



**CENTRO DE INVESTIGACIÓN Y DE  
ESTUDIOS AVANZADOS DEL  
INSTITUTO POLITÉCNICO  
NACIONAL**

# **Navegación relativa de un vehículo submarino utilizando visión artificial**

Tesis que presenta

**Ing. Isaac Alejandro García Briones**

Para obtener el grado de

**Maestro en Ciencias**

en

**Sistemas Autónomos de Navegación Aérea y Submarina**

Asesores de Tesis:

Dr. Filiberto Muñoz Palacios

Dr. Sergio Salazar Cruz

Ciudad de México

Agosto 2023.



# Agradecimientos

---

A mis padres y a mi hermano, que siempre me han brindado su apoyo incondicional y me han ayudado a alcanzar mis metas personales y académicas. Todo esto ha sido posible gracias a ustedes.

Mi agradecimiento al Dr. Filiberto Muñoz y al Dr. Sergio Salazar, mis guías principales durante este proceso. Su dedicación como docentes e investigadores ha sido un gran ejemplo que me motivó a crecer en el ámbito académico.

Agradezco a los profesores de la UMI-LAFMIA por transmitirme sus conocimientos y por apoyarme, impulsándome siempre a dar lo mejor de mí.

A Benjamín y Salatiel, les agradezco su compañerismo y apoyo durante mi trabajo de tesis. De ustedes aprendí mucho.

Mi reconocimiento al Dr. Fabrice Le Bars por la excelente labor que realizó al orientarme durante mi estancia de investigación, la cual fue clave para el desarrollo de mi tesis.

A Baptiste y Guillaume, agradezco su compañerismo y ayuda, a pesar del corto tiempo que nos conocemos. Me hicieron sentir como en casa.

Al CINVESTAV, que se convirtió en mi casa de estudios durante dos años y cuya cultura de excelencia siempre llevaré conmigo.

Agradezco a ENSTA Bretagne por recibirme con las puertas abiertas, permitiéndome hacer uso de sus instalaciones para llevar a cabo mi investigación. También agradezco al personal del laboratorio que siempre me brindó apoyo cuando lo necesitaba.

Al CONACYT, por brindarme la oportunidad de continuar mi formación como investigador con el apoyo económico que me otorgaron, sin el cual no hubiera podido continuar mis estudios de posgrado.

A Liz, que estuvo a mi lado durante los momentos más difíciles.



# Resumen

---

En esta tesis se presenta el desarrollo de un sistema de navegación relativa basado en visión artificial para un vehículo autónomo submarino (AUV por sus siglas en inglés: Autonomous Underwater Vehicle). El sistema de navegación relativa presentado aquí tiene dos componentes principales que lo conforman. El primero, es un sistema de visión artificial basado en la estimación de la posición relativa de la cámara con respecto a un marcador de LEDs diseñado para ser observado a cortas distancias. El segundo es un sistema de control neurodifuso de orden superior que no tiene la necesidad de conocer previamente el modelo dinámico del sistema, además de ser un controlador robusto capaz de absorber el error de posicionamiento que tiene el sistema de visión relativa. En la tesis se presenta además del desarrollo del sistema de navegación relativa, resultados experimentales que fueron obtenidos como producto de una estancia de investigación en el laboratorio Lab-STICC de ENSTA Bretagne. Finalmente se presentan en las conclusiones el futuro trabajo tanto para mejorar el desempeño del sistema como para probarlo en otros escenarios prácticos.

# Abstract

---

In this thesis, the development of a relative navigation system based in computer vision for an autonomous underwater vehicle (AUV) is presented. The aforementioned relative navigation system is composed mainly by two principal components. The first one is a computer vision system based in the estimation of the relative position of a camera regarding a LED light marker which was designed to be observed at short distances in both low-light and high-light subaquatic environments. The second, is the development of robust control strategies based on sliding-mode and higher order neurofuzzy models. In this thesis the experimental results of the relative navigation system are showcased, which were performed on the facilities of Lab-STICC of ENSTA Bretagne during a research internship as well as in CINVESTAV's laboratories. Finally the conclusions of this thesis and the future work proposals for improving the performance of the proposed relative navigation system are presented.

# Nomenclatura

---

- **USVs**. Unmanned Surface Vehicles (Vehículos de superficie no tripulados).
- **ROVs**. Remotely Operated Vehicles (Vehículos remotamente operados).
- **UMVs**. Unmanned Maritime Vehicles (Vehículos marítimos no tripulados).
- **UUVs**. Unmanned Underwater Vehicles (Vehículos submarinos no tripulados).
- **AUVs**. Autonomus Underwater Vehicles (Vehículos submarinos autónomos).
- **DGPS**. Differential Global Positioning System (Sistema de posicionamiento global diferencial).
- **GPS** Global Positioning System (Sistema de posicionamiento global)
- **RHONNs** Recurrent High-Order Neural Networks (Redes neuronales recurrentes de alto orden)
- **LED** Light Emitting Diode (Diodo emisor de luz)
- **PTAM** Parallel Tracking and Mapping (Seguimiento y mapeo en paralelo)
- **SLAM** Simultaneous Localization and Mapping (Localización y mapeo simultáneos)
- **ORB-SLAM** Oriented FAST and Rotated BRIEF Simultaneous Localization and Mapping
- **ROS** Robotic Operating System (Sistema operativo para robots)
- **FPS** Frames per Second (Fotogramas por segundo)
- **UDP** User Datagram Protocol (Protocolo de datagramas de usuario)
- **GPU** Graphics Processing Unit (Unidad de procesamiento grafico)
- **HSV** Hue, Saturation, Value (Matiz, saturación y valor)
- **RGB** Red, Green, Blue (Rojo, verde, azul)
- **HD** High Definition (Alta definicion)
- **USB** Universal Serial Bus ( Bus universal en serie)

# Índice general

---

<b>Agradecimientos</b>	<b>I</b>
<b>Resumen</b>	<b>II</b>
<b>Abstract</b>	<b>III</b>
<b>Nomenclatura</b>	<b>IV</b>
<b>Índice de tablas</b>	<b>VII</b>
<b>Índice de figuras</b>	<b>VIII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes y motivación . . . . .	1
1.2. Planteamiento del problema . . . . .	3
1.3. Justificación . . . . .	4
1.4. Estado del arte . . . . .	4
1.5. Objetivos . . . . .	11
1.5.1. Objetivo general . . . . .	12
1.5.2. Objetivos particulares . . . . .	12
1.6. Metodología . . . . .	12
1.7. Publicaciones y estancias . . . . .	14
1.8. Estructura de la tesis . . . . .	15
<b>2. Modelado matemático</b>	<b>16</b>
2.1. Notaciones . . . . .	16
2.2. Modelo cinemático . . . . .	19
2.2.1. Marcos de referencia . . . . .	19
2.2.2. Transformación entre marcos de referencia . . . . .	20
2.3. Modelo dinámico . . . . .	23
2.4. Conclusiones . . . . .	30
<b>3. Diseño del control con RHONNs</b>	<b>31</b>
3.1. Identificación utilizando RHONNs . . . . .	31
3.1.1. Estructura de las RHONNs . . . . .	32
3.1.2. Propiedades de aproximación de la RHONN . . . . .	33
3.1.3. Entrenamiento basado en el algoritmo de gradiente descendente . . . . .	34

3.2.	Identificación y control control utilizando FRHONNs . . . . .	34
3.2.1.	Estructura de las FRHONNs . . . . .	35
3.2.2.	Estrategia de control backstepping utilizando FRHONNs . . . . .	37
3.3.	Algoritmo Super-Twisting . . . . .	41
3.4.	Conclusiones . . . . .	43
<b>4.</b>	<b>Desarrollo del método de estimación basado en visión</b>	<b>44</b>
4.1.	Descripción de la cámara monocular . . . . .	45
4.1.1.	Requisitos del sistema de navegación relativa . . . . .	46
4.1.2.	Calibración de la cámara . . . . .	48
4.2.	Descripción del método propuesto . . . . .	49
4.3.	Descripción del algoritmo de visión . . . . .	52
4.3.1.	Captura del stream de vídeo . . . . .	52
4.3.2.	Lectura de imagen . . . . .	53
4.3.3.	Conversión a espacio de color HSV . . . . .	53
4.3.4.	Segmentación . . . . .	54
4.3.5.	Detección de contornos . . . . .	54
4.3.6.	Filtrado . . . . .	55
4.3.7.	Detección de color . . . . .	55
4.3.8.	Registro de luces candidatas . . . . .	56
4.3.9.	Comprobación de la disposición geométrica . . . . .	56
4.3.10.	Cálculo de la distancia y estimación del estado . . . . .	57
4.3.11.	Publicación en ROS . . . . .	58
4.4.	Conclusiones . . . . .	58
<b>5.</b>	<b>Resultados</b>	<b>59</b>
5.1.	Resultados de simulación para las FRHONNs . . . . .	59
5.2.	Pruebas Experimentales . . . . .	63
5.2.1.	Pruebas de punto fijo . . . . .	64
5.2.2.	Pruebas de seguimiento de trayectoria . . . . .	65
<b>6.</b>	<b>Conclusiones y Trabajo a futuro</b>	<b>67</b>
6.1.	Conclusión . . . . .	67
6.2.	Trabajo a futuro . . . . .	69

# Índice de tablas

---

2.1. Notación SNAME para vehículos marinos (SNAME 1950) . . . . .	17
2.2. Grados de libertad del BlueROV2 . . . . .	19
5.1. Parámetros físicos y dinámicos del BlueROV2. . . . .	60
5.2. Trayectoria deseada para la validación del algoritmo de identificación . . .	60



# Índice de figuras

---

1.1.	Vehículo submarino BlueROV2 de la empresa Blue Robotics <sup>1</sup> . . . . .	1
1.2.	Modulo GPS submarino de la empresa Celurean Sonar <sup>2</sup> . . . . .	4
1.3.	ROV realizando tareas de detección y navegación usando visión artificial <sup>3</sup> .	5
1.4.	Adquisición de características y descripciones de imagen mediante algoritmo ORB <sup>4</sup> . . . . .	6
1.5.	Vehículo submarino navegando en una alberca con obstáculos . . . . .	7
1.6.	Diagrama de funcionamiento para la implementación de <i>Light Beacons</i> <sup>5</sup> . .	8
1.7.	Efecto de deriva por consecuencia de corrientes oceánicas <sup>6</sup> . . . . .	8
1.8.	Intercambio de información para estimación de posición entre vehículos . .	9
1.9.	Disposición de marcadores activos, en un ambiente simulado de oscuridad <sup>7</sup>	10
1.10.	Medición de marcadores con respecto de la cámara <sup>8</sup> . . . . .	10
1.11.	Seguimiento de un vehículo submarino <i>líder</i> equipado por un sensor de precisión DVL, por los <i>seguidores</i> <sup>9</sup> . . . . .	11
1.12.	Vehículo submarino BlueROV2 disponible en el Laboratorio UMI-LAFMIA	13
2.1.	Sistema referencial NED (fuente: BasicAirData) . . . . .	20
2.2.	Sistema referencial del cuerpo (créditos a: T.I.Fossen) . . . . .	20
2.3.	Marco referencial del cuerpo $O_B$ e inercial $O_I$ . . . . .	21
4.1.	Cámara a bordo del vehículo BlueROV2 . . . . .	45
4.2.	Se muestra una imagen obtenida de la cámara monocular del Bluerov a diferentes niveles de iluminación . . . . .	46
4.3.	Difracción en cambios de medio . . . . .	48
4.4.	Marcador de <b>LEDs</b> propuesto para usar como referencia en la estimación de la posición relativa. Los <b>LEDs</b> están conectados al <b>AUV</b> submarino OctoSub	49
4.5.	Diagrama de la estimación de la distancia $L^{CO}$ que hay entre los marcos de referencia de la cámara $C$ y del marcador $O$ . . . . .	50
4.6.	Imagen nativa obtenida de la cámara monocular a bordo del BlueROV2 . .	53
4.7.	Imagen en el espacio de color HSV. . . . .	53
4.8.	Máscara usada en la segmentación de las fuentes de luz blanca en la imagen.	54
4.9.	Fuentes de luz detectadas en la imagen con la técnica de detección de contornos. . . . .	54
4.10.	Máscaras usadas para detectar el color RGB dominante en la vecindad de los contornos detectados. . . . .	56
4.11.	Luces detectadas con la etiqueta del color dominante en su vecindad. . . . .	56

4.12. Detección del marcador de LEDs entre un ambiente con diversas fuentes de luz. En la imagen se muestran todas las luces detectadas y se muestra el marcador formado por las luces que pasaron los filtros del algoritmo. . . . .	57
5.1. Seguimiento de trayectoria con FRHONNs . . . . .	61
5.2. Error de seguimiento . . . . .	61
5.3. Actualización de los pesos de las FRHONNs para el subsistema X . . . . .	62
5.4. Actualización de los pesos de las FRHONNs para el subsistema Y . . . . .	62
5.5. Actualización de los pesos de las FRHONNs para el subsistema Z . . . . .	62
5.6. Actualización de los pesos de las FRHONNs para el subsistema Yaw . . . . .	63
5.7. El BlueROV2 haciendo seguimiento al marcador de LEDs montado al OctoSub	63
5.8. La posición del BlueROV2 en el primer experimento usando las coordenadas relativas $(-2.5, 0)$ como referencia . . . . .	64
5.9. La posición del BlueROV2 en el primer experimento usando las coordenadas relativas $(-2.5, -0.5)$ como referencia . . . . .	64
5.10. La posición del BlueROV2 en el primer experimento usando las coordenadas relativas $(-1.5, 0.5)$ como referencia . . . . .	65
5.11. El seguimiento de trayectoria del BlueROV2 usando un controlador Super Twisting obteniendo el estado del algoritmo de visión . . . . .	65

## 1.1. Antecedentes y motivación

En la actualidad, los vehículos acuáticos son cada vez una herramienta más popular en las tareas de investigación tales como exploración, monitoreo de especies marinas, mantenimiento predictivo en infraestructura marítima, entre muchas otras. Por su arquitectura estos pueden ser divididos en vehículos operados remotamente (*Remotely Operated Vehicles*, por sus siglas en inglés **ROVs**), vehículos submarinos autónomos (*Autonomous Underwater Vehicles*, por sus siglas en inglés **AUVs**) y vehículos autónomos de superficie (*Autonomous Surface Vehicles*, por sus siglas en inglés **ASVs**). Un ejemplo de un vehículo que puede funcionar como **ROV** o **AUV** se puede observar en la Figura 1.1.



Figura 1.1: Vehículo submarino BlueROV2 de la empresa Blue Robotics <sup>1</sup>

Los **ROVs**, como su nombre lo indica, son manipulados por un operador remotamente y pueden estar cableados tanto para transferencia de datos como para alimentación eléctrica. Estos vehículos son adecuados para operaciones de corto alcance donde el lastre del cable no representa un problema, además la transferencia de vídeo e imagen puede llevarse a cabo de manera más efectiva. Por otro lado, el hecho de que dependan de un operador y que su zona

---

<sup>1</sup>Imagen tomada de <https://bluerobotics.com/>

operativa esté limitada por el cableado los hace poco recomendables para tareas de duración prolongada y/o que requieran una amplia cobertura.

A lo largo de los años, ha habido una demanda cada vez mayor de vehículos submarinos autónomos para una amplia gama de aplicaciones, como el mapeo del lecho marino, la exploración de las profundidades del océano, el monitoreo rutinario de puertos y el mantenimiento de oleoductos. Para realizar con éxito estas tareas, se requiere una solución de localización eficiente y precisa para los **AUVs**. Sin embargo, esto es un desafío para la navegación submarina, en parte porque la señal **GPS** no se puede recibir bajo el agua y los sensores acústicos requieren una instalación costosa antes de las aplicaciones. Aunque existen diferentes alternativas para la localización de los **AUVs**, tales como los *Doppler Velocity Log* (**DVL** por sus siglas en Inglés), los cuales pueden proporcionar una buena localización, el alto costo limita sus implementaciones generalizadas

Uno de los principales problemas dentro de este tipo de vehículos al momento de realizar la navegación, es el problema de afrontar la ausencia de **GPS**, dispositivo que es primordial en diferentes vehículos autónomos para determinar la posición de éste, esto para convertir a un vehículo **ROV** en un vehículo **AUV**, ya que en los últimos años ha aumentado el interés por los vehículos autónomos para misiones de intervención, es decir, misiones con vehículos submarinos autónomos y, en consecuencia, la necesidad de localización que debe ser razonablemente precisa.

Una información de navegación precisa es esencial no solo para la operación segura y la recuperación del **AUV**, sino también para que los datos recopilados por un **AUV** sean útiles. Para muchos tipos de misiones de un **AUV**, la calidad de los datos adquiridos por el vehículo depende de manera crítica de la precisión del sistema de navegación del vehículo. El costo puede ser un factor importante en el diseño del sistema de navegación **AUV**. Esto es especialmente cierto para aplicaciones que involucran la operación coordinada de múltiples vehículos.

El uso de datos de visión computacional es uno de los métodos mas empleados para la navegación submarina autónoma. Los seres humanos y la muchos de los sistemas terrestres, aéreos y también submarinos autónomos utilizan información de visión para la navegación, ya que contienen mucha información sobre el medio ambiente. Las cámaras se consideran una solución más económica con información más rica en comparación con otros sensores submarinos, como los sensores acústicos. Aunque la gama limitada de sensores de visión y

---

la gran parte de regiones sin características en entornos submarinos han sido las desventajas, los datos visuales aún pueden desempeñar un papel importante en la navegación submarina, especialmente para la detección de objetos cercanos. Se sabe que las imágenes submarinas se degradan debido a una serie de factores, como la turbidez, las partículas flotantes y la atenuación de la luz en el medio [1].

## 1.2. Planteamiento del problema

Dentro de las principales características de los vehículos autónomos (submarinos, aéreos, terrestres) se encuentra el poder realizar tareas de navegación de forma autónoma. Una de las muchas tareas que pueden realizar los vehículos submarinos es la de realizar la navegación relativa, es decir, seguir a algún objetivo valioso para realizar tareas de monitoreo o para realizar navegación en grupo junto con otros vehículos. El realizar estas tareas de navegación autónoma en vehículos submarinos implica diferentes retos y problemas a solucionar tanto en la parte teórica, como en las partes de software y hardware.

Desde el punto de vista teórico, los vehículos submarinos son sistemas no-lineales y el modelo dinámico que se obtiene para estos vehículos generalmente incluye dinámicas no modeladas y parámetros desconocidos, que para obtener, es necesaria la identificación experimental, proceso que además de tardado, puede resultar costoso.

Desde el punto de vista de la implementación en el hardware, realizar estas tareas de navegación, necesita de un conjunto de módulos de sensores especializados para la operación submarina, entre ellos, se encuentra el sistema de posicionamiento **GPS** submarino para realizar el posicionamiento del vehículo. Al día de hoy es posible encontrar módulos **GPS** comerciales para vehículos terrestres y aéreos por precios que rondan los 2,000 pesos mexicanos. Para vehículos submarinos, uno de los sistemas de posicionamiento comerciales más accesible es el kit *ROV Locator Bundle Mark II* de la compañía Celurean Sonar, el cual se muestra en la Figura 1.2, tiene un costo de 2,400 dólares, un precio elevado para el sector de la investigación pública. Por lo tanto, el posicionamiento relativo o global de los vehículos submarinos es un problema aún en la actualidad debido a los altos costos de los sistemas de posicionamiento.

---

<sup>2</sup>Imagen tomada de <https://ceruleansonar.com/>

---



Figura 1.2: Modulo GPS submarino de la empresa Celurean Sonar <sup>2</sup>.

### 1.3. Justificación

Debido al elevado costo monetario derivado de la realización de las tareas de navegación y navegación relativa para vehículos submarinos, es pertinente en la actualidad el estudio de nuevas estrategias de posicionamiento y control que puedan dar soluciones más accesibles desde el punto de vista económico.

Por lo tanto, esta tesis busca desarrollar algoritmos de visión que permitan estimar la posición del vehículo con respecto a un objetivo específico dentro del rango de visión del vehículo. Esto, sin la necesidad de utilizar hardware especializado de precios generalmente elevados para la investigación financiada con fondos públicos. De igual manera, el desarrollo de estrategias de control robustas e inteligentes que le permitan al vehículo realizar tareas de seguimiento de trayectoria de forma autónoma.

### 1.4. Estado del arte

La implementación de sistemas y algoritmos inteligentes en los vehículos autónomos ha dado pie a un amplio rango de avances en el área, desde nuevas soluciones a los problemas clásicos de la navegación autónoma hasta expandir el horizonte de las tareas posibles a realizar para un vehículo autónomo [1]-[3]. Estos sistemas inteligentes están basados en técnicas de inteligencia artificial, como las *redes neuronales*, el *control difuso*, la *probabilidad bayesiana*, el *machine learning*, la *computación evolutiva*, los *algoritmos genéticos*, entre otras.

Para la identificación y el control, se han implementado extensivamente algoritmos de control difusos [4]-[8], filtros de Kalman basados en la probabilidad bayesiana [9], [10] y en

---

los últimos años ha habido un incremento en la investigación del uso de las redes neuronales [11], [12] debido a la alta capacidad que han mostrado para el aprendizaje en la presencia de perturbaciones externas e incertidumbres en el modelado.

