



**Escuela Técnica Superior de Ingeniería  
Universidad de Huelva**

Doble Grado de Ingeniería Electrónica  
Industrial e Ingeniería Mecánica

**Trabajo Fin de Grado**

Desarrollo y simulación de un modelo  
matemático de la dinámica y cinemática de  
un ROV sumergible

Autora: Olga Marín Cañas

Tutores: Fernando Gómez Bravo

Rafael López De Ahumada Gutiérrez

Septiembre 2022





## Agradecimientos

Terminar esta doble titulación ha sido una tarea larga y ardua, y nunca hubiera sido posible sin la ayuda de las muchas personas a las que guardo un afecto especial. A todas ellas quiero expresar mi agradecimiento.

En primer lugar, a mis padres y a mi hermana, por apoyarme durante estos años y aguantar mis momentos de máximo estrés. A mis amigos, por llenar este camino de momentos felices. También a mis compañeros del laboratorio, especialmente a Alejandro Garrocho Cruz y Sara Rua Tirado, por su continua ayuda y motivación.

En segundo lugar, a mi tutores Dr. Fernando Gómez Bravo, director del Departamento de Ingeniería Electrónica, Sistemas Informáticos y Automática de la Escuela Técnica Superior de Ingeniería de la Universidad de Huelva, y Dr. Rafael López De Ahumada Gutiérrez, profesor del grado de Ingeniería Electrónica Industrial, por introducirme en el campo de la investigación y guiarme durante la realización de este trabajo.

Por último, a todo el profesorado del Doble Grado de Ingeniería Electrónica Industrial e Ingeniería Mecánica, por contribuir a mi aprendizaje.





## Resumen

En los últimos años, se han desarrollado diferentes soluciones tecnológicas que apoyan la explotación sostenible de instalaciones acuáticas, tales como piscifactorías o granjas acuícolas, y garantizan su viabilidad. Entre las tecnologías emergentes más prometedoras destaca el uso de vehículos submarinos operados remotamente (ROVs), gracias a su capacidad para obtener información utilizando técnicas no destructivas y mínimamente intrusivas.

Estos vehículos pueden ser más eficientes si navegan bajo el agua de forma autónoma. Por esto es preciso desarrollar sistemas de navegación y algoritmos de control de elevada exactitud, que, a su vez, requieren de modelos dinámicos que reflejen correctamente el funcionamiento del vehículo.

Además, los modelos matemáticos nos permiten analizar y predecir el comportamiento del vehículo submarino en cuestión en distintas circunstancias, y nos ayudan a tomar decisiones y proponer mejoras. En este sentido, los modelos se pueden utilizar para realizar simulaciones que permiten observar dicho comportamiento en varios escenarios sin la participación del ROV, que es costoso y toma más tiempo.

Es por ello que este Trabajo de Fin de Grado se divide de dos fases principales: la primera es el modelado y la identificación del sistema de un ROV acuático, denominado Sibiu PRO, y la segunda es el diseño y desarrollo de una simulación en Matlab del movimiento de este vehículo y del funcionamiento del sonar que el vehículo incorpora, denominado Ping360.

## Palabras clave

Vehículos submarinos operados remotamente, ROV, modelo matemático, simulación, sonar.





## Abstract

In recent years, different technological solutions that support the sustainable exploitation of aquatic facilities, such as fish farms or aquaculture farms, and guarantee their viability, have been developed. Among the most promising emerging technologies to be applied, it is worth highlighting the use of Remotely Operated Vehicles (ROVs), due to their ability to obtain information using non-destructive and minimally intrusive techniques.

These vehicles can be more efficient if they navigate underwater autonomously. For this reason, it is necessary to develop highly accurate navigation systems and control algorithms, which, in turn, require dynamic models that correctly reflect the operation of the vehicle.

In addition, mathematical models allow us to analyze and predict the behavior of the underwater vehicle in question in different circumstances, and help us make decisions and propose improvements. In this regard, models can be used to perform simulations that allow observing such behavior in various scenarios without the involvement of the ROV, which is expensive and takes more time.

That is why this Final Degree Project is divided into two main phases: the first one involves modeling an underwater ROV system, called Sibiu PRO, and the second one includes the design and development of a Matlab simulation of the dynamics of this vehicle and the operation of the sonar that the vehicle incorporates, called Ping360.

## Key words

Underwater Remotely Operated Vehicles, ROV, mathematical model, simulation, sonar.





## Lista de abreviaturas

CAD	Diseño asistido por computador (Computer Aided Design)
CB	Centro de flotabilidad (Centre of Buoyancy)
CFD	Mecánica de fluidos computacional (Computational Fluid Dynamics)
CG	Centro de gravedad (Centre of gravity)
DoF	Grados de libertad (Degree of Freedom)
GCS	Estación de control en tierra (Ground Control Station)
GNC	Guía, navegación y control (Guide, Navigation and Control)
IMU	Unidad de medida inercial (Inertial Measurement Unit)
LiDAR	Light Detection And Ranging
MIMO	Múltiples entradas y múltiples salidas (Multiple Input Multiple Output)
PMM	Mecanismo de movimiento plano (Planar Motion Mechanism)
ROV	Vehículo Operado Remotamente (Remotely Operated Vehicle)



## Lista de figuras

Fig. 1. Sibiu PRO [6].....	11
Fig. 2. Elementos del sistema.....	12
Fig. 3. Movimientos y rotaciones independientes de un vehículo submarino.....	13
Fig. 4. Configuración de los propulsores del Sibiu PRO.....	14
Fig. 5. Sonar Ping360.....	14
Fig. 6. Haz acústico del sonar.....	15
Fig. 7. Muestreo del sonar.....	15
Fig. 8. Imagen obtenida con el sonar.....	15
Fig. 9. Sistema de referencia NED.....	19
Fig. 10. Sistema de referencia BODY.....	19
Fig. 11. Posición del ROV en el plano.....	30
Fig. 12. Fuerza aplicada por el primer propulsor.....	34
Fig. 13. Fuerza aplicada por el segundo propulsor.....	35
Fig. 14. Fuerza aplicada por el tercer propulsor.....	36
Fig. 15. Fuerza aplicada por el cuarto propulsor.....	37
Fig. 16. Fuerza ejercida por cada propulsor.....	38
Fig. 17. Velocidades correspondientes a cada DoF: $v_a$ , $v_l$ , $\omega$ .....	39
Fig. 18. Canales de flujo a) Memorial University of Newfoundland [12]; b) University of Southampton [13]; c) Laboratorio de Mecánica de Fluidos, ETSI, Universidad de Huelva.....	42
Fig. 19. Mecanismo de movimiento plano [14].....	42
Fig. 20. Simulación de las líneas de corriente sobre la estructura de un ROV en ANSYS [8].....	44
Fig. 21. Modelado del Sibiu PRO en SolidWorks [15].....	45
Fig. 22. Sección transversal del ROV [15].....	45
Fig. 23. Simulación de las líneas de corriente sobre el Sibiu PRO [15].....	46
Fig. 24. Momento de inercia respecto al eje vertical.....	47
Fig. 25. Control del ROV mediante el mando.....	50
Fig. 26. Simulación manual (I).....	51
Fig. 27. Representación gráfica de los empujes (I).....	51
Fig. 28. Representación gráfica de las velocidades (I).....	52



Fig. 29. Simulación manual (II) .....	52
Fig. 30. Representación gráfica de los empujes (II) .....	52
Fig. 31. Representación gráfica de las velocidades (II).....	53
Fig. 32. Simulación manual (III).....	53
Fig. 33. Representación gráfica de los empujes (III).....	54
Fig. 34. Representación gráfica de las velocidades (III) .....	54
Fig. 35. Diagrama de flujo (I).....	55
Fig. 36. Secuencia de la simulación (I) .....	57
Fig. 37. Simulación del LiDAR original.....	58
Fig. 38. Diagrama de flujo (II) .....	58
Fig. 39. Simulación del Ping360 (I).....	59
Fig. 40. Simulación del Ping360 (II) .....	60
Fig. 41. Diagrama de flujo (III).....	61
Fig. 42. Simulación del Ping360 (III) .....	61
Fig. 43. Simulación del Ping360 (IV).....	62
Fig. 44. Simulación del Ping360 (V) .....	62
Fig. 45. Navegación realizando transectos.....	64
Fig. 46. Diagrama de flujo (II) .....	65
Fig. 47. Diagrama de flujo (V) .....	65
Fig. 48. Secuencia de la simulación (II).....	66
Fig. 49. Pruebas experimentales (I) [21] .....	67
Fig. 50. Navegación hacia objeto .....	68
Fig. 51. Diagrama de flujo (VI).....	68
Fig. 52. Diagrama de flujo (III).....	69
Fig. 53. Muestreo del entorno de 360°.....	69
Fig. 54. Identificación de objetos.....	70
Fig. 55. Interfaz para la selección del objetivo .....	70
Fig. 56. Secuencia de simulación (III).....	71
Fig. 57. Muestreo con el sonar Ping360 [21].....	72
Fig. 58. Pruebas experimentales (II) [21].....	72



# Índice

Agradecimientos .....	III
Resumen .....	V
Abstract.....	VII
Lista de abreviaturas .....	IX
Lista de figuras.....	XI
Capítulo 1: Introducción .....	1
1.1. Antecedentes .....	2
1.2. Motivación .....	4
1.3. Objetivos.....	4
1.4. Estado del arte y definición del proyecto.....	6
Capítulo 2: Descripción del Sibiu PRO .....	9
2.1. Vehículos operados remotamente .....	9
2.2. Sibiu PRO.....	11
2.2.1. Características generales.....	11
2.2.2. Grados de libertad y diseño estructural.....	12
2.3. Sonar Ping360.....	14
Capítulo 3: Modelo dinámico.....	17
3.1. Notación.....	17
3.2. Sistemas de referencia.....	19
3.3. Transformaciones entre los sistemas de referencia.....	20
3.3.1. Transformación de la velocidad lineal.....	20
3.3.2. Transformación de la velocidad angular .....	21



3.3.3. Matriz de rotación global.....	22
3.4. Ecuaciones del modelo .....	22
3.4.1. Matriz de inercia.....	24
3.4.2. Matriz de Coriolis .....	25
3.4.3. Matriz de resistencia hidrodinámica .....	26
3.4.4. Vector de fuerzas de flotación y gravitacional .....	27
Capítulo 4: Modelo dinámico simplificado .....	29
4.1. Ecuaciones del modelo propuesto.....	30
4.2. Modelado de los propulsores .....	34
4.3. Estimación de los coeficientes hidrodinámicos.....	41
4.3.1. Metodologías de estimación .....	41
4.3.2. Coeficientes hidrodinámicos del Sibiu PRO.....	44
4.4. Estimación del resto de parámetros.....	47
Capítulo 5: Simulación en Matlab.....	49
5.1. Simulación del movimiento del ROV .....	49
5.1.1. Movimiento manual .....	50
5.1.2. Movimiento automatizado .....	54
5.2. Simulación del sonar Ping360.....	57
5.3. Simulación conjunta del ROV y sonar Ping360.....	63
5.3.1. Navegación realizando transectos.....	63
5.3.2. Navegación hacia un objeto .....	67
Capítulo 6: Conclusiones.....	73
6.1. Resultados.....	73



6.2. Trabajo futuro .....	73
Bibliografía .....	75
Anexo I: Código.....	77
A. Ecuación de estado del ROV y representación de empujes .....	77
B. Sonar Ping360 .....	82
C. Representar haces del sonar Ping360 .....	83
D. Simulación del movimiento realizando transectos.....	84
E. Simulación del movimiento de navegación hacia objeto .....	91





# Capítulo 1

## Introducción

---

Este Trabajo Fin de Grado (TFG) se centra en el modelado matemático y simulación por ordenador del Sibiu PRO. Este es un Vehículo Operado Remotamente (ROV) sumergible, con el que se pueden desarrollar trabajos de inspección y mantenimiento de instalaciones acuáticas de manera eficiente y segura, ya que cuenta con la capacidad de recoger y transmitir información sobre el estado general del entorno que lo rodea.

Para ello, el vehículo dispone de diversos sensores, entre los que destacan sensores de presión, cámaras, sensores sonar, etc. Estos últimos son especialmente útiles en circunstancias donde la visibilidad es baja, pues permiten obtener imágenes acústicas del entorno donde se sitúa el vehículo.

En los últimos años, el desarrollo de este tipo de vehículos acuáticos ha crecido significativamente, lo que ha permitido su uso en una amplia variedad de aplicaciones, tales como la exploración, estudio y mantenimiento de hábitats marinos. En estas aplicaciones, el ROV generalmente es dirigido por un operador humano.

Puesto que las condiciones ambientales pueden aumentar la dificultad de maniobrar eficazmente el ROV, diversos esfuerzos de investigación se han centrado en mejorar la autonomía del vehículo, minimizando así la necesidad de la presencia de operadores.

Sin embargo, antes de implementar sistemas de control, es de gran utilidad desarrollar modelos matemáticos que reflejen de forma precisa el comportamiento de estos vehículos. El desarrollo de estos modelos es complejo y requiere mucho tiempo, tanto en el desarrollo teórico como en las pruebas experimentales.

Esta complejidad se debe a las propiedades altamente no lineales de un ROV, parámetros desconocidos en los modelos, información de estado incompleta proporcionada por sensores cuyas mediciones contienen ruido, etc. Es por esto que resulta indispensable estudiar las posibles simplificaciones que faciliten el modelado, a la vez que preserven su exactitud con la realidad.

Además, los modelos matemáticos nos permiten analizar y predecir el comportamiento del vehículo submarino en cuestión en distintas circunstancias, y nos ayudan a tomar decisiones y proponer mejoras. En este sentido, los modelos se pueden utilizar para realizar simulaciones que permiten observar dicho comportamiento en varios escenarios sin la participación del ROV, que es costoso y toma más tiempo.

### 1.1. Antecedentes

El presente trabajo constituye el Trabajo Fin de Grado de Ingeniería Mecánica, siendo el segundo de dos TFG realizados para la obtención del título de Doble Grado de Ingeniería Electrónica Industrial e Ingeniería Mecánica por la Universidad de Huelva. Ha sido realizado por Olga Marín Cañas, estudiante del mencionado doble grado. Se lleva a cabo bajo la supervisión del Dr. Fernando Gómez Bravo, director del Departamento de Ingeniería Electrónica, Sistemas Informáticos y Automática de la Escuela Técnica Superior de Ingeniería de la Universidad de Huelva, y del Dr. Rafael López De Ahumada Gutiérrez, profesor del grado de Ingeniería Electrónica Industrial.

Este TFG se enmarca en el proyecto internacional KTT SeaDrones (POCTEP 0622\_KTTSEADRONES\_5\_E) cofinanciado por el Fondo Europeo de Desarrollo Regional FEDER a través del Programa Interreg V-A España-Portugal (POCTEP) 2014-2020. Este proyecto promueve el conocimiento y la transferencia de tecnología sobre vehículos aéreos y acuáticos para el desarrollo transfronterizo de ciencias marinas y pesqueras. La innovación tecnológica y la transferencia son factores esenciales para lograr el desarrollo del sector pesquero-acuícola regional a la vez que se protege el medio ambiente y se fomenta un uso racional y sostenible de los recursos naturales. En concreto, este TFG se ha llevado a cabo en los últimos meses de duración del proyecto, por lo que su contenido se ha desarrollado a partir del trabajo previo realizado por otros investigadores.

Asimismo, este proyecto ha contado con la concesión de la Beca de Colaboración convocada por el Banco Santander y la Universidad de Huelva, cuyo objetivo es la promoción entre el estudiantado universitario de la mejora de su empleabilidad y la potenciación de su capacidad futura de emprendimiento en los ámbitos tecnológico, de la investigación y el conocimiento.

En este trabajo se han desarrollado las siguientes competencias vinculadas con el Grado en Ingeniería Mecánica de la Universidad de Huelva:



Competencias básicas	
<b>CB1</b>	Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área general de estudio que parte de la base de educación secundaria general, y se suele encontrar un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.
<b>CB2</b>	Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.
<b>CB3</b>	Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.
<b>CB4</b>	Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
<b>CB5</b>	Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

Competencias generales	
<b>G01</b>	Capacidad para la resolución de problemas.
<b>G02</b>	Capacidad para toma de decisiones.
<b>G03</b>	Capacidad de organización y planificación.
<b>G04</b>	Capacidad de aplicar los conocimientos en la práctica.
<b>G06</b>	Actitud de motivación por la calidad y mejora continua.
<b>G07</b>	Capacidad de análisis y síntesis.
<b>G08</b>	Capacidad de adaptación a nuevas situaciones.
<b>G09</b>	Creatividad y espíritu inventivo en la resolución de problemas científico-técnicos.
<b>G10</b>	Capacidad para comunicarse con personas no expertas en la materia.
<b>G12</b>	Capacidad para el aprendizaje autónomo y profundo.
<b>G14</b>	Capacidad de gestión de la información en la solución de situaciones problemáticas.
<b>G16</b>	Sensibilidad por temas medioambientales.
<b>G17</b>	Capacidad para el razonamiento crítico.

Competencias transversales	
<b>TC2</b>	Desarrollo de una actitud crítica en relación con la capacidad de análisis y síntesis.
<b>TC3</b>	Desarrollo de una actitud de indagación que permita la revisión y avance permanente del conocimiento.
<b>TC4</b>	Capacidad de utilizar las Competencias Informáticas e Informacionales (CI2) en la práctica profesional.

Competencias específicas	
<b>B01</b>	Capacidad para la resolución de los problemas matemáticos que puedan plantearse en la ingeniería. Aptitud para aplicar los conocimientos sobre: álgebra lineal; geometría; geometría diferencial; cálculo diferencial e integral; ecuaciones diferenciales y en derivadas parciales; métodos numéricos; algorítmica numérica; estadística y optimización.

<b>B02</b>	Comprensión y dominio de los conceptos básicos sobre las leyes generales de la mecánica, termodinámica, campos y ondas y electromagnetismo y su aplicación para la resolución de problemas propios de la ingeniería.
<b>B03</b>	Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.
<b>B05</b>	Capacidad de visión espacial y conocimiento de las técnicas de representación gráfica, tanto por métodos tradicionales de geometría métrica y geometría descriptiva, como mediante las aplicaciones de diseño asistido por ordenador.
<b>C02</b>	Conocimientos de los principios básicos de la mecánica de fluidos y su aplicación a la resolución de problemas en el campo de la ingeniería. Cálculo de tuberías, canales y sistemas de fluidos.
<b>C04</b>	Conocimiento y utilización de los principios de teoría de circuitos y máquinas eléctricas. Conocimientos de los fundamentos de la electrónica.
<b>C05</b>	Conocimientos de los fundamentos de la electrónica.
<b>C06</b>	Conocimientos sobre los fundamentos de automatismos y métodos de control.
<b>E06</b>	Conocimiento aplicado de los fundamentos de los sistemas y máquinas fluidomecánicas.

## 1.2. Motivación

En los últimos años, los Vehículos Operados Remotamente submarinos se han convertido en elementos muy útiles y fiables para el ser humano, puesto que su sofisticado diseño permite en muchas ocasiones realizar trabajos peligrosos sin necesidad de poner en riesgo la vida de las personas.

Los resultados de este proyecto contribuirán al conocimiento y desarrollo de estos vehículos, especialmente al estudio de su comportamiento en diferentes circunstancias, a través de una simulación en ordenador. Esto permite diseñar algoritmos que controlen el movimiento del vehículo sin contar con la presencia del ROV, ahorrando recursos y tiempo.

## 1.3. Objetivos

Este Trabajo Fin de Grado se centra en el modelado matemático y simulación por ordenador del Sibiu PRO. Realizar la simulación del vehículo en un ordenador nos permite estudiar su comportamiento sin la participación directa del vehículo, lo que conlleva numerosas ventajas.

Para obtener una simulación precisa, es necesario disponer del modelo hidrodinámico del vehículo. Este proceso de modelado implica ciertos aspectos fundamentales, tales como llevar a cabo un análisis previo del sistema, estudiar las posibles simplificaciones, estimar los parámetros relevantes, etc.

Por otro lado, como se explica con más detalle posteriormente, uno de los sensores más importantes del Sibiú PRO es el sonar Ping360, que le permite obtener una imagen acústica del entorno que lo rodea. Es por esto que la simulación de este sensor adquiere asimismo gran relevancia.

El desarrollo de los objetivos del proyecto, dada su mayor complejidad, se divide en diferentes paquetes de trabajo, que se describen a continuación.

☒ Primer paquete de trabajo: desarrollo del modelo matemático

---

**Tarea 1**

Desarrollar un modelo matemático de la dinámica y cinemática del ROV.

**Tarea 2**

Establecer ciertas suposiciones con respecto al ROV y al entorno que simplifiquen el modelo.

**Tarea 3**

Establecer una estimación de los parámetros inerciales e hidrodinámicos del vehículo.

☒ Segundo paquete de trabajo: desarrollo de la simulación en un ordenador

---

**Tarea 4**

Desarrollar un modelo de simulación del movimiento del vehículo que incluya las características dinámicas del ROV y la arquitectura de control existente.

**Tarea 5**

Validar con resultados experimentales tanto el modelo dinámico como el modelo de simulación.

**Tarea 6**

Desarrollar un modelo de simulación para el sensor sonar Ping360.

**Tarea 7**

Integración de la simulación del movimiento del vehículo junto con la simulación del sonar Ping360.

## 1.4. Estado del arte y definición del proyecto

El modelado hidrodinámico y control de vehículos marinos ha sido ampliamente estudiado durante las últimas décadas por numerosos investigadores [16], que han tratado de derivar la dinámica de este tipo de sistemas mediante diferentes métodos.

En 1990, Ioi e Itoh emplearon las ecuaciones de la mecánica clásica de Newton-Euler para determinar una fórmula recursiva que describiera el comportamiento dinámico del vehículo submarino [17]. En 1991, Mahesh, Vuh y Lakshmi derivaron un modelo de tiempo discreto del sistema [18], mientras que Janocha y Papadimitriou llevaron a cabo un estudio de simulación sobre el comportamiento dinámico de un robot submarino, y presentaron un algoritmo recursivo para determinar su modelo dinámico [19].

En 1994, Thor I. Fossen [20] deriva un modelo matemático para todo tipo de embarcaciones marinas con 6 DoF, mediante ecuaciones diferenciales en forma matricial. De hecho, su modelo propuesto ha sido adoptado por la comunidad internacional como un modelo estándar para el diseño de sistemas de control de vehículos marinos.

Por lo tanto, en este trabajo, se expone el marco teórico propuesto por Fossen [1], [20] para el modelado de vehículos marinos. Posteriormente, se adapta este modelo a un vehículo en particular, Sibiu PRO, simplificando y determinando los parámetros relevantes. Finalmente, se lleva a cabo una simulación en ordenador, tanto del vehículo en cuestión, como del sensor sonar que incorpora en su estructura, el sonar Ping360.

Esta memoria se compone de 6 capítulos:

- En el primer capítulo se exponen los objetivos planteados, y se establecen las diferentes etapas a llevar a cabo durante el proceso de modelado del vehículo y su posterior simulación.
- El segundo capítulo describe el Sibiu PRO, vehículo cuyo modelado y simulación se han llevado a cabo.
- El tercer capítulo describe el modelado matemático del vehículo submarino.
- El cuarto capítulo incluye una simplificación del modelo presentado en el capítulo anterior.



- El quinto capítulo se centra en la simulación diseñada en Matlab a partir del modelo dinámico propuesto.
- El sexto capítulo termina con las conclusiones obtenidas de todo el proceso llevado a cabo a lo largo del presente documento.



## Capítulo 2

### Descripción del Sibiu PRO

---

Los vehículos submarinos se clasifican generalmente en dos categorías: vehículos tripulados y no tripulados. Dentro de los vehículos no tripulados, podemos distinguir entre Vehículos Submarinos Autónomos (AUVs), Vehículos Operados Remotamente (ROVs) y Vehículos Operados Remotamente Híbridos (HROVs).

Un AUV puede viajar bajo el agua largas distancias de forma independiente, sin tener cables conectados ni recibir indicaciones o comandos de los operadores. En cambio, un ROV es controlado por un operador a través de un cable umbilical y, generalmente, opera a bajas velocidades.

La ventaja que presentan los HROVs es que, además de poder realizar las mismas funciones automatizadas que un AUV, permiten la intervención humana o incluso el control humano total en su navegación. De esta forma, en un HROV podemos complementar el control automático y manual, según lo requieran las circunstancias.

#### 2.1. Vehículos operados remotamente

Los primeros ROVs se comenzaron a utilizar en la década de los 60 para realizar operaciones en el ámbito militar [18], tales como rastreo de naves, mapeo de campos de escombros o minas submarinas, preparación y ejecución de actividades de rescate y salvamento, actividades de seguridad y protección de instalaciones costeras y submarinas, entre otras.

En los últimos años de la década de los 90, se hicieron las primeras investigaciones sobre la evaluación de poblaciones de la fauna acuática [10]. Desde mediados de esta década, ya se apreciaban los beneficios que estos equipos aportaban a la sociedad respecto a las metodologías tradicionalmente usadas, esto es, tienen una alta capacidad de maniobra, respetan totalmente el medio que les rodea, en el sentido de que no llegan a producir variaciones sobre el mismo, o estas variaciones son mínimas, etc. [16].

A lo largo de los años se fue investigando hasta llegar a conseguir la combinación de la videograbación junto a la obtención de datos físico-químicos in situ, pudiendo realizar estudios y análisis precisos de mega faunas en condiciones realmente complicadas.

En la actualidad, los ROVs son elementos muy útiles y fiables para el ser humano, puesto que su sofisticado diseño permite en muchas ocasiones realizar trabajos peligrosos sin necesidad de poner en riesgo la vida de las personas. Algunas de las aplicaciones donde más se utilizan estos vehículos se exponen a continuación:

- En el campo de la acuicultura, algunas de sus aplicaciones son la prevención e inspección de bioincrustaciones o incrustaciones biológicas (cúmulo indeseable de microorganismos, plantas, algas y/o animales sobre estructuras mojadas), el análisis de datos e imágenes en zonas estuáricas y explotaciones acuícolas, realización de controles visuales de especies, estudios sobre animales (tamaño, posición, etc), entre otras [11].
- Manipulación e inspección de instalaciones petroleras o de gas, presas, tuberías, barcos, cables, cimentaciones, etc. Las exigencias del trabajo realizado en infraestructuras submarinas son altas, ya que frecuentemente incluyen la inspección de grietas y defectos, y la posterior intervención para realizar tareas como mantenimiento, ensamblaje, limpieza, vigilancia, apertura y cierre de válvulas, entre otras. La tendencia en el uso de ROVs se ve incrementada a medida que las infraestructuras cerca de la costa se trasladan hacia aguas más profundas.
- En cuanto a la investigación científica, los ROVs se emplean para realizar tareas de estudio de la fauna y flora submarina, recolección de muestras para estudios (arqueológicos, geológicos o ecológicos) y análisis físico-químico del agua (composición, pH, presión, temperatura, etc).
- Salvamento, investigación policial, recuperación de cuerpos, misiones de rescate.
- Construcción submarina, excavaciones y enterrado de zanjas, tendido de tubos, cimentaciones, extensión de cables, perforaciones, etc.

## 2.2. Sibiu PRO

El ROV que ha sido objeto de estudio es el Sibiu Pro (mostrado en la Figura 1), comercializado por *Nido Robotics*. Este ROV fue adquirido por el grupo de Electrónica y Robótica Inteligentes TIC-266 de la Universidad de Huelva en el marco del proyecto internacional KTTSeaDrones.



Fig. 1. Sibiu PRO [6]

A continuación, se describen las características generales del Sibiu PRO, así como otros aspectos relevantes para este proyecto.

### 2.2.1. Características generales

El Sibiu PRO incorpora ocho propulsores, lo que le concede suavidad y estabilidad en la navegación, así como una cámara de 1080p, específicamente optimizada para el entorno submarino, y cuatro luces regulables de 1500 lúmenes. Utiliza baterías recargables de litio (LiFePO<sub>4</sub>) como fuente de energía para su funcionamiento bajo el agua.

