

# Análisis de algoritmos de navegación para una flota de robots móviles mediante gemelos digitales

Alberto Martínez-Gutiérrez<sup>1</sup>, Javier Díez-González<sup>1</sup>, Paula Verde<sup>1</sup>, Rubén Ferrero-Guillén<sup>1</sup> and Hilde Perez<sup>1</sup>

<sup>1</sup>Sistemas inteligentes para la fabricación, Universidad de León, España.

Email: [amartg, jdieg, pverg, rferrg, hperg]@unileon.es

## Resumen

La manutención industrial se encuentra en un proceso de digitalización y automatización mediante el uso de tecnologías de la industria 4.0. Un ejemplo son los gemelos digitales los cuales permiten virtualizar activos como la flota de Robots Móviles Autónomos (RMA). En este entorno digitalizado son analizados dos algoritmos de navegación local; Aproximación de Ventana Dinámica (DWA) y el Tiempo de Banda Elástico (TEB). Para ello, se implementa un gemelo digital en el ecosistema ROS donde se virtualiza un entorno industrial. En este entorno es simulada una misión de una flota de tres RMA donde navegan conjuntamente con el fin de comparar el comportamiento de los algoritmos en entornos industriales dinámicos. Como resultado, se analizan las trayectorias y los tiempos de cada RMA que son clave para la planificación de los costes de producción. Además, en este trabajo se desarrolla un servidor web para interactuar con el gemelo digital facilitando su uso.

**Palabras clave:** Industria 4.0; Robots autónomos; Gemelos Digitales; simulación.

## Abstract

Industrial handling is undergoing a process of digitalization and automation through the use of Industry 4.0 technologies. For instance, digital twins allow the virtualization of assets such as the fleet of Autonomous Mobile Robots (AMR). In this digitized environment, two local navigation algorithms are analyzed; Dynamic Window Approximation (DWA) and Time Elastic Band (TEB). For this purpose, a Digital Twin is implemented in the ROS ecosystem where an industrial environment is virtualized. On this environment, a mission of a fleet of three RMAs is simulated where they will navigate simultaneously in order to compare the behavior of the algorithms in dynamic industrial environments. As results, the trajectories and times of each RMA are analyzed, which are key for the planning of production costs. Furthermore, in this work a web server is developed to interact with the digital twin, facilitating its use.

**Keywords:** Industry 4.0; Autonomous Mobile Robots; Digital Twin; Simulation.

## 1. Introducción

La industria manufacturera se encuentra en un proceso de transformación digital donde las diferentes tecnologías convergen hacia una producción flexible, individualizada y eficiente [1]. Para ello, la digitalización de los activos en los procesos industriales es esencial en la creación de un entorno colaborativo con el fin de generar mayor valor añadido. Bajo este paradigma nace el concepto de la industria 4.0 [2] donde la integración de tecnologías heterogéneas es un requisito para alcanzar una

producción más eficiente mediante la eliminación de las islas de automatización. En este contexto, los Sistemas Ciber-Físicos (SCF) son clave para la fusión del nivel físico con el digital habilitando la interconexión de toda la cadena de valor [3]. La comunicación de los activos industriales presenta requisitos específicos como, tiempos de respuesta reducidos, seguridad e integridad respecto a las redes convencionales. Por este motivo, se desarrolla el concepto Internet Industrial de las Cosas (IIIC) donde se emplean políticas y métodos para la transmisión de

datos e información que permitan la fabricación integrada y colaborativa [4].

Dentro de este contexto de digitalización y conectividad, la manutención industrial es un activo donde se necesita la coordinación de otros procesos de fabricación con el fin de ser eficiente. Para alcanzar un transporte eficiente es necesario la generación de datos de los diferentes procesos mediante SCF con el fin de obtener información para la toma de decisiones [5]. En este sentido, la toma de decisiones puede ser ejecutada por un sistema experto debido a que la información necesaria está digitalizada. Sin embargo, para la automatización completa del transporte interno se requiere de vehículos autónomos capaces de navegar de forma segura por la planta de fabricación. Estos vehículos se denominan en la literatura científica como Robots Móviles Autónomos (RMA) los cuales equipan diferentes sensores que les permiten reconocer el entorno, así como cierta inteligencia embarcada para navegar con seguridad [6]. Además, los RMA disponen de comunicaciones inalámbricas para recibir las diferentes misiones y comunicar su estado en tiempo real.

Sin embargo, la adaptación de los RMA a los entornos industriales es desconocida debido a la heterogeneidad de las distribuciones de las máquinas, así como la posición dinámica del personal. No obstante, la evaluación del comportamiento de los RMA es fundamental para la previsión de los costes y de la planificación de la producción. En este contexto, los Gemelos Digitales (GD) permiten la virtualización de un entorno, así como la modelización de los RMA. Mediante el empleo de GD se es capaz de analizar las trayectorias, tiempos e incluso consumos de batería, claves para la planificación y optimización de la actividad productiva. Además, el empleo de GD permite la simulación en diferentes hipótesis sin necesidad de implementación en la realidad con el ahorro de coste que supone para la industria manufacturera.

En este contexto, existe un estudio [7] el cual analiza el comportamiento de un único RMA en un GD comparándolo con robot real. En dicho estudio se recrean dos escenarios con obstáculos estáticos, alcanzando un alto nivel de similitud; sin embargo, este trabajo no analiza el comportamiento con varios RMA de forma simultánea. Además, la navegación de una flota de RMA presenta diversas peculiaridades en el ámbito de la navegación, planificación o simulación. Por este motivo, en este trabajo se estudiará el comportamiento de la navegación de una flota de RMA haciendo uso de un GD. Además, en este estudio se implementará una interfaz desde la que facilitar el uso por parte de personal no cualificado con el fin de extender la aplicación a usuarios no expertos.

En este trabajo además de la creación del GD con varios RMA, se pretende evaluar el comportamiento de dos algoritmos diferentes de navegación local. El

objetivo de este estudio es establecer cuál ofrece menores tiempos de transporte una vez configurado y adaptados sus parámetros para evitar colisiones entre ellos. Además, para que la evaluación sea comparable, se realizará en un mismo entorno industrial de simulación, así como casos de estudio idénticos.

