CEA Comité Español de Automática

XLVI JORNADAS AUTOMÁTICA CARTAGENA 2025

UNIVERSIDADE DA CORUÑA

Jornadas de Automática

Robótica Colaborativa Marina: Explorando Escenarios con NauSim y YOLO

Ortiz-Toro, César Antonio[a,*], Gutiérrez, Alvaro[a], Chaos-García, Dictino[b], Cerrada-Collado, Cristina[b], Luque-Nieto, Miguel Ángel[c], Clemente-Medina, M. Carmen[c]

[a]*Escuela Técnica Superior de Ingenieros de Telecomunicación, Universidad Politécnica de Madrid, Avenida Complutense, 30, 28040 Madrid, España*
[b]*Departamento de Ciencias de la Computación y Control Automático UNED , Madrid, Madrid, España.*
[c]*Instituto de Ingeniería Oceánica, Escuela Técnica Superior de Ingeniería de Telecomunicación, Universidad de Málaga, Cervantes, 2. 29071 Málaga España.*

To cite this article: Ortiz-Toro, C.A., Gutiérrez, A., Chaos-García, D., Luque-Nieto, M.A., Cerrada-Collado, C., Clemente-Medina, M.C. 2025. Collaborative Marine Robotics: Exploring Scenarios with NauSim and YOLO. Jornadas de Automática, 46. https://doi.org/10.17979/ja-cea.2025.46.12260

Resumen

Este artículo presenta los resultados de las pruebas con NauSim, un simulador de código abierto para drones acuáticos, junto con el sistema de visión por computador YOLO en el contexto de la robótica colaborativa para el control y posicionamiento de vehículos de superficie no tripulados (USV). Se desarrollan algoritmos de control para tareas como formaciones, seguimiento de rutas y evasión de obstáculos, que se prueban en simulación y en escenarios reales. Los resultados muestran un notable rendimiento general del sistema y buena correspondencia entre la simulación y la realidad, apoyando la viabilidad del sistema propuesto.

Palabras clave: Simulación, Vehículos Marítimos Autónomos, Sistemas multivehículo, Sensores/actuadores, Navegación, Programación y Visión

Collaborative Marine Robotics: Exploring Scenarios with NauSim and YOLO

Abstract

This work presents the results of tests with NauSim, an open source simulator for waterborne drones, together with the YOLO computer vision system in the context of collaborative robotics for the control and positioning of unmanned surface vehicles (USVs). Control algorithms for tasks such as formations, route following and obstacle avoidance are developed and tested in simulation and real scenarios. The results show remarkable overall system performance and good correspondence between simulation and reality, supporting the feasibility of the proposed system.

Keywords: Simulation, Unmanned marine vehicles, Multi-vehicle systems, Sensors and actuators, Robot Navigation, Programming and Vision

1. Introducción

La navegación autónoma de vehículos de superficie no tripulados (USV) ha emergido como un campo de investigación de gran importancia en diversas aplicaciones, desde la monitorización ambiental y la inspección de infraestructuras hasta la seguridad y defensa como se muestra en Barrera et al. (2021). La capacidad de los USV para operar de manera autónoma, individualmente o en formaciones, ofrece ventajas significativas en términos de eficiencia, seguridad y cobertura.

Sin embargo, el desarrollo de sistemas de control robustos para la navegación en entornos marinos presenta desafíos debido a la dinámica no lineal de los vehículos, las perturbaciones ambientales (corrientes, viento, olas) y la necesidad de percibir y reaccionar ante un entorno dinámico. Esto hace el uso de simuladores esencial en el desarrollo de sistemas de control, especialmente en las fases iniciales del diseño y las pruebas iniciales de vehículos marinos, ya que reducen la necesidad de acceso constante a un entorno acuático real y un control sobre el entorno que, de otra manera, no sería posi-

*Autor para correspondencia: ca.ortiz@upm.es

ble. Se han desarrollado numerosas herramientas para este fin, como UWSim en Prats et al. (2012) y la extensión para Gazebo "UUV Simulator" en Manhães et al. (2016). También existen paquetes que integran funcionalidades de simulación en plataformas conocidas como MATLAB™/Simulink™(ver de Cerqueira Gava et al. (2022)). La tendencia actual se inclina por simuladores que ofrecen gran fidelidad visual utilizando motores 3D comerciales, como HoloOcean presentado en Potokar et al. (2022) y UNav-Sim (Amer et al., 2023). Sin embargo, esto puede generar dependencia de la estructura y las limitaciones de dichos motores.