Sus especificaciones técnicas se muestran a continuación [6]:

Masa	16 kg
Tamaño	52 x 39 x 29 cm
Profundidad máxima	300 m
Velocidad máxima	3 nudos
Cámara	1080p
Luces	4x1500 lúmenes

Tabla 1. Especificaciones técnicas del Sibiu PRO

Además del vehículo, el sistema requiere una estación de control en tierra (GCS, Ground Control Station). Ambos elementos se conectan mediante un cable umbilical, a través de una caja de superficie, tal como muestra la Figura 2.



**Fig. 2.** Elementos del sistema

La estación de control es operada y supervisada por un piloto humano en la superficie del agua. Se encarga de ejecutar el programa que nos permite comunicarnos con el ROV, ya sea *QGroundControl* o un script en *Matlab*. Además, en el PC se recibe y muestra la transmisión de vídeo, la información de telemetría y otros datos sobre el estado general del vehículo. También permite que el operador controle el vehículo desde un joystick.

El cable umbilical es responsable de transmitir las señales de control desde la estación de tierra al vehículo, además de enviar la señal de vídeo y los datos de telemetría a la estación de tierra.

### 2.2.2. Grados de libertad y diseño estructural

El nivel de control del movimiento de un ROV depende del número de grados de libertad (DoF) que se pueden alcanzar dada la configuración de sus propulsores. Los grados de libertad son el conjunto de desplazamientos y rotaciones que pueden especificar completamente la posición y orientación del vehículo. Cualquier cuerpo que puede moverse libremente en el espacio tridimensional tiene un máximo de 6 DoFs, tres elementos de traslación y tres de rotación.

En un vehículo submarino [2], estos 6 DoF se traducen en un conjunto de tres movimientos independientes a lo largo de tres direcciones X, Y, Z definidos como:

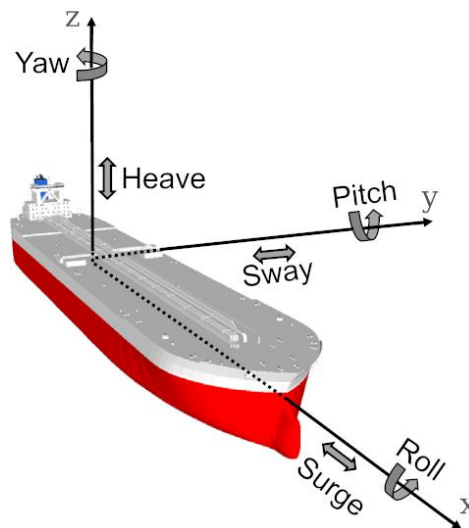
- Avance o *surge* (movimiento longitudinal)
- Ronza o *sway* (movimiento lateral)
- Ascenso o *heave* (movimiento vertical)

Así como tres rotaciones alrededor de los ejes X, Y, Z definidos como:

- Balanceo o *roll* (rotación sobre el eje X)
- Inclinación o *pitch* (rotación sobre el eje Y)
- Guiñada o *yaw* (rotación sobre el eje Z). Para referirse a este ángulo frecuentemente se utiliza el término *heading*.

Estos tres ángulos se conocen como ángulos de Euler. También se podría describir la orientación del vehículo mediante cuaterniones, aunque los ángulos de Euler son más intuitivos y fáciles de visualizar.

Estos 6 grados de libertad se ilustran en la Figura 3.



**Fig. 3.** Movimientos y rotaciones independientes de un vehículo submarino

Aunque es posible usar un modelo de 6 DoF para diseñar sistemas de control, en muchos casos, las embarcaciones marinas no utilizan todos los grados de libertad posibles durante su navegación, por lo que los modelos de orden reducido se usan comúnmente en los sistemas de control de movimiento.

Por ejemplo, los modelos de 3 DoF solo consideran los movimientos en el plano horizontal (*roll*, *pitch* y *yaw*). En este caso, también es común suponer que los dos movimientos de traslación pueden desacoplarse en un movimiento longitudinal y otro movimiento lateral.

En cualquier caso, el número de grados de libertad permitidos está definido por la configuración estructural del vehículo objeto de estudio. En el Sibiu PRO, esta

configuración está formada por ocho propulsores del tipo T200, dispuestos tal como se muestra en la Figura 4.

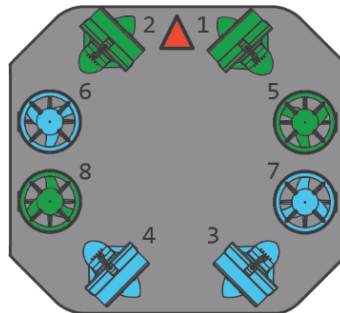


Fig. 4. Configuración de los propulsores del Sibiu PRO

Esta configuración se muestra desde la vista superior. Los propulsores verdes giran en sentido antihorario y los propulsores azules en sentido horario. Además, los propulsores 1, 2, 3 y 4 desplazan el ROV en el plano horizontal, mientras que los propulsores 5, 6, 7 y 8 lo desplazan verticalmente. De esta forma, la configuración proporciona la capacidad de control de 6 DoF.

### 2.3. Sonar Ping360

El Sibiu PRO incluye un conjunto de sensores integrados en su plataforma, cuya información es vital para una correcta navegación: unidad de medida inercial (IMU), sensor de presión, varios sonares acústicos, etc. Entre ellos, cabe destacar el Ping360, mostrado en la Figura 5. Este es un sensor SONAR (SOUND Navigation And Ranging) que permite visualizar los elementos que se encuentran alrededor del ROV.



Fig. 5. Sonar Ping360

Los sensores sonar son un tipo de sonar activo que funciona transmitiendo pulsos de sonido al agua y registrando los ecos que se devuelven a medida que cada pulso de sonido se refleja en los objetos frente a él.

Cuando se trata de explorar instalaciones en entornos naturales o al aire libre, es normal que el agua presente un alto grado de turbidez, lo que perjudica la visión submarina e impide, por tanto, que las cámaras, que tradicionalmente portan los ROVs, generen información útil.

Por este motivo, la detección de elementos de interés en estas circunstancias requiere el uso de sensores de proximidad, tales como sonares, que proporcionen al operador información sobre el entorno que rodea al ROV.

Es por esto por lo que la propuesta presentada en este trabajo se basa en la información proporcionada por el sonar Ping360, que aporta información del entorno que rodea al vehículo, posicionando distintos tipos de elementos que se encuentren a su alrededor.

Los sonares de barrido, como lo es el Ping360, se definen por tener un haz acústico en forma de “abanico” con un haz vertical ancho y un haz horizontal estrecho [15], que le permite obtener secciones transversales acústicas del entorno, tal como se muestra en la Figura 6.

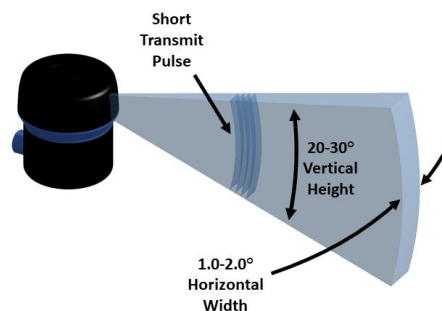


Fig. 6. Haz acústico del sonar

Este transductor está montado sobre un motor que lo gira en incrementos de 0.9 grados y, mientras lo hace, genera una imagen circular de los alrededores de la sonda con un alcance máximo de 50 metros (ver Figuras 7 y 8).

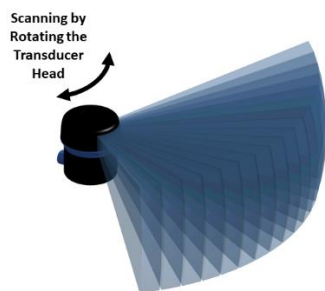


Fig. 7. Muestreo del sonar

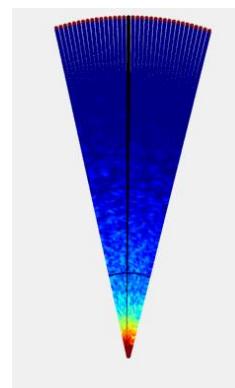


Fig. 8. Imagen obtenida con el sonar

Al combinar la velocidad conocida del sonido en el agua con el tiempo en el que se recibieron los ecos, el sonar puede calcular la distancia que ha viajado el sonido. La ecuación para determinar la distancia acústicamente es la siguiente:

$$distancia = \frac{1}{2} \cdot velocidad\ del\ sonido\ en\ el\ agua \cdot tiempo \quad (1)$$

Por lo general, la velocidad del sonido en agua salada es de aproximadamente 1500 m/s, pero esto puede variar según la temperatura del agua, la salinidad y la profundidad de funcionamiento del sonar.

Los objetivos materiales con densidades muy diferentes a las del agua (como gas, roca, hormigón o metal) serán muy reflectantes y tendrán fuertes ecos. Los ecos de materiales como barro, limo, arena y plantas serán más débiles, ya que tienen una densidad similar a la del agua o absorben energía acústica.

El sonar Ping360 obtiene 1200 valores de la intensidad del eco en cada ángulo. Cada uno de ellos hace referencia a una distancia concreta y consiste en un valor entre 0 y 255 (sin rebote e intensidad máxima, respectivamente). Estos se muestran en el interfaz de visualización a través de una paleta de colores, teniendo un color diferente dependiendo de la intensidad del rebote.

En determinadas circunstancias donde las condiciones de visibilidad sean mínimas, la información del sonar acústico será imprescindible, puesto que la función de la cámara será mínima.

# Capítulo 3

## Modelo dinámico

---

Aunque el Sibiu PRO tiene un sistema de control implementado en su plataforma, utiliza un controlador en bucle abierto, que proporciona capacidades de control para la operación manual, donde no se requiere un alto nivel de exactitud. Sin embargo, para operaciones automáticas, se necesita un sistema de control robusto, debido a los requisitos de precisión y seguridad del vehículo.

Para estudiar el comportamiento del ROV y lograr su control autónomo resulta de gran utilidad el empleo de modelos matemáticos que reflejen de la forma más fidedigna posible la dinámica real del sistema. La exactitud de este modelo matemático basado en las propiedades hidrodinámicas del vehículo es crucial para llevar a cabo simulaciones del ROV y predecir su comportamiento.

La dinámica de un vehículo submarino ha sido ampliamente estudiada desde los inicios de la ingeniería naval, siendo frecuentemente difícil de definir con exactitud debido a las no linealidades presentes en el sistema y a las incertidumbres inherentes al proceso de modelado.

En este capítulo, se presenta un modelo dinámico del Sibiu PRO. Las teorías fundamentales aplicadas en este TFG para modelar el vehículo han sido descritas ampliamente por Fossen [2], que demuestra los modelos matemáticos para todo tipo de embarcaciones marinas con 6 DoF, mediante ecuaciones diferenciales en forma matricial.

### 3.1. Notación

El movimiento de un ROV con 6 grados de libertad se puede representar en forma vectorial utilizando la notación SNAME (Society of Naval Architects and Marine Engineers) [4], tal como se muestra en la Tabla 2, donde se establece la posición, velocidad, y fuerza o momento correspondiente a cada grado de libertad del vehículo (ver Figura 3).

DoF	Tipo de movimiento	Fuerza	Velocidad lineal	Posición
1	Traslación en el eje X (surge)	$X$	$u$	$x$
2	Traslación en el eje Y (sway)	$Y$	$v$	$y$
3	Traslación en el eje Z (heave)	$Z$	$w$	$z$
DoF	Tipo de movimiento	Momento	Velocidad angular	Ángulo de Euler
4	Rotación sobre el eje X (roll)	$K$	$p$	$\phi$
5	Rotación sobre el eje Y (pitch)	$M$	$q$	$\theta$
6	Rotación sobre el eje Z (yaw)	$N$	$r$	$\psi$

Tabla 2. Notación SNAME para vehículos acuáticos

donde:

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^3 \text{ es la posición lineal;}$$

$$v = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \in \mathbb{R}^3 \text{ es la velocidad lineal;}$$

$$\theta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \in (SO)3 \text{ es la posición angular, también denominada actitud;}$$

$$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \in \mathbb{R}^3 \text{ es la velocidad angular;}$$

$$f = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \in \mathbb{R}^3 \text{ es la fuerza que actúa sobre el ROV;}$$

$$m = \begin{bmatrix} K \\ M \\ N \end{bmatrix} \in \mathbb{R}^3 \text{ es el momento que actúa sobre el ROV;}$$

siendo  $\mathbb{R}^3$  el espacio Euclídeo tridimensional y  $SO(3)$  la esfera tridimensional en la que se definen los tres ángulos de Euler.

De acuerdo con la notación SNAME, las coordenadas de posición, velocidad y fuerza generalizadas se pueden representar mediante los vectores  $\eta$ ,  $v$  y  $\tau$ , respectivamente:

$$\eta = [x \ y \ z \ \phi \ \theta \ \psi]^T \in \mathbb{R}^3 \times (SO)3 \quad (2)$$

$$v = [u \ v \ w \ p \ q \ r]^T \in \mathbb{R}^6 \quad (3)$$

$$\tau = [X \ Y \ Z \ K \ M \ N]^T \in \mathbb{R}^6 \quad (4)$$

### 3.2. Sistemas de referencia

Al modelar un cuerpo en el espacio tridimensional, se necesitan al menos los dos sistemas de referencia siguientes para describir su movimiento:

#### ○ NED

Es el sistema de coordenadas *North-East-Down* formado por los ejes  $\{n\} = (x_n, y_n, z_n)$  y origen en  $o_n$ , tal como ilustra la Figura 9. Este es el sistema de coordenadas que empleamos en nuestra vida cotidiana. Suele definirse como el plano tangente a la superficie de la Tierra que se mueve con el vehículo. En este sistema, el eje  $x_n$  apunta hacia el norte, el eje  $y_n$  apunta hacia el este, y el eje  $z_n$  es normal a la superficie de la Tierra y apunta hacia su centro.

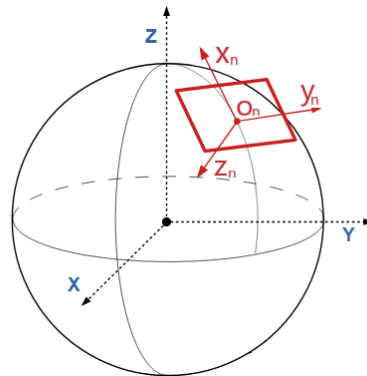


Fig. 9. Sistema de referencia NED

#### ○ BODY

Es el sistema de coordenadas de la estructura física del vehículo formado por los ejes  $\{b\} = (x_b, y_b, z_b)$  y origen en  $o_b$ , tal como ilustra la Figura 10. Es un sistema móvil fijado al vehículo. El origen  $o_b$  generalmente se define en el centro geométrico del vehículo, para aprovechar las propiedades de simetría. Los ejes  $x_b, y_b$  y  $z_b$  se definen de tal manera que coincidan con los ejes principales de inercia:  $x_b$  apunta hacia la dirección de avance del ROV,  $y_b$  apunta hacia la dirección transversal del ROV y  $z_b$  apunta verticalmente hacia abajo desde el ROV.

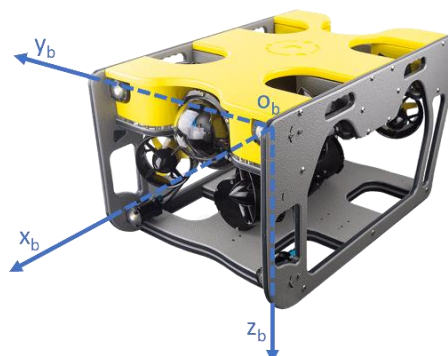


Fig. 10. Sistema de referencia BODY

### 3.3. Transformaciones entre los sistemas de referencia

#### 3.3.1. Transformación de la velocidad lineal

Un vector definido en un sistema de referencia se puede transformar en otro utilizando una matriz de rotación. En este caso, la velocidad lineal del ROV viene dada por:

$$v = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (5)$$

Este vector está definido en el sistema de referencia BODY. Para obtener la velocidad lineal expresada en el sistema de coordenadas NED, utilizamos la matriz de rotación  $R_b^n$ , tal como se muestra a continuación:

$$v^n = R_b^n \cdot v^b \quad (6)$$

donde  $v^n$  es el vector de velocidad lineal expresado en el sistema de referencia NED y  $v^b$  es el mismo vector expresado en el sistema BODY.

La matriz de rotación  $R_b^n$  depende de los ángulos de rotación sobre los ejes  $x_b, y_b, z_b$  (ángulos de Euler), definidos en la Tabla 2:

$$\theta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \quad (7)$$

Por lo tanto, la matriz de rotación  $R_b^n(\theta)$  se puede calcular de la siguiente forma:

$$R_b^n(\theta) = R_z(\psi) \cdot R_y(\theta) \cdot R_x(\phi) \quad (8)$$

donde:

$$R_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \quad (10)$$

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \quad (11)$$

Siendo  $R_z$  la matriz de rotación sobre el eje  $z_b$ ,  $R_y$  la matriz de rotación sobre el eje  $y_b$  y  $R_x$  la matriz de rotación sobre el eje  $x_b$ .

Cabe destacar que el orden en el que se realizan estas rotaciones no es arbitrario. En aplicaciones de guiado, navegación y control (GNC), al definir las rotaciones es común usar la convención  $zyx$  para realizar la transformación de  $\{n\}$  a  $\{b\}$  especificada en términos de los ángulos de Euler  $\phi$ ,  $\theta$  y  $\psi$ .

Por lo tanto, la matriz de rotación queda de la siguiente forma:

$$R_b^n(\theta) = \begin{bmatrix} \cos\psi \cdot \cos\theta & -\sin\psi \cdot \cos\phi + \cos\psi \cdot \sin\theta \cdot \sin\phi & \sin\psi \cdot \sin\phi + \cos\psi \cdot \sin\theta \cdot \cos\phi \\ \sin\psi \cdot \cos\theta & \cos\psi \cdot \cos\phi + \sin\psi \cdot \sin\theta \cdot \sin\phi & -\cos\psi \cdot \sin\phi + \sin\psi \cdot \sin\theta \cdot \cos\phi \\ -\sin\theta & \cos\theta \cdot \sin\phi & \cos\theta \cdot \cos\phi \end{bmatrix} \quad (12)$$

Esta matriz de rotación se denomina matriz de rotación de ángulos de Euler, y se puede comprobar que  $R \cdot R^T = R^T \cdot R = I$ , y además  $\det(R) = 1$ , lo que implica que  $R$  es ortogonal.

### 3.3.2. Transformación de la velocidad angular

De la misma forma, la velocidad angular del ROV viene dada por:

$$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (13)$$

Este vector está definido en el sistema de referencia BODY. Para obtener la velocidad angular expresada en el sistema de coordenadas NED, utilizamos la matriz de rotación  $T_\theta$ , tal como se muestra a continuación:

$$\dot{\theta} = T_\theta(\theta) \cdot \omega^b \quad (14)$$

donde  $\dot{\theta}$  es el vector de velocidad angular expresado en el sistema de referencia NED y  $\omega^b$  es el mismo vector expresado en el sistema BODY, siendo  $\dot{\theta} = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$ .

La matriz de rotación  $T_\theta(\theta)$  se calcula de la siguiente forma:

$$T_\theta(\theta) = \begin{bmatrix} 1 & \sin\phi \cdot \tan\theta & \cos\phi \cdot \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \quad (15)$$

Como vemos,  $T_\theta(\theta)$  no está definida para un ángulo  $\theta = \pm 90^\circ$  ( $\tan 90^\circ = \pm\infty$ ). En general, para los vehículos submarinos esto no es un problema, ya que no suelen operar cerca de esta singularidad.

Para solucionar la discontinuidad de los ángulos de Euler, puede emplearse un método de cuatro parámetros basado en cuaterniones [4]. Un cuaternión  $q$  se define como un número complejo formado por cuatro unidades:

$$q = [q_0 \ \varepsilon_1 \ \varepsilon_2 \ \varepsilon_3]^T \quad (16)$$

donde  $q_0$  es un parámetro real y las otras tres unidades son parámetros imaginarios.

### 3.3.3. Matriz de rotación global

A partir de las dos matrices de rotación definidas anteriormente, podemos determinar la matriz de rotación que nos permite pasar del sistema de referencia BODY al sistema de referencia NED, tanto las velocidades lineales como las velocidades angulares:

$$J_\theta(\eta) = \begin{bmatrix} R_b^n(\theta) & 0_{3 \times 3} \\ 0_{3 \times 3} & T_\theta(\theta) \end{bmatrix} \quad (17)$$

### 3.4. Ecuaciones del modelo

Las ecuaciones de movimiento que componen el modelo dinámico de un vehículo submarino pueden ser obtenidas a partir de las ecuaciones de Newton-Euler o Lagrange.

La segunda ley de Newton establece que la suma de todas las fuerzas externas que actúan sobre un cuerpo es igual al producto de la masa del cuerpo por su aceleración:

$$\sum F = m \cdot \frac{d^2x}{dt^2} \quad (18)$$

Las fuerzas que actúan sobre el ROV son las fuerzas generadas por los propulsores y varias fuerzas externas. La mayoría de las veces no es posible conocer todas las fuerzas ajenas al ROV, ya que existen numerosas perturbaciones no medibles que actúan sobre el sistema, siendo las fuerzas externas principales las corrientes submarinas y las perturbaciones del cable tether.

Las corrientes submarinas podrían modelarse como fuerzas verticales y horizontales que actúan sobre el ROV en el sistema de referencia NED y, aplicando la matriz de rotación, podrían moverse al sistema de coordenadas BODY e implementarse como ruido blanco gaussiano. Sin embargo, el alcance de este TFG no incluirá el modelado de corrientes submarinas y del cable del vehículo.

El modelo presentado por Fossen en 2002 [7] ha sido adoptado por la comunidad internacional como un modelo estándar para el diseño de sistemas de control de vehículos marinos. Este modelo contiene la ecuación cinemática (Ec. 19) y la ecuación cinética (Ec. 20), como se muestra a continuación [2]:

$$\dot{\eta} = J(\eta) \cdot v \quad (19)$$

$$M \cdot \dot{v} + C(v) \cdot v + D(v) \cdot v + g(\eta) = \tau \quad (20)$$

donde:

- $M$  es la matriz de inercia del sistema;
- $C$  es la matriz de aceleración de Coriolis y centrípeta;
- $D$  es la matriz de rozamiento hidrodinámico o resistencia viscosa (*damping*);
- $g$  es el vector que agrupa las fuerzas de flotación y gravitacional;
- $\tau$  es el vector de fuerzas y momentos que actúan sobre el ROV;
- $v$  es el vector de velocidad del ROV representado en  $\{b\}$ .

La ecuación cinemática (Ec. 19) describe la representación del movimiento en diferentes sistemas de referencia. Es decir, dado que todas las matrices y vectores definidos anteriormente basados en la mecánica newtoniana están representados en el sistema de referencia BODY, esta ecuación los transforma al sistema de referencia NED.

Por lo tanto, esta ecuación también puede expresarse de la siguiente forma:

$$\dot{\eta} = J(\eta) \cdot v \Leftrightarrow \begin{bmatrix} \dot{p} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} R_b^n(\theta) & 0_{3 \times 3} \\ 0_{3 \times 3} & T_\theta(\theta) \end{bmatrix} \begin{bmatrix} v^b \\ v^n \end{bmatrix} \quad (21)$$

La ecuación cinética (Ec. 20), por otra parte, analiza las fuerzas y los momentos que provocan el movimiento del ROV. Este modelo asume ciertas simplificaciones:

- El vehículo es un cuerpo rígido de masa constante, con empuje y par constantes.
- Las corrientes submarinas son despreciables.
- El vehículo no experimenta el paso por su propia estela.

- La masa añadida del sistema es despreciable. Este término representa una masa virtual añadida al sistema debido a que el cuerpo que acelera debe mover un determinado volumen del fluido circundante a medida que se mueve a través de él [13].

Las ecuaciones del modelo están escritas en forma vectorial, ya que esta representación facilita el diseño de controladores de múltiples entradas y múltiples salidas (MIMO), puesto que las propiedades y la reducción del modelo se derivan de las propiedades básicas de las matrices. Además, cabe señalar que los modelos vectoriales presentan mayores ventajas desde el punto de vista computacional y para realizar manipulaciones algebraicas. La legibilidad también ha mejorado significativamente debido a su notación más compacta.

Las matrices  $M$ ,  $C(v)$ ,  $D(v)$  y el vector  $g(\eta)$  contienen más de 300 parámetros desconocidos, por lo que la estimación de todos ellos resulta inviable. Sin embargo, según las características y las velocidades de operación del vehículo, se pueden hacer varias suposiciones para simplificar el modelo dinámico y reducir la cantidad de parámetros desconocidos en el modelo, tal como se muestra a continuación.

### 3.4.1. Matriz de inercia

La matriz de inercia del sistema viene dada por:

$$M = \begin{bmatrix} m & 0 & 0 & 0 & m \cdot z_g & -m \cdot y_g \\ 0 & m & 0 & -m \cdot z_g & 0 & m \cdot x_g \\ 0 & 0 & m & m \cdot y_g & -m \cdot x_g & 0 \\ 0 & -m \cdot z_g & m \cdot y_g & I_x & I_{xy} & I_{xz} \\ m \cdot z_g & 0 & -m \cdot x_g & I_{yx} & I_y & I_{yz} \\ -m \cdot y_g & m \cdot x_g & 0 & I_{zx} & I_{zy} & I_z \end{bmatrix} \quad (22)$$

donde:

- $m$  es la masa del vehículo;
- $I_x$ ,  $I_y$ ,  $I_z$  son los momentos de inercia sobre los ejes  $x_b$ ,  $y_b$ ,  $z_b$  de  $\{b\}$ , respectivamente;
- $I_{xy} = I_{yx}$ ,  $I_{xz} = I_{zx}$ ,  $I_{yz} = I_{zy}$  son los productos de inercia;
- $r_g = [x_g, y_g, z_g]^T$  es el centro de gravedad (CG) del vehículo, en relación con el centro del vehículo ( $o_b$ );

Además, el centro de flotabilidad (CB) del vehículo se define como:

$$r_{CB} = [x_{CB}, y_{CB}, z_{CB}]^T \quad (23)$$

Puesto que el centro de la estructura del vehículo ( $o_b$ ) se coloca en el centro de flotabilidad,  $r_{CB}$  se convierte en:

$$r_{CB} = [0, 0, 0]^T \quad (24)$$

Por otro lado, dado que el vehículo tiene simetría en el plano xz y en el plano xy, la posición del centro de gravedad (CG) del vehículo viene dada por:

$$r_g = [x_g, y_g, z_g]^T = [0, 0, z_g]^T \quad (25)$$

Por lo tanto,  $I_{xy} = I_{xz} = I_{yz} = 0$ .

De esta forma, la matriz de inercia  $M$  se simplifica de la siguiente forma:

$$M = \begin{bmatrix} m & 0 & 0 & 0 & m \cdot z_g & 0 \\ 0 & m & 0 & -m \cdot z_g & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & -m \cdot z_g & 0 & I_x & 0 & 0 \\ m \cdot z_g & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & 0 & 0 & I_z \end{bmatrix} \quad (26)$$

### 3.4.2. Matriz de Coriolis

La matriz de Coriolis representa el momento generado por  $M$  debido a la rotación del sistema de referencia BODY sobre sistema de referencia NED, y viene dada por la siguiente expresión:

$$C = \begin{bmatrix} 0 & 0 & 0 & m(y_g q + z_g r) & -m(x_g p - w) & -m(x_g r + v) \\ 0 & 0 & 0 & -m(y_g p + w) & m(z_g r + x_g p) & -m(y_g r - u) \\ 0 & 0 & 0 & -m(z_g p - v) & -m(z_g q + u) & m(x_g p + y_g q) \\ -m(y_g q + z_g r) & m(y_g p + w) & m(z_g p - v) & 0 & -I_{yz}q - I_{xz}p + I_z r & I_{yz}r + I_{xy}p - I_y q \\ m(x_g q - w) & -m(z_g r + x_g p) & m(z_g q + u) & I_{yz}q + I_{xz}p - I_z r & 0 & -I_{xz}r - I_{xy}q + I_x p \\ m(x_g r + v) & m(y_g r - u) & -m(x_g p + y_g q) & -I_{yz}r - I_{xy}p + I_y q & I_{xz}r + I_{xy}q - I_x p & 0 \end{bmatrix} \quad (27)$$

Simplificando esta matriz:

$$C = \begin{bmatrix} 0 & 0 & 0 & m z_g r & m w & -m v \\ 0 & 0 & 0 & -m w & m z_g r & m u \\ 0 & 0 & 0 & -m(z_g p - v) & -m(z_g q + u) & 0 \\ -m z_g r & m w & m(z_g p - v) & 0 & I_z r & -I_y q \\ -m w & -m z_g r & m(z_g q + u) & -I_z r & 0 & I_x p \\ m v & -m u & 0 & I_y q & -I_x p & 0 \end{bmatrix} \quad (28)$$

### 3.4.3. Matriz de resistencia hidrodinámica

El arrastre o rozamiento hidrodinámico de un vehículo acuático se debe principalmente a los siguientes factores [2]:

- Rozamiento potencial: este tipo de rozamiento aparece cuando un cuerpo es forzado a oscilar con la frecuencia de excitación de la ola en ausencia de ondas incidentes.
- Rozamiento en la superficie del vehículo: se debe a la teoría de la capa límite laminar, y adquiere relevancia cuando se considera el movimiento de baja frecuencia del vehículo.
- Rozamiento debido a las olas: este tipo de rozamiento se puede interpretar como una resistencia adicional que el vehículo debe vencer para avanzar entre las olas.
- Rozamiento debido al desprendimiento de vórtices: la paradoja de D'Alambert establece que ninguna fuerza hidrodinámica actúa sobre un sólido que se mueve con velocidad constante completamente sumergido en un fluido no viscoso. En un fluido viscoso, las fuerzas de fricción están presentes de tal manera que el sistema no conserva su energía. Esto se debe al desprendimiento de vórtices en bordes afilados o salientes.