El trabajo está organizado de la siguiente manera: en el capítulo 2 se analizará el problema, así como el caso de estudio. En el capítulo 3 se plantea el marco teórico y en el capítulo 4 la metodología propuesta. Los resultados serán mostrados en el capítulo 5. Posteriormente, se discutirán los algoritmos de navegación en el capítulo 6, finalizando con las conclusiones en el capítulo 7.

## 2. Caso de estudio

Los RMA son un activo fundamental para la ejecución de las decisiones para el transporte interno en la planta industrial [8]. Para realizar este demostrador y estudio con una flota de RMA, se recreará de forma virtual una planta industrial para la fabricación de piezas. Para ello, las piezas necesitan su traslado de unas máquinas a otras para su terminación en un almacén.

Con este caso, se tiene como objetivo analizar las trayectorias y los tiempos asociados al transporte autónomo de estos vehículos. Sin embargo, para analizar el comportamiento de los algoritmos de navegación con obstáculos móviles, ha sido recreado un escenario de pequeñas dimensiones con el fin de maximizar los encuentros con los RMA.

### 2.1. Descripción del RMA

El vehículo autónomo industrial empleado es un robot de movilidad no holónica [9]. En este caso, el robot dispone de dos ruedas motrices independientes aportando dos Grados De Libertad (GDL) frente a los 3 GDL de los robots holónicos (i.e., dos para posición y uno para orientación).

En cuanto a su cinemática, se basa en un movimiento de tracción diferencial mediante dos ruedas convencionales fijas. No obstante, en sus extremos dispone de ruedas descentradas orientables para distribución de carga y estabilidad las cuales no son motrices, como se puede observar en la Figura 1.



Figura 1. Robot autónomo industrial objeto del estudio.

De acuerdo con la Figura 1, las ruedas motrices se encuentran en los laterales del centro geométrico del robot permitiendo rotar sobre sí mismo sin necesidad de desplazarse. De esta forma se logra una alta movilidad en espacios reducidos, la cual es considerada en los algoritmos de navegación.

Además, este vehículo autónomo objeto de estudio es comercializado por la empresa *Mobile Industrial Robots* (MIR) y es ampliamente usado en la industria. Concretamente, el modelo es el MIR100, el cual es capaz de llevar 100 kg de carga útil con una velocidad máxima de 1.5 m/s.

## 2.2. Entorno de simulación

En este trabajo es necesaria la definición precisa de un entorno de navegación para establecer el comportamiento de las trayectorias y tiempos de los algoritmos. Este hecho, diferencia los GD con otros programas de simulación basados en tiempos como *FlexSim*, los cuales no tienen en cuenta la navegación de forma activa como sucede en los GD.

Para analizar los algoritmos de navegación, los autores plantean una planta de fabricación equipada con máquinas de mecanizado, simulando un entorno industrial como se puede ver en la Figura 2. Este tipo de producción requiere de un transporte con el fin de proveer de tochos de material a las máquinas las cuales devuelven piezas intermedias o finales.

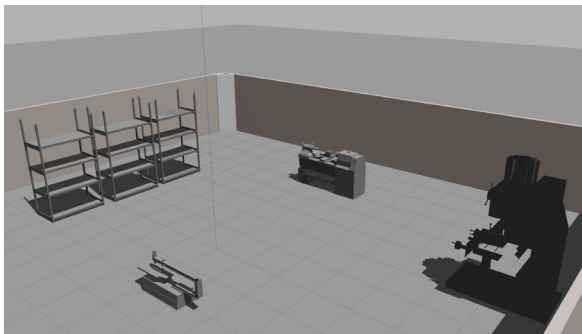


Figura 2: Virtualización del entorno de fabricación del gemelo digital.

En este contexto, se ha simulado una planta de fabricación rectangular. La planta virtual cuenta con 3 tipos de máquinas (Sierra, torno y taladradora) y un almacén, es decir, cuatro objetivos de navegación. Dichos objetivos tienen dos coordenadas cartesianas en el plano del suelo, así como la orientación en la que se debe estacionar el vehículo. En este caso se implementan tres RMA para satisfacer la mantención del proceso de fabricación.

La distribución de las máquinas se realiza por lotes estando espacialmente distribuidas para que los RMA puedan maniobrar. Sin embargo, en este trabajo se busca la intersección de las trayectorias para evaluar el comportamiento de los algoritmos de navegación ante

obstáculos móviles. Por lo tanto, las máquinas se distribuirán más cerca que en la realidad resultando un caso más complejo para los algoritmos.

## 2.3. Definición de misión

Para evaluar ambos algoritmos, el entorno y los objetivos han de ser los mismos con el fin de ser comparables. Para ello se ha establecido una misión para cada uno de los RMA. De esta forma el entorno y las circunstancias son idénticas, pudiendo ser comparables los resultados de los algoritmos.

Para este caso de estudio, a todos los RMA se les envían las misiones al mismo tiempo; desplazándose hasta otros dos objetivos diferentes. Las restricciones para la misión estipula que dos vehículos no pueden estar estacionados en una misma máquina. Las misiones de cada RMA son mostradas en la Tabla 1.

Tabla 1. Definición de misiones para la evaluación de los algoritmos de navegación.

Misiones de RMA			
Posición	RMA-1	RMA-2	RMA-3
Inicial	Sierra	Fresado	Torno
Intermedia	Torno	Almacén	Sierra
Final	Fresado	Sierra	Almacén

Una vez establecidas las misiones y el caso de estudio se procede a la resolución del problema.

## 3. Marco teórico

El entorno de desarrollo para la evaluación de los algoritmos y simulaciones dinámicas se realiza con el Sistema Operativo de Robots o ROS por sus siglas en inglés. Este entorno dispone de librerías de software y herramientas para el desarrollo de aplicaciones y algoritmos en código abierto.