El uso de *redes neuronales recurrentes de alto orden* (por sus siglas en inglés **RHONNs**) va más allá de solo la identificación de los sistemas dinámicos, sino también diversos algoritmos de control neuronal en tiempo real se han desarrollado para controlar vehículos utilizando las **RHONNs** [13], en conjunto con técnicas como backstepping [14] o los modos deslizantes [15]. Así mismo, se han propuesto diversas estrategias de control basadas en redes neuronales para la navegación aérea donde los efectos aerodinámicos no lineales y las perturbaciones externas hacen relevante la implementación de algoritmos de identificación y las estrategias de control robusto. En menor medida, se han realizado investigaciones utilizando control neuronal en la navegación submarina, que a comparación de los **UAVs** (*Unmanned Aerial Vehicles*, por sus siglas en inglés), los **AUVs** aún puede considerarse como un campo emergente.

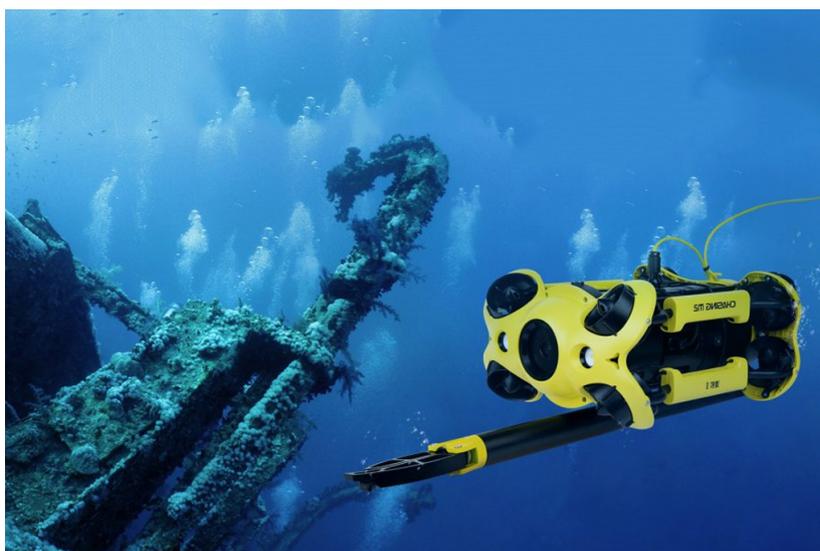


Figura 1.3: ROV realizando tareas de detección y navegación usando visión artificial <sup>3</sup>

Para la realización de un gran número de tareas en robótica móvil, la tarea de posicionamiento para navegación en diferentes entornos es difícil completar, en vehículos submarinos no es la excepción, por lo que en un gran número de ocasiones los sistemas de visión artificial es la respuesta. Visión por computadora es la transformación de datos de un vídeo, en una nueva representación, donde todas las transformaciones están hechas para lograr un objetivo más específico. La entrada de vídeo incluye información de la ubicación de donde se encuen-

<sup>3</sup>Imagen tomada de <https://aerotop.cl/drone-sub-acuatico/>

tre la cámara, o la cantidad de fotogramas por segundo. Una nueva representación, puede significar cambiar una imagen en color a una representación en escala de grises, o remover el movimiento de una secuencia de imagen. Uno de los tantos problemas que pueden resolverse usando la visión artificial es el seguimiento y la detección de objetos de interés, con la finalidad de resolver tareas autónomas utilizando vehículos móviles [16]. Actualmente, se han utilizado algoritmos de visión extensivamente para realizar tareas de reconocimiento de objetos [17], [18], así como en tareas de navegación [19]. En la Figura 1.3 se puede apreciar un **ROV** realizando tareas de detección utilizando visión artificial.

Entre los algoritmos de visión que se han estudiado en vehículos submarinos, son los que están basados en la detección de color, esto es debido a que el procesamiento necesario para realizar la detección es relativamente sencillo de implementar en comparación con los algoritmos de detección de formas o de detección de características propias de la imagen. Otro algoritmo que también es ampliamente estudiado en la literatura, es el algoritmo “*Simultaneous Localization and Mapping*”, (por sus siglas en inglés **SLAM**) y sus modificaciones como el **ORB-SLAM** (*Oriented FAST and Rotated BRIEF Simultaneous Localization and Mapping*) [20]-[22] y el algoritmo **PTAM** (*Parallel Tracking and Mapping*), diseñado para espacios pequeños y con la diferencia de que los procesos de seguimiento y mapeo se dividen en tareas separas, procesadas en hilos paralelos [23] entre otras modificaciones al algoritmo. [24]. La obtención de características en un ambiente submarino utilizando el algoritmo ORB se puede apreciar en la Figura 1.4.

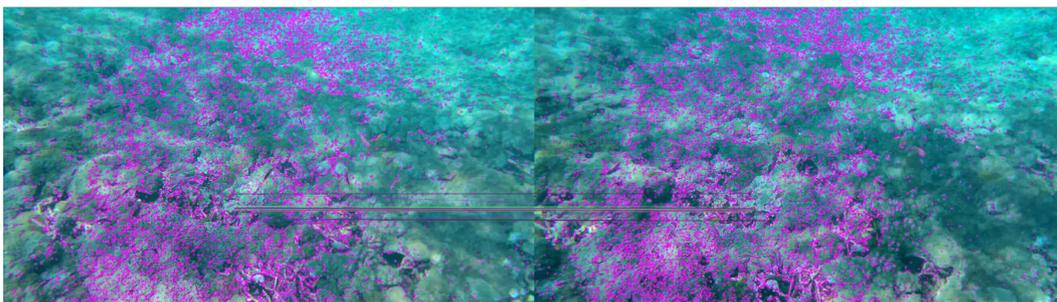


Figura 1.4: Adquisición de características y descripciones de imagen mediante algoritmo ORB <sup>4</sup>.

Sin embargo, este tipo de vehículos son susceptibles a las complejas condiciones a las que se encuentran expuestos. Con respecto a las dificultades que representa la operación de los vehículos submarinos, una de ellas es durante la exploración en terrenos desconocidos,

<sup>4</sup>Imagen tomada de <https://cse.hkust.edu.hk/pg/research/projects/saikit/underwater-slam/>

donde se pueden utilizar estos algoritmos para llevar a cabo tareas de evasión de obstáculos [25], [26]. En la Figura 1.5 se observa un vehículo BlueROV2 realizando tareas de evasión de obstáculos mediante visión computacional.



Figura 1.5: Vehículo submarino navegando en una alberca con obstáculos

Otra dificultad que se presenta, es que en entornos submarinos profundos la escasez de iluminación hace que tengan bajo rendimiento los algoritmos basados en visión artificial [27]. Una solución propuesta para esto, es el sistema de “*Light Beacons*”, que combina un algoritmo utilizando un *DVL* y *Extended Kalman Filter* con giroscopios de fibra óptica y marcadores activos, que son diodos emisores de luz con destellos continuos a una frecuencia determinada. La implementación de este algoritmo se basa en dos fuentes de información: la comunicación acústica para el intercambio de información del sensor de desviación y un grupos de marcadores ópticos en cada vehículo submarino. Con esto, la comunicación entre el objetivo y el vehículo se realiza con comunicación acústica para largas distancias y comunicación a base de visión para cortas distancias. Este sistema es especialmente útil para la navegación cooperativa [28]. En la Figura 1.6 se observa la configuración utilizada para esta solución.

Existen diferentes métodos para tratar de calcular o determinar la posición de un vehículo submarino, uno de los métodos más simple para obtener la posición de un vehículo en movimiento es integrando su velocidad con respecto del tiempo. Por lo que se requiere conocer la velocidad y la dirección del vehículo, lo que generalmente se logra con una brújula y un sensor de velocidad del agua. El problema principal está relacionado con la presencia de una corriente oceánica como se ve en la Figura 1.7, porque agregará un componente de

velocidad al vehículo que no es detectado por el sensor de velocidad. Entonces, la precisión del método se verá fuertemente afectada, especialmente cuando el vehículo navega a baja velocidad. Como lo plantean los autores en [29], es posible implementar sensores inerciales para intentar compensar los errores de medición y mejorar la estimación de posición de los vehículos, sin embargo el error en las estimaciones de posición aumenta con el tiempo y depende de la precisión de los sensores utilizados.

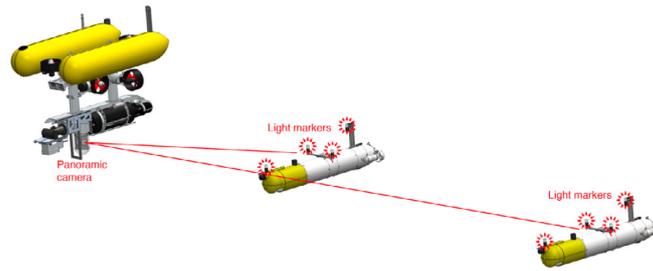


Figura 1.6: Diagrama de funcionamiento para la implementación de *Light Beacons*<sup>5</sup>

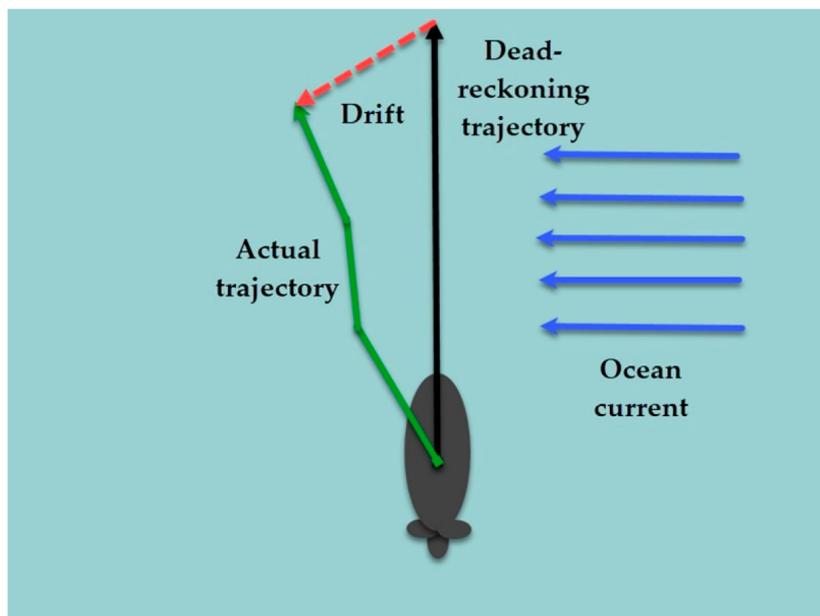


Figura 1.7: Efecto de deriva por consecuencia de corrientes oceánicas<sup>6</sup>

En la literatura es posible encontrar métodos que funcionan mediante el uso de sensores ubicados en diferentes lugares de tal manera que sea posible generar una malla intercomunicada mediante comunicación acústica para que el nodo principal pueda localizarse [30].

<sup>5</sup>Imagen tomada de Close-Range Tracking of Underwater Vehicles Using Light Beacons [28]

<sup>6</sup>Imagen tomada de Autonomous Underwater Vehicles: Localization, Navigation, and Communication for Collaborative Missions[29]

Los algoritmos de posicionamiento distribuido generalmente asumen que los nodos de sensores de anclaje se distribuyen aleatoriamente a lo largo de la red de sensores como puede observarse en la Figura 1.8, y el porcentaje de nodos de anclaje en la red también es alto (5% – 20%). Sin embargo, en el caso de un campo submarino, la red establece la columna vertebral del ancla distribuida aleatoriamente, cuya ubicación exacta se conoce previamente.

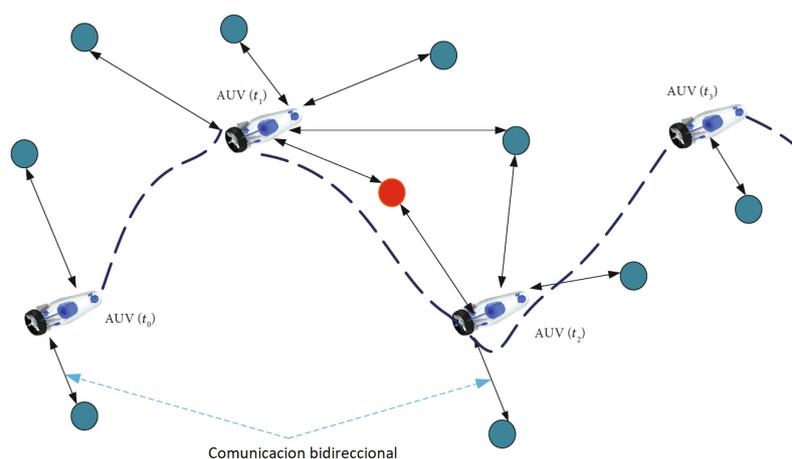


Figura 1.8: Intercambio de información para estimación de posición entre vehículos

Estimar la posición de un vehículo submarino mediante una estimación 3D ha sido propuesto en [31], [32], con las características que se agrega una variable de turbidez al agua, esto para tratar de simular las condiciones a las que se enfrenta un AUV, en un entorno no controlado, los marcadores fueron dispuestos como se muestra en la Figura 1.9, de tal forma que el problema se convierta en localizar el marcador 3D y calcular su posición a la vez de calcular la correlación entre un modelo previamente descrito en el sistema con el adquirido mediante la cámara.

A lo largo de las investigaciones se encontró que de igual forma se ha tratado de emplear marcadores pasivos [33], para estimar la posición de con respecto a estos, mediante algoritmos simples, para ello es necesario realizar máscaras para realizar filtrados a lo largo de los diferentes colores de los marcadores, intercambiando el modelo de color de **RGB** a **HSV**. Para calcular la posición del vehículo se conoce la resolución del sensor de la cámara y el tamaño del píxel en la imagen, por lo que es posible determinar en milímetros las distancias entre el centro de la imagen,  $d'$ , y cada uno de los tres puntos, y en consecuencia las distancias de cada punto a los otros dos. Usando el factor de escala es posible escalar eso al plano

<sup>7</sup>Imagen tomada de Monocular vision-based system for docking a hovering AUV [33]

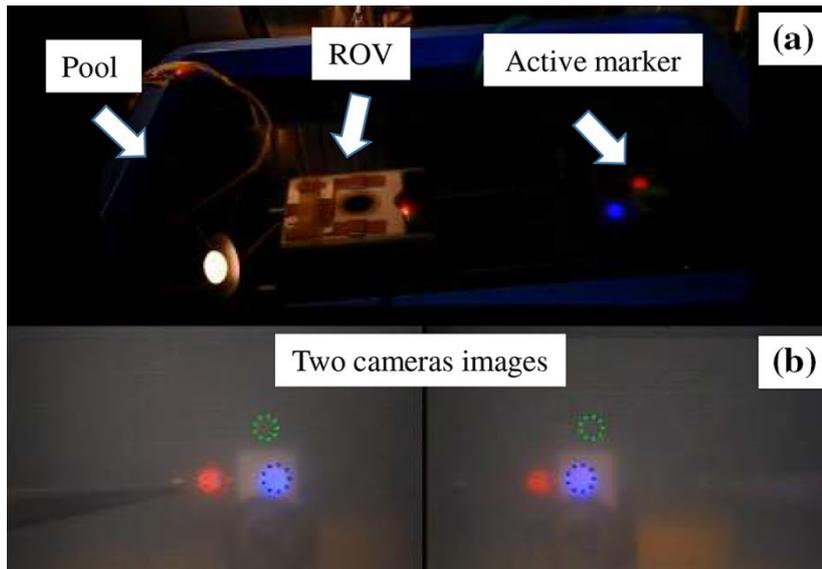


Figura 1.9: Disposición de marcadores activos, en un ambiente simulado de oscuridad<sup>7</sup>

de la escena. Se tiene conocimiento de la distancia focal de la lente  $f$ , de esta manera podemos determinar los ángulos de proyección para cada punto como puede verse en la Figura 1.10.

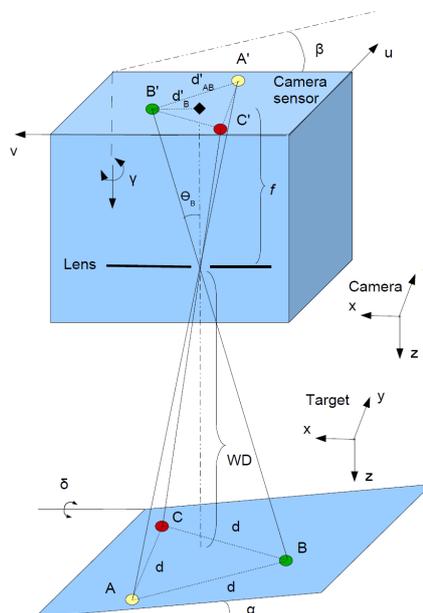


Figura 1.10: Medición de marcadores con respecto de la cámara<sup>8</sup>

Aunque por razones obvias, el seguimiento visual se puede aplicar a aplicaciones en tierra y bajo el agua, la mayoría de los algoritmos de seguimiento visual basados en redes

<sup>8</sup>Imagen tomada de Monocular vision-based system for docking a hovering AUV [33]

neuronales convolucionales o aprendizaje profundo están orientados en tierra. Se han realizado algunas investigaciones en vehículos submarinos, intentando realizar seguimiento con resultados exitosos [34].

El uso de marcadores activos como pasivos, ha sido una constante en el análisis del estado del arte de esta investigación lo cual es un enfoque de visión para realizar el seguimiento de un vehículo en movimiento en distancias cortas, las ventajas de estos métodos es que se puede implementar con una sola cámara monocular y múltiples LED sin necesidad de dispositivos acústicos. El error debe ser pequeño para un seguimiento de alta precisión a corta distancia. Como la información visual se puede obtener con una frecuencia mucho más alta que la acústica, un enfoque basado en la visión es adecuado para posicionar o rastrear vehículos submarinos a corta distancia. El uso de sistemas de sensores de alta precisión como un DVL [35], también es factible, pero debido a los altos costos de implementación, resulta en ocasiones prohibitivo, para algunas áreas, debido a esto los métodos antes propuestos resultan mejores opciones para tratar de estimar la posición relativa con respecto a otros vehículos u objetivos como puede verse en la Figura 1.11, es la opción mas fácil y viable para su implementación

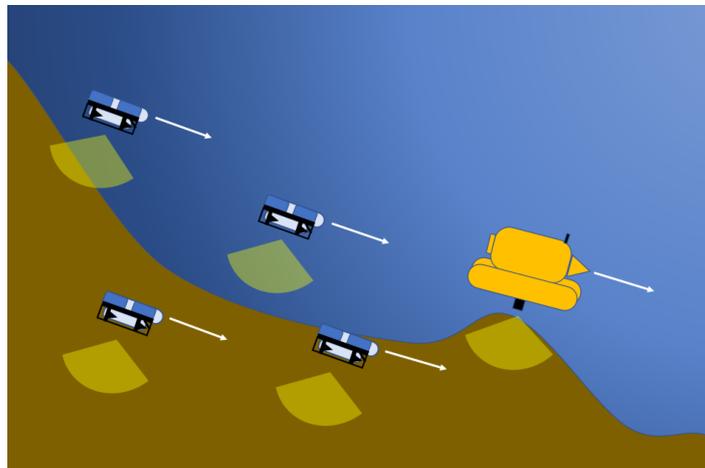


Figura 1.11: Seguimiento de un vehículo submarino líder equipado por un sensor de precisión DVL, por los seguidores<sup>9</sup>.

## 1.5. Objetivos

A continuación se presentan los objetivos general y particulares del presente trabajo de tesis.

<sup>9</sup>Imagen tomada de Short Range Tracking Method of Underwater Vehicles Based on Vision[35]

### 1.5.1. Objetivo general

Diseñar e implementar estrategias de control robusto e inteligentes para realizar tareas de navegación relativa de un vehículo submarino utilizando algoritmos de visión artificial.

### 1.5.2. Objetivos particulares

- Realizar el análisis del estado del arte de las estrategias de control robustas e inteligentes aplicadas a vehículos submarinos.
- Obtener el modelo matemático del vehículo a utilizar.
- Diseñar una estrategia de control inteligente basada en redes neuronales recurrentes de alto orden combinadas con sistemas difusos para realizar tareas de seguimiento de trayectoria.
- Diseñar e implementar una estrategia de control robusta para realizar tareas de seguimiento de trayectoria.
- Diseñar e implementar algoritmos de visión para la localización de un vehículo submarino en línea de vista, basados en técnicas de visión convencionales.
- Realizar la implementación de los algoritmos desarrollados en una plataforma experimental.

## 1.6. Metodología

Primero se realizó un estudio analítico para obtener modelo dinámico del vehículo *Blue-ROV2*, mientras se llevaba a cabo una investigación en la literatura acerca de las soluciones propuestas previamente para la determinación de los parámetros del modelo así como de las dinámicas no modeladas de un **ROV**. También se realizó una investigación en la literatura acerca de soluciones propuestas recientemente para la navegación relativa basada en la visión artificial. El vehículo utilizado en el desarrollo de esta tesis es el presentado en la Figura 1.12.

Después, se desarrolló analíticamente una estrategia de control inteligente para realizar las tareas de navegación sin la necesidad de conocer todos los parámetros del modelo. El

---

acercamiento para formular dicha estrategia fue basado en redes neuronales recurrentes de orden superior y sistemas difusos. Esta estrategia fue validada a través de una simulación en Matlab Simulink, para así comprobar que el comportamiento del sistema en simulación sea el deseado. De igual manera, se diseñó una estrategia de control robusta basada en modos deslizantes, específicamente utilizando el algoritmo Super-Twisting.