Sin embargo, los efectos debidos al rozamiento potencial y al rozamiento debido a las olas pueden despreciarse para los vehículos submarinos, ya que suelen operar por debajo de la zona afectada por las olas.

En definitiva, la matriz de rozamiento hidrodinámico del ROV,  $D(v)$ , se puede aproximar con un término de rozamiento lineal  $D_L$  causado por la fricción superficial y un término de rozamiento cuadrático  $D_{NL}(v)$  debido principalmente al desprendimiento de vórtices, tal como se muestra a continuación:

$$D(v) = D_L + D_{NL}(v) \quad (29)$$

Para la mayoría de las aplicaciones prácticas, los términos no diagonales de  $D_L$  y  $D_{NL}(v)$  son pequeños en comparación con los de la diagonal, por lo que pueden despreciarse. Por lo tanto, las matrices de rozamiento lineal y cuadrático vienen dadas por:

$$D_L = -diag(X_u, Y_v, Z_w, K_p, M_q, N_r) \quad (30)$$

$$D_{NL} = -diag(X_{u|u}|u|, Y_{v|v}|v|, Z_{w|w}|w|, K_{p|p}|p|, M_{q|q}|q|, N_{r|r}|r|) \quad (31)$$

Por lo tanto, la matriz de rozamiento hidrodinámico se calcula se la siguiente forma:

$$D = - \begin{bmatrix} X_u + X_{u|u}|u| & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_v + Y_{v|v}|v| & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_w + Z_{w|w}|w| & 0 & 0 & 0 \\ 0 & 0 & 0 & K_p + K_{p|p}|p| & 0 & 0 \\ 0 & 0 & 0 & 0 & M_q + M_{q|q}|q| & 0 \\ 0 & 0 & 0 & 0 & 0 & N_r + N_{r|r}|r| \end{bmatrix} \quad (32)$$

donde:

- $X_u, Y_v, Z_w$  son los coeficientes de rozamiento lineales de traslación en el agua;
- $X_{u|u}, Y_{v|v}, Z_{w|w}$  son los coeficientes de rozamiento cuadráticos de traslación en el agua;
- $K_p, M_q, N_r$  son los coeficientes de rozamiento lineales de rotación en el agua;
- $K_{p|p}, M_{q|q}, N_{r|r}$  son los coeficientes de rozamiento cuadráticos de rotación en el agua.

#### 3.4.4. Vector de fuerzas de flotación y gravitacional

En hidrostática, las fuerzas que actúan sobre el ROV debido a la fuerza gravitacional y a la fuerza de flotación se denominan fuerzas de sustentación o fuerzas de elevación.

Dado que  $m$  es la masa del vehículo,  $g$  es la aceleración de la gravedad,  $\rho$  es la densidad del agua donde está sumergido el vehículo y  $\nabla$  es el volumen de fluido desplazado por el ROV, el peso del vehículo  $W$  y la fuerza de flotación  $B$  vienen dados por las siguientes expresiones:

$$W = m \cdot g \quad (33)$$

$$B = \rho \cdot \nabla \cdot g \quad (34)$$

Como se mencionó anteriormente, el centro de flotabilidad (CB) del vehículo es  $r_b = [x_b, y_b, z_b]^T = [0, 0, 0]^T$ , al estar situado en el origen del sistema de referencia BODY. Por otro lado, la posición del centro de gravedad (CG) del vehículo viene dada por  $r_g = [x_g, y_g, z_g]^T = [0, 0, z_g]^T$ . Además, el ROV está diseñado para ser neutralmente flotante ( $W = B$ ) en agua salada.

Por lo tanto, el vector de fuerza restauradora  $g(\eta)$  se puede calcular en función de los ángulos de Euler de la siguiente forma:

$$g(\eta) = \begin{bmatrix} (W - B)\sin\theta \\ -(W - B) \cdot \cos\theta \cdot \sin\phi \\ -(W - B) \cdot \cos\theta \cdot \cos\phi \\ z_g W \cdot \cos\theta \cdot \sin\phi \\ z_g W \cdot \sin\theta \\ 0 \end{bmatrix} \quad (35)$$

Como vemos, el vector de fuerza restauradora solo afecta al eje vertical (eje Z de la Figura 3).

## Capítulo 4

### Modelo dinámico simplificado

---

Con el objetivo de crear una simulación del Sibiu PRO en Matlab, en este capítulo se lleva a cabo una simplificación del modelo presentado en el capítulo anterior (Ec. 20). Para ello, es necesario que se cumplan las siguientes condiciones:

- El vehículo se desplaza únicamente en el plano  $xy$ , por lo que no vamos a considerar el movimiento de ascensión o *heave*, reduciendo así los grados de libertad del ROV a 3 DoF.
- La velocidad del vehículo debe ser pequeña. De esta forma, las fuerzas de Coriolis y las fuerzas centrípetas se pueden despreciar.

Este modelo simplificado seguirá reflejando la dinámica principal del ROV. La ecuación cinética queda de la siguiente forma:

$$M \cdot \dot{v} + D(v) \cdot v = \tau \quad (36)$$

Esa ecuación considera los siguientes movimientos:

- Movimiento longitudinal (*surge*)
- Movimiento lateral (*sway*)
- Rotación sobre el eje Z (*yaw*)

A partir de la Ec. 36, podemos obtener la ecuación de estado del sistema:

$$M \cdot \dot{v} = -D(v) \cdot v + \tau \quad (37)$$

$$\dot{v} = -M^{-1} \cdot D(v) \cdot v + M^{-1} \cdot \tau \quad (38)$$

Como vemos, las entradas del modelo propuesto son las fuerzas que actúan sobre el sistema físico, es decir, las fuerzas o empujes generados por los motores de propulsión.

Además, al reducir los grados de libertad del vehículo, el número de ecuaciones necesarias para definir el movimiento del vehículo disminuye. Puesto que el orden del sistema generalmente viene dado por la siguiente expresión:

$$\text{Orden} = 2 \times \text{DoF} \quad (39)$$

obtendremos un modelo de 6 ecuaciones considerando 3 grados de libertad.

#### 4.1. Ecuaciones del modelo propuesto

La posición del ROV en el plano está definida por tres variables independientes, tal como se muestra en la Figura 11:

$$\vec{r} = (x, y, \theta) \quad (40)$$

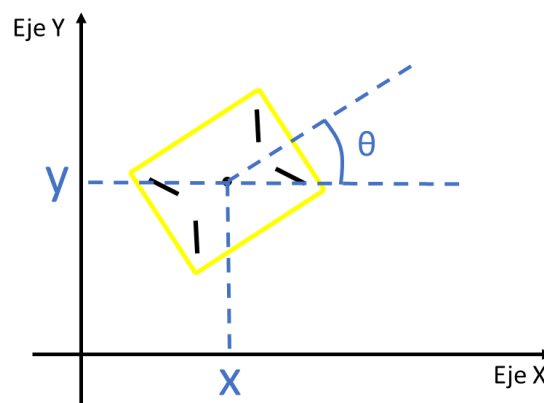


Fig. 11. Posición del ROV en el plano

Además, los motores del ROV generan dos fuerzas y un momento independientes, que provocan el desplazamiento del vehículo:

$$\vec{F} = (F_x, F_y, \tau) \quad (41)$$

A continuación, analizamos el movimiento correspondiente a cada grado de libertad del vehículo.

#### Movimiento lineal en el eje X

En primer lugar, definimos la posición, velocidad y aceleración (magnitudes lineales) en este eje:

- $x = x_1$ : posición;
- $v_x = \dot{x}_1 = x_2$ : velocidad;
- $a_x = \ddot{x}_1 = \dot{x}_2$ : aceleración.

A continuación, aplicamos la segunda ley de Newton:

$$\sum F = m \cdot \frac{d^2x}{dt^2} \quad (18)$$

En este caso, las fuerzas que actúan sobre el vehículo son las siguientes:

- Fuerza causada por los propulsores:  $F_x$
- Fuerza de arrastre o resistencia hidrodinámica. Es la fuerza de rozamiento que impide el movimiento del cuerpo a través del fluido con facilidad, que se genera por una depresión existente entre la parte trasera y la parte delantera del vehículo. Dicho de otro modo, la resistencia hidrodinámica se podría definir como la fuerza opuesta al movimiento que sufre un objeto sólido al desplazarse a través de un fluido, y generada por el mismo fluido que lo rodea. Esta fuerza se calcula mediante la siguiente expresión:

$$F_D = \frac{1}{2} \cdot C_D \cdot A_D \cdot \rho \cdot v^2 \quad (42)$$

donde:

- $C_D$  es el coeficiente de arrastre o coeficiente de resistencia hidrodinámica. Es un parámetro adimensional usado para cuantificar el arrastre de un objeto en un medio fluido. Este coeficiente es independiente del tamaño del cuerpo y de la velocidad del fluido hasta cierta medida.
- $A_D$  es el área de la sección transversal a la dirección de movimiento;
- $\rho$  es la densidad del fluido;
- $v$  es la velocidad del vehículo.

$C_D$  es función del número de Reynolds ( $Re$ ), que a su vez depende de la velocidad del vehículo en el fluido. En general, se calcula a partir de tablas empíricas. Agrupando todas las constantes de la expresión anterior, obtenemos el coeficiente hidrodinámico:

$$\mu = \frac{1}{2} \cdot C_D \cdot A_D \cdot \rho \quad (43)$$

Por lo tanto, la fuerza de rozamiento viscoso viene dada por:

$$F_D = \mu \cdot v^2 = \mu \cdot x_2^2 \quad (44)$$

donde  $\mu$  es un coeficiente hidrodinámico por determinar.

Finalmente, a partir de la segunda ley de Newton, obtenemos la siguiente expresión:

$$F_x - F_D = F_x - \mu \cdot x_2^2 = m \cdot \dot{x}_2 \quad (45)$$

Despejando  $\dot{x}_2$ :

$$\dot{x}_2 = \frac{F_x}{m} - \frac{\mu \cdot x_2^2}{m} \quad (46)$$

### Movimiento lineal en el eje Y

En primer lugar, definimos la posición, velocidad y aceleración (magnitudes lineales) en este eje:

- $y = x_3$ : posición
- $v_y = \dot{x}_3 = x_4$ : velocidad
- $a_y = \ddot{x}_3 = \dot{x}_4$ : aceleración

De nuevo, las fuerzas que actúan sobre el vehículo son las siguientes:

- Fuerza causada por los propulsores:  $F_y$
- Fuerza de rozamiento debida a la viscosidad del agua, que viene dada por:

$$F_D = \lambda \cdot v^2 = \lambda \cdot x_4^2 \quad (47)$$

donde  $\lambda$  es un coeficiente hidrodinámico por determinar.

Por lo tanto, a partir de la segunda ley de Newton, obtenemos la siguiente expresión:

$$F_y - F_D = F_y - \lambda \cdot x_4^2 = m \cdot \dot{x}_4 \quad (48)$$

Despejando  $\dot{x}_4$ :

$$\dot{x}_4 = \frac{F_y}{m} - \frac{\lambda \cdot x_4^2}{m} \quad (49)$$

### Rotación en el eje Z

En primer lugar, definimos la posición, velocidad y aceleración (magnitudes angulares) en este eje:

- $\theta = x_5$ : posición angular

- $\omega = \dot{x}_5 = x_6$ : velocidad angular
- $\alpha = \ddot{x}_5 = \dot{x}_6$ : aceleración angular

A continuación, aplicamos la segunda ley de Newton (expresada para movimientos rotacionales):

$$\sum M = I \cdot \frac{d^2\theta}{dt^2} \quad (50)$$

donde:

- M son los momentos o torques aplicados sobre el cuerpo;
- I es el momento de inercia del vehículo.

En este caso, los momentos que actúan sobre el vehículo son las siguientes:

- Momento causado por los propulsores:  $\tau$
- Momento de rozamiento viscoso, que viene dado por:

$$M_D = \xi \cdot \omega^2 = \xi \cdot x_6^2 \quad (51)$$

donde  $\xi$  es un coeficiente hidrodinámico por determinar.

Por lo tanto, a partir de la segunda ley de Newton, obtenemos la siguiente expresión:

$$\tau - M_D = \tau - \xi \cdot x_6^2 = I \cdot \dot{x}_6 \quad (52)$$

Despejando  $\dot{x}_6$ :

$$\dot{x}_6 = \frac{\tau}{I} - \frac{\xi \cdot x_6^2}{I} \quad (53)$$

Por lo tanto, las seis ecuaciones obtenidas son las siguientes:

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{F_x}{m} - \frac{\mu \cdot x_2^2}{m} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{F_y}{m} - \frac{\lambda \cdot x_4^2}{m} \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = \frac{\tau}{I} - \frac{\xi \cdot x_6^2}{I} \end{array} \right. \quad (54)$$

En forma matricial:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{\mu}{m}x_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{\lambda}{m}x_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -\frac{\xi}{I}x_6 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I} \end{pmatrix} \cdot \begin{pmatrix} F_x \\ F_y \\ \tau \end{pmatrix} \quad (55)$$

Como vemos, esta expresión tiene la misma forma que la Ec. 38:

$$\dot{v} = -M^{-1} \cdot D(v) \cdot v + M^{-1} \cdot \tau \quad (38)$$

De la misma forma, las entradas del modelo propuesto son las componentes de la fuerza generada por los motores de propulsión del vehículo. En el siguiente apartado, determinamos estas componentes.

## 4.2. Modelado de los propulsores

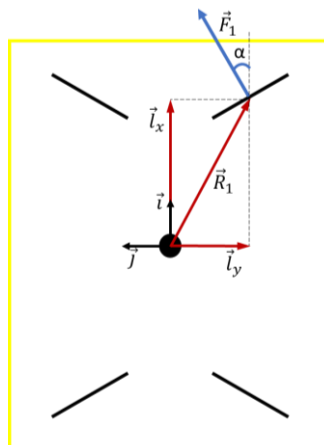
Como se vio en el apartado anterior, los motores del ROV generan dos fuerzas y un momento independientes, que provocan el desplazamiento del vehículo:

$$\vec{F} = (F_x, F_y, \tau) \quad (56)$$

A continuación, vamos a calcular el par aplicado por cada uno de los propulsores, y así poder calcular el par total,  $\tau$ .

### Primer propulsor

La fuerza o empuje aplicado por el primer propulsor se representa en la Figura 12:



**Fig. 12.** Fuerza aplicada por el primer propulsor

Por lo tanto, el par aplicado viene dado por:

$$\vec{\tau}_1 = \vec{R}_1 \wedge \vec{F}_1 \quad (57)$$

Desarrollando esta expresión:

$$\vec{\tau}_1 = (l_x \vec{i} - l_y \vec{j}) \wedge (F_1 \sin \alpha \vec{i} + F_1 \cos \alpha \vec{j}) \quad (58)$$

Observando la estructura del vehículo, vemos que  $\alpha = 45^\circ$ , por lo tanto:

$$\vec{\tau}_1 = \frac{\sqrt{2}}{2} F_1 (l_x \vec{i} - l_y \vec{j}) \wedge (\vec{i} + \vec{j}) \quad (59)$$

$$\vec{\tau}_1 = \frac{\sqrt{2}}{2} F_1 (l_x \vec{k} + l_y \vec{k}) \quad (60)$$

$$\vec{\tau}_1 = \frac{\sqrt{2}}{2} F_1 (l_x + l_y) \vec{k} \quad (61)$$

Siendo  $\vec{\tau}_1$  el par aplicado por el primer propulsor.

### Segundo propulsor

De acuerdo con la Figura 13, el par aplicado viene dado por:

$$\vec{\tau}_2 = \vec{R}_2 \wedge \vec{F}_2 \quad (62)$$

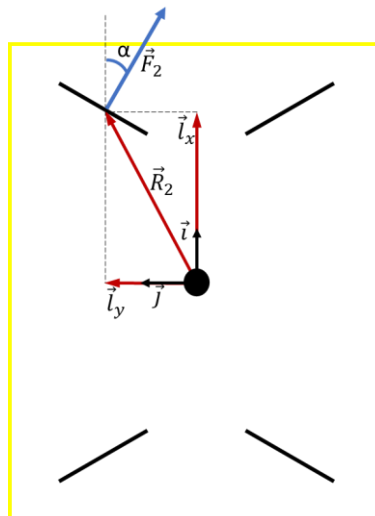


Fig. 13. Fuerza aplicada por el segundo propulsor

Desarrollando esta expresión:

$$\vec{\tau}_2 = (l_x \vec{i} + l_y \vec{j}) \wedge (F_2 \sin \alpha \vec{i} - F_2 \cos \alpha \vec{j}) \quad (63)$$

De nuevo, vemos que  $\alpha = 45^\circ$ , por lo tanto:

$$\vec{\tau}_2 = \frac{\sqrt{2}}{2} F_2 (l_x \vec{i} + l_y \vec{j}) \wedge (\vec{i} - \vec{j}) \quad (64)$$

$$\vec{\tau}_2 = \frac{\sqrt{2}}{2} F_2 (-l_x \vec{k} - l_y \vec{k}) \quad (65)$$

$$\vec{\tau}_2 = -\frac{\sqrt{2}}{2} F_2 (l_x + l_y) \vec{k} \quad (66)$$

Siendo  $\vec{\tau}_2$  el par aplicado por el segundo propulsor.

### Tercer propulsor

De acuerdo con la Figura 14, el par aplicado viene dado por:

$$\vec{\tau}_3 = \vec{R}_3 \wedge \vec{F}_3 \quad (67)$$

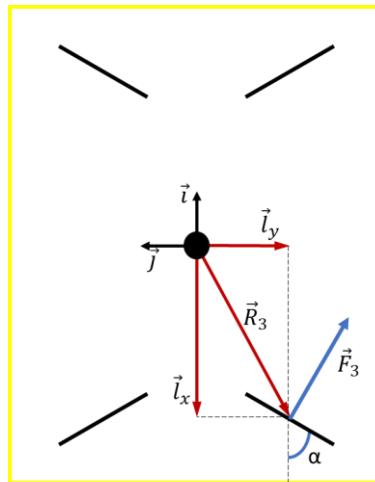


Fig. 14. Fuerza aplicada por el tercer propulsor

Desarrollando esta expresión:

$$\vec{\tau}_3 = (-l_x \vec{i} - l_y \vec{j}) \wedge (F_3 \sin \alpha \vec{i} - F_3 \cos \alpha \vec{j}) \quad (68)$$

De nuevo, vemos que  $\alpha = 45^\circ$ , por lo tanto:

$$\vec{\tau}_3 = \frac{\sqrt{2}}{2} F_3 (-l_x \vec{i} - l_y \vec{j}) \wedge (\vec{i} - \vec{j}) \quad (69)$$

$$\vec{\tau}_3 = \frac{\sqrt{2}}{2} F_3 (l_x \vec{k} + l_y \vec{k}) \quad (70)$$

$$\vec{\tau}_3 = \frac{\sqrt{2}}{2} F_3 (l_x + l_y) \vec{k} \quad (71)$$

Siendo  $\vec{\tau}_3$  el par aplicado por el tercer propulsor.

### Cuarto propulsor

De acuerdo con la Figura 15, el par aplicado viene dado por:

$$\vec{\tau}_4 = \vec{R}_4 \wedge \vec{F}_4 \quad (72)$$

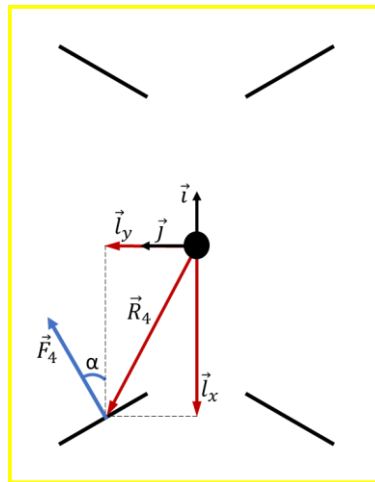


Fig. 15. Fuerza aplicada por el cuarto propulsor

Desarrollando esta expresión:

$$\vec{\tau}_4 = (-l_x \vec{i} + l_y \vec{j}) \wedge (F_4 \sin \alpha \vec{i} + F_4 \cos \alpha \vec{j}) \quad (73)$$

De nuevo, vemos que  $\alpha = 45^\circ$ , por lo tanto:

$$\vec{\tau}_4 = \frac{\sqrt{2}}{2} F_4 (-l_x \vec{i} + l_y \vec{j}) \wedge (\vec{i} + \vec{j}) \quad (74)$$

$$\vec{\tau}_4 = \frac{\sqrt{2}}{2} F_4 (-l_x \vec{k} - l_y \vec{k}) \quad (75)$$

$$\vec{\tau}_4 = -\frac{\sqrt{2}}{2} F_4 (l_x + l_y) \vec{k} \quad (76)$$

Siendo  $\vec{\tau}_4$  el par aplicado por el cuarto propulsor.

### Cálculo del par total y fuerza total

Para calcular el par total, sumamos el par producido por cada propulsor:

$$\vec{\tau}_T = \vec{\tau}_1 + \vec{\tau}_2 + \vec{\tau}_3 + \vec{\tau}_4 = \tau \vec{k} \quad (77)$$

Desarrollando esta expresión:

$$\vec{\tau}_T = \frac{\sqrt{2}}{2} [F_1 (l_x + l_y) \vec{k} - F_2 (l_x + l_y) \vec{k} + F_3 (l_x + l_y) \vec{k} - F_4 (l_x + l_y) \vec{k}] \quad (78)$$

$$\vec{\tau}_T = \frac{\sqrt{2}}{2} (l_x + l_y) (F_1 - F_2 + F_3 - F_4) \vec{k} \quad (79)$$

Por lo tanto:

$$\tau = \frac{\sqrt{2}}{2} (l_x + l_y) (F_1 - F_2 + F_3 - F_4) \quad (80)$$

De forma similar, calculamos las fuerzas  $F_x$  y  $F_y$  a partir de la fuerza total, sumando la fuerza generada por cada propulsor (ver Figura 16):

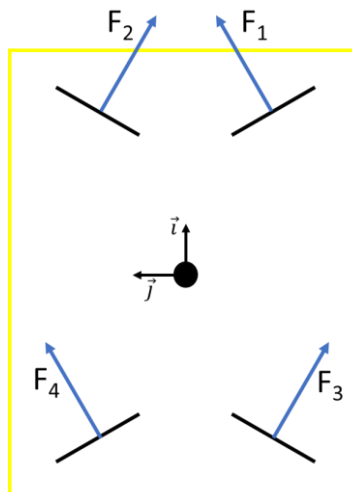


Fig. 16. Fuerza ejercida por cada propulsor

$$\vec{F}_T = \vec{F}_1 + \vec{F}_2 + \vec{F}_3 + \vec{F}_4 = F_x \vec{i} + F_y \vec{j} \quad (81)$$

$$\vec{F}_T = \frac{\sqrt{2}}{2} [F_1(\vec{i} + \vec{j}) + F_2(\vec{i} - \vec{j}) + F_3(\vec{i} - \vec{j}) + F_4(\vec{i} + \vec{j})] \quad (82)$$

$$\vec{F}_T = \frac{\sqrt{2}}{2} (F_1 + F_2 + F_3 + F_4) \vec{i} + \frac{\sqrt{2}}{2} (F_1 - F_2 - F_3 + F_4) \vec{j} \quad (83)$$

Por lo tanto:

$$F_x = \frac{\sqrt{2}}{2} (F_1 + F_2 + F_3 + F_4) \quad (84)$$

$$F_y = \frac{\sqrt{2}}{2} (F_1 - F_2 - F_3 + F_4) \quad (85)$$

### Determinación de las fuerzas de los propulsores

El modelo obtenido permite simular el comportamiento del vehículo para cualquier conjunto de valores que tomen las fuerzas de los propulsores. Sin embargo, para establecer un procedimiento que permita conducir el vehículo intuitivamente, se propone que estas fuerzas sean tales que provoquen que, por una parte, las componentes de la fuerza resultante  $F_x$  y  $F_y$  sean respectivamente proporcionales a dos velocidades de referencia, una la velocidad deseada de desplazamiento hacia adelante ( $v_a$ ), y otra la velocidad deseada de desplazamiento lateral ( $v_l$ ). Y, por otra parte, se propone que el par resultante sea proporcional a la velocidad de rotación deseada ( $\omega$ ). Con esta idea, la fuerza de cada propulsor se calculará a partir de dichas velocidades (ver Figura 17) según se propone en las ecuaciones (84)-(87).

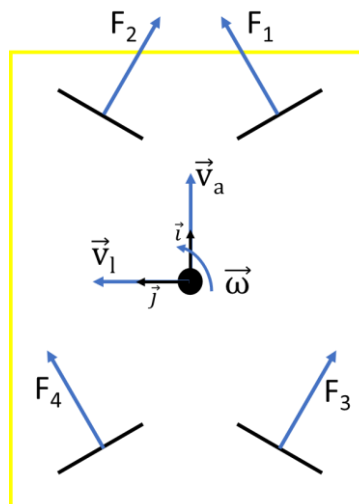


Fig. 17. Velocidades correspondientes a cada DoF:  $v_a$ ,  $v_l$ ,  $\omega$

$$F_1 = v_a + v_1 + \omega \quad (86)$$

$$F_2 = v_a - v_1 - \omega \quad (87)$$

$$F_3 = v_a - v_1 + \omega \quad (88)$$

$$F_4 = v_a + v_1 - \omega \quad (89)$$

A continuación, se comprobará que esta elección hace que la fuerza y par resultantes tomen valores que se adecúan al comportamiento inicialmente especificado. Efectivamente, calculando  $F_x$ ,  $F_y$  y  $\tau$  en función de  $v_a$ ,  $v_1$  y  $\omega$  obtenemos:

○  $\tau$

$$\tau = \frac{\sqrt{2}}{2} (l_x + l_y) (F_1 - F_2 + F_3 - F_4) \quad (90)$$

$$\tau = \frac{\sqrt{2}}{2} (l_x + l_y) (v_a + v_1 + \omega - v_a + v_1 + \omega + v_a - v_1 + \omega - v_a - v_1 + \omega) \quad (91)$$

$$\tau = \frac{\sqrt{2}}{2} (l_x + l_y) (4\omega) \quad (92)$$

$$\tau = 2\sqrt{2} (l_x + l_y) \omega \quad (93)$$

○  $F_x$

$$F_x = \frac{\sqrt{2}}{2} (F_1 + F_2 + F_3 + F_4) \quad (94)$$

$$F_x = \frac{\sqrt{2}}{2} (v_a + v_1 + \omega + v_a - v_1 - \omega + v_a - v_1 + \omega + v_a + v_1 - \omega) \quad (95)$$

$$F_x = \frac{\sqrt{2}}{2} (4v_a) \quad (96)$$

$$F_x = 2\sqrt{2} v_a \quad (97)$$

○  $F_y$

$$F_y = \frac{\sqrt{2}}{2} (F_1 - F_2 - F_3 + F_4) \quad (98)$$

$$F_y = \frac{\sqrt{2}}{2} (v_a + v_l + \omega - v_a + v_l + \omega - v_a + v_l - \omega + v_a + v_l - \omega) \quad (99)$$

$$F_y = \frac{\sqrt{2}}{2} (4v_l) \quad (100)$$

$$F_y = 2\sqrt{2}v_l \quad (101)$$

Lo cual implica que los valores resultantes son proporcionales a las velocidades de referencia establecidas.