Para simular el comportamiento de un robot se requiere de un entorno simulado donde se incorpora un modelo virtual del vehículo con sus restricciones cinemáticas, sensores y actuadores. El modelo debe de analizar los datos de los sensores en tiempo real para determinar cuáles son las acciones necesarias para el cumplimiento de las misiones u objetivos. Para ello, se requiere de nodos los cuales analizan los datos para obtener información necesaria para la navegación como la ubicación, la velocidad actual, la huella del robot, o los obstáculos entre otros.

Con el fin de alcanzar una interoperabilidad entre los diferentes algoritmos y nodos, se ha establecido una serie de mensajes estructurados y preestablecidos. Estos mensajes son transmitidos mediante la filosofía publicador-suscriptor mediante la cual un mensaje puede ser retransmitido a varios nodos de interés. Del mismo modo funciona la navegación autónoma para el envío de acciones.

### 3.1. Pila de navegación

La pila de navegación consta de diversos programas y algoritmos interconectados entre sí. En robótica móvil, la navegación se estructura de acuerdo con el esquema la Figura 3.

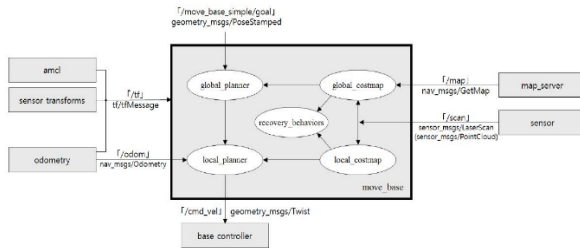


Figura 3. Esquema de navegación en ROS donde se muestran los nodos y sus relaciones.

En el esquema, los nodos vienen representados por elipses mientras que las librerías, sensores y actuadores son rectángulos. En el interior del recuadro negro se encuentran los nodos encargados de establecer las trayectorias globales y locales, así como sus mapas de costos.

No obstante, para que la librería de navegación funcione, se requiere de la conexión con otras librerías como las de posicionamiento, servidor de mapas, o el controlador de movimiento donde se realiza la función cinemática inversa. Además, este proceso de navegación debe ser iterativo y en tiempo real para adaptarse a los entornos dinámicos que se encuentra el RMA.

Para navegar desde un punto A hacia un punto B se establece una trayectoria inicial conocida la posición inicial, el objetivo y el mapa del entorno. Para ello, se emplea el algoritmo de búsqueda de grafos A\* donde se alcanza la trayectoria global con el menor costo [10].

#### 3.1.1. Mapa de costos

El algoritmo A\* requiere definir un mapa de costos para encontrar el camino con el mínimo coste. Para ello, se emplea un mapa discretizado a partir de una matriz de pesos, los cuales se determinan en función de los obstáculos existentes. De acuerdo con el esquema, existen dos mapas de costos: el global y el local.

El mapa de costo global requiere la creación de un mapa de obstáculos estáticos. Este mapa bidimensional se puede generar de forma manual mediante una imagen bicolor. No obstante, el mapa también se puede generar con un escaneo del entorno de forma virtual mediante mapeado con un único robot o con varios robots [11].

Sin embargo, el algoritmo A\* se basa en la trayectoria de un solo punto y no tienen en cuenta las dimensiones del robot, lo que puede desencadenar colisiones. Esto es debido a que la resolución de la matriz de mapa de costo es superior a la dimensión del robot móvil. Por lo

tanto, se debe generar una huella del tamaño del RMA con el fin de que el algoritmo evalúe el mínimo coste considerando todo el contorno de la huella. La huella es definida por las coordenadas de un polígono siendo el centro geométrico del robot el origen de coordenadas.

En este sentido, también se establece una distancia de seguridad con los obstáculos fijos mediante la penalización de sus vecindades en función de la cercanía al obstáculo, denominada capa de inflación. La capa de inflación viene determinada por una relación exponencial.

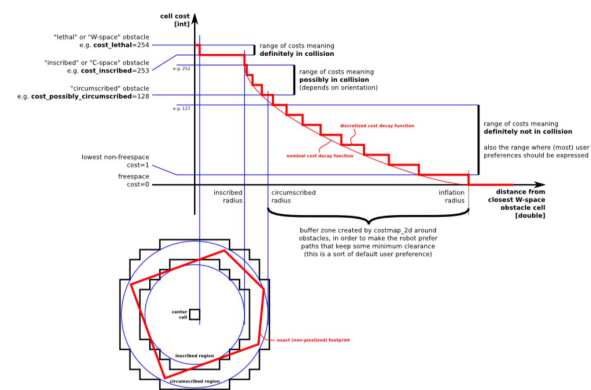


Figura 4. Esquema de asignación pesos en función de la huella y la capa de inflación.

Como se puede observar en la Figura 4, el contorno rojo inferior define la huella de un robot el cual es un obstáculo para otros robots. En cambio, la gráfica superior muestra la asignación del coste en función de la distancia al obstáculo.

Por otra parte, el mapa de costos local se basa solamente en la información recabada por los diferentes sensores en tiempo real. De esta manera, ante un obstáculo no registrado se es capaz de establecer una nueva trayectoria alternativa a la global para alcanzar el objetivo.

#### 3.1.2. Localización y pose

Ubicar en tiempo real el robot en el espacio es una de las premisas necesarias para la navegación. En este sentido, el robot necesita su posición en el entorno, descrita mediante sus coordenadas y orientación (i.e., pose). Para ello, se emplean la convergencia de varias técnicas como la odometría y nube de partículas.

La odometría analiza el movimiento de las ruedas para estimar su posición. Sin embargo, esta técnica es imprecisa debido a la existencia de deslizamiento entre las ruedas y el suelo.

En cambio, el escáner LIDAR genera una nube de puntos la cual es comparada con el mapa precargado para generar una pose estimada. Además, esta información requiere el paso por una función

denominada transformada puesto que el sensor no se ubica en el centro geométrico del robot.

Como consecuencia del ruido existente en estos sensores, se calcula la función de densidad para todas las posibles poses partiendo de un estado inicial donde convergen hacia una zona de interés. Este método se conoce como aproximación de localización de Monte Carlo, AMCL por sus siglas en inglés [12].