Por otro lado, si bien existen múltiples formas de obtener la posición de un USV como puede verse en Qiao et al. (2023) y Peng et al. (2020), la posibilidad de usar posicionamiento mediante análisis de imagen externa en tiempo real no ha sido convenientemente explotada. Ciertamente el concepto presenta múltiples problemas, como un rango limitado y el ser vulnerable a los cambios de iluminación del entorno o a condiciones atmosféricas inestables. Sin embargo, su bajo coste tanto de desarrollo, al tener disponible una cámara, como de despliegue lo convierte en una opción muy conveniente en las primeras fases de un proyecto o cuando la simplicidad del robot o la cantidad de vehículos en el sistema desaconsejan métodos más complejos.

Este trabajo se centra en el diseño, simulación y validación experimental de algoritmos de control para USV, utilizando YOLO (You Only Look Once) (específicamente YOLOv8, ver Varghese and Sambath (2024)), como sistema de posicionamiento abordando tres capacidades fundamentales: el mantenimiento de formaciones (estáticas y dinámicas), el seguimiento de rutas con evasión de obstáculos y el comportamiento de bandada. La metodología propuesta integra el uso de un simulador de desarrollo propio, NAUSIM (Ortiz-Toro et al., 2024), para la fase de diseño y testeo inicial de los controladores, seguida de pruebas con los vehículos reales para validar el rendimiento de los algoritmos en condiciones operativas, tanto en un entorno controlado, como en instalaciones portuarias.

2. Materiales y métodos

Este trabajo se encuadra dentro del proyecto "Heterogeneous submarine networks for collaborative, dynamic and profitable exploration" (NEMO4EX). Este proyecto busca crear enjambres de vehículos autónomos pequeños y económicos para gestionar tareas coordinadas sin intervención humana directa. Estos enjambres actuarán como un sistema descentralizado que decidirá autónomamente el despliegue de cada vehículo y adaptará su cobertura según las necesidades del entorno y la información colectiva, ofreciendo servicios como posicionamiento, recogida de datos y recarga de baterías para otros nodos.

Se plantea un enfoque de desarrollo híbrido, donde la fase inicial de diseño y validación se realiza en un entorno simulado, y posteriormente se transfiere a una plataforma real para pruebas exhaustivas. Como entorno simulado, tanto para el diseño y la validación como para el control de las pruebas en entornos reales se utiliza el simulador NauSIM.

El sistema descrito emplea tres vehículos BlueROV2 (Robotics, 2016) en su configuración pesada (Figura 1), adaptados para operar como vehículo de superficie no tripulado(USV). Cada BlueROV2, en su configuración pesada, tiene cuatro propulsores horizontales y cuatro verticales, permitiendo seis grados de libertad. Están controlados por una Raspberry Pi con una placa de expansión que integra una unidad de medición inercial, un magnetómetro y un sensor de presión. Además, cada vehículo está equipado con un aparato de ecosondeo y un sonar mecánico de barrido para navegación.

Los tres robots están conectados a un único ordenador central a través de enlaces USB configurados como interfaces de red IP, con cada robot tratado como una entidad autónoma con acceso exclusivo a sus propios sensores. El movimiento de cada robot se gestiona mediante dos controladores PID independientes (uno para el avance lineal y otro para el giro angular), cuyos parámetros se afinaron inicialmente en simulación y luego experimentalmente. Estos controladores generan valores de activación que son convertidos en comandos MAVLink (Koubâa et al., 2019) por la unidad central y enviados a cada vehículo, cerrando el ciclo de control. La coordinación entre los robots se facilita mediante el intercambio de mensajes dentro del simulador, lo que es el único método para compartir información entre ellos. Este diseño enfatiza la modularidad, escalabilidad y la capacidad de simular interacciones entre vehículos completamente autónomos.

Figura 1: Imagen del BlueROV2 en su configuración pesada.