Figura 1.12: Vehículo submarino BlueROV2 disponible en el Laboratorio UMI-LAFMIA

Consecuentemente, se desarrolló un sistema de navegación relativa basada en visión artificial para realizar las tareas de navegación relativa. El acercamiento para dar solución al problema de la navegación relativa fue propuesto a partir de la implementación de un algoritmo para estimar el posicionamiento del vehículo con respecto a un punto de referencia. Aquí el punto de referencia fue escogido como una agrupación de faros submarinos, colocados estratégicamente a distancias fijas entre sí, con la finalidad de estar dentro del rango de visión de una cámara monocular montada en el vehículo, el algoritmo pueda estimar la posición del vehículo usando las distancias y ángulos observados entre los faros. Este algoritmo fue programado en **Python** para ser validado usando imágenes muestra de los faros a diferentes distancias y orientaciones. El método propuesto se describe brevemente de la siguiente manera

- Se instalaron marcadores activos en posiciones determinadas del AUV, mediante 5 luces **LED**, donde cada distancia entre los marcadores  $d = [d_1, \dots, d_n], n = 5$

- Se realizó la captura de video en tiempo real mediante la implementación de un nodo en *ROS*
- Se convirtió el espacio de color *RGB* al espacio *HSV*, para facilitar la implementación de las máscaras de color necesarias.
- Se realizó una segmentación de color, para hacer una detección de color de los diferentes marcadores.
- Se aplicaron filtros para eliminar destellos de luz, sombras, reflejos.
- Se determinaron los rangos para la detección del color de los marcadores.
- Se determinó que formas detectadas, son candidatas a ser un posible marcador real.
- Se calculó la distancia entre cada marcador, y se estimó la distancia aproximada real.

Finalmente, ambos algoritmos desarrollados (el de control y el de posicionamiento basado en visión artificial) fueron implementados en la plataforma experimental para evaluar su desempeño en un ambiente real. Los algoritmos se programaron en **Python** para implementarse en el framework **ROS**. Los algoritmos se implementaron en **ROS** en una estación en tierra que se comunica con la plataforma experimental usando el protocolo **MavLink**, esto con la finalidad de poder hacer modificaciones al algoritmo sin tener la necesidad de sacar el vehículo del agua como sería de ser implementado en el firmware de **ArduSub** programado en la computadora embebida a bordo del vehículo.

## 1.7. Publicaciones y estancias

Se realizó la publicación del siguiente artículo de congreso internacional

- **I. Alejandro García**, Benjamín Nicolas Trinidad, Salatiel García Nava, Fabrice Le Bars, Sergio Salazar Cruz, Filiberto Muñoz Palacios, *Development of a vision algorithm for close-range relative navigation of underwater vehicles*, 2022, 19Th International Conference on Electrical Engineering, Computing Science and Automatic Control, Ciudad de México, Noviembre, 2022.
-

De igual manera, se realizó una estancia de investigación en el laboratorio Lab-STICC de ENSTA Bretagne bajo la supervisión del Dr. Fabrice Le Bars, durante los meses de mayo y junio de 2022. Durante esta estancia se desarrolló parte del sistema de visión así como resultados preeliminares del controlador Super-Twisting aplicado al vehículo BlueROV2.

## **1.8. Estructura de la tesis**

Este presente trabajo de tesis está organizado en seis capítulos distribuidos de la siguiente forma. En el Capítulo 1 se plantea la problemática a resolver, el panorama del estado del arte, los objetivos a alcanzar y la metodología con lo que se consiguieron. En el Capítulo 2 se presenta la obtención de los modelos cinemático y dinámico para el vehículo BlueROV2. En el Capítulo 3 se presenta la teoría de las redes neuronales recurrentes de orden superior, la propiedad que tienen estas para la identificación de funciones no lineales y el diseño de una estrategia de control basada en RHONNs desarrollada para el BlueROV2, así como la mejora de dicha estrategia con la incorporación de lógica difusa. De igual manera se presenta la estructura del algoritmo Super-Twisting aplicado al vehículo submarino. En el Capítulo 4 se muestra las bases de los algoritmos de visión para realizar la navegación relativa y el desarrollo del algoritmo que se implementó en la plataforma experimental así como su incorporación al sistema de control. En el Capítulo 5 se presentan los resultados experimentales tanto de la estrategia de control robusta como de los algoritmos de visión desarrollados. Finalmente, en el Capítulo 6 se presentan las conclusiones y propuestas de trabajo a futuro para continuar con esta línea de investigación.

---

## Modelado matemático

En este capítulo se presenta la obtención analítica del modelo dinámico que describe el comportamiento de un ROV, dicho modelo dinámico es de suma importancia para el desarrollo de estrategias de control que permitan estabilizar la posición y la orientación del vehículo y también realizar el seguimiento de una trayectoria deseada. En la literatura se han presentado diferentes enfoques para la obtención del modelo matemático de este tipo de vehículos, entre las que destaca el propuesto por Fossen [36], donde se obtuvo el modelo matemático para representar la cinemática eq. (2.1) y la ecuación dinámica eq. (2.2) de un vehículo submarino adoptado de su modelo vectorial para un robot [37].

$$\dot{\eta} = J(\eta)\nu \quad (2.1)$$

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\nu) = \tau + \omega \quad (2.2)$$

Este modelo vectorial cuenta con la ventaja de aprovechar las propiedades matriciales usadas bastante en la literatura de la robótica y el control para así facilitar el diseño de leyes de control y el análisis de la estabilidad. La notación que utiliza este modelo dinámico será descrita en la Sección 2.1. En la Sección 2.2. se describirá el modelo cinemático para un ROV y los marcos de referencia usado en la navegación submarina. En la Sección 2.3 se mostrará a detalle el modelo dinámico y las fuerzas y momentos que lo conforman.

### 2.1. Notaciones

La notación usada comúnmente para representar el movimiento de un **ROV** de 6 GDL es la notación SNAME, que recibe el mismo nombre de la Sociedad de Arquitectos Navales e

Ingenieros Marinos, quienes la propusieron en 1950. En esta notación, mostrada en la Tabla 2.1, se propone tomar seis coordenadas individuales que describen la fuerza o momento, la velocidad lineal o angular y la posición u orientación sobre su respectiva coordenada. En

Tabla 2.1: Notación SNAME para vehículos marinos (SNAME 1950)

GDL	Fuerzas y momentos	Velocidades lineales y angulares	Posiciones y ángulos de Euler
Surge	$X$	$u$	$x$
Sway	$Y$	$v$	$y$
Heave	$Z$	$w$	$z$
Roll	$K$	$p$	$\phi$
Pitch	$M$	$q$	$\theta$
Yaw	$N$	$r$	$\psi$

forma vectorial, se tiene la pose del vehículo expresada en el marco inercial como

$$\eta = [x \ y \ z \ \phi \ \theta \ \psi]^T = \begin{bmatrix} p \\ \Theta \end{bmatrix} \quad (2.3)$$

y los subvectores que la conforman, la posición dada en la ecuación (2.4) y la orientación dada en la ecuación (2.5) (descrita por ángulos de Euler)

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^3 \quad (2.4)$$

$$\Theta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \in SO(3) \quad (2.5)$$

A su vez, el vector que describe las velocidades lineales y angulares expresadas en el marco del cuerpo como

$$\dot{\eta} = [u \ v \ w \ p \ q \ r]^T = \begin{bmatrix} \nu \\ \omega \end{bmatrix} \quad (2.6)$$

y los subvectores que la conforman, la velocidad lineal dada en la ecuación (2.7) y la velocidad angular presentada en la ecuación (2.8)

$$\nu = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \in \mathbb{R}^3 \quad (2.7)$$

$$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \in \mathbb{R}^3 \quad (2.8)$$

El vector que describe las fuerzas y momentos actuando en el ROV está dado como

$$\tau = \begin{bmatrix} X & Y & Z & K & M & N \end{bmatrix}^T = \begin{bmatrix} f \\ \mathcal{M} \end{bmatrix} \quad (2.9)$$

conformado por los subvectores de fuerzas dado por la ecuación (2.10) y momentos presentado en la ecuación (2.11)

$$f = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \in \mathbb{R}^3 \quad (2.10)$$

$$\mathcal{M} = \begin{bmatrix} K \\ M \\ N \end{bmatrix} \in \mathbb{R}^3 \quad (2.11)$$

En el caso particular del vehículo BlueROV2, es un vehículo subactuado de cuatro grados de libertad mostrados en la tabla 2.2, los vectores que describen su dinámica quedan de la siguiente forma

$$\eta = \begin{bmatrix} x & y & z & \psi \end{bmatrix}^T \quad (2.12)$$

$$\nu = \begin{bmatrix} u & v & w & r \end{bmatrix}^T \quad (2.13)$$

$$\tau = \begin{bmatrix} X & Y & Z & N \end{bmatrix}^T \quad (2.14)$$

Tabla 2.2: Grados de libertad del BlueROV2

GDL	Fuerzas y momentos	Velocidades lineales y angulares	Posiciones y ángulos de Euler
Surge	$X$	$u$	$x$
Sway	$Y$	$v$	$y$
Heave	$Z$	$w$	$z$
Yaw	$N$	$r$	$\psi$

## 2.2. Modelo cinemático

### 2.2.1. Marcos de referencia

Para describir el modelo cinemático y dinámico de un **ROV** de una forma que ayude a realizar la navegación, se necesitan definir con respecto a dos marcos de referencia, un marco de referencia inercial y un marco de referencia del cuerpo del vehículo. Comúnmente en la literatura de navegación submarina se utiliza los sistemas de referencia inercial y de cuerpo propuestos por SNAME en 1950 [38].

#### Marco de referencia inercial

El sistema de referencia inercial es un sistema que no rota y su aceleración es igual a cero, por lo que la inercia permanece constante. Los ejes y origen de este sistema referencial están representados por  $n = (x_n, y_n, z_n)$  y  $o_n$  respectivamente. Este sistema inercial que está ilustrado en la Figura 2.1 tiene el origen colocado en la superficie de cuerpo de agua sobre un punto en la superficie de la tierra y tiene los ejes coordenados según la convención *NED* (North, East, Down, por sus siglas en inglés) donde el eje  $x_n$  es positivo en dirección al norte geométrico de la tierra, el eje  $y_n$  es positivo en dirección al este geométrico de la tierra y el eje  $z_n$ , que es normal a la superficie de la tierra y positivo hacia abajo.

#### Marco de referencia del cuerpo

El sistema de referencia del cuerpo del vehículo cuyo origen reside en el vehículo, por lo que está sujeto a aceleraciones y rotaciones. Siguiendo el consejo de SNAME [38] el origen del sistema referencial del cuerpo es colocado en el centro de gravedad (*CG*) del vehículo, como está ilustrado en la Figura 2.2, ya que su *CG* está ubicado sobre el plano principal de simetría del vehículo y de esta forma se pueden utilizar las simetrías del vehículo.

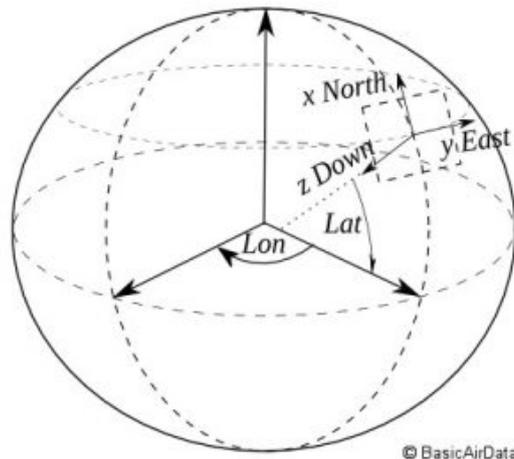


Figura 2.1: Sistema referencial NED (fuente: BasicAirData)

El sistema referencial del cuerpo tiene los ejes representados por  $b = (x_b, y_b, z_b)$  y el origen representado por  $o_b$ . El eje  $x_b$  es positivo hacia la parte frontal del vehículo, el eje  $y_b$  es positivo hacia el lado derecho del vehículo y el eje  $z_b$  es positivo hacia abajo del vehículo.

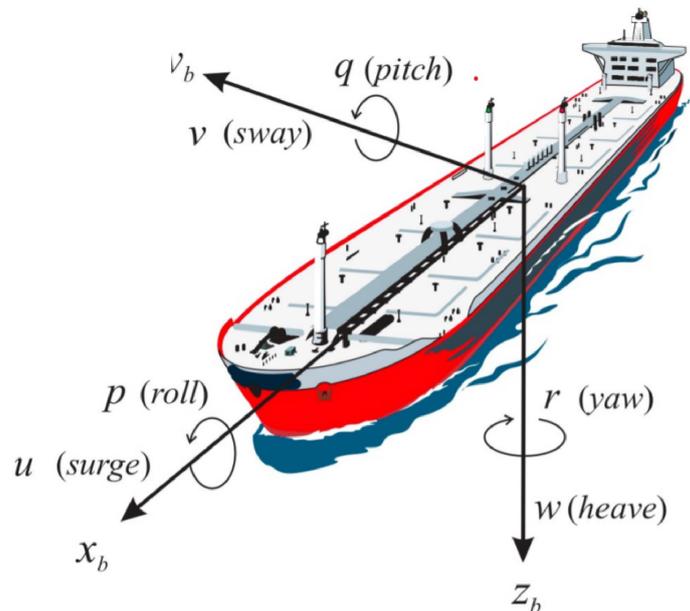


Figura 2.2: Sistema referencial del cuerpo (créditos a: T.I.Fossen)

### 2.2.2. Transformación entre marcos de referencia

Ya que se tienen determinados los sistemas referenciales para el vehículo BlueROV2, como está ilustrado en la Figura 2.3, se debe buscar una forma transformar las coordenadas

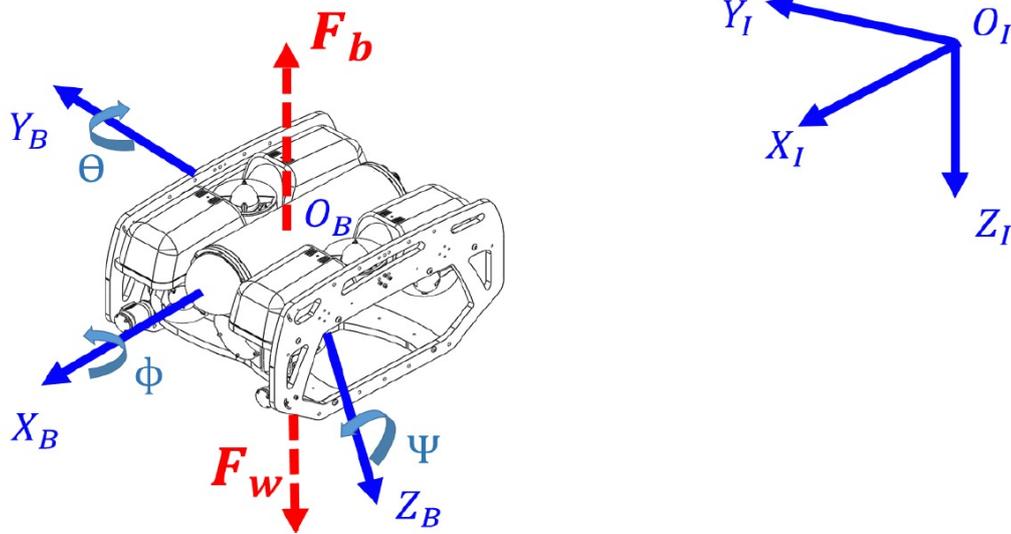


Figura 2.3: Marco referencial del cuerpo  $O_B$  e inercial  $O_I$

de un marco referencial a otro.

Un vector que está descompuesto en las coordenadas de un marco de referencia en específico puede ser transformado al otro marco de referencia usando una matriz de rotación, por ejemplo un vector  $\chi^b \in \mathbb{R}^3$  en el marco de referencia del cuerpo puede ser transformado al marco de referencia inercial, obteniendo así un vector  $\chi^n \in \mathbb{R}^3$  que está descompuesto en las coordenadas del marco inercial. La transformación está dada por

$$\chi^n = R_b^n \chi^b \quad (2.15)$$

donde  $R_b^n$  es una matriz de transformación para llevar del sistema referencial  $b$  al sistema referencial  $n$ . Esta matriz  $R_b^n$  tiene la propiedad de que  $(R_b^n)^{-1} = R_n^b$ , por lo que su inversa puede llevar a la cabo la transformación del sistema referencial  $n$  al sistema referencial  $b$ .

### Transformación con ángulos de Euler

Los ángulos de Euler, ilustrados en la Figura 2.3, están definidos como roll( $\phi$ ), pitch( $\theta$ ) y yaw( $\psi$ ) y sus giros son positivos a la derecha en sentido del eje coordenado sobre el que actúan. Los ángulos de Euler tienen la facultad de describir la transformación de un sistema referencial a otro por medio de tres rotaciones del cuerpo rígido alrededor de los ejes coordenados del sistema referencial del cuerpo. La transformación para las velocidades lineales

del marco del cuerpo  $b$  al marco inercial  $n$  está dada por

$$\nu^n = R_b^n(\Theta)\nu^b \quad (2.16)$$

donde la matriz de rotación  $R_b^n$  se descompone en tres rotaciones consecutivas sobre sus respectivos ejes coordenados de la siguiente forma

$$R_b^n = R_\phi^T R_\theta^T R_\psi^T \quad (2.17)$$

donde la matriz de rotación de pitch (eq. (2.18)), de roll (eq. (2.19)) y de yaw (eq. (2.20)) están representadas de forma matricial como

$$R_\phi^T = \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.18)$$

$$R_\theta^T = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.19)$$

$$R_\psi^T = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.20)$$

De forma similar, la transformación de las velocidades angulares del marco del cuerpo al marco inercial está dada por

$$\omega^n = T_b^n(\Theta)\omega^b \quad (2.21)$$

donde la matriz de transformación  $T_b^n$  está dada por

$$T_b^n = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{bmatrix} \quad (2.22)$$

De forma particular, la transformación completa para llevar las coordenadas del Blue-

ROV2 del marco del cuerpo al marco inercial está dada de la siguiente forma

$$\dot{\eta} = J_{\Theta}(\eta)\nu \quad (2.23)$$

donde

$$J_{\Theta}(\eta) = \begin{bmatrix} R_b^n(\Theta) & 0_{2 \times 2} \\ 0_{2 \times 2} & T_b^n(\Theta) \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 & 0 \\ \sin \psi & \cos \psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.24)$$

por lo tanto, la matriz de transformación para llevarlo del marco inercial al cuerpo queda de la siguiente manera

$$J_{\Theta}^{-1}(\eta) = \begin{bmatrix} \cos \psi & \sin \psi & 0 & 0 \\ -\sin \psi & \cos \psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.25)$$

### 2.3. Modelo dinámico

El modelo dinámico en el marco del cuerpo para el BlueROV2 está representado por las siguientes ecuaciones derivadas de la formulación de Newton-Euler [37]

$$\begin{aligned} M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) &= \tau \\ \dot{\eta} &= J(\eta)\nu \end{aligned} \quad (2.26)$$

donde  $M$  es la matriz de inercia,  $C(\nu)$  es la matriz de Coriolis y fuerzas centrífugas inducidas por  $M$  debido a la rotación del marco del cuerpo con respecto al marco inercial,  $D(\nu)$  es la matriz de amortiguamiento hidrodinámico,  $g(\eta)$  es el vector de fuerza gravitacional y empuje hidrostático y  $\tau$  es el vector de entradas de control. A continuación, se presenta de forma detallada la obtención de estas matrices.

#### Matriz de inercia

Esta matriz considera los efectos del tensor de inercia y la masa del vehículo. Está contiene una parte que considera la dinámica del cuerpo rígido, denominada como  $M_{RB}$  y una parte que considera la hidrodinámica del cuerpo dentro de un fluido, denominada como  $M_A$ .

Por lo tanto la matriz de inercia está definida como

$$M(\nu) = M_{RB}(\nu) + M_A(\nu) \quad (2.27)$$

La matriz de inercia del cuerpo rígido  $M_{RB}$  puede ser obtenida para seis grados de libertad a través de la aplicación de la formulación de Newton [36] teniendo como matriz de inercia del cuerpo rígido a

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & mz_g & -my_g \\ 0 & m & 0 & -mz_g & 0 & mx_g \\ 0 & 0 & m & my_g & -mx_g & 0 \\ 0 & -mz_g & my_g & I_x & -I_{xy} & -I_{xz} \\ mz_g & 0 & -mx_g & -I_{yx} & I_y & -I_{yz} \\ -my_g & mx_g & 0 & -I_{zx} & -I_{zy} & I_z \end{bmatrix} \quad (2.28)$$

donde  $m$  es la masa del vehículo,  $I_x$ ,  $I_y$  e  $I_z$  son los momentos de inercia alrededor de los ejes coordenados del marco referencial del cuerpo;  $I_{xy} = I_{yx}$ ,  $I_{xz} = I_{zx}$  y  $I_{yz} = I_{zy}$  son los productos de inercia;  $x_g$ ,  $y_g$  y  $z_g$  son las posiciones del centro de gravedad  $CG$  en relación al centro geométrico del vehículo.

En el caso particular del BlueROV2, la matriz de inercia del cuerpo rígido queda reducida para cuatro grados de libertad ( $x$ ,  $y$ ,  $z$  y  $\psi$ ) de la siguiente forma

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & -my_g \\ 0 & m & 0 & mx_g \\ 0 & 0 & m & 0 \\ -my_g & mx_g & 0 & I_z \end{bmatrix} \quad (2.29)$$

Se considera que el origen del marco de referencia del cuerpo se encuentra en el centro de gravedad  $CG$  del vehículo, por lo que  $x_g = y_g = z_g = 0$  y  $I_{xy} = I_{xz} = I_{yz} = 0$ , entonces la

matriz de inercia del cuerpo rígido para el BlueROV2  $M_{RB}$  queda reducida a

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 \\ 0 & m & 0 & 0 \\ 0 & 0 & m & 0 \\ 0 & 0 & 0 & I_z \end{bmatrix} \quad (2.30)$$

La matriz de inercia de masa añadida para un vehículo submarino de seis grados de libertad puede ser definida como [37]

$$M_A = - \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix} \quad (2.31)$$

donde, de acuerdo a la notación de SNAME [38], las derivadas hidrodinámicas representan las fuerzas y momentos hidrodinámicos de la masa añadida en la dirección de un eje coordenado causados por la aceleración del vehículo en el grado de libertad indicado ( $X_{\dot{u}}$  es la fuerza hidrodinámica actuando en  $x$  causado por la aceleración  $\dot{u}$ ).