### 3.2. Algoritmos de navegación local

El algoritmo de navegación local permite la adaptación de las trayectorias ante objetos que no estaban en la planificación global u otros objetos móviles como otros robots. En la literatura científica existen numerosos algoritmos de navegación local, sin embargo, en este trabajo son objeto de estudio los algoritmos DWA y TEB.

#### 3.2.1. Algoritmo DWA

El algoritmo de aproximación de ventana dinámica DWA [13] por sus siglas en inglés tienen como objetivo evitar colisiones dinámicas mediante la maximización de una función objetivo. El algoritmo evalúa múltiples posibles soluciones y escoge la que maximice la función de la ecuación 1.

$$G(v, w) = \alpha \cdot \text{heading}(v, w) + \beta \cdot \text{dist\_obs}(v, w) + \gamma \cdot \text{vel}(v_t, w_t) \quad (1)$$

Dicha función evalúa la viabilidad en función de la velocidad actual del robot  $\text{vel}(v_t, w_t)$ , de la trayectoria objetivo  $\text{heading}(v, w)$  y las distancias con los obstáculos  $\text{dist\_obs}(v, w)$ . Estas variables son ajustadas mediante los pesos  $\alpha, \beta, \gamma$  en función del entorno y las políticas de navegación.

Sin embargo, no todas las velocidades lineales y angulares son válidas, ya que existen diversas restricciones físicas que deben ser incluidas en el algoritmo:

- Restricción de velocidades lineal y angular máximas y mínimas debido a la limitación de los controladores del motor del RMA. Esta limitación se representa en la ecuación 2.

$$V_m = \{(v, w) \mid v \in [v_{min}, v_{max}], w \in [w_{min}, w_{max}]\} \quad (2)$$

Siendo  $v$  la velocidad lineal y  $w$  la velocidad angular.

- Restricciones debidas a la inercia del vehículo, existiendo unos límites de aceleración y deceleración. Esta restricción queda definida por la ecuación 3.

$$V_d = \{(v, \omega) \mid v \in [v_t - \dot{v}_b dt, v_t + \dot{v}_a dt], \omega \in [\omega_t - \dot{\omega}_b dt, \omega_t + \dot{\omega}_a dt]\} \quad (3)$$

Donde  $v_t$  y  $w_t$  representan las velocidades lineales y angulares en la operación actual del robot. En cambio  $\dot{v}_a$  y  $\dot{v}_b$  son las aceleraciones lineales y angulares máximas y mínimas respectivamente mientras  $\dot{\omega}_a$  y  $\dot{\omega}_b$  representan las aceleraciones angulares.

- Restricción de acuerdo con la distancia de frenado del robot móvil. Su función es garantizar que el robot no colisione con ningún obstáculo. Por lo tanto, el conjunto de velocidades restringidas en función de la deceleración máxima se muestra en la ecuación 4.

$$V_a = \left\{ \begin{array}{l} (v, \omega) \mid v \leq [2 \text{dist}(v, \omega) \cdot \dot{v}_b]^{1/2} \\ \omega \leq [2 \text{dist}(v, \omega) \cdot \dot{\omega}_b]^{1/2} \end{array} \right\} \quad (4)$$

Donde  $\text{dist}(v, \omega)$  refleja la distancia más corta entre el vehículo y el objeto para una cierta velocidad.

Por lo tanto, las velocidades evaluables resultarán del conjunto de las intersecciones sobre las restricciones quedando matemáticamente definido por la ecuación 5.

$$V_e = V_a \cap V_d \cap V_m \quad (5)$$

#### 3.2.2. Algoritmo TEB

El algoritmo de banda elástica temporizada [14], TEB por sus siglas en inglés es una evolución del algoritmo de bandas elásticas cuyo objetivo es ofrecer una trayectoria a nivel local sin obstáculos. Para ello se emplea el concepto de fuerza de repulsión y concentración donde la concentración elimina las holguras de las trayectorias globales y la repulsión le aleja de obstáculos dinámicos. De esta manera el algoritmo es capaz de cambiar las velocidades lineal y angular del RMA para evitar obstáculos móviles.

A diferencia del algoritmo de banda elásticas, TEB contiene una secuencia de poses intermedias que modifican el plan global, resultando más eficiente computacionalmente.

Matemáticamente, el algoritmo es expresado en las ecuaciones 6 y 7 como una secuencia de intervalos de eventos.

$$\begin{cases} Q = \{S_i\}, i = 1, 2, \dots, n \in \mathbb{N} \\ \tau = \{\Delta T_i\}, i = 1, 2, \dots, n - 1 \end{cases} \quad (6)$$

$$B := (Q, \tau) = [S_1, \Delta T_1, S_2, \Delta T_2, \dots, \Delta T_{n-1}, S_n] \quad (7)$$

Donde  $S_i$  es la pose para cada instante de tiempo  $i$  y  $Q$  es la secuencia de la pose. En cambio,  $\Delta T_i$  es el intervalo de tiempo entre poses contiguas.  $\tau$  es la secuencia de intervalos de tiempo.  $B$  es la trayectoria resultada de la combinación entre la secuencia de pose y la de intervalos de tiempo.

El algoritmo resuelve las poses en los diferentes intervalos de tiempo mediante una optimización multiobjetivo ponderada. Para alcanzar la ruta optima el TEB ajusta la pose y los intervalos de tiempo utilizando el modelo de sumas ponderadas.