Se realizan pruebas en dos escenarios reales: en el tanque de pruebas del Laboratorio de Puertos, ubicado en las instalaciones de la Escuela Técnica Superior de Ingenieros Civiles de la UPM y en uno de los muelles situados en la base naval de Puntales, en Cádiz (ver Figura 2). En estos escenarios, la posición estimada de cada robot se obtiene a partir de una cámara externa capturando vídeo en tiempo real, conectada directamente al ordenador central. Puesto que en un entorno acuático real es muy difícil posicionarse en una vista cenital, en cada uno de los escenarios se realiza una calibración de perspectiva sobre la cámara, haciendo la corrección correspondiente en cada uno de los "frames"del vídeo, para evitar la que la distorsión de perspectiva afecte al posicionamiento. La detec-

ción y posicionamiento de cada uno de los vehículos se realiza mediante un modelo generado partiendo de YOLOv8. YOLO es una arquitectura de red neuronal convolucional (CNN) de etapa única diseñada para la detección de objetos en tiempo real. YOLO procesa la imagen completa en una sola pasada, prediciendo directamente las cajas delimitadoras y las probabilidades de clase. Esta arquitectura permite un procesamiento notablemente rápido, ideal para aplicaciones en tiempo real como el guiado de vehículos autónomos.

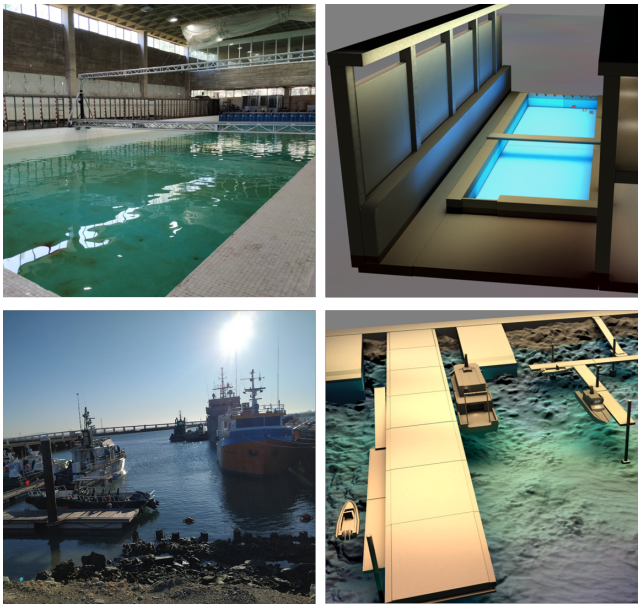

Figura 2: Escenarios de prueba junto con sus homólogos virtuales; tanque de pruebas del Laboratorio de Puertos en la fila superior y base naval de Puntales en la inferior.

La posición estimada de cada robot se asocia a su identificador y es suministrada exclusivamente al proceso de control correspondiente, configurándose, desde el punto de vista de los vehículos, como un sensor más.

2.1. NAUSIM

El núcleo de la arquitectura de NauSim se basa en el modelo sensor-controlador-actuador. El modelo de arquitectura sensor-controlador-actuador es un marco fundamental en el diseño y operación de drones, asegurando una ejecución eficiente y precisa de las tareas. Este modelo divide la funcionalidad del dron en tres capas interconectadas: sensores, controladores y actuadores. Este enfoque estructurado permite una retroalimentación continua y ajustes en tiempo real, configurando el dron sea como una herramienta versátil y adaptable. No solo eso, la arquitectura aísla los diferentes componentes del dron, permiten tanto la reusabilidad de partes ya implementadas como una transición sencilla entre el robot simulado y el real. La interacción de esta arquitectura con el mundo simulado se hace a través de un modelo físico, que traduce las acciones de los actuadores al estado en el que se encontrará el robot dentro del espacio simulado.

- Sensores: Incluyen tanto sensores reales (Posicionamiento, cámaras, LIDAR, etc.) como virtuales (p ej. , detectores de colisión). Cada sensor se ejecuta en un hilo independiente con una frecuencia de actualización configurable, simulando de forma realista su comportamiento en el mundo real. No se distingue entre sensores simulados y reales, lo que permite una integración directa con hardware real tras el entrenamiento de modelos de control.

- Controladores: Procesan la información proveniente de los sensores para generar ordenes para los actuadores. Pueden variar desde simples scripts hasta algoritmos complejos de aprendizaje. El sistema permite jerarquías de controladores, facilitando transiciones entre modos (por ejemplo, manual y automático).

- Actuadores: Los actuadores (motores, servos, etc.) ejecutan las órdenes del controlador. En simulación, se modelan mediante interfaces físicas que traducen ordenes en movimientos en el entorno virtual. En robots reales, pueden estar conectados directamente al hardware o gestionados mediante middleware como MAVLink o ROS2.