Como se explica en el siguiente capítulo, dependiendo de si el vehículo es operado de forma manual o se desplaza de manera autónoma, se establecerán los valores de  $v_a$ ,  $v_l$  y  $\omega$  de una forma u otra.

### 4.3. Estimación de los coeficientes hidrodinámicos

Como hemos visto en la Ecuación (54), el modelo matemático propuesto depende de tres coeficientes hidrodinámicos:  $\mu$ ,  $\lambda$ ,  $\xi$ . A continuación, se exponen las principales metodologías propuestas a lo largo de los años para estimar los coeficientes hidrodinámicos que caracterizan a los vehículos submarinos, así como la estimación de dichos coeficientes.

#### 4.3.1. Metodologías de estimación

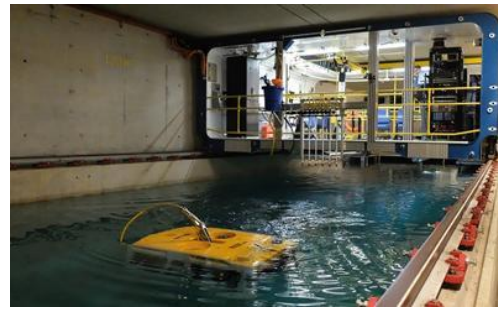
Los métodos más convencionales incluyen procedimientos experimentales, mientras que los más avanzados precisan cálculos numéricos de gran complejidad. Ambos se describen a continuación.

#### Métodos experimentales

Esos experimentos habitualmente son llevados a cabo en canales de ensayo especiales, denominados canales de flujo (ver Figura 18), utilizando el propio vehículo submarino o un modelo a escala del mismo, mientras se miden las fuerzas y los momentos aplicados al vehículo en diversas circunstancias operativas.



a)



b)



c)

**Fig. 18.** Canales de flujo a) Memorial University of Newfoundland [12]; b) University of Southampton [13]; c) Laboratorio de Mecánica de Fluidos, ETSI, Universidad de Huelva

Generalmente, se emplea un mecanismo de movimiento plano (PMM, Planar Motion Mechanism) sobre el canal de ensayo para desplazar el ROV en un movimiento horizontal, tal como ilustra la Figura 19.



**Fig. 19.** Mecanismo de movimiento plano [14]

Dado que el mecanismo PMM montado en el canal de ensayo puede mover el vehículo en múltiples direcciones, al rotarlo permite determinar los coeficientes hidrodinámicos en los 6 grados de libertad. Sin embargo, los PMM son bastante costosos y los procedimientos de prueba consumen una cantidad significativa de tiempo.

Otro enfoque de identificación de los coeficientes hidrodinámicos utiliza los datos medidos por los sensores a bordo junto con la información de las señales de control

de los propulsores durante un conjunto de pruebas simples, en las que el movimiento del vehículo debe restringirse en un solo DoF.

La principal ventaja de usar sensores a bordo es que no se requiere equipo externo y se puede llevar a cabo cada vez que se modifica la configuración del vehículo. De esta forma, este método resulta más económico y altamente repetible.

Sin embargo, la precisión de los resultados depende considerablemente de la exactitud de los sensores y los procedimientos de prueba realizados. De hecho, usar solo los datos de sensores a bordo para identificar las fuerzas aplicadas al ROV por los propulsores puede ser complejo, debido a los efectos de las interacciones entre los mismos.

### Métodos numéricos

En los últimos años, se ha utilizado ampliamente la Mecánica de Fluidos Computacional (CFD) para la determinación de los coeficientes hidrodinámicos de vehículos submarinos. Esta es una de las ramas de la Mecánica de Fluidos que resuelve problemas sobre el flujo de gases y líquidos basándose en métodos numéricos y algoritmos. Estos métodos numéricos tratan de resolver las ecuaciones de Navier-Stokes de dinámica de fluidos, que son un conjunto de ecuaciones en derivadas parciales no lineales.

Los programas CFD, tales como ANSYS Fluent o Wave Analysis MIT, generalmente van acompañados de softwares de diseño asistido por computador (CAD). Este tipo de programas permiten obtener numerosos parámetros y simulaciones, tales como:

- Velocidad alrededor del ROV
- Líneas de corriente
- Formación de vórtices
- Turbulencia formada por la energía cinética

La forma de trabajo de ANSYS consiste en determinar una región alrededor del modelo 3D diseñado [16]. A esta región se le denomina volumen de control y está formada por un gran número de celdas pequeñas que componen una malla. En cada una de estas celdas ANSYS resuelve una matriz algebraica, y mediante un proceso de iteración, llega a una convergencia sobre el resultado final al cabo de un determinado número de iteraciones. Sobre el número de iteraciones es importante señalar que la solución será más precisa cuanto mayor número de iteraciones realice el programa para hallar el resultado final.

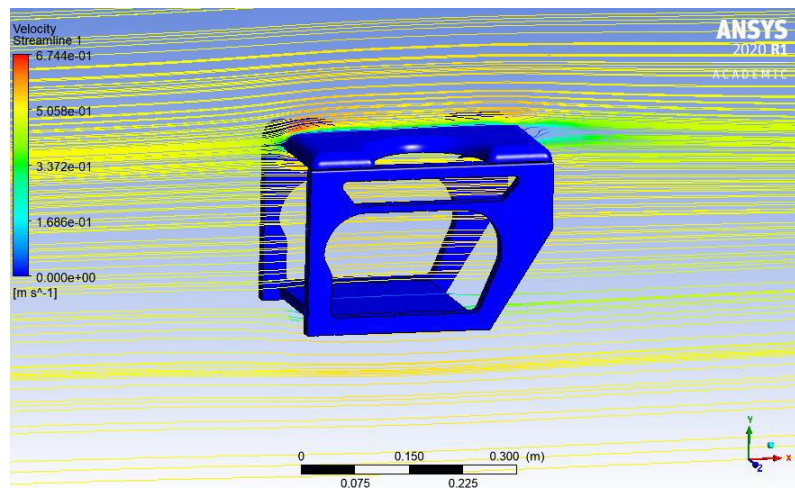


Fig. 20. Simulación de las líneas de corriente sobre la estructura de un ROV en ANSYS [8]

En general, los métodos numéricos proporcionan una alternativa factible cuando no se dispone de las instalaciones y la instrumentación necesarias para realizar las pruebas hidrodinámicas. Además, los softwares CFD facilitan la generación de gráficos postprocesados, que permiten una mejor comprensión del resultado obtenido.

También cabe destacar que los métodos experimentales son más caros y lentos cuantos más puntos de medida se tomen, mientras que los métodos CFD solo requieren de capacidad de procesamiento y almacenamiento de datos.

Sin embargo, la precisión del análisis basado en CFD queda limitada al nivel de similitud con la realidad que presente el software utilizado. De hecho, en determinados casos se necesita simplificar el fenómeno a estudiar para que el software en cuestión pueda ejecutarlo.

#### 4.3.2. Coeficientes hidrodinámicos del Sibiu PRO

Para determinar los parámetros hidrodinámicos del ROV Sibiu PRO, nos hemos basado en el Trabajo Fin de Máster realizado por Juan José Toscano Angulo [15], en el que plantea una metodología de simulación CFD y postprocesado mediante las herramientas de código abierto OpenFOAM y ParaView.

Para ello, en primer lugar, realiza el modelado del vehículo en el software CAD SolidWorks, tal como ilustra la Figura 21.

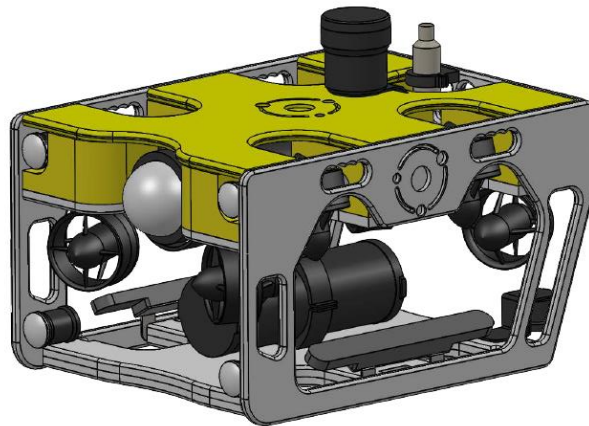


Fig. 21. Modelado del Sibiu PRO en SolidWorks [15]

A partir de este modelo, determina el área transversal del vehículo que interactúa con el fluido en el plano ortogonal a la velocidad de avance,  $A_D$ , tal como ilustra la Figura 22.

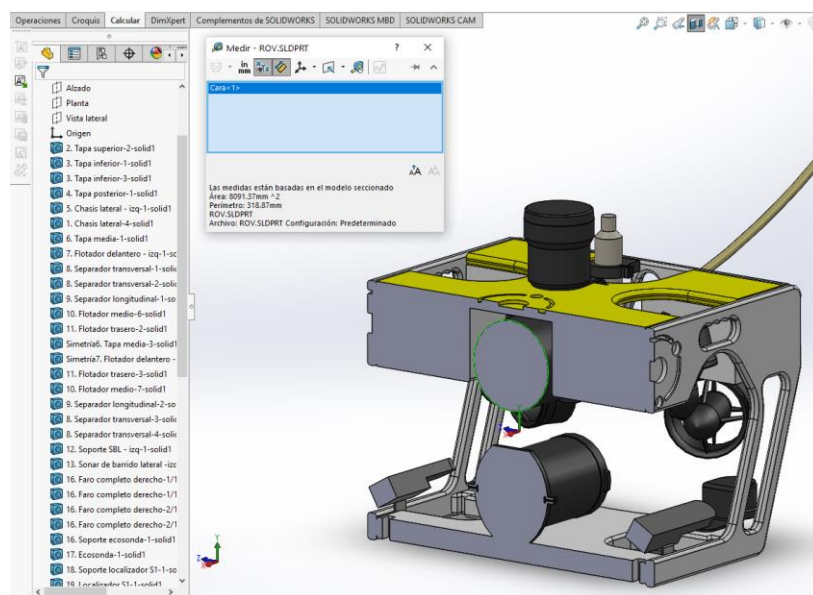


Fig. 22. Sección transversal del ROV [15]

Mediante la herramienta *medir* de SolidWorks, se recopila cada pequeña área que forma la sección transversal. Sumando estas áreas, se concluye un área transversal de arrastre de  $0.05479 \text{ m}^2$ .

A continuación, exporta este modelo a OpenFOAM, donde simula el movimiento de avance del vehículo a una velocidad constante de  $1 \text{ m/s}$  (ver Figura 23).

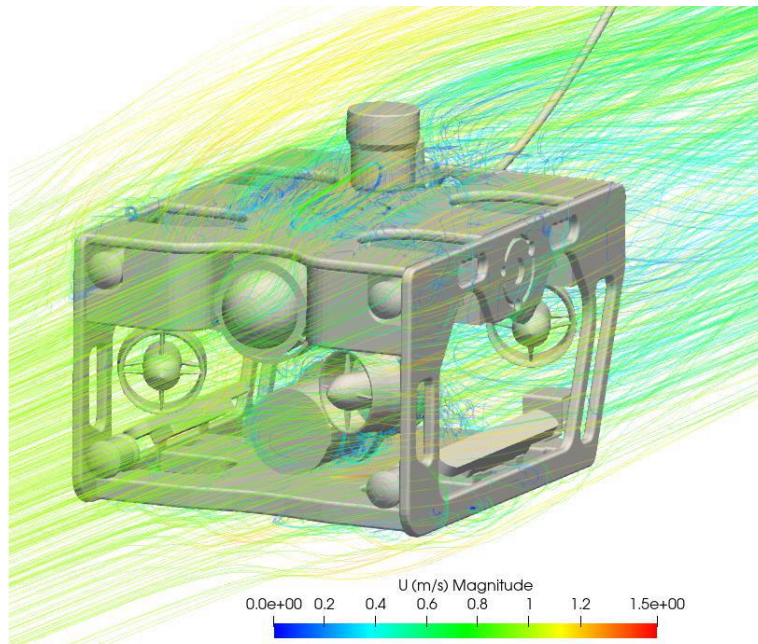


Fig. 23. Simulación de las líneas de corriente sobre el Sibiu PRO [15]

A partir del procedimiento de postprocesado, obtiene los resultados del vector de fuerza arrastre ( $F_D$ ), entre otras, a lo largo de los 125 instantes de tiempo de simulación, siendo su valor medio de convergencia de 50 N.

A partir del valor medio de fuerza de arrastre y el área transversal obtenida en SolidWorks, se puede entonces calcular el coeficiente de arrastre,  $C_D$ , reordenando la ecuación (40):

$$C_D = \frac{2 |F_D|}{\rho A_D v^2} \quad (102)$$

$$C_D = \frac{2 |50 \text{ N}|}{1000 \frac{\text{kg}}{\text{m}^3} \cdot 0.05479 \text{ m}^2 \cdot \left(1 \frac{\text{m}}{\text{s}}\right)^2} = 1.825 \quad (103)$$

De forma similar, obtiene el coeficiente de arrastre para el movimiento transversal. En la Tabla 3, se muestran los coeficientes de arrastre obtenidos en los dos movimientos lineales, así como las áreas transversales utilizadas en cada caso:

	Movimiento	$A_D$ (m <sup>2</sup> )	$C_D$
X	Movimiento longitudinal (surge)	0.05479	1.825
Y	Movimiento lateral (sway)	0.09728	1.830

Tabla 3. Áreas transversales y coeficientes de arrastre de Sibiu PRO [15]

Tal como describe la Ecuación (54), el modelo matemático propuesto depende de tres coeficientes hidrodinámicos:  $\mu$ ,  $\lambda$ ,  $\xi$ . Como vimos anteriormente, calculamos  $\mu$  a partir de la siguiente expresión:

$$\mu = \frac{1}{2} \cdot C_D \cdot A_D \cdot \rho = \frac{1}{2} \cdot 1.825 \cdot 0.05479 \text{ m}^2 \cdot 1000 \frac{\text{kg}}{\text{m}^3} = 49.9959 \frac{\text{kg}}{\text{m}} \quad (104)$$

De la misma forma, calculamos  $\lambda$ :

$$\lambda = \frac{1}{2} \cdot C_D \cdot A_D \cdot \rho = \frac{1}{2} \cdot 1.830 \cdot 0.09728 \text{ m}^2 \cdot 1000 \frac{\text{kg}}{\text{m}^3} = 89.0112 \frac{\text{kg}}{\text{m}} \quad (105)$$

En cuanto a  $\xi$ , para determinar su valor no contamos con información previa. Modificando este parámetro en varias pruebas en la simulación diseñada, podemos aproximar su valor a  $15 \frac{\text{kg}}{\text{m}}$ .

En definitiva, el valor estimado de los tres coeficientes hidrodinámicos  $\mu$ ,  $\lambda$ ,  $\xi$  se muestra en la Tabla 4:

$\mu$	$49.9959 \frac{\text{kg}}{\text{m}}$
$\lambda$	$89.0112 \frac{\text{kg}}{\text{m}}$
$\xi$	$15 \frac{\text{kg}}{\text{m}}$

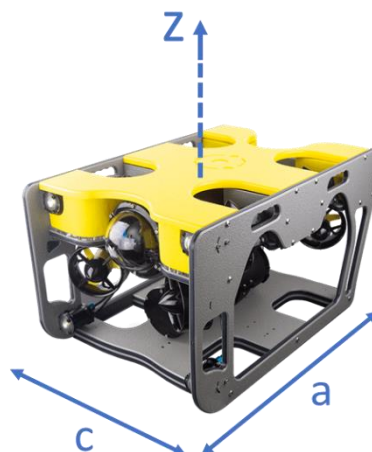
**Tabla 4.** Estimación de los coeficientes hidrodinámicos del Sibiu PRO

#### 4.4. Estimación del resto de parámetros

De acuerdo con el modelo propuesto en la Ecuación (54), es preciso estimar la masa del vehículo, así como su momento de inercia respecto al eje vertical, puesto que es el único eje de giro considerado en el modelo simplificado.

La masa del vehículo  $m$  se ha obtenido a partir de la documentación publicada por *Nido Robotics* [14], siendo 16 kg.

En cuanto al momento de inercia  $I$ , ha sido calculado asimilando el vehículo a un prisma sólido (ver Figura 24).



**Fig. 24.** Momento de inercia respecto al eje vertical

Para un prisma sólido de longitud  $a$  y ancho  $c$ , el momento de inercia con respecto al eje  $z$  viene dado por:

$$I_z = \frac{1}{12} m (a^2 + c^2) \quad (106)$$

Tal como recoge la Tabla 1: Especificaciones técnicas del Sibiu PRO,  $a = 52$  cm,  $c = 39$  cm. Por lo tanto:

$$I_z = \frac{1}{12} \cdot 16 \cdot (0.52^2 + 0.39^2) = 0.5633 \text{ kg} \cdot \text{m}^2 \quad (107)$$

En definitiva, el valor estimado de la masa y momento de inercia se muestra en la Tabla 5:

$m$	$16 \text{ kg}$
$I_z$	$0.5633 \text{ kg} \cdot \text{m}^2$

**Tabla 5.** Estimación de la masa y momento de inercia del Sibiu PRO

# Capítulo 5

## Simulación en Matlab

---

En este capítulo, se describen las simulaciones llevadas a cabo en Matlab, tanto del modelo dinámico del Sibiu PRO, como del sonar Ping360, y la fusión de ambas.

### 5.1. Simulación del movimiento del ROV

Una vez que hemos obtenido las ecuaciones de movimiento del vehículo tal como describe el capítulo 4, procedemos a escribir estas ecuaciones en un script de Matlab. Para resolver las ecuaciones diferenciales del modelo, se ha empleado el método de Runge-Kutta de cuarto orden.

Este método numérico, aunque proporciona un pequeño margen de error con respecto a la solución real del problema, es fácilmente programable en un software para realizar las iteraciones necesarias. Para aplicar este método, es necesario conocer las condiciones iniciales.

Existen variaciones en el método de Runge-Kutta de cuarto orden, pero el más utilizado es el método en el que se elige un paso de integración  $h$  y un número máximo de iteraciones  $n$  tal que:

$$k_1 = h \cdot f(t_i, y_i) \quad (108)$$

$$k_2 = h \cdot f\left(t_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right) \quad (109)$$

$$k_3 = h \cdot f\left(t_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right) \quad (110)$$

$$k_4 = h \cdot f(t_i + h, y_i + k_3) \quad (111)$$

Siendo la iteración:

$$y_{i+1} = y_i + \frac{1}{6} \cdot (k_1 + 2k_2 + 2k_3 + k_4) \quad (112)$$

Para  $i = 0, \dots, n-1$ . La solución se da en el intervalo  $(t_0, t_0 + h \cdot n)$ .

Podemos simular el movimiento del ROV en sus dos modos de funcionamiento: manual y automatizado. Ambos se describen a continuación.

### 5.1.1. Movimiento manual

En este caso, el operador dirige el vehículo a través de un mando tipo joystick. Por lo tanto, las tres velocidades independientes del ROV ( $x_2$ ,  $x_4$ ,  $x_6$ ) son proporcionales a las entradas del mando, tal como se muestra a continuación:

$$\begin{aligned} v_a &= -k_1 * \text{axis}(\text{JOY}, 5); \\ v_l &= -k_2 * \text{axis}(\text{JOY}, 4); \\ w &= -k_3 * \text{axis}(\text{JOY}, 1); \end{aligned}$$

donde:

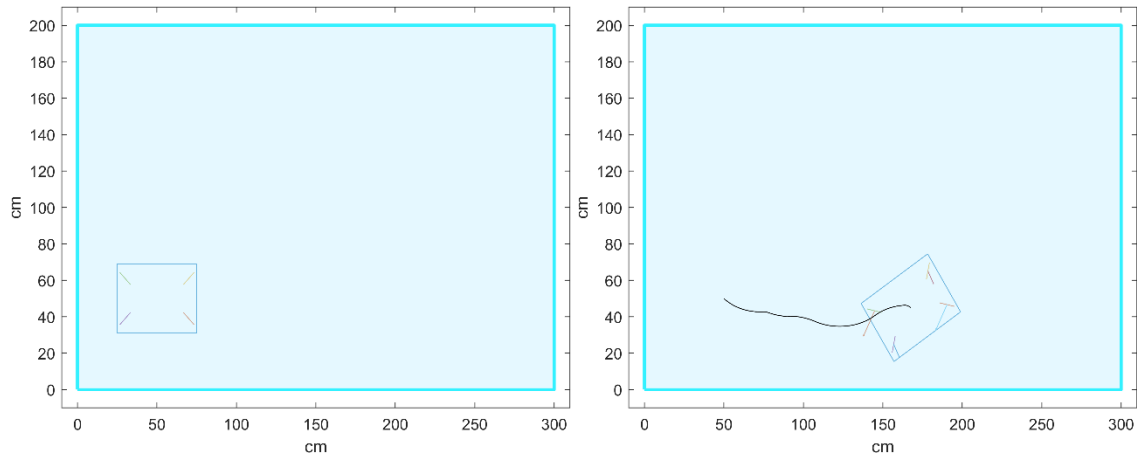
- $v_a$  es la velocidad de avance (eje  $x$ ), por lo que tiene la misma dirección que  $x_2$ ;
- $v_l$  es la velocidad de desplazamiento lateral (eje  $y$ ), por lo que tiene la misma dirección que  $x_4$ ;
- $w$  es la velocidad de giro (eje  $z$ ), al igual que  $x_6$ ;

Como vemos, controlamos la velocidad de avance con el eje vertical de un joystick, la velocidad lateral con el eje horizontal del mismo, y la velocidad de giro con el eje horizontal del otro joystick (ver Figura 25).



**Fig. 25.** Control del ROV mediante el mando

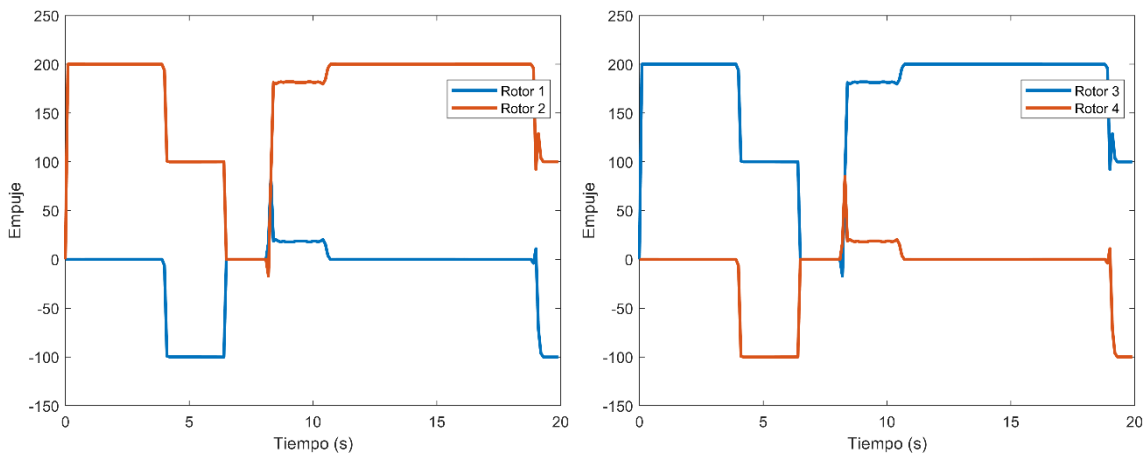
En la Figura 26, se muestra la simulación después de haber realizado un movimiento aleatorio:



**Fig. 26.** Simulación manual (I)

Como vemos, durante la simulación del movimiento se dibuja la trayectoria seguida por el vehículo. Además, representamos las fuerzas o empujes de cada rotor del ROV, para tener una idea más clara de su dirección y magnitud durante el movimiento.

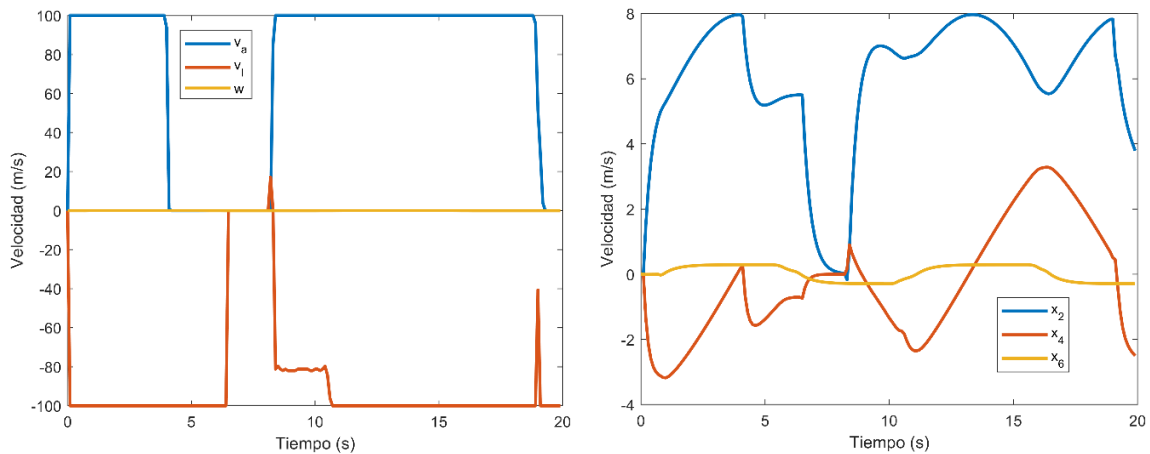
La siguiente imagen muestra la representación gráfica de estos empujes durante la simulación (ver Figura 27):



**Fig. 27.** Representación gráfica de los empujes (I)

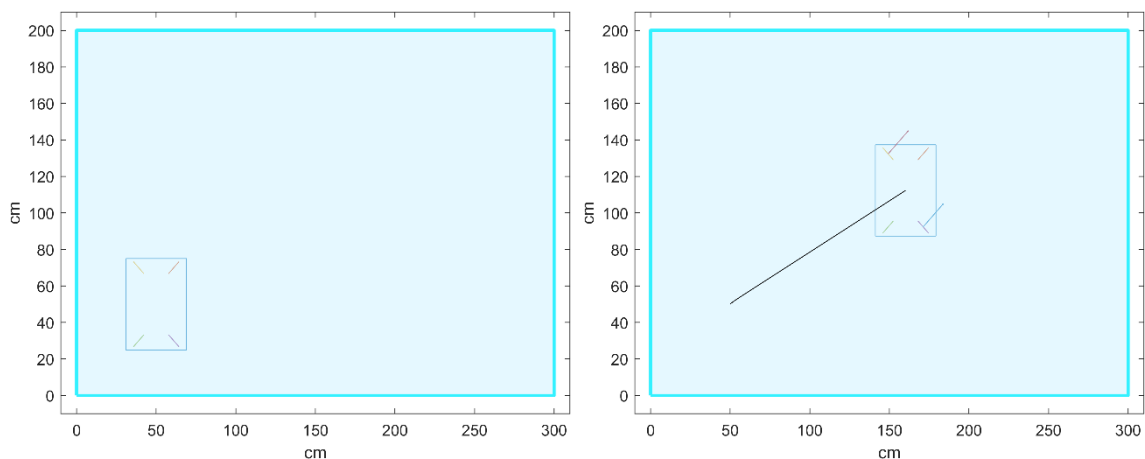
Como vemos, el empuje en los rotores 1 y 3 es idéntico, al igual que el empuje en los rotores 2 y 4. Esto se debe a la propia simetría del vehículo.

A continuación, en la Figura 28 representamos las velocidades introducidas desde el mando ( $v_a$ ,  $v_l$ ,  $w$ ), así como las velocidades del propio vehículo ( $x_2$ ,  $x_4$ ,  $x_6$ ).



