluación de transitabilidad para el sector i .

En resumen, es necesario ajustar cuatro parámetros de navegación k_1, k_2, k_3 y k_4 para la función de coste J . La evaluación de las direcciones es relativamente simple y solo toma $t_m = 0.15$ s aproximadamente en la computadora de a bordo.

La Figura 5.7 muestra el cronograma de tareas para la navegación local reactiva. Se introduce un retraso t_d intencionalmente para proporcionar actuaciones tres veces por cada barrido láser 3D construyendo tres mapas de transitabilidad 2D, cada $t_s/3 = 1.1$ s, aproximadamente. Para este propósito, la demora debe cumplir:

$$t_m + t_d \approx t_s/3 \Rightarrow t_d \approx 0.95 \text{ s}. \quad (5.5)$$

Sin embargo, el intervalo entre los cambios de dirección no es del todo constante porque t_c depende fuertemente del número de puntos de cada barrido 3D. También se puede observar en la Figura 5.7 que la adquisición de una nube de puntos 3D ocurre simultáneamente con la clasificación de la transitabilidad de la nube de puntos previa porque se ejecutan en paralelo en distintos nodos ROS.

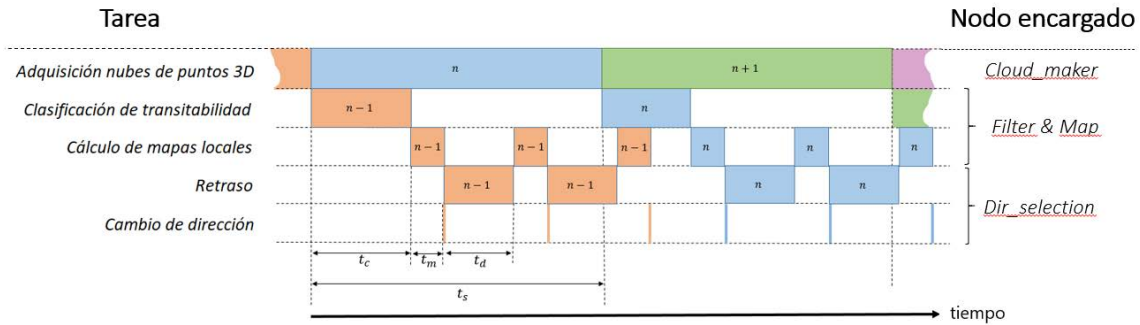


Figura 5.7: Cronograma de tareas y nodos encargados para la navegación reactiva.

5.4. Experimentos en simulación

La estrategia reactiva para la navegación local se ha probado ampliamente con simulaciones de Gazebo para ajustar sus cuatro parámetros. El parámetro de mayor relevancia es k_3 que regula cómo se combinan los comportamientos de perseguir el objetivo y el de evitar obstáculos. Los parámetros k_1 y k_2 intentan evitar cambios de dirección y k_4 favorece las direcciones libres de obstáculos. Como resultado de un proceso de prueba y error, los siguientes parámetros se han seleccionado manualmente: $k_1 = k_2 = 0,15$, $k_3 = 1$, y $k_4 = 0,3$.

La Figura 5.8 muestra con una línea azul el camino global seguido por Andábata mientras persigue tres puntos objetivo distantes en el entorno natural con $v^{sp} = 0.3 \text{ ms}^{-1}$. En esta figura, los datos del GPS se representan con puntos rojos, los objetivos se dibujan con un pequeño círculo verde rodeado por un círculo verde discontinuo de proximidad de 3 m y la equis negra marca el comienzo del camino.

Se puede observar en la Figura 5.8 que el componente reactivo del sistema de

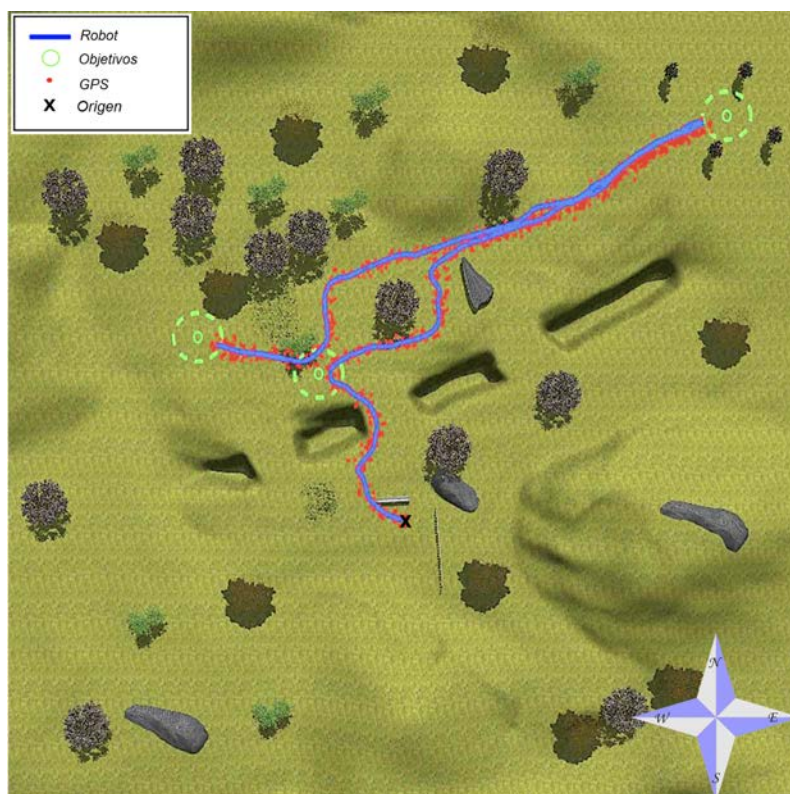


Figura 5.8: Vista aérea del camino seguido por Andábata en el entorno simulado.

navegación permite evitar obstáculos tanto positivos como negativos. Concretamente, en el camino hacia el primer objetivo, Andábata evita una barrera artificial y una zanja profunda. Luego, cuando intenta alcanzar el segundo objetivo, rodea un árbol y una gran roca. Finalmente, elude la hierba alta cerca del último punto de objetivo.

La Figura 5.9 contiene la trayectoria de 161 m de longitud de la Figura 5.8 junto con marcas de tiempo y coordenadas horizontales. En total se han adquirido 186 nubes de puntos 3D y se han construido sus correspondientes 558 mapas de navegación local 2D.

La altura a la que se encuentra el robot y sus cambios en orientación durante esta trayectoria, se representan en la Figura 5.10. Se pueden observar cambios suaves de rumbo con la excepción del giro de 180° cuando se alcanza el segundo objetivo a los 480 s. La elevación máxima que ascendió y descendió el robot móvil fue de 2.15 m.

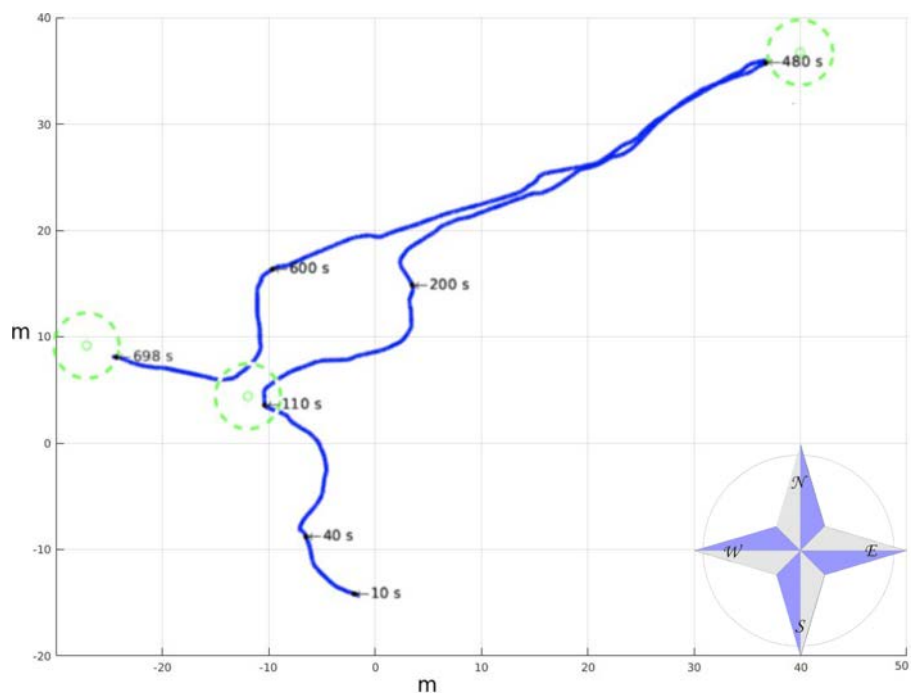


Figura 5.9: Trayectoria seguida por Andábata con marcas de tiempo.

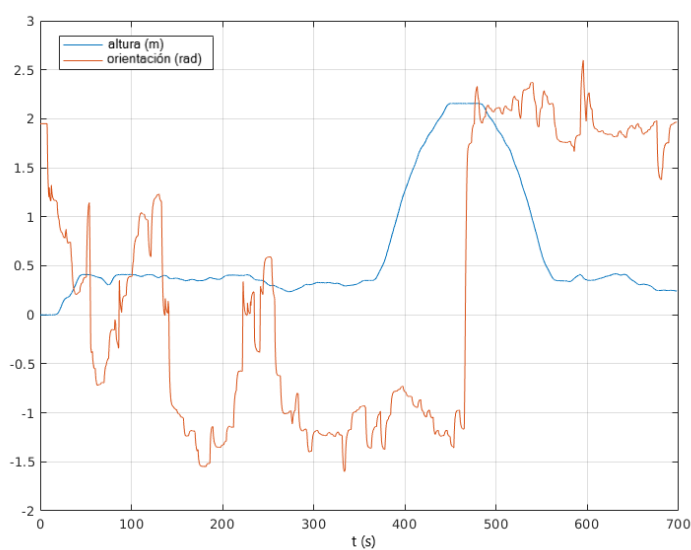


Figura 5.10: Altura y cambios en la orientación del robot durante la prueba simulada.