Independientemente de las funcionalidades que incluya, un simulador para robótica se suele organizar alrededor de un espacio virtual, una representación del mundo real donde el proceso de simulación tiene lugar. En este simulador, la visualización del entorno virtual se realiza mediante el motor Panda3D (Goslin and Mine, 2004). Actualmente bajo licencia BSD y mantenido por la activa comunidad de usuarios, Panda3D (originalmente ”Platform Agnostic Networked Display Architecture”, aunque el acrónimo en sí ha caído en desuso) fue desarrollado inicialmente por Disney Interactive en 2002 para su división de realidad virtual en parques temáticos. Panda3D ofrece un completo conjunto de funcionalidades, es completamente multiplataforma, y presenta una ”interface” completamente desarrollada en Python.

2.1.1. Modelo físico

Como modelo físico para la simulación de este vehículo se han adaptado e implementado dos versiones, una simplificada, una completa, de los modelos matemáticos de simulación de dinámicas para BlueROV2. Se utilizará la notación compacta de Fossen Fossen (2011). Definimos la velocidad del vehículo en relación con el marco de su cuerpo como $\boldsymbol{\nu} = [u, v, w, p, q, r]^T, \boldsymbol{\nu} \in \mathbb{R}^6$, donde las seis variables de velocidad corresponden a la oscilación, el balanceo, el cabeceo y la guiñada, y $\boldsymbol{\eta} = [x, y, z, \phi, \theta, \psi]^T, \boldsymbol{\eta} \in \mathbb{R}^6$, como el vector que combina las posiciones y los ángulos de Euler con respecto al marco del mundo. Si se supone que no hay fuerzas externas tales como corrientes o la influencia de la correa de sujeción, su modelo cinemático se puede expresar como Ecuación 1.

$$(\mathbf{M}_{RB} + \mathbf{M}_A)\,\dot{\boldsymbol{\nu}} + (\mathbf{C}_{RB}(\boldsymbol{\nu}) + \mathbf{C}_A(\boldsymbol{\nu}))\,\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \quad (1)$$

Donde:

- $\mathbf{M}_{RB} \in \mathbb{R}^{6\times6}$: matriz de masa del cuerpo rígido.

- $\mathbf{M}_A \in \mathbb{R}^{6\times6}$: matriz de masa añadida. La matriz de masa añadida cuantifica la inercia añadida debida al fluido circundante.

- $\mathbf{C}_{RB}(\boldsymbol{\nu})$: matriz de Coriolis y centrífuga del cuerpo rígido. Esta matriz describe las fuerzas que actúan sobre una partícula debido a la rotación del sistema de referencia.

- $\mathbf{C}_A(\boldsymbol{\nu})$: matriz de Coriolis para la masa añadida. Cuantifica la inercia en un en sistema en rotación resultado tanto de la rotación del sistema de referencia como de la inercia adicional debida al fluido desplazado.

- $\mathbf{D}(\boldsymbol{\nu})$: matriz de amortiguamiento hidrodinámico. Las fuerzas de amortiguación son una fuerza de arrastre resultado de la viscosidad del fluido.

- $\mathbf{g}(\boldsymbol{\eta})$: vector de fuerzas y momentos hidrostáticos (gravedad y flotabilidad).

- $\boldsymbol{\tau} \in \mathbb{R}^6$: vector de esfuerzos generados por los actuadores.

El BlueROV2 en su configuración pesada está equipado con ocho propulsores T200. La relación entre los comandos de entrada a cada propulsor y las fuerzas/momentos generados se modela según la expresión 2.

$$\boldsymbol{\tau} = \mathbf{B} \cdot \mathbf{G}(s) \cdot \boldsymbol{\mu} \tag{2}$$

Donde $\boldsymbol{\mu} \in \mathbb{R}^8$ es un vector de señales de control a los propulsores (normalizado en $[-1, 1]$), $\mathbf{B} \in \mathbb{R}^{6\times8}$ corresponde a la matriz de asignación de empuje (posición y orientación de los propulsores) y $\mathbf{G}(s) \in \mathbb{R}^{8\times8}$ es la matriz de funciones de transferencia de los propulsores.

Finalmente, además de la dinámica, se considera la relación cinemática entre los marcos de referencia del vehículo y del mundo mostrada en la Ecuación 3:

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta}) \cdot \boldsymbol{\nu} \tag{3}$$

donde $\mathbf{J}(\boldsymbol{\eta})$ es la matriz Jacobiana que transforma las velocidades del *body frame* al *world frame*, dependiendo de la orientación del vehículo.