Al considerar el movimiento del BlueROV2 desacoplado, las fuerzas fuera de la diagonal se pueden despreciar y tomando en cuenta que es de cuatro grados de libertad, queda una matriz simplificada de la siguiente forma

$$M_A = - \begin{bmatrix} X_{\dot{u}} & 0 & 0 & 0 \\ 0 & Y_{\dot{v}} & 0 & 0 \\ 0 & 0 & Z_{\dot{w}} & 0 \\ 0 & 0 & 0 & N_{\dot{r}} \end{bmatrix} \quad (2.32)$$

Finalmente, la matriz de inercia  $M$  para el vehículo BlueROV2 queda como

$$M = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 & 0 \\ 0 & m - Y_{\dot{v}} & 0 & 0 \\ 0 & 0 & m - Z_{\dot{w}} & 0 \\ 0 & 0 & 0 & m - N_{\dot{r}} \end{bmatrix} \quad (2.33)$$

### Matriz de Coriolis

La matriz de Coriolis y fuerzas centrífugas es una matriz que contiene las fuerzas de inercia que actúan perpendiculares a la dirección del movimiento. Esta matriz está conformada por las fuerzas causadas por la inercia del cuerpo rígido  $C_{RB}$  y por las fuerzas causadas por la inercia de la masa añadida  $C_A$ , con lo cual se tiene  $C(\nu) = C_A(\nu) + C_{RB}(\nu)$ . Para obtener la matriz de Coriolis a partir de la matriz de inercia, se puede aplicar las ecuaciones de Kirchhoff con un enfoque basado en la energía cinética [37], teniendo así la matriz de Coriolis y fuerzas centrífugas del cuerpo rígido para un vehículo de seis grados de libertad

$$C_{RB} = \begin{bmatrix} 0 & 0 & 0 & 0 & mw & -mv \\ 0 & 0 & 0 & -mw & 0 & mu \\ 0 & 0 & 0 & mw & -mu & 0 \\ 0 & -mw & mv & 0 & I_z r & -I_y q \\ mw & 0 & -mu & -I_z r & 0 & I_x p \\ -mv & mu & 0 & I_y q & -I_x p & 0 \end{bmatrix} \quad (2.34)$$

de la misma forma se puede obtener la matriz de Coriolis y fuerzas centrífugas de la masa añadida para un vehículo de seis grados de libertad

$$C_A = \begin{bmatrix} 0 & 0 & 0 & 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v \\ 0 & 0 & 0 & Z_{\dot{w}}w & 0 & -X_{\dot{u}}u \\ 0 & 0 & 0 & -Y_{\dot{v}}v & X_{\dot{u}}u & 0 \\ 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v & 0 & -N_{\dot{r}}r & M_{\dot{q}}q \\ Z_{\dot{w}}w & 0 & -X_{\dot{u}}u & N_{\dot{r}}r & 0 & -K_{\dot{p}}p \\ -Y_{\dot{v}}v & X_{\dot{u}}u & 0 & -M_{\dot{q}}q & K_{\dot{p}}p & 0 \end{bmatrix} \quad (2.35)$$

Con lo cual se obtiene la matriz de Coriolis para un vehículo submarino de seis grados de libertad

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & (m - Z_{\dot{w}})w & -(m - Y_{\dot{v}})v \\ 0 & 0 & 0 & -(m - Z_{\dot{w}})w & 0 & (m - X_{\dot{u}})u \\ 0 & 0 & 0 & (m - Y_{\dot{v}})v & -(m - X_{\dot{u}})u & 0 \\ 0 & (m - Z_{\dot{w}})w & -(m - Y_{\dot{v}})v & 0 & (I_z - N_{\dot{r}})r & -(I_y - M_{\dot{q}})q \\ -(m - Z_{\dot{w}})w & 0 & (m - X_{\dot{u}})u & -(I_z - N_{\dot{r}})r & 0 & (I_x - K_{\dot{p}})p \\ (m - Y_{\dot{v}})v & -(m - X_{\dot{u}})u & 0 & (I_y - M_{\dot{q}})q & -(I_x - K_{\dot{p}})p & 0 \end{bmatrix} \quad (2.36)$$

Al igual que la matriz de inercia, para el caso del BlueROV2, esta matriz queda reducida a su versión de cuatro grados de libertad

$$C = \begin{bmatrix} 0 & 0 & 0 & -(m - Y_{\dot{v}})v \\ 0 & 0 & 0 & (m - X_{\dot{u}})u \\ 0 & 0 & 0 & 0 \\ (m - Y_{\dot{v}})v & -(m - X_{\dot{u}})u & 0 & 0 \end{bmatrix} \quad (2.37)$$

### Matriz de amortiguamiento hidrodinámico

La matriz de amortiguamiento hidrodinámico contiene la fuerza resultado de la fricción que ejerce la viscosidad del fluido sobre el cuerpo. De manera general se puede decir que la matriz de amortiguamiento dinámico, dada por la ecuación (2.38), está constituida por cuatro causas principales: el amortiguamiento potencial  $D_P$ , el amortiguamiento por fricción  $D_S$ , el amortiguamiento por oleaje  $D_W$  y el amortiguamiento debido al desprendimiento de vórtices  $D_M$  [36]

$$D(\nu) = D_P(\nu) + D_S(\nu) + D_W(\nu) + D_M(\nu) \quad (2.38)$$

Al tratarse de un vehículo submarino, el amortiguamiento potencial  $D_P$  y el amortiguamiento por oleaje  $D_W$  pueden ser despreciados. El resto de los términos pueden ser resumidos de forma aproximada a una matriz de amortiguamiento lineal causado principalmente por la fricción  $D_L = D_S$  y una matriz de amortiguamiento no lineal causado principalmente por el desprendimiento de vórtices  $D_{NL} = D_M$ . Además de esta aproximación, si se toma en cuenta que la dinámica está desacoplada, entonces los términos fuera de las diagonales son

despreciados, quedando cada una de las matrices de la siguiente forma

$$D_S = - \begin{bmatrix} X_u & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_v & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_w & 0 & 0 & 0 \\ 0 & 0 & 0 & K_p & 0 & 0 \\ 0 & 0 & 0 & 0 & M_q & 0 \\ 0 & 0 & 0 & 0 & 0 & N_r \end{bmatrix} \quad (2.39)$$

$$D_M = - \begin{bmatrix} X_{u|u}|u| & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{v|v}|v| & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_{w|w}|w| & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{p|p}|p| & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{q|q}|q| & 0 \\ 0 & 0 & 0 & 0 & 0 & N_{r|r}|r| \end{bmatrix} \quad (2.40)$$

Y así la matriz de amortiguamiento hidrodinámico (2.38) para un vehículo submarino de seis grados de libertad está dada por

$$D = - \begin{bmatrix} X_u + X_{u|u}|u| & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_v + Y_{v|v}|v| & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_w + Z_{w|w}|w| & 0 & 0 & 0 \\ 0 & 0 & 0 & K_p + K_{p|p}|p| & 0 & 0 \\ 0 & 0 & 0 & 0 & M_q + M_{q|q}|q| & 0 \\ 0 & 0 & 0 & 0 & 0 & N_r + N_{r|r}|r| \end{bmatrix} \quad (2.41)$$

Al igual que con la matriz de inercia y la de Coriolis, para el caso particular del BlueROV2 se reduce a cuatro grados de libertad y queda como

$$D = - \begin{bmatrix} X_u + X_{u|u}|u| & 0 & 0 & 0 \\ 0 & Y_v + Y_{v|v}|v| & 0 & 0 \\ 0 & 0 & Z_w + Z_{w|w}|w| & 0 \\ 0 & 0 & 0 & N_r + N_{r|r}|r| \end{bmatrix} \quad (2.42)$$

## Vector de fuerza gravitacional y fuerzas de restitución

En este vector se toman en cuenta la fuerza gravitacional, definida en la ecuación (2.43), y las fuerzas de restitución dadas en la ecuación (2.44), que son causadas por el conjunto de fuerzas de flotabilidad, las cuales son fuerzas verticales ascendentes (negativas en la convención **NED**) presentes cuando hay un objeto sumergido en un fluido. Estas fuerzas están representadas por

$$W = mg \quad (2.43)$$

$$B = \rho_f g v_f \quad (2.44)$$

donde  $g$  es la aceleración gravitacional,  $\rho_f$  es la densidad del fluido, en este caso la densidad del agua por la que está navegando el vehículo y  $v_f$  es el volumen del fluido desplazado. El efecto que tengan estas fuerzas en el submarino depende de la ubicación del centro de gravedad y el centro de flotabilidad del vehículo con respecto al origen del marco del cuerpo.

Como el BlueROV2 por diseño tiene los centros de gravedad y flotabilidad alineados, los vectores de fuerza gravitacional  $r_g$  y de fuerza de flotabilidad  $r_b$  están dados por

$$r_g = \begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ z_g \end{bmatrix} \quad (2.45)$$

$$r_b = \begin{bmatrix} x_b \\ y_b \\ -z_b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -z_b \end{bmatrix} \quad (2.46)$$

Estos vectores al ser transformados al sistema referencial inercial del cuerpo dan como resultado para un vehículo de 6 GDL lo siguiente

$$g_B(\eta) = \begin{bmatrix} (W - B) \sin \theta \\ -(W - B) \cos \theta \sin \phi \\ -(W - B) \cos \theta \cos \phi \\ z_g W \cos \theta \sin \phi \\ z_g W \sin \theta \\ 0 \end{bmatrix} \quad (2.47)$$

Para el BlueROV2, al ser de cuatro grados de libertad, queda reducido a

$$g_B(\eta) = \begin{bmatrix} (W - B) \sin \theta \\ -(W - B) \cos \theta \sin \phi \\ -(W - B) \cos \theta \cos \phi \\ 0 \end{bmatrix} \quad (2.48)$$

## 2.4. Conclusiones

En este capítulo se presentó la obtención de los modelos no lineales cinemático y dinámico de un vehículo de 6GDL y para el vehículo BlueROV2 en su configuración de 4 GDL. Para obtenerlos se asumieron algunas condiciones para simplificar el modelo dinámico y que tuviera la menor cantidad de parámetros desconocidos. Sin embargo, para obtener los parámetros restantes es necesario realizar estimaciones mediante un estudio experimental o una simulación de la hidrodinámica del vehículo. El modelo aquí obtenido será utilizado en el siguiente capítulo para diseñar el sistema de control del vehículo, donde se empleará una estrategia de control que permita controlar el vehículo de una forma robusta sin la necesidad de conocer este tipo de parámetros previamente.

---

---

## Diseño del control con RHONNs

---

En este capítulo se presenta el diseño de las estrategias de control inteligentes y robustas para el vehículo BlueROV2. La estrategia de control propuesta se basa en un algoritmo de redes neuronales recurrentes difusas de orden superior (**FRHONN**) debido a su capacidad para realizar aproximaciones de funciones no lineales únicamente con las entradas y salidas del sistema, que en este caso es el vehículo submarino BlueROV2. Como se mencionó al final del capítulo pasado, para obtener los parámetros desconocidos del modelo del **ROV** dado por la ecuación (2.2) es necesario hacer análisis hidrodinámicos y pruebas experimentales, por lo que un algoritmo de aproximación resulta benéfico cuando no se tienen estos parámetros. De igual manera, se presenta el diseño de una estrategia de control robusta basada en el algoritmo Super-Twisting, este algoritmo fue elegido ya que se busca evitar los fenómenos de chattering, se utilizará la continuidad del control de supertwisting para compensar las incertidumbres acotadas [39], [40]. Por tanto, la continuidad y robustez de nuestro controlador atenuará el efecto de chattering, mejorando la estabilidad y maniobrabilidad del vehículo submarino.

### 3.1. Identificación utilizando RHONNs

Con propósitos de clarificar la estructura de las **FRHONNs**, primeramente se presenta en esta sección la estructura de las redes neuronales recurrentes de orden superior **RHONNs**.

### 3.1.1. Estructura de las RHONNs

Considere el modelo de una red neuronal recurrente de orden superior donde el estado de cada neurona está gobernado por una ecuación diferencial con la forma

$$\dot{x}_j = -a_j x_j + b_j \sum_{k=1}^L w_{jk} \prod_{j \in I_k} y_j^{d_j(k)} \quad (3.1)$$

donde  $x_j$  es el estado de cada neurona,  $a_j, b_j$  son constantes positivas reales,  $w_{jk}$  es el  $k$ -ésimo peso ajustable que conecta el estado  $j$  a la neurona,  $L$  representa el número total de pesos usados para identificar el comportamiento del sistema,  $y_j$  es el vector de entradas externas y estados de cada neurona evaluados a través de una función de activación sigmoïdal.

El vector de entradas para cada neurona está dado por

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_i \\ y_{i+1} \\ \vdots \\ y_{i+j} \end{pmatrix} = \begin{pmatrix} s(x^1) \\ \vdots \\ s(x^i) \\ u_1 \\ \vdots \\ u_j \end{pmatrix} \quad (3.2)$$

donde  $u = [u^1, u^2, \dots, u^j]^T$  es el vector de entradas externas a la red neuronal y  $s(\cdot)$  es una función no lineal de tipo sigmoïdal. Definiendo el vector  $z$  como

$$z = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_L \end{pmatrix} = \begin{pmatrix} \prod_{j \in I_1} y_j^{d_j(1)} \\ \prod_{j \in I_2} y_j^{d_j(2)} \\ \vdots \\ \prod_{j \in I_L} y_j^{d_j(L)} \end{pmatrix} \quad (3.3)$$

el modelo de la **RHONN** puede ser reescrito como

$$\dot{x}_j = -a_j x_j + b_j \sum_{k=1}^L w_{jk} z_{jk} \quad (3.4)$$

Definiendo un vector de parámetros ajustables como  $w_{jk} = b_j [w_{j1}, w_{j2}, \dots, w_{jL}]^T$ , el modelo

se reescribe como

$$\dot{x}_j = -a_j x_j + (w_{jk})^T z_{jk} \quad (3.5)$$

donde los vectores  $w_{jk}$  representan los pesos ajustables de la red neuronal mientras que los coeficientes  $a_j$  son parte de la arquitectura de la red neuronal y se fijan durante el entrenamiento. Para garantizar que cada neurona  $x_j$  sea **BIBO**-estable (*Bounded Input-Bounded Output*) se asume que  $a_j > 0$ .

### 3.1.2. Propiedades de aproximación de la RHONN

El sistema dinámico que se busca aproximar, en este caso el del BlueROV2 está dado por

$$\dot{\chi}_j^i = F(\chi_j^i, u^i) \quad (3.6)$$

donde  $u^i \in \mathbb{R}^i$  es la entrada del sistema,  $\chi_j^i \in \mathbb{R}^j$  es el estado del sistema y  $F : \mathbb{R}^{i+j}$  es un campo vectorial suave definido en el conjunto compacto  $\Upsilon \subset \mathbb{R}^{i+j}$ , donde  $i$  y  $j$  son constantes. La aproximación consiste en determinar si existen pesos  $w_{jk}^i$  tal que la ecuación (3.5) aproxime el comportamiento entrada-salida de un sistema con la forma de la ecuación (3.6).

A continuación, se muestra el teorema que muestra que se puede realizar esta aproximación a algún nivel de precisión si se permiten suficientes conexiones recursivas de orden superior en la **RHONN** de la ecuación (3.5). Para este teorema se asume que

- $F(\chi_j^i, u^i)$  es continuo y localmente Lipschitz ya que la ecuación (3.6) tiene un solución única.
- $(\chi_j^i(t), u^i(t)) \in \Upsilon$  para toda  $t$  en un intervalo de tiempo  $J_T = (t_0 \leq t \leq T)$ , donde  $J_T$  representa el tiempo en el que la aproximación se realizará.

#### **Teorema 3.1 [13]**

*Supóngase que el sistema representado por la ecuación (3.6) y la **RHONN** de la ecuación (3.5) tienen inicialmente el mismo estado  $\chi_j^i(0) = x_j^i(0)$ . Entonces, para toda  $\epsilon > 0$  y toda  $T > 0$  finita, existe un entero  $L$  y un vector  $w_{jk}^{*i} \in \mathbb{R}^L$  ya que el estado  $x_j^i(t)$  del modelo de **RHONN** de la ecuación (3.5), con  $L$  conexiones recursivas de orden superior y valores de*

pesos  $w_{jk}^i = w_{jk}^{*i}$ , satisface que

$$\sup_{0 \leq t \leq T} |x_j^i(t) - \chi_j^i(t)| \leq \epsilon \quad (3.7)$$

G. Rovithakis demuestra este teorema en su libro [13], sin embargo, este teorema no proporciona ningún método para encontrar el valor óptimo de los pesos  $w_{jk}^{*i}$ , por lo que se debe implementar un algoritmo de entrenamiento para encontrar los pesos de la red.

### 3.1.3. Entrenamiento basado en el algoritmo de gradiente descendente

Para el entrenamiento de la red con este algoritmo, se asume que el sistema desconocido está modelado exactamente por una arquitectura de RHONN como la de la eq. (3.13). Basado en el supuesto de que no hay errores de modelado, existe un vector de pesos desconocidos  $w_{jk}^*$  ya que cada estado  $\chi_j$  del sistema dinámico desconocido satisface

$$J(w_1, \dots, w_n) = \frac{1}{2} \sum_{i=1}^n \xi_i^2 = \frac{1}{2} \sum_{i=1}^n [(w_i - w_i^*)^T - \xi_i]^2 \quad (3.8)$$

Un método para esto es el de gradiente descendente que da la siguiente ley de entrenamiento

$$\dot{w}_{jk} = -\Gamma_{jk} z_{jk} \xi_j \quad (3.9)$$

donde la ganancia adaptable  $\Gamma_{jk} \in R^{L \times L}$  es una matriz definida positiva.

A continuación, se mostrará el teorema que garantiza la estabilidad y convergencia de la ley de adaptación de los pesos de la RHONN con el método del descenso de gradiente.

#### Teorema 3.2 [13]

Considere el modelo de RHONN dado por la eq. (3.5), cuyos pesos están ajustados de acuerdo a la eq. (3.9). Entonces, para  $i = 1, 2, \dots, n$ :

- $\xi_i, \phi_i \in \mathcal{L}_\infty$  ( $\xi_i$  y  $\phi$  son uniformemente acotados)
- $\lim_{t \rightarrow \infty} \xi_i(t) = 0$

## 3.2. Identificación y control control utilizando FRHONNs

En la sección anterior se propuso una estructura de identificación basada en un algoritmo de redes neuronales recurrentes de orden superior. Esta estructura de identificación puede ser

mejorada, expandiendo el algoritmo de redes neuronales recurrentes de orden superior para que, mediante lógica difusa, se utilizaran  $q$  particiones difusas para identificar la dinámica del submarino con un menor error estacionario.

### 3.2.1. Estructura de las FRHONNs

Considere una función no lineal a aproximar, en este caso el sistema dinámico de un vehículo submarino, está dada por

$$\dot{\chi}^i = f(\chi^i, u^i) \quad (3.10)$$

donde  $u \in \mathbb{R}^s$  es el vector de entradas del sistema,  $\chi \in \mathbb{R}^n$  es el vector de  $i = 1, 2, \dots, n$  estados del sistema y  $f : \mathbb{R}^{n+s} \rightarrow \mathbb{R}^n$  es un campo vectorial suave definido en el conjunto compacto  $\Upsilon \subset \mathbb{R}^{n+s}$ , donde  $n$  y  $s$  son constantes.

Ahora, considere el modelo de un grupo de  $q$  redes neuronales recurrentes de orden superior, cada una definida en la  $p = 1, 2, \dots, q$  subregión  $\Omega_{f_i}^p$  de la variable de estado  $i$ , con la siguiente forma:

$$\dot{x}_i^p = -ax_i + \sum_{p=1}^q \tilde{x}_i^p \cdot \left( \sum_{l=1}^k w_i^{pl} \cdot s_l(x(t), u(t)) \right) \quad (3.11)$$

donde  $x_i$  es la neurona que representa cada uno de los estados del sistema,  $\tilde{x}_i^p$  es el centro de la partición difusa  $p$  de el estado  $i$ ,  $w_i^{pl}$  es el peso neuronal de la conexión de la entrada  $l = 1, 2, \dots, k$  a la neurona  $i$  para una partición difusa  $p$ ,  $s_l$  es el vector de entradas externas y estados de cada neurona evaluados a través de una función de activación sigmoideal.

En este trabajo de tesis, para el caso particular de un vehículo submarino, se propone una forma de segundo orden de este grupo de FRHONNs con la siguiente forma

$$\begin{aligned} \dot{x}_{1j}^p &= -a_1 x_{1j} + \sum_{p=1}^q \tilde{x}_{1j}^p \cdot \left( \sum_{l=1}^k w_{1j}^{pl} \cdot s_l(x(t), u(t)) \right) + x_2 \\ \dot{x}_{2j}^p &= -a_2 x_{2j} + \sum_{p=1}^q \tilde{x}_{2j}^p \cdot \left( \sum_{g=1}^d w_{2j}^{pg} \cdot s_g(x(t), u(t)) \right) + u \end{aligned} \quad (3.12)$$

donde  $i = 1, 2$  que representan la posición y la velocidad de cada uno de los  $j = 1, 2, \dots, m$  subsistemas de segundo orden en los que puede dividirse el modelo dinámico. La estructura propuesta es basada en la presentada en [14], donde los autores utilizan RHONNs en el sistema (3.12). A diferencia de [14], se propone una estructura basada en FRHONNs, las

cuales mejoran la identificación del sistema.

