

Jornadas de Automática

Percepción, aproximación y agarre robótico autónomo subacuático de tuberías usando visión monocular

Pérez Edo, Inés^{a,*}, López Barajas, Salvador^a, Marín Prades, Raúl^a, Solís Jiménez, Alejandro^a, Pino Jarque, Andrea^a, Sanz Valero, Pedro José^a

^aCentro de Investigación en Robótica y Tecnología Subacuática (CIRTESU), Calle Universitat Jaume I, 134, 12006 Castelló de la Plana, España.

To cite this article: Pérez Edo, Inés, López Barajas, Salvador, Marín Prades, Raúl, Solís Jiménez, Alejandro, Sanz Valero, Pedro José, Pino Jarque, Andrea. 2025. Perception, approach, and grasping of underwater pipes by an autonomous robot using monocular vision. *Jornadas de Automática*, 46. <https://doi.org/10.17979/ja-cea.2025.46.12227>

Resumen

Este trabajo presenta un sistema completo de percepción, aproximación y agarre de un tubo en un entorno subacuático, utilizando un robot equipado únicamente con una cámara monocular como sensor visual. La ausencia de sensores de profundidad plantea un desafío adicional, ya que toda la información espacial se obtiene a partir de imágenes 2D, lo que incrementa la complejidad de la percepción y la planificación de acciones. La detección y segmentación del tubo se realiza con un modelo YOLOv8, entrenado específicamente para este tipo de entorno. A partir de la imagen segmentada, se calculan tanto las características del tubo como los puntos de agarre. Esta información permite al robot posicionarse correctamente frente al tubo y realizar el agarre con una pinza simple. El sistema se desarrolló en ROS Noetic y se han realizado algunas pruebas en tres escenarios diferentes: en el simulador Stonefish, en el tanque del CIRTESU y en condiciones reales en el Puerto de Castellón.

Palabras clave: Navegación y control de sistemas marinos, Vehículos submarinos autónomos, Percepción, Planificación de movimiento, Procesamiento de imágenes

Perception, approach, and grasping of underwater pipes by an autonomous robot using monocular vision

Abstract

This work presents a complete system for the perception, approach, and grasping of a pipe in an underwater environment, using a robot equipped solely with a monocular camera as its visual sensor. The absence of depth sensors introduces an additional challenge, as all spatial information must be obtained from 2D images, increasing the complexity of both perception and motion planning. The detection and segmentation of the pipe are performed using a YOLOv8 model specifically trained for this type of environment. Based on the segmented image, both the geometric features of the pipe and the grasping points are computed. This information enables the robot to position itself correctly in front of the pipe and perform the grasp using a simple gripper. The system was developed in ROS Noetic, and several tests have been conducted in different scenarios: the Stonefish simulator, the CIRTESU test tank, and real-world conditions in the Port of Castellón.

Keywords: Marine system navigation and control, Autonomous underwater vehicles, Perception and sensing, Motion planning, Image processing

* Autor para correspondencia: al416992@uji.es
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

1. Introducción

La robótica subacuática tiene una gran relevancia debido a sus aplicaciones en inspección, mantenimiento y exploración en entornos marinos y acuáticos. Sin embargo, la percepción visual bajo el agua presenta retos concretos, como la distorsión de la imagen causada por la refracción y la dispersión de la luz en el medio, las variaciones en la iluminación ambiental y la presencia de partículas en suspensión que afectan la claridad (Pi et al., 2021), (López-Barajas et al., 2025).

En este contexto, los sistemas robóticos equipados únicamente con cámaras monoculares representan una solución económica y versátil, pero su uso implica dificultades adicionales, dado que la percepción espacial debe inferirse a partir de imágenes 2D. Este trabajo presenta un sistema completo que aborda estos retos para la detección, aproximación y agarre de un tubo subacuático, el cual se encuentra colocado horizontalmente y sumergido en la piscina, sin estar en contacto con el fondo, utilizando un robot BlueROV (Blue Robotics, 2016) cuya percepción se basa exclusivamente en una cámara monocular.

Para la detección y segmentación del tubo, se emplea un modelo de aprendizaje profundo basado en YOLOv8, entrenando específicamente para condiciones subacuáticas y orientado a reconocer la forma particular del tubo. A partir de la segmentación, se extraen características geométricas relevantes que facilitan la planificación de la trayectoria de aproximación del robot hacia el objeto. Asimismo, se calculan puntos óptimos de agarre, lo que permite un posicionamiento preciso y una manipulación eficaz mediante una pinza simple.