Este modelo ha sido implementado dentro del entorno de simulación NauSim utilizando los parámetros hidrodinámicos medidos experimentalmente para el BlueROV2, según los datos presentados en von Benzon et al. (2022) a partir de los trabajos desarrollados en Wu (2018).

2.2. Casos de uso

Los módulos de control se plantean como máquinas de estado simples, donde se actualiza una posición objetivo dependiendo de las variables relevantes en cada momento y se deja en manos del controlador PID alcanzar dicha posición. Se han desarrollado e implementado los siguientes algoritmos de control :

- Formaciones Estáticas: Los USV se posicionan formando patrones predefinidos,(e.g., línea, triángulo, cuadrado. Los vehículos comunican al resto del enjambre si han alcanzado su posición asignada y manteniendo la posición, en caso de ser necesario, hasta que el grupo completa la formación especificada.

- Formaciones Dinámicas: La formación se mueve como una unidad, manteniendo la estructura relativa entre los vehículos mientras siguen una trayectoria global. Esto implica que cada USV ajusta su posición y velocidad para seguir a un líder virtual o a un punto de referencia en movimiento, manteniendo la distancia y orientación deseadas con respecto a sus vecinos.

- Formaciones de Bandada (Flocking): El algoritmo de bandada permite que los vehículos de superficie no tripulados (USV) se muevan juntos de forma coordinada siguiendo a un líder o sin control centralizado, inspirándose en la naturaleza. Cada USV sigue reglas locales como las esbozadas en Reynolds (1987) para interactuar con sus vecinos: evitan colisiones, igualan su dirección y velocidad con los USV cercanos, y se mantienen cerca del centro del grupo. Todos los vehículos comparten su posición con la bandada.

- Seguimiento de Rutas y Evasión: Los USV siguen una ruta predefinida mientras evitan colisiones con otros vehículos. La estrategia de evasión se integra con el controlador de seguimiento de ruta, utilizando la información de posicionamiento de YOLO para detectar y predecir las trayectorias de los obstáculos y generar trayectorias evasivas. Los vehículos comunican su posición al resto del enjambre.

3. Resultados

A partir de una serie de 1330 conteniendo los USV objetivo (BlueROV2 rojo ,BlueROV2 amarillo y BlueROV2 azul) en diversas condiciones de iluminación, ángulos y fondos acuáticos, se ha construido un conjunto datos etiquetado con las apariciones de cada vehículo, divididos tal y como se muestra en la Tabla 1. En la Figura 3 pueden verse algunos ejemplos de imágenes etiquetadas parte del conjunto.

Utilizando este conjunto se ha generado un modelo YOLO para realizar el posicionamiento en la escena mediante una cámara externa. La validación realizada con el conjunto de prueba muestra una notable capacidad para diferenciar entre las categorías de objetos definidas, con una precisión 0,97, y un "recall" de 0,96. La matriz de confusión (Tabla 2) ofrece una visión detallada de la calidad de la clasificación con clasificaciones erróneas mínimas y un número muy limitado de casos donde los vehículos se confunden con el fondo. En cuanto a la calidad del posicionamiento en sí, el modelo muestra una precisión promedio (mAP) de 0,97 para un IoU (intersección sobre la unión) de 0,5. Aunque el mAP a umbrales de IoU más estrictos (0,95) es más bajo ,0,62 ,sigue siendo un resultado notable y demuestra que el modelo es capaz de una localización precisa.

Tabla 1: Composición y distribución del conjunto de datos usado en la generación del modelo YOLO.

	Entrenamiento	Validación	Test	Total
V. azul	363	102	99	564
V. amarillo	289	97	92	478
V. rojo	279	98	103	480
Total	931	297	294	1522

Figura 3: Conjunto de datos; ejemplos de imágenes etiquetadas.

Tabla 2: Modelo para la detección de los USV en imágenes. Pruebas con el conjunto de test; Matriz de confusión.

		Actual			
		V.azul	V.amarillo	V.rojo	Fondo
Predicción	V. azul	**93**	1	0	4
	V. amarillo	0	**87**	0	2
	V. rojo	1	0	**101**	3
	Fondo	5	4	2	-

Aunque no es estrictamente necesario para estas pruebas, puesto que a cada vehículo le corresponde una clase específica, para obtener una evaluación más realista del rendimiento de YOLO en escenarios complejos, se utiliza BoT-SORT (*Bounding Box Tracker with Simple Online Real-time Tracking*) para realizar el seguimiento de objetos. BoT-SORT es una extensión común para detectores como YOLO, que mantiene identificadores únicos para los objetos a lo largo del vídeo, incluso con oclusiones o cambios de pose. La pérdida de seguimiento en BoT-SORT representa un fallo sostenido en la identificación de un objeto a lo largo del tiempo, indicando situaciones límite en la detección de imágenes que podrían afectar negativamente un programa de control, lo que la convierte en una métrica más útil que por ejemplo simplemente el número de fallos de YOLO.

