

# Proyecto Fin de Máster Máster en Ingeniería Electrónica, Robótica y Automática

## Planificación de Trayectorias para Vehículos de Superficie no Tripulados

Autor: Víctor Quesada Conejero

Tutor: Ángel Rodríguez Castaño

Dpto. Ingeniería de Sistemas y Automática  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2023





Proyecto Fin de Máster  
Máster en Ingeniería Electrónica, Robótica y Automática

# **Planificación de Trayectorias para Vehículos de Superficie no Tripulados**

Autor:

Víctor Quesada Conejero

Tutor:

Ángel Rodríguez Castaño

Profesor Contratado Doctor Interino

Dpto. Ingeniería de Sistemas y Automática  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2023



Proyecto Fin de Máster: Planificación de Trayectorias para Vehículos de Superficie no Tripulados

Autor: Víctor Quesada Conejero

Tutor: Ángel Rodríguez Castaño

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:



# Agradecimientos

---

A mi familia y amigos





# Resumen

---

**E**n la actualidad los sistemas autónomos nos acompañan en nuestro día a día pero aun quedan muchos retos por superar en este área de conocimiento. Los vehículos autónomos representan un caso de uso muy interesante tanto desde el punto de vista científico como tecnológico y en el mundo naval se comienzan a ver los primeros avances para el desarrollo de sistemas de navegación autónomos.

El objetivo de este trabajo ha sido investigar y desarrollar algoritmos de la familia C-Space sampling-based para generar trayectorias para vehículos de superficie no tripulados. El objetivo principal se ha basado en la integración de restricciones cinemáticas y dinámicas en el algoritmo RRT\* durante la fase de exploración. Mediante experimentos de simulación se ha probado el algoritmo desarrollado y se han evaluado sus características principales. Entre las sugerencias para trabajos futuros se incluyen incorporar información dinámica en la generación de trayectorias y la detección de inexactitudes del modelo en entornos reales debido a los efectos de distintos estados de mar, aumentando así los márgenes de seguridad cuando sea necesario.



# Abstract

---

Nowadays, autonomous systems accompany us in our daily lives, but there are still many challenges to overcome in this area of knowledge. Autonomous vehicles represent a very exciting use case from both a scientific and technological point of view, and the naval world is beginning to see the first advances in the development of autonomous navigation systems.

The goal of this final project has been to research and develop algorithms based on C-Space sampling to generate trajectories for unmanned surface vehicles operating in marine environments. The focus has been on integrating kinodynamic constraints in the RRT\* algorithm during the exploration phase. Through simulation experiments, the developed algorithm has been tested and its key features evaluated. Future work suggestions include incorporating dynamic information in trajectory generation and considering disturbances to estimate the model's error in the real system and to set appropriate safety margins.



# Índice

---

<i>Agradecimientos</i>	I
<i>Resumen</i>	III
<i>Abstract</i>	V
<b>1 Introducción</b>	<b>1</b>
1.1 Trabajo realizado	1
1.2 Proyecto asociado	1
1.3 Estructura del documento	2
<b>2 Estado del arte</b>	<b>3</b>
2.1 Planificación	3
2.2 Planificación con restricciones dinámicas	5
2.3 Sistemas autónomos para embarcaciones	6
<b>3 Arquitectura del sistema e implementación</b>	<b>7</b>
3.1 Arquitectura software	7
3.1.1 Planificador de trayectorias	8
3.1.2 Gestor de misiones	8
3.1.3 Visualizador	8
3.1.4 Mapa de planificación	8
3.2 Medios utilizados	8
3.2.1 Lenguajes utilizados	8
3.2.2 Programas utilizados	8
<b>4 Método de planificación</b>	<b>11</b>
4.1 RRT*-DA	11
4.2 Sistema bucle cerrado	13
<b>5 Experimentos</b>	<b>15</b>
5.1 Mapa de planificación	15
5.1.1 Motivación	15
5.1.2 Condiciones	15
5.1.3 Resultados	15
5.1.4 Conclusiones	16
5.2 Replanificación de trayectorias	16
5.2.1 Motivación	17
5.2.2 Condiciones	17
5.2.3 Resultados	17
5.2.4 Conclusiones	18
5.3 Seguimiento de trayectorias	18
5.3.1 Motivación	19

---

5.3.2	Condiciones	19
5.3.3	Resultados	19
5.3.4	Conclusiones	19
5.4	Tiempos de planificación	19
5.4.1	Motivación	19
5.4.2	Condiciones	20
5.4.3	Resultados	20
5.4.4	Conclusiones	20
<b>6</b>	<b>Conclusiones y trabajos futuros</b>	<b>23</b>
6.1	Conclusiones	23
6.2	Trabajos futuros	23
6.2.1	Información semántica del entorno	23
6.2.2	Evaluación del error del modelo mediante estimación del estado del mar	24
6.2.3	Planificación con posicionamiento dinámico	24
6.2.4	Planificación con COLREGs	24
6.2.5	Replanificación frente a elementos dinámicos	24
6.2.6	Paralelización del proceso de planificación	24
	<i>Índice de Figuras</i>	25
	<i>Bibliografía</i>	27

# 1 Introducción

---

*Yo es que he pensado que a mí también me gustaría ser intelectual, como no tengo nada que perder.*

AMANECE QUE NO ES POCO, 1989

Nos encontramos en un mundo en constante evolución donde la automatización de procesos cada día está más presente en nuestras vidas. En la vida cotidiana puede suponer una mejora en la calidad de vida y comodidad, como pueden ser los ejemplos de un climatizador o algo tan sencillo como preparar un café con el único esfuerzo de pulsar un botón. La industria, por otro lado, centra sus esfuerzos en automatizar los procesos productivos para conseguir una ventaja competitiva en el mercado mediante distintos enfoques: reduciendo costes de producción o mejorando la calidad de sus productos mediante sofisticada maquinaria de producción industrial [5]. La automatización en la conducción de vehículos es un campo con un alto interés desde el punto de vista científico, debido al reto y a la disposición de desarrollos tecnológicos previos que evocan a su abordaje; o desde el punto de vista comercial, ya que, mediante la conducción autónoma, puede aumentarse tanto la seguridad como la comodidad del usuario [2].

Claro ejemplo de lo expuesto puede observarse en el estado actual de los coches autónomos. Empresas como: Tesla, Google, Toyota o Cruise, centran su modelo de negocio en el desarrollo de la tecnología necesaria para ofrecer un producto como el coche autónomo y sostienen una alta popularidad, así como un alto capital invertido. Por otro lado, la navegación de vehículos marinos presenta una oportunidad en la actualidad poco explorada y llena de retos a abordar.

## 1.1 Trabajo realizado

Dentro de este marco se presenta este trabajo de fin de máster, donde se expone una solución al problema de planificación para embarcaciones autónomas que utiliza tanto un modelo del sistema cuya trayectoria se desea planificar como información estática y dinámica del entorno para asegurar la seguridad y la viabilidad de las trayectorias generadas. Uno de los aspectos críticos para la planificación es el tiempo de cálculo, siendo este excesivamente alto para su cálculo "online", uno de los objetivos que se ha impuesto ha sido mantener unos tiempos de cálculo razonables para su uso. Para la validación del sistema se han realizado diversos experimentos para comprobar las distintas funcionalidades y casos de uso que presenta la aplicación desarrollada.

## 1.2 Proyecto asociado

Este trabajo de fin de máster ha sido desarrollado gracias a FAVENTAN, un proyecto liderado por la empresa Navantia junto con la colaboración de la Universidad de Sevilla para la concepción del primer USV desarrollado y fabricado en territorio nacional. Las capacidades de este USV van desde la teleoperación del mismo a la evitación de obstáculos de forma completamente reactiva gracias a los sensores embarcados [16].



**Figura 1.1** Imagen de la embarcación Vendaval.

### **1.3 Estructura del documento**

La estructura de la memoria se divide en los siguientes apartados: primero se ha presentado una introducción del trabajo realizado, seguido por un segundo capítulo centrado en el estado del arte, tercer y cuarto capítulo tratarán sobre la arquitectura software planteada y el método teórico implementado respectivamente y para finalizar un capítulo con los experimentos realizados y otro con las conclusiones obtenidas del trabajo realizado.



## 2 Estado del arte

---

- *¿Es un niño o una niña?*  
- *Creo que es un poco pronto para empezar a imponerle roles, ¿no cree?*

EL SENTIDO DE LA VIDA, 1983

En este capítulo se expone el estado actual del estado del arte de la planificación en general y en concreto dentro de la planificación de trayectorias con restricciones cinemáticas y dinámicas, además de estudios planteados para la planificación de trayectorias para USV.