La implementación del sistema se realizó sobre ROS Noetic (Open Source Robotics Foundation, 2020), lo que garantiza una arquitectura modular y escalable. Se han llevado a cabo diversas pruebas en distintos entornos, desde simulación con el simulador Stonefish hasta escenarios reales en el tanque de CIRTESU y el Puerto de Castellón.

2. Percepción

2.1. Segmentación del tubo

Para la detección y segmentación del tubo subacuático se utilizó el modelo de aprendizaje profundo YOLOv8 (Ultralytics, 2023), reconocido por su capacidad para realizar detección en tiempo real con alta precisión y eficiencia computacional. YOLO (You Only Look Once) es una arquitectura que procesa la imagen completa en una sola pasada, lo que permite una detección rápida y precisa de objetos dentro de la escena, siendo especialmente adecuada para entornos donde la velocidad y la eficiencia son cruciales.

Se preparó un conjunto de datos compuesto por 1140 imágenes del tubo capturadas desde distintas posiciones, condiciones lumínicas y diversos entornos, incluyendo simulación y tanque del CIRTESU. Para mejorar la capacidad de generalización del modelo, se aplicaron técnicas de aumento de datos, como rotaciones, cambios de brillo y saturación, ampliando el *dataset* a un total de 2742 imágenes. Estas imágenes se dividieron en un 90 % para entrenamiento, 5 % para validación y 5 % para prueba.

Se utilizó la versión ligera YOLOv8n con capacidad de segmentación (YOLOv8n-seg.pt) por ser adecuada para sistemas robóticos con recursos limitados. El modelo fue entrenado durante 50 epochs, aprendiendo a detectar y segmentar el tubo en diversas condiciones subacuáticas (Figura 1).

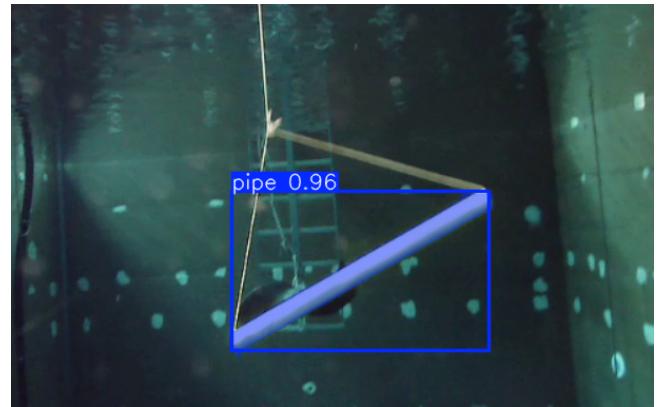


Figura 1: Resultado de la detección y segmentación del tubo en el tanque del CIRTESU utilizando el modelo YOLOv8n.

2.2. Calculo de características y puntos de agarre

Una vez realizada la segmentación del tubo, se procede a calcular los puntos óptimos de agarre que guiarán la manipulación del robot. Para ello, se utilizan los momentos geométricos de la imagen segmentada hasta el segundo orden (1) (González and Wintz, 1987).

$$\begin{aligned} \mu_{00} &= m_{00} & ; \quad \mu_{10} = \mu_{01} = 0 \\ \mu_{02} &= m_{02} - \bar{y}m_{01} \\ \mu_{11} &= m_{11} - \bar{y}m_{10} \\ \mu_{12} &= m_{12} - 2\bar{y}m_{11} - \bar{x}m_{02} + 2\bar{y}^2m_{10} \\ \mu_{20} &= m_{20} - \bar{x}m_{10} \\ \mu_{21} &= m_{21} - 2\bar{x}m_{11} - \bar{y}m_{20} + 2\bar{x}^2m_{01} \end{aligned} \quad (1)$$

Los valores \bar{x} y \bar{y} representan las coordenadas del centroide (o centro de masa) de la distribución de píxeles de la máscara y m_{pq} hace referencia a los momentos de primer orden (2).

$$m_{pq} = \sum_{j=0}^{N_x} \sum_{i=0}^{N_y} x_j^p y_i^q I(x_j, y_i) \quad (2)$$

Donde las coordenadas x_j y y_i indican la posición horizontal y vertical de cada píxel, respectivamente, mientras que $I(x_j, y_i)$ es el valor de intensidad del color del píxel en esa posición.

Los momentos y centro de masas de y se pueden calcular mediante funciones disponibles en OpenCV (OpenCV Team, 2024). Estos momentos permiten determinar el centroide del tubo, así como los ejes de máxima (I_{\max}) y mínima inercia (I_{\min}), que caracterizan su orientación principal.