En la Figura 5.11 se muestra un ejemplo de barrido 3D clasificado por su transitabilidad. Esta nube de puntos nivelada se adquirió en el camino hacia el primer punto objetivo cerca de la zanja. Los tres mapas consecutivos de transitabilidad creados a partir de este barrido 3D se representan en la Figura 5.12. La zanja aparece en

estos mapas como una gran región blanca a la izquierda y arriba atravesada por la dirección objetivo al Noroeste. Sin embargo, la dirección seleccionada mantiene al robot lejos de este obstáculo negativo como se puede observar en los tres mapas, donde la orientación del vehículo apunta al Noreste.

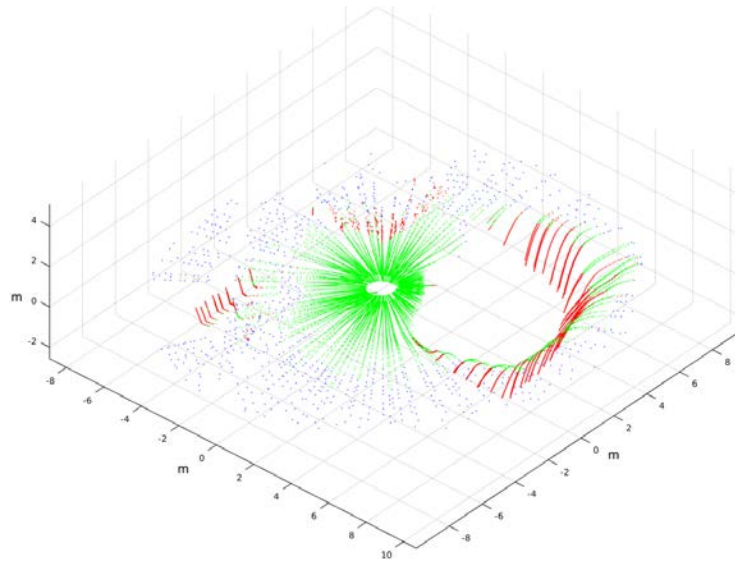


Figura 5.11: Un barrido 3D obtenido durante la navegación y clasificado según su transitabilidad.

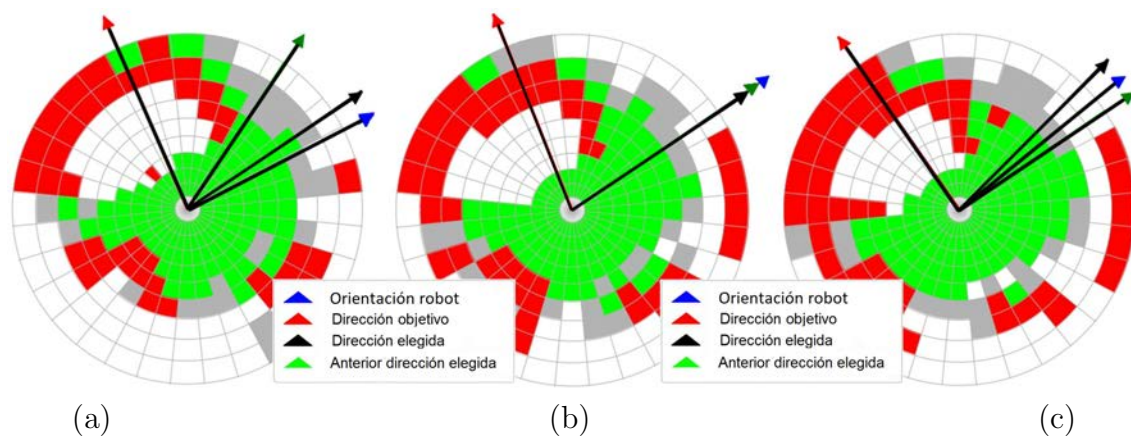


Figura 5.12: Primer (a), segundo (b) y tercer mapas de transitabilidad para la nube de puntos 3D de la Figura 5.11

5.5. Experimentos reales

Una vez ajustados los parámetros del control reactivo mediante simulaciones, se ha probado con Andábata en experimentos reales en el sendero que atraviesa una hondonada y en un parque urbano descuidado.

5.5.1. Sendero en una hondonada

Se han elegido dos puntos objetivo para atravesar un sendero con pendientes en el interior de una hondonada. En general, los bordes del sendero consisten en maleza seca y colinas (ver Figura 5.13).



Figura 5.13: Andábata avanzando por el sendero dentro de la hondonada. Se muestran fotografías sucesivas desde el comienzo (a) hasta el final (f) de la trayectoria.

La Figura 5.14 muestra una vista aérea de la ruta seguida por el robot móvil según lo registrado por los datos del GPS. Con $v^{sp} = 0.3 \text{ m s}^{-1}$, la trayectoria fue 133 m de largo y duró 462 s. En total, se adquirieron 137 barridos 3D de un promedio de 27,694 puntos.

En la Figura 5.15 se muestra la vista superior de una nube de puntos en 3D real clasificada por transitabilidad. Este barrido en particular se adquirió en el camino hacia el segundo punto objetivo, unos metros después superar el primer objetivo (ver Figura 5.13 d). Los tres mapas de transitabilidad consecutivos construidos a partir



Figura 5.14: Mediciones GPS donde se indica el lugar donde se tomó cada fotografía de la Figura 5.13.

de este barrido 3D se representan en la Figura 5.16. En ellos, aparece una colina y vegetación en dirección al segundo objetivo, por lo que el robot debe desviarse de su dirección actual, como se puede comprobar por el cambio de rumbo entre el primer y los sucesivos mapas de transitabilidad.

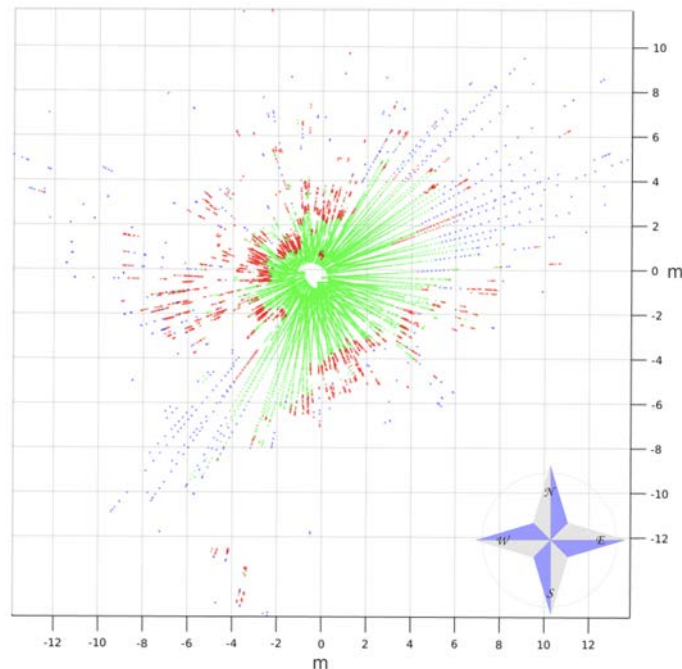


Figura 5.15: Vista superior de un barrido 3D real clasificado por transitabilidad.

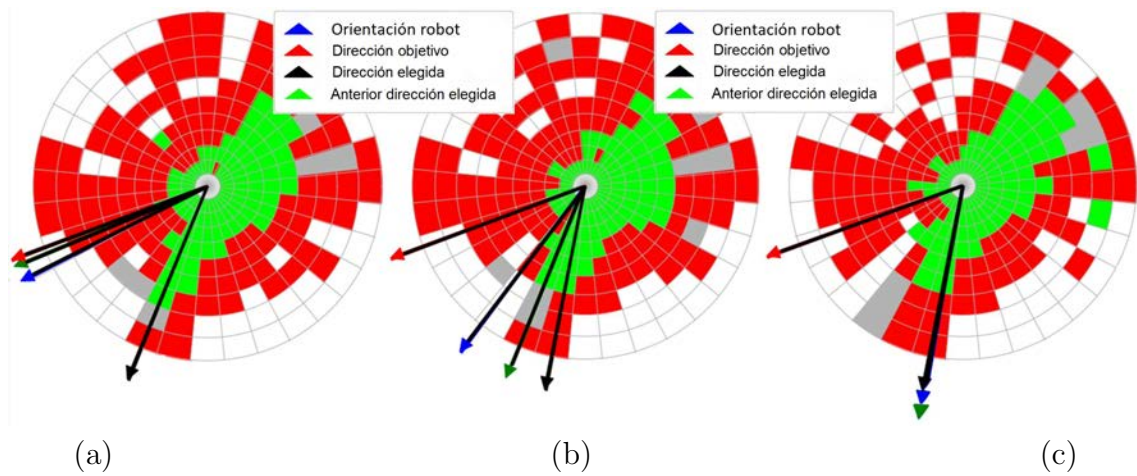


Figura 5.16: Primer (a), segundo (b) y tercer mapas de transitabilidad para el barrido real de la Figura 5.15.

5.5.2. Parque urbano

También se ha probado la navegación autónoma en un parque urbano descuidado utilizando tres puntos objetivo (ver Figura 5.17). El punto intermedio aparece dos veces en el camino hacia los puntos extremos en el Oeste y en el Este. La vista aérea de la Figura 5.17 muestra la trayectoria GPS cuando la velocidad lineal del robot estaba fijada a $v^{sp} = 0.3 \text{ m s}^{-1}$.