### 2.1 Planificación

Planificar puede ser definido como el proceso de idear una serie de acciones que nos llevan a un objetivo deseado, este concepto puede aplicarse a campos muy distantes de la robótica como la organización industrial o juegos matemáticos como las torres de Hanói. Uno de los problemas matemáticos más famosos relacionados con el concepto de planificar es el problema del vendedor viajero, también llamado problema del viajante, enunciado por los matemáticos William Rowan Hamilton y Thomas Kirkman. El problema busca la respuesta al siguiente enunciado "Dado una lista de ciudades y una lista de distancia entre cada par de ciudades, ¿Cuál es la ruta con la menor distancia que recorra todas las ciudades una sola vez?", dicho problema puede ser una definición precisa de lo que hoy consideramos en robótica "Path planner".

En concreto, en la robótica aparece una denominación más concreta "Motion planner" que puede ser tratada como un subproblema del "Path planner". En concreto, en "Motion planner" se busca el conjunto de acciones de control que llevan al robot de un estado inicial a un estado final deseado.

En este trabajo nos centraremos en el "Path planning" con algunas consideraciones adicionales (2.2). Como podemos observar en la Figura 2.1, en la actualidad podemos dividir los enfoques teóricos de resolución de trayectorias en los siguientes campos:

- **Reactive-Computing-Based Path Planning:** Esta categoría agrupa los algoritmos de planificación en los que el entorno solo indica la ubicación y la forma de los obstáculos presentes. Generalmente, el entorno conocido es limitado debido a que es generado a través de los sensores a bordo del sistema. Estos algoritmos suelen funcionar como planificadores locales, minimizando el tiempo de cómputo y generando trayectorias de maniobra para evitar obstáculos que aparecen durante el seguimiento de una ruta generada por otro algoritmo de planificación global. Podemos diferenciar dos subcategorías:
  - Reactive Manoeuvre: donde la presencia de un obstáculo genera una nueva maniobra de forma inmediata [10].
  - Local Optimization: donde la actual trayectoria es modificada para evitar el obstáculo detectado [12].
- **Soft-Computing-Based Path Planning:** Esta categoría de algoritmos de planificación se caracteriza por no intentar obtener soluciones exactas, sino aproximarse a ellas con una tolerancia de imprecisión

acotada. Son especialmente ventajosos en problemas con entornos dinámicos, un gran número de variables a tener en cuenta o un sistema con muchos grados de libertad. Por regla general necesitan de un ajuste específico de los parámetros de los algoritmos para cada uno de los casos de uso, lo cual conlleva a la necesidad de realizar experimentos de calibración tanto en simulación como en entornos reales para poder medir el desempeño real del algoritmo. Podemos diferenciar tres subcategorías:

- Evolutionary Computation: Estos algoritmos, también llamados Meta-heurísticos o inspirados en la naturaleza, están inspirados en el comportamiento biológico, dado una serie de individuos iniciales con características únicas entre ellos. Los individuos con mejor resultado van mutando hasta encontrar un individuo con el set de configuración que mejor resultado da para un problema dado.
  - Fuzzy control: Mediante una serie de instrucciones lógicas se consigue desarrollar un sistema de planificación que resuelve problemas como la exploración de zonas no estructuradas.
  - Machine learning: Mediante machine learning se puede generar un sistema de planificación de trayectorias, dependiendo de la técnica usada pueden evaluarse distintas funciones objetivos y conseguir resultados.
- **C-Space-Search-Based Path Planning:** Esta categoría de algoritmos de planificación se caracteriza por definir el espacio de búsqueda como una discretización de todos los posibles estados de configuración del vector de estados del robot. Dentro del espacio de búsqueda se encuentran tanto el estado inicial como el estado final, o al menos estados con un error asumible definido por la discretización realizada. El resultado de estos algoritmos es una sucesión de estados que producen una evolución tal que transforma del estado inicial al estado final. Dichos algoritmos son altamente dependientes de la discretización de los estados y las restricciones impuestas en la transición de estos, es común encontrar algoritmos auxiliares de suavizado a la salida de estos para paliar los efectos de la discretización. Podemos diferenciar dos subcategorías:
    - Graph Search: En este caso el algoritmo dispone de un grafo preexistente, cada uno de dichos nodos representan un estado del sistema, el cual es explorado por un algoritmo para encontrar la mejor ruta posible según la función de coste definida.
    - Sampling-Based: En contraposición con los algoritmos graph search no existen un grafo preexisten y el algoritmo va explorando el espacio de estados, incluso una vez encontrada una solución factible puede continuar explorándolo para mejorar la solución existente.
  - **Optimal-Control-Based Path Planning:** Como el propio nombre de esta categoría indica, el problema de planificación se aborda desde el punto de vista de control óptimo. En este caso, en contraposición al soft-computing, el problema está completamente cerrado y no existen parámetros de ajuste ni tolerancia frente a la solución óptima. Podemos diferenciar dos subcategorías:
    - PDE solving: El algoritmo resuelve una ecuación del tipo Partial Derivate Equation (PDE) sobre una discretización, basado en el Dynamic Programming Principle (DPP).
    - Numerical Optimization: En este caso se resuelve una optimización numérica de una solución inicial existente para que sea factible realizarla dada unas restricciones cinemáticas y dinámicas.



Figura 2.1 Esquema de los distintos enfoques existentes en la actualidad al problema de "Path planning" [17].

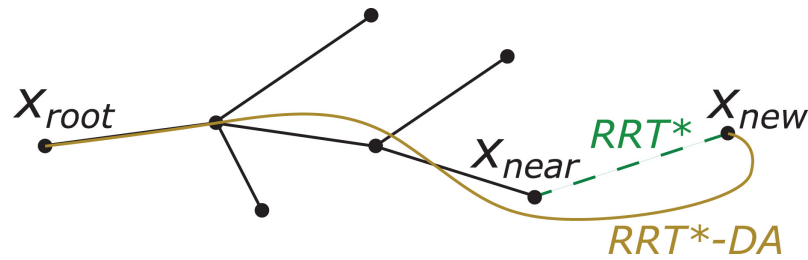
## 2.2 Planificación con restricciones dinámicas

La planificación de trayectorias puede centrarse únicamente en la búsqueda de una serie de estados de un sistema que permiten evolucionar entre un estado inicial hasta un estado objetivo final. Incluso puede centrarse únicamente en la búsqueda de una solución geométrica, estando fuera de su estudio si el sistema tiene la capacidad o no de realizar dicha trayectoria. En esta sección se presentarán algunos planteamientos del problema de planificación con restricciones cinemáticas y dinámicas.

Una de las primeras contribuciones que reflejan la importancia de las restricciones cinemáticas y dinámicas se puede observar en el artículo publicado por LaValle [7], donde se estudia el caso de un planificador "sampling-based" en el que se tiene en cuenta la cinemática a la hora de generar nuevos estados. Se pueden observar resultados de planificaciones cuyo espacio de estado es de dimensión 12 y se planifica sobre un sistema subactuado con cinco entradas. El principal inconveniente de este sistema es su alto coste computacional.

Otra aproximación a la planificación con restricciones plantea dividir el problema en dos etapas [15]. Durante la primera etapa se realiza una planificación sin restricciones cinemáticas, únicamente teniendo en cuenta la geometría del sistema. En la segunda etapa la trayectoria se transforma hasta ser compatible con las restricciones cinemáticas y dinámicas. Una de las principales desventajas de este método es la posibilidad de encontrar rutas en la primera etapa que sean compatibles con la geometría, pero no con las restricciones, lo que produce como resultado un fallo en el algoritmo, pudiendo existir otras rutas que sí sean factible pero impliquen un coste superior en la primera etapa.

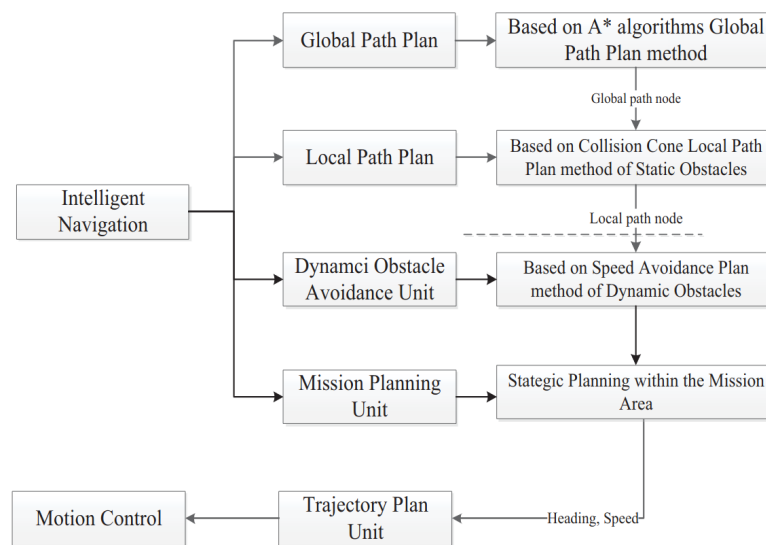
Por último podemos encontrar el trabajo presentado en el artículo [1], donde se plantea un algoritmo de planificación para quadrotors con manipuladores. El objetivo de dicho algoritmo es conseguir trayectorias físicamente factibles mediante la evolución de estados interpolando curvas "spline" de cuarto orden, como se puede observar en la Figura 2.2. Esta es una técnica eficiente desde del punto de vista del coste computacional.