Los valores de I_{\min} y I_{\max} se calculan según las ecuaciones:

$$I_{\min} = \frac{\mu_{20} + \mu_{02} - \sqrt{4\mu_{11}^2 + (\mu_{20} - \mu_{02})^2}}{2} \quad (3)$$

$$I_{\max} = \frac{\mu_{20} + \mu_{02} + \sqrt{4\mu_{11}^2 + (\mu_{20} - \mu_{02})^2}}{2} \quad (4)$$

Con estos valores se ajusta una elipse que aproxima la forma del tubo (Jain, 1989), cuyos semiejes mayor a y menor b se calculan mediante:

$$a = \left(\frac{4}{\pi}\right)^{1/4} \left(\frac{I_{\max}^3}{I_{\min}}\right)^{1/8} ; \quad b = \left(\frac{4}{\pi}\right)^{1/4} \left(\frac{I_{\min}^3}{I_{\max}}\right)^{1/8} \quad (5)$$

Estos puntos y ejes se representan visualmente en la imagen capturada por la cámara, mostrando la orientación y posición del tubo (Figura 2).

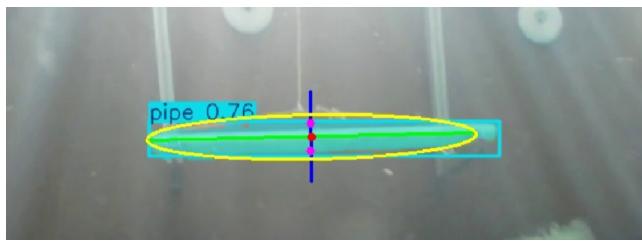


Figura 2: Representación gráfica de las características geométricas del tubo: el punto rojo indica el centroide, las líneas azul y verde muestran los ejes de máxima y mínima inercia respectivamente, y los puntos magentas señalan los puntos óptimos de agarre.

Para estimar la distancia entre el tubo y el robot, es medido el diámetro aparente del tubo en píxeles a partir de imágenes capturadas a distintas distancias conocidas, expresadas en metros. Sobre estos datos se aplica una regresión basada en una relación inversa, dado que el diámetro observado aumenta conforme disminuye la distancia al objeto. Adicionalmente, se compararon los resultados experimentales con el modelo teórico de cámara estenopeica (modelo pinhole), que permite estimar la distancia según la ecuación (6) (Torralba et al., 2024).

$$Z : \text{distancia cámara-objeto (m)} \\ f = 470,48 : \text{distancia focal (px)} \\ D = 0,04 : \text{diámetro real del tubo (m)} \\ d : \text{diámetro observado (px)} \quad (6)$$

Los datos de distancia focal de la cámara se obtienen rea- lizando su calibración.

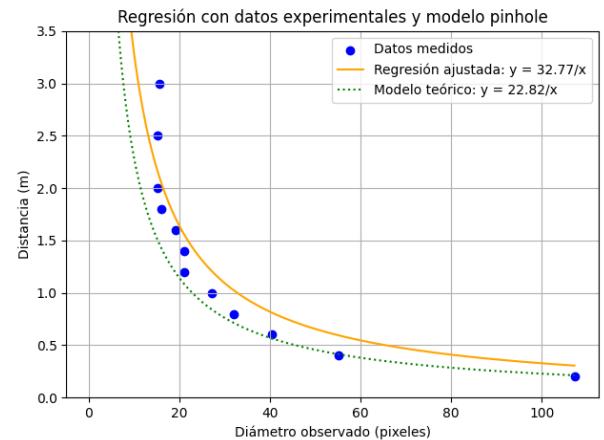


Figura 3: Regresión ajustada con datos experimentales y modelo teórico pinhole representados en colores amarillo y verde, respectivamente. Los puntos azules corresponden a las mediciones experimentales.

Tabla 1: Métricas de error entre la regresión ajustada, el modelo teórico y los datos experimentales

	MAE	MSE	RMSE
Regresión vs. Teórico	0.4257	0.2492	0.4992
Teórico vs. Datos	0.3797	0.3374	0.5809
Regresión vs. Datos	0.2176	0.0882	0.2970

En la Tabla 1 se aprecia que la regresión ajustada proporciona un mejor ajuste a los datos experimentales en comparación con el modelo teórico. Aunque la imagen de la cámara ha sido calibrada, es posible que existan desviaciones atribuibles a la morfología particular del tubo o a las condiciones variables de iluminación durante la captura. No obstante, el comportamiento de la regresión se aproxima al modelo teórico, lo que confirma su validez y utilidad para estimar distancias en este contexto.