Comparando estos pérdidas de seguimiento en los entornos de la piscina de la piscina y el puerto, encontramos que existe un aumento sustancial del número de veces que el sistema de posicionamiento perdió la pista de los robots en el escenario del muelle. De esta forma aunque tenemos el tiempo de pruebas registrado es similar en los dos casos (43:26 en 18 pruebas diferentes, en el caso del puerto, 39:17 en la piscina, en 16 pruebas) el sistema perdió es seguimiento de los vehículos casi el doble de veces (121 frente a 62). La marcada diferencia de rendimiento entre los dos entornos nos da una idea de cómo la complejidad y la variabilidad del entorno influye en la precisión del sistema de posicionamiento basado en YOLO.

Centrándonos en el rendimiento de los algoritmos de control, en la Figura 4 se presenta una comparación de las trayectorias seguidas por los vehículos en diferentes situaciones, tanto en simulación como en un entorno real. Dado que siempre va a existir diferencias tanto en las condiciones iniciales de las pruebas como en la posibles perturbaciones del entorno entre escenarios simulados y reales, buscaremos verificar si el comportamiento programado en el simulador se reproduce de manera similar en un escenario físico con condiciones análogas, antes que una validación puramente cuantitativa.

Los vehículos muestran un comportamiento similar en entornos reales al observado en las simulaciones, especialmente cuando se utilizan controladores sencillos para tareas como el mantenimiento de formaciones estáticas y dinámicas. Los experimentos han confirmado que los USV pueden sostener tanto formaciones estáticas como dinámicas, pueden seguir rutas y ser capaces de evadir otros vehículos en movimiento de forma segura en caso de ser necesario, de forma similar a como se comportan en las simulaciones. Este comportamiento también se da en el ámbito del comportamiento de bandada, donde se evidencian los comportamiento emergentes de formaciones coherentes y dinámicas tipos de las reglas de bandada tanto en entornos simulados como reales.

Si bien es un situación esperada y también presente en escenarios simulados, existe una tendencia en los vehículos a ”deslizarse” de sus posiciones en las formaciones estáticas con mayor facilidad en un entorno real. De manera similar, las pruebas en escenarios reales, particularmente las realizadas en el puerto, revelan trayectorias más irregulares para los vehículos. A pesar de esta inestabilidad, el resultado general no se ve comprometido gracias a la acción compensatoria de los controladores PID, que gestionan eficazmente estas perturbaciones. Cabe destacar que, a pesar de su simplicidad, los controladores probados demuestran una notable resiliencia ante fallos en el posicionamiento, lo que subraya su robustez en condiciones operativas diversas.

4. Conclusiones

Los resultados obtenidos en estas pruebas muestran tanto la eficacia de NauSim como una herramienta robusta para el desarrollo, prueba y validación experimental de algoritmos de control para vehículos de superficie no tripulados (USV). Se ha demostrado con éxito el desarrollo de algoritmos de control para el mantenimiento de formaciones (estáticas y dinámicas), el seguimiento de rutas con evasión de obstáculos y el comportamiento de bandada. La integración de YOLO como sistema de posicionamiento en tiempo real se muestra como una solución económica y de rápida implementación para entornos acuáticos, proporcionando resultados precisos y confiables, a pesar de los fallos en el seguimiento observados debido a variaciones en iluminación, fondos complejos y oclusiones.