Para la aproximación de cada subsistema de segundo orden, se consideran un número diferente de entradas recursivas en cada neurona del subsistema, por lo tanto, para las conexiones de la neurona que representa la posición ya sea lineal o angular  $i = 1$ , se tienen  $l = 1, 2, \dots, k$  entradas, mientras que para la neurona que representa la velocidad del subsistema, se tienen  $g = 1, 2, \dots, d$  entradas. Por simplicidad se considera que tanto las neuronas de la posición como de la velocidad tienen el mismo número de particiones difusas  $q$ .

Esta estructura propuesta de red neuronal formada por **FRHONNs** puede ser reescrita a una forma más compacta como

$$\begin{aligned}\dot{x}_1 &= A_1 x_1 + X_1 W_1 s_1(x, u) + x_2 \\ \dot{x}_2 &= A_2 x_2 + X_2 W_2 s_2(x, u) + u\end{aligned}\quad (3.13)$$

donde  $A_i$  es una matriz diagonal estable de dimensiones  $m \times m$ ,  $A_i = \text{diag}[-a_{i1}, -a_{i2}, \dots, -a_{im}]$ , con  $a_{ij} > 0$ .  $X_i$  es una matriz que contiene los centros de las particiones difusas de cada variable de estado, de dimensiones  $m \times m \cdot q$  y con la forma de  $X_i = \text{diag}[\tilde{x}_{i1}, \tilde{x}_{i2}, \dots, \tilde{x}_{im}]$ , con  $\tilde{x}_{ij}$  siendo un vector fila de dimensión  $q$  con la siguiente forma:

$$\tilde{x}_{ij} = \begin{bmatrix} \tilde{x}_{ij}^1 & \tilde{x}_{ij}^2 & \dots & \tilde{x}_{ij}^q \end{bmatrix}\quad (3.14)$$

Entonces  $X_i$  en una forma más detallada:

$$X_i = \begin{bmatrix} \tilde{x}_{i1}^1 & \dots & \tilde{x}_{i1}^q & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & \tilde{x}_{i2}^1 & \dots & \tilde{x}_{i2}^q & 0 & \dots & 0 \\ \dots & \dots \\ 0 & \dots & 0 & 0 & \dots & 0 & \tilde{x}_{im}^1 & \dots & \tilde{x}_{im}^q \end{bmatrix}\quad (3.15)$$

$s_i(x)$  es un vector columna donde cada elemento ya sea  $s_{1l}(x)$  o  $s_{2g}(x)$  contienen  $l = 1, 2, \dots, k, g = 1, 2, \dots, d$  combinaciones de orden superior de las variables de estado y señales de entrada evaluadas en una función sigmoideal.

Finalmente,  $W_1$  y  $W_2$  son matrices de dimensiones  $m \cdot q \times k$  y  $m \cdot q \times d$  respectivamente, que contienen los pesos de la red neuronal.  $W_i$  tiene la forma de  $W_i = \begin{bmatrix} W_{i11} & \dots & W_{i1m} \end{bmatrix}^T$ ,

donde cada  $W_{1j}$  es una matriz de tamaño  $q \times k$  con la siguiente estructura

$$W_{1j} = \begin{bmatrix} w_{1j}^{11} & w_{1j}^{12} & \cdots & w_{1j}^{1k} \\ w_{1j}^{21} & w_{1j}^{22} & \cdots & w_{1j}^{2k} \\ \cdots & \cdots & \cdots & \cdots \\ w_{1j}^{q1} & w_{1j}^{q2} & \cdots & w_{1j}^{qk} \end{bmatrix} \quad (3.16)$$

y de forma general se tiene

$$W_1 = \begin{bmatrix} w_{11}^{11} & w_{11}^{12} & \cdots & w_{11}^{1k} \\ w_{11}^{21} & w_{11}^{22} & \cdots & w_{11}^{2k} \\ \cdots & \cdots & \cdots & \cdots \\ w_{11}^{q1} & w_{11}^{q2} & \cdots & w_{11}^{qk} \\ \cdots & \cdots & \cdots & \cdots \\ w_{1m}^{11} & w_{1m}^{12} & \cdots & w_{1m}^{1k} \\ w_{1m}^{21} & w_{1m}^{22} & \cdots & w_{1m}^{2k} \\ \cdots & \cdots & \cdots & \cdots \\ w_{1m}^{q1} & w_{1m}^{q2} & \cdots & w_{1m}^{qk} \end{bmatrix} \quad (3.17)$$

Las matrices  $W_1$  y  $W_2$  comparten la misma forma, sólo cambian el tamaño de la dimensión  $k$  por  $d$ , que se refiere al número de entradas a las que se conecta el nodo de la variable de estado. Estas dimensiones podrían ser iguales para mayor simpleza, pero para ahorrar costo computacional, se elige una  $d > k$  debido a que debido al modelo que se busca aproximar, generalmente se necesitan más entradas de orden superior para estimar la velocidad que para estimar la posición.

### 3.2.2. Estrategia de control backstepping utilizando FRHONNs

El objetivo es diseñar una ley de control por retroalimentación que haga que el sistema siga una trayectoria deseada. Teniendo como error de identificación a  $\xi = x - \chi$  y como error de seguimiento a  $\epsilon = \chi - x_d$ , el error de seguimiento de las salidas queda reescrito como  $\tilde{\chi} = \xi + \epsilon$  y la dinámica del error queda dada por

$$\dot{\tilde{\chi}} = \dot{\xi} + \dot{\epsilon} \quad (3.18)$$

Primeramente, se analizan las dinámicas del error de identificación y los pesos de la red neuro-difusa. Para esto, considere la función candidata de Lyapunov

$$V(\xi, \phi_j) = \frac{1}{2} \sum_{j=1}^m (\xi_j^2 + \phi_j^T \Gamma_j^{-1} \phi_j) \quad (3.19)$$

donde  $\phi_j = w_j - w_j^*$ , con  $w_j^*$  los pesos óptimos de la red. La derivada de la función de Lyapunov está dada por

$$\dot{V}(\xi, \tilde{w}) = \sum_{j=1}^m (\xi_j \dot{\xi}_j + \dot{\phi}_j^T \Gamma_j^{-1} \phi_j) \quad (3.20)$$

donde  $\dot{\phi}_j = \dot{w}_j$  al ser  $w_j^*$  una constante. Al evaluar la derivada de la función candidata de Lyapunov se obtiene

$$\begin{aligned} \dot{V}(\xi, \phi) &= \sum_{j=1}^m (\xi_j - a_j \xi_j + \tilde{x}_j \phi_{jk}^T s_{jk} + \tilde{x}_j \dot{\phi}_j^T \Gamma_j^{-1} - \Gamma_{jk} s_{jk} \xi_j) \\ \dot{V}(\xi, \tilde{w}) &= \sum_{j=1}^m (-a_j \xi_j^2 + \xi_j \tilde{x}_j \phi_{jk}^T s_{jk} - \tilde{x}_j \dot{\phi}_j^T \Gamma_j^{-1} \phi_j) \\ \dot{V}(\xi, \tilde{w}) &= \sum_{j=1}^m -a_j \xi_j^2 \leq 0 \end{aligned} \quad (3.21)$$

la cual al ser semidefinida negativa, garantiza que tanto el error de identificación como los pesos de la red neuro-difusa son acotados.

Ahora, se procede a diseñar la estrategia de control basada en el método de backstepping que estabilice el error de seguimiento, definido como

$$\begin{aligned} \epsilon_1 &= x_1 - \chi_{1d} \\ \epsilon_2 &= x_2 - \chi_{2d} \end{aligned} \quad (3.22)$$

La dinámica del error de seguimiento está dada por

$$\begin{aligned} \dot{\epsilon}_1 &= \dot{x}_1 - \dot{\chi}_{1d} \\ \dot{\epsilon}_1 &= -a_1 \epsilon_1 - a_1 \chi_{1d} + \tilde{x}_j w_1^T s_1 + x_2 - \dot{\chi}_{2d} \\ \dot{\epsilon}_1 &= -a_1 \epsilon_1 - a_1 \chi_{1d} + \tilde{x}_j w_1^T s_1 + \epsilon_2 \\ \dot{\epsilon}_2 &= \dot{x}_2 - \dot{\chi}_{2d} \\ \dot{\epsilon}_2 &= -a_2 \epsilon_2 - a_2 \chi_{2d} + \tilde{x}_j w_2^T s_2 + u - \dot{\chi}_{2d} \end{aligned} \quad (3.23)$$

Para este controlador se utilizará el método del *backstepping*, considerando a  $\epsilon_2$  como una entrada de control virtual, se diseñará una entrada de control  $\epsilon_2 = \alpha_2$  para estabilizar el origen  $\epsilon_1 = 0$ .

Escogiendo la entrada de control virtual como

$$\epsilon_2 = \alpha_2 = a_1 \chi_{1d} - \tilde{x}_j w_1^T s_1 - c_1 \epsilon_1 \quad (3.24)$$

los términos no lineales de  $\epsilon_1$  son cancelados y entonces queda como

$$\dot{\epsilon}_1 = -a_1 \epsilon_1 - c_1 \epsilon_1 \quad (3.25)$$

Para comprobar la estabilidad del error  $\epsilon_1$ , se propone la siguiente función candidata de Lyapunov

$$V_1(\epsilon_1, \xi, \phi) = V(\chi, \phi) + \frac{1}{2} \epsilon_1^2 \quad (3.26)$$

cuya derivada temporal está dada por

$$\dot{V}_1(\epsilon_1, \xi, \phi) = \dot{V}(\xi, \phi) + \epsilon_1 \dot{\epsilon}_1 \quad (3.27)$$

que al ser evaluada a lo largo de la trayectoria de la ecuación (3.25) queda como

$$\dot{V}_1(\epsilon_1, \xi, \phi) = \dot{V}(\xi, \phi) + \epsilon_1 (-a_1 \epsilon_1 - c_1 \epsilon_1) \quad (3.28)$$

que puede ser reescrita como

$$\dot{V}_1(\epsilon_1, \xi, \phi) = \dot{V}(\xi, \phi) - (a_1 + c_1) \epsilon_1^2 \quad (3.29)$$

Continuando el método del *backstepping*, se hace el cambio de variable

$$\zeta_2 = \epsilon_2 - \alpha_2 = \epsilon_2 - a_1 \chi_{1d} + \tilde{x}_j w_1^T s_1 + c_1 \epsilon_1 \quad (3.30)$$

despejando  $\epsilon_2$  se tiene que

$$\epsilon_2 = \zeta_2 + \alpha_2 = \zeta_2 + a_1 \chi_{1d} - \tilde{x}_j w_1^T s_1 - c_1 \epsilon_1 \quad (3.31)$$

sustituyendo esta ecuación en la ecuación (3.23) se tiene

$$\dot{\epsilon}_1 = -a_1\epsilon_1 - c_1\epsilon_1 + \zeta_2 \quad (3.32)$$

La derivada de la ecuación (3.24) da

$$\dot{\alpha}_2 = a_1\chi_{2d} - \tilde{x}_j\dot{w}_1^T s_1 - \dot{s}_1^T \tilde{x}_j w_1 - c_1\dot{\epsilon}_1 \quad (3.33)$$

sustituyendo la ecuación (3.32) se obtiene

$$\dot{\alpha}_2 = a_1\chi_{2d} - \tilde{x}_j\dot{w}_1^T s_1 - \dot{s}_1^T \tilde{x}_j w_1 - c_1(-a_1\epsilon_1 - c_1\epsilon_1 + \zeta_2) \quad (3.34)$$

La derivada de  $\zeta$  queda dada por

$$\dot{\zeta}_2 = \dot{\epsilon}_2 - \dot{\alpha}_2 \quad (3.35)$$

y sustituyendo las ecuaciones (3.23) y (3.34) , queda reescrito como

$$\begin{aligned} \dot{\zeta}_2 = & -a^2\epsilon_2 - a_2\chi_{2d} + \tilde{x}_j w_2^T s_2 + u - \dot{\chi}_{2d} - a_1\chi_{2d} \\ & + \tilde{x}_j\dot{w}_1^T s_1 + \dot{s}_1^T \tilde{x}_j w_1 + c_1(-a_1\epsilon_1 - c_1\epsilon_1 + \zeta_2) \end{aligned} \quad (3.36)$$

Entonces el sistema puede ser reescrito a la forma

$$\begin{aligned} \dot{\epsilon}_1 = & -a_1\epsilon_1 - c_1\epsilon_1 + \zeta_2 \\ \dot{\zeta}_2 = & -a_2\epsilon_2 - a_2\chi_{2d} + \tilde{x}_j w_2^T s_2 + u - \dot{\chi}_{2d} - a_1\chi_{2d} + \dot{w}_1^T s_1 + \dot{s}_1^T \tilde{x}_j w_1 \\ & + c_1(-a_1\epsilon_1 - c_1\epsilon_1 + \zeta_2) \end{aligned} \quad (3.37)$$

Ahora, para encontrar la entrada de control que estabilice dicho sistema se propone una función candidata de Lyapunov, que será una expansión de la última función de Lyapunov

$$V_2(\zeta_2, \epsilon_1, \chi, \psi) = V(\chi, \psi) + \frac{1}{2}\epsilon_1^2 + \frac{1}{2}\zeta_2^2 \quad (3.38)$$

La derivada de la función de Lyapunov a lo largo de las trayectorias de la ecuación (3.36) da

$$\begin{aligned}\dot{V}_2(\zeta_2, \epsilon_1, \xi, \psi) &= \dot{V}(\xi, \psi) + \epsilon_1 \dot{\epsilon}_1 + \zeta_2 \dot{\zeta}_2 \\ \dot{V}_2(\zeta_2, \epsilon_1, \xi, \psi) &= \dot{V}(\xi, \psi) + \epsilon_1(-a_1 \epsilon_1 - c_1 \dot{\epsilon}_1) + \zeta_2^i(-a^2 \epsilon_2 - a_2 \chi_{2d}^i + \tilde{x}_j w_2^T s_2 + u - \dot{\chi}_{2d} \\ &\quad - a_1 \chi_{2d} + \tilde{x}_j \dot{w}_1^T s_1 + \dot{s}_1^T \tilde{x}_j w_1 + c_1(-a_1^i \epsilon_1 - c_1 \dot{\epsilon}_1 + \zeta_2))\end{aligned}\quad (3.39)$$

expandiendo y ordenando los términos se tiene que

$$\begin{aligned}\dot{V}_2(\zeta_2, \epsilon_1, \xi, \psi) &= \dot{V}(\xi, \psi) - a_1 \epsilon_1^2 - c_1 \dot{\epsilon}_1^2 + \zeta_2(\epsilon_1 - a_2 \epsilon_2 - a_2 \chi_{2d} + \tilde{x}_j w_2^T s_2 + u - \dot{\chi}_{2d} \\ &\quad - a_1 \chi_{2d} + \tilde{x}_j \dot{w}_1^T s_1 + \dot{s}_1^T \tilde{x}_j w_1 + c_1(-a_1 \epsilon_1 - c_1 \dot{\epsilon}_1 + \zeta_2))\end{aligned}\quad (3.40)$$

la entrada de control se obtendrá con la finalidad de hacer la ecuación (3.40) semidefinida negativa, por lo tanto, la entrada de control queda como

$$\begin{aligned}u &= -\epsilon_1 + a_2 \epsilon_2 + a_2 \chi_{2d} - \tilde{x}_j w_2^T s_2 + \dot{\chi}_{2d} + a_1 \chi_{2d} - \tilde{x}_j \dot{w}_1^T s_1 \\ &\quad - \dot{s}_1^T \tilde{x}_j w_1 - c_1(-a_1 \epsilon_1 - c_1 \dot{\epsilon}_1 + \zeta_2) - c_2 \zeta_2\end{aligned}\quad (3.41)$$

que al sustituir en la ecuación (3.39) da como resultado

$$\dot{V}_2(\zeta_2, \epsilon_1, \xi, \psi) = \dot{V}(\xi, \psi) - (a_1 + c_1) \epsilon_1^2 - (a_2 + c_2) \zeta_2^2 \leq 0 \quad (3.42)$$

lo cual garantiza que el error de seguimiento así como el error de la entrada de control virtual son acotados.

### 3.3. Algoritmo Super-Twisting

El algoritmo Super-Twisting (**STA**) es un conocido algoritmo de control de modo deslizante de segundo orden introducido por Levant[41]. Se considera adecuado para aquellos sistemas que tienen un grado relativo uno con respecto a la variable de deslizamiento. El algoritmo garantiza la robustez con respecto a los errores de modelado y las perturbaciones externas al tiempo que reduce el fenómeno de vibración que se encuentra en el algoritmo de control de modo deslizante (**SMC**) convencional. El **STA** produce una acción de control continua porque el término de conmutación está oculto por el término integral utilizado en

el ecuación de control. Si consideramos un sistema afín genérico de la siguiente forma

$$\dot{x} = f(x, t) + g(x, t)u \quad (3.43)$$

y, si definimos la variable deslizante  $s$  tal que la señal de control aparece en la primera derivada con respecto al tiempo  $\dot{s}$  de la función de conmutación, el sistema tiene un grado relativo 1

$$\dot{s}(x, t) = h(x, t) + g(x, t)u \quad (3.44)$$

La entrada de control **STA**, puede definirse como sigue

$$u_{STA} = -K_1|s|^{\frac{1}{2}}\text{sign}(s) + u_1 \quad (3.45)$$

$$\dot{u}_1 = \begin{cases} -u_{STA} & , |u_{STA}| > U_M \\ -K_2\text{sign}(s) & , |u_{STA}| \leq U_M \end{cases} \quad (3.46)$$

donde algunas suposiciones son necesarias sobre la cota de la función  $h(x, t)$  y  $g(x, t)$

$$|\dot{h}| + U_M|\dot{g}| \leq C, \quad 0 \leq K_m \leq g(x, t) \leq K_M, \quad |h/g| < qU_M, \quad 0 < q < 1 \quad (3.47)$$

donde  $C$ ,  $K_M$ ,  $K_m$ ,  $U_M$  y  $q$  son constantes positivas que definen la función y los límites de control, mientras que las constantes positivas  $K_2$  y  $K_1$  son los parámetros de ajuste del controlador. Para garantizar el inicio del modo deslizante en el sistema debemos tener  $K_M > C$  y  $K_1$  lo suficientemente grandes. Además, si el control entra en el límite  $[-U_M, U_M]$ , permanece allí. Desde el punto de vista práctico, una condición suficiente para la elección del parámetro  $K_1$  viene dada por

$$K_1 > \sqrt{\frac{2}{(K_m K_2 - C)}} \cdot \frac{(K_m K_2 + C) K_M (1 + q)}{k_m^2 (1 - q)} \quad (3.48)$$

donde las constantes positivas  $K_1$  y  $K_2$  son los parámetros de sintonización de la **STA**. Finalmente, la constante  $U_M$  define el valor máximo de la acción de control.

### **3.4. Conclusiones**

En este capítulo se presentó el diseño de las estrategias de control inteligentes y robustas para el vehículo BlueROV2. Un controlador (FRHONN) basado en redes neuronales recurrentes difusas de orden superior y un algoritmo super-twisting para solucionar la problemática de la navegación relativa. También se mostraron las demostraciones de estabilidad y los parámetros a afinar para mejorar el desempeño del controlador. En el siguiente capítulo se mostrarán los resultados de las pruebas utilizando estos controladores.

---

---

# Desarrollo del método de estimación basado en visión

---

En este capítulo se presenta el desarrollo del método de estimación de la posición relativa basada en visión artificial. El algoritmo de visión puede detectar y posteriormente, estimar la posición y velocidad relativa del vehículo BlueROV2 con respecto un marcador hecho con luces con una disposición geométrica determinada. La selección de las luces a detectar y la disposición geométrica entre ellas se decidió en base a las condiciones de iluminación, turbiedad del agua y tamaño de las albercas de pruebas del laboratorio LAB-STICC de ENSTA Bretagne y UMI LAFMIA en las que se realizaron las pruebas de estimación y seguimiento.

El algoritmo de visión desarrollado está basado en la segmentación y detección de contornos para la detección de las luces debido a la facilidad con la que resaltan las luces en ambientes subacuáticos y tiene una serie de filtros para distinguir las luces que pertenecen al marcador de otras luces externas al sistema que pueden aparecer en la práctica, esta característica es especialmente útil para que el algoritmo pueda ser empleado cerca de la superficie debido a que el reflejo de las luces puede hacer que un algoritmo convencional que no tenga contemplada la aparición de luces ajenas pueda tomar lecturas equivocadas que puedan resultar peligrosas, esto le otorga al algoritmo un grado de robustez. Otra medida tomada en el diseño del algoritmo de visión para aumentar la robustez del algoritmo es el uso de luces redundantes para la estimación de la posición. Debido a que la estimación de la posición se puede realizar únicamente con un par de luces, los pares extra en el marcador sirven en el escenario en el que sólo una parte del marcador es visible por oclusión o por fallas de hardware. Cabe señalar que el algoritmo es capaz de funcionar con solo dos marcadores visibles,

---

esto ha sido comprobado durante los experimentos realizados.