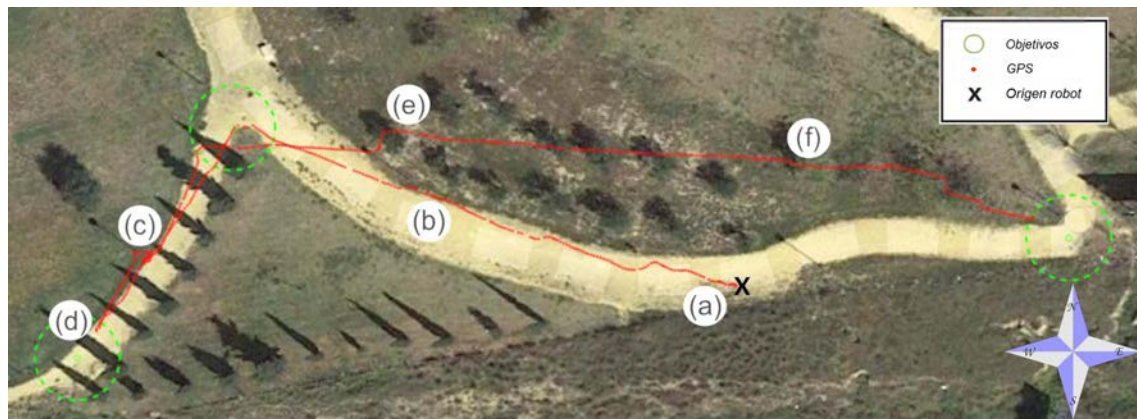


Figura 5.17: Mediciones GPS durante la navegación en el parque. Se indica el lugar que corresponde a cada fotografía de la Figura 5.18

La superficie casi plana del parque contiene obstáculos tanto naturales (árboles, arbustos y malas hierbas) como artificiales (farolas, vallas y papeleras). La mayor

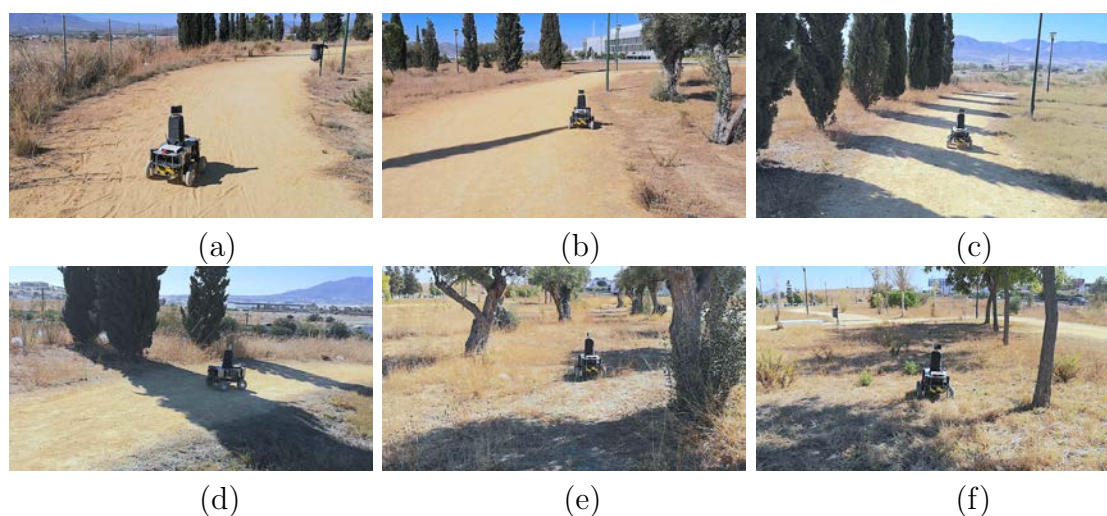


Figura 5.18: Fotografías de Andábata en el parque urbano desde el inicio (a) hasta el final (f) de la trayectoria.

parte de la trayectoria se realiza sobre el camino amarillo con la excepción del último tramo, donde el robot atraviesa una zona irregular con árboles y malezas mientras persigue el último punto objetivo al Este (ver Figura 5.18).

La trayectoria del parque fue 181 m de largo y duró 660 s. En total, Andábata adquirió 183 barridos 3D. El valor promedio de 34,498 puntos por barrido 3D es mayor que en el experimento anterior porque la visibilidad del cielo se reduce principalmente debido a las copas de los árboles.

5.5.3. Discusión de los resultados

La selección de la ubicación de los puntos objetivos es muy relevante para completar la navegación. Para probarlo, se ha intentado repetir el recorrido del parque urbano eliminando el punto intermedio. En este caso, Andábata no logra alcanzar el objetivo al Oeste (ver Figura 5.19 a). Este fallo se debe a que el robot no encontró un camino hacia el objetivo a través de la maleza y permaneció dando vueltas.

En este caso, existe un conflicto de comportamientos en el controlador reactivo: si el vehículo avanza en una dirección libre de obstáculos d_i sobre el camino amarillo, aumenta la diferencia angular con respecto a la dirección objetivo Δ_i . Por lo tanto,

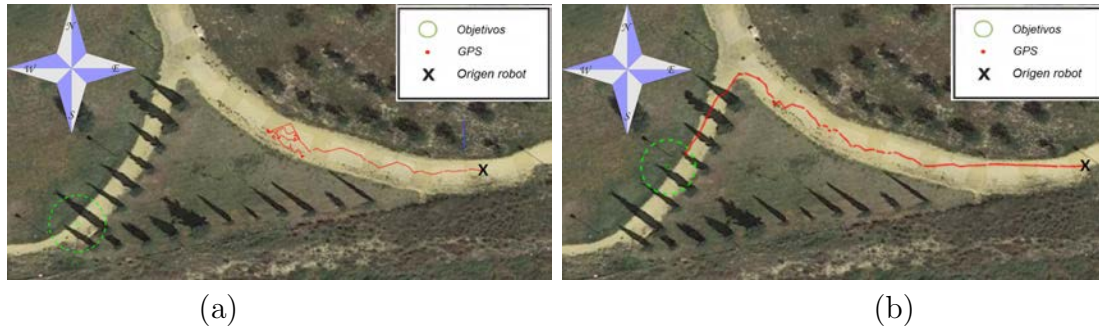


Figura 5.19: Navegación reactiva en el parque urbano sin el punto objetivo intermedio: fracaso (a) y éxito (b).

valores altos de $G(d_i)$ (5.3) y $T(i)$ (5.4) se obtienen a la vez en el numerador y denominador de la función de coste $J(d_i, i)$ (5.2), respectivamente. De hecho, si el punto de objetivo al Oeste se mueve unos metros hacia el Noreste, el robot logra alcanzarlo circunnavegando las malezas y los cipreses (ver Figura 5.19 b).

Generalmente, un vehículo no tripulado debe evitar los cuerpos de agua (charcos, lagos, etc) durante la navegación para evitar daños eléctricos o quedarse atascado dentro [125]. Las aguas profundas se pueden detectar indirectamente con un escáner láser 3D por la falta de medidas provocadas por las deflexiones de los rayos láser, que los hacen comportarse como obstáculos negativos [35].

Sin embargo, Andábata no pudo evitar los charcos que encontró en su camino. Esto se debe a que es suficiente con tener un punto clasificado como transitable dentro de una celda cerca del charco en el mapa de transitabilidad para etiquetar toda la celda casi vacía como verde (ver Figura 5.20).

Otro tema relevante para la navegación exterior son los voladizos como las copas de los árboles o los túneles [87]. La Figura 5.21 a muestra una nube de puntos 3D tomada del experimento del parque urbano donde el robot estaba cerca de un olivo. Se puede observar que los puntos de la copa de los árboles están correctamente clasificados como no transitables en color rojo. Sin embargo, la proyección de estos puntos en el mapa de transitabilidad 2D de la Figura 5.21 b provoca que la mayoría de las celdas del suelo alrededor del vehículo se consideren como no transitables, reduciendo considerablemente el espacio navegable. En la Figura 5.21 c se muestra

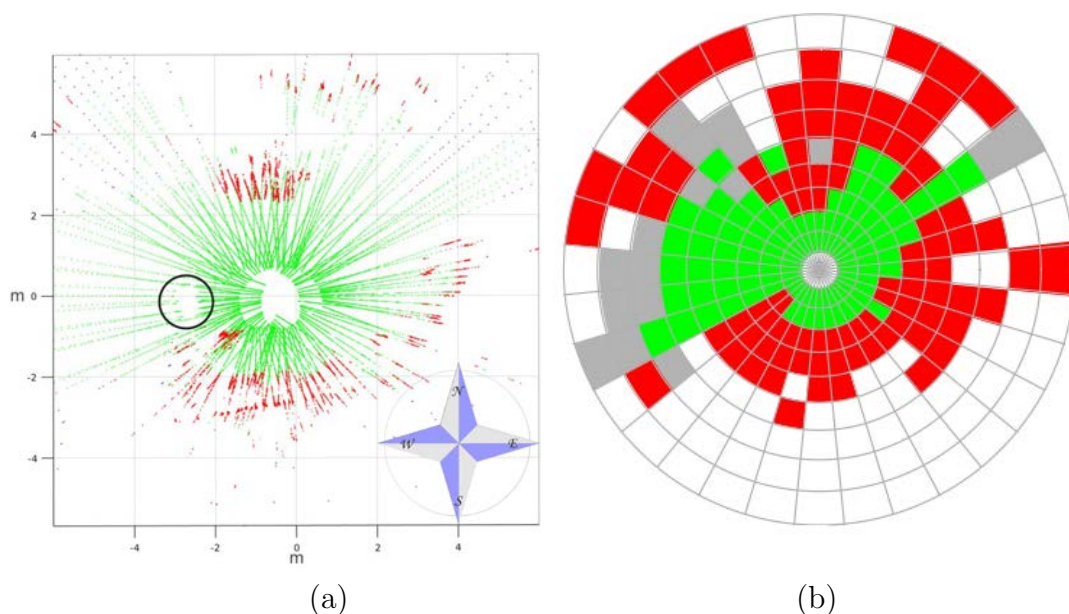


Figura 5.20: Vista cercana de barrido 3D desde arriba que contiene un charco frente al vehículo indicado por un círculo negro (a) y su primer mapa de transitabilidad (b).

el mapa de transitabilidad si se hubieran eliminado voladizos como en la Sección 3.2. En este caso se pueden observar un mayor número de celdas verdes alrededor del vehículo y de celdas blancas lejos del robot.