Definidos los conjuntos de estados  $Q, \tau, B$ :

$$f(B) = \sum_k \gamma_k f_k(B) \quad (8)$$

$$B^* = \underset{B}{\operatorname{argmin}} f(B) \quad (9)$$

Siendo  $B^*$  el resultado optimo que minimiza la función objetivo de la restricción correspondiente mientras  $\gamma$

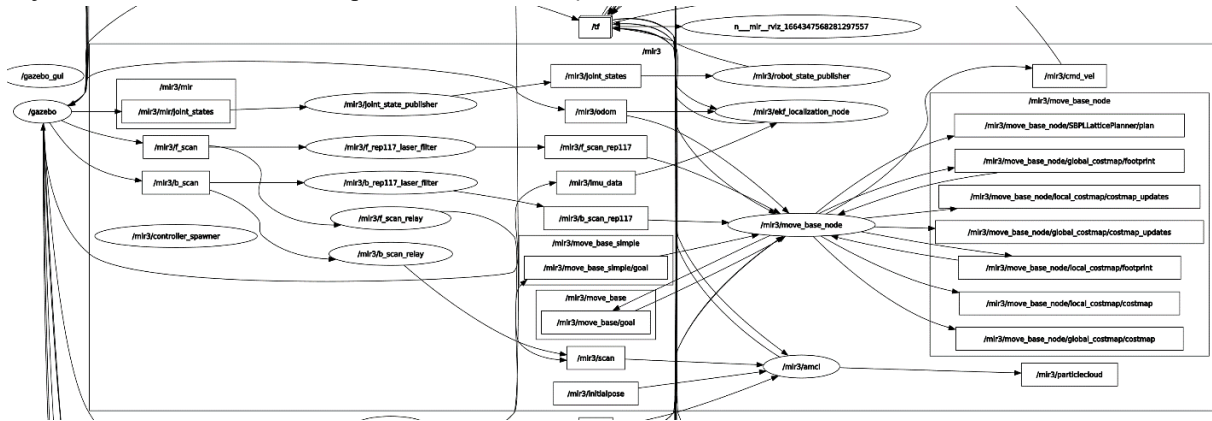


Figura 5: Flujograma de las librerías y herramientas empleadas para la navegación de un único RMA.

De acuerdo con el esquema de la Figura 5, los nodos se representan mediante elipses mientras que los temas o canales se representan con rectángulos. La pila de navegación viene resumida en el nodo *move\_base\_node* donde recibe la información del posicionamiento de acuerdo con el marco teórico.

Para integrar múltiples robots, los autores han decidido crear prefijos con el fin de cargar las mismas características y parámetros sin necesidad de clonar el código. De esta forma con un único archivo XML somos capaces de reproducir los RMA en un mismo entorno de simulación.

#### 4.1.1. Parámetros

Los algoritmos se han ajustado específicamente al entorno para que no se registren colisiones [15]. En este sentido se han ajustado tanto los parámetros de trayectorias globales como mapa de costos. Además, se adaptado los pesos y otros parámetros de los algoritmos de los planificadores locales los cuales son objetivo de

es su peso. En este sentido las restricciones hacen referencia a las velocidades y aceleraciones máxima como se mencionó en el algoritmo DWA.

## 4. Metodología

Una vez definido el marco teórico, se procede al desarrollo del entorno de trabajo para alcanzar los objetivos.

### 4.1. Comparativa de algoritmos

Para el análisis comparativo de los algoritmos de navegación, se ha empleado el ecosistema ROS donde se ha establecido la interconexión entre un conjunto de librerías y herramientas las cuales se pueden observar en la Figura 5.

comparación. Para el algoritmo DWA se ha empleado los parámetros que se observan en la Tabla 2.

Tabla 2: Parámetros para el algoritmo de navegación local DWA.

Parámetros DWA	
Parámetro	Valor
Velocidad máxima	0.5 m/s
Velocidad mínima	0.2 m/s
Aceleración máxima	1.5 m/s <sup>2</sup>
Deceleración máxima	1.5 m/s <sup>2</sup>
Ponderación para seguimiento del plan global	32
Ponderación para alcanzar el objetivo.	48.0
Ponderación para evitar obstáculos	0.1

En cambio, para el algoritmo TEB se han empleado los parámetros que se observan en la Tabla 3.

Tabla 3: Parámetros para el algoritmo de navegación local TEB.

Parámetros TEB	
Parámetro	Valor
Velocidad máxima	0.5 m/s
Velocidad mínima	0.2 m/s
Aceleración máxima	1.5 m/s <sup>2</sup>
Deceleración máxima	1.5 m/s <sup>2</sup>
Distancia de inflación	3.5 m
Ponderación de obstáculos	15.0
Peso de obstáculos dinámicos	50.0

Los parámetros del planificador global, así como del mapa de costos no se han variado con el fin de poder comparar los resultados.

#### 4.1.2. Gazebo

El entorno virtualizado de simulación dinámico se realiza con gazebo donde los RMA son engendrados para la navegación. El entorno ha sido diseñado y exportado en archivos de datos espaciales (SDF) para su interpretación en el ecosistema ROS. La planta de fabricación es de forma cuadrangular con un tamaño de 16 metros de lado.

Tras la creación del entorno se han definido las poses de los RMA en los puntos de interés definidos en la misión. A continuación, en la Tabla 4 se puede ver los vectores que definen la pose.

Tabla 4: Vectores de pose para los objetivos de navegación.

Poses objetivo		
Objetivo	Posición [m]	Orientación [rad]
Sierra	{11,9,0}	{0,0,0,1}
Fresadora	{14,12,0}	{0,0,0.7,0.7}
Torno	{10,13,0}	{0,0,1,0}
Almacén	{7,10,0}	{0,0,0.7,0.7}

En la Figura 6 se aprecia la ubicación y orientación de las poses objetivo en el entorno de simulación.

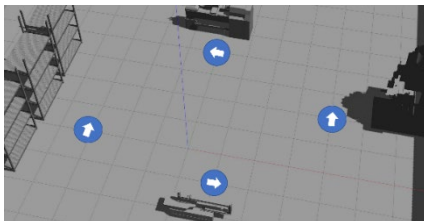


Figura 6: Ubicación y orientación de los objetivos en el entorno virtualizado.

## 4.2. Interfaz de usuario

La interfaz de usuario es necesaria para facilitar la adquisición de datos, así como el envío de órdenes. Para ello, se ha programado un servidor web donde interactuar con el gemelo digital. Un nodo ha sido creado para la comunicación entre el servidor web y el ecosistema ROS. En este contexto, se han adaptado la estructura de los mensajes al estándar de ROS para establecer una pasarela de comunicaciones entre ambos entornos.