Agradecimientos

Este trabajo cuenta con el apoyo de la subvención PID2023-146540OB financiada por MCIN/AEI/10.13039/501100011033. Este trabajo ha sido parcialmente financiado por la Comisión Europea en el contexto del proyecto SMAUG, financiado por Horizon Europe Grant Agreement Acuerdo 101121129

Referencias

Amer, A., Álvarez-Tuñón, O., Uğurlu, H. İ., Sejersen, J. L. F., Brodskiy, Y., Kayacan, E., 2023. Unav-sim: A visually realistic underwater robotics si-

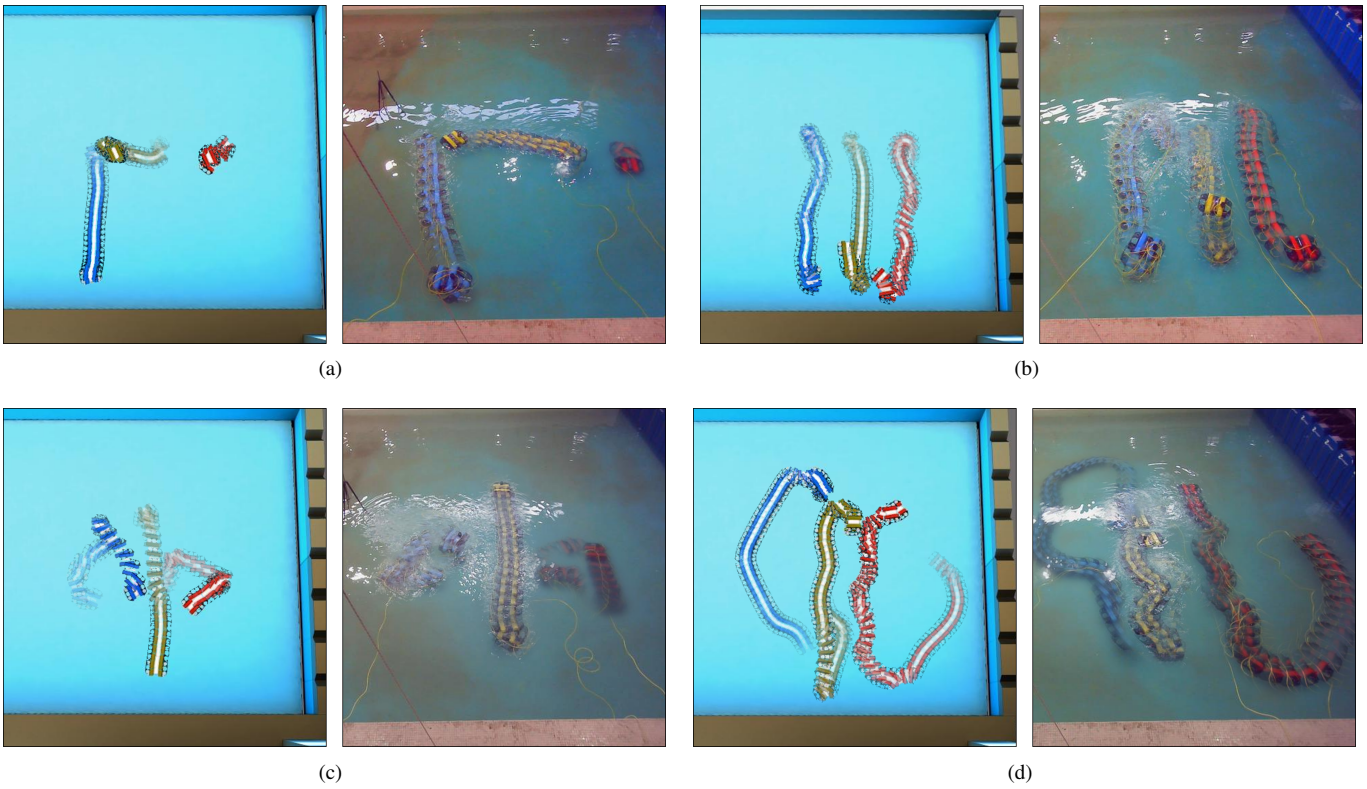

Figura 4: Rastro de movimiento; comparativa del comportamiento de los controladores en varias situaciones en escenarios virtuales y reales utilizando 3 vehículos diferentes. En la Figura 4(a) se muestra el movimiento seguido al cambiar de una formación en línea a una formación en triángulo. En Figura 4(b) se compara el cambio de dirección en una formación dinámica en triangulo. La Figura 4(c) muestra el comportamiento de una bandada al realizar un cambio de dirección. En la Figura 4(d) se pueden ver las similitudes en el movimiento en un escenario virtual y un aprueba real en una situación de evasión de colisiones entre vehículos.

mulator and synthetic data-generation framework. In: 2023 21st International Conference on Advanced Robotics (ICAR). IEEE, pp. 570–576.
DOI: 10.48550/arXiv.2310.11927