Un vehículo también puede encontrar obstáculos dinámicos, como animales o personas, en entornos naturales. Con esto en mente, se ha probado la navegación autónoma mientras Andábata se cruzaba con un peatón. Sin embargo, el robot no pudo prevenir la colisión correctamente porque la tasa de adquisición de barridos 3D es claramente insuficiente para este propósito.

5.6. Conclusiones

En este Capítulo se ha descrito cómo Andábata distingue suelo transitable en las nubes de puntos 3D adquiridas en movimiento de su entorno cercano mediante el uso de un clasificador de transitabilidad previamente entrenado con datos sintéticos. Se ha propuesto un esquema de navegación reactiva a baja velocidad para alcanzar puntos objetivo GPS mientras se evitan obstáculos tanto positivos como negativos

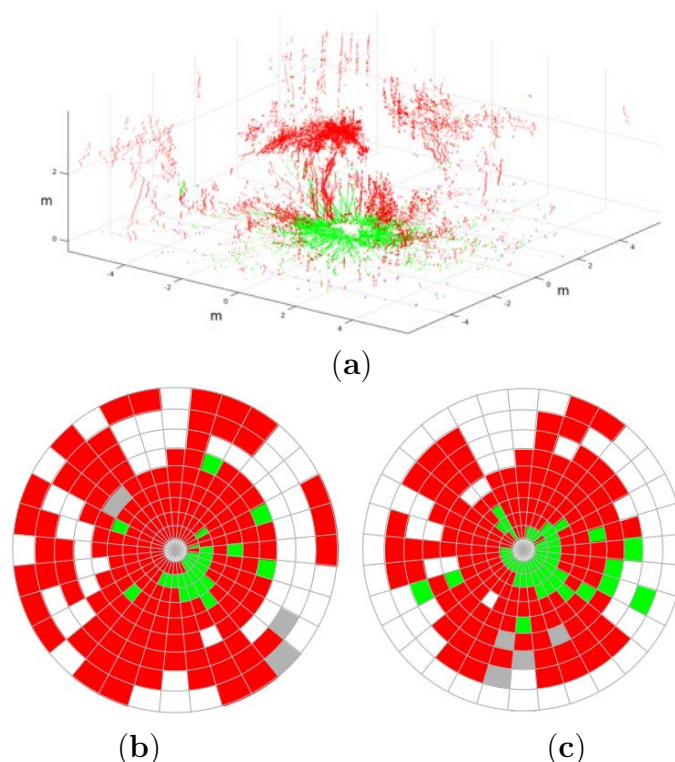


Figura 5.21: Un barrido láser 3D clasificado con Andábata debajo de un árbol (a), su primer mapa de transitabilidad (b) y el mapa 2D resultante de eliminar los voladizos (c).

en entornos naturales.

Se han empleado simulaciones robóticas realistas con Gazebo para ajustar los parámetros reactivos de manera apropiada. De esta forma, se han evitado numerosos experimentos con Andábata, lo que representa una ganancia considerable en el tiempo de pruebas y en la integridad del robot. Además, se han presentado experimentos de campo, en los que Andábata ha seguido con éxito diferentes caminos utilizando solo unos pocos puntos objetivo GPS distantes.

En este Capítulo se ha mejorado la estrategia de navegación presentada anteriormente en el Capítulo 3. Ésto se ha logrado procesando todos los barridos láser 3D adquiridos por el robot en movimiento. Además, la reactividad se ha mejorado sustancialmente mediante la construcción de tres mapas de transitabilidad 2D para cada nube de puntos 3D nivelada a medida que el robot se mueve a través de ella. Aún así, el tiempo de adquisición de los barridos láser 3D de 3.3s limita

considerablemente poder desarrollar comportamientos reactivos más rápidos.

El autor de esta tesis en este Capítulo ha participado en el desarrollo y pruebas del software implementado y en el análisis del funcionamiento de las técnicas propuestas.

Capítulo 6

Conclusiones

6.1. Recapitulación

En esta tesis, el robot móvil terrestre Andábata ha sido empleado como caso de estudio para abordar varios aspectos necesarios para la navegación autónoma en entornos naturales. Se han descrito sus sensores, la arquitectura software en ROS y la generación de nubes de puntos 3D niveladas durante el movimiento del vehículo. La navegación global empleada durante los experimentos se realiza mediante el uso de GPS para poder seguir puntos objetivo distantes y la navegación local se implementa de forma reactiva. Para cada una de las estrategias de navegación reactiva se han aportado soluciones originales y se han estudiado los resultados que proporcionan.

Así, se ha desarrollado la generación de mapas de elevación borrosos (FEM) y su uso en la navegación en Andábata. Cada barrido 3D nivelado se emplea para producir un FEM y su correspondiente máscara de confiabilidad borrosa (FRM), que se procesan para elegir la mejor dirección de movimiento para alcanzar un punto de objetivo distante mientras se cruzan áreas transitables.

Ha sido necesario filtrar las nubes de puntos 3D y la ejecución multiproceso para aumentar la frecuencia de construcción de FEM y FRM, ya que su tiempo de cálculo promedio es superior a la tasa de adquisición de los barridos. Se han presentado resultados experimentales de Andábata navegando de forma autónoma en un parque urbano.

También se han desarrollado clasificadores de transitabilidad para barridos láser 3D adquiridos por el robot móvil Andábata en entornos naturales. Para este propósito, se han probado siete potentes métodos de aprendizaje supervisado de la librería Python *Scikit-learn*. Además de ser muy completo y ser software libre, esta librería también ha facilitado el flujo de trabajo en gran medida.

Para realizar el entrenamiento y la validación ha sido necesario utilizar nubes de puntos 3D con etiquetas binarias obtenidas automáticamente con el simulador robótico Gazebo. Estos datos se parecen mucho a los obtenidos por el sensor 3D de Andábata en la realidad, pero cada punto sintético tiene etiqueta de transitabilidad, que depende de su medición de intensidad.

Para el cómputo de la transitabilidad, se han calculado tres características espaciales simples para cada punto Cartesiano. Sin embargo, la extracción de características es un proceso que requiere mucho tiempo y ha sido necesario acelerar mediante compilación. Los tiempos de predicción, una vez obtenidas las características, son generalmente insignificantes. Con todo, se ha logrado que Andábata determine la transitabilidad de un barrido completo con alta fiabilidad mucho antes de que esté disponible el siguiente.

Se ha propuesto un método de generación de mapas de transitabilidad 2D a partir de puntos clasificados de las nubes de puntos 3D adquiridas en movimiento para la navegación reactiva evitando obstáculos tanto positivos como negativos en terrenos naturales.

Se ha empleado Gazebo para ajustar los parámetros reactivos en simulación. De esta forma se han evitado numerosos experimentos con Andábata, lo que representa una ganancia considerable en el tiempo y en la integridad del robot. Finalmente, se han realizado pruebas de navegación reales con Andábata en entornos naturales con resultados satisfactorios y se han analizado en qué situaciones existe la posibilidad de mejorar el comportamiento.

El tiempo de adquisición de barridos láser 3D limita considerablemente la adopción de comportamientos más rápidos para poder trabajar con obstáculos dinámicos

como podrían ser, otros vehículos, animales o peatones.

6.1.1. Comparativa con otros trabajos

Otros trabajos realizan investigaciones similares en las que se proponen sistemas de navegación autónomos. Sin embargo, es difícil de comparar en términos absolutos las soluciones propuestas porque los parámetros utilizados en cada experimento son diferentes y tienen un gran impacto en los resultados como podrían ser el sensor 3D empleado y su tasa de generación de nubes de puntos, los obstáculos del entorno o la velocidad del vehículo.

En cualquier caso, se ha incluido la Tabla A.1 en el Apéndice A que ilustra una comparativa con trabajos previos realizados por otros autores.

6.2. Publicaciones derivadas

Como fruto del trabajo realizado en esta tesis, el autor ha obtenido las siguientes publicaciones en congresos internacionales y revistas científicas.

En revistas indexadas en el JCR se han publicado los siguientes artículos (en orden cronológico):

- J. L. Martínez, **M. Morán**, J. Morales, A. J. Reina y M. Zafra (2018). Field Navigation using Fuzzy Elevation Maps Built with Local 3D Laser Scans. En *Applied Sciences*. Vol. 8 (3), 397. Año JCR 2018: Factor de Impacto 2'217, segundo cuartil. puesto 67 de 148 revistas en la categoría Física Aplicada [51].
- J. L. Martínez, **M. Morán**, J. Morales, A. Robles y M. Sánchez (2020). Supervised Learning of Natural-Terrain Traversability with Synthetic 3D Laser Scans. En *Applied Sciences*. Vol. 10 (3), 1140. Año JCR 2019: Factor de Impacto 2'474, segundo cuartil, puesto 32 de 91 en la categoría de Ingeniería Multidisciplinar [97].

- J. L. Martínez, J. Morales, M. Sánchez, **M. Morán**, A. Reina y J. Fernández-Lozano (2020). Reactive Navigation on Natural Environments by Continuous Classification of Ground Traversability. En *Sensors*. Vol. 20 (22), 6423. Año JCR 2019: Factor de Impacto 3'275, primer cuartil, puesto 15 de 64 en la categoría Instrumentos e Instrumentación [122].

Por otra parte, en congresos internacionales se han presentado los siguientes artículos (en orden cronológico):

- A. Pomares, J. L. Martínez, A. Mandow, M. A. Martínez, **M. Morán** y J. Morales (2018). Ground Extraction from 3D Lidar Point Clouds with the Classification Learner App. En *26th Mediterranean Conference on Control and Automation (MED)*. Pag. 400-405. Zadar, Croacia [90].
- M. Sánchez, J. L. Martínez, J. Morales, A. Robles y **M. Morán** (2019). Automatic Generation of Labeled 3D Point Clouds of Natural Environments with Gazebo. En *IEEE International Conference on Mechatronics (ICM)*. Pag. 161-166. Ilmenau, Alemania [108].