## 5. Resultados

De acuerdo con los objetivos propuestos se han alcanzado los siguientes resultados: virtualización de la simulación, navegación, e interfaz de usuario.

### 5.1. Virtualización de la simulación

La virtualización del entorno es un resultado de la incorporación de los RMA al entorno preestablecido. Mediante la virtualización se extrae los parámetros necesarios para la ejecución de los algoritmos.

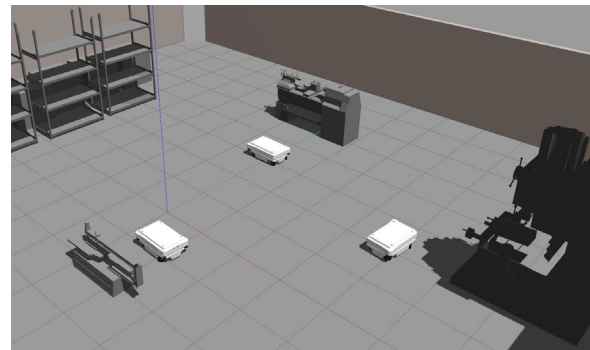


Figura 7: Virtualización del gemelo digital con gazebo.

Como se puede observar en la Figura 7, se ha logrado virtualizar de forma tridimensional el comportamiento de los diferentes RMA para la evaluación de los algoritmos de navegación local en tiempo real.

### 5.2. Navegación

En este apartado se muestran los tiempos de navegación de cada RMA, así como las trayectorias con el fin de compararlas.

#### 5.2.1. Trayectorias

Las trayectorias de los RMA es un indicativo de cómo funcionan los algoritmos de navegación. En este caso, se muestran como resultado las trayectorias reales de los RMA. Para alcanzar la trayectoria resultante, los algoritmos de planificación local se basan también en la trayectoria global, así como en su entorno más cercano. Los autores han empleado la herramienta *rviz*

para obtener las trayectorias resultantes, así como el entorno mapeado. En primer lugar, se mostrarán las trayectorias con el algoritmo de planificación local DWA. Por motivos de

claridad en la Figura 8 solo se muestra la trayectoria resultante obviando las trayectorias globales de cada RMA.

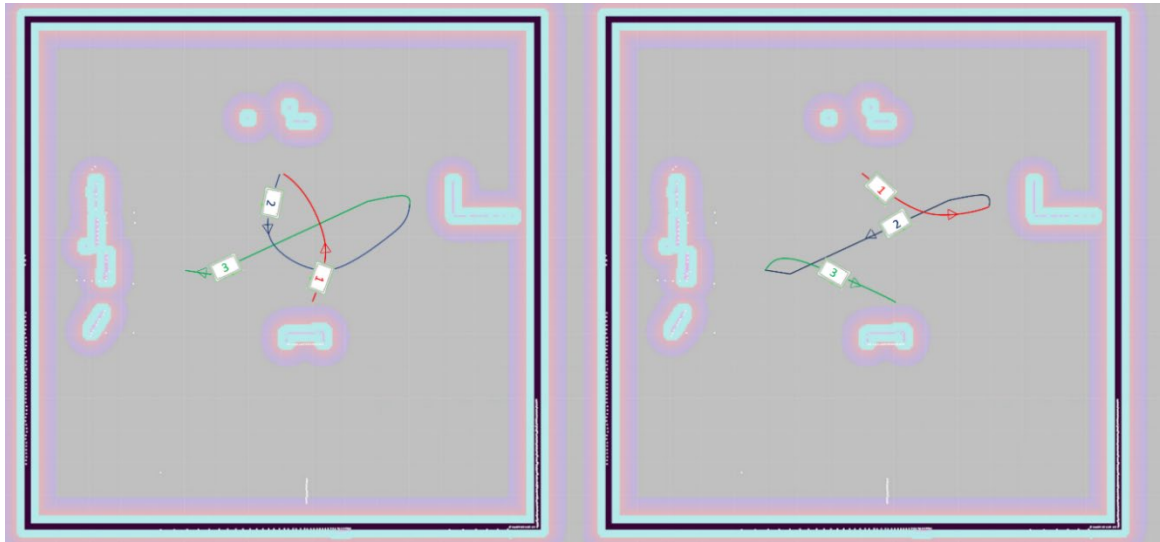


Figura 8: Trayectorias de los RMA en el gemelo digital con el algoritmo de planificación local DWA.

De acuerdo con la Figura 8, cada RMA es diferenciado con un número. Además, la misión se ha dividido en dos mapas como consecuencia de las dos trayectorias.

Del mismo modo que en el algoritmo DWA se presentan las trayectorias del algoritmo de planificación local TEB en la Figura 9.

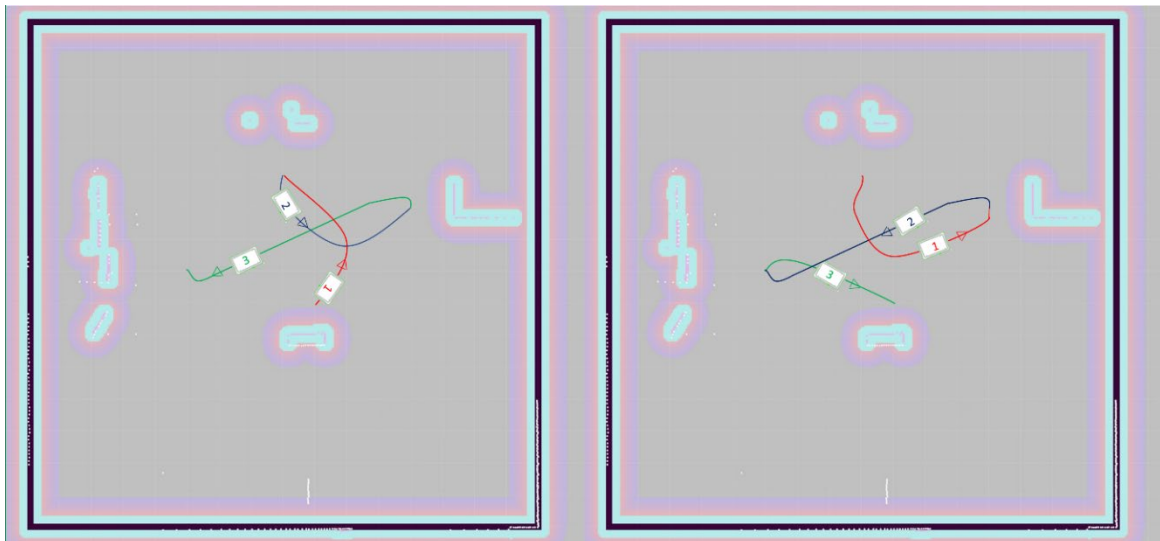


Figura 9: Trayectorias de los RMA en el gemelo digital con el algoritmo de planificación local TEB.