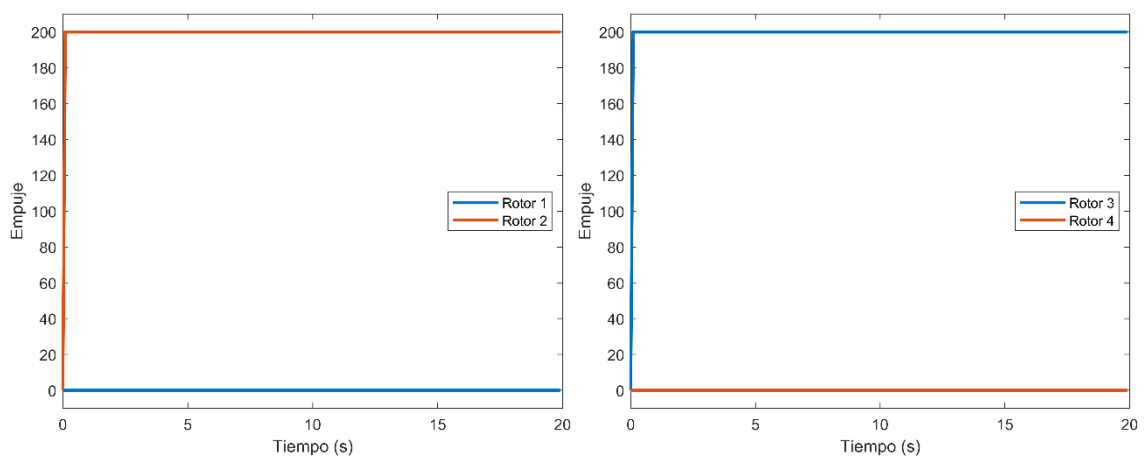
**Fig. 28.** Representación gráfica de las velocidades (I)

Podemos ver que el vehículo sigue aproximadamente la consigna introducida por el joystick. A continuación, simulamos un movimiento en diagonal, en el que la velocidad de avance es positiva, y la velocidad lateral es negativa (ver Figura 29):



**Fig. 29.** Simulación manual (II)

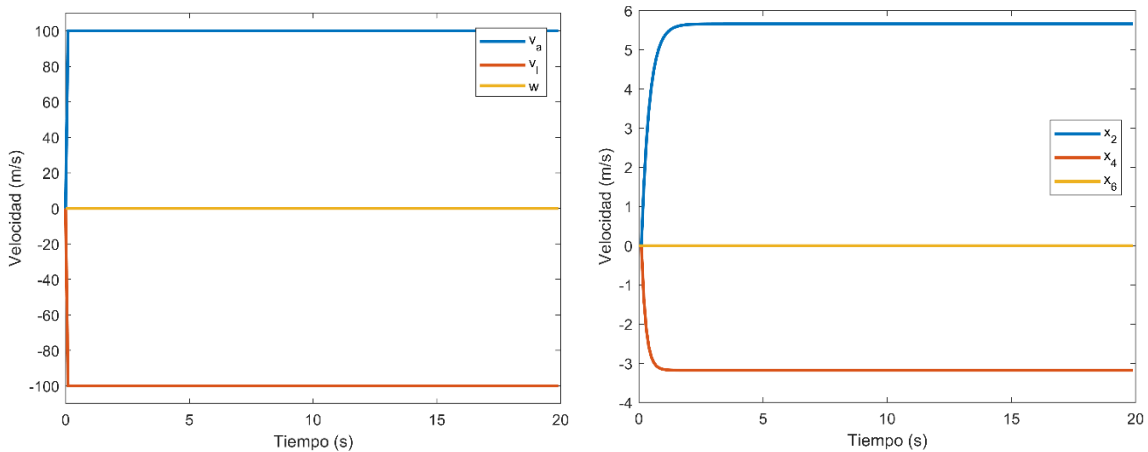
De nuevo, representamos los empujes de los cuatro rotores (ver Figura 30):



**Fig. 30.** Representación gráfica de los empujes (II)

De la misma forma que en la simulación anterior, el empuje en los rotores 1 y 3 es idéntico, al igual que el empuje en los rotores 2 y 4.

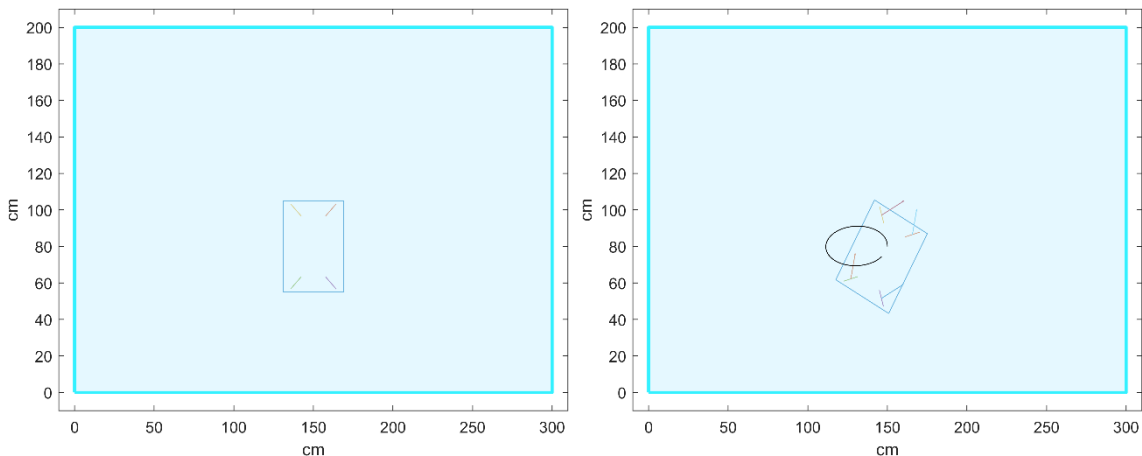
Asimismo, representamos las velocidades introducidas desde el mando, así como las velocidades del propio vehículo (ver Figura 31):



**Fig. 31.** Representación gráfica de las velocidades (II)

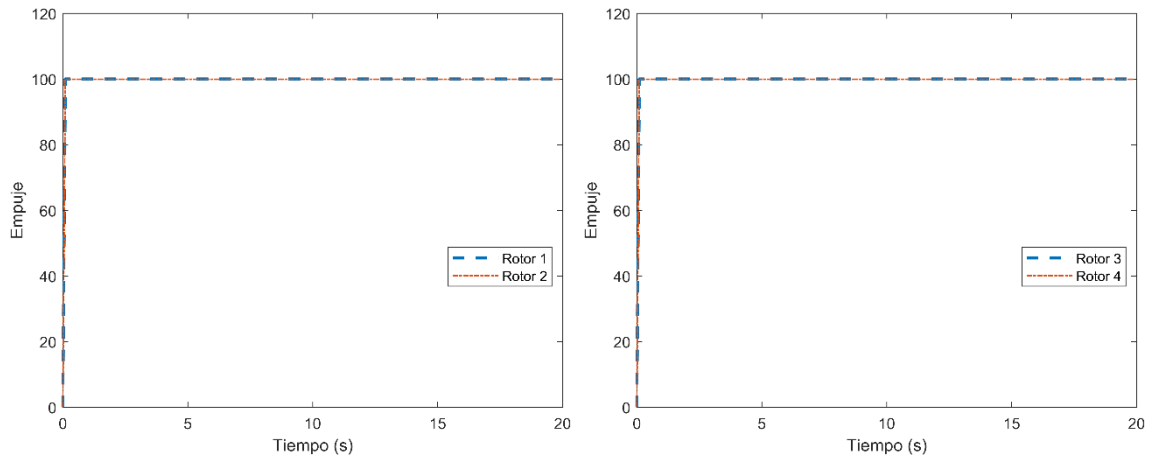
De nuevo, el vehículo ( $x_2$ ,  $x_4$ ,  $x_6$ ) sigue aproximadamente la consigna introducida por el joystick ( $v_a$ ,  $v_l$ ,  $w$ ).

Finalmente, simulamos un movimiento en el que el vehículo gira mientras la velocidad lateral es nula (ver Figura 32):

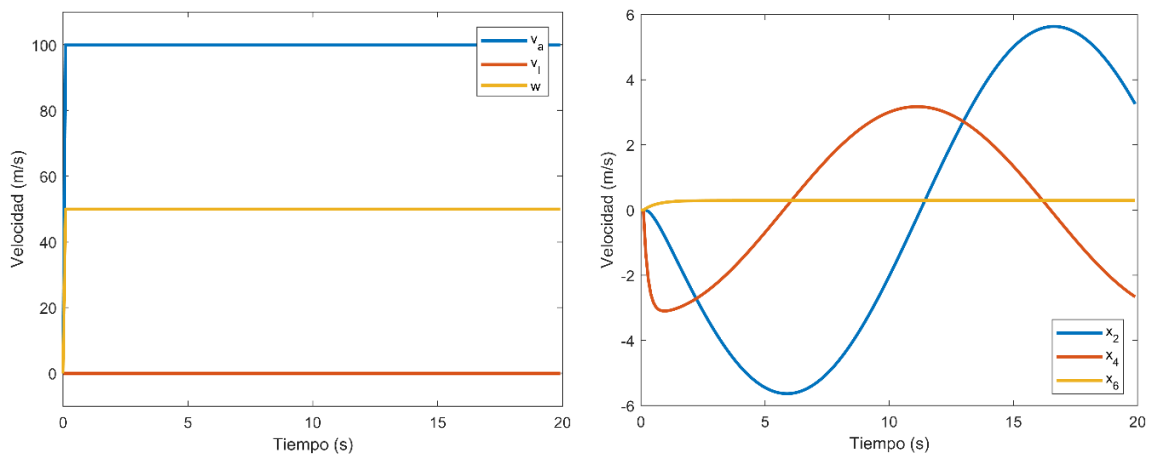


**Fig. 32.** Simulación manual (III)

A continuación, representamos los empujes de los cuatro rotores, así como las velocidades introducidas desde el mando y las velocidades del propio vehículo (ver Figuras 33 y 34):



**Fig. 33.** Representación gráfica de los empujes (III)



**Fig. 34.** Representación gráfica de las velocidades (III)

### 5.1.2. Movimiento automatizado

En este caso, definimos un punto objetivo en el sistema de referencia global al que queremos que se dirija el vehículo. El proceso de navegación se ha dividido en dos etapas: orientación y acercamiento al punto indicado. Esto se debe a que en las pruebas experimentales realizadas con el Sibiu PRO en la piscina de la ETSI de la Universidad de Huelva llevamos a cabo maniobras similares [21].

En la simulación, el vehículo gira hasta que se orienta adecuadamente con respecto al punto objetivo. A continuación, se acerca al mismo, tal como muestra el siguiente diagrama de flujo (ver Figura 35):

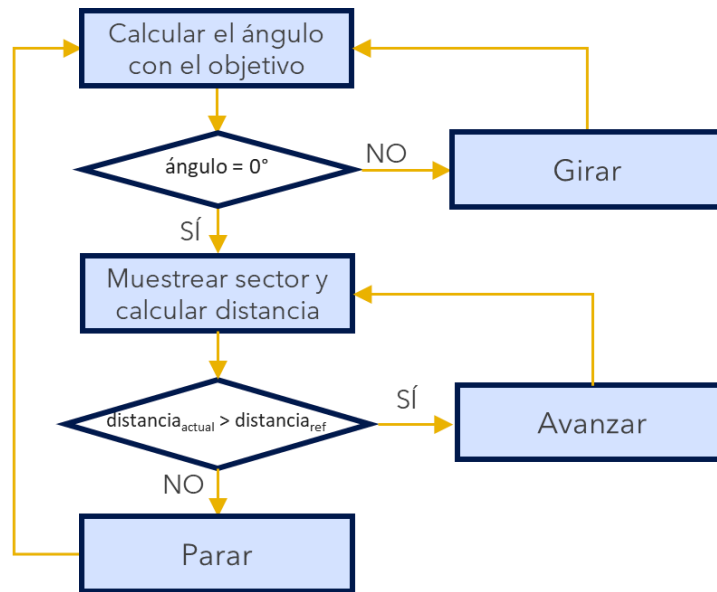


Fig. 35. Diagrama de flujo (I)

En la etapa de orientación, las velocidades de movimiento longitudinal y lateral son nulas, mientras que la velocidad de giro es proporcional al ángulo comprendido entre el objetivo y el ROV ( $theta_e$ ), tal como se muestra a continuación:

$$\begin{aligned}
 v_a(k) &= 0; \\
 v_l(k) &= 0; \\
 w(k) &= k_{p1} * theta_e(k);
 \end{aligned}$$

donde  $k$  es un índice que se incrementa en una unidad en cada iteración.

Para calcular este ángulo, determinamos en primer lugar la matriz de rotación del sistema de referencia local del ROV al sistema de referencia global,  $T$ :

$$T = \begin{bmatrix} \cos(pose(5,k)) & -\sin(pose(5,k)) & pose(1,k); \\ \sin(pose(5,k)) & \cos(pose(5,k)) & pose(3,k); \\ 0 & 0 & 1 \end{bmatrix};$$

donde:

- $pose(1, k)$  es la posición del ROV en el eje  $x$  en la iteración  $k$ ;
- $pose(3, k)$  es la posición del ROV en el eje  $y$  en la iteración  $k$ ;
- $pose(5, k)$  es la orientación del ROV en la iteración  $k$ ;

A continuación, transformamos las coordenadas del punto objetivo, definidas en el sistema de referencia global, al sistema de referencia local del ROV:

$$target = T^{-1} * [objetivo(1) \quad objetivo(2) \quad 1]';$$

donde:

- $T$  es la matriz de rotación del sistema local del ROV al sistema global;
- $objetivo(1)$  es la coordenada x del punto objetivo;
- $objetivo(2)$  es la coordenada y del punto objetivo;

Finalmente, calculamos el ángulo comprendido entre el ROV y el punto objetivo:

$$\theta_e(k) = \text{atan2}(\text{target}(2), \text{target}(1));$$

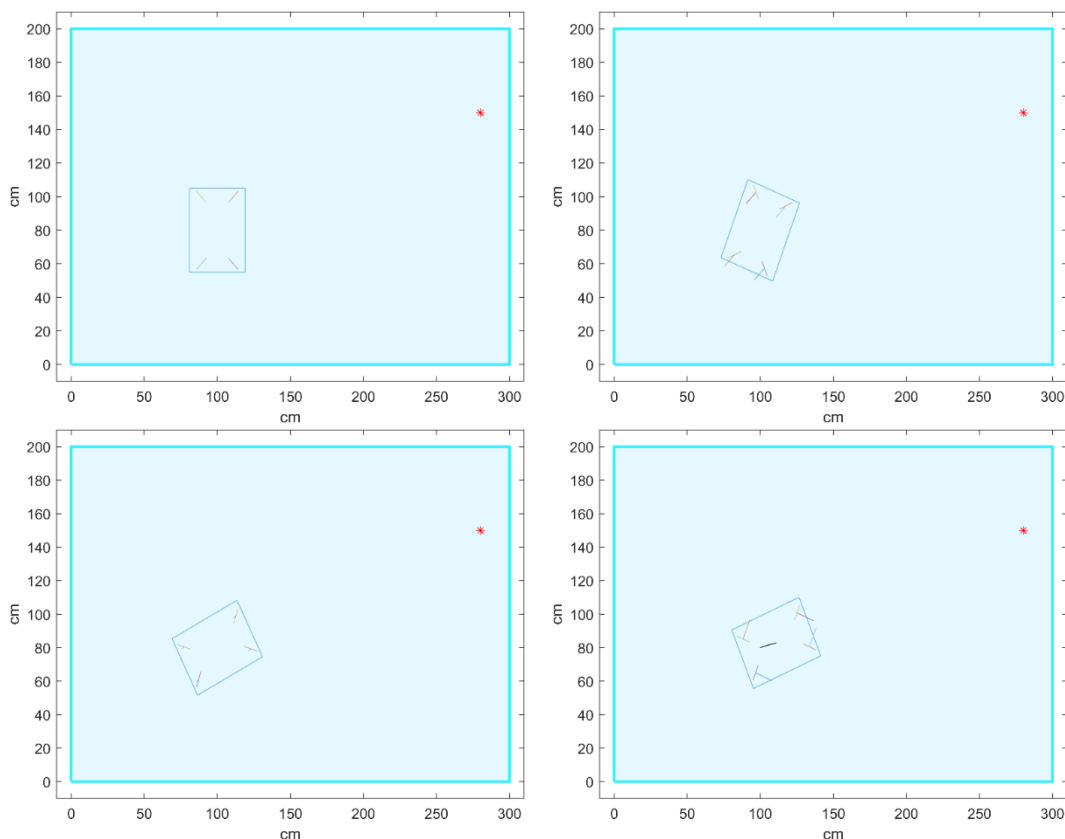
En la etapa de acercamiento, las velocidades de movimiento lateral y giro son nulas, mientras que la velocidad de avance es proporcional a la distancia comprendida entre el objetivo y el ROV ( $dist$ ), tal como se muestra a continuación:

$$\begin{aligned} v_a(k) &= k_p \cdot dist(k); \\ v_l(k) &= 0; \\ w(k) &= 0; \end{aligned}$$

Calculamos esta distancia a partir de la siguiente expresión:

$$dist(k) = \sqrt{(\text{objetivo}(1) - \text{pose}(1, k))^2 + (\text{objetivo}(2) - \text{pose}(3, k))^2};$$

En la Figura 36, podemos ver la secuencia de la simulación descrita:



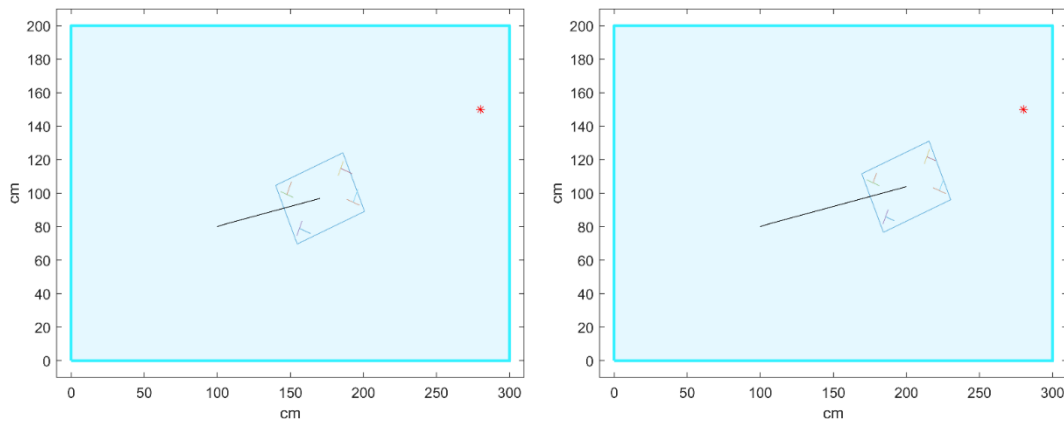


Fig. 36. Secuencia de la simulación (I)

## 5.2. Simulación del sonar Ping360

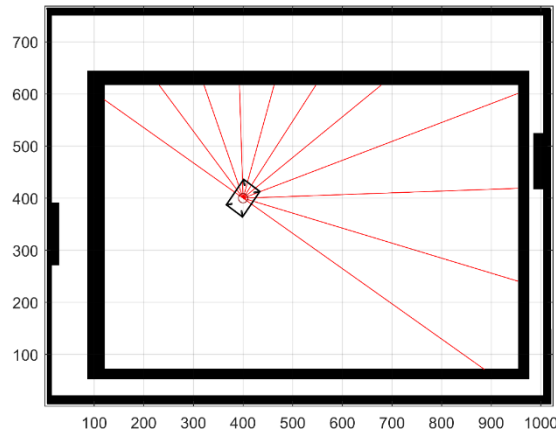
Como se explicó en el capítulo 2, el sonar acústico Ping360 que el Sibiu PRO incorpora adquiere gran importancia durante la navegación, ya que nos permite explorar el entorno que rodea al vehículo, identificar objetos de interés para posteriormente diseñar maniobras automáticas de acercamiento a dichos objetos, etc.

Para que la información proporcionada por el sonar resulte de utilidad, es necesario diseñar algoritmos que procesen y filtren esta información, y proporcionen resultados válidos. Estos algoritmos presentan, en ocasiones, cierta complejidad, por lo que su comprobación previa a las pruebas experimentales in situ puede suponer un gran ahorro de tiempo. Es por esto que la simulación del sonar Ping360 adquiere gran relevancia, puesto que nos permite reproducir el comportamiento del mismo.

La simulación del sonar Ping360 se ha desarrollado a partir de la simulación de un LiDAR. Este es un dispositivo que permite determinar la distancia desde un emisor a un objeto o superficie utilizando un haz láser pulsado. La distancia al objeto se determina midiendo el tiempo de retraso entre la emisión del pulso y su detección a través de la señal reflejada. Es decir, este sensor devuelve un único valor por cada haz emitido.

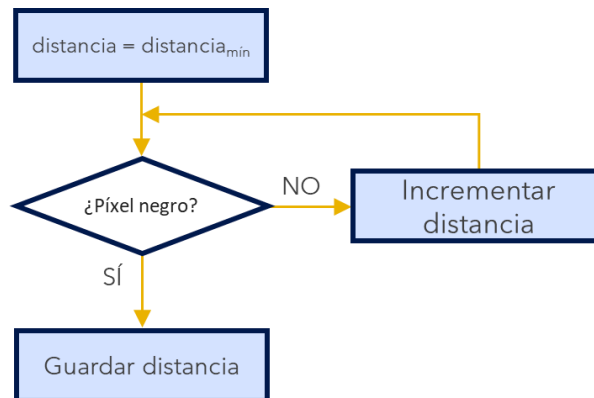
En cambio, el Ping360 recoge 1200 datos por cada haz acústico emitido, siendo un total de 400 haces emitidos en cada vuelta. Cada uno de estos 1200 datos hace referencia a una distancia concreta y almacena la intensidad del eco devuelto, que consiste en un valor entre 0 y 255 (sin rebote e intensidad máxima, respectivamente). Estos se muestran en el interfaz de visualización a través de una paleta de colores, teniendo un color diferente dependiendo de la intensidad del

rebote. Todas estas diferencias deben ser tenidas en cuenta al adaptar la simulación disponible del LiDAR. Esta simulación original se muestra en la Figura 37:



**Fig. 37.** Simulación del LiDAR original

Como vemos, en esta simulación se han empleado píxeles únicamente de dos intensidades, blanco y negro. El sensor LiDAR emite un determinado número de haces, cuya longitud máxima está definida. Para cada haz, se comprueba si ha alcanzado un píxel negro a una determinada distancia, empezando por la distancia mínima a la que no detecta al propio vehículo. En caso negativo, se incrementa la distancia y se vuelve a comprobar, hasta alcanzar la distancia máxima. En caso positivo, se guarda el valor de la distancia, tal como muestra el siguiente diagrama de flujo (ver Figura 38):



**Fig. 38.** Diagrama de flujo (II)

A continuación, se realizan las siguientes modificaciones.

# 1

En primer lugar, acotamos el sector de muestreo a un ángulo determinado. Además, definimos una variable (*Sector*) para que el ángulo de muestreo del sensor sea fácilmente modificable.

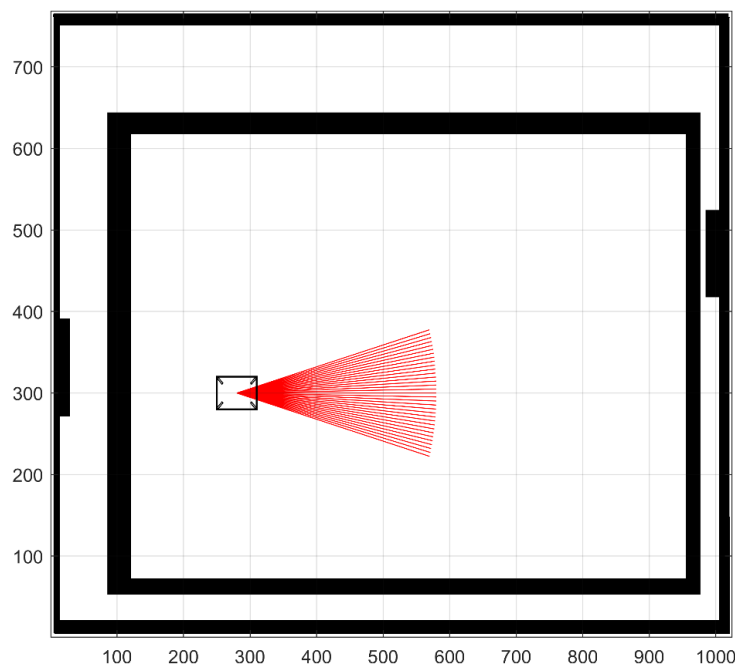
Asimismo, corregimos el número de haces emitido por el sensor. Como se explicó en el Capítulo 2, el motor integrado en el sonar Ping360 gira en incrementos de  $0.9^\circ$ . Por lo tanto, el número de haces para un sector determinado se calcula según la siguiente expresión:

$$n_{haces} = \frac{Sector}{0.9} \quad (113)$$

El resultado obtenido debe ser aproximado, para obtener valores enteros en cualquier caso.

Además, definimos la distancia de muestreo del sonar, es decir, la longitud de los haces.

La simulación obtenida se muestra en la Figura 39. En este caso, el sector de muestreo es  $30^\circ$ .



**Fig. 39.** Simulación del Ping360 (I)

## 2

A continuación, creamos un interfaz en el que se muestren simultáneamente la simulación del vehículo y sensor, así como los datos obtenidos por este último, tal como ilustra la Figura 40.

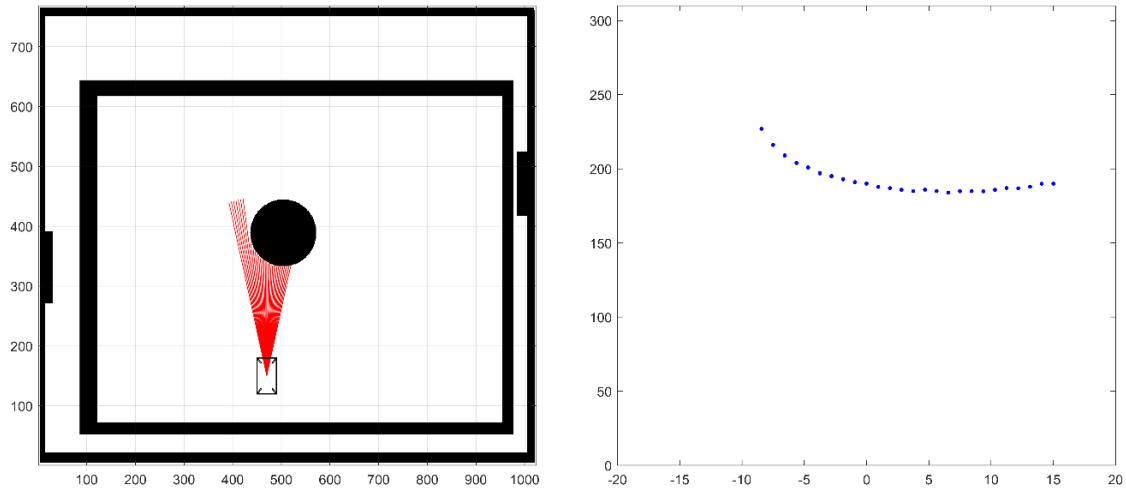


Fig. 40. Simulación del Ping360 (II)

Como vemos, cada haz del sonar guarda la distancia con respecto al vehículo a la que se encuentra el primer obstáculo situado en su trayectoria.

## 3

Posteriormente, modificamos el funcionamiento del sensor: cada haz debe almacenar 1200 valores, que tienen asociado un valor de intensidad y una distancia. Estos 1200 valores se sitúan de manera equidistante entre sí. Por lo tanto, es necesario definir el valor de la distancia que separa cada punto del anterior, a partir de la siguiente expresión:

$$distancia_{separación} = \frac{distancia\ de\ muestreo}{1200} \quad (114)$$

A partir de la distancia de separación entre puntos y la posición actual del vehículo, calculamos las coordenadas del píxel que el sonar debe almacenar:

$$x = x_{vehículo} + distancia_{separación} \cdot (i - 1) \cdot \cos(\theta_{vehículo}) \quad (115)$$

$$y = y_{vehículo} + distancia_{separación} \cdot (i - 1) \cdot \sin(\theta_{vehículo}) \quad (116)$$

donde  $i$  es un índice que se va incrementado en una unidad, desde 1 hasta 1200.