6.3. Trabajos futuros

Existen numerosas formas de continuar los temas tratados a lo largo de esta tesis. Entre las líneas futuras de investigación más prometedoras caben destacar las siguientes:

La inclusión de sensores adicionales como cámaras podría reducir la incertidumbre que presenta el hecho de utilizar sólo el LIDAR como fuente de datos del entorno para el cómputo de la transitabilidad. Así, se podría incluir mejoras para detectar obstáculos dinámicos frente al vehículo procesando las imágenes de la cámara del teléfono inteligente a bordo.

La generación de un mapa global incremental a medida que el robot navegara aprovecharía la información de regiones espaciales ya visitadas en algún barrido

previo. También se podría estudiar la detección de cuándo el robot se atasca en un bucle intentado encontrar un camino seguro y sea capaz de suspender la misión.

Otra posibilidad sería utilizar otras fuentes de datos para la evaluación de la transitabilidad como imágenes por satélite o realizadas desde un dron, ya fueran con un LIDAR o con cámaras, para complementar la información disponible para la navegación. Si se dispusiera de un mapa global, se podrían poner en funcionamiento técnicas de planificación de camino globales que podrían arrojar un resultado más eficiente en la navegación.

Respecto los métodos de clasificación empleados, se podría ampliar utilizando redes neuronales con algoritmos de *deep-learning* (aprendizaje profundo) para comparar sus prestaciones y resultados con los empleados en esta tesis. Se ha observado que, en los escenarios reales, hay menos espacios libres que en el entorno simulado, por lo que es necesario trabajar con un escenario simulado más complicado incluyendo más elementos, como maleza y charcos.



UNIVERSIDAD
DE MÁLAGA

Apéndice A

Trabajos relacionados

En la Tabla A.1 de este Apéndice se incluyen los trabajos de otros autores más cercanos a esta tesis doctoral. Para cada uno de éstos, se comentan el sensor 3D empleado, el mapa 2D utilizado, el entorno en el que se encuentra el robot móvil, los procedimientos empleados y algunos detalles sobre el funcionamiento.

Ref.	Sensor	Mapa 2D	Procedimientos	Entorno	Funcionamiento
[3]	Láser 3D y cámaras estéreo	Mapa de transitabilidad y contornos de obstáculos negativos	Análisis de transitabilidad y de contornos de obstáculos	Escenarios de búsqueda y rescate	Realiza sólo análisis, no navegación ni planificación
[7]	LIDAR 3D Velodyne HDL-64E	Mapa de superficie multinivel	Análisis de transitabilidad y planificación de camino	Escenarios naturales	Detecta obstáculos pero el robot es teleoperado a 4.5 m/s
[27]	Combinación de LIDAR, cámaras e imágenes de satélite	Mapa de coste de transitabilidad	Análisis de transitabilidad, planificación de caminos y navegación	Escenarios naturales	Navegación autónoma pero supervisada por operador a 2.3 m/s
[28]	LIDAR 3D Velodyne	Mapa de transitabilidad	Sólo análisis de transitabilidad	Entornos no estructurados	Pruebas con 2 robots distintos teleoperados
[30]	Láser 3D basado en Hokuyo UTM-30LX y cámaras TOF	Mapa binario: navegable o no navegable	Análisis de transitabilidad sólo en frente del robot	Escenarios al aire libre	Análisis de transitabilidad de alto nivel y planificación de la trayectoria realizada fuera de línea
[31]	Láser 3D Sick	DEM del terreno	Generación de mapa, análisis de transitabilidad y control de movimiento	Terreno complicado simulado	Pruebas realizadas en escenarios interiores
[35]	Lidar Velodyne HDL-64	Histograma LIDAR	Análisis de transitabilidad y detección de obstáculos	Fuera de carretera	Detección en tiempo real de obstáculos positivos, negativos y de agua
[36]	LIDAR 3D Velodyne HDL-32E	DEM que incluyen modelos cinemáticos	Análisis de transitabilidad, planificación y navegación	Suelo irregular y suelo rocoso	Planificación realizada fuera de línea

Tabla A.1: Listado de trabajos relacionados (continúa en página siguiente).

Ref.	Sensor	Mapa 2D	Procedimientos	Entorno	Funcionamiento
[39]	Láser 3D basado en Hokuyo UTM-30LX	Mapas locales multiresolución	Creación de mapa y análisis transitabilidad	Entornos naturales	Robot teleoperado
[41]	LIDAR 3D Velodyne HDL-32E	Mapa-nube de puntos	Análisis de transitabilidad y planificación de trayectoria	Entornos 3D complejos	Se navega con un mapa previo generado con nubes de puntos
[43]	Láser 3D basado en Sick LMS291	Mapa de obstáculos	Clasificación de puntos y creación de mapa de obstáculos	Entornos naturales	Realiza clasificación online y la generación de mapas offline
[45]	LIDAR Velodyne HDL-32E y cámara con resolución 640x480	Mapa probabilístico de ocupación	Generación de mapas, análisis de transitabilidad y evitación de obstáculos	Fuera de carretera	Combina información de cámara y del LIDAR
[47]	Láser 3D basado en Sick LMS291	Mallas triangulares irregulares	Creación de mapas, planificación de camino y navegación	Entornos planetarios	Utiliza <i>stop and go</i>
[48]	Láser 3D basado en Hokuyo y 8 cámaras estéreo	Mapas de superficie multiresolución	Análisis de transitabilidad y creación de mapas	Parking y entorno planetario	No realiza navegación
[50]	Escáner láser Sick LMS 200	Obstáculos de clase pendiente y obstáculos de clase altura	Análisis de transitabilidad y Realiza planificación pero no navegación	Carretera, escenario natural, escalera y rampa	Emplea <i>stop, scan and go</i> con velocidad 0.6 m/s
[52]	Cámara RGB-D	Mapa de transitabilidad	Creación de mapa, análisis de transitabilidad y planificación de camino	Tunel y entorno interiores a una mina con terreno accidentado	Se centra en la planificación de trayectorias con algoritmo RRT

Tabla A.1: Listado de trabajos relacionados (continúa en página siguiente).

Ref.	Sensor	Mapa 2D	Procedimientos	Entorno	Funcionamiento
[54]	LIDAR Sick LMS 300	Mapa de ocupación	Análisis de transitabilidad y planificación de trayectoria	Entornos dinámicos	Combina uso de comportamientos y planificación de alto nivel
[55]	Láser 3D basado en Sick LMS 291	Mapa de ocupación	Evitación de obstáculos	Terreno con vegetación	Emplea navegación semi-autónoma en modo teleoperado
[56]	4 pares de cámaras estéreo	DEM	Análisis de transitabilidad, navegación y planificación de trayectoria	Entornos planetarios en Marte	Combina uso teleoperado y autónomo
[57]	Láser 3D personalizado	Mapas de Voronoi	Creación de mapa incremental, análisis de transitabilidad, y navegación en tiempo real	Escenarios al aire libre	Navegación en diferentes entornos a una velocidad de 3 m/s
[58]	Sólo análisis teórico	Mapa por sectores borrosos	Evitación de obstáculos y movimiento hacia objetivos	Sólo análisis teórico	Combinación de reglas de comportamientos para evitar obstáculos y alcanzar objetivos pero no realiza pruebas de campo
[59]	2 conjuntos de cámaras estéreo	DEM y una rejilla de ocupación	Creación de mapa, análisis de transitabilidad y navegación	Entornos planetarios	Realizado en movimiento, pero la velocidad del robot es de 0.1 m/s
[60]	8 sensores s3nar	Rejilla de transitabilidad	Detección de obstaculos, análisis de transitabilidad y planificación de trayectoria	Terreno accidentado	Planificaci3n de trayectorias y navegaci3n, usa <i>stop-scan-go</i>
[61]	Tel3metro l3ser Erim 64x256	DEM	Análisis de transitabilidad y planificación de trayectoria y navegaci3n	Terreno accidentado	Navegaci3n aut3noma reactiva a 2 m/s

Tabla A.1: Listado de trabajos relacionados (continúa en página siguiente).

Ref.	Sensor	Mapa 2D	Procedimientos	Entorno	Funcionamiento
[62]	LIDAR Velodyne PUCK VLP-16	Mapa de ocupación	Construcción de mapa, análisis de transitabilidad y planificación de camino	Fuera de camino	Construcción de mapa global incremental
[74]	Láser 3D y cámaras estéreo	DEM	Clasificación de puntos, construcción de mapa y navegación autónoma	Terreno accidentado	Velocidades entre 0.8 y 1.9 m/s
[79]	Cámaras estéreo	Mapa de transitabilidad y de ocupación	Análisis de transitabilidad y navegación	Entornos planetarios	Planificación fuera de línea
[82]	Láser basado en 3D UTM-30 LX-E	DEM en coordenadas cilíndricas	Creación de mapa local, análisis de transitabilidad y navegación	Entornos planetarios	Realiza todo en línea y navega a velocidad promedio de 0.0287 m/s
[92]	Cámara Kinect RGB-D	Emplea puntos 3D clasificados	Análisis de transitabilidad de puntos 3D	Entorno agrícola	No realiza navegación ni construcción de mapas
[93]	Láser 3D basado en escáner Sick	Rejilla de ocupación	Análisis de transitabilidad	Entorno natural	Barridos obtenidos con <i>stop-scan-go</i> , no realiza navegación
[94]	LIDAR Velodyne HDL-32E	Mapa de ocupación con transformación de distribución normal (NDT-OM)	Análisis de transitabilidad, creación de mapa y navegación	Entornos naturales	Navegación autónoma a velocidades entre 0.6 y 1.5 m/s
[96]	Cámara Tango RGB-D	Mapa de alturas	Análisis de transitabilidad	Terrenos no estructurados	Realiza entrenamiento de clasificadores en entorno virtual. No realiza navegación

Tabla A.1: Listado de trabajos relacionados (continúa en página siguiente).

Ref.	Sensor	Mapa 2D	Procedimientos	Entorno	Funcionamiento
[110]	Láser 3D con Hokuyo URG-04LX y cámaras estéreo	Rejilla de celdas	Análisis de transitabilidad y generación de mapas	Entorno planetarios	No realiza navegación
[119]	2 LIDARs	<i>Dense-DEM</i>	Análisis de transitabilidad, detección de objetos dinámicos, cálculo de mapa de coste	Terreno accidentado	Emplea aceleración GPU

Tabla A.1: Listado de trabajos relacionados.

Referencias

- [1] J. A. Marshall. *Mining Robotics*, pages 1–6. Springer Berlin Heidelberg, 2020. [CrossRef].
- [2] B. Shah and H. Choset. Survey on urban search and rescue robots. *Journal of the Robotics Society of Japan*, 22(5):582–586, 2004. [CrossRef].
- [3] A. Sinha and P. Papadakis. Mind the gap: detection and traversability analysis of terrain gaps using LIDAR for safe robot navigation. *Robotica*, 31:1085–1101, 2013. [CrossRef].
- [4] U. Weiss and P. Biber. Plant detection and mapping for agricultural robots using a 3D LIDAR sensor. *Robotics and Autonomous Systems*, 59(5):265–273, 2011. [CrossRef].
- [5] S. S. H. Hajjaj and K. S. M. Sahari. Review of research in the area of agriculture mobile robots. In *The 8th International Conference on Robotic, Vision, Signal Processing & Power Applications*, pages 107–117. Springer, Singapore, 2014. [CrossRef].
- [6] J. Khurshid and H. Bing-Rong. Military robots-a glimpse from today and tomorrow. In *ICARCV 2004 8th Control, Automation, Robotics and Vision Conference, 2004.*, volume 1, pages 771–777. IEEE, 2004. [CrossRef].
- [7] J. Larson and M. Trivedi. Lidar based off-road negative obstacle detection and analysis. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), Washington, DC, USA*. IEEE, oct 2011. [CrossRef].

-
- [8] W. Xu, B. Liang, and Y. Xu. Survey of modeling, planning, and ground verification of space robotic systems. *Acta Astronautica*, 68(11-12):1629–1649, 2011. [CrossRef].
- [9] C. H. Tong, T. D. Barfoot, and E. Dupuis. Three-dimensional SLAM for mapping planetary work site environments. *Journal of Field Robotics*, 29(3):381–412, jan 2012. [CrossRef].
- [10] A. Tsalatsanis, K.P. Valavanis, A. Kandel, and A. Yalcin. Multiple sensor based UGV localization using fuzzy extended Kalman filtering. In *2007 Mediterranean Conference on Control & Automation*, pages 1–8. IEEE, 2007. [CrossRef].
- [11] L. Matthies, T. Litwin, K. Owens, A. Rankin, K. Murphy, D. Coombs, J. Gillsinn, Tsai hong, S. Legowik, M. Nashman, and B. Yoshimi. Performance evaluation of UGV obstacle detection with CCD/FLIR stereo vision and LADAR. In *Proceedings of the 1998 IEEE International Symposium on Intelligent Control (ISIC), Intell, Gaithersburg, MD, USA*, pages 658–670. IEEE, 1998. [CrossRef].
- [12] K. Cho, S. Baeg, and S. Park. Natural terrain detection and SLAM using LIDAR for UGV. In *Advances in Intelligent Systems and Computing*, pages 793–805. Springer, 2013. [CrossRef].
- [13] P. Papadakis. Terrain traversability analysis methods for unmanned ground vehicles: A survey. *Eng. Appl. Artif. Intel.*, 26(4):1373–1385, 2013. [CrossRef].
- [14] A. Phunopas, A. Blattler, and N. Pudchuen. iRAP robot: World RoboCup rescue championship 2016 and best in class mobility award. In *RoboCup 2016: Robot World Cup XX*, pages 554–564. Springer, 2016. [CrossRef].
- [15] T. Mueller-Sim, M. Jenkins, J. Abel, and G. Kantor. The robotanist: a ground-based agricultural robot for high-throughput crop phenotyping. In *2017 IEEE*

- International Conference on Robotics and Automation (ICRA), Singapore*, pages 3634–3639, 2017. [CrossRef].
- [16] R. Volpe, T. Estlin, S. Laubach, C. Olson, and J. Balam. Enhanced mars rover navigation techniques. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), San Francisco, CA, USA*, volume 1, pages 926–931, 2000. [CrossRef].
- [17] F. Sanfilippo, J. Azpiazu, G. Marafioti, A. A. Transeth, Ø. Stavdahl, and P. Liljebäck. Perception-driven obstacle-aided locomotion for snake robots: the state of the art, challenges and possibilities. *Applied Sciences*, 7:336, 2017. [CrossRef].
- [18] K. Daniec, K. Jedrasiak, R. Koterak, and A. Nawrat. Embedded micro inertial navigation system. In *Applied mechanics and materials*, volume 249, pages 1234–1246. Trans Tech Publ, 2013. [CrossRef].
- [19] B. Tian, V. A. Shim, M. Yuan, C. Srinivasan, H. Tang, and H. Li. RGB-d based cognitive map building and navigation. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan*, pages 1562–1567, 2013. [CrossRef].
- [20] Y. Cong, X. Li, J. Liu, and Y. Tang. A stairway detection algorithm based on vision for UGV stair climbing. In *2008 IEEE International Conference on Networking, Sensing and Control, Sanya, China*, pages 1806–1811, 2008. [CrossRef].
- [21] A. Downs, R. Madhavan, and T. Hong. Temporal range registration for unmanned ground and aerial vehicles. *Journal of Intelligent and Robotic Systems*, 44(1):47–69, 2005. [CrossRef].
- [22] S. Liu, M. M. Atia, T. B. Karamat, and A. Noureldin. A LiDAR-aided indoor

- navigation system for UGVs. *Journal of Navigation*, 68(2):253–273, 2015. [CrossRef].
- [23] Q. Li, L. Zhang, Q. Mao, Q. Zou, P. Zhang, S. Feng, and W. Ochieng. Motion field estimation for a dynamic scene using a 3D LiDAR. *Sensors*, 14(9):16672–16691, 2014. [CrossRef].
- [24] M. Agrawal and K. Konolige. Real-time localization in outdoor environments using stereo vision and inexpensive GPS. In *18th International Conference on Pattern Recognition (ICPR'06), Hong Kong*, volume 3, pages 1063–1068, 2006. [CrossRef].
- [25] G. A. Demetriou. A survey of sensors for localization of unmanned ground vehicles (UGVs). In *IC-AI*, pages 659–668, 2006. [CrossRef].
- [26] T. Chu, N. Guo, S. Backén, and D. Akos. Monocular camera/IMU/GNSS integration for ground vehicle navigation in challenging GNSS environments. *Sensors*, 12(3):3162–3185, 2012. [CrossRef].
- [27] J. A. Bagnell, D. Bradley, D. Silver, B. Sofman, and A. Stentz. Learning for autonomous navigation. *IEEE Robot. Autom. Mag.*, 17(2):74–84, 2010. [CrossRef].
- [28] B. Suger, B. Steder, and W. Burgard. Traversability analysis for mobile robots in outdoor environments: A semi-supervised learning approach based on 3D-LIDAR data. In *IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA*, pages 3941–3946, 2015. [CrossRef].
- [29] I. Kostavelis, L. Nalpantidis, and A. Gasteratos. Supervised traversability learning for robot navigation. *Lect. Notes Artif. Int.*, 6856:289–298, 2011. [CrossRef].
- [30] A. Santamaria-Navarro, E. H. Teniente, M. Morta, and J. Andrade-Cetto.

- Terrain classification in complex three-dimensional outdoor environments. *J. Field Robot.*, 32:42–60, 2015. [CrossRef].
- [31] C. Ye and J. Borenstein. T-transformation: Traversability analysis for navigation on rugged terrain. In *In Proceedings of the Defense and Security Symposium, Orlando, USA*, pages 473–483, 2004. [CrossRef].
- [32] S. Thrun, M. Montemerlo, and A. Aron. Probabilistic terrain analysis for high-speed desert driving. In *In Proceedings of Robotics: Science and Systems II, Philadelphia, USA*, pages 1–7, 2006. [CrossRef].
- [33] C. Pang, X. Zhong, H. Hu, J. Tian, . Peng, and J. Zeng. Adaptive obstacle detection for mobile robots in urban environments using downward-looking 2D LiDAR. *Sensors*, 18(6):1749, 2018. [CrossRef].
- [34] Q. Zhu, J. Wu, H. Hu, C. Xiao, and W. Chen. LIDAR point cloud registration for sensing and reconstruction of unstructured terrain. *Appl. Sci.*, 8(11):2318, 2018. [CrossRef].
- [35] L. Chen, J. Yang, and H. Kong. LIDAR-histogram for fast road and obstacle detection. In *Proceedings of the IEEE International Conference on Robotics and Automation, Marina Bay Sands, Singapore*, pages 1343–1348, 2017. [CrossRef].
- [36] K. Zhang, Y. Yang, M. Fu, and M. Wang. Traversability assessment and trajectory planning of unmanned ground vehicles with suspension systems on rough terrain. *Sensors*, 19(20):4372, 2019. [CrossRef].
- [37] J. L. Martínez, A. Mandow, A. J. Reina, T.J. Cantador, J. Morales, and A. García-Cerezo. Navigability analysis of natural terrains with fuzzy elevation maps from ground-based 3D range scans. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan*, pages 1576–1581, 2013. [CrossRef].

- [38] H. Almqvist, M. Magnusson, and A. Lilienthal. Improving point cloud accuracy obtained from a moving platform for consistent pile attack pose estimation. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 75:101–128, 2014. [CrossRef].
- [39] D. Droeschel, M. Schwarz, and S. Behnke. Continuous mapping and localization for autonomous navigation in rough terrain using a 3D laser scanner. *Robot. Auton. Syst.*, 88:104–115, 2017. [CrossRef].
- [40] L. Chen, D. Hoang, H. Lin, and T. Nguyen. Innovative methodology for multi-view point cloud registration in robotic 3D object scanning and reconstruction. *Applied Sciences*, 6:132, 2016. [CrossRef].
- [41] P. Krusi, P. Furgale, M. Bosse, and R. Siegwart. Driving on point clouds: Motion planning, trajectory optimization, and terrain assessment in generic nonplanar environments. *J. Field Robot.*, 34(5):940–984, 2017. [CrossRef].
- [42] H. Macher, T. Landes, and P. Grussenmeyer. From point clouds to building information models: 3D semi-automatic reconstruction of indoors of existing buildings. *Applied Sciences*, 7:1030, 2017. [CrossRef].
- [43] J. Lalonde, N. Vandapel, D. F Huber, and M. Hebert. Natural terrain classification using three-dimensional ladar data for ground robot mobility. *J. Field Robot.*, 23(10):839–861, 2006. [CrossRef].
- [44] V. Plaza-Leiva, J. A. Gomez-Ruiz, A. Mandow, and A. García-Cerezo. Voxel-based neighborhood for spatial shape pattern classification of LIDAR point clouds with supervised learning. *Sensors*, 17(594):1–17, 2017. [CrossRef].
- [45] J. Sock, J. Kim, J. Min, and K. Kwak. Probabilistic traversability map generation using 3D-LIDAR and camera. In *2016 IEEE International Conference on Robotics and Automation (ICRA, Stockholm, Sweden)*, pages 5631–5637, 2016. [CrossRef].

- [46] K. Janschek, V. Janschek, and M. Beck. Performance analysis for visual planetary landing navigation using optical flow and DEM matching. In *AIAA Guidance, Navigation, and Control Conference and Exhibit, Dresden, Germany*, page 6706, 2006. [CrossRef].
- [47] I. Rekleitis, J.L. Bedwani, E. Dupuis, T. Lamarche, and P. Allard. Autonomous over-the-horizon navigation using LIDAR data. *Autonomous Robots*, 34(1-2):1–18, 2013. [CrossRef].
- [48] M. Schadler, J. Stückler, and S. Behnke. Rough terrain 3D mapping and navigation using a continuously rotating 2D laser scanner. *Künstliche Intelligenz*, 28(2):93–99, 2014. [CrossRef].
- [49] R. Zlot and M. Bosse. Efficient large-scale three-dimensional mobile mapping for underground mines. *Journal of Field Robotics*, 31:731–752, 2014. [CrossRef].
- [50] Y. Yi, F. Mengyin, Y. Xin, X. Guangming, and J.W. Gong. Autonomous ground vehicle navigation method in complex environment. In *IEEE Intelligent Vehicles Symposium*, pages 1060–1065, San Diego, USA, 2010. [CrossRef].
- [51] J. L. Martínez, M. Morán, J. Morales, A. J. Reina, and M. Zafra. Field navigation using fuzzy elevation maps built with local 3D laser scans. *Appl. Sci.*, 8(3):1–18, 2018. [CrossRef].
- [52] N. Pérez-Higueras, A. Jardón, A. Rodríguez, and C. Balaguer. 3D exploration and navigation with optimal-RRT planners for ground robots in indoor incidents. *Sensors*, 20(1):220, 2020. [CrossRef].
- [53] B. K. Patle, G. Babu-L, A. Pandey, D. R. K. Parhi, and A. Jagadeesh. A review: On path planning strategies for navigation of mobile robot. *Defense Technol.*, 15(4):582–606, 2019. [CrossRef].

- [54] B. Sebastian and P. Ben-Tzvi. Physics based path planning for autonomous tracked vehicle in challenging terrain. *J. Intel. Robot. Sys.*, 95:511–526, 2019. [CrossRef].
- [55] H. Schäfer, A. Hach, M. Proetzsch, and K. Berns. 3d obstacle detection and avoidance in vegetated off-road terrain, pasadena, ca, usa. In *IEEE International Conference on Robotics and Automation*, pages 923–928, 2008. [CrossRef].
- [56] J. J. Biesiadecki, P. C. Leger, and M. W. Maimone. Tradeoffs between directed and autonomous driving on the Mars exploration rovers. *Int. J. Robot. Res.*, 26(1):91–104, 2007. [CrossRef].
- [57] C. Castejon, D. Blanco, and L. Moreno. Compact modeling technique for outdoor navigation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 38(1):9–24, 2008. [CrossRef].
- [58] H. Seraji. Smartnav: A rule-free fuzzy approach to rover navigation. *Journal of Field Robotics*, 22(12):795–808, 2005. [CrossRef].
- [59] S. Lacroix, A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury, M. Herrb, and R. Chatila. Autonomous rover navigation on unknown terrains: Functions and integration. *The International Journal of Robotics Research*, 21(10-11):917–942, 2002. [CrossRef].
- [60] A. Howard, Ho. Seraji, and B. Werger. Global and regional path planners for integrated planning and navigation. *Journal of Field Robotics*, 22(12):767–778, 2005. [CrossRef].
- [61] D. Langer, J. Rosenblatt, and M. Hebert. An integrated system for autonomous off-road navigation. In *IEEE International Conference on Robotics and Automation, San Diego, USA*, pages 414–419, 1994. [CrossRef].
- [62] A. Pfrunder, P. V. K. Borges, A. R. Romero, G. Catt, and A. Elfes. Real-time autonomous ground vehicle navigation in heterogeneous environments using a

- 3D LiDAR. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, Canada*, pages 2601–2608, 2017. [CrossRef].
- [63] M. Quigley, B. Gerkey, K. Conley J., Faust T., Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. ROS: an open-source robot operating system. In *IEEE International Conference on Robotics and Automation: Workshop on Open Source Software (ICRA), Kobe, Japan*, 2009. [CrossRef].
- [64] J. L. Martínez. Proyecto de investigación de excelencia de la Junta de Andalucía P10-TEP-6101-R. “Navegación autónoma de un robot móvil 4x4 en entornos naturales mediante GPS diferencial y telémetro láser tridimensional”, 2013. [Web del proyecto].
- [65] O. Cordero. Puesta en marcha y documentación del robot móvil para exteriores Andábata, Proyecto Fin de carrera de Ingeniero Industrial, 2016. [Documento].
- [66] J. L. Martínez, J. Morales, A. J. Reina, A. Mandow, A. Pequeño-Boter, and A. García-Cerezo. Construction and calibration of a low-cost 3D laser scanner with 360° field of view for mobile robots. In *IEEE International Conference on Industrial Technology (ICIT), Seville, Spain*, pages 149–154, 2015. [CrossRef].
- [67] P. Bovbel and F. Andargie. Android-based mobile robotics platform, Mover-Bot, 2012. [Enlace a App].
- [68] C. Rockey and A. Furlan. ROS Android Sensors Driver, 2012. [Github del proyecto].
- [69] A. Mandow, J. L. Martínez, J. Morales, J. L. Blanco, A. García-Cerezo, and J. González. Experimental kinematics for wheeled skid-steer mobile robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, USA*, pages 1222 – 1227, 2007. [CrossRef].
- [70] J. L. Martínez, J. Morales, and M. Zafra. Obtención de barridos láser 3D

- nivelados con el robot móvil Andábata en movimiento. In *XXXVII Jornadas de Automática, Madrid, Spain*, pages 661–666, 2016. [CrossRef].
- [71] T. Moore and D. Stouch. A generalized extended Kalman filter implementation for the robot operating system. In *Intelligent Autonomous Systems 13*, pages 335–348. Springer International Publishing, 2015. [CrossRef].
- [72] B. M. Leedy, J. S. Putney, C. Bauman, S. Cacciola, J.M. Webster, and C. F. Reinholtz. Virginia tech’s twin contenders: A comparative study of reactive and deliberative navigation. In *The 2005 DARPA Grand Challenge*, pages 155–182. Springer, 2007. [CrossRef].
- [73] J. Morales, J.L. Martínez, A. Mandow, A.J. García-Cerezo, and S. Pedraza. Power consumption modeling of skid-steer tracked mobile robots on rigid terrain. *IEEE Transactions on Robotics*, 25(5):1098–1108, 2009. [CrossRef].
- [74] D. Silver, B. Sofman, N. Vandapel, J A. Bagnell, and A. Stentz. Experimental analysis of overhead data processing to support long range navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China*, pages 2443–2450, 2006. [CrossRef].
- [75] E. Marder-Eppstein and V. Pradeep. Ros actionlib, 2013. [Github del proyecto].
- [76] K. Song and J. Tai. Fuzzy navigation of a mobile robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Raleigh, NC, USA*, pages 621–627. IEEE, 1992. [CrossRef].
- [77] A. Saffiotti. The uses of fuzzy logic in autonomous robot navigation. *Soft Computing*, 1(4):180–197, 1997. [CrossRef].
- [78] P. Rusu, E.M. Petriu, T.E. Whalen, A. Cornell, and H.J.W. Spoelder. Behavior-based neuro-fuzzy controller for mobile robot navigation. *IEEE*

- Transactions on Instrumentation and Measurement*, 52(4):1335–1340, 2003. [CrossRef].
- [79] A. Howard, H. Seraji, and B. Werger. Fuzzy terrain-based path planning for planetary rovers. In *2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No.02CH37291)*. IEEE, 2002. [CrossRef].
- [80] M. A. Porta, O. Montiel, O. Castillo, R. Sepúlveda, and P. Melin. Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. *Applied Soft Computing*, 9(3):1102–1110, 2009. [CrossRef].
- [81] C. Brenneke, O. Wulf, and B. Wagner. Using 3D laser range data for SLAM in outdoor environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA*, volume 1, pages 188–193, 2003. [CrossRef].
- [82] Ishigami G., Otsuki M., and Kubota T. Path planning and navigation framework for a planetary exploration rover using a laser range finder. In *Springer Tracts in Advanced Robotics*, pages 431–447. Springer Berlin Heidelberg, dec 2013. [CrossRef].
- [83] P. Pfaff, R. Triebel, and W. Burgard. An efficient extension to elevation maps for outdoor terrain mapping and loop closing. *The International Journal of Robotics Research*, 26(2):217–230, 2007. [CrossRef].
- [84] A. Mandow, T.-J. Cantador, A. García-Cerezo, A. J. Reina, J.-L. Martínez, and J. Morales. Fuzzy modeling of natural terrain elevation from a 3D scanner point cloud. In *In Proceedings of the 7th IEEE International Symposium on Intelligent Signal Processing (WISP), Floriana, Malta*, pages 171–175, 2011. [CrossRef].

- [85] J. Hou, Y. Chang, M. Hsu, S. Lee, and C. Wu. Construction of fuzzy map for autonomous mobile robots based on fuzzy confidence model. *Mathematical Problems in Engineering*, 2014:1–8, 2014. [CrossRef].
- [86] A. Mandow, T. J. Cantador, A. J. Reina, J. L. Martínez, J. Morales, and A. García-Cerezo. Building fuzzy elevation maps from a ground-based 3D laser scan for outdoor mobile robots. In *Advances in Intelligent Systems and Computing*, volume 417, pages 29–41. Springer, 2016. [CrossRef].
- [87] A. J. Reina, J. L. Martínez, A. Mandow, J. Morales, and A. García-Cerezo. Collapsible cubes: Removing overhangs from 3D point clouds to build local navigable elevation maps. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Besançon, France*, pages 1012–1017, 2014. [CrossRef].
- [88] A. García-Cerezo, M. J. López-Baldán, and A. Mandow. An efficient least squares fuzzy modelling method for dynamic systems. In *In Proceedings of the IMACS Multiconference on Computational Engineering in Systems Applications (CESA), Symposium on Modelling, Analysis and Simulation, Lille, France*, pages 885–890, 1996. [Pascal-Francis].
- [89] Alphabet Inc. Google Earth, 2005. [Enlace Web].
- [90] A. Pomares, J. L. Martínez, A. Mandow, M. A. Martínez, M. Morán, and J. Morales. Ground extraction from 3D LIDAR point clouds with the classification learner app. In *In Proceedings of the 26th Mediterranean Conference on Control and Automation (MED), Zadar, Croatia*, pages 400–405, 2018. [CrossRef].
- [91] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel. On the segmentation of 3D LIDAR point clouds. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China*, pages 2798–2805, 2011. [CrossRef].

- [92] M. Bellone, G. Reina, N.I. Giannoccaro, and L. Spedicato. 3D traversability awareness for rough terrain mobile robots. *Sensor Rev.*, 34(2):220–232, 2014. [CrossRef].
- [93] S. K. Reddy and Prabir K. Pal. Computing an unevenness field from 3D laser range data to obtain traversable region around a mobile robot. *Robot. Auton. Syst.*, 84:48–63, 2016. [CrossRef].
- [94] J. Ahtiainen, T. Stoyanov, and J. Saarinen. Normal distributions transform traversability maps: Lidar-only approach for traversability mapping in outdoor environments. *J. Field Robot.*, 34(3):600–621, 2017. [CrossRef].
- [95] M. Bellone, G. Reina, L. Caltagirone, and M. Wahde. Learning traversability from point clouds in challenging scenarios. *IEEE T. Intell. Transp. Syst.*, 19(1):296–305, 2018. [CrossRef].
- [96] R. O. Chavez-Garcia, J. Guzzi, L. M. Gambardella, and A. Giusti. Learning ground traversability from simulations. *IEEE Robot. Autom. Lett.*, 3(3):1695–1702, 2018. [CrossRef].
- [97] J. L. Martínez, M. Morán, J. Morales, A. Robles, and M. Sánchez. Supervised learning of natural-terrain traversability with synthetic 3D laser scans. *Appl. Sci.*, 10(1140):1–12, 2020. [CrossRef].
- [98] H. Vu, H. T. Nguyen, P. M. Chu, W. Zhang, S. Cho, Y. W. Park, and K. Cho. Adaptive ground segmentation method for real-time mobile robot control. *Int. J. Adv. Robot. Syst.*, 14(6), 2017. [CrossRef].
- [99] M. Kragh, R.N. Jorgensen, and H. Pedersen. Object detection and terrain classification in agricultural fields using 3D LIDAR data. *Lect. Notes Comput. Sci.*, 9163:188–197, 2015. [CrossRef].
- [100] A. Cuenca, U. Ojha, J. Salt, and M. Chow. A non-uniform multi-rate control

- strategy for a Markov chain-driven Networked Control System. *Information Sciences*, 321:31–47, 2015. [CrossRef].
- [101] T. Furukawa, B. Lavis, and H. F. Durrant-Whyte. Parallel grid-based recursive bayesian estimation using GPU for real-time autonomous navigation. In *IEEE International Conference on Robotics and Automation, Anchorage, AK, USA*, pages 316–321. IEEE, 2010. [CrossRef].
- [102] C. Qingyang, S. Zhenping, L. Daxue, F. Yuqiang, and L. Xiaohui. Local path planning for an unmanned ground vehicle based on SVM. *International Journal of Advanced Robotic Systems*, 9(6):246, 2012. [CrossRef].
- [103] S. Lee and D. Kwak. A terrain classification method for ugv autonomous navigation based on SURF. In *8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Incheon, South Korea*, pages 303–306. IEEE, 2011. [CrossRef].
- [104] P. Moghadam and W. S. Wijesoma. Online, self-supervised vision-based terrain classification in unstructured environments. In *2009 IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX, USA*, pages 3100–3105. IEEE, 2009. [CrossRef].
- [105] H. Liu, G. He, H. Yu, Y. Zhuang, and W. Wang. Fast 3D scene segmentation and classification with sequential 2D laser scanning data in urban environments. In *2016 35th Chinese Control Conference (CCC), Chengdu, China*, pages 7131–7136. IEEE, 2016. [CrossRef].
- [106] J. Balado, L. Díaz-Vilariño, P. Arias, and H. González-Jorge. Automatic classification of urban ground elements from mobile laser scanning data. *Automation in Construction*, 86:226–239, 2018. [CrossRef].
- [107] X. Xiong, D. Munoz, J.A. Bagnell, and M. Hebert. 3-D scene analysis via sequenced predictions over points and regions. In *Proceedings of the IEEE Inter-*

- national Conference on Robotics and Automation (ICRA), Shanghai, China*, pages 2609–2616, 2011. [CrossRef].
- [108] M. Sánchez, J. L. Martínez, J. Morales, A. Robles, and M. Morán. Automatic generation of labeled 3D point clouds of natural environments with Gazebo. *In Proc. IEEE International Conference on Mechatronics, Ilmenau, Germany*, 2019. [CrossRef].
- [109] K. Doherty, T. Shan, J. Wang, and B. Englot. Learning-aided 3D occupancy mapping with bayesian generalized kernel inference. *IEEE Transactions on Robotics*, 35(4):953–966, 2019. [CrossRef].
- [110] R. A. Hewitt, . Ellery, and A. de Ruiter. Training a terrain traversability classifier for a planetary rover through simulation. *International Journal of Advanced Robotic Systems*, 14(5):1–14, 2017. [CrossRef].
- [111] N. Koenig and A. Howard. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), Sendai, Japan*. IEEE, 2004. [CrossRef].
- [112] A. Pomares. Clasificación de puntos 3D para evaluar la navegabilidad del terreno con las herramientas de aprendizaje por computadora de Matlab, Trabajo Final del Máster en Ingeniería Mecatrónica, 2017.
- [113] Documentación de entorno de simulación Gazebo, 2014. [Enlace Web].
- [114] Robotics system toolbox for Matlab. [Enlace librería].
- [115] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12:2825–2830, 2011. [CrossRef].

-
- [116] Jean-Fransois Lalonde, Nicolas Vandapel, and Martial Hebert. Data structures for efficient dynamic processing in 3-D. *The International Journal of Robotics Research*, 26(8):777–796, 2007.
- [117] S. K. Lam, A. Pitrou, and S. Seibert. Numba: A LLVM-Based Python JIT Compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, LLVM '15*, New York, NY, USA, 2015. Association for Computing Machinery. [Crossref].
- [118] Multiprocessing. Process based parallelism, 2001. [Web del proyecto].
- [119] Y. Pan, X. Xu, Y. Wang, X. Ding, and R. Xiong. GPU accelerated real-time traversability mapping. In *In Proceeding of the IEEE International Conference on Robotics and Biomimetics, Dali, China*, pages 734–740, 2019. [CrossRef].
- [120] E. Rohmer, S. P. N. Singh, and M. Freese. V-rep: a versatile and scalable robot simulation framework. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan*, pages 1321–1326, 2013. [CrossRef].
- [121] J. Hsu, N. Koenig, and D. Coleman. Gazebo ROS pkgs, 2015. [Web del proyecto].
- [122] J. L. Martínez, J. Morales, M. Sánchez, M. Morán, A. Reina, and J. Fernández-Lozano. Reactive navigation on natural environments by continuous classification of ground traversability. *Sensors*, 20(22):6423, 2020. [CrossRef].
- [123] S. Kohlbrecher and J. Meyer. Hector gazebo plugins, 2013. [Web del proyecto].
- [124] ZED-F9P module u-blox F9 high precision GNSS module, 2018. [Datasheet].
- [125] A. Rankin, M. Bajracharya, A. Huertas, A. Howard, B. Moghaddam, S. Brennan, A. Ansar, B. Tang, M. Turmon, and L. Matthies. Stereo-vision-based perception capabilities developed during the robotics collaborative technology

alliances program. In *In Proceedings of the SPIE Defense, Security, and Sensing, Orlando, USA*, pages 1–15, 2010. [CrossRef].



UNIVERSIDAD
DE MÁLAGA