Además, en los mapas de las Figuras 8 y 9 los cuales han sido generados por *rviz* se aprecia el mapa de costos y su valor mediante un degradado de color en tonos azules. Sin embargo, los obstáculos dinámicos provocados por los RMA no han sido representados para mayor claridad.

### 5.2.2. Tiempos de navegación

Otro parámetro de análisis son los tiempos de navegación, los cuales son relevantes para la planificación industrial. No obstante, para la misión se han ejecutado los movimientos en bloque con el fin de aumentar las colisiones para evaluar los algoritmos. De esta forma, hasta que todos los RMA no alcancen el objetivo, no se inicia el siguiente bloque de acciones. Por lo tanto, la misión consta de dos bloques de



acciones: posición de inicio-intermedia e intermedia-final. A continuación, en la Tabla 5 se comparan los tiempos de ambos algoritmos de planificación local.

Tabla 5: Comparación de los tiempos de navegación entre los algoritmos DWA y TEB.

Tiempos de navegación [s]		
Misión	DWA	TEB
Inicial-intermedio	31.23	28.12
Intermedio-final	30.25	25.24
Total	61.48	53.36

De acuerdo con los resultados de la Tabla 5, el algoritmo TEB logra menores tiempos de navegación. No obstante, ante cualquier variación del entorno o cualquiera de los parámetros desencadenan acciones de los RMA diferentes.

### 5.3. Interfaz de usuario

Otro resultado es la interfaz de usuario donde se puede interactuar con el GD desde un navegador web. De esta forma, es más sencillo el acceso a la información, así como la toma de acciones sin la necesidad de manipular el complejo ecosistema ROS.

En este caso, y a modo de demostrador, se han representado los diferentes objetivos que pueden tomar cada uno de los tres RMA mediante casillas. Al presionar el botón, el nodo de navegación establece un nuevo objetivo. Estas acciones generan diversas métricas (e.g., velocidades, aceleraciones, estado de navegación). Por este motivo, se han mostrado en la interfaz las velocidades tanto lineal como angular del primer RMA como se puede apreciar en la Figura 10.

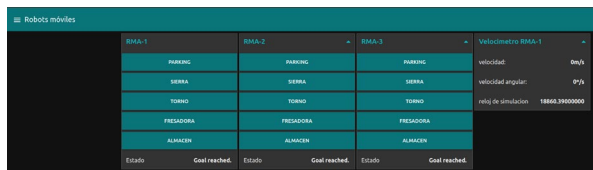


Figura 10: Interfaz del servidor web donde se puede interactuar con el gemelo digital.

Además, se ha incluido el reloj de simulación, debido a que el gemelo digital no se ejecuta en tiempo real. Esta diferencia varía dinámicamente en función de la carga computacional del servidor. Por lo tanto, la referencia de tiempo en la simulación no es el tiempo real si no el de simulación el cual ha sido empleado para evaluar ambos algoritmos.

## 6. Discusión de los algoritmos

Ambos algoritmos tienen el objetivo de planificar o predecir el movimiento en un espacio mediante la optimización de una función objetivo. El algoritmo DWA realiza una optimización basada en muestras

mientras el algoritmo TEB busca la solución de tiempo óptimo, motivo por el cual los tiempos son menores. No obstante, el comportamiento para de TEB es mejor para funciones de costos uniformes puesto que aplica un esquema de optimización numérica continua. En cambio, DWA se adapta mejor a las funciones de costo menos suaves debido a que la optimización se realiza mediante muestras de velocidad en un espacio físico y temporal delimitados.

Si se comparan los resultados obtenidos con otros trabajos [16], [17] se comprueba que los tiempos con el algoritmo TEB son menores que con DWA. Por lo tanto, podemos afirmar que TEB también es mejor con obstáculos dinámicos.

Además, la optimización de trayectorias se acerca a la solución óptima en TEB. Sin embargo, las restricciones en el algoritmo se han implementado mediante penalizaciones, lo que no garantiza el cumplimiento de las restricciones.

En cuanto a la carga computacional, TEB presenta una mayor carga en comparación con el algoritmo DWA. No obstante, esta puede ser configurada en los parámetros del `teb_local_planner`.

## 7. Conclusiones

La manutenzione industrial es un activo estratégico en los procesos productivos el cual requiere un análisis detallado con el fin de establecer las necesidades y costes. En este sentido, los GD son clave para la virtualización del entorno y la generación de información para la toma de decisiones de forma inteligente. En este sentido, se ha analizado dos algoritmos de navegación: DWA y TEB. Para ello, se ha virtualizado un entorno industrial con tres RMA con el fin de adquirir las trayectorias y los tiempos de navegación para una misma misión. Consecuentemente, se ha empleado el ecosistema ROS para evaluación de los algoritmos. De acuerdo con los resultados obtenidos, el algoritmo TEB ofrece mejores tiempos que el DWA para obstáculos dinámicos. Por lo tanto, se puede concluir que TEB sigue siendo mejor al tratar con obstáculos móviles. Asimismo, las trayectorias con TEB resultan más idóneas para la función objetivo, aunque no se garantiza el cumplimiento de las restricciones. Además, se ha desarrollado una interfaz web que permite la interacción entre los usuarios y el GD dentro del ecosistema ROS, facilitando su uso. Por lo tanto, los objetivos iniciales de este trabajo han sido completados con éxito mediante el análisis de algoritmos de navegación local mediante un gemelo digital interactivo.