Barrera, C., Padron, I., Luis, F., Llinas, O., 2021. Trends and challenges in unmanned surface vehicles (usv): From survey to shipping. TransNav: International Journal on Marine Navigation and Safety of Sea Transportation 15.
DOI: 10.12716/1001.15.01.13

de Cerqueira Gava, P. D., Nascimento Júnior, C. L., Belchior de França Silva, J. R., Adabo, G. J., 2022. Simu2vita: A general purpose underwater vehicle simulator. Sensors 22 (9), 3255.
DOI: 10.3390/s22093255

Fossen, T. I., 2011. Handbook of marine craft hydrodynamics and motion control. John Wiley & Sons.
DOI: 10.1002/9781119994138

Goslin, M., Mine, M. R., 2004. The panda3d graphics engine. Computer 37 (10), 112–114.
DOI: 10.1109/MC.2004.180

Koubâa, A., Allouch, A., Alajlan, M., Javed, Y., Belghith, A., Khalgui, M., 2019. Micro air vehicle link (mavlink) in a nutshell: A survey. IEEE Access 7, 87658–87680.
DOI: 10.1109/ACCESS.2019.2924410

Manhães, M. M. M., Scherer, S. A., Voss, M., Douat, L. R., Rauschenbach, T., 2016. Uuv simulator: A gazebo-based package for underwater intervention and multi-robot simulation. In: OCEANS 2016 MTS/IEEE Monterey. Ieee, pp. 1–8.
DOI: 10.1109/OCEANS.2016.7761080

Ortiz-Toro, C. A., Cerrada-Collado, C., Moreno-Salinas, D., Chaos-Garcia, D., Garcia-Suárez, K. L., Otero-Roth, P., Vidal-Perez, J. M., Luque-Nieto, M. A., Vázquez, A. I., Fraile-Ardanuy, J. J., Negro-Valdecantos, V., Jimenez-Yguacel, E., Aranda-Almansa, J., Zazo-Bello, S., Zufiria, P. J., Magdalena-Layos, L., Parras-Moral, J., Gutierrez, A., 2024. Exploring uuv development with nausim: An open-source simulation platform. In: 2024 Global Conference on Wireless and Optical Technologies (GCWOT). pp. 1–7.
DOI: 10.1109/GCWOT63882.2024.10805602

Peng, Z., Wang, J., Wang, D., Han, Q.-L., 2020. An overview of recent advances in coordinated control of multiple autonomous surface vehicles. IEEE Transactions on Industrial Informatics 17 (2), 732–745.
DOI: 10.1109/TII.2020.3004343

Potokar, E., Ashford, S., Kaess, M., Mangelson, J. G., 2022. Holoocean: An underwater robotics simulator. In: 2022 International Conference on Robotics and Automation (ICRA). IEEE, pp. 3040–3046.
DOI: 10.1109/ICRA46639.2022.9812353

Prats, M., Perez, J., Fernández, J. J., Sanz, P. J., 2012. An open source tool for simulation and supervision of underwater intervention missions. In: 2012 IEEE/RSJ international conference on Intelligent Robots and Systems. IEEE, pp. 2577–2582.
DOI: 10.1109/IROS.2012.6385788

Qiao, Y., Yin, J., Wang, W., Duarte, F., Yang, J., Ratti, C., 2023. Survey of deep learning for autonomous surface vehicles in marine environments. IEEE Transactions on Intelligent Transportation Systems 24 (4), 3678–3701.
DOI: 10.1109/TITS.2023.3235911

Reynolds, C. W., 1987. Flocks, herds and schools: A distributed behavioral model. In: Proceedings of the 14th annual conference on Computer graphics and interactive techniques. pp. 25–34.
DOI: 10.1145/37402.37406

Robotics, B., 2016. Bluerov2: The world's most affordable high-performance rov. BlueROV2 Datasheet; Blue Robotics: Torrance, CA, USA.

Varghese, R., Sambath, M., 2024. Yolov8: A novel object detection algorithm with enhanced performance and robustness. In: 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS). IEEE, pp. 1–6.
DOI: 10.1109/ADICS58448.2024.10533619

von Benzon, M., Sørensen, F. F., Uth, E., Jouffroy, J., Liniger, J., Pedersen, S., 2022. An open-source benchmark simulator: Control of a bluerov2 underwater robot. Journal of Marine Science and Engineering 10 (12), 1898.
DOI: 10.3390/jmse10121898

Wu, C.-J., 2018. 6-dof modelling and control of a remotely operated vehicle. Ph.D. thesis.