## 4.1. Descripción de la cámara monocular

La cámara a bordo del BlueROV2 es una cámara **HD USB** para baja luminosidad de un ángulo de visión amplio con baja distorsión. El sensor que utiliza es el sensor *Sony IMX323* con relativamente una baja cantidad de píxeles (2MP) y de alta definición, esto quiere decir que el tamaño físico de cada píxel es grande y esto permite una sensibilidad a la luz alta, por lo que es ideal para utilizarse a bajas profundidades y aun así recibir la poca luminosidad disponible en el ambiente. La cámara cuenta con un chip de compresión H.264 a bordo para que el *stream* de video enviado por **UDP** a través del cable *tether* del BlueROV2 ya esté comprimido y únicamente se deba descomprimir en la estación de tierra con un pipeline de *Gstreamer*. La Figura 4.1 muestra la cámara utilizada

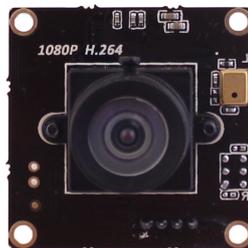


Figura 4.1: Cámara a bordo del vehículo BlueROV2

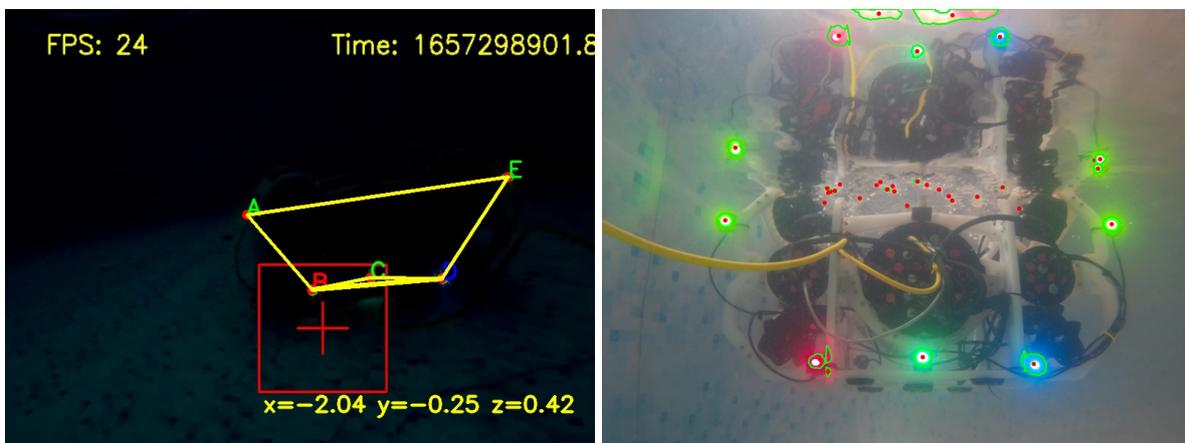
---

### 4.1.1. Requisitos del sistema de navegación relativa

La amplia variedad de condiciones en ambientes subacuáticos dicta una necesidad para que el sistema de navegación relativa sea versátil para que pueda tener un uso efectivo en la práctica. A diferencia de los ambientes de desarrollo y experimentación controlados, como en las albercas de pruebas, las condiciones de visión en los escenarios reales pueden tener las siguientes variantes que deben considerarse

- Luminosidad del ambiente.
- Turbidez del agua.
- Reflejos en la superficie.
- Fuentes de luz ajenas.
- Oclusión.

Por lo tanto, para el desarrollo del sistema de visión se deben buscar soluciones para cada uno de estos puntos. Para la luminosidad del ambiente se aprovechan las capacidades de la cámara a bordo del BlueROV2 para funcionar en condiciones de baja luminosidad, y así proponer un sistema de visión que no sólo sea capaz de funcionar a diferentes horas del día, sino que también sea capaz de funcionar a diferentes niveles de profundidad. La Figura 4.2 muestra la imagen obtenida con la cámara monocular a bordo del BlueROV2 a diferentes niveles de luminosidad.



(a) Imagen a baja iluminacion

(b) Imagen a alta iluminacion

Figura 4.2: Se muestra una imagen obtenida de la camara monocular del Bluerov a diferentes niveles de iluminacion

Al considerar la turbiedad del agua, se descartan sistemas de visión que requieran utilizar los faros a bordo del BlueROV2 debido a que, en ambientes turbios, la luz de los faros puede iluminar las partículas presentes en el agua turbia y la visión de la cámara es obstruida por estas partículas iluminadas hasta puntos donde la cámara capta imágenes blancas y ruidosas. Por esta razón se debe elegir un sistema de visión en el que la cámara no necesite de luces para observar a su objetivo como en el caso del uso de marcadores fiduciales que no podrían ser visibles a baja luminosidad a menos que se utilicen los faros del BlueROV2. La solución para esto es utilizar un marcador hecho de luces **LED**, cuya disposición geométrica es conocida y por lo tanto se pueden usar las distancias entre cada par de **LEDs** para estimar la posición de la cámara que los observa. En la siguiente sección se muestra el marcador diseñado.

Un algoritmo que únicamente detecta las luces puede ser susceptible a detectar luces que no son parte del sistema como luces que pertenecen a otros vehículos o que están colocadas como indicador en estructuras subacuáticas, así mismo se podrían detectar por error el reflejo de las luces en la superficie. Estos errores de detección podrían hacer que el algoritmo que estime la posición se equivoque drásticamente y causar estragos en la navegación, por lo que el algoritmo debe identificar las luces que pertenecen al sistema de entre un grupo de luces candidatas detectadas. Se propone un método para la identificación de las luces que pertenecen al marcador basado en una serie de filtros para las luces candidatas, estos filtran las luces por color, tamaño, forma y por disposición geométrica que forman entre ellas hasta encontrar las luces que pertenecen al marcador. Esta técnica además de filtrar las fuentes de luz ajenas permite reconocer cada par de luces del marcador de forma independiente y así se tiene una solución para la oclusión.

---

### 4.1.2. Calibración de la cámara

Una parte vital para el correcto funcionamiento del algoritmo de visión es que la cámara esté correctamente calibrada. Para la calibración se usó un script de *OpenCV* con la técnica para detectar las esquinas de un tablero de ajedrez. Cabe aclarar que la calibración de la cámara debe hacerse con fotos tomadas debajo del agua, esto se debe a que la cámara está situada detrás de un domo en la cápsula principal del BlueROV2, al haber un cambio de medio de agua a aire, la luz se refracta, esto ocasiona que el campo de visión disminuya como puede verse en la Figura 4.3. La geometría del domo ayuda a eliminar esa refracción, pero esto es considerando que la cámara esté correctamente centrada y además el domo actúa como un lente divergente y aparece otro tipo de distorsión dependiendo del radio del domo. Los datos obtenidos en la calibración son siguientes

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

donde  $K$  es la matriz intrínseca,  $f_x$  y  $f_y$  son las longitudes focales,  $c_x$  y  $c_y$  son las coordenadas del centro óptico en el plano de la imagen.

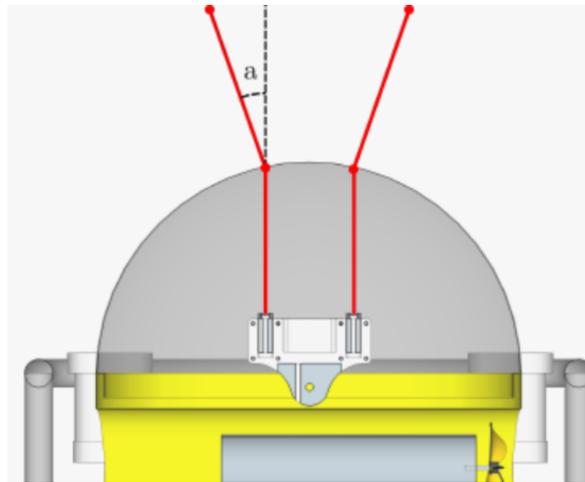


Figura 4.3: Difracción en cambios de medio

## 4.2. Descripción del método propuesto

Para el marcador se eligieron 5 luces **LED** Luxeon Star debido a su tamaño compacto y su capacidad de iluminación que es suficiente para las albercas de pruebas en las que se implementó el algoritmo de visión. La Figura 4.4 muestra la disposición de las luces propuesta. El marcador se colocó sobre otro **AUV** submarino, el **AUV** OctoSub, esto con el propósito de que las pruebas se puedan hacer tanto para seguimiento de objetivos estáticos como de objetivos en movimiento.

Se utilizaron colores **RGB** diferentes para poder identificar, a través de los colores, cada luz específica del marcador; de esta forma se tienen más restricciones a la hora de determinar qué luces pertenecen al marcador entre todas las detectadas.

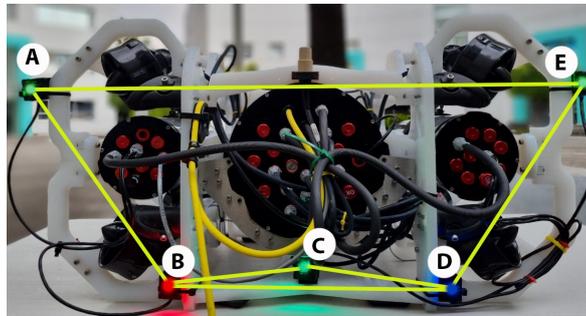


Figura 4.4: Marcador de **LEDs** propuesto para usar como referencia en la estimación de la posición relativa. Los **LEDs** están conectados al **AUV** submarino OctoSub

La distancia entre cada par de **LEDs** es designada como  $d_i$ , donde  $i = 1, \dots, n$  indica un par determinado de luces en el marcador y  $n$  es el número total de pares usados para la estimación, que en este caso se eligieron  $n = 6$  pares.

Para hacer la estimación de la posición relativa del BlueROV2 con respecto al marcador, se consideran dos sistemas coordenados: el primero es el sistema coordenado de la cámara  $C$ , cuyo origen  $O^C$  se encuentra en el centro focal de la cámara a bordo del BlueROV2; el segundo sistema coordenado es el sistema del marcador  $O$ , que está trazado en el plano formado por las luces y con origen  $O^O$  al centro de la cápsula principal del vehículo OctoSub. Los dos sistemas coordenados son considerados paralelos entre sí y también que los dos vehículos tienen la misma orientación. La Figura 4.5 muestra un diagrama con los sistemas coordenados.

La distancia entre los sistemas coordenados  $L^{CO}$  es estimada usando cada uno de los pares individuales detectados y presentes en el fotograma, se realiza un promedio de las

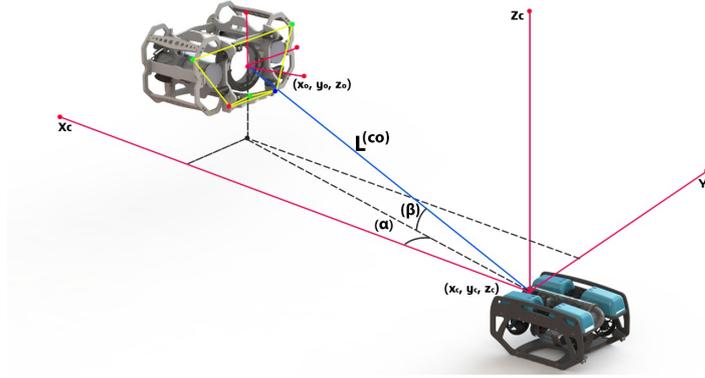


Figura 4.5: Diagrama de la estimación de la distancia  $L^{CO}$  que hay entre los marcos de referencia de la cámara  $C$  y del marcador  $O$

estimaciones de cada par para utilizar la redundancia de las luces para dar robustez a la estimación y disminuir el error causado por el desfase en la detección del centroide de cada **LED**, con lo cual la distancia  $L^{CO}$  se calcula como

$$L^{CO} = \frac{\sum_i^n l_i}{n} \quad (4.2)$$

donde  $l_i$  es la distancia estimada usando cada uno de los  $i$  pares de luces. Esta estimación se realiza usando la relación lineal que hay entre la longitud de cada par de luces en píxeles y la longitud de cada par que ya es conocida. La estimación está dada por la siguiente ecuación

$$l_i^{CO} = \frac{f \times d_i \times k_i}{s} \quad (4.3)$$

donde  $f$  es la longitud focal obtenida durante el proceso de calibración,  $s$  es la altura del sensor de la cámara que es dado por el fabricante,  $d_i$  es la distancia conocida entre cada par de luces y  $k_i$  es el factor de escala dado por la tasa entre la distancia en píxeles entre cada par de luces en el fotograma y el tamaño del frame.

También, se calculan los ángulos formados entre los dos sistemas coordenados,  $\alpha^C$ , que está trazado sobre el plano  $XY$  y  $\beta$ , que está trazado sobre el plano  $XZ$ . Estos ángulos se obtienen usando una relación que hay entre el ángulo del campo de visión y el porcentaje de la distancia en píxeles con respecto al tamaño del fotograma

$$\alpha = p_x \times \frac{FOV_x}{2c_x} \quad (4.4)$$

$$\beta = p_y \times \frac{FOV_y}{2c_y} \quad (4.5)$$

Conociendo los ángulos y la distancia euclidiana entre los sistemas coordenados, se puede lograr el posicionamiento del BlueROV2 con respecto del marcador

$$x^C = L^{CO} \cos(\alpha) \quad (4.6)$$

$$y^C = L^{CO} \sin(\alpha) \quad (4.7)$$

$$z^C = L^{CO} \cos(\beta) \quad (4.8)$$

Tomando en cuenta que se asume que ambos vehículos tienen la misma orientación, la orientación relativa y su derivada son iguales a 0, entonces para completar la estimación del vector de estado, hace falta obtener la velocidad relativa lineal entre el marcador y el BlueROV2.

Como la intención es proponer un método basado en visión artificial que estime tanto posición como velocidad, las velocidades se deben obtener derivando las posiciones estimadas por el algoritmo propuesto. Se proponen dos métodos de derivación a utilizar

- Derivada de Euler, que es definida por la siguiente ecuación:

$$\dot{f}(t) = \frac{f(t) - f(t-1)}{\Delta t} \quad (4.9)$$

- Diferenciador de Levant de primer orden [42]

### 4.3. Descripción del algoritmo de visión

El algoritmo fue desarrollado para ser procesado en la estación de tierra usando **ROS** y **QGroundControl** para comunicarse con el **AUV**. Los principales pasos del algoritmo de visión se presentan en el Algoritmo 1. Se utiliza **GStreamer** para obtener el stream de video en la estación de tierra y procesar la imagen usando **OpenCV** y **Python**. El stream de video se hace a **25 FPS** y una resolución de **800x600**.

---

**Algorithm 1** Algoritmo de estimación de la posición

---

```
1: while not rospy is shutdown() do
2:   Captura del stream de video
3:   Lectura de imagen
4:   for frame do
5:     Corrección de la distorsión
6:     Conversión a espacio de color HSV
7:     Segmentación
8:     Detección de contornos
9:     for contour in contours do
10:      Filtrado
11:      Detección de color
12:      Registro de luces candidatas
13:     for light in lights do
14:       Comprobación de la disposición geométrica
15:       Calculo de distancia
16:       Estimación de coordenadas
17:       Publicación en ROS
```

---

A continuación, se presenta una descripción detallada de cada etapa del algoritmo

#### 4.3.1. Captura del stream de vídeo

El script de **Python** para obtener el stream de video consiste en usar un pipeline multi-fuente de **GStreamer** que obtiene los datos del vídeo enviados por el tether con un protocolo de comunicación **UDP**. Se elige multi-fuente para enviar el vídeo por diferentes puertos para obtener el vídeo en **QGroundControl** y en el algoritmo de visión al mismo tiempo. El pipeline de **GStreamer** se configuró para recibir el stream nativo, ya que de fuente está a una resolución baja que se puede procesar fácilmente en la estación de tierra usando un **GPU**.

---

### 4.3.2. Lectura de imagen

El fotograma emitido por GStreamer es leído en el algoritmo para procesarse con funciones de OpenCV. Si no hay un fotograma disponible al momento en el que se manda a llamar la función, se salta el resto del loop hasta que haya un fotograma disponible. En la Figura 4.6 se muestra un frame de la imagen nativa.

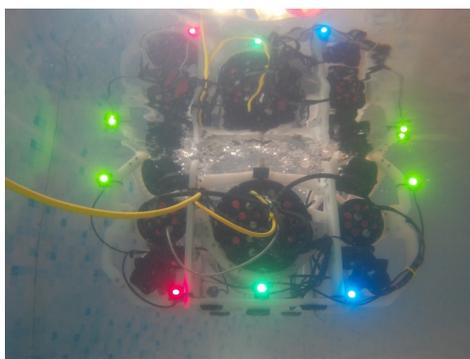


Figura 4.6: Imagen nativa obtenida de la cámara monocular a bordo del BlueROV2

### 4.3.3. Conversión a espacio de color HSV

La conversión al espacio de color **HSV** (Hue, Saturation and Value) es realizada al frame para hacer una segmentación basada en color, que posteriormente servirá para detectar tanto las fuentes de luz en el frame como su color. El espacio de color nativo del fotograma es **RGB** (Red, Green and Blue), el cual tiene un método aditivo para describir los colores y por ello tiene un rendimiento menor al del **HSV** en el procesamiento de imagen. Al usar el espacio de color **HSV** es más fácil segmentar los tonos de color deseados usando los tres parámetros, tono, saturación y brillo. La Figura 4.7 muestra el frame en el espacio de color **HSV**.

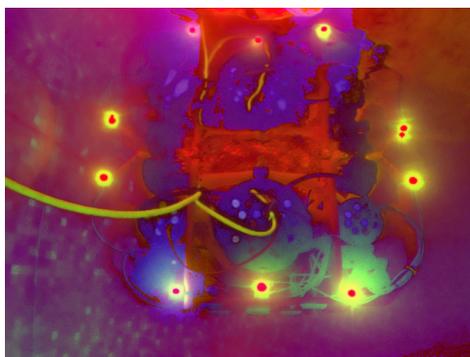


Figura 4.7: Imagen en el espacio de color HSV.

#### 4.3.4. Segmentación

Debido a la absorción de la longitud de onda de la luz en el agua, los **LEDs** de cualquier color tienen un centro blanco cuando se observan debajo del agua, por lo que la primera forma de detección es a través de localizar esos centros blancos de alta intensidad en la imagen. Para esto se aplica una máscara que segmenta los tonos blancos de alta intensidad en el fotograma en espacio de color **HSV**. La Figura 4.8 muestra la imagen binarizada para segmentar las fuentes de luz blanca de alta intensidad.

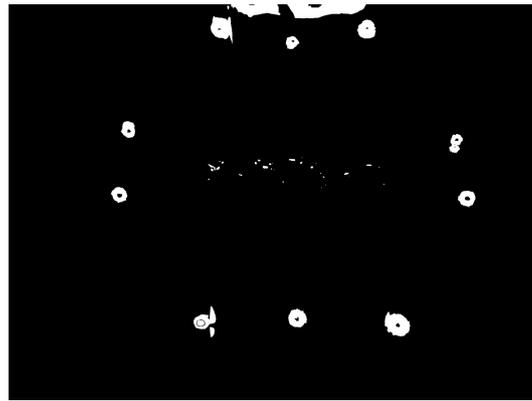


Figura 4.8: Máscara usada en la segmentación de las fuentes de luz blanca en la imagen.

#### 4.3.5. Detección de contornos

Se utiliza una función de OpenCV para detectar contornos en la imagen, estos contornos serán tratados como candidatos para ser una luz que pertenece al marcador del sistema de navegación relativa. Posteriormente, se filtrarán estos contornos con respecto a las condiciones de disposición y colores que tienen los **LEDs** del marcador. La Figura 4.9 muestra los contornos detectados en la imagen.

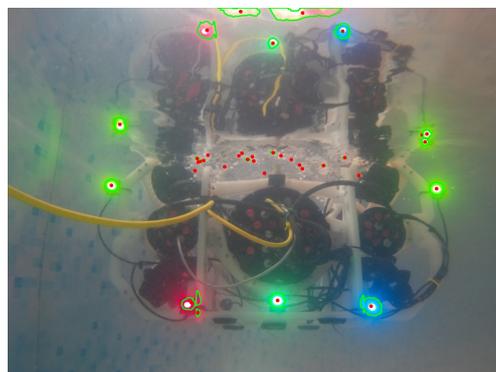


Figura 4.9: Fuentes de luz detectadas en la imagen con la técnica de detección de contornos.

---

### 4.3.6. Filtrado

Los contornos pasan por varios filtros en el siguiente orden

- Filtro de área: Se determinan los límites máximos y mínimos de área que debe tener un contorno que sea una luz del sistema, estos límites se determinan a través del tamaño de los contornos cuando las distancias relativas entre el marcador y la cámara se encuentran en el límite operacional. Siendo el límite mínimo cuando más lejano se encuentran entre sí los vehículos y el límite máximo cuando más cerca se encuentran entre sí.
- Filtro de forma: Como la forma de los contornos que se buscan es circular, se aplican filtros aquí para no contar los contornos que tengan formas aproximadas a polígonos de un límite máximo de lados. Después se filtran los contornos que tengan formas cóncavas usando un límite del porcentaje que debe tener el área del contorno con respecto al área circular máxima que encierra todos los puntos del contorno.
- Filtro de ruido: Para filtrar los puntos blancos que pueden aparecer esporádicamente en la imagen como ruido, se guardan todas las fuentes de luz detectadas en un arreglo para comparar las posiciones actuales de las luces con las posiciones previas y así eliminar las luces que no se encuentran en el frame pasado.