3. Planificación de agarre

En esta sección se presenta el sistema de control que permite al robot BlueROV desplazarse y posicionarse de forma autónoma para manipular un tubo subacuático, utilizando únicamente la información de una cámara monocular.

3.1. Aproximación y control de posición

El control de posición horizontal utiliza IBVS (Image-Based Visual Servoing) (Raviv and Herman, 1993), basado en medir el error visual en píxeles entre el centroide del tubo detectado por la cámara monocular y la posición objetivo de la garra asumida en el centro de la imagen. Estas desviaciones en los ejes x e y se transforman directamente en comandos cinemáticos mediante un controlador proporcional P, simplificando así la aproximación precisa al tubo sin necesidad de reconstrucción tridimensional.

Una vez que el robot logra centrar la garra respecto al tubo en los ejes, inicia el movimiento de aproximación lineal hacia el objeto. Durante esta aproximación, el sistema continúa realizando ajustes en tiempo real para mantener la alineación, corrigiendo cualquier desviación que pueda surgir debido al movimiento o a las perturbaciones del entorno. El robot avanza

hasta alcanzar una distancia estimada de 0.2 metros del tubo, considerada óptima para cerrar la pinza y realizar un agarre seguro y efectivo.

3.2. Corrección de orientación

La corrección del ángulo de orientación (yaw) del robot respecto al tubo se aborda mediante un método empírico basado en la relación visual entre el ancho y el largo del tubo detectado en la imagen monocular. Intentar calcular directamente el ángulo utilizando fórmulas trigonométricas para triangular y estimar la orientación resultó poco fiable, debido a errores acumulados y a la complejidad del entorno subacuático. Para determinar un rango fiable de esta proporción, se ha monitoreado durante un largo periodo de tiempo la relación ancho/largo del tubo en diferentes distancias y bajo variadas condiciones de iluminación usando PlotJuggler (Rocco, 2016). Los valores obtenidos se resumen en la Figura 4, donde se puede observar que los valores de esta proporción están dentro del rango (11,5-7,5), nunca superando las 11,5 unidades.

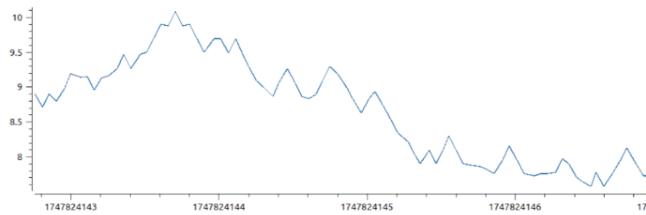


Figura 4: Valores de la proporción ancho/largo del tubo en posiciones con variaciones de yaw, registrados durante un período prolongado bajo diferentes condiciones.

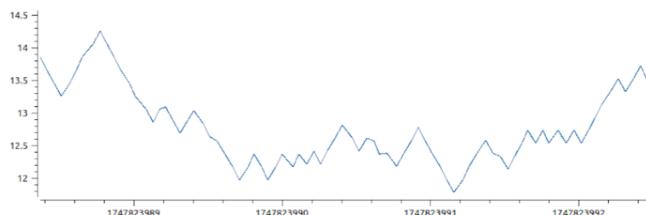


Figura 5: Valores de la proporción ancho/largo del tubo en posición recta, monitoreados durante un período prolongado bajo distintas condiciones.

Asimismo, se ha desarrollado un método basado en los resultados mostrados en la Figura 5, donde se observa que los valores de la proporción ancho/largo superan consistentemente las 12 unidades cuando el tubo está torcido. Con base en este umbral, se puede asumir que el tubo presenta una torsión significativa, por lo que es necesario calcular la dirección de esta torsión (o ángulo de yaw) para corregir la orientación del robot. Para ello, se mide el diámetro en ambos extremos del tubo, como se muestra en la Figura 6, y a partir de estas medidas se determina hacia qué lado debe realizarse el giro en yaw. La corrección se aplica mediante un controlador proporcional basado en la diferencia del diámetro de los segmentos del tubo. Se calcula la media móvil de los últimos 10 valores de diámetro en cada extremo; si la media del diámetro en un extremo es consistentemente mayor que la del otro, se interpreta que el tubo está orientado hacia ese lado, indicando la dirección del yaw necesaria para corregir la orientación. De

esta forma, el robot ajusta su giro en función del extremo con mayor diámetro aparente, logrