

JORNADAS de ROBÓTICA, EDUCACIÓN y BIOINGENIERÍA 2022

XII Jornadas Nacionales de Robótica
VIII Seminario de Educación en Automática
XIII Simposio CEA de Bioingeniería

Málaga, 18-20 de mayo, 2022



UNIVERSIDAD
DE MÁLAGA



DEPARTAMENTO DE
INGENIERÍA DE SISTEMAS
Y AUTOMÁTICA



CEA
Comité Español de Automática

Colaboran



Libro de actas

de las

JORNADAS de ROBÓTICA, EDUCACIÓN y BIOINGENIERÍA 2022

XII Jornadas Nacionales de Robótica
VIII Seminario de Educación en Automática
XIII Simposio CEA de Bioingeniería

Málaga, 18-20 de mayo, 2022

Edita: Dpto. de Ingeniería de Sistemas y Automática
Universidad de Málaga

ISBN: 978-84-09-41095-8

Índice

COMUNICACIONES

UNA NUEVA OPORTUNIDAD DE ATENCIÓN A LAS PERSONAS MAYORES MEDIANTE ROBÓTICA SOCIAL EN LAS RESIDENCIAS LACORT (UNO MÁS) <i>Eduardo Zalama, Jaime Gómez-García-Bermejo, Pablo Francisco Viñas, Luis Cobo Hurtado, José María Delgado Sanz, Beatriz Vielba</i>	4
ROBOT DE CABLES PARA LA LIMPIEZA DE VENTANAS <i>S. Santamaría, E. Zalama, J. Gómez</i>	12
HACIA LA RECOGIDA DE RESIDUOS DOMESTICOS EN EXTERIORES: ENFOQUE VISUAL-TACTIL <i>P. Gil, S.T. Puente, J. Castaño-Amorós, I. L. Páez-Ubieta</i>	17
SISTEMA ROBOTICO INTELIGENTE DE EXPLORACION TERMICA PARA MISIONES DE BUSQUEDA Y RESCATE <i>Christyan Cruz Ulloa, Guillermo Prieto, Jaime del Cerro, Antonio Barrientos</i>	25
APLICANDO HERRAMIENTAS PSICOLÓGICAS PARA INCREMENTAR EL VÍNCULO ENTRE LOS USUARIOS Y LOS ROBOTS SOCIALES <i>Fernando Alonso Martín, Sara Carrasco Martínez, Marcos Maroto Gómez, Javier Sevilla Salcedo, Miguel Á. Salichs</i>	32
ROMERIN: ORGANISMO ROBÓTICO ESCALADOR BASADO EN PATAS MODULARES CON VENTOSAS ACTIVAS <i>Carlos Prados, Miguel Hernando, Ernesto Gambao, Alberto Brunete</i>	40
SISTEMA ROBÓTICO MÓVIL PARA EL ANÁLISIS DE LA MARCHA HUMANA SIN MARCADORES E IDENTIFICACIÓN DE PATOLOGÍAS NEUROLÓGICAS, ROBOGAI <i>Diego Guffanti, Alberto Brunete, Miguel Hernando, Ernesto Gambao, David Álvarez</i>	50
CREACIÓN DE MAPAS DE NAVEGACIÓN PARA UN ROBOT TREPADOR DE ESTRUCTURAS RETICULARES <i>F.J. Soler, A. Peidró, M. Fabregat, L. Payá, O. Reinoso</i>	57
ANÁLISIS Y VALIDACIÓN DE UN EXOESQUELETO HÍBRIDO PARA REHABILITACIÓN DE LAS EXTREMIDADES SUPERIORES <i>J. Carrillo-Ríos, D. Cantalejo-Escobar, D. Pont-Esteban, A. Contreras González, M. A. Sánchez-Urán, M. Ferre</i>	64
ORIGAMI KRESLING PATTERN AS SOFT SCALABLE LINK <i>L. Mena, C.A. Monje, C. Balaguer</i>	70
SISTEMA DE MONITORIZACIÓN NO INTRUSIVA PARA EL RECONOCIMIENTO DE ACTIVIDADES DE LA VIDA DIARIA EN ENTORNOS MULTIUSUARIO <i>R.G. Ramos, J.D. Domingo, E. Zalama, J. G. García-Bermejo</i>	75
MÉTODO DE ODOMETRÍA BASADA EN PLANOS PARA CÁMARAS DE PROFUNDIDAD <i>Andres Galeote-Luque, Jose-Raul Ruiz-Sarmiento, Javier Gonzalez-Jimenez</i>	83
INTEGRACIÓN DE SMARTPHONES 5G EN REDES DE SENSORES DISTRIBUIDAS PARA ROBÓTICA DE EXTERIORES MEDIANTE ROS Y ANDROID <i>Germán Ruiz Mudarra, Juan Bravo Arrabal, J.J. Fernández Lozano, Alfonso García Cerezo</i>	91

DISEÑO DE LA CONFIGURACIÓN DE BRAZOS ROBÓTICOS VESTIBLES PARA TAREAS DE INSTALACIÓN Y MANTENIMIENTO <i>J. Álvarez-Pastor, D. Martínez-Pascual, A. Blanco, L.D. Lledó, C. Fernández-Irles, N. García-Aracil, F. López-Labrador, Á. Bernal-Plaza</i>	100
HIMTAE: SISTEMA HETEROGÉNEO MULTIROBOT PARA AYUDA DE PERSONAS MAYORES EN UN AMBIENTE ASISTIDO EN EL HOGAR <i>Ramón Barber, Francisco J. Ortiz, Francisco M. Calatrava, Santiago Garrido, Luis Moreno, José Alfonso Vera, Alicia Mora, Adrián Prados, Joaquín Roca, Manuel Jiménez, Inmaculada Méndez, Cecilia Ruiz-Esteban, Óscar M. Mozos</i>	106
EXPLORACIÓN DE ENTORNOS DESCONOCIDOS MEDIANTE VEHÍCULOS TERRESTRES BASADA EN LA ENTROPÍA <i>Jaime Godoy Calvo, Dahui Lin Yan y Ricardo Vázquez-Martín</i>	118
IDENTIFICACIÓN DE UN CUELLO ROBÓTICO BLANDO MEDIANTE APRENDIZAJE AUTOMÁTICO <i>Nicole Continelli, Luis Nagua, Concepción A. Monje y Carlos Balaguer</i>	124
ROBOMINER: ROBOT MODULAR RESILIENTE PARA OPERACIONES MINERAS <i>V. Gómez, D. Collados, E. Aguado, M. Hernando, R. Sanz, C. Rossi</i>	132
AUTOMATIZACIÓN DEL PEGADO DE LAS GARRAS MAGNÉTICAS DE UN ROBOT TREPADOR BÍPEDO <i>Marc Fabregat, Adrián Peidró, Óscar Reinoso, Francisco José Soler, Luis Miguel Jiménez</i>	139
EVALUACIÓN AUTOMATIZADA DE LA COORDINACIÓN Y DESTREZA MANUAL FINA USANDO SERIOUS GAMES <i>E.D. Oña, C. Balaguer, y A. Jardón</i>	149
MÉTODO DE PLANIFICACIÓN DE TRAYECTORIAS Y NAVEGACIÓN PARA UN ROBOT AUTÓNOMO SUBTERRÁNEO (uSLAM) <i>E. Menéndez, J. Yagüez, S. Martínez, C. Balaguer</i>	156
DETECCIÓN DE PARTES DEL CUERPO EN IMÁGENES MULTIMODALES DE BÚSQUEDA Y RESCATE <i>Alejandro González Centeno, Ricardo Vázquez-Martín, Anthony Mandow y Alfonso García-Cerezo</i>	161
RESÚMENES EXTENDIDOS	
DETECCIÓN Y RECOGIDA DE MUESTRAS POR VEHÍCULOS DE EXPLORACIÓN PLANETARIA <i>Laura M. Mantoani, Raúl Castilla Arquillo, Gonzalo J. Paz Delgado, Carlos J. Pérez del Pulgar and Martin Azkarate</i>	169
FROM TRADITIONAL TO FEW-SHOT LEARNING AND ITS APPLICATION IN NATURAL LANGUAGE GENERATION IN ROBOTICS <i>Javier Sevilla Salcedo, Sara Carrasco Martínez, J. C. Castillo, Á Castro-González, M. Malfaz, F. Alonso-Martín, Miguel A. Salichs</i>	172

CREACIÓN DE MAPAS DE NAVEGACIÓN PARA UN ROBOT TREPADOR DE ESTRUCTURAS RETICULARES

Francisco José Soler

Grupo de Automatización, Robótica y Visión por Computador, Departamento de Ingeniería de Sistemas y
Automática, Universidad Miguel Hernández de Elche
francisco.soler05@goumh.umh.es

Adrián Peidro

Grupo de Automatización, Robótica y Visión por Computador, Departamento de Ingeniería de Sistemas y
Automática, Universidad Miguel Hernández de Elche
apeidro@umh.es

Marc Fabregat

Grupo de Automatización, Robótica y Visión por Computador, Departamento de Ingeniería de Sistemas y
Automática, Universidad Miguel Hernández de Elche
marc.fabregat@goumh.umh.es

Luis Payá

Grupo de Automatización, Robótica y Visión por Computador, Departamento de Ingeniería de Sistemas y
Automática, Universidad Miguel Hernández de Elche
lpaya@umh.es

Óscar Reinoso

Grupo de Automatización, Robótica y Visión por Computador, Departamento de Ingeniería de Sistemas y
Automática, Universidad Miguel Hernández de Elche
o.reinoso@umh.es

Resumen

En este artículo se presenta un estudio sobre la creación de mapas de estructuras reticulares, con el objetivo de permitir a un robot trepador navegar a través de estas últimas. Para la tarea de mapeado se ha estudiado el uso de un sensor LiDAR montado sobre el robot que vaya integrando el mapa a medida que éste se mueve por la estructura. Para obtener datos del sensor (nubes de puntos) y realizar el estudio con facilidad, se ha modelado un entorno de simulación con el robot y una estructura reticular. Por último, se han probado distintos algoritmos para integrar las nubes de puntos obtenidas y crear un mapa de la estructura.

Palabras clave: Estructura reticular; Robot trepador; Simulación; ROS; Gazebo; LiDAR Mapping.

1 INTRODUCCIÓN

Las estructuras reticulares metálicas están muy presentes en la sociedad actual, las podemos encontrar en multitud de construcciones artificiales, como torres de alta tensión, aeropuertos, refinerías, puentes, etc. Como cualquier construcción, dichas

estructuras requieren de un mantenimiento e inspección para prever posibles daños o fallas, como pueden ser tareas de pintado para evitar corrosión, trabajos de soldadura, apriete de tornillos, etc. Tradicionalmente, estas tareas han sido llevadas a cabo por un operario, el cual, ha de recorrer la estructura en busca de dichas fallas y corregirlas. Dado que las estructuras mencionadas suelen tener considerables dimensiones y alcanzar alturas elevadas, supone un gran riesgo para el operario trabajar sobre ellas, quedando expuesto a caídas, traumatismos o electrocuciones pese a las medidas de seguridad pertinentes. Para evitar este último hecho, se propone emplear robots capaces de realizar dichas tareas sin riesgo alguno para los operarios. La aplicación de la robótica en el desempeño de dichas tareas es algo que dese hace más de una década [2] se está estudiando pero con el paso de los años y el avance de la tecnología, permite mejorar muchos de sus aspectos, como el peso, precisión, navegación, etc.

Existen principalmente dos tipos de robots capaces de realizar tareas en estas estructuras [12]: de movimiento continuo o paso a paso. Este artículo se centra en el uso del robot híbrido (paso a paso) desarrollado en [5]. Sobre el robot descrito en el

artículo mencionado, se ha colocado un sensor LiDAR que permite recibir información sobre la estructura y mediante el cual se pretende realizar un mapeado de esta. Para agilizar todo el proceso, minimizar costes y facilitar el testeado de los algoritmos, se optado por trabajar en un entorno de simulación capaz de simular sensores LiDAR y el movimiento del robot a través de la estructura.

Como entorno de simulación, se ha decidido emplear Gazebo Simulator. Se trata de un simulador tridimensional de código abierto, capaz de incorporar múltiples robots de forma simultánea. Es capaz de simular la dinámica de los elementos incorporados, así como incluye variedad de sensores, robots comerciales y la posibilidad de que el usuario desarrolle plugins para incorporar nuevas funciones en el simulador. Además de lo anterior, Gazebo está integrado con ROS (*Robot Operating System*), una de las principales razones por las que se ha elegido este simulador, dado que se pretende que el control del robot trepador se desarrolle bajo esta plataforma y quede todo integrado en ella.

ROS es una plataforma de código abierto, muy popular en el ámbito de la robótica, pues incluye muchas facilidades para desarrollar nuevos proyectos o implementar proyectos ya realizados. Tiene multitud de paquetes desarrollados para diferentes aplicaciones, mapeado, navegación, control de robots, etc. Además, actualmente los fabricantes de sensores y robots suelen proporcionar paquetes para poder emplear sus equipos bajo esta plataforma. Otra de las ventajas que ofrece, es la posibilidad de integrar librerías externas, como PCL (Point Cloud Library), que será la empleada para trabajar con las nubes de puntos obtenidas del LiDAR y hacer la integración de estas.

Bajo la plataforma ROS y el simulador Gazebo, este artículo se centra en crear un modelo digital del robot y de la estructura, para hacer un mapa de dicha estructura.

El artículo se organiza en las siguientes secciones. Una descripción del robot híbrido y sus capacidades se desarrolla en la sección 2. Posteriormente, en la sección 3, se procede a describir la creación del modelo digital del robot y la estructura en el entorno de simulación Gazebo. Casi para acabar, se exponen los algoritmos empleados para la creación de mapas de la estructura reticular y los resultados obtenidos para cada uno de ellos. Por último, se realiza una conclusión del artículo, donde se comentan los aspectos más destacados, resultados obtenidos, así como trabajos futuros.

2 ROBOT HÍBRIDO

Los robots híbridos están formados por varios mecanismos paralelos conectados en serie, lo que les proporciona las ventajas de ambas arquitecturas, es decir, el amplio espacio de trabajo de los robots serie y la elevada capacidad de carga y rigidez de los robots paralelos [4]. El robot híbrido empleado, ha sido desarrollado por el grupo de investigación de Automatización, Robótica y Visión por computador (ARVC) de la Universidad Miguel Hernández de Elche (UMH). Un exhaustivo desarrollo y estudio del robot se puede encontrar en [4] o de forma más breve en [6].

El robot en concreto es bípedo y posee una arquitectura híbrida serie-paralela, ya que cada pata es la combinación serie de dos mecanismos paralelos. La combinación serie de ambas patas se realiza mediante articulaciones rotativas que unen cada una de las patas con una “cadera”.

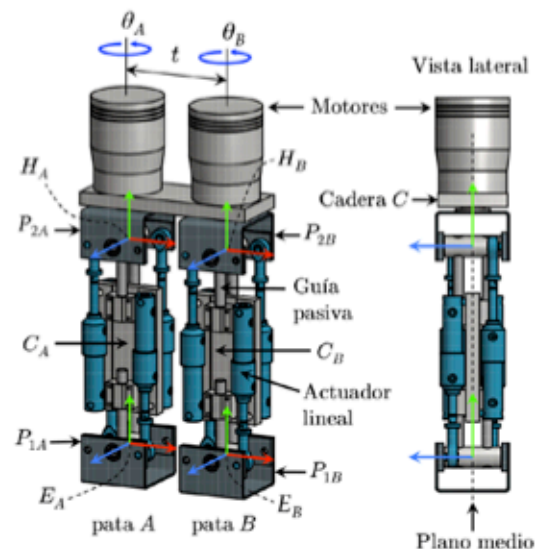


Figura 1: Modelo CAD del robot híbrido

En la Figura 1 se muestra un modelo 3D del robot. Cada pata (A o B) está formada por tres elementos: un núcleo o “core” (C#) y dos plataformas (P1#, P2#). Las plataformas inferiores (P1#) cumplen la función de pie adhiriendo el robot a la estructura metálica mediante un sistema de imanes permanentes conmutables. Durante el movimiento del robot, sólo una de las patas estará fija a la estructura. Las plataformas superiores (P2#), se conectan a la “cadera” mediante articulaciones rotativas. Además, cada plataforma está conectada al “core” a través de dos actuadores prismáticos y un carril deslizante constituyendo un mecanismo paralelo de 2 GDL (Figura 2). Cada pata dispone de dos plataformas, por lo que cada una está compuesta por dos mecanismos paralelos planares de 2 GDL mencionados arriba. De este modo, cada pata tiene un total de 4 GDL. Los actuadores prismáticos se encuentran a ambos lados del “core” de cada pata.

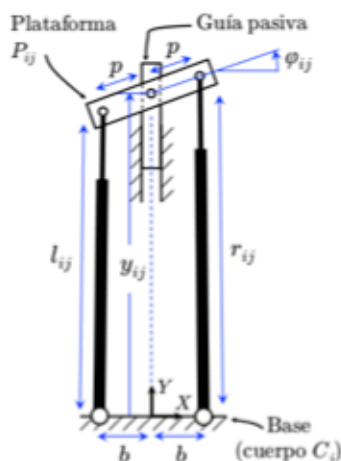


Figura 2: Mecanismo paralelo que forma las patas del robot

En total, el robot tiene 10 GDL entre los pies: las dos rotaciones de la “cadera” y los cuatro actuadores lineales de cada pata. Dado que bastan 6 GDL para posicionar y orientar un pie con respecto al otro, se trata de un robot redundante.

A pesar de que el robot es híbrido, es posible obviar su estructura interna (es decir, sus mecanismos paralelos) y analizarlo como un robot de tipo serie “equivalente” con 8 GDL. El robot “equivalente” serie, está compuesto por dos módulos RPR conectados a un mismo elemento mediante articulaciones de rotación (RPR-R-R-RPR). La notación anterior representa las articulaciones de rotación por una R y las prismáticas por una P.

Puesto que el objetivo de este artículo es la creación de mapas de navegación, se ha empleado el equivalente serie del robot híbrido, para simular el movimiento del robot a través de la estructura lo que facilita considerablemente las tareas de simulación sin afectar demasiado a la generación de datos ni al mapeado.

Como ya se ha mencionado anteriormente el robot queda detallado por completo en [3].

3 SIMULACIÓN DE ENTORNOS EN ROS GAZEBO

Para obtener datos LiDAR de diferentes estructuras y situaciones, se ha optado por realizar una simulación del robot híbrido que permita generar datos y testear los algoritmos de mapeado. En este apartado se reflejan las principales tareas llevadas a cabo para el modelado y la simulación del robot.

Esta tarea se ha llevado a cabo a través del simulador Gazebo, en conjunción con ROS. Gazebo es una

plataforma de simulación 3D que permite al usuario una recreación realista de un entorno donde comprobar, analizar y estudiar el desarrollo de tareas realizadas por sistemas robóticos. Es una herramienta muy útil desde la que un usuario puede crear su propio robot, así como testear su correcto control para una posible implementación futura en la realidad.

Para este artículo se emplea la distribución de ROS- Noetic junto con Gazebo 11.9.0 sobre el sistema operativo Ubuntu 20.04.

3.1 MODELO DEL ROBOT

Para realizar la simulación en Gazebo-ROS, En primer lugar, se crea un paquete de ROS, el cual contendrá todos los elementos necesarios para la simulación.

Una vez dispuesto dicho paquete, es necesario crear un archivo que describa al robot, así como todos sus elementos y articulaciones. Esta descripción se hace mediante el uso del lenguaje XML siguiendo el formato denominado Unified Robot Description Format (URDF). Este formato contiene una serie de especificaciones que permiten al usuario describir los elementos y articulaciones del robot [8]. Cabe destacar que URDF es el formato final del archivo que describe el robot, siendo lo más común redactar la descripción del robot en el formato XACRO [9], cuyo objetivo es simplificar la redacción de archivos URDF permitiendo realizar declaraciones condicionales, parametrizaciones, macros, lo que reduce considerablemente el tamaño del archivo y simplifica su desarrollo. La conversión entre XACRO y URDF es automática, simplemente es necesario ejecutar un comando.

Llegados a este punto, y durante la descripción del robot, es necesario emplear la herramienta de ROS denominada RViz. Esta herramienta es muy útil a la hora de realizar desarrollos en ROS permitiendo ver en una interfaz gráfica nubes de puntos 3D, un modelo 3D del robot, representar un mapa de ocupación, etc. Cabe destacar que RViz es sólo una herramienta de visualización, es decir, no realiza ningún tipo de simulación. En la Figura 3 se muestra el robot descrito en URDF mediante la herramienta RViz.

Para el modelado del robot se han empleado formas geométricas primitivas, cubos o cilindros, de modo que se reduzca la necesidad de los recursos gráficos y de computación, sin modificar las dimensiones reales del robot.

Al robot en cuestión, se le ha añadido un LiDAR en la parte superior, representado por el cilindro rojo que se muestra en la Figura 3.

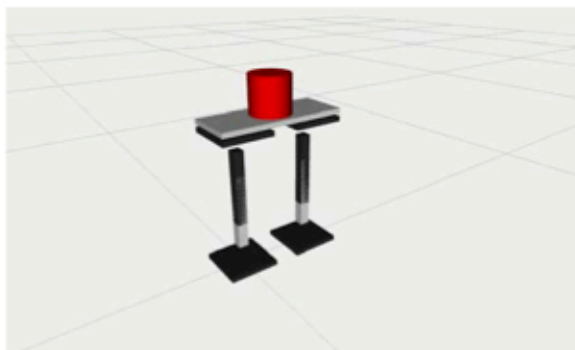


Figura 3: Modelo 3D de la descripción URDF del robot visualizado en RViz

El LiDAR en concreto que se simula, es un Ouster OS1 de 128 canales [3], el cual se ha basado en la descripción del Ouster OS1 de 64 canales dispuesto por Wil Selby en [11]. Junto con el paquete desarrollado por Wil Selby, se proporciona un plugin de Gazebo (porción de código que desarrolla una determinada función) que permite la correcta simulación del sensor.

3.2 MODELO DE LA ESTRUCTURA

El modelo de la estructura reticular se ha desarrollado desde la herramienta que incorpora Gazebo denominada “Model Editor” mediante formas geométricas simples. El modelo de estructura en cuestión se puede observar en la Figura 4.

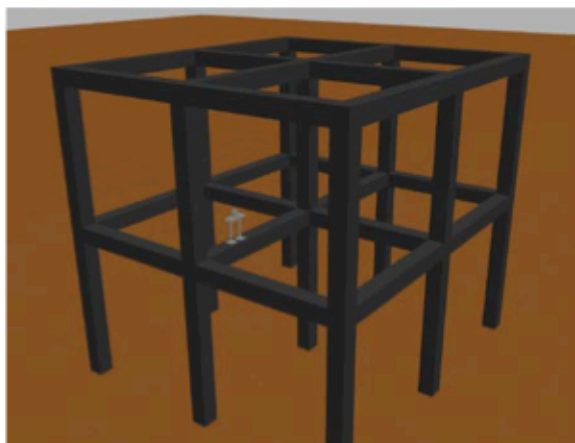


Figura 4: Modelo 3D de la estructura reticular

3.3 PLUGIN DE GAZEBO

Un plugin en Gazebo es un fragmento de código previamente compilado, que tiene comunicación con el entorno de simulación y se ejecuta durante el transcurso de esta. Los plugins de Gazebo se desarrollan a través de su interfaz de programación de aplicaciones (API) en lenguaje C++ [1].

Para este caso en concreto, como lo que se busca es la obtención de datos LiDAR para realizar pruebas con algoritmos de creación de mapas, se ha optado por realizar una simulación en Gazebo con el sistema de simulación de físicas desactivado, puesto que el objetivo es desarrollar una herramienta de creación de mapas y no simular la dinámica del robot. Para realizar el control de las articulaciones del robot con el sistema de físicas desactivado, se ha desarrollado un plugin de Gazebo que permita establecer las posiciones articulares del robot desde tópicos de ROS. El valor de las articulaciones se puede establecer publicando el mensaje directamente en el tópico correspondiente o a través de una interfaz con deslizaderas tal y como se muestra en la Figura 5. Los valores de las articulaciones de rotación (q_1 , q_3 , q_4 , q_5 , q_6 , q_8) se establecen en radianes, mientras que las articulaciones prismáticas se establecen en milímetros (q_2 , q_7).



Figura 5: Interfaz para establecer valores articulares

La funcionalidad del plugin es la siguiente. Cada vez que se publica un nuevo valor en un determinado tópico, referente a la posición de cada una de las articulaciones, se llama a una función encargada de actualizar las posiciones articulares. Previo a llamar a la función para actualizar las posiciones articulares, se comprueba qué pata está fijada, indicado mediante un parámetro, en función del cual se establecerán unos valores articulares u otros.

La simulación del cambio en la fijación de una pata a la otra se realiza empleando dos modelos URDF del robot, para cada uno de los cuales se encuentra una de las patas fijadas. En función de la pata que se desee fijar en cada momento y los valores articulares en el instante del cambio de pata, se carga en Gazebo

un modelo u otro en la configuración y posición correspondientes.

El análisis cinemático y las posibles trayectorias o movimientos del robot se describen en [4] quedando fuera del alcance de este artículo.

4 CREACIÓN DE MAPAS DE ESTRUCTURAS RETICULARES

Para la creación de mapas, se ha desarrollado un paquete en ROS que permite trabajar con nubes de puntos publicadas en el tipo de mensajes de ROS (`sensor_msgs/PointCloud2`) mediante una librería externa. Como librería externa se ha optado por emplear la librería abierta denominada Point Cloud Library (PCL) [7]. Dicha librería está desarrollada en C++ y contiene gran cantidad de algoritmos, funciones y clases que facilitan trabajar con nubes de puntos. Dicha librería está dividida en una serie de módulos que permiten desde la visualización, hasta el filtrado, segmentación y el registro entre nubes de puntos.

El paquete desarrollado, incorpora una serie de nodos, que permiten almacenar lecturas de nubes de puntos en formato “.pcd”, pasar de formato “.pcd” a “.ply”, visualizar nubes de puntos en tiempo real a través del visualizador de PCL, aplicar filtros, extracción de planos y algoritmos para realizar registros entre dos nubes de puntos.

Como norma general a las nubes de puntos recibidas se les hace un filtrado por distancia en alguno de los ejes, si no en todos ellos, y además se les hace un filtrado de voxel para reducir el número de puntos de la nube. Tras esto se han realizado distintas pruebas con diferentes algoritmos para tratar de casar las nubes de puntos y posteriormente integrarlas en un mapa.

Para la creación del mapa se toma como sistema de referencia fijo el correspondiente a la captura de la primera nube de puntos. A partir de ese punto, se van registrando nubes de puntos consecutivas, lo que proporciona una matriz de transformación homogénea (MTH) que permite pasar de un sistema de referencia posterior al sistema previo. Acumulando las MTH que pasan de un sistema al sistema previo, se puede obtener cualquier nube de puntos respecto del sistema de referencia inicial. Para el testeo de los algoritmos empleados, se ha realizado la captura de 74 nubes de puntos en la que el robot ha seguido una trayectoria para cambiar de una viga a otra. En la Figura 6 se muestra la trayectoria aproximada seguida por el robot.

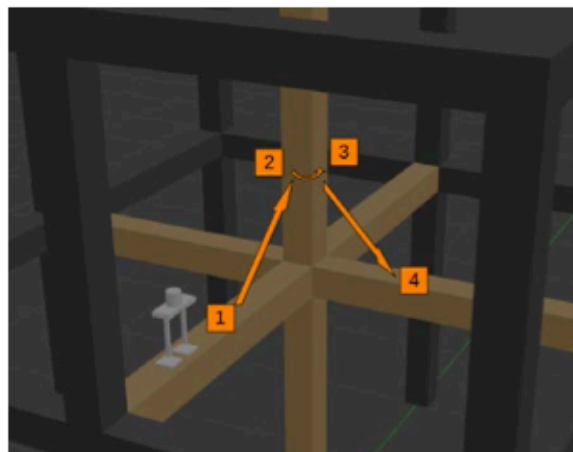


Figura 6: Trayectoria realizada para el testeo de los algoritmos y reconstrucción del mapa

A las nubes de puntos capturadas se les aplica un filtrado de voxel de tamaño (0.1, 0.1, 0.1) metros en cada uno de sus ejes previamente a ser introducidas en los algoritmos de creación de mapa. En las siguientes subsecciones se presentan los algoritmos empleados para la creación del mapa, así como sus ventajas, la calidad del mapa obtenido y el tiempo de cómputo necesario.

4.1 ITERATIVE CLOSEST POINT (ICP)

El primero de los algoritmos testado ha sido el conocido algoritmo “Iterative Closest Point” más comúnmente conocido como ICP y emplea un cálculo de distancia punto-a-punto. Este algoritmo trata de encontrar una MTH tal que la distancia entre los puntos de ambas nubes sea mínima basándose en la descomposición de valores singulares (SVD). La clase de la librería PCL que implementa este algoritmo se denomina `pcl::IterativeClosestPoint` y requiere de dos entradas, la nube que se desea transformar y la nube sobre la que se desea transformar la primera.

El mapa generado mediante este algoritmo se muestra en la Figura 7, donde se puede observar un mapa que se aproxima a la estructura real, pero que dista mucho de ser una reconstrucción fidedigna de la estructura. En la Tabla 1 se pueden observar los resultados obtenidos.

4.2 ICP WITH NORMALS

En segundo lugar, se ha empleado una variante del algoritmo ICP, en el que se tiene en cuenta la normal de cada punto de la nube. Este algoritmo busca una transformación basada en distancias de punto-a-plano, está implementado en la clase `pcl::IterativeClosestPointWithNormals` y de igual modo que el anterior, requiere de dos entradas, pero

en este caso, cada punto de las nubes de entrada ha de tener una normal asociada que se ha de calcular previamente. Para el cálculo de las normales se puede emplear la clase `pcl::NormalEstimation` que permite obtener la normal de cada punto dado un radio o un número de vecinos sobre los que calcular la normal a cada punto.

El mapa obtenido se observa en la Figura 8. Este mapa presenta mejoras respecto al anterior pero aún se puede mejorar.

4.3 GENERALIZED ICP

Por último, se ha empleado el algoritmo denominado Generalized ICP desarrollado en [10]. Tal y como se comenta en [10], este algoritmo es una variante del ICP la cual se puede considerar que emplea distancia plano-a-plano. Dado que nuestro entorno está compuesto básicamente por planos, esta premisa parece muy prometedora y como veremos más adelante, efectivamente lo es.

En la Figura 9 se puede observar como el mapa que se ha generado en este caso representa más fielmente la estructura, con unos bordes y planos más definidos.

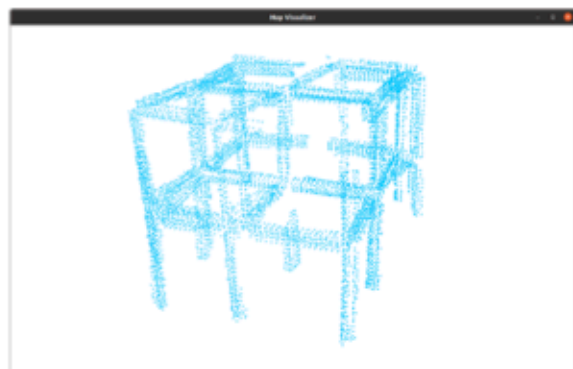


Figura 7: Mapa obtenido con ICP

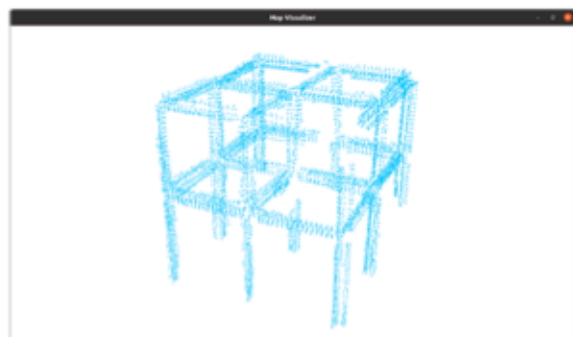


Figura 8: Mapa obtenido con ICP with Normals

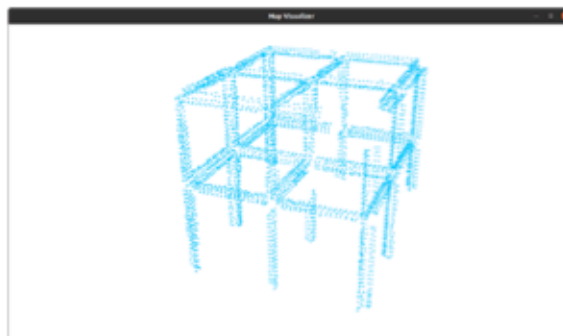


Figura 9: Mapa obtenido con Generalized ICP

Para comparar el resultado de los 3 algoritmos de forma cuantitativa y puesto que en el mapa que se ha creado se deberían detectar 11 planos, se propone usar como medida del error en la Tabla 1, el número de puntos que no pertenecen a ninguno de los 11 planos correspondientes al mapa.

Tabla 1: Tabla de resultados

Algoritmo empleado	Tiempo requerido (s)	Error
ICP	9	2609
ICP with Normals	8	940
Generalized ICP	50	37

5 CONCLUSIONES

Este artículo muestra la capacidad de ROS, bajo el cual se pueden implementar multitud de herramientas, desde simulaciones, trabajar con nubes de puntos y creación de mapas.

Por otro lado, se muestran algunos de los principales algoritmos para el registro de nubes de puntos, así como el Generalized ICP como algoritmo más preciso en el caso de disponer de muchos planos.

Como trabajos futuros se pretende implementar funciones para extracción de características que permitan realizar una primera estimación de la transformación entre nubes de puntos, además de emplear una cámara RGB-D para tratar de implementar esta misma creación de mapas.

Agradecimientos

Este trabajo forma parte del proyecto PID202-116418RB-I00 financiado por el Ministerio de Ciencia e Innovación MCIN/AEI/10.13039/501100011033.

English summary

CREATION OF NAVIGATION MAPS FOR A RETICULAR STRUCTURE CLIMBING ROBOT

Abstract

This paper presents a study on the creation of maps of reticular structures, with the objective of allowing a climbing robot to navigate through them. For the mapping task, it has been studied the use of a LiDAR sensor mounted on the robot, for this to integrate the map as it moves through the structure. To obtain data from the sensor (point clouds) and for the study to be performed easily, it has been modelled a simulation environment with the robot and the reticular structure. Lastly, different algorithms have been tested to integrate the point clouds obtained and create a map of the structure.

Keywords: Reticular Structure; Climbing Robot; Simulation; ROS; Gazebo; LiDAR Mapping.

Referencias

- [1] amazonaws.com. *Gazebo: Main Page - Amazon AWS*. [online] Disponible en: <<http://osrf-distributions.s3.amazonaws.com/gazebo/api/11.0.0/index.html>> [Accedido el 26 abril 2022].
- [2] Aracil, R., Saltarén, R., Reinoso, O., (2003) *Robotics and Autonomous Systems - Volumen 42(2)*, pp 125-134. ISSN:0921-8890.
- [3] Ouster.com. *High-resolution OS1 lidar sensor: robotics, trucking, mapping*. [online] Disponible en: <<https://ouster.com/products/scanning-lidar/os1-sensor/>> [Accedido el 26 abril 2022].
- [4] Peidró, A., (2018) *Kinematic Analysis and Design of the HyReCRo Robot: a Serial-Parallel and Redundant Structure-Climbing Robot*. Doctorado. Universidad Miguel Hernández, Departamento de Ingeniería de Sistemas y Automática.
- [5] Peidró, A., Gil, A., Marín, J., Berenguer, Y. and Reinoso, Ó., (2015). Kinematic analysis and simulation of a hybrid biped climbing robot. *2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, pp.24-34.
- [6] Peidró, A., Marín, J., Gil, A., Berenguer, Y., & Reinoso, Ó. (2018). Analysis and Design of a Serial-Parallel Redundant Biped Climbing

Robot. *Jornadas Nacionales De Robótica 2018, (Valladolid, 14-15 de junio de 2018)*.

- [7] Point Cloud Library. Acceso el 26 de abril de 2022, desde <https://pointclouds.org/>
- [8] ROS Wiki. 2013. *urdf/XML - ROS Wiki*. [online] Disponible en: <<http://wiki.ros.org/urdf/XML>> [Accedido el 26 abril 2022].
- [9] ROS Wiki. 2022. *xacro - ROS Wiki*. [online] Disponible en: <<http://wiki.ros.org/xacro>> [Accedido el 26 abril 2022].
- [10] Segal, A., Hähnel, D., Thrun, S. (2009). Generalized-ICP. *Proc. of Robotics: Science and Systems*. 10.15607/RSS.2009.V.021.
- [11] Selby, W., (2020). *wilselby - ouster_example*. [online] Github.com. Disponible en: <https://github.com/wilselby/ouster_example> [Accedido el 26 abril 2022].
- [12] Tavakoli, M., Marques, L., and De Almeida, A. T. (2011). 3DCLIMBER: Climbing and manipulation over 3D structures. *Mechatronics*, 21(1):48-62.



© 2022 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution CC BY-NC-SA 4.0 license (<https://creativecommons.org/licenses/by-ncsa/4.0/deed.es>).