Finalmente, guardamos el valor de ese píxel, y buscamos el siguiente. El funcionamiento del algoritmo descrito se muestra en la Figura 41.

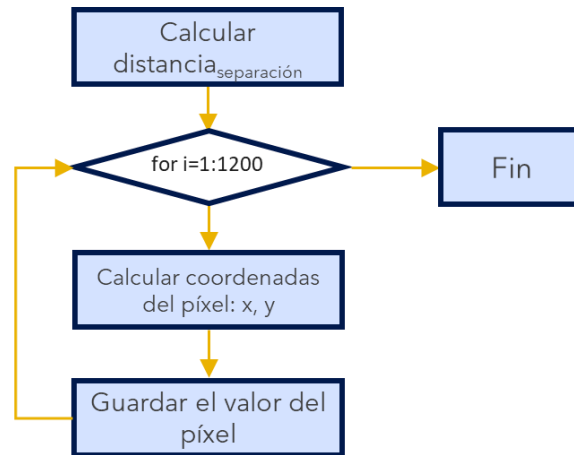


Fig. 41. Diagrama de flujo (III)

A continuación, representamos estos valores en un interfaz de forma circular, tal como funciona el sonar Ping360. El resultado de esta simulación se muestra en la Figura 42.

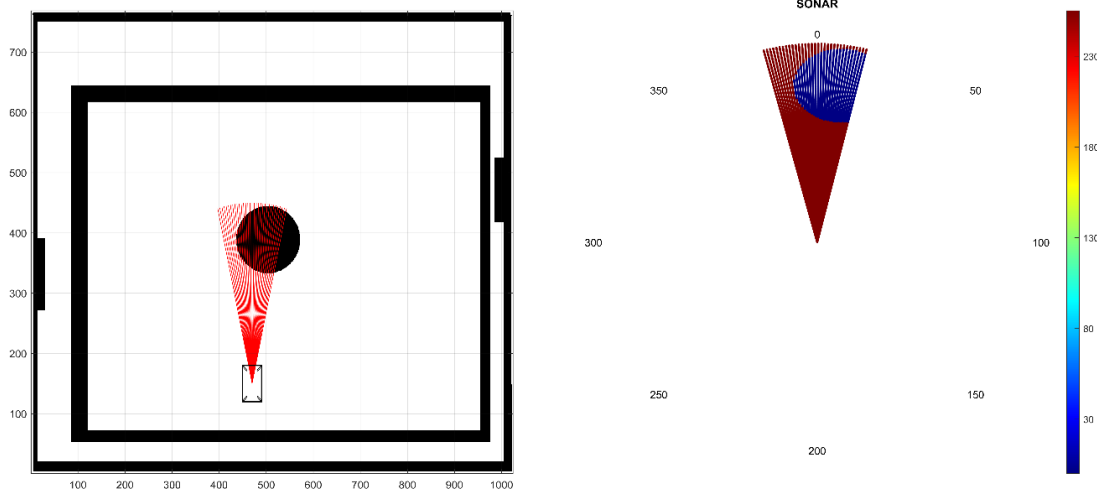


Fig. 42. Simulación del Ping360 (III)

## 4

Finalmente, modificamos la imagen donde se simula el vehículo, para que incluya 256 colores, no solo el blanco y negro. A cada color le corresponde un valor de intensidad (ver Figura 43).

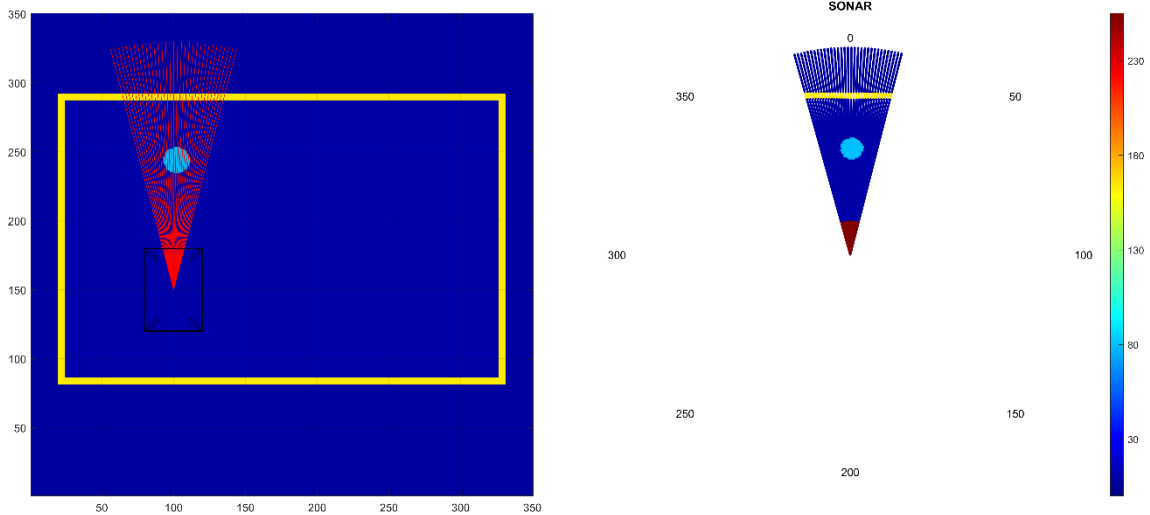


Fig. 43. Simulación del Ping360 (IV)

Modificando el ángulo de barrido, conseguimos que el sonar muestree el entorno completo (ver Figura 44).

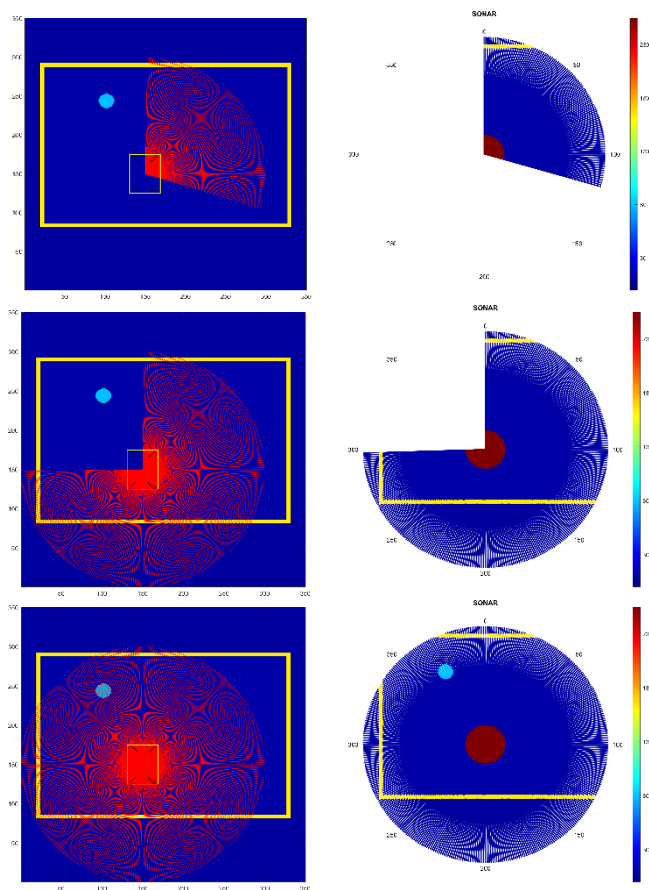


Fig. 44. Simulación del Ping360 (V)

### 5.3. Simulación conjunta del ROV y sonar Ping360

Para llevar a cabo maniobras automáticas de forma eficiente, resulta imprescindible conocer el entorno que rodea al vehículo. En este sentido, el Sibiu PRO está equipado con un sonar acústico de barrido, el Ping360, cuyo funcionamiento y simulación se han descrito en el apartado anterior.

Por esta razón, en este apartado se han desarrollado dos algoritmos que fusionan las dos simulaciones explicadas anteriormente: movimiento automatizado del vehículo y sonar Ping360. Estos algoritmos representan dos objetivos de navegación, descritos posteriormente en este capítulo:

- Navegación realizando transectos perpendiculares a las paredes frontales de la piscina.
- Reorientación y acercamiento a un objeto seleccionado.

Estas simulaciones son el equivalente virtual de los experimentos realizados con el vehículo real, descritos en el TFG correspondiente al grado de Ingeniería Electrónica Industrial [21]. Asimismo, los algoritmos implementados en las simulaciones son similares a los presentados en dicho TFG.

#### 5.3.1. Navegación realizando transectos

Entre las diferentes aplicaciones que un ROV puede desarrollar en una instalación acuática, cabe destacar la navegación autónoma dentro de la instalación mientras toma datos del entorno. De esta forma, el operador puede centrar su atención en otras tareas, tales como analizar la información recogida por el vehículo en tiempo real, etc.

En este caso, el vehículo recorre la piscina de un extremo a otro, manteniendo la perpendicularidad con la pared a la que se acerca. Una vez que se sitúa a cierta distancia de la pared, gira  $180^\circ$  y comienza a navegar hacia la pared opuesta, tal como ilustra la Figura 45:

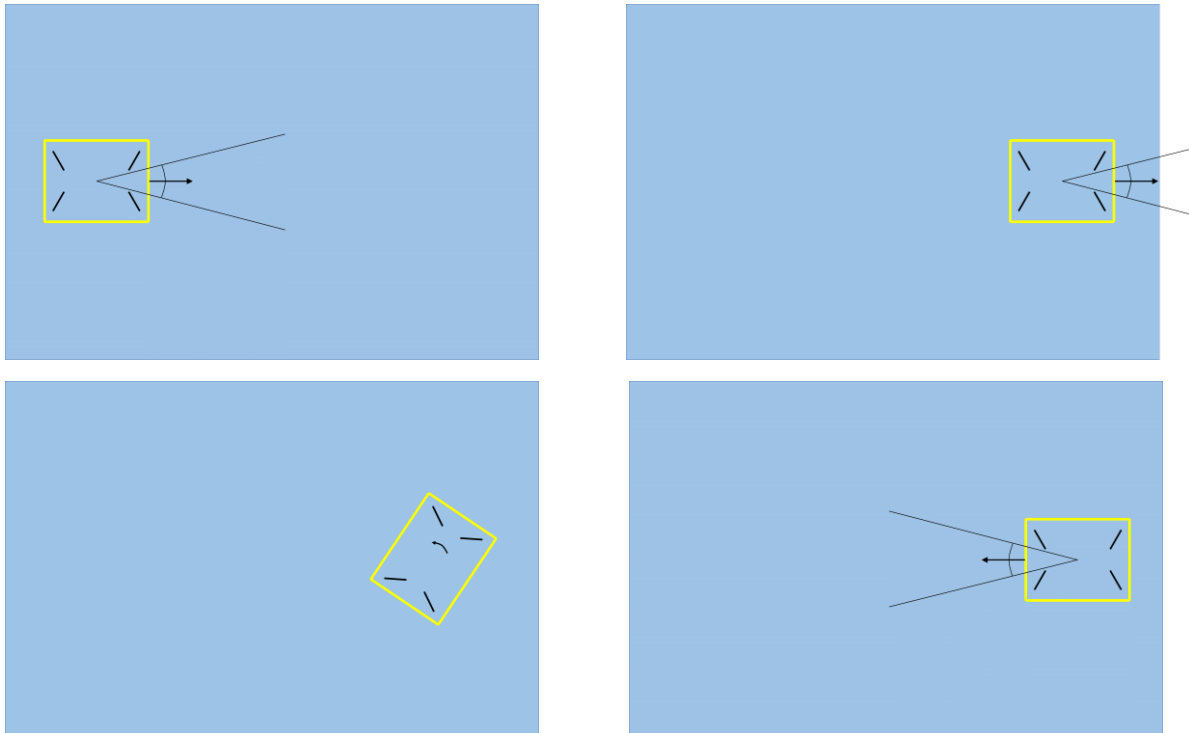


Fig. 45. Navegación realizando transectos

Para conseguir que el algoritmo funcione correctamente, es necesario acotar el ángulo de muestreo del sonar Ping360 a  $20^{\circ}$ - $30^{\circ}$ , aumentando así la frecuencia de actualización de los datos más relevantes para la navegación. Estos datos, que nos permiten determinar el rumbo hacia la pared, son la distancia y el ángulo relativo entre esta y el ROV.

Una vez determinado el rumbo, el controlador de vuelo se encarga de modular las velocidades de avance y de giro para acercar la nave a la pared. Por lo tanto, para regular estas dos velocidades, se han implementado dos controladores independientes que corrigen constantemente el rumbo:

- El controlador de la velocidad de avance, que depende de la distancia a la pared.
- El controlador de la velocidad de giro, que depende del ángulo entre la pared y el ROV.

Estos controladores son de tipo proporcional (P). El controlador proporcional es uno de los más utilizados en la industria. Su principal ventaja es la simplicidad de implementación, ya que, aunque el modelo del sistema sea completamente desconocido, mediante un ajuste manual es posible encontrar la ganancia  $k_p$  que obtenga un rendimiento adecuado del controlador. En concreto, la ganancia proporcional disminuye el tiempo de subida, aunque aumenta el sobreimpulso y degrada la estabilidad del sistema.

El funcionamiento de este algoritmo se muestra en el siguiente diagrama de flujo (ver Figura 46).

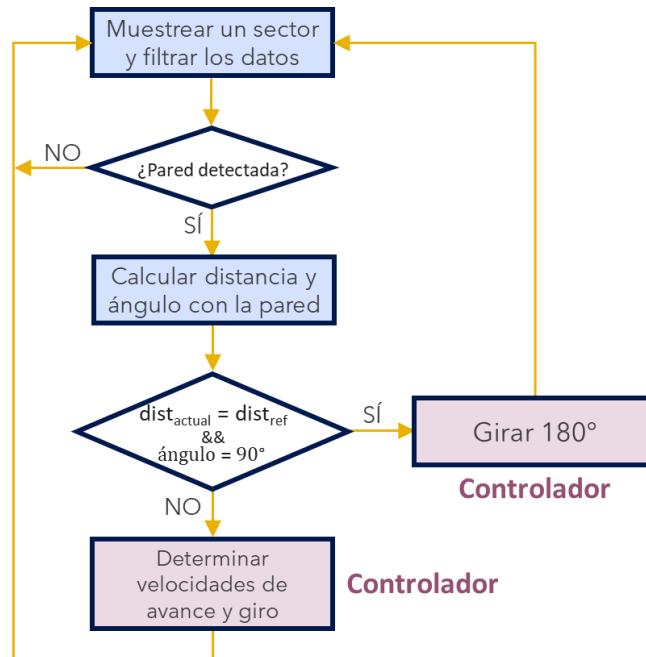


Fig. 46. Diagrama de flujo (II)

Para realizar el giro de 180°, el controlador sigue el siguiente diagrama de flujo (ver Figura 47):

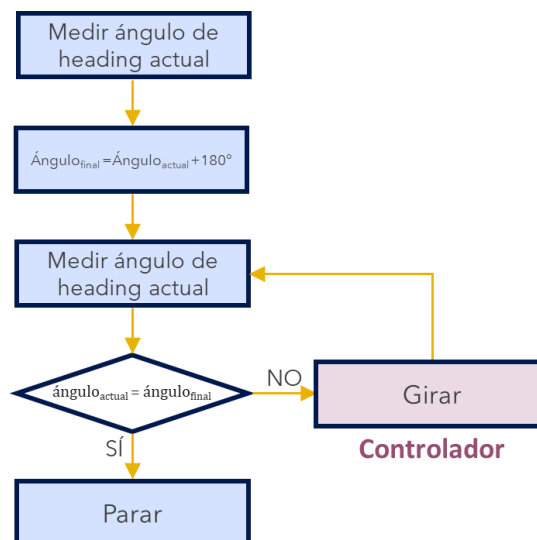


Fig. 47. Diagrama de flujo (V)

Si repetimos este proceso indefinidamente, el vehículo navega realizando transectos entre los extremos de la piscina.

El resultado de esta simulación se muestra en la Figura 48:

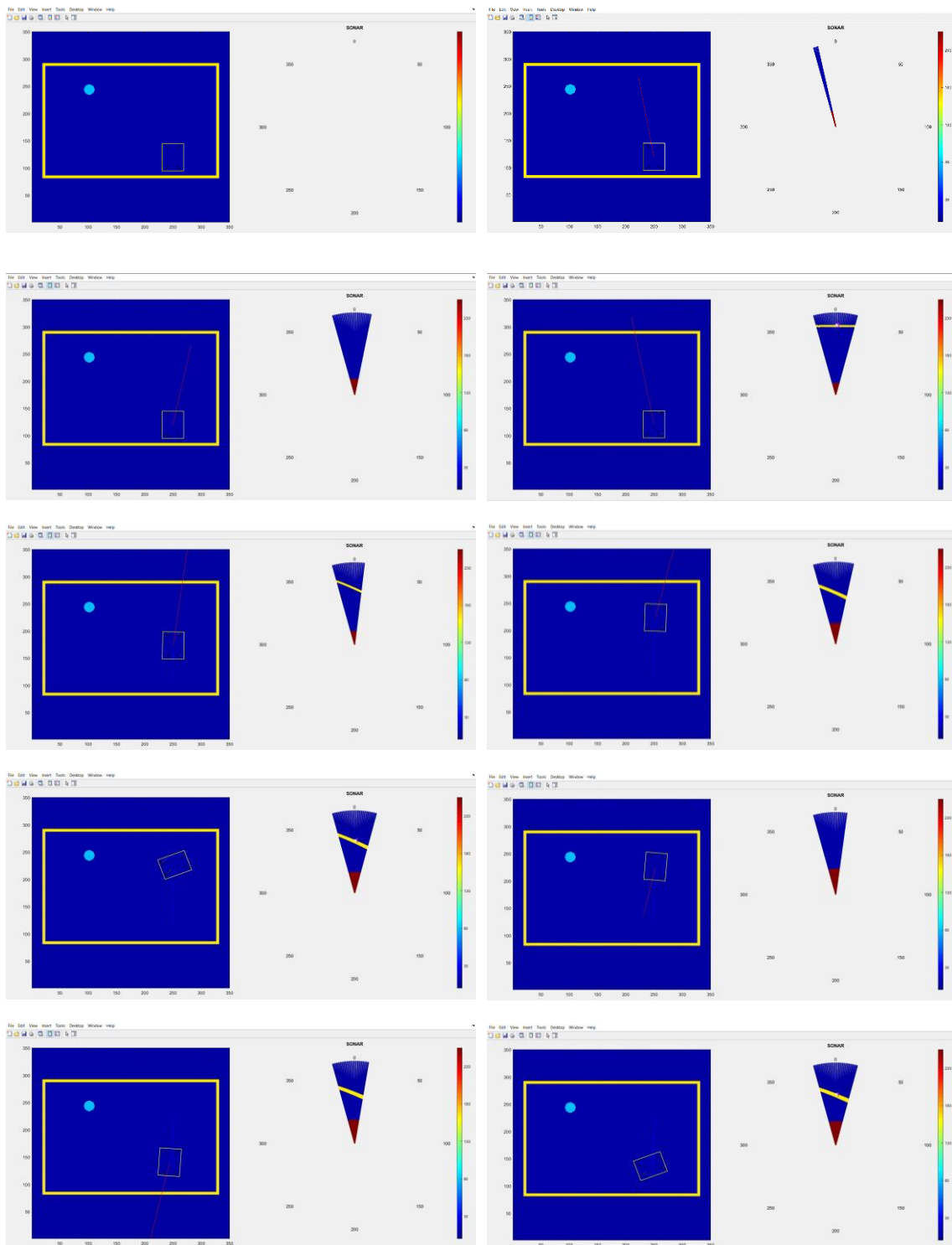


Fig. 48. Secuencia de la simulación (II)

Los resultados experimentales presentados en [21] se muestran en la Figura 49:

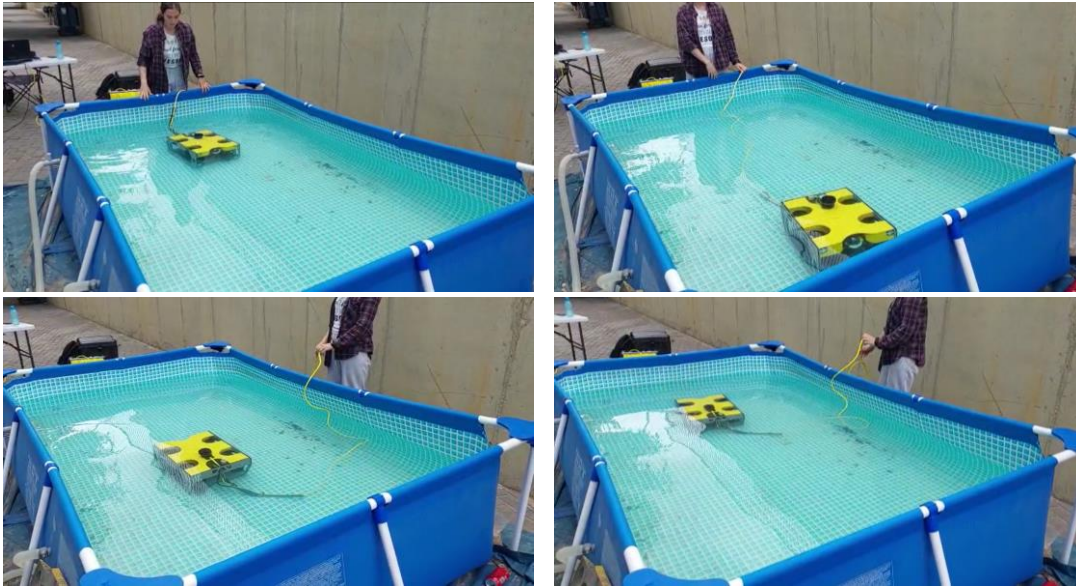


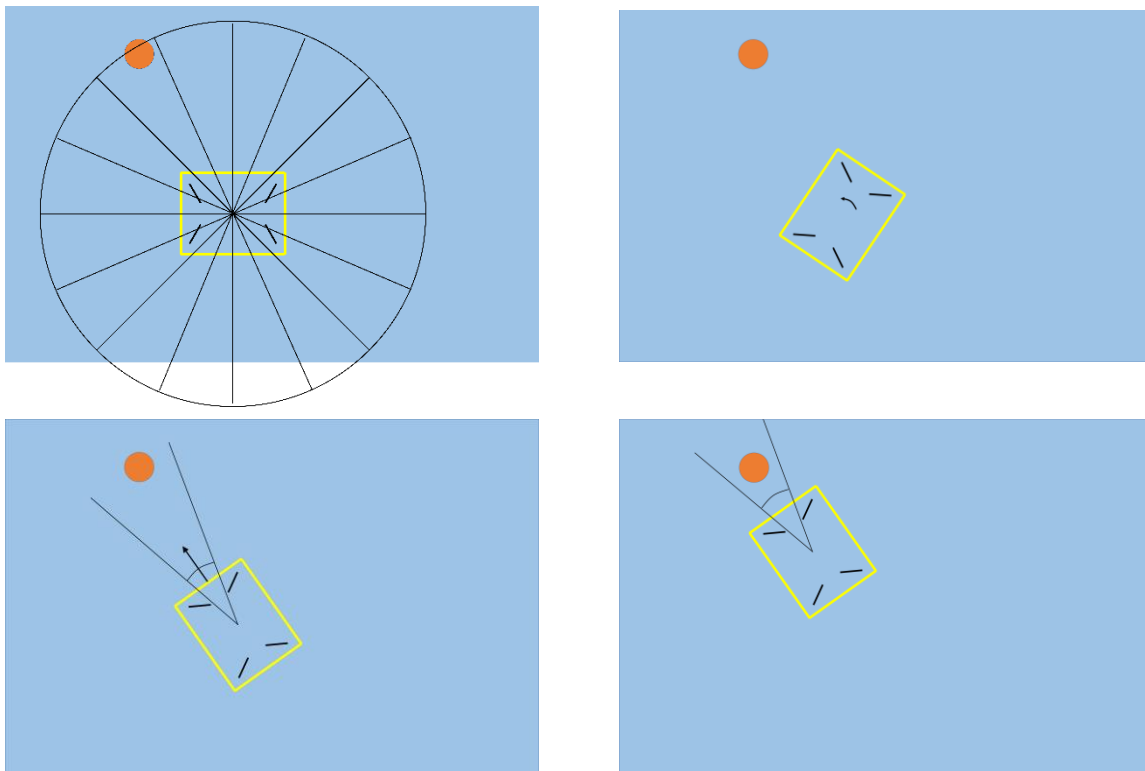
Fig. 49. Pruebas experimentales (I) [21]

Como vemos, la simulación recrea de forma bastante similar el comportamiento observado durante las diversas pruebas llevadas a cabo con el propio vehículo.

### 5.3.2. Navegación hacia un objeto

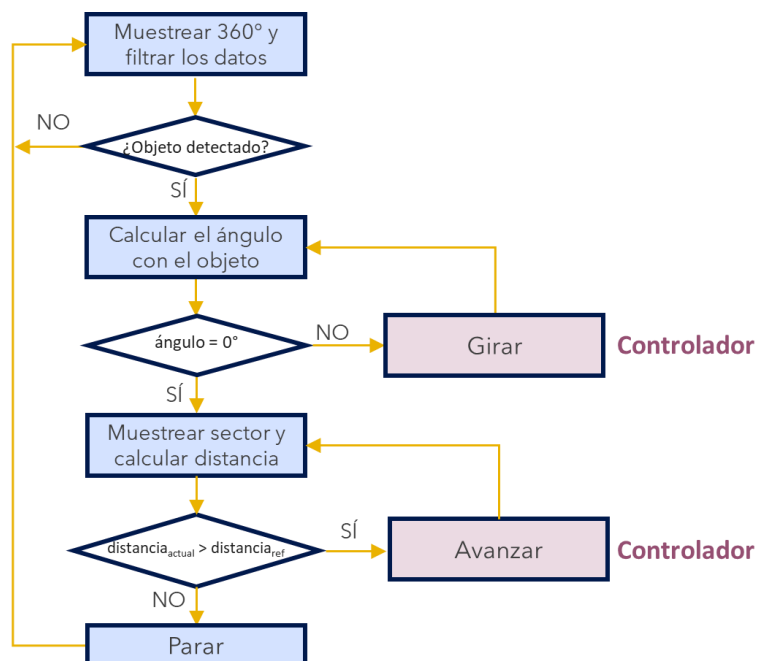
Otra de las aplicaciones que un ROV puede llevar a cabo consiste en la localización de elementos de interés, tales como peces, cardúmenes u otros objetos de naturaleza variada. Una vez se han localizado dichos elementos, el siguiente paso es dotar al ROV con la capacidad de navegación autónoma hacia el objetivo de interés seleccionado.

En este tipo de navegación, en primer lugar, realizamos un barrido con el sonar de 360°, y así identificamos los objetivos disponibles. Una vez seleccionado el objetivo, el ROV se orienta hasta quedar alineado con el mismo. Finalmente, avanza hasta situarse a cierta distancia del objeto en cuestión. Cuando alcanza la posición deseada, muestrea de nuevo el entorno para identificar nuevos objetivos si los hubiera, tal como ilustra la Figura 50.



**Fig. 50.** Navegación hacia objeto

El funcionamiento de este algoritmo se muestra en el siguiente diagrama de flujo (ver Figura 51).



**Fig. 51.** Diagrama de flujo (VI)

En este caso, el algoritmo de control implementado se divide tres etapas diferentes, tal como ilustra la Figura 52.

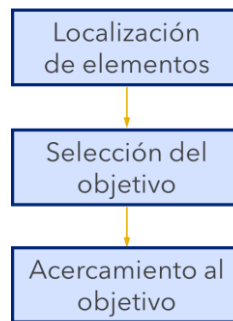


Fig. 52. Diagrama de flujo (III)

Las etapas del algoritmo de control se describen a continuación.

### Localización de elementos

Este módulo está orientado a dar información al usuario sobre los elementos que se encuentran alrededor del ROV. Para identificar los posibles objetivos a seguir, se lleva a cabo un procesamiento de los datos obtenidos, dividido en las siguientes etapas:

- Identificación de elementos a partir de la agrupación de elementos (clustering) tomando como característica la intensidad del eco devuelto.
- Cálculo de los datos que caracterizan la posición de los objetos identificados (distancia y ángulo con respecto al ROV).