**Figura 2.2** Algoritmo RRT\*-DA: Optimal Rapidly-exploring Random Tree algorithm that incorporates dynamics awareness [1].

### 2.3 Sistemas autónomos para embarcaciones

Para finalizar, en esta sección se expondrá el estado del arte referente a la planificación para USV. Frente al estudio de la planificación para otros sistemas, nos encontramos con una serie de características que lo hacen especialmente interesante y propone un amplio espectro de enfoques en su estudio [21].



**Figura 2.3** Planteamiento de la arquitectura para realización de misiones totalmente autónomas mediante USVs [19].

Una de ellas es la disparidad de distancias entre distintos tipos de misiones, desde algoritmos que consideran la climatología para determinar rutas en largas distancias [18] a planificaciones en el interior de puertos, donde los obstáculos dinámicos toman una importancia crítica y la interpretación de las COLREGs entra en juego para la planificación [3].

Otro punto de vista a evaluar es el tipo de misión que se va a solicitar realizar, pudiendo estar centrada en la realización de una ruta compleja debido al entorno, buscando maximizar la información obtenida de un área delimitada o incluso realizar de forma eficiente el seguimiento de otras embarcaciones. Algunos ejemplos del estado del arte los podemos encontrar en el caso de mantener formaciones de varios USV para el seguimiento de otras embarcaciones [13] o misiones de patrullaje para control de costas [20].

La planificación en USV tiene la peculiaridad del alto número de perturbaciones asociadas, esto provoca que la dimensión del espacio de planificación crece de manera desmedida, lo que provoca que sea de especial interés la planificación bajo control óptimo [8] [9].

# 3 Arquitectura del sistema e implementación

---

*K.I.S.S.*

KELLY JOHNSON, 1960 FRAN REAL, 2022

Tras exponer en los capítulos anteriores aspectos generales del trabajo y el estado del arte actual en planificación de vehículos autónomos de superficie continuaremos con la arquitectura e implementación realizada. Para ello, este capítulo se dividirá en las siguientes secciones: primero, un planteamiento general de la arquitectura, tras el cual, se detallarán las herramientas utilizadas y la justificación de su uso.

## 3.1 Arquitectura software

La definición de una arquitectura software nunca es una tarea sencilla, requiere de un análisis riguroso de las necesidades reales del sistema para poder adaptar soluciones que sean óptimas para el desempeño del mismo. Esto entra en conflicto directo con las metodologías actuales de gestión de proyectos que, alejándose del tradicional método en cascada, buscan una evolución constante de los requisitos del proyecto y una revisión crítica de los anteriores. Es por ello que la arquitectura software no puede ceñirse en buscar el rendimiento como objetivo principal, sino un compromiso entre la flexibilidad y el rendimiento.

En esta arquitectura se ha asociado dicha flexibilidad mediante el desarrollo modular del software. De esta forma, siempre que se respeten las interfaces de comunicación, podemos adaptarnos a los requisitos que vayan siendo descubiertos a lo largo del desarrollo.

Como se puede observar en la figura 3.1, el sistema completo consta de un total de 4 aplicaciones independientes, y cada una de ellas ha sido desarrollada para realizar una tarea concreta, lo cual facilita la depuración de errores del sistema completo.

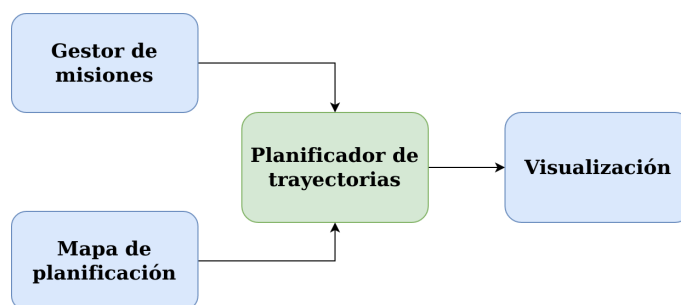


Figura 3.1 Esquema de la arquitectura implementada.

### 3.1.1 Planificador de trayectorias

Se trata del núcleo principal de funcionamiento del sistema. Tras recibir la definición de la misión por parte del gestor de misiones y mediante la información recopilada en el mapa de planificación proporciona una serie de waypoints. Dichos waypoints constituyen una ruta realizable para el sistema cuyo modelo es utilizado para comprobar las colisiones. En el Capítulo 4 se expone en detalle el funcionamiento del método propuesto.

### 3.1.2 Gestor de misiones

Aplicación auxiliar del sistema que proporcionará los waypoints que definen la misión a realizar. Este trabajo se ha realizado mediante la implementación de una aplicación con interfaz gráfica que permite enviar una serie de misiones previamente definidas.

En una implementación rigurosa del sistema, esta función estaría embebida en un sistema HMI que usaría el operario responsable de la embarcación.

### 3.1.3 Visualizador

Aplicación auxiliar del sistema que proporciona una representación gráfica del estado del sistema. Mediante esta aplicación se mostrará por pantalla información de utilidad como: la misión solicitada, el estado de la embarcación, el mapa considerado en la planificación, así como la ruta propuesta.

En una implementación rigurosa del sistema, esta función estaría embebida en un sistema HMI que usaría el operario responsable de la embarcación.

### 3.1.4 Mapa de planificación

Esta aplicación proporcionará la información requerida por la aplicación "Planificador de trayectorias" para poder determinar las zonas navegables. Queda fuera del alcance de este trabajo el funcionamiento de esta aplicación. Desde el punto de vista del sistema completo, esta aplicación recibirá un punto en coordenadas geodésicas, un radio en kilómetros y una profundidad en metros y devolverá un listado de polígonos cerrados complejos definidos por una serie de puntos ordenados en coordenadas geodésicas. Todo elemento devuelto por esta aplicación será tratado como un obstáculo para el sistema.

## 3.2 Medios utilizados

En esta sección se detallarán las herramientas utilizadas para llevar a cabo del trabajo, así como una justificación del uso de estas. Se ha tratado de usar, siempre que ha sido posible, herramientas de software libre.

### 3.2.1 Lenguajes utilizados

Para el desarrollo del proyecto se han utilizado distintos lenguajes de programación, dependiendo de las necesidades de cada tarea:

1. **MATLAB-SimuLink** Se trata del lenguaje sobre el que se ha desarrollado el planificador. La decisión de usar MATLAB y no otro entorno de desarrollo fue tomada debido a la facilidad de integrar el modelado del sistema en bucle cerrado del USV, además de ofrecer una plataforma de depuración fácil e integración con ROS.
2. **C++** Mediante el generador de código que proporciona MATLAB se ha implementado el sistema de planificación en C++ permitiendo mayor eficiencia y, por ende, reduciendo los tiempos de planificación. Dicho esto, supone una desventaja importante a la hora de desarrollar, ya que se ha incluido un paso intermedio entre el desarrollo e implementación que conlleva un aumento sustancial de los tiempos de despliegue y depuración.
3. **Python** Python ha sido utilizado como lenguaje auxiliar para desarrollar aplicaciones de test y de interfaz para realizar simulaciones del sistema completo. Gracias a la facilidad y flexibilidad del lenguaje es idóneo para este tipo de tareas que no requieren un alto coste computacional.

### 3.2.2 Programas utilizados

Los programas utilizados durante el desarrollo del proyecto han sido los siguientes:

1. **ROS** *Robot Operating System* es un *framework* de código abierto que proporciona un entorno estructurado y robusto para el desarrollo de software para robots. Se puede definir como una colección de herramientas, librerías y convenciones que facilitan el desarrollo de software enfocado a la robótica [14].
2. **Git** Por último, he querido mencionar el uso del software de control de versiones durante la totalidad del proyecto. La decisión de usar Git y no otro software es debido a que ya tenía experiencia con el mismo y lo considero una herramienta importante a la hora de desarrollar un proyecto.