## Financiación

Esta investigación ha sido desarrollada y financiada por el proyecto del Ministerio de Ciencia e Innovación con número de subvención PID2019-108277GB-C21.

## 8. Referencias

- [1] K. Foit, G. Gołda, and A. Kampa, “Integration and Evaluation of Intra-Logistics Processes in Flexible Production Systems Based on OEE Metrics, with the Use of Computer Modelling and Simulation of AGVs,” *Processes*, vol. 8, no. 12, p. 1648, Dec. 2020, doi: 10.3390/pr8121648.
- [2] H. Lasi, P. Fettke, H. Kemper, ... T. F.-B. & information, and undefined 2014, “Industry 4.0,” *Springer*, doi: 10.1007/s12599-014-0334-4.
- [3] Y. Zhang, Z. Zhu, and J. Lv, “CPS-Based Smart Control Model for Shopfloor Material Handling,” *IEEE Trans Industr Inform*, vol. 14, no. 4, pp. 1764–1775, Apr. 2018, doi: 10.1109/TII.2017.2759319.
- [4] S. Mumtaz, A. Alsoghaily, Z. Pang, A. Rayes, K. F. Tsang, and J. Rodriguez, “Massive Internet of Things for Industrial Applications: Addressing Wireless IIoT Connectivity Challenges and Ecosystem Fragmentation,” *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 28–33, Mar. 2017, doi: 10.1109/MIE.2016.2618724.
- [5] M. L. Hoffmann Souza, C. A. da Costa, G. de Oliveira Ramos, and R. da Rosa Righi, “A survey on decision-making based on system reliability in the context of Industry 4.0,” *J Manuf Syst*, vol. 56, pp. 133–156, Jul. 2020, doi: 10.1016/J.JMSY.2020.05.016.
- [6] R. Siegart, I. Nourbakhsh, D. S.-A. B. Book, and undefined 2011, “Autonomous mobile robots,” *asl.ethz.ch*, Accessed: Sep. 29, 2022. [Online]. Available: [https://asl.ethz.ch/content/dam/ethz/special-interest/mavt/robotics-n-intelligent-systems/asl-dam/documents/lectures/autonomous\\_mobile\\_robots/spring-2015/5\\_Summary\\_Add-on\\_Slides.pdf](https://asl.ethz.ch/content/dam/ethz/special-interest/mavt/robotics-n-intelligent-systems/asl-dam/documents/lectures/autonomous_mobile_robots/spring-2015/5_Summary_Add-on_Slides.pdf)
- [7] A. Martínez-Gutiérrez, J. Díez-González, R. Ferrero-Guillén, P. Verde, R. Álvarez, and H. Perez, “Digital Twin for Automatic Transportation in Industry 4.0,” *Sensors*, vol. 21, no. 10, p. 3344, May 2021, doi: 10.3390/s21103344.
- [8] M. Kosacka-Olejnik, M. Kostrzewski, M. Marczewska, B. Mrówczyńska, and P. Pawlewski, “How Digital Twin Concept Supports Internal Transport Systems?—Literature Review,” *Energies 2021, Vol. 14, Page 4919*, vol. 14, no. 16, p. 4919, Aug. 2021, doi: 10.3390/EN14164919.
- [9] M. Udengard and K. Iagnemma, “Kinematic analysis and control of an omnidirectional mobile robot in rough terrain,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 795–800, 2007, doi: 10.1109/IROS.2007.4398996.
- [10] F. Duchon *et al.*, “Path Planning with Modified a Star Algorithm for a Mobile Robot,” *Procedia Eng*, vol. 96, pp. 59–69, Jan. 2014, doi: 10.1016/J.PROENG.2014.12.098.
- [11] I. Afanasyev, A. Sagitov, and E. Magid, “ROS-based SLAM for a gazebo-simulated mobile robot in image-based 3D model of Indoor environment,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9386, Springer Verlag, 2015, pp. 273–283. doi: 10.1007/978-3-319-25903-1\_24.
- [12] F. R. Q. Aini, A. N. Jati, and U. Sunarya, “A study of Monte Carlo localization on robot operating system,” *2016 International Conference on Information Technology Systems and Innovation, ICITSI 2016 - Proceedings*, Feb. 2017, doi: 10.1109/ICITSI.2016.7858235.
- [13] L. I. U. Tianyu, Y. A. N. Ruixin, W. E. I. Guangrui, and S. U. N. Lei, “Local Path Planning Algorithm for Blind-guiding Robot Based on Improved DWA Algorithm,” *Proceedings of the 31st Chinese Control and Decision Conference, CCDC 2019*, pp. 6169–6173, Jun. 2019, doi: 10.1109/CCDC.2019.8833259.
- [14] Z. Yongzhe, B. Ma, and C. K. Wai, “A Practical Study of Time-Elastic-Band Planning Method for Driverless Vehicle for Auto-parking,” *2018 International Conference on Intelligent Autonomous Systems, ICoIAS 2018*, pp. 196–200, Oct. 2018, doi: 10.1109/ICOIAS.2018.8494025.
- [15] K. Zheng, “ROS Navigation Tuning Guide,” *Studies in Computational Intelligence*, vol. 962, pp. 197–226, 2021, doi: 10.1007/978-3-030-75472-3\_6/FIGURES/17.
- [16] B. Cybulski, A. Wegierska, and G. Granosik, “Accuracy comparison of navigation local planners on ROS-based mobile robot,” *12th International Workshop on Robot Motion and Control, RoMoCo 2019 - Workshop Proceedings*, pp. 104–111, Jul. 2019, doi: 10.1109/ROMOCO.2019.8787346.
- [17] Y. Li, J. Li, W. Zhou, Q. Yao, J. Nie, and X. Qi, “Robot Path Planning Navigation for Dense Planting Red Jujube Orchards Based on the Joint Improved A\* and DWA Algorithms under Laser SLAM,” *Agriculture 2022, Vol. 12, Page 1445*, vol. 12, no. 9, p. 1445, Sep. 2022.