### 4.3.7. Detección de color

Se aplica a la imagen en **HSV** una máscara que segmenta cada uno de los colores **RGB** a una alta intensidad. La Figura 4.10 muestra las imágenes binarias con la segmentación de cada uno de los colores **RGB**. Posteriormente, para detectar el color de cada una de las luces candidatas, se genera una máscara circular con centro en el centroide de cada uno de los contornos detectados. Se realiza una operación bitwise **AND** de la máscara circular y de cada una de las máscaras **RGB**, el propósito de esto es calcular el número de píxeles “on” (valor de 1 binario) que hay de cada uno de los colores segmentados en la vecindad de las luces candidatas, el mayor número de píxeles que coinciden en esta área nos dará el color **RGB** dominante en la vecindad de cada contorno. La Figura 4.11 nos muestra los colores detectados de cada una de las luces candidatas.

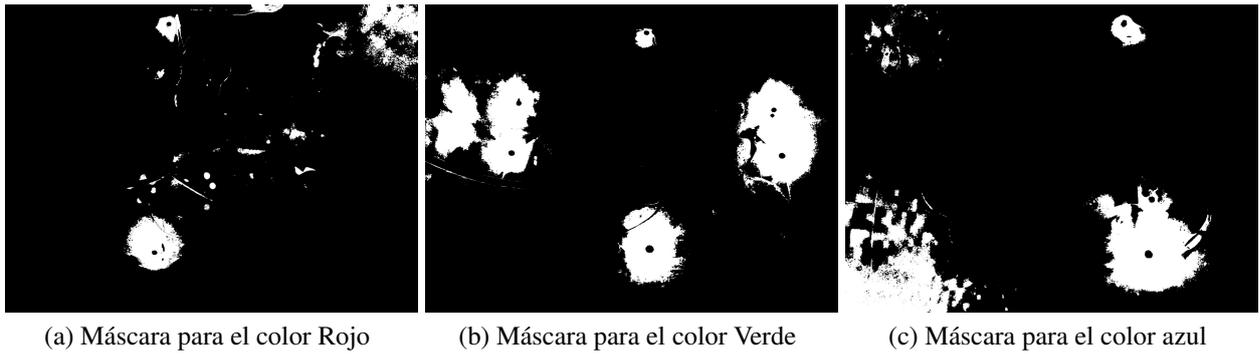


Figura 4.10: Máscaras usadas para detectar el color RGB dominante en la vecindad de los contornos detectados.

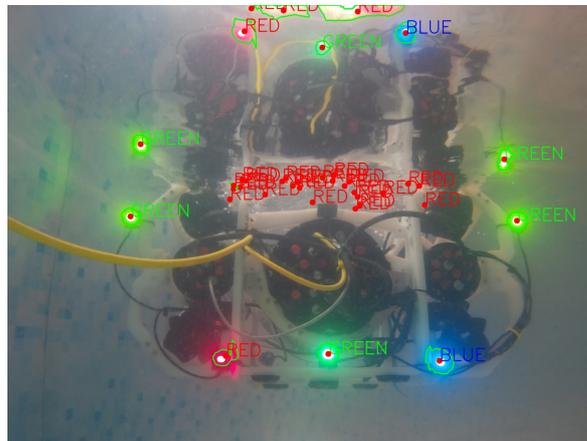


Figura 4.11: Luces detectadas con la etiqueta del color dominante en su vecindad.

#### 4.3.8. Registro de luces candidatas

Después de la aplicación de los filtros y de la detección de color, se guarda toda la información obtenida en arreglos, esta información es: coordenadas del centro de cada luz detectada y color dominante en su vecindad. Esta información permitirá evaluar si pertenecen o no al marcador.

#### 4.3.9. Comprobación de la disposición geométrica

Basado en el color de cada luz candidata, se forman pares de luces para determinar si las luces pertenecen al marcador. Aquí se usan las luces rojas y azul como referencias para encontrar los pares AB, BD, BC, CD y DE primero. Se considera que ambos vehículos mantienen la misma orientación, por lo tanto las condiciones limitantes que se toman en cuenta son:

- La luz roja se encuentra a la izquierda de la luz azul.
- Entre las luces roja y azul se encuentra una luz verde, que también está arriba de esas luces.
- A la izquierda y arriba de la luz roja se encuentra una luz verde.
- A la derecha y arriba de la luz azul se encuentra una luz verde.
- La distancia entre las luces verdes de las esquinas debe ser mayor que las distancias entre los demás pares de luces.

La Figura 4.12 muestra las luces que pertenecen al marcador detectadas en la imagen, también se muestran las demás luces detectadas que no fueron identificadas como parte del marcador.

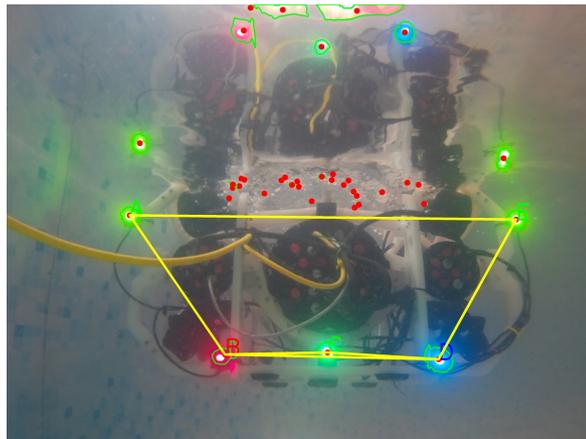


Figura 4.12: Detección del marcador de LEDs entre un ambiente con diversas fuentes de luz. En la imagen se muestran todas las luces detectadas y se muestra el marcador formado por las luces que pasaron los filtros del algoritmo.

#### 4.3.10. Cálculo de la distancia y estimación del estado

En esta parte del algoritmo se realizan los cálculos para el cálculo de la distancia  $L^{CO}$  y la estimación de las coordenadas relativas  $(x, y, z)$ . Estos cálculos se hacen conforme a las ecuaciones (4.6), (4.7) y (4.8). Las posiciones estimadas se guardan en un valor que se sobrescribe cada iteración para tener acceso al dato anterior de cada estimación, esto con la finalidad de usarse en las técnicas de derivación propuestas y así obtener las velocidades lineales.

### **4.3.11. Publicación en ROS**

Las posiciones y velocidades estimadas se publican en un nodo de ROS para usarlos en un script separado para el control del BlueROV2. Este nodo hace la conexión al BlueROV2 a través de MavROS, para enviar las señales vía UDP a través del tether.

## **4.4. Conclusiones**

En este capítulo se presentó el desarrollo del método de estimación de la posición relativa del ROV basada en visión artificial. Se presentaron los requisitos con los que debe cumplir este sistema de navegación y el desarrollo de la propuesta que ataca cada uno de esos puntos. Este algoritmo se probó tanto dentro, como fuera del agua durante su desarrollo para que tuviera la capacidad de utilizarse en una plataforma experimental en tiempo real. En el siguiente capítulo se mostrarán los resultados experimentales del algoritmo de visión relativa en conjunto de una de las estrategias de control expuestas en el capítulo pasado.

---

# Resultados

Antes de realizar las pruebas experimentales con los algoritmos desarrollados, se realizaron pruebas simuladas para validar el funcionamiento de las estrategias de control y el algoritmo de visión desarrollado. Cada una de las pruebas se enfocó en probar y validar el correcto funcionamiento de una parte específica del método de navegación relativa para vehículos submarinos que se propone en esta tesis. En este capítulo se presentarán los resultados de las simulaciones que se hicieron para verificar el desempeño de la identificación y control utilizando la **FRHONNs**, así como la pruebas experimentales del control Super-Twisting. Además de los resultados obtenidos por el algoritmo de visión desarrollado.

## 5.1. Resultados de simulación para las FRHONNs

Los resultados en simulación se realizaron en Matlab para verificar el desempeño de la estrategia FRHONN desarrollada en el Capítulo 3. Para la identificación del modelo dinámico del submarino puede dividirse en  $m = 4$  subsistemas con la forma aproximada

$$\begin{aligned}\dot{\chi}_1 &= \chi_2 \\ \dot{\chi}_2 &= f(\chi_1, \chi_2, u) + g(\chi, \dot{\chi}, u)\end{aligned}\tag{5.1}$$

Se toma el modelo de FRHONN propuesto en la ecuación (3.13) con  $k = 1$  y  $d = 3$  y las siguientes entradas recursivas

$$s_{1j}(x, u) = \left[ s(x_{1j}) \right]^T\tag{5.2}$$

$$s_{2j}(x, u) = \left[ s(x_{1j}) \quad s(x_{2j}) \quad s(x_{1j})s(x_{2j}) \right]^T\tag{5.3}$$

que en su forma vectorial para todos los subsistemas son

$$s_1(x, u) = \begin{bmatrix} s(x_{11}) & s(x_{12}) & s(x_{13}) & s(x_{14}) \end{bmatrix}^T \quad (5.4)$$

$$s_2(x, u) = \begin{bmatrix} s(x_{21}) & s(x_{22}) & s(x_{23}) & s(x_{24}) \end{bmatrix}^T \quad (5.5)$$

donde la función de activación sigmoïdal a utilizar  $s(x, u)$  es la tangente hiperbólica

$$s(x, u) = \tanh(x, u) \quad (5.6)$$

Para llevar a cabo la simulación, se utilizó el modelo dinámico obtenido en el Capítulo 2 presentado en la ecuación (2.2), tomando los parámetros del modelo dinámico del BlueROV2 de la Tabla 5.1 [43].

Tabla 5.1: Parámetros físicos y dinámicos del BlueROV2.

Param.	Valor	Param.	Valor	Param.	Valor
$W$	$[0, 0, 176]^T$	$X_{\dot{u}}$	5.5	$Y_{\dot{v}}$	12.7
$r_b$	$[0, 0, -0.1]^T$	$Z_{\dot{w}}$	14.57	$I_{xx}$	0.16
$B$	$[0, 0, 9.81m_f]^T$	$I_{yy}$	0.16	$I_{zz}$	0.16
$K_{\dot{p}}$	0.002	$M_{\dot{q}}$	0.002	$N_{\dot{r}}$	0.12
$X_u$	3.03	$Y_v$	3.21	$Z_w$	8.18
$K_p$	2	$M_q$	0.1	$N_r$	0.07
$X_{u_c}$	4.03	$Y_{v_c}$	6.21	$Z_{w_c}$	5.18
$K_{p_c}$	0.07	$M_{q_c}$	0.07	$N_{r_c}$	0.07

Las trayectorias deseadas para cada una de las dinámicas del vehículo BlueROV2 son presentadas en la Tabla 5.2

Tabla 5.2: Trayectoria deseada para la validación del algoritmo de identificación

Periodo	$(x_d)_i \quad i = 1 : 3$	$(\dot{x}_d)_i \quad i = 1 : 3$	$(x_d)_4$
0 – 20s	$0.5 \sin(0.25t)$	$0.125 \cos(0.25t)$	$10\pi/180$
20s – 40s	$\sin(0.5t)$	$0.5 \cos(0.5t)$	$-20\pi/180$
40s – 70s	$2 \sin(0.25t) \cos(0.5t)$	$0.5 \cos(0.25t) \cos(0.5t) - \sin(0.25t) \sin(0.5t)$	0

En esta simulación se eligieron  $q = 5$  particiones difusas. El seguimiento de la trayectoria puede observarse en la Figura 5.1 donde puede verse que realiza el seguimiento de trayectoria de forma satisfactoria para las dinámicas del vehículo BlueROV2.

En la Figura 5.2 muestra el error de seguimiento utilizando las **FRHONNs**. Se puede observar que el error de seguimiento es acotado tanto para posición como para velocidad. Cabe

destacar que la simulación es realizada en lazo cerrado con la metodología backstepping basada en el identificador **FRHONNs**.

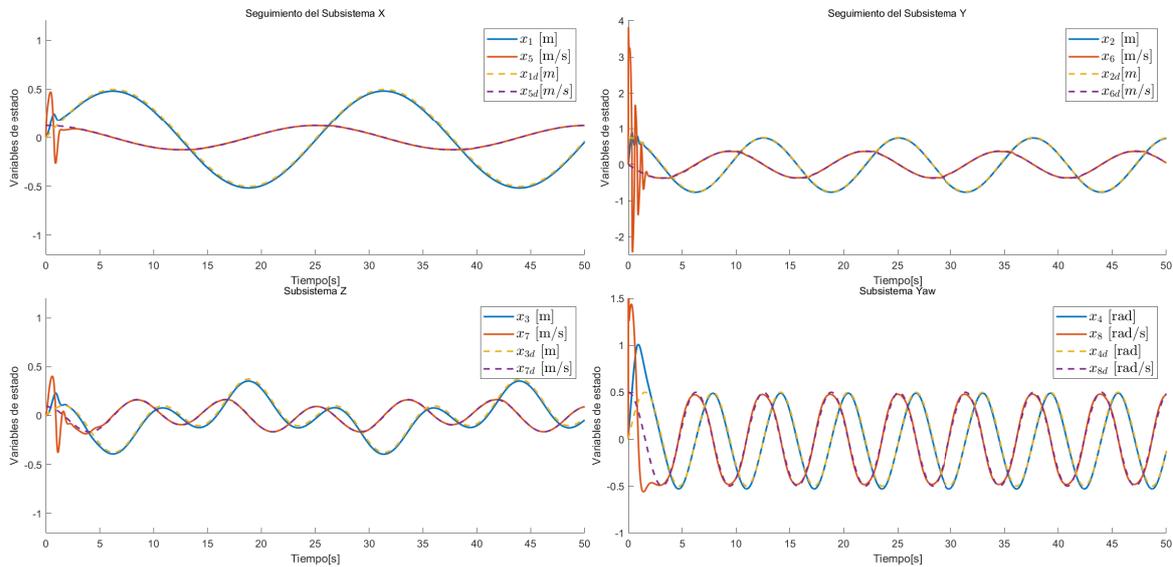


Figura 5.1: Seguimiento de trayectoria con FRHONNs

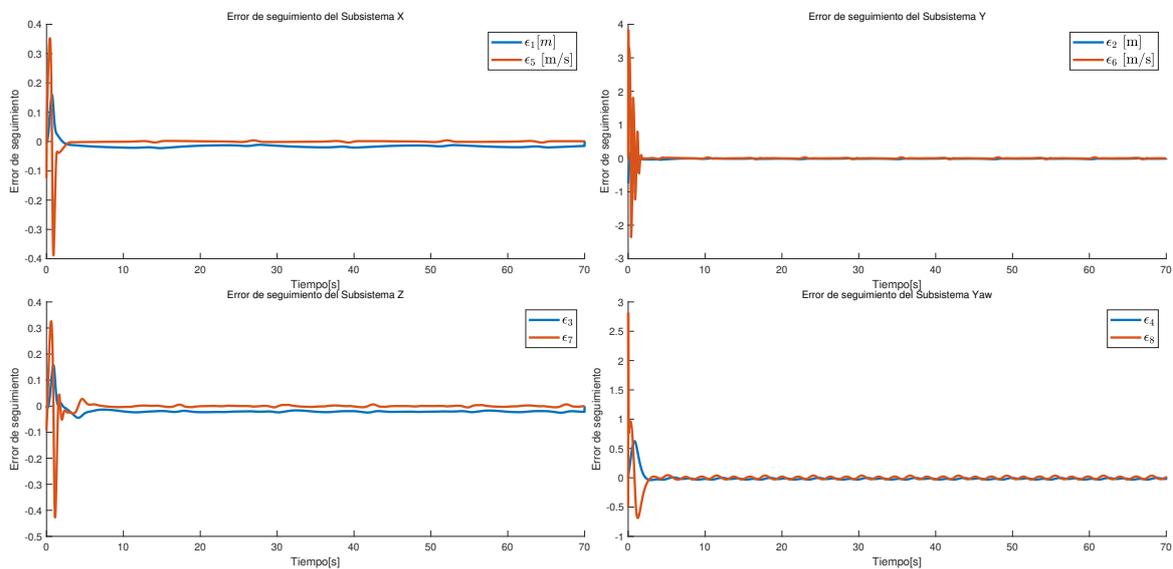


Figura 5.2: Error de seguimiento

En la Figuras 5.3, 5.4, 5.5, 5.6 pueden observarse los pesos adaptables de las FRHONNs para cada subsistema, aquí se puede apreciar que el mismo parámetro de entrenamiento para los pesos da como resultado, diferentes comportamientos para diferentes trayectorias.

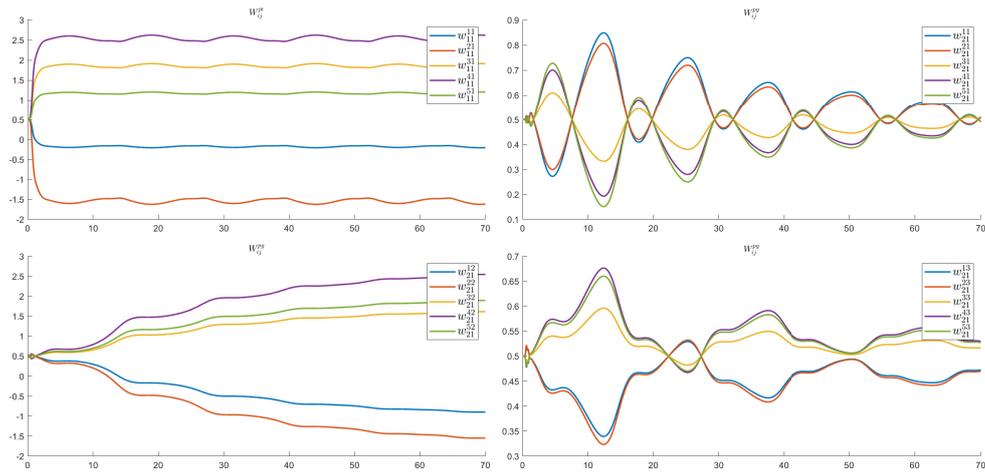


Figura 5.3: Actualización de los pesos de las FRHONNs para el subsistema X

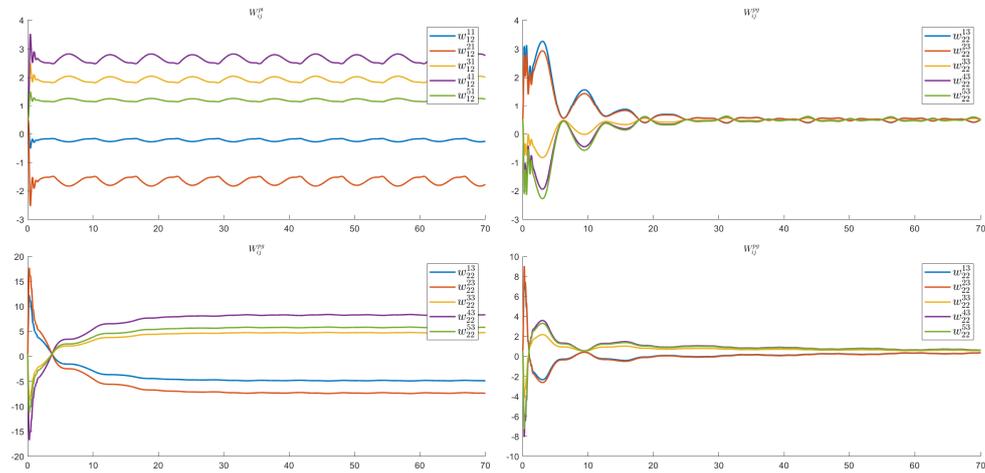


Figura 5.4: Actualización de los pesos de las FRHONNs para el subsistema Y

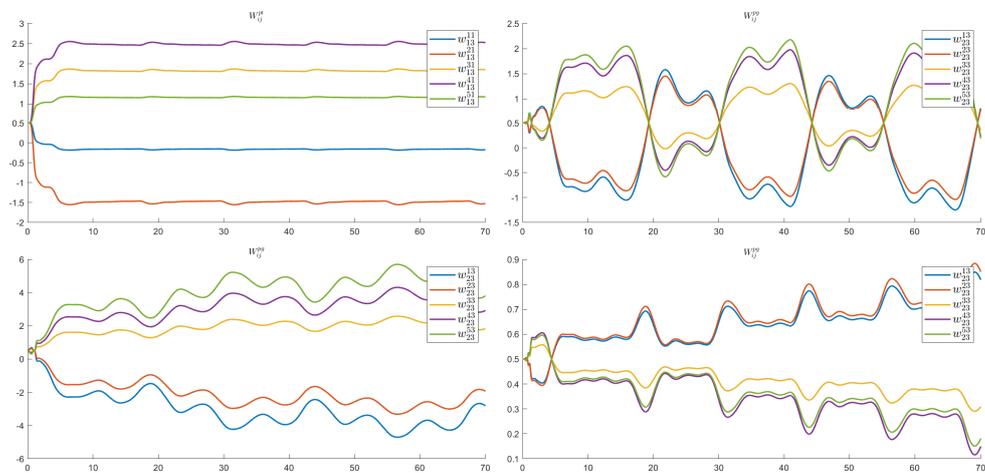


Figura 5.5: Actualización de los pesos de las FRHONNs para el subsistema Z

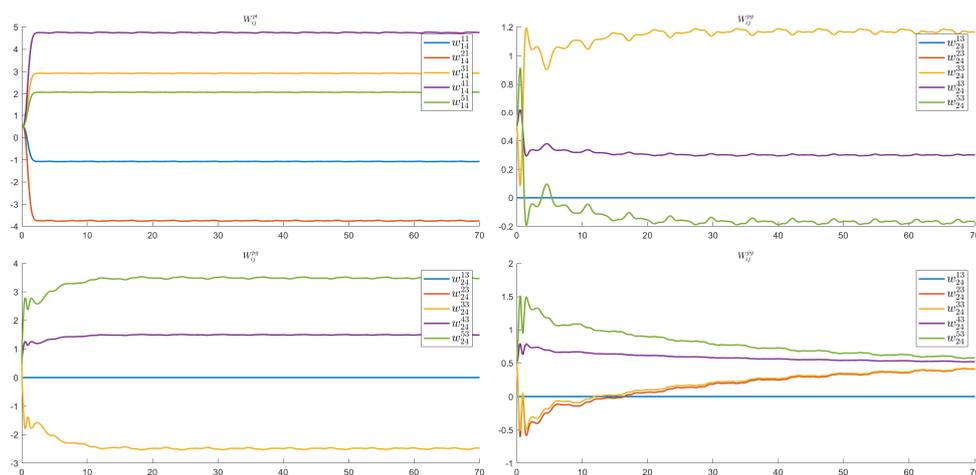


Figura 5.6: Actualización de los pesos de las FRHONNs para el subsistema Yaw

## 5.2. Pruebas Experimentales

Se realizaron pruebas experimentales con el algoritmo de visión propuesto en esta tesis y la ley de control Super Twisting presentada en el Capítulo 3. Las pruebas fueron realizadas en tiempo real a una frecuencia de 20Hz, por lo tanto el tiempo de muestreo para el control era de  $T = 50ms$ . Estas pruebas se llevaron a cabo en la alberca de pruebas del laboratorio LANAVEX de la UMI-LAFMIA.