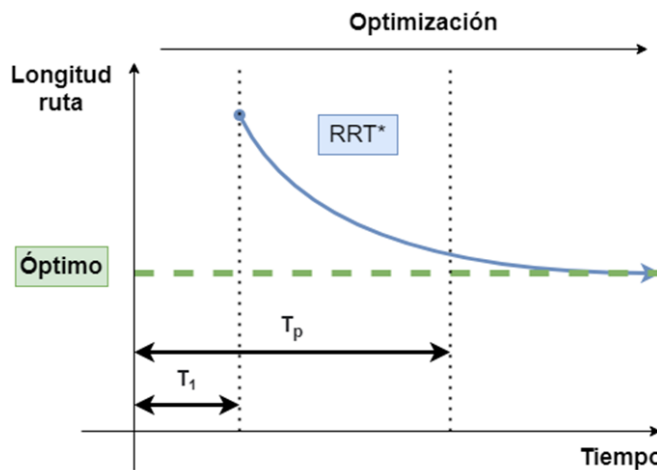
## 4 Método de planificación

*¡Mira detrás de ti! ¡Un mono con tres cabezas!.*

GUYBRUSH THREEPWOOD, 1990

En este capítulo se expondrá el método desarrollado para este trabajo, un algoritmo de planificación de la rama de exploración aleatoria enfocado en asegurar la factibilidad de las rutas generadas mediante la comprobación de colisiones utilizando tanto una geometría aproximada del elemento móvil como un modelo dinámico del mismo. Este capítulo se dividirá en dos secciones: la primera de ellas enfocada al algoritmo de exploración y la segunda a presentar el modelo dinámico empleado por el algoritmo.

### 4.1 RRT\*-DA



**Figura 4.1** Evolución de la longitud de la ruta planificada durante el proceso de planificación mediante un algoritmo RRT\*. La primera solución se encuentra en el instante  $T_1$ . El resto del tiempo de planificación disponible  $T_p$  se invierte en optimizar la solución..

En línea con las consideraciones anteriores, se planteará una estrategia de planificación basada en algoritmos de muestreo aleatorio. En concreto, se pretende hacer uso de la familia de algoritmos RRT\* [6]. Como se representa en la Figura 4.1, el algoritmo RRT\* se caracteriza por ser un algoritmo asintóticamente óptimo, ya que la longitud de la ruta planificada tiende al valor óptimo a medida que se aumenta el tiempo de planificación. Dicho algoritmo RRT\* permite encontrar rutas navegables en tiempos reducidos ( $T_1$  en la figura), optimizando su longitud durante el tiempo de planificación disponible restante ( $T_p$  en la figura). Adicionalmente, con el fin de minimizar los tiempos de cálculo mientras se maximiza la eficiencia de la ruta,

se pretende también hacer uso de técnicas de posprocesado basadas en suavizado de rutas, ya que aportan eficiencia a la ruta sin un incremento significativo en los tiempos de computación.

Más en detalle, el algoritmo RRT\* basa su funcionamiento en la construcción de un árbol de búsqueda que crece de forma aleatoria y uniforme en todo el espacio de planificación (zona delimitada por los límites de operación), siempre comenzando desde el nodo origen (punto inicial). Así, cualquier nodo del árbol estará siempre conectado a través de un número determinado de ramas con el nodo origen, por lo que, tras detenerse el crecimiento del árbol de búsqueda, solo es necesario buscar aquel nodo del árbol que se encuentra más cerca del punto objetivo del USV y reconstruir la ruta hasta llegar al nodo origen. Para la construcción del árbol de búsqueda se sigue el proceso descrito en el Algoritmo 1.

---

**Algorithm 1:** Algoritmo RRT\*.

---

**Data:** *mapa*, punto inicial ( $x_{init}$ ), punto final ( $x_{goal}$ ), *parametros*

**Result:** *ruta*

$Tree \leftarrow INITIALIZATION(x_{init}, parametros);$

**for**  $i \leftarrow 1$  **to**  $iter_{max}$  **do**

$x_{rand} \leftarrow SAMPLE();$

$x_{nearest} \leftarrow NEAREST(Tree, x_{rand});$

$x_{new} \leftarrow STEER(x_{nearest}, x_{rand});$

**if**  $COLLISION(x_{nearest}, x_{new}, mapa)$  **then**

$x_{near} \leftarrow NEAR(Tree, x_{new});$

$Tree \leftarrow ADD(x_{nearest}, x_{near}, x_{new});$

$Tree \leftarrow REWIRE(x_{near}, x_{new});$

**end**

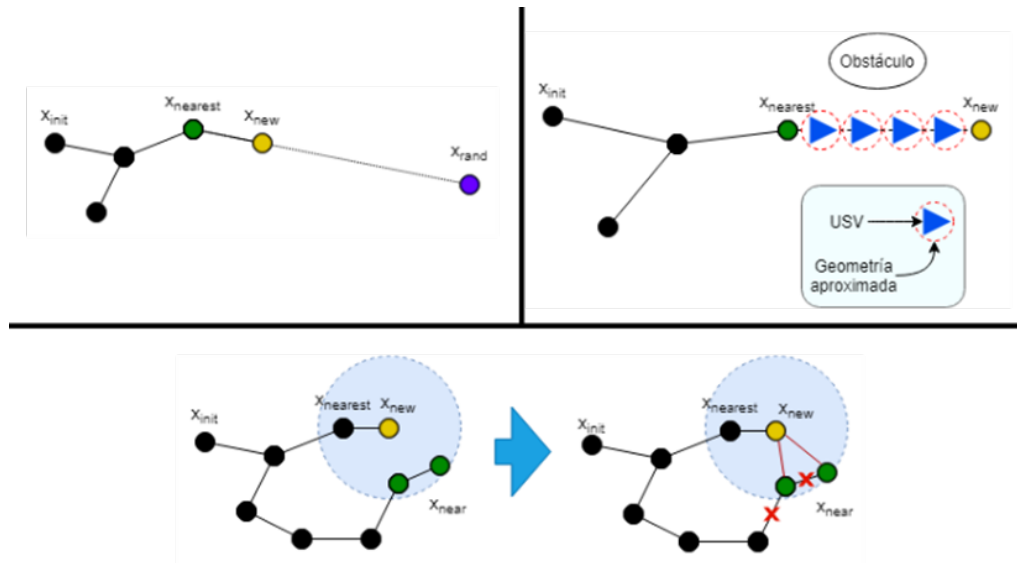
**end**

$ruta \leftarrow ROUTE(Tree, x_{goal});$

---

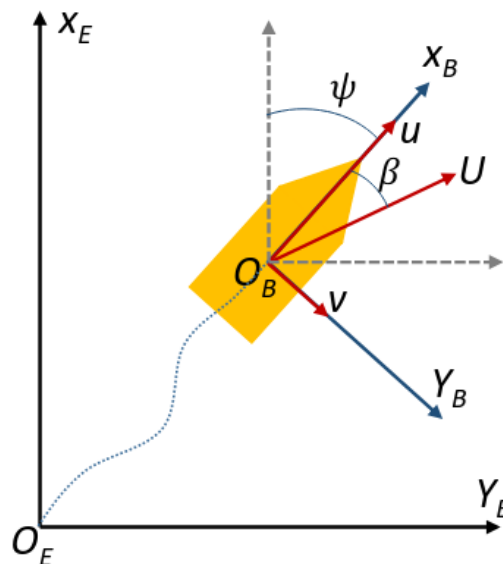
Tras inicializar el árbol de búsqueda *Tree* haciendo uso de todos los parámetros necesarios para la planificación (punto inicial  $x_{init}$ , distancia mínima entre puntos, etc.) a través de la función *INITIALIZATION*, se lleva a cabo un proceso iterativo para la expansión del árbol *Tree*. En cada iteración, el primer paso consiste en proponer un potencial punto de paso aleatorio  $x_{rand}$  dentro del espacio de planificación mediante la función *SAMPLE*. A continuación, de entre todos los puntos ya existentes en el árbol (inicialmente solo el nodo origen), se busca aquel más cercano  $x_{nearest}$  a  $x_{rand}$  en términos de distancia euclídea mediante la función *NEAREST*. Este punto  $x_{nearest}$  será al que se proponga unir el nuevo punto  $x_{rand}$ . Dado que el punto aleatorio  $x_{rand}$  puede estar demasiado cerca o alejado del punto  $x_{nearest}$ , se limita la distancia mínima y máxima del punto  $x_{rand}$  proponiendo para ello un punto alternativo  $x_{new}$  (ver Figura 4 arriba izquierda). Lo anterior se hace con la función *STEER* y evita tanto una expansión demasiado rápida del árbol como que haya puntos de paso muy próximos entre ellos, lo que puede ser incompatible con la dinámica del barco. Una vez disponible el potencial nuevo punto de paso  $x_{new}$  a añadir al árbol *Tree*, se comprueba mediante la función *COLLISION* si la rama que uniría los puntos  $x_{nearest}$  y  $x_{new}$  es una rama libre de conflictos. Para ello, se discretiza dicha rama en una serie de puntos de paso intermedios y en cada uno de ellos se comprueba si la geometría simplificada del barco generaría colisiones con los obstáculos presentes en el entorno (ver Figura 4 arriba derecha). El criterio de colisión dependerá de la geometría que se use para aproximar a la del barco y del modelo dinámico que se utilice para obtener la trayectoria que define la rama. No obstante, en una primera aproximación, la geometría simplificada puede considerarse como un círculo cuyo radio  $r_b$  permita envolver a todo el USV y, por tanto, el criterio de colisión estaría basado en comprobar si alguno de los puntos intermedios se encuentra a una distancia de todos los obstáculos cercanos menor a la del radio  $r_b$  (más una distancia de seguridad  $r_s$ ). A partir de este momento, se pueden dar dos casos. Si se considera que el punto  $x_{new}$  genera un conflicto, este punto es descartado y se comienza con una nueva iteración. En caso de que el punto no genere conflicto, se llevan a cabo tres operaciones adicionales, representadas en la Figura 4 abajo. Primero, mediante la función *NEAR*, se buscan todos los puntos  $x_{near}$  del árbol *Tree* que se encuentren a una distancia del punto  $x_{new}$  menor de un cierto valor  $r_n$ . Segundo, se añade, mediante la función *ADD*, el punto  $x_{new}$  al árbol *Tree* y se conecta al punto  $x_{near}$  (o  $x_{nearest}$ ) que minimice la longitud total de la ruta desde el punto inicial  $x_{init}$  hasta el punto  $x_{new}$  (previa comprobación de que no se genera conflicto entre estos puntos). Finalmente, para el resto de los puntos  $x_{near}$ , se comprueba si pueden ser reconectados al árbol *Tree* a través del nuevo punto  $x_{new}$  de forma que la longitud total de la trayectoria desde el punto inicial  $x_{init}$  hasta dichos puntos se vea reducida. En caso de reducirse, se lleva a cabo la reconexión. Esta última parte del proceso se lleva a cabo