En la simulación, obtenemos la siguiente imagen (ver Figura 53):

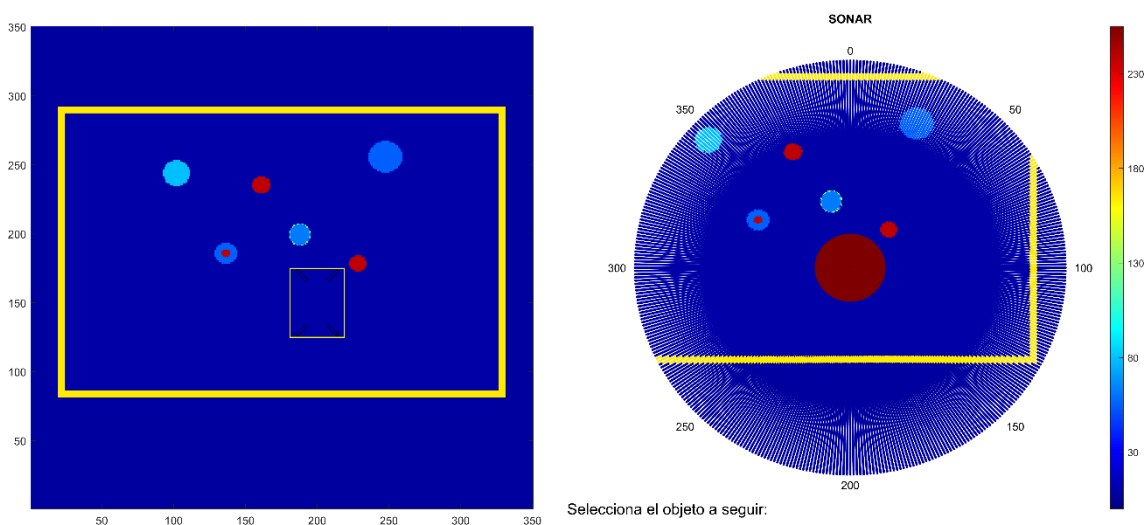


Fig. 53. Muestreo del entorno de 360°

A continuación, se lleva a cabo el procesamiento de datos. Tras aplicar el filtro descrito anteriormente, obtenemos la Figura 54:

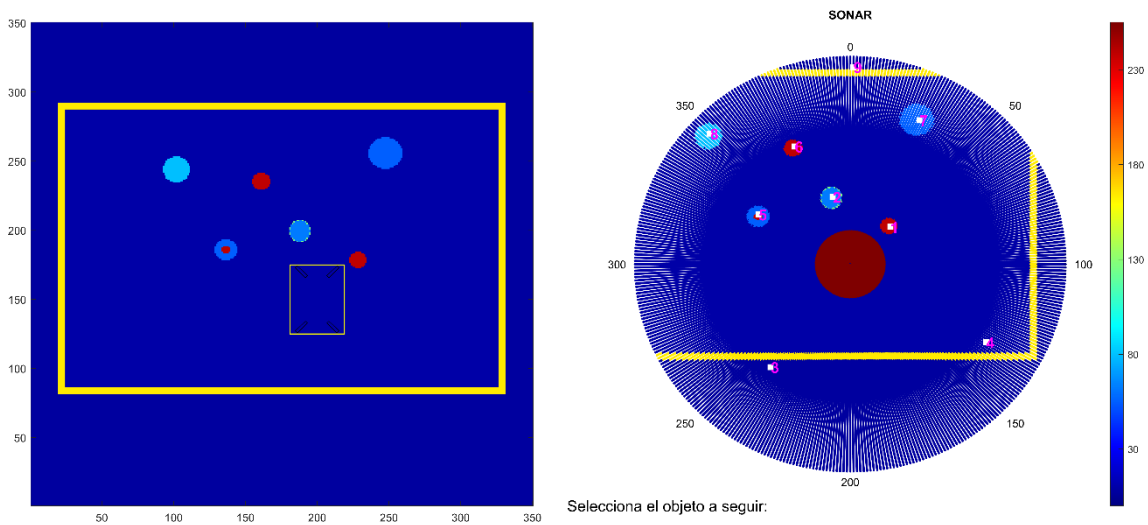


Fig. 54. Identificación de objetos

Como vemos en esta figura, el algoritmo identifica los objetos presentes, que aparecen numerados en dicha imagen.

### Selección del objetivo de navegación

A continuación, a través de un interfaz, el operario escoge el elemento identificado que sea de su interés. En ese momento, el elemento seleccionado se convierte en el objetivo de navegación. Dicho interfaz es la figura donde se representa la imagen acústica obtenida, en la que aparece un menú desplegable donde se muestran los números identificativos de todos los objetos detectados, tal como ilustra la Figura 55:

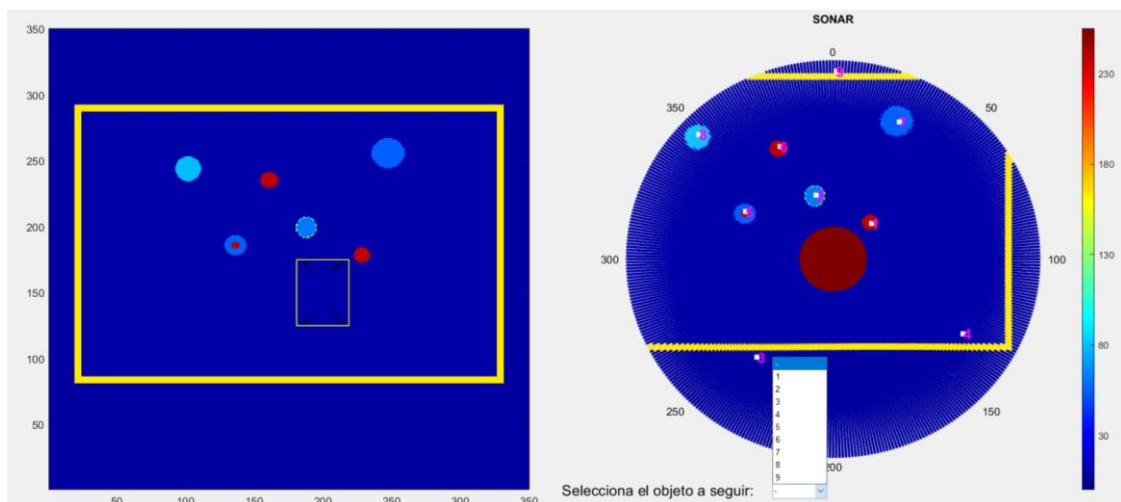


Fig. 55. Interfaz para la selección del objetivo

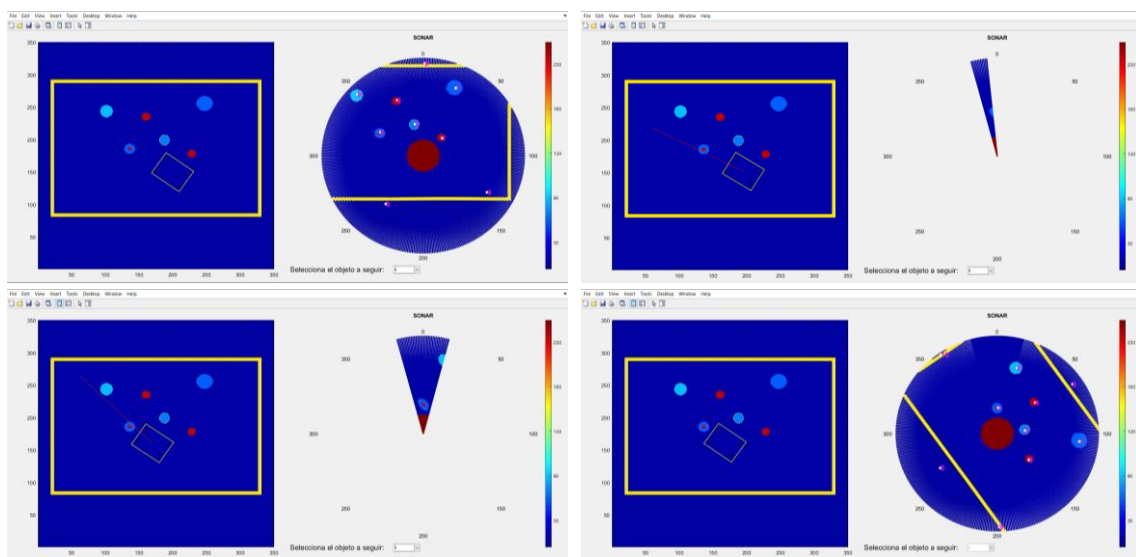
Este interfaz permite seleccionar, mediante el uso del ratón, de forma cómoda y sencilla el objetivo de navegación, sin necesidad de utilizar un teclado.

### Acercamiento al objetivo

Los controladores independientes de velocidad de avance y velocidad de giro que corrigen constantemente el rumbo del vehículo son los mismos que los explicados en el apartado anterior.

Además, cabe destacar que mientras el ROV se acerca al objeto, es necesario acotar el ángulo de muestreo del sonar a  $20^{\circ}$ - $30^{\circ}$ , aumentando así la frecuencia de actualización de los datos más relevantes para la navegación. Estos datos, que nos permiten determinar el rumbo hacia el objeto, son la distancia y el ángulo relativo entre este y el ROV.

El resultado de esta simulación se muestra en la Figura 56. En este caso, partiendo de la Figura 55, seleccionamos el objeto nº 5:



**Fig. 56.** Secuencia de simulación (III)

Los resultados experimentales presentados en [21] se muestran en las Figuras 57 y 58:

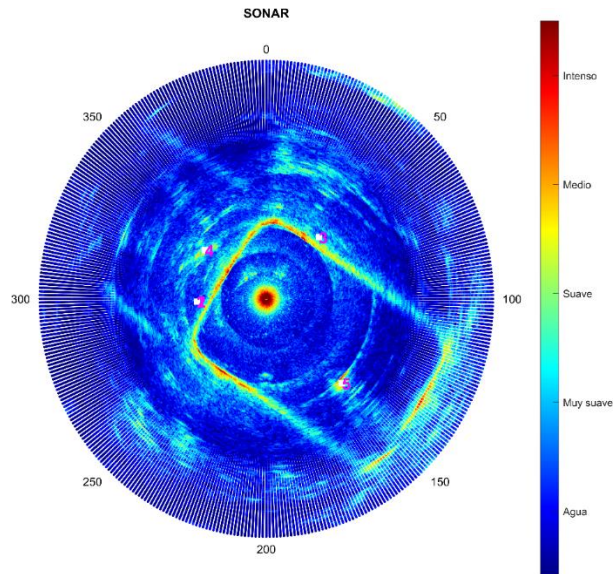


Fig. 57. Muestreo con el sonar Ping360 [21]

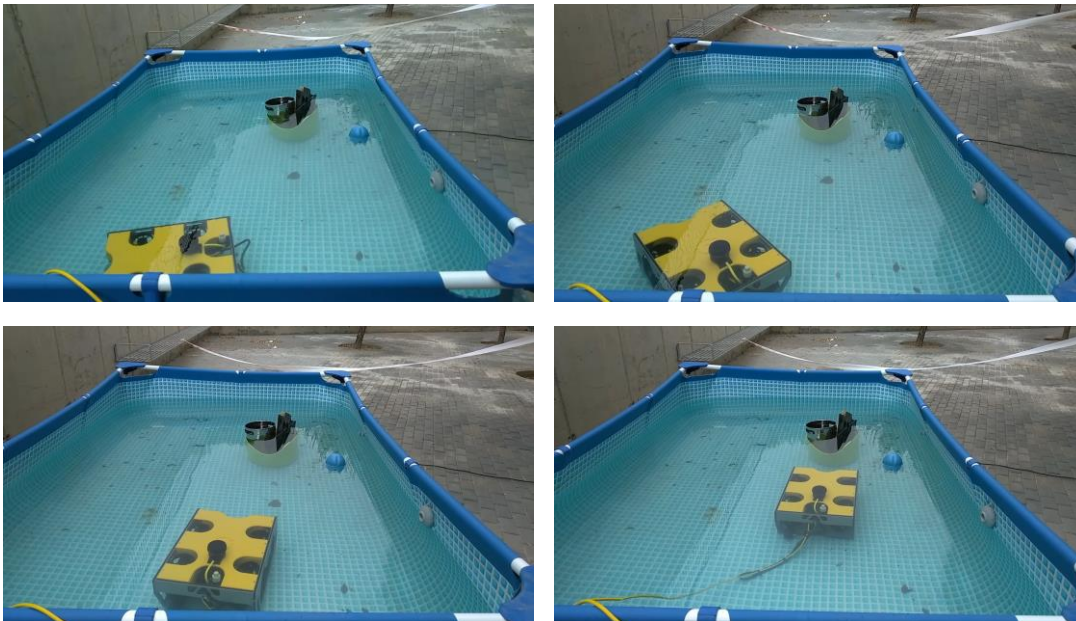


Fig. 58. Pruebas experimentales (II) [21]

Como vemos, de nuevo la simulación recrea de forma bastante similar el comportamiento del vehículo observado durante las pruebas llevadas a cabo.

# Capítulo 6

## Conclusiones

---

En este capítulo se exponen las reflexiones y resultados de este trabajo de fin de grado, así como estrategias para futuras mejoras.

### 6.1. Resultados

La aplicación de nuevas tecnologías es un factor clave para lograr un correcto aprovechamiento y una explotación sostenible de los recursos pesqueros y acuícolas. En este sentido, la implantación de tecnologías 4.0, como la robótica, serán determinantes en la gestión y desarrollo de estos sectores.

Este trabajo aborda el desarrollo de un modelo matemático simplificado para la dinámica de un vehículo submarino comercializado, el Sibiu PRO. Posteriormente, se llevó a cabo una simulación en ordenador, utilizando Matlab, tanto del movimiento del ROV como del sonar Ping360 que incorpora, que permite estudiar el comportamiento del vehículo en diferentes circunstancias.

Además, la simulación del Sibiu PRO supone el primer paso para el control autónomo del vehículo submarino, permitiendo el estudio de varios diseños de control antes de la implementación en el vehículo.

### 6.2. Trabajo futuro

Si bien este TFG logró resolver todos los problemas planteados al principio, todavía hay espacio para más mejoras.

- Modelo dinámico del Sibiu PRO

El modelo de simulación generalmente no es una aproximación exacta del vehículo real debido a los coeficientes hidrodinámicos poco conocidos, perturbaciones difíciles de modelar e incertidumbres en los parámetros del modelo. Un refinamiento del modelo de simulación para reducir el error de aproximación entre el vehículo real y el modelo de simulación se puede mediante un análisis experimental del vehículo submarino.



- Simulación del Sibiu PRO

En la navegación del vehículo intervienen diversos factores externos difícilmente medibles que debemos tratar de corregir, tales como la tensión del cable umbilical, la deriva del propio vehículo, etc. Estos podrán ser incluidos en la simulación en ordenador.

## Bibliografía

- [1] Fossen, T. I.: Handbook of marine craft hydrodynamics and motion control (2011).
- [2] SNAME: *The Society of Naval Architects and Marine Engineers. Nomenclature for Treating the Motion of a Submerged Body Through a Fluid*, In: Technical and Research Bulletin No. 1–5, 1950.
- [3] Fossen, T. I.: Marine Control Systems: Guidance, Navigation and Control of Ships and Underwater Vehicles (2002).
- [4] Fossen, T. I.: Guidance and Control of Ocean Vehicles (1999).
- [5] T. I. Fossen, *Validación experimental del modelo dinámico de un cuerpo en maniobras de inmersión*, John Wiley and Sons, 1999.
- [6] <https://www.nidorobotics.com/sibiu-pro> (Accedido en 03/2022)
- [7] <https://bluerobotics.com/learn/understanding-and-using-scanning-sonars/> (Accedido en 03/2022)
- [8] Vélez-Bermejo, C.: Modelación hidrodinámica de un vehículo submarino operado remotamente (ROV) usando CFD, Trabajo Fin de Grado, Universidad de Huelva (2020).
- [9] Ridao, P., Carreras, M., Hernández, E., Palomeras : Underwater telerobotics for collaborative research, *Advances in telerobotics*, Springer 501 pp (2007).
- [10] Bergström, B.I., Larsson, J., Pettersson, J.O.: Use of a Remotely Operated Vehicle (ROV) to study marine phenomena: I. Pandalid shrimp densities, *Marine Ecology Progress Series* 37(1), 97-101 (1987).
- [11] Gutiérrez-Estrada, J.C., Peregrín-Rubio, A., Gómez-Bravo, F., Pulido-Calvo, I., González-Cabrera, M.: Informe introductorio sobre desarrollo y evaluación de sistemas fijos (boyas), pequeños vehículos marinos (ROVs) y análisis de datos e imágenes en zona estuáricas y explotaciones acuícolas, Informe de la Actividad 1 del Proyecto POCTEP 0622-KTTSEADRONES-5-E (2019).
- [12] Wang, Y., Thanyamanta, W., Bulger, C., Bose, N., Hwang, J.: Microbubbles as Proxies for Oil Spill Delineation in Field Tests, *Journal of Marine Science and Engineering* (2021).
- [13] <https://www.southampton.ac.uk/engineering/research/facilities/towing-tank-news/12-saab-seaeye-tuning-test.page> (Accedido en 05/2022)
- [14] Derradji-Aouat, A., Thiel, A.: Terry fox Resistance Tests - Phase III (PMM) ITTC Experimental Uncertainty Analysis Initiative (2004).
- [15] Toscano-Angulo, J.J.: Simulación hidrodinámica y validación experimental de un vehículo submarino operado remotamente, Trabajo Fin de Máster, Universidad de Huelva (2021).



- [16] Fossen, T. I., Schjølberg, I.S.: Modelling and Control of Underwater Vehicle-Manipulator Systems (1994).
- [17] Ioi, K., Itoh, K.: Modeling and simulation of an underwater manipulator (1990)
- [18] Mahesh, H., Yuh, J., Lakshmi, R.: A coordinated control of an underwater vehicle and robotic manipulator, Journal of Robotics Systems (1991).
- [19] Janocha, H., Papadimitriou, I.: Simulation of the dynamic behavior of robots in an extreme environment (1991).
- [20] Fossen, T.: Guidance and Control of Ocean Vehicles, John Wiley & Sons Ltd. (1994).
- [21] Marín-Cañas, O.: Diseño de un controlador para la realización de maniobras automáticas en un ROV sumergible, Trabajo Fin de Grado, Universidad de Huelva (2022).

## Anexo I: Código

En este anexo se muestran los códigos utilizados en la simulación diseñada en Matlab.

### A. Ecuación de estado del ROV y representación de empujes

```
function salida=kuta(t,Y,h,Empujes_local)

Empujes_global=animacion(Y,Empujes_local);

salida=zeros(size(Y));%la salida debe ser un vector columna

k1=h*feval('modelo_ROV',t,Y,Empujes_global);
k2=h*feval('modelo_ROV',t+h/2,Y+(1*k1/2),Empujes_global);
k3=h*feval('modelo_ROV',t+h/2,Y+(1*k2/2),Empujes_global);
k4=h*feval('modelo_ROV',t+h,Y+(1*k3),Empujes_global);
salida=Y+(k1+2*k2+2*k3+k4)/6;

end

function S=modelo_ROV(t,Estado,Entradas)

%Valores de las componentes de las fuerzas y el par aplicado
Fx=Entradas(1);
Fy=Entradas(2);
Tau=Entradas(3);

%Parámetros inerciales
global nu %rozamiento desplazamiento longitudinal
global lambda %rozamiento desplazamiento lateral
global chi %rozamiento rotación
global m
global I

%Valores del estado
S=zeros(size(Estado));
x=Estado(1,1);
Vx=Estado(2,1);
y=Estado(3,1);
Vy=Estado(4,1);
teta=Estado(5,1);
w=Estado(6,1);

%Modelo dinámico
S(1,1)=Vx;
S(2,1)=(Fx/m)-((nu/m)*Vx);
S(3,1)=Vy;
S(4,1)=(Fy/m)-((lambda/m)*Vy);
S(5,1)=w;
S(6,1)=(Tau/I)-((chi/I)*w);
```

end

```
function Empujes_global=animacion(estado,empujes_local)

    global pose
    global punto

    % Dibujo de los elementos

    cla;

    Empujes_global=ROV(estado, empujes_local);

    % pose(1,:)->x, pose(2,:)->Vx,
    % pose(3,:)->y, pose(4,:)->Vy,
    % pose(5,:)->theta, pose(6,:)->w,

    %Piscina
    x=[0,300]; %cm
    y=[200,200];
    area(x, y, 'LineStyle','none', 'FaceColor', [0.8010, 0.9450, 1],
    'FaceAlpha', 0.5, 'ShowBaseLine','off');
    hold on
    x1=0;
    x2=300;
    y1=0;
    y2=200;
    x = [x1, x2, x2, x1, x1];
    y = [y1, y1, y2, y2, y1];
    plot(x, y, '-', 'Color', [0.2, 0.9450, 01], 'LineWidth', 2);
    xlim([-10,310]);
    ylim([-10,210]);
    xlabel("cm");
    ylabel("cm");
    plot(punto(1),punto(2),'*r');
    plot(pose(1,:),pose(3:),'k');
    drawnow;
```

end

```
function Empujes_global=ROV(pose, empujes_local)
```

```
    % Dibuja el ROV

    x=pose(1);
    y=pose(3);
    theta=pose(5);
    Vx=pose(2);
    Vy=pose(4);
    w=pose(6);

    %Empuje de los rotores
    empuje1=empujes_local(1);
    empuje2=empujes_local(2);
    empuje3=empujes_local(3);
    empuje4=empujes_local(4);
    %Robot
    largo=50; %cm
```

```
ancho=38; %cm

p1=[largo/2 ancho/2 0 1]';
p2=[largo/2 -ancho/2 0 1]';
p3=[-largo/2 -ancho/2 0 1]';
p4=[-largo/2 ancho/2 0 1]';

robot=[p1 p2 p3 p4 p1];

%Rotor básico
radio_ruedas=4.7; %cm

p1=[radio_ruedas 0 0 1]';
p2=[-radio_ruedas 0 0 1]';

rotor_basico=[p1 p2];

%Matriz de transformación

Transformacion_1=Desplazamiento(x,y,0)*Rotacionz(theta);

% Transformaciones de los rotores en el local del robot

dist_ruedas_largo=40; %cm
dist_ruedas_ancho=22; %cm

T_rotor_d_d=Desplazamiento(dist_ruedas_largo/2,-
dist_ruedas_ancho/2,0)*Rotacionz(-pi/4);

T_rotor_i_d=Desplazamiento(dist_ruedas_largo/2,dist_ruedas_ancho/2,0)*
Rotacionz(5*pi/4);

T_rotor_d_t=Desplazamiento(-dist_ruedas_largo/2,-
dist_ruedas_ancho/2,0)*Rotacionz(5*pi/4);
T_rotor_i_t=Desplazamiento(-
dist_ruedas_largo/2,dist_ruedas_ancho/2,0)*Rotacionz(-pi/4);

% Transformaciones de los rotores en el global

T_rotor_d_d_g=Transfromacion_1*T_rotor_d_d;
T_rotor_i_d_g=Transfromacion_1*T_rotor_i_d;

T_rotor_d_t_g=Transfromacion_1*T_rotor_d_t;
T_rotor_i_t_g=Transfromacion_1*T_rotor_i_t;

%Configuración del robot y rotores en el global

robot_0=Transfromacion_1*robot;

rotor_d_d=T_rotor_d_d_g*rotor_basico;
rotor_i_d=T_rotor_i_d_g*rotor_basico;
rotor_d_t=T_rotor_d_t_g*rotor_basico;
rotor_i_t=T_rotor_i_t_g*rotor_basico;

% Dibujo de los elementos
```

```
plot(robot_0(1,:),robot_0(2,:));
hold on

plot(rotor_d_d(1,:),rotor_d_d(2,:));

plot(rotor_i_d(1,:),rotor_i_d(2,:));

plot(rotor_d_t(1,:),rotor_d_t(2,:));

plot(rotor_i_t(1,:),rotor_i_t(2,:));

% Dibuja los vectores de empuje

k1=0.00001;
k2=20;
k3=10;

%empuje rotor 1
%empuje en el global
    E1_g=T_rotor_d_d_g(1:3,1:3)*[0 empuje1 0]';
%dibujo el vector
    quiver(T_rotor_d_d_g(1,4),T_rotor_d_d_g(2,4),E1_g(1)/k3+Vx*k1+w*
    k2,E1_g(2)/k3+Vx*k1+w*k2);

%empuje rotor 2
%empuje en el global
    E2_g=T_rotor_i_d_g(1:3,1:3)*[0 empuje2 0]';
%dibujo el vector

    quiver(T_rotor_i_d_g(1,4),T_rotor_i_d_g(2,4),E2_g(1)/k3+Vx*k1+w*
    k2,E2_g(2)/k3+Vx*k1+w*k2);

%empuje rotor 3
%empuje en el global
    E3_g=T_rotor_d_t_g(1:3,1:3)*[0 empuje3 0]';
    %dibujo el vector
    quiver(T_rotor_d_t_g(1,4),T_rotor_d_t_g(2,4),E3_g(1)/k3+Vx*k1+w*
    k2,E3_g(2)/k3+Vx*k1+w*k2);

%empuje rotor 4

%empuje en el global
    E4_g=T_rotor_i_t_g(1:3,1:3)*[0 empuje4 0]';
%dibujo el vector
    quiver(T_rotor_i_t_g(1,4),T_rotor_i_t_g(2,4),E4_g(1)/k3+Vx*k1+w*
    k2,E4_g(2)/k3+Vx*k1+w*k2);

% CALCULOS EMPUJE Y PAR EN EL GLOBAL

% Calculo de la fuerza total en el sistema de referencia global

Fuerza_x= E1_g(1)+E2_g(1)+E3_g(1)+E4_g(1);
Fuerza_y= E1_g(2)+E2_g(2)+E3_g(2)+E4_g(2);

% Cálculo de los momentos en el local del robot
```



```
%empuje rotor 1 en el local
    E1_l=T_rotor_d_d(1:3,1:3)*[0 empuje1 0]';
%momento respecto al CG
    Tau1=cross(T_rotor_d_d(1:3,4),E1_l);

%empuje rotor 2 en el local
    E2_l=T_rotor_i_d(1:3,1:3)*[0 empuje2 0]';
%momento respecto al CG
    Tau2=cross(T_rotor_i_d(1:3,4),E2_l);

%empuje rotor 3 en el local
    E3_l=T_rotor_d_t(1:3,1:3)*[0 empuje3 0]';
%momento respecto al CG
    Tau3=cross(T_rotor_d_t(1:3,4),E3_l);

%empuje rotor 4 en el local
    E4_l=T_rotor_i_t(1:3,1:3)*[0 empuje4 0]';
%momento respecto al CG
    Tau4=cross(T_rotor_i_t(1:3,4),E4_l);

Tau=Tau1(3)+Tau2(3)+Tau3(3)+Tau4(3);

%empuje total en el local
ET_l=E1_l+E2_l+E3_l+E4_l;

axis([-10 90 -10 90])

Empujes_global=[Fuerza_x Fuerza_y Tau];

end

function D=Desplazamiento(x0,y0,z0)
    D=[1 0 0 x0 ; 0 1 0 y0 ; 0 0 1 z0; 0 0 0 1];
end

function Rz=Rotacionz(theta)
    Rz=[cos(theta) -sin(theta) 0 0;sin(theta) cos(theta) 0 0; 0 0 1 0 ; 0
    0 0 1];
end

end
```

## B. Sonar Ping360

```
function salida=salida_laser3(u, n, MAPA, n_puntos)

    px=u(1);
    py=u(2);
    orientacion=u(3);

    global Sector
    global i
    sec=Sector*2*pi/360/2; %Sector/2 en rad
    incremento=(sec)/n;

    [distancias, intensidad]=medida(px,py,orientacion+sec-(i-
    1)*incremento,MAPA);

    %generamos un ruido de la dimensión del vector con las medidas
    ruido=randn([1,n_puntos])/10;
    %intensidad=intensidad+ruido;
    salida=[transpose(distancias) transpose(intensidad)];
end

function [d, valor] = medida(px,py,orientacion,MAPA)
    global dmax
    global n_puntos
    global Width
    global Height
    d = zeros(1, n_puntos);
    valor = zeros(1, n_puntos);
    incremento=dmax/n_puntos;

    for k=1:n_puntos
        d(k)=incremento*(k-1);
        n = round(px+d(k)*cos(orientacion));
        m = round(py+d(k)*sin(orientacion));
        if (n < 0) | (m < 0) | (n >= Width) | (m >= Height) |(d >dmax)
            valor(k)=0;
        else
            valor(k)=MAPA(m+1, n+1);
        end
    end
end

end
```