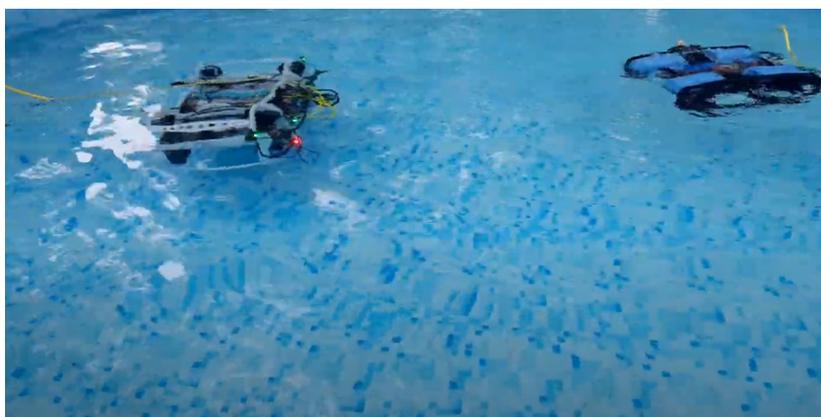


Figura 5.7: El BlueROV2 haciendo seguimiento al marcador de LEDs montado al OctoSub

Se realizaron dos tipos de pruebas para probar el comportamiento del sistema de navegación relativa propuesto:

- La primera prueba fue una prueba de un punto fijo, donde el BlueROV2 debe alcanzar un punto en específico y permanecer en él durante una cantidad de minutos para observar el error en el estado estacionario.

- La segunda prueba consistió en realizar el seguimiento de una trayectoria para probar las capacidades del algoritmo de visión para ser usado como un sistema de referencia inercial, así como para comprobar las propiedades de estabilidad de la ley de control en ambientes reales.

### 5.2.1. Pruebas de punto fijo

Para el primer experimento, se registró la posición del BlueROV2 al ser controlado por la ley de control Super Twisting con tres puntos de referencia diferentes, el primero en  $(x_{ref} = -2.5, y_{ref} = 0)$  que se muestra en la Figura 5.8, el segundo en  $(x_{ref} = -2.5, y_{ref} = -0.5)$  que se muestra en la Figura 5.9 y el tercero en  $(x_{ref} = -1.5, y_{ref} = 0.5)$  que es mostrado en la Figura 5.10.

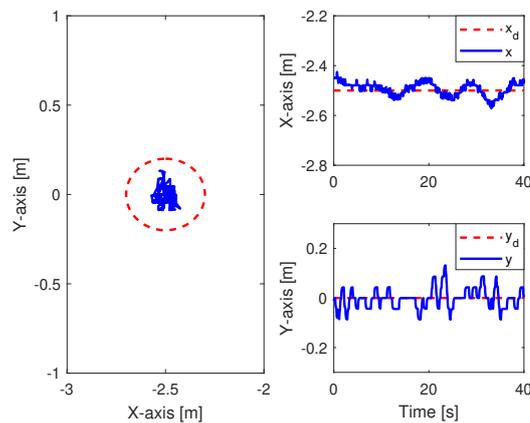


Figura 5.8: La posición del BlueROV2 en el primer experimento usando las coordenadas relativas  $(-2.5, 0)$  como referencia

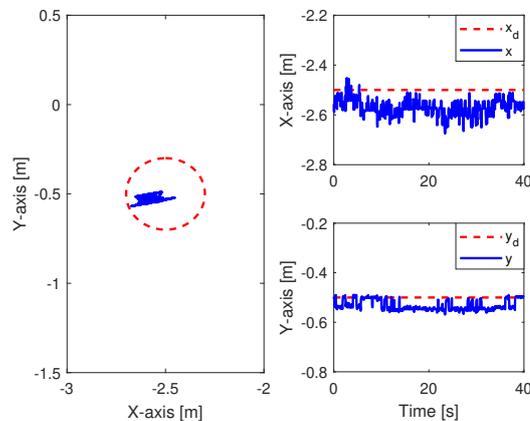


Figura 5.9: La posición del BlueROV2 en el primer experimento usando las coordenadas relativas  $(-2.5, -0.5)$  como referencia

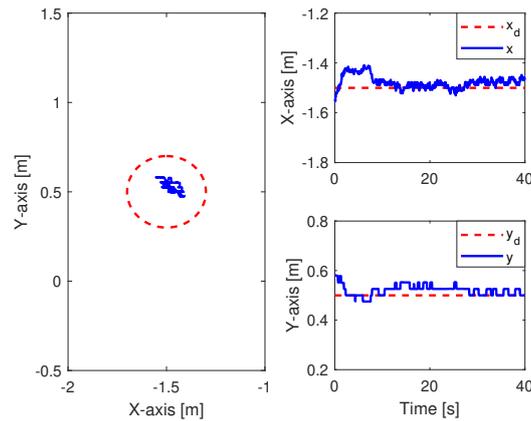


Figura 5.10: La posición del BlueROV2 en el primer experimento usando las coordenadas relativas  $(-1.5, 0.5)$  como referencia

Como pueden observarse en las tres figuras, el error permanece acotado y es relativamente bajo ( $|e_1| < 0.1, |e_2| < 0.1$ ) por lo que se puede afirmar que el desempeño de la ley de control Super Twisting cuando se usa el algoritmo de visión como estimación de la posición y velocidad es satisfactorio.

### 5.2.2. Pruebas de seguimiento de trayectoria

Para verificar el seguimiento de trayectoria usando el algoritmo de visión junto con la ley de control Super Twisting, se utilizó una trayectoria lineal en X y Y pasada por un filtro de primer orden para que el error fuera incrementando gradualmente. La Figura 5.11 muestra la gráfica del seguimiento de trayectoria.

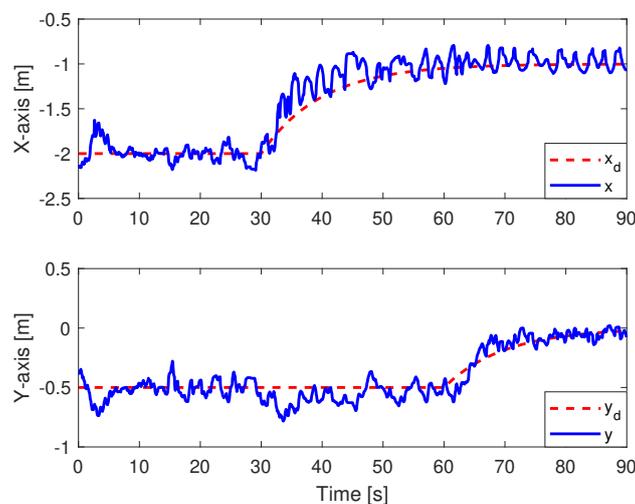


Figura 5.11: El seguimiento de trayectoria del BlueROV2 usando un controlador Super Twisting obteniendo el estado del algoritmo de visión

Con esta prueba se observa la capacidad del BlueROV2 de poder usar el método propuesto de estimación de posiciones y velocidades relativas basado en visión artificial, así mismo, que esta estimación puede usarse adecuadamente para el seguimiento de trayectoria aplicando la estrategia de control robusta por modos deslizantes basada en el algoritmo Super Twisting.

---

---

## Conclusiones y Trabajo a futuro

---

### 6.1. Conclusión

De acuerdo a los resultados de las pruebas experimentales, se comprobó que el método propuesto de navegación relativa basada en visión artificial nos permite adecuadamente realizar el seguimiento de trayectorias mientras se esté observando el marcador de LEDs diseñado. Esto puede ser muy útil para misiones de exploración subacuático donde los vehículos tienen que usar marcadores fijos en lugares de interés para realizar la navegación relativa apoyados de este sistema de navegación, así mismo para inspección de estructuras submarinas que pueden tener marcadores como estos equipados en puntos estratégicos para que los vehículos puedan posicionarse con respecto a la estructura.

Otra potencial aplicación que puede tener este método de localización relativa a un objetivo con el marcador equipado es para operaciones multiagentes, donde puede haber un vehículo líder que tenga este marcador equipado y los vehículos equipados con una cámara monocular pueden seguirlo, utilizando la posición relativa como estimada como un punto de referencia en la trayectoria a seguir, y así mantener una distancia fija entre el líder y los vehículos que lo siguen, esto es posible debido a que en la estimación de la posición y la velocidad relativa no hay ninguna restricción de que el marcador esté estático o esté en movimiento. Faltarían hacer pruebas para probar esta aplicación en específico, que será analizada en la siguiente sección más a profundidad.

El alcance de esta tesis era proponer un método de navegación relativa basada en visión artificial como estimación del vector de estados un vehículo submarino y un controlador inteligente y no linear basado en redes neuronales para tener un controlador inteligente que no

necesite de conocer del modelo dinámico previamente ni que tenga que tener un proceso de ajuste de parámetros exhaustivo. Si bien se desarrolló un controlador basado en redes neuronales recursivas de orden superior (FRHONN) que fue validado por medio de simulaciones, las pruebas experimentales no lograron llevarse a cabo por falta de tiempo. Sin embargo tanto las pruebas de simulación como las pruebas experimentales usando el control Supertwisting indican que una fuerte posibilidad de que el la ley de control de FRHONNs propuesta tendrá un buen desempeño en pruebas experimentales.

Una de las aportaciones finales de este proyecto, es todo el marco de desarrollo e implementación de software de control que se creó durante la realización de este proyecto, que consiste en los nodos de ROS para la integración de scripts de algoritmos de visión y control al stack ROS+Ardupilot+QGroundControl, esto permitirá a futuros investigadores de la UMI-LAFMIA para implementar de una forma más rápida y directa los controladores que diseñen para AUVs que usen el firmware de ArduSub. Así mismo, la creación del ambiente de simulación de Gazebo le permitirá a futuros investigadores poder probar el comportamiento de sus leyes de control en una simulación confiable antes de probarlos en las plataformas experimentales, así aumentando la seguridad y dando una plataforma para encontrar bugs antes de la implementación.

---

## 6.2. Trabajo a futuro

La continuación a este proyecto se puede dar desde varios enfoques:

- La primera, es continuar con los objetivos no alcanzados en esta tesis, que es implementar el algoritmo de control basado en redes neuronales recursivas (FRHONNs) en la plataforma experimental para probar su comportamiento.
  - El segundo, es realizar las modificaciones pertinentes al algoritmo de visión artificial para que también pueda estimar la orientación tanto del marcador como la de la cámara, así como las velocidades angulares.
  - Otra aplicación que podría estudiarse es la de vehículos multiagentes usando el marcador de LEDs montado en un vehículo líder, al que le harán seguimiento otros vehículos.
  - Una aplicación podría ser la incorporación de la localización acústica para hacerse en conjunto con la localización visual y así mezclar los datos de dos sensores para tener mejor precisión y robustez en la localización.
  - Una aplicación frecuente en la literatura reciente es la de usar sistemas de localización visual para asistir a un AUV en estacionarse en una estación de carga subacuática. El método propuesto en esta tesis puede explorarse con esa aplicación en ambientes de iluminación variable.
-

# Bibliografía

---

- [1] J. J. Leonard y A. Bahr, “Autonomous underwater vehicle navigation,” *Springer handbook of ocean engineering*, págs. 341-358, 2016.
- [2] A. Branca, E. Stella y A. Distante, “Autonomous navigation of underwater vehicles,” en *IEEE Oceanic Engineering Society. OCEANS’98. Conference Proceedings (Cat. No. 98CH36259)*, IEEE, vol. 1, 1998, págs. 61-65.
- [3] P. A. Miller, J. A. Farrell, Y. Zhao y V. Djapic, “Autonomous underwater vehicle navigation,” *IEEE Journal of Oceanic Engineering*, vol. 35, n.º 3, págs. 663-678, 2010.
- [4] L. Doitsidis, K. P. Valavanis, N. C. Tsourveloudis y M. Kontitsis, “A framework for fuzzy logic based UAV navigation and control,” en *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA’04. 2004*, IEEE, vol. 4, 2004, págs. 4041-4046.
- [5] S. Jatsun, O. Emelyanova, A. S. M. Leon y S. Stykanyova, “Control flight of a UAV type tricopter with fuzzy logic controller,” en *2017 Dynamics of Systems, Mechanisms and Machines (Dynamics)*, IEEE, 2017, págs. 1-5.
- [6] A. Sharma y A. Barve, “Controlling of quad-rotor uav using pid controller and fuzzy logic controller,” *International Journal of electrical, electronics and computer engineering*, vol. 1, n.º 2, págs. 38-41, 2012.
- [7] K. Ishaque, S. S. Abdullah, S. Ayob y Z. Salam, “A simplified approach to design fuzzy logic controller for an underwater vehicle,” *Ocean Engineering*, vol. 38, n.º 1, págs. 271-284, 2011.
- [8] S. Smith, G. Rae, D. Anderson y A. Shein, “Fuzzy logic control of an autonomous underwater vehicle,” *Control Engineering Practice*, vol. 2, n.º 2, págs. 321-331, 1994.

- 
- [9] Y. Dai, S. Yu, Y. Yan y X. Yu, “An EKF-based fast tube MPC scheme for moving target tracking of a redundant underwater vehicle-manipulator system,” *IEEE/ASME Transactions on Mechatronics*, vol. 24, n.º 6, págs. 2803-2814, 2019.
- [10] A. Alcocer, P. Oliveira y A. Pascoal, “Study and implementation of an EKF GIB-based underwater positioning system,” *Control engineering practice*, vol. 15, n.º 6, págs. 689-701, 2007.
- [11] H. Wang, K. Liu y S. Li, “Command filter based globally stable adaptive neural control for cooperative path following of multiple underactuated autonomous underwater vehicles with partial knowledge of the reference speed,” *Neurocomputing*, vol. 275, págs. 1478-1489, 2018.
- [12] T. Salloom, X. Yu, W. He y O. Kaynak, “Adaptive neural network control of underwater robotic manipulators tuned by a genetic algorithm,” *Journal of Intelligent & Robotic Systems*, vol. 97, n.º 3, págs. 657-672, 2020.
- [13] J. D. Rios, A. Y. Alanis, C. Lopez-Franco y N. Arana-Daniel, “RHONN identifier-control scheme for nonlinear discrete-time systems with unknown time-delays,” *Journal of the Franklin Institute*, vol. 355, n.º 1, págs. 218-249, 2018.
- [14] F. Jurado y S. Lopez, “Continuous-time decentralized neural control of a quadrotor UAV,” en *Artificial Neural Networks for Engineering Applications*, Elsevier, 2019, págs. 39-53.
- [15] M.-j. Zhang y Z.-z. Chu, “Adaptive sliding mode control based on local recurrent neural networks for underwater robot,” *Ocean Engineering*, vol. 45, págs. 56-62, 2012.
- [16] Z. Chen, R. Wang, W. Ji, M. Zong, T. Fan y H. Wang, “A novel monocular calibration method for underwater vision measurement,” *Multimedia Tools and Applications*, vol. 78, n.º 14, págs. 19 437-19 455, 2019.
- [17] A. Handa, M. Bloesch, V. Pătrăucean, S. Stent, J. McCormac y A. Davison, “gvnn: Neural network library for geometric computer vision,” en *European Conference on Computer Vision*, Springer, 2016, págs. 67-82.
- [18] Y. Chen, S. Liu, X. Shen y J. Jia, “Dsgn: Deep stereo geometry network for 3d object detection,” en *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, págs. 12 536-12 545.
-

- [19] D. Lee, G. Kim, D. Kim, H. Myung y H.-T. Choi, "Vision-based object detection and tracking for autonomous navigation of underwater robots," *Ocean Engineering*, vol. 48, págs. 59-68, 2012.
  - [20] X. Wang, J. Zou y D. Shi, "An improved ORB image feature matching algorithm based on SURF," en *2018 3rd International Conference on Robotics and Automation Engineering (ICRAE)*, IEEE, 2018, págs. 218-222.
  - [21] E. Rublee, V. Rabaud, K. Konolige y G. Bradski, "ORB: An efficient alternative to SIFT or SURF," en *2011 International conference on computer vision*, Ieee, 2011, págs. 2564-2571.
  - [22] R. Mur-Artal, J. M. M. Montiel y J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE transactions on robotics*, vol. 31, n.º 5, págs. 1147-1163, 2015.
  - [23] G. Klein y D. Murray, "Parallel tracking and mapping for small AR workspaces," en *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, IEEE, 2007, págs. 225-234.
  - [24] Q. Zhang, G. Xu y N. Li, "Improved SLAM closed-loop detection algorithm based on DBoW2," en *Journal of Physics: Conference Series*, IOP Publishing, vol. 1345, 2019, pág. 042 094.
  - [25] G. L. Foresti y S. Gentili, "A vision based system for object detection in underwater images," *International journal of pattern recognition and artificial intelligence*, vol. 14, n.º 02, págs. 167-188, 2000.
  - [26] M. Leonardi, A. Stahl, M. Gazzea, M. Ludvigsen, I. Rist-Christensen y S. M. Nornes, "Vision based obstacle avoidance and motion tracking for autonomous behaviors in underwater vehicles," en *OCEANS 2017-Aberdeen*, IEEE, 2017, págs. 1-10.
  - [27] R. Pérez-Alcocer, L. A. Torres-Méndez, E. Olgun-Díaz y A. A. Maldonado-Ramírez, "Vision-based autonomous underwater vehicle navigation in poor visibility conditions using a model-free robust control," *Journal of Sensors*, vol. 2016, 2016.
  - [28] N. Gracias, J. Bosch y M. E. Karim, "Pose estimation for underwater vehicles using light beacons," *IFAC-PapersOnLine*, vol. 48, n.º 2, págs. 70-75, 2015.
-

- 
- [29] J. González-García, A. Gómez-Espinosa, E. Cuan-Urquiza, L. G. García-Valdovinos, T. Salgado-Jiménez y J. A. Escobedo Cabello, “Autonomous underwater vehicles: Localization, navigation, and communication for collaborative missions,” *Applied sciences*, vol. 10, n.º 4, pág. 1256, 2020.
- [30] X. Su, I. Ullah, X. Liu y D. Choi, “A review of underwater localization techniques, algorithms, and challenges,” *Journal of Sensors*, vol. 2020, 2020.
- [31] K. N. Lwin, N. Mukada, M. Myint et al., “Docking at pool and sea by using active marker in turbid and day/night environment,” *Artificial Life and Robotics*, vol. 23, n.º 3, págs. 409-419, 2018.
- [32] R. T. Amjad, M. Mane, A. A. Amjad, W. Ge, Z. Zhang y J. Xu, “Tracking of light beacons in highly turbid water and application to underwater docking,” en *Ocean Sensing and Monitoring XIV*, SPIE, vol. 12118, 2022, págs. 90-97.
- [33] A. B. d. A. A. Figueiredo, “Monocular vision-based system for docking a hovering AUV,” 2020.
- [34] R.-J. Huang, Y.-C. Lai, C.-Y. Tsao, Y.-P. Kuo, J.-H. Wang y C.-C. Chang, “Applying convolutional networks to underwater tracking without training,” en *2018 IEEE International Conference on Applied System Invention (ICASI)*, IEEE, 2018, págs. 342-345.
- [35] Y. Hamamatsu, T. Matsuda, T. Sakamaki y T. Maki, “Short Range Tracking Method of Underwater Vehicles Based on Vision,” en *OCEANS 2019-Marseille*, IEEE, 2019, págs. 1-5.
- [36] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.
- [37] T. I. Fossen, *Nonlinear modelling and control of underwater vehicles*. Universitetet i Trondheim (Norway), 1991.
- [38] T. SNAME, “Nomenclature for treating the motion of a submerged body through a fluid,” *The Society of Naval Architects and Marine Engineers, Technical and Research Bulletin*, n.º 1950, págs. 1-5, 1950.
-

- [39] J. A. Moreno, "A linear framework for the robust stability analysis of a generalized super-twisting algorithm," en *2009 6th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, IEEE, 2009, págs. 1-6.
  - [40] J. A. Moreno y M. Osorio, "Strict Lyapunov functions for the super-twisting algorithm," *IEEE transactions on automatic control*, vol. 57, n.º 4, págs. 1035-1040, 2012.
  - [41] A. Levant, "Sliding order and sliding accuracy in sliding mode control," *International journal of control*, vol. 58, n.º 6, págs. 1247-1263, 1993.
  - [42] A. Levant, "Higher-order sliding modes, differentiation and output-feedback control," *International journal of Control*, vol. 76, n.º 9-10, págs. 924-941, 2003.
  - [43] M. Á. Garcia Rangel, "Inspección y reconstrucción 3D de estructuras marinas usando un ROV," 2021.
-