mediante la función REWIRE, y permite la optimización de la ruta planificada. Una vez acabado el proceso iterativo, se extrae la ruta planificada del árbol de búsqueda generado Tree mediante la función ROUTE, reconstruyendo la ruta desde el punto del árbol más cercano al punto final  $x_{goal}$  hasta el punto inicial  $x_{init}$ . Para una descripción más detallada del algoritmo RRT\* puede consultarse [6].



**Figura 4.2** Detalles del funcionamiento del algoritmo RRT\*: función STEER (arriba izquierda), función COLLISION (arriba derecha) y funciones NEAR, ADD y REWIRE (abajo).

## 4.2 Sistema bucle cerrado



**Figura 4.3** Ejes tierra y cuerpo del sistema [4].

El sistema de bucle cerrado utilizado en el planificador ha sido extraído del desarrollo propuesto en [4]. Como se puede observar en la Figura 4.3 se trata del modelo de una embarcación con tres grados de libertad. La

cinemática del sistema queda definida tal que:

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\boldsymbol{\psi})\mathbf{v} \quad \mathbf{R}(\boldsymbol{\psi}) = \begin{bmatrix} \cos(\boldsymbol{\psi}) & -\sin(\boldsymbol{\psi}) & 0 \\ \sin(\boldsymbol{\psi}) & \cos(\boldsymbol{\psi}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

Siendo  $\boldsymbol{\eta} := [x, y, \boldsymbol{\psi}]^T$  el vector que expresa la posición y orientación de la embarcación respecto al Sistema de Referencia Fijo en Tierra,  $\mathbf{v} := [u, v, r]^T$  el vector que contiene las velocidades lineales y angulares del Sistema de Referencia Fijado al Barco y  $\mathbf{R}(\boldsymbol{\psi})$  la matriz de rotación del Sistema de Referencia Fijado al Barco al Sistema de Referencia Fijo en Tierra.

Por otro lado el modelo simplificado de dinámica del sistema queda de la siguiente forma:

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{D}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau}_D + \boldsymbol{\tau}_E \quad (4.2)$$

Representando  $\mathbf{M}$  la masa inercial y masa agregada,  $\mathbf{D}$  la matriz de efectos amortiguadores,  $\boldsymbol{\tau}_D$  son las fuerzas en el plano y el momento realizado por los actuadores del sistema y por último  $\boldsymbol{\tau}_E$  representa las fuerzas externas sufridas por el sistema. Las fuerzas y el momento realizado por los actuadores del sistema  $\boldsymbol{\tau}_D$  quedan definidos por la siguiente ecuación:

$$\boldsymbol{\tau}_D = \begin{pmatrix} \tau_x \\ \tau_y \\ \tau_n \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & -l_x \end{pmatrix} \begin{pmatrix} T_{act} - D_{atc} \\ L_{act} \end{pmatrix} \quad (4.3)$$

Donde  $l_x$  es la distancia del actuador al centro de gravedad de la embarcación,  $T_{act}$  representa el empuje generado por la hélice y  $D_{atc}$  y  $L_{act}$  son tanto el arrastre como las fuerzas transversales generadas por el timón, respectivamente. Dichas magnitudes son generadas por el sistema propulsor de la embarcación, en este caso de estudio se ha considerado un motor fuera-borda, siendo su modelo matemático el siguiente:

$$\begin{aligned} T_{act} &= T_{|n|n}|n|n - T_{|n|u}|n|u_r \\ D_{act} &= D_{|\delta|}|\delta||u_r|u_r \\ L_{act} &= (L_{\delta}\delta - L_{|\delta|}|\delta||\delta|)|u_r|u_r \end{aligned} \quad (4.4)$$

Para finalizar se define el control utilizado en el sistema, para ello se han definido dos estructuras PID para el control del rumbo y la velocidad, definidas en la ecuación 4.5. Los controladores han sido discretizados mediante el método implícito de Euler con un periodo de muestreo de  $T_m = 0.05$  s

$$\begin{aligned} \delta_c(k) &= \underbrace{K_{P_{\psi}}e_{\psi}(k)}_{\delta_P(k)} + \underbrace{T_m K_{I_{\psi}}e_{\psi}(k) + \delta_I(k-1)}_{\delta_I(k)} + \underbrace{\frac{K_{D_{\psi}}}{T_m}(e_{f_{\psi}}(k) - e_{f_{\psi}}(k-1))}_{\delta_D(k)} \\ n_c(k) &= \underbrace{K_{P_u}e_u(k)}_{n_P(k)} + \underbrace{T_m K_{I_u}e_u(k) + n_I(k-1)}_{n_I(k)} + \underbrace{\frac{K_{D_u}}{T_m}(e_{f_u}(k) - e_{f_u}(k-1))}_{n_D(k)} \\ e_{f_i}(k) &= \frac{1}{c_f + 1}(e_i(k) + c_f e_{f_i}(k-1)), \quad e_i(k) = sp_i(k) - i(k) \end{aligned} \quad (4.5)$$

Los errores quedan definidos como:

$$\begin{aligned} e_{\psi}(k) &= sp_{\psi}(k) - \boldsymbol{\psi}(u) \\ e_u(k) &= sp_u(k) - u(k) \end{aligned} \quad (4.6)$$

# 5 Experimentos

---

*El Señor habló diciendo: primero sacaréis el santo sello, luego contaréis hasta tres, no más no menos. Tres es el número al que habréis de llegar y el número del conteo deberá ser tres. No contaréis hasta cuatro, ni tampoco hasta dos, salvo que preceda al tres. El cinco está descartado. Una vez que el tres sea alcanzado lanzaréis la Santa Granada de Mano de Antioquía hacia vuestro enemigo, quien al no ser agradable a mis ojos, se desintegrará.*

LOS CABALLEROS DE LA MESA CUADRADA, 1975

**E**n este capítulo se expondrán los experimentos realizados en simulación para la validación del sistema de planificación desarrollado. Cada uno de los experimentos busca evaluar alguna de las características principales del planificador: como el tiempo de planificación o la capacidad de encontrar rutas en entornos densos. La estructura del capítulo será de una sección para cada experimento, donde se indicarán: descripción general del experimento, la motivación del experimento, condiciones, resultados y conclusiones extraídas.

## 5.1 Mapa de planificación

El mapa de planificación constituye la base para que el algoritmo de exploración consiga generar trayectorias coherentes. Es por ello que el primer experimento a realizar es comprobar el contenido de dicho mapa. En este experimento se evaluará la calidad del mapa generado.

### 5.1.1 Motivación

La motivación principal de este experimento es evaluar la capacidad del generador de mapas de aportar al planificador un entorno de planificación realista.

### 5.1.2 Condiciones

Se solicitarán distintas áreas de planificación al generador de mapas y se representarán junto con las imágenes satélites de las zonas.

### 5.1.3 Resultados

Tras la realización del experimento bajo distintas condiciones se ejemplifica el resultado de este mediante las Figuras 5.1 y 5.2 donde la zona roja representa los obstáculos presentes en el mapa de planificación.



**Figura 5.1** Visualización amplia de los elementos considerados como obstáculos (rojo) sobre el Puerto de Ceuta.



**Figura 5.2** Visualización en detalle de los elementos considerados como obstáculos (rojo) sobre el Puerto de Ceuta.

#### 5.1.4 Conclusiones

Mediante un análisis visual de los mapas de planificación generados respecto a imágenes satélite georeferenciadas se puede observar una alta fidelidad y coherencia del mapa generado respecto a la realidad. Además, se comprueba la existencia de zonas no navegables debido a la poca profundidad de estas aguas.

## 5.2 Replanificación de trayectorias

Las zonas de exclusión son un elemento útil en la planificación de trayectorias. Mediante estos elementos, el operario tiene la capacidad de delimitar peligros que no puedan ser detectados por los sistemas de detección embarcados. En este experimento se evaluará la capacidad del planificador de generar una ruta alternativa una vez detecte un nuevo obstáculo (zona de exclusión) y comprobar su viabilidad tanto en el tiempo de planificación como en el resultado obtenido.

### 5.2.1 Motivación

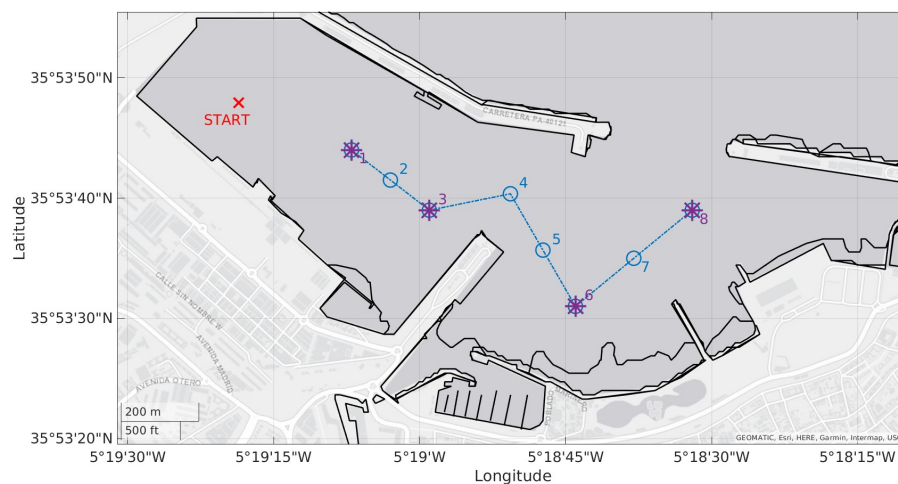
La motivación principal de este experimento es evaluar la capacidad del planificador de calcular una ruta alternativa frente a un elemento reconocido. Adicionalmente, se pondrá de manifiesto la capacidad de generar zonas de exclusión.

### 5.2.2 Condiciones

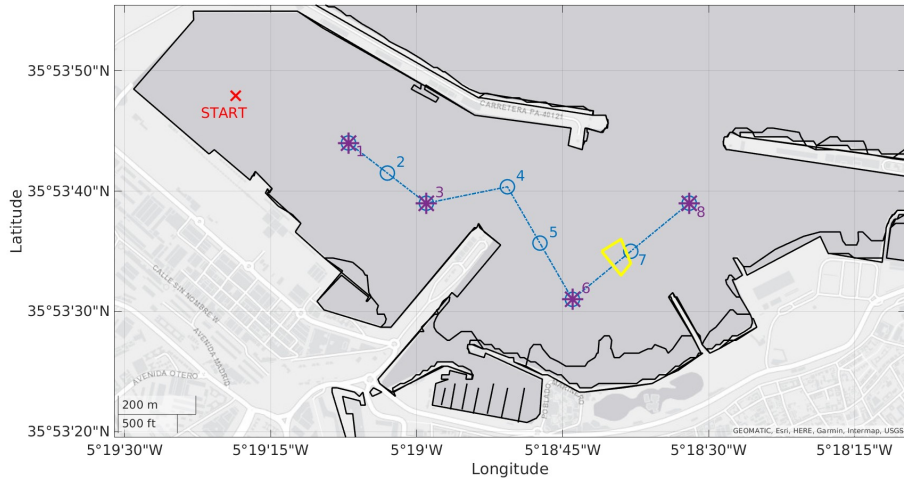
Para el experimento se define una ruta de planificación con colisión con costa, tras recibir la misión planificada se enviará mediante el gestor de zonas de exclusión una zona a validar que entre en conflicto con la ruta generada de forma autónoma. Deberá de producirse una detección del tramo de la ruta no realizable y producirse una nueva planificación para el tramo conflictivo.

### 5.2.3 Resultados

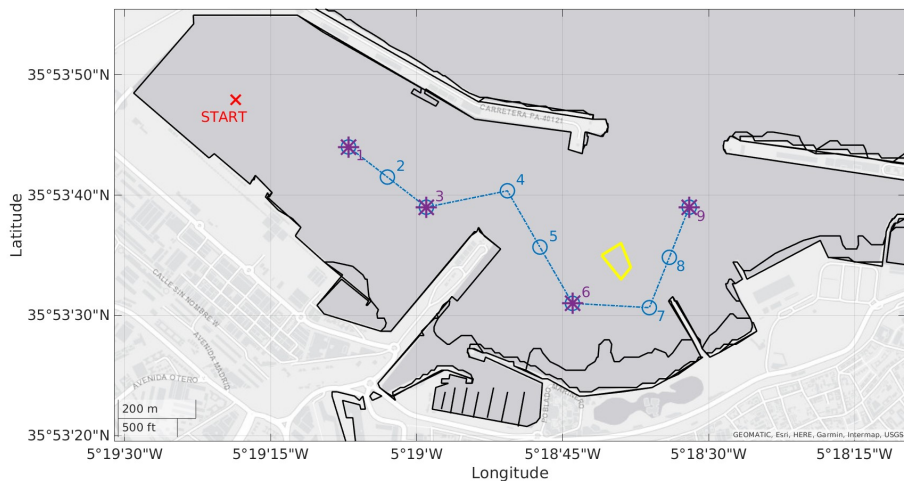
Tras la realización del experimento bajo distintas condiciones se ejemplifica el resultado de este mediante las Figuras 5.3, 5.4 y 5.5 donde las zonas negras representan los obstáculos del mapa de planificación, las marcas violetas presentan los puntos objetivos de la misión, las circunferencias azules representan los waypoints generados por el planificador y la zona amarilla representa una zona de exclusión añadida durante la realización de la misión



**Figura 5.3** Misión inicial planificada para el experimento de replanificación.



**Figura 5.4** Misión inicial planificada para el experimento de replanificación y zona de exclusión validada.



**Figura 5.5** Misión replanificada tras la validación de una zona de exclusión.

## 5.2.4 Conclusiones

Mediante este experimento se ha validado la capacidad del planificador para ofrecer rutas alternativas, evitando así la colisión con elementos no considerados en la planificación inicial.

## 5.3 Seguimiento de trayectorias

Una vez validada la capacidad del planificador de generar rutas coherentes, se continuará realizando experimentos para la validación de la capacidad de generar rutas realizables para un sistema igual al modelo impuesto en las restricciones cinemáticas y dinámicas durante la etapa de exploración. En este experimento se evaluará la capacidad del planificador de surtir al sistema de control de los waypoints para generar las trayectorias de seguimiento. Además, se comprobará que las trayectorias son factibles bajo las restricciones cinemáticas y dinámicas impuestas por el modelo.



### 5.3.1 Motivación

La motivación principal de este experimento es evaluar la capacidad del planificador de calcular una ruta factible para el modelo implementado. Adicionalmente, se evaluará la capacidad del planificador de surtir de puntos objetivo al seguimiento de rutas mientras calcula posibles nuevas colisiones provocadas por las desviaciones del sistema durante la realización de la ruta.

### 5.3.2 Condiciones

Se define una misión con colisión con costa, se generará la ruta de forma autónoma y se realizará una simulación del modelo que complete dicha ruta.

### 5.3.3 Resultados

Tras la realización del experimento bajo distintas condiciones se ejemplifica el resultado de este mediante la Figura 5.6 de las zonas negras representan los obstáculos del mapa de planificación, las marcas violetas presentan los puntos objetivos de la misión, las circunferencias azules representan los waypoints generados por el planificador.

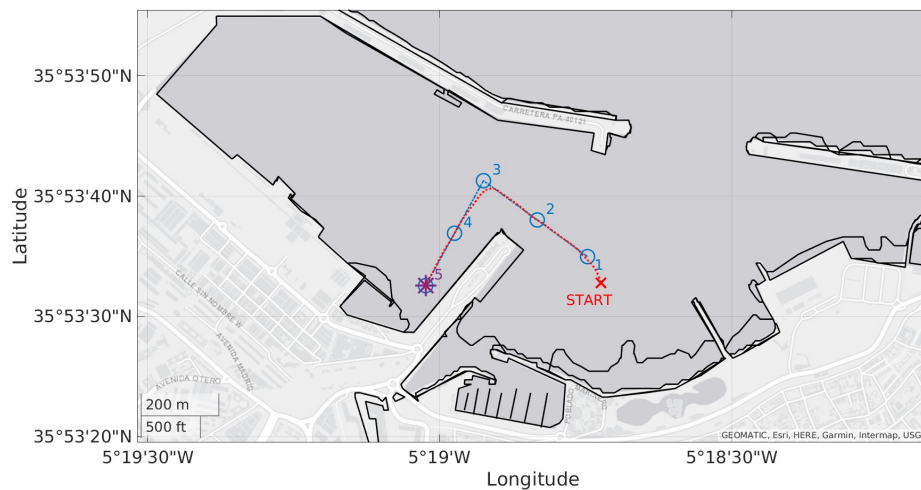


Figura 5.6 Misión planificada y realizada de forma autónoma.

### 5.3.4 Conclusiones

Mediante este experimento se ha validado la capacidad del planificador para generar rutas factibles para las condiciones del modelo implementado, así como la capacidad de aportar waypoints al seguimiento de rutas mientras calcula posibles nuevas colisiones.

## 5.4 Tiempos de planificación

Tras las validaciones anteriores restará comprobar que los tiempos de planificación son suficientemente rápidos como para ser utilizados de forma online. Esto quiere decir sería posible realizar la completa planificación de una ruta dentro de un tiempo tolerable. Los tiempos de planificación son altamente variables debido a una amplia variedad de parámetros que pueden afectar a los cálculos, es por ello que las condiciones de este experimento estarán acotadas bajo suposiciones de uso real.

### 5.4.1 Motivación

La motivación principal de este experimento es evaluar la capacidad del planificador de calcular una ruta factible dentro de un tiempo tolerable que permita su ejecución.

### 5.4.2 Condiciones

Se establecen las siguientes condiciones en este experimento:

1. Se considera un tiempo máximo de planificación de 20 segundos. Si durante este tiempo no se ha encontrado una ruta a efectos de este experimento se considerará como una ruta no realizable.
2. La distancia máxima de la ruta será de 4 kilómetros.
3. La ruta deberá contener tramos que colisionen con costa, esta condición descarta rutas que puedan ser resueltas únicamente uniendo los tramos proporcionados sin añadir WP intermedios.
4. Se realizará un total de 20 ejercicios, un número suficiente para caracterizar el comportamiento no determinista del método presentado.

### 5.4.3 Resultados

Durante distintos ejercicios se recogen los tiempos correspondientes a la obtención de la solución inicial (T1) y el tiempo total de planificación (TP), tal y como se puede observar en la Figura 5.7. Cabe destacar que se han detectado dos comportamientos inesperados durante la experimentación:

El primero de ellos se puede observar en los ejercicios 3 y 8, en los cuales el tiempo de planificación se ve reducido respecto al máximo valor establecido, además de forma no consistente en el instante de finalización. Tras un análisis detallado se ha determinado que este comportamiento se produce al planificar rutas especialmente cortas, las cuales, producen un área de planificación pequeña y produciendo un aumento en la velocidad de cómputo hasta llegar al máximo de iteraciones establecidas, 3000 en estos experimentos. El segundo de ellos se puede observar en el ejercicio 16, en el cual el tiempo de planificación y el tiempo hasta obtener una solución inicial son igual. Este comportamiento se manifiesta tras la incapacidad del planificador de obtener una ruta factible y, por tanto, fallar el experimento.

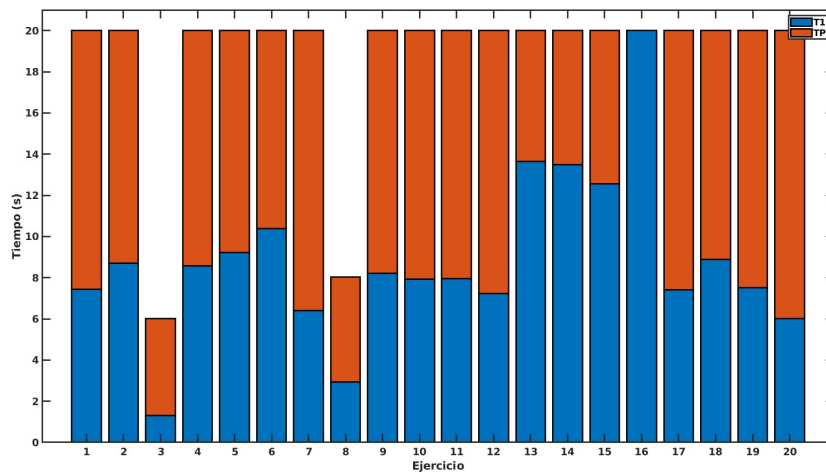
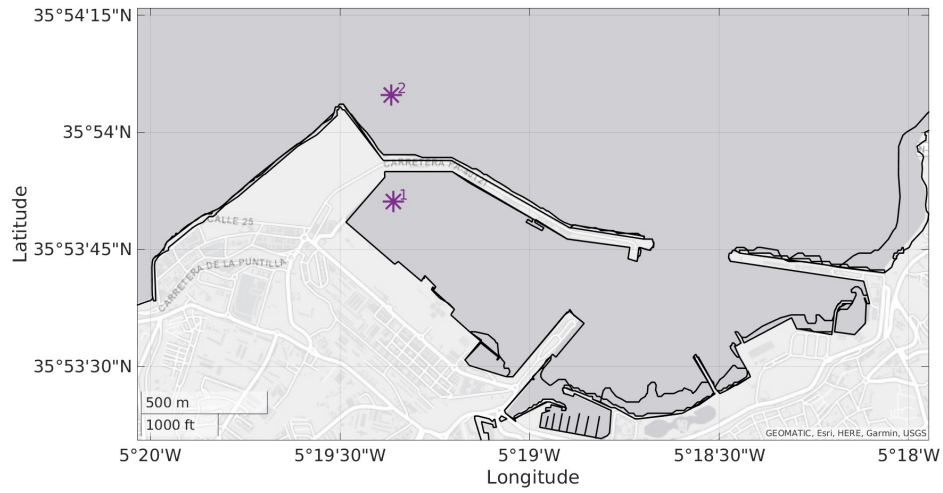


Figura 5.7 Registro de instantes en los que se encuentra una solución inicial..

### 5.4.4 Conclusiones

Mediante este experimento se ha validado la capacidad del planificador para proporcionar rutas factibles bajo limitaciones temporales que habilitan la posibilidad de emplearlo para el cálculo de rutas online. Se ha detectado que bajo misiones especialmente complejas, debido a los obstáculos implicados, se incrementa considerablemente el porcentaje de error. Un caso puede observarse en la Figura 5.8.



**Figura 5.8** Ejemplo de misión especialmente compleja para el planificador debido a los obstáculos implicados.



## 6 Conclusiones y trabajos futuros

---

*Always look on the bright side of life.*

LA VIDA DE BRYAN, 1979

Como capítulo final de este trabajo de fin de máster se presentan las conclusiones recogidas a lo largo de su desarrollo y validación tras realizar los experimentos, así como un apartado final dedicado posibles líneas de trabajos futuros.

### 6.1 Conclusiones

Tras evaluar los experimentos realizados, los siguientes hitos del proyecto han sido superados:

1. **Tiempo de planificación:** Se ha conseguido desarrollar e implementar una metodología de planificación que permite cumplir con las restricciones impuestas en el tiempo de planificación.
2. **Planificación con restricciones cinemáticas y dinámicas:** Queda demostrado en los experimentos que las rutas realizadas mediante las restricciones cinemáticas y dinámicas impuestas por el modelo produce trayectorias factibles para el sistema definido.
3. **Replanificación de trayectorias:** Tras la detección de nuevos obstáculos el planificador genera trayectorias coherentes para la evasión del obstáculo y la continuación con la misión definida.

### 6.2 Trabajos futuros

Debido a la limitación de tiempo algunas ideas se han ido descartando o aplazando a lo largo del desarrollo del proyecto en pos de centrar los recursos en elementos más importantes. Tras el estudio del estado del arte de la planificación de trayectorias y la experiencia ganada tras el desarrollo de este proyecto, surgen las siguientes líneas de trabajo futuro:

#### 6.2.1 Información semántica del entorno

En la planificación de trayectorias para robots marítimos contamos con elemento muy interesante, la cartografía náutica. Como se puede observar en la Figura 6.1 la cartografía contiene un mapa detallado de información valiosa para la planificación como: líneas de costa, veriles, boyas con información referente a la restricción que imponen. Toda esta información y más puede ser integrada en un planificador para obtener rutas que tengan en cuenta la normativa naval.