## C. Representar haces del sonar Ping360

```
sec=Sector*2*pi/360/2; %Sector/2 en rad

for km=i:i      %1:2*nh+1

    phi=sec-(sec/nh)*(km);

    haz_local=[0 sonar(km)*cos(phi); 0 sonar(km)*sin(phi); 0 0; 1 1];

    % Transformación del robot

    Transformacion_1=Desplazamiento(pose(1,k),pose(3,k),0)*Rotacionz(pose(
5,k));

    %Haces del sonar situadas en el global
    haz_global=Transformacion_1*haz_local;

    %Dibuja la representación del haz
    addpoints(haces_sonar(km),haz_global(1,:),haz_global(2,:));

    %Borra la anterior representación del haz
    if km==1
        clearpoints(haces_sonar(2*nh));
        pause(0.01);
    elseif km== 2*nh
        clearpoints(haces_sonar(km-1));
        pause(0.01);
        clearpoints(haces_sonar(km));
    else
        clearpoints(haces_sonar(km-1));
        pause(0.01);
    end

    %Dibuja el muestreo
    nexttile(2);
    angle=-Sector*400/360/2+(i-1);
    distancia_puntos=dmax/n_puntos;
    puntos_vector=1:n_puntos;
    Paso_motor_rad=0.9*pi/180;

    x = distancia_puntos*puntos_vector*sin(angle*Paso_motor_rad);
    y = distancia_puntos*puntos_vector*cos(angle*Paso_motor_rad);
    scatter(transpose(x),transpose(y), 50, sonar_1(:, 2),'.');
    drawnow

end
```



## D. Simulación del movimiento realizando transectos

```
clear all
clc

%Sector de muestreo
    global Sector
%Distancia máxima
    global dmax %cm
    dmax=150;
%Número de haces (inicial)
    nh=round(360*(400/360)*(1/2));

global pose
global punto
global nu %rozamiento desplazamiento
global lambda
global chi %rozamiento rotación
global m
global I

f1=crea_sonar(dmax);
nexttile(1);

%Cargamos el mapa y lo adecuamos para representarlo adecuadamente

MAPA = imread('.\piscina10.bmp');
%Transformación para colocar correctamente el origen del Sistema de
%Referencia
MAPA(1:end, :, :)=MAPA(end:-1:1, :, :);
image(MAPA);
axis xy
global Width
global Height
[Height, Width]=size(MAPA);

%-----
%Definición de las variables animatedline para la representación de los
%haces del sonar
%-----

global i
i=1;
global n_puntos
n_puntos=1200; %En el ping son 1200

for km=1:2*nh
    haces_sonar(km)=animatedline('Color',[1 0 0]);
end

nu=49.9959;
lambda=89.0112;
chi=15;
m=16; %kg
I=0.5633; %kg*m^2

%Condiciones iniciales
pose0=[250; 0; 120 ; 0; pi/2; 0];
```



```
objetivo=[0 0];

t0=0;

%final de la simulación
tf=120;
%paso de integración
h=0.1;
%vector tiempo
t=0:h:tf*10;
%índice de la matriz
k=1;

%inicialización valores iniciales
pose(:,k)=pose0;
t(k)=t0;

%-----
%Sistema de Referencia del ROV del que dependerán todos los demás elementos
%de la animación
%-----
SRF_ROV = hgtransform;
%-----

%% Pintar ROV
objetivo=[pose0(1) pose0(3)];

va(k)=0;
vl(k)=0;
w(k)=0;

%Configuración de los empujes de los rotores
E1(k)=va(k)+w(k)+vl(k); E2(k)=va(k)-w(k)-vl(k);
E3(k)=va(k)+w(k)-vl(k); E4(k)=va(k)-w(k)+vl(k);
Empujes_local= [E1(k) E2(k) E3(k) E4(k)];

pose(:,k+1)=kuta_ROV_new(t(k),pose(:,k),h,Empujes_local);

d_ant=dmax;

%%
while (t0+h*k) < tf

    %% Muestrear 30º
    Sector=30;
    nh=round(Sector*(400/360)*(1/2));
    Sector_1=round(Sector*400/360);

    %Simulación de la medida del sonar en el sistema de referencia del ROV
    sonar=simulacion_sonar(Sector);

    matriz_datos=zeros(n_puntos, Sector_1);
    if d_ant~=dmax
        nexttile(2);
        cla;
        texto(dmax);
        pause(1);
        d_ant=dmax;
        nexttile(1);
    end
end
```



```
end

for n=1:2*nh

    %Obtener datos de sonar
    sonar_1=salida_laser3([pose(1,k) pose(3,k) pose(5,k)], nh, MAPA,
        n_puntos);

    %Añadimos el ruido del ROV
    d_rov=round(n_puntos/dmax*30);
    for j=1:1
        sonar_1(j, 2)=0;
    end
    for j=2:d_rov
        sonar_1(j, 2)=255;
    end

    %Guardamos matriz completa
    matriz_datos(:, i)=sonar_1(:,2);

    %Script donde se simula y representa la medida del sonar
    medir_pintar_sonar;

    i=i+1;
end
datos=transpose(matriz_datos);
i=1;

%% Filtrar e identificar objetos

conn=4; % conectividad vertical-horizontal (4) o en asterisco (8)
umbral_conectividad=20; % minimo nº pixeles para ser componente
distancia_limite=1; % si el objeto está a una distancia mayor que este
valor, se hace filtro estricto
lim_inf_intensidad=50; %por debajo de este valor se considera agua
(primerá mitad de la muestra)
lim_min_cant_pez=50;

distancia_puntos=dmax/n_puntos;
Paso_motor_rad=0.9*pi/180;

[N_objetos,ID_objetos,centroids_r]=IdentificacionYFiltro(datos,conn,umbral_co
nectividad,d_rov,lim_inf_intensidad,lim_min_cant_pez,distancia_limite,distanc
ia_puntos,Paso_motor_rad);

if N_objetos>0
    for r=1:N_objetos
        x_cent = distancia_puntos*centroids_r(ID_objetos(r,1),1)*sin
            ((centroids_r(ID_objetos(r,1),2)-Sector*400/360/2)*
            Paso_motor_rad);
        y_cent = distancia_puntos*centroids_r(ID_objetos(r,1),1)*cos
            ((centroids_r(ID_objetos(r,1),2)-Sector*400/360/2)*
            Paso_motor_rad);

        Centroide_objetos(r)=scatter(x_cent,y_cent,50,'w','s','filled');
        if ID_objetos(1,1) ~=1
            for s=1:N_objetos
                ID_objetos(s, 1)=s;
            end
        end
    end
end
```



```
end
text(x_cent,y_cent,num2str(ID_objetos(r, 1)), 'Color', [1 0 1],
'FontSize', 14, 'FontWeight','bold');

drawnow
end
objeto=1;
x_cent = distancia_puntos*centroids_r(ID_objetos(objeto,1),1)*sin(
(centroids_r(ID_objetos(objeto,1),2)-Sector*400/360/2)*
Paso_motor_rad);
y_cent = distancia_puntos*centroids_r(ID_objetos(objeto,1),1)*cos
((centroids_r(ID_objetos(objeto,1),2)-Sector*400/360/2)*
Paso_motor_rad);

if x_cent>0 && y_cent>0
    alpha=-(90-atand(y_cent/x_cent));
elseif x_cent<0 && y_cent>0
    alpha=(atand(y_cent/x_cent)+90);
end

objetivo(1)=pose0(1)+y_cent*cos(pose0(5))+x_cent*sin(pose0(5));
objetivo(2)=pose0(3)-x_cent*cos(pose0(5))+y_cent*sin(pose0(5));
else
disp("Objetos no detectados");
%Aumentar distancia de muestreo
dmax=dmax+50;
end

if N_objetos > 0

%% Muestrear sector y avanzar

nexttile(2);

while (abs(objetivo(2)-pose(3, k))>80 || abs(objetivo(1)-pose(1,
k))>80) && (t0+h*k) < tf && N_objetos>0
matriz_datos=zeros(n_puntos, Sector_1);
if d_ant~=dmax
nexttile(2);
cla;
texto(dmax);
d_ant=dmax;
nexttile(1);
end
for n=1:2*nh

%Obtener datos de sonar
sonar_1=salida_laser3([pose(1,k) pose(3,k) pose(5,k)], nh,
MAPA, n_puntos);

%Añadimos el ruido del ROV
d_rov=round(n_puntos/dmax*30);
for j=1:1
sonar_1(j, 2)=0;
end
for j=2:d_rov
sonar_1(j, 2)=255;
end
```



```
matriz_datos(:, i)=sonar_1(:,2);

nexttile(1);
medir_pintar_sonar;
i=i+1;

punto=[objetivo(1) objetivo(2)];

va(k)=2*sqrt((objetivo(1)-pose(1,k))^2+(objetivo(2)-
pose(3,k))^2);
vl(k)=0;

%Controlador de rumbo
%matriz del Local del Rov al global
T=[cos(pose(5,k)) -sin(pose(5,k)) pose(1,k);
   sin(pose(5,k))  cos(pose(5,k)) pose(3,k);
   0                0                1    ];
%coordenadas del objetivo en el local
target=T^(-1)*[objetivo(1) objetivo(2) 1]';

%Angulo de dirección del objetivo en el local del ROV
theta_e(k)=atan2(target(2), target(1));

%Velocidad angular para alcanzar el objetivo
w(k)=1/20*theta_e(k);

%Configuración de los empujes de los rotores
E1(k)=va(k)+w(k)+vl(k); E2(k)=va(k)-w(k)-vl(k);
E3(k)=va(k)+w(k)-vl(k); E4(k)=va(k)-w(k)+vl(k);

Empujes_local= [E1(k) E2(k) E3(k) E4(k)];

pose(:,k+1)=kuta_ROV_new(t(k),pose(:,k),h,Empujes_local);

%actualización
k=k+1;
end
datos=transpose(matriz_datos);
i=1;

if (abs(objetivo(2)-pose(3, k))>20 || abs(objetivo(1)-pose(1,
k))>20)

% Volver a calcular objetivo

conn=4;
umbral_conectividad=20; e
distancia_limite=1;
lim_inf_intensidad=50;
lim_min_cant_pez=50;

distancia_puntos=dmax/n_puntos;
Paso_motor_rad=0.9*pi/180;

[N_objetos,ID_objetos,centroids_r]=IdentificacionYFiltro(datos,conn,umbral_co
nectividad,d_rov,lim_inf_intensidad,lim_min_cant_pez,distancia_limite,distanc
ia_puntos,Paso_motor_rad);
```

```
        if N_objetos>0
            for r=1:1
                x_cent =
distancia_puntos*centroids_r(ID_objetos(r,1),1)*sin((centroids_r(ID_objetos(r
,1),2)-Sector*400/360/2)*Paso_motor_rad);
                y_cent =
distancia_puntos*centroids_r(ID_objetos(r,1),1)*cos((centroids_r(ID_objetos(r
,1),2)-Sector*400/360/2)*Paso_motor_rad);

Centroide_objetos(r)=scatter(x_cent,y_cent,50,'w','s','filled');
                text(x_cent,y_cent,num2str(ID_objetos(r, 1)),
'Color', [1 0 1], 'FontSize', 14, 'FontWeight','bold');
                drawnow
            end
            objeto=1;
            x_cent =
distancia_puntos*centroids_r(ID_objetos(objeto,1),1)*sin((centroids_r(ID_obje
tos(objeto,1),2)-Sector*400/360/2)*Paso_motor_rad);
            y_cent =
distancia_puntos*centroids_r(ID_objetos(objeto,1),1)*cos((centroids_r(ID_obje
tos(objeto,1),2)-Sector*400/360/2)*Paso_motor_rad);
            objetivo(1)=pose(1, k)+y_cent*cos(pose(5,
k))+x_cent*sin(pose(5, k));
            objetivo(2)=pose(3, k)-x_cent*cos(pose(5,
k))+y_cent*sin(pose(5, k));
        else
            disp("Objetos no detectados");
            N_objetos=0;
        end
        dmax=round(sqrt(x_cent^2+y_cent^2))+20;
    end
end

%% Girar 180º
pause(1);
nexttile(2);

hdg_final=pose(5,k)+pi;
dif(k)=hdg_final-pose(5,k);

while abs(dif(k)) > 0.05 && (t0+h*k) < tf

    nexttile(1);

    va(k)=0;
    vl(k)=0;
    w(k)=1/20*dif(k);

    %%Configuración de los empujes de los rotores
    E1(k)=va(k)+w(k)+vl(k); E2(k)=va(k)-w(k)-vl(k);
    E3(k)=va(k)+w(k)-vl(k); E4(k)=va(k)-w(k)+vl(k);

    Empujes_local= [E1(k) E2(k) E3(k) E4(k)];

    pose(:,k+1)=kuta_ROV_new(t(k),pose(:,k),h,Empujes_local);
    %actualización
    k=k+1;
```



```
        dif(k)=hdg_final-pose(5,k);
    end
    pose0=pose(:, k);

end

end

%hold off
grid on

%% Funciones

function sonar=simulacion_sonar(Sector)
    global dmax
    Sector_1=round(Sector*400/360);
    sonar=zeros(Sector_1,2);
    sonar(:,1)=dmax;
    v_inicial=round(-Sector/2);
    for j=1:Sector_1
        sonar(j,2)=v_inicial;
        v_inicial=v_inicial+0.9;
    end
end

function []=texto(distance)
    axis ([-distance-distance/6 distance+distance/6 -distance-distance/6
distance+distance/6]);
    axis off
    text(0, distance+distance/20, '0', 'HorizontalAlignment', 'center');
    text(distance/1.5+distance/10, distance/1.5+distance/10, '50',
'HorizontalAlignment', 'center');
    text(distance+distance/12, 0, '100', 'HorizontalAlignment', 'center');
    text(distance/1.5+distance/10, -(distance/1.5+distance/10), '150',
'HorizontalAlignment', 'center');
    text(0, -(distance+distance/20), '200', 'HorizontalAlignment', 'center');
    text(-(distance/1.5+distance/10), -(distance/1.5+distance/10), '250',
'HorizontalAlignment', 'center');
    text(-(distance+distance/12), 0, '300', 'HorizontalAlignment', 'center');
    text(-(distance/1.5+distance/10), distance/1.5+distance/10, '350',
'HorizontalAlignment', 'center');
    drawnow;
end
```

## E. Simulación del movimiento de navegación hacia objeto

```
clear all
clc

%Sector de muestreo
global Sector
%Distancia máxima en cm
global dmax
dmax=150;
%Número de haces (inicial)
nh=round(360*(400/360)*(1/2));

global pose
global punto
global lambda %rozamiento desplazamiento longitudinal
global nu %rozamiento desplazamiento lateral
global chi %rozamiento rotación
global m
global I

f1=crea_sonar(dmax);
nexttile(1);

%Cargamos el mapa y lo adecuamos para representarlo adecuadamente

MAPA = imread('.\piscina11.bmp');
%Transformación para colocar correctamente el origen del Sistema de
%Referencia
MAPA(1:end, :, :) = MAPA(end:-1:1, :, :);
image(MAPA);
axis xy
global Width
global Height
[Height, Width] = size(MAPA);

global i
i=1;
global n_puntos
n_puntos=1200; %En el ping son 1200

%-----
%Definición de las variables animatedline para la representación de los
%haces del sonar
%-----

for km=1:2*nh
    haces_sonar(km)=animatedline('Color',[1 0 0]);
end

nu=49.9959;
lambda=89.0112;
chi=15;
m=16; %kg
I=0.5633; %kg*m^2

%Condiciones iniciales
pose0=[200; 0; 150; 0; pi/2; 0];
objetivo=[0 0];
```



```
t0=0;

%final de la simulación
tf=100;
%paso de integración
h=0.1;
%vector tiempo
t=0:h:tf*10;
%índice de la matriz
k=1;

%inicialización valores iniciales
pose(:,k)=pose0;
t(k)=t0;

%-----
%Sistema de Referencia del ROV del que dependerán todos los demás elementos
%de la animación
%-----
SRF_ROV = hgtransform;
%-----

%% Pintar ROV
objetivo=[pose0(1) pose0(3)];

va(k)=0;
vl(k)=0;
w(k)=0;

%Configuración de los empujes de los rotores
E1(k)=va(k)+w(k)+vl(k); E2(k)=va(k)-w(k)-vl(k);
E3(k)=va(k)+w(k)-vl(k); E4(k)=va(k)-w(k)+vl(k);
Empujes_local= [E1(k) E2(k) E3(k) E4(k)];

pose(:,k+1)=kuta_ROV_new(t(k),pose(:,k),h,Empujes_local);

%pause(10)

%%
while (t0+h*k) < tf

    %% Muestrear 360º
    Sector=360;
    nh=round(Sector*(400/360)*(1/2));
    Sector_1=round(Sector*400/360);

    %Simulación de la medida del sonar en el sistema de referencia del ROV
    sonar=simulacion_sonar(Sector);

    matriz_datos=zeros(n_puntos, Sector_1);

    for n=1:2*nh

        %Obtener datos de sonar
        sonar_1=salida_laser3([pose(1,k) pose(3,k) pose(5,k)], nh, MAPA,
n_puntos);

        %Añadimos el ruido del ROV
        d_rov=round(n_puntos/dmax*25);
```

```
for j=1:1
    sonar_1(j, 2)=0;
end
for j=2:d_rov
    sonar_1(j, 2)=255;
end

%Guardamos matriz completa
matriz_datos(:, i)=sonar_1(:,2);

%Script donde se simula y representa la medida del sonar
medir_pintar_sonar;

i=i+1;
end
datos=transpose(matriz_datos);
i=1;

%% Seleccionar objetivo

%Filtrar e identificar objetos

conn=4; % conectividad vertical-horizontal (4) o en asterisco (8)
umbral_conectividad=20; %mínimo nº píxeles para ser componente
%lim_cant_datos=120000; %a partir de este límite se considera ruido del
ROV
distancia_limite=1; %si el objeto está a una distancia mayor que este
valor, se hace filtro estricto
lim_inf_intensidad=50; %por debajo de este valor se considera agua
(primerá mitad de la muestra)
lim_min_cant_pez=50;

distancia_puntos=dmax/n_puntos;
Paso_motor_rad=0.9*pi/180;

[N_objetos,ID_objetos,centroids_r]=IdentificacionYFiltro(datos,conn,umbral_co
nectividad,d_rov,lim_inf_intensidad,lim_min_cant_pez,distancia_limite,distanc
ia_puntos,Paso_motor_rad);

if N_objetos>0
    for r=1:N_objetos
        x_cent =
distancia_puntos*centroids_r(ID_objetos(r,1),1)*sin((centroids_r(ID_objetos(r
,1),2)-Sector*400/360/2)*Paso_motor_rad);
        y_cent =
distancia_puntos*centroids_r(ID_objetos(r,1),1)*cos((centroids_r(ID_objetos(r
,1),2)-Sector*400/360/2)*Paso_motor_rad);

        Centroide_objetos(r)=scatter(x_cent,y_cent,50,'w','s','filled');
        if ID_objetos(1,1) ~=1
            for s=1:N_objetos
                ID_objetos(s, 1)=s;
            end
        end
        text(x_cent,y_cent,num2str(ID_objetos(r, 1)), 'Color', [1 0 1],
'FontSize', 14, 'FontWeight','bold');

        drawnow
    end
end
```

```
end

objeto=0;
opciones=zeros(1,N_objetos);
for q=1:N_objetos
    opciones(q)=(ID_objetos(q, 1));
end
opciones_str=string(opciones);
op_0='-';
opciones_completo=[op_0 opciones_str];

if k==1
    annotation('textbox', [0.5, 0.08, 0, 0], 'string', 'Selecciona el
objeto a seguir:', ...
    'FitBoxToText','on', 'LineStyle','none', 'FontSize',14);
end
c = uicontrol(f1,'Style','popupmenu');
c.Position = [1000 16 70 30];
c.String = opciones_completo;
drawnow;
while objeto==0
    drawnow;
    if c.Value>1
        objeto=c.Value-1;
    end
end

x_cent =
distancia_puntos*centroids_r(ID_objetos(objeto,1),1)*sin((centroids_r(ID_obje
tos(objeto,1),2)-Sector*400/360/2)*Paso_motor_rad);
y_cent =
distancia_puntos*centroids_r(ID_objetos(objeto,1),1)*cos((centroids_r(ID_obje
tos(objeto,1),2)-Sector*400/360/2)*Paso_motor_rad);

if x_cent>0 && y_cent>0
    alpha=(atand(y_cent/x_cent)-90);
elseif x_cent<0 && y_cent>0
    alpha=(atand(y_cent/x_cent)+90);
elseif x_cent<0 && y_cent<0
    alpha=(atand(y_cent/x_cent)+90);
elseif x_cent>0 && y_cent<0
    alpha=(atand(y_cent/x_cent)-90);
end
else
    disp("Objetos no detectados");
end

if N_objetos > 0

%% Girar ROV
hdg_final=pose(5,k)+alpha*pi/180;
dif(k)=hdg_final-pose(5,k);
objetivo(1)=pose0(1)+y_cent*cos(pose0(5))+x_cent*sin(pose0(5));
objetivo(2)=pose0(3)-x_cent*cos(pose0(5))+y_cent*sin(pose0(5));

while abs(dif(k)) > 0.05 && (t0+h*k) < tf

    nexttile(1);
```



```
va(k)=0;
v1(k)=0;
w(k)=1/10*dif(k);

%Configuración de los empujes de los rotores
E1(k)=va(k)+w(k)+v1(k); E2(k)=va(k)-w(k)-v1(k);
E3(k)=va(k)+w(k)-v1(k); E4(k)=va(k)-w(k)+v1(k);

Empujes_local= [E1(k) E2(k) E3(k) E4(k)];

pose(:,k+1)=kuta_ROV_new(t(k),pose(:,k),h,Empujes_local);

%actualización
k=k+1;

dif(k)=hdg_final-pose(5,k);
end
pose0=pose(:, k);

%% Muestrear sector y avanzar

nexttile(2);
cla();
texto(dmax);

Sector=30;
nh=round(Sector*(400/360)*(1/2));
Sector_1=round(Sector*400/360);
%Simulación de la medida del sonar en el sistema de referencia del
ROV
sonar=simulacion_sonar(Sector);

while (abs(objetivo(2)-pose(3, k))>35 || abs(objetivo(1)-pose(1,
k))>35) && (t0+h*k) < tf && N_objetos>0
    matriz_datos=zeros(n_puntos, Sector_1);
    for n=1:2*nh

        %Obtener datos de sonar
        sonar_1=salida_laser3([pose(1,k) pose(3,k) pose(5,k)], nh,
MAPA, n_puntos);

        %Añadimos el ruido del ROV
        d_rov=round(n_puntos/dmax*30);
        for j=1:1
            sonar_1(j, 2)=0;
        end
        for j=2:d_rov
            sonar_1(j, 2)=255;
        end

        matriz_datos(:, i)=sonar_1(:,2);

        nexttile(1);
        medir_pintar_sonar;
        i=i+1;

        punto=[objetivo(1) objetivo(2)];
```

```
pose(3,k))^2);
va(k)=2*sqrt((objetivo(1)-pose(1,k))^2+(objetivo(2)-
pose(3,k))^2);
vl(k)=0;

%Controlador de rumbo
%matriz del Local del Rov al global
T=[cos(pose(5,k)) -sin(pose(5,k)) pose(1,k);
   sin(pose(5,k))  cos(pose(5,k)) pose(3,k);
   0                0                1    ];
%coordenadas del objetivo en el local
target=T^(-1)*[objetivo(1) objetivo(2) 1]';

%Angulo de dirección del objetivo en el local del ROV
theta_e(k)=atan2(target(2), target(1));

%Velocidad angular para alcanzar el objetivo
w(k)=1/40*theta_e(k);

%Configuración de los empujes de los rotores
E1(k)=va(k)+w(k)+vl(k); E2(k)=va(k)-w(k)-vl(k);
E3(k)=va(k)+w(k)-vl(k); E4(k)=va(k)-w(k)+vl(k);

Empujes_local= [E1(k) E2(k) E3(k) E4(k)];

pose(:,k+1)=kuta_ROV_new(t(k),pose(:,k),h,Empujes_local);

%actualización
k=k+1;
end
datos=transpose(matriz_datos);
i=1;
if (abs(objetivo(2)-pose(3, k))>20 || abs(objetivo(1)-pose(1,
k))>20)

    % Volver a calcular objetivo

    conn=4; % conectividad vertical-horizontal (4) o en asterisco
    umbral_conectividad=20; %mínimo nº pixeles para ser
    componente
    %lim_cant_datos=120000; %a partir de este límite se considera
    ruido del ROV
    distancia_limite=1; %si el objeto está a una distancia mayor
    que este valor, se hace filtro estricto
    lim_inf_intensidad=50; %por debajo de este valor se considera
    agua (primera mitad de la muestra)
    lim_min_cant_pez=50; %

    distancia_puntos=dmax/n_puntos;
    Paso_motor_rad=0.9*pi/180;

    [N_objetos,ID_objetos,centroids_r]=IdentificacionYFiltro(datos,conn,umbral_co
nectividad,d_rov,lim_inf_intensidad,lim_min_cant_pez,distancia_limite,distanc
ia_puntos,Paso_motor_rad);

    if N_objetos>0
        for r=1:1
```

```
        x_cent =
distancia_puntos*centroids_r(ID_objetos(r,1),1)*sin((centroids_r(ID_objetos(r
,1),2)-Sector*400/360/2)*Paso_motor_rad);
        y_cent =
distancia_puntos*centroids_r(ID_objetos(r,1),1)*cos((centroids_r(ID_objetos(r
,1),2)-Sector*400/360/2)*Paso_motor_rad);

Centroide_objetos(r)=scatter(x_cent,y_cent,50,'w','s','filled');
        text(x_cent,y_cent,num2str(ID_objetos(r, 1)),
'Color', [1 0 1], 'FontSize', 14, 'FontWeight','bold');
        drawnow
    end
    objeto=1;
    x_cent =
distancia_puntos*centroids_r(ID_objetos(objeto,1),1)*sin((centroids_r(ID_obje
tos(objeto,1),2)-Sector*400/360/2)*Paso_motor_rad);
    y_cent =
distancia_puntos*centroids_r(ID_objetos(objeto,1),1)*cos((centroids_r(ID_obje
tos(objeto,1),2)-Sector*400/360/2)*Paso_motor_rad);

objetivo(1)=pose0(1)+y_cent*cos(pose0(5))+x_cent*sin(pose0(5));
    objetivo(2)=pose0(3)-
x_cent*cos(pose0(5))+y_cent*sin(pose0(5));
    else
        disp("Objetos no detectados");
        N_objetos=0;
    end
end
end
end
end
nexttile(2);
end

hold off
grid on

%% Funciones

function sonar=simulacion_sonar(Sector)
    global dmax
    Sector_1=round(Sector*400/360);
    sonar=zeros(Sector_1,2);
    sonar(:,1)=dmax;
    v_inicial=round(-Sector/2);
    for j=1:Sector_1
        sonar(j,2)=v_inicial;
        v_inicial=v_inicial+0.9;
    end
end

function []=texto(distancia)
    distance=distancia;
    text(0, distance+distance/20, '0', 'HorizontalAlignment', 'center');
    text(distance/1.5+distance/10, distance/1.5+distance/10, '50',
'HorizontalAlignment', 'center');
    text(distance+distance/12, 0, '100', 'HorizontalAlignment', 'center');
```



```
text(distance/1.5+distance/10, -(distance/1.5+distance/10), '150',  
'HorizontalAlignment', 'center');  
text(0, -(distance+distance/20), '200', 'HorizontalAlignment', 'center');  
text(-(distance/1.5+distance/10), -(distance/1.5+distance/10), '250',  
'HorizontalAlignment', 'center');  
text(-(distance+distance/12), 0, '300', 'HorizontalAlignment', 'center');  
text(-(distance/1.5+distance/10), distance/1.5+distance/10, '350',  
'HorizontalAlignment', 'center');  
drawnow;  
end
```