Figura 6.1 Captura de la cartografía náutica de Ceuta [11].

### 6.2.2 Evaluación del error del modelo mediante estimación del estado del mar

El modelo que se emplea para la definición de las restricciones dinámicas y cinemáticas estaba caracterizado bajo unas condiciones del estado de mar determinada. Mediante la estimación del estado de mar durante la planificación se podría variar la geometría de la embarcación para tener en cuenta posibles inexactitudes del modelo y aumentar la seguridad.

### 6.2.3 Planificación con posicionamiento dinámico

La embarcación del proyecto FAVENTAN cuenta con la posibilidad de ser comandado mediante posicionamiento dinámico, de esta forma el sistema pasa a ser tratado como un sistema holónimo y es posible realizar trayectorias que serían imposibles para el sistema no holónimo. Una posible línea de desarrollo futuro podría ser la planificación bajo estas condiciones en entornos de puerto o durante maniobras de atraque y desatraque.

### 6.2.4 Planificación con COLREGs

Las COLREGs definen la actuación esperada de una embarcación durante una situación de aproximación a otra embarcación, dicho con otras palabras define la actuación frente a situaciones con posibilidad de colisión entre dos embarcaciones. En el proyecto FAVENTAN el sistema reactivo cuenta con una interpretación de dicha normativa, una posible línea de desarrollo futuro sería implementar dicho reglamento en el planificador para aumentar sus capacidades autónomas y seguridad.

### 6.2.5 Replanificación frente a elementos dinámicos

Actualmente, el planificador considera todos los obstáculos como elementos estáticos, velocidad y aceleración nula. Esto es así debido al incremento en el coste computacional que sería integrar en el espacio de planificación el tiempo como variable, además del tratamiento de la estimación de los vectores velocidad y aceleración de los obstáculos. Una línea de futuro abierta en este trabajo es el estudio de la planificación frente a obstáculos dinámicos.

### 6.2.6 Paralelización del proceso de planificación

Aunque en el estado final del proyecto hay procesos que aprovechan las bondades de la paralelización, aún quedan muchas tareas que pueden paralelizarse y con ello mejorar el rendimiento total del sistema. Queda pendiente dicha tarea para trabajos futuros.

# Índice de Figuras

---

1.1	Imagen de la embarcación Vendaval	2
2.1	Esquema de los distintos enfoques existentes en la actualidad al problema de "Path planning" [17]	5
2.2	Algoritmo RRT*-DA: Optimal Rapidly-exploring Random Tree algorithm that incorporates dynamics awareness [1]	6
2.3	Planteamiento de la arquitectura para realización de misiones totalmente autónomas mediante USVs [19]	6
3.1	Esquema de la arquitectura implementada	7
4.1	Evolución de la longitud de la ruta planificada durante el proceso de planificación mediante un algoritmo RRT*. La primera solución se encuentra en el instante T1. El resto del tiempo de planificación disponible $T_p$ se invierte en optimizar la solución.	11
4.2	Detalles del funcionamiento del algoritmo RRT*: función STEER (arriba izquierda), función COLLISION (arriba derecha) y funciones NEAR, ADD y REWIRE (abajo)	13
4.3	Ejes tierra y cuerpo del sistema [4]	13
5.1	Visualización amplia de los elementos considerados como obstáculos (rojo) sobre el Puerto de Ceuta	16
5.2	Visualización en detalle de los elementos considerados como obstáculos (rojo) sobre el Puerto de Ceuta	16
5.3	Misión inicial planificada para el experimento de replanificación	17
5.4	Misión inicial planificada para el experimento de replanificación y zona de exclusión validada	18
5.5	Misión replanificada tras la validación de una zona de exclusión	18
5.6	Misión planificada y realizada de forma autónoma	19
5.7	Registro de instantes en los que se encuentra una solución inicial.	20
5.8	Ejemplo de misión especialmente compleja para el planificador debido a los obstáculos implicados	21
6.1	Captura de la cartografía náutica de Ceuta [11]	24





# Bibliografía

---

- [1] Alvaro Caballero, Manuel Bejar, Angel Rodriguez-Castaño, and Anibal Ollero, Motion planning with dynamics awareness for long reach manipulation in aerial robotic systems with two arms, *International Journal of Advanced Robotic Systems* **15** (2018), no. 3, 1729881418770525.
- [2] Laurène Claussmann, Marc Revilloud, Dominique Gruyer, and Sébastien Glaser, A review of motion planning for highway autonomous driving, *IEEE Transactions on Intelligent Transportation Systems* **21** (2020), no. 5, 1826–1848.
- [3] Yang Gu, Zhenwei Rong, Huzhou Tong, Jia Wang, Yulin Si, and Shujie Yang, Unmanned surface vehicle collision avoidance path planning in restricted waters using multi-objective optimisation complying with colregs, *Sensors* **22** (2022), no. 15.
- [4] Rafael Guardado, Manuel J. López, Jesús Sánchez, and Agustín Consegliere, Autotuning environment for static obstacle avoidance methods applied to usvs, *Journal of Marine Science and Engineering* **8** (2020), no. 5.
- [5] Robert Harrison, Dynamically integrating manufacturing automation with logistics, 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2019, pp. 21–22.
- [6] Sertac Karaman and Emilio Frazzoli, Sampling-based algorithms for optimal motion planning, *CoRR abs/1105.1186* (2011).
- [7] Steven M LaValle and James J Kuffner Jr, Randomized kinodynamic planning, *The International Journal of Robotics Research* **20** (2001), no. 5, 378–400.
- [8] Luis Lüttgens, Benjamin Jurgelucks, Heinrich Wernsing, Sylvain Roy, Christof Büskens, and Kathrin Flaßkamp, Autonomous navigation of ships by combining optimal trajectory planning with informed graph search, *Mathematical and Computer Modelling of Dynamical Systems* **28** (2022), no. 1, 1–27.
- [9] Dongfang Ma, Shunfeng Hao, Weihao Ma, Huarong Zheng, and Xiuli Xu, An optimal control-based path planning method for unmanned surface vehicles in complex environments, *Ocean Engineering* **245** (2022), 110532.
- [10] Helen Oleynikova, Dominik Honegger, and Marc Pollefeys, Reactive avoidance using embedded stereo vision for mav flight, 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 50–56.
- [11] OpenSeaMap, Visor cartográfico online, 2023.
- [12] Lucía Hilario Pérez, Marta Covadonga Mora Aguilar, Nicolás Montés Sánchez, and Antonio Falcó Montesinos, Path planning based on parametric curves, *Advanced Path Planning for Mobile Entities* (Rastislav Róka, ed.), IntechOpen, Rijeka, 2017.
- [13] Guohu Qian, Xiang Zheng, Jianhua Wang, Zhigang Xie, Qiwen Wu, and Wei Xu, Equilateral triangular formation of unmanned surface vehicles for target tracking with event-triggered collision avoidance, *Ocean Engineering* **267** (2023), 113211.

- [14] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Ng, Ros: an open-source robot operating system, vol. 3, 01 2009.
- [15] Charles Richter, Adam Bry, and Nicholas Roy, Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments, pp. 649–666, 04 2016.
- [16] Navantia S.A. S.M.E, Usv embarcación no tripulada, 2023.
- [17] José Ricardo Sánchez-Ibáñez, Carlos J. Pérez-del Pulgar, and Alfonso García-Cerezo, Path planning for autonomous mobile robots: A review, *Sensors* **21** (2021), no. 23.
- [18] Roberto Vettor and C. Guedes Soares, Development of a ship weather routing system, *Ocean Engineering* **123** (2016), 1–14.
- [19] Ai Wei, Leng Yue, Wang Yanfeng, Huang Yong, Chen Guoqing, and Huang Genwang, Design and research of intelligent navigation system for unmanned surface vehicle, 2020 3rd International Conference on Unmanned Systems (ICUS), 2020, pp. 1102–1107.
- [20] Samuel Yanes Luis, Daniel Gutiérrez-Reina, and Sergio Toral Marín, Censored deep reinforcement patrolling with information criterion for monitoring large water resources using autonomous surface vehicles, *Applied Soft Computing* **132** (2023), 109874.
- [21] Chunhui Zhou, Shangding Gu, Yuanqiao Wen, Zhe Du, Changshi Xiao, Liang Huang, and Man Zhu, The review unmanned surface vehicle path planning: Based on multi-modality constraint, *Ocean Engineering* **200** (2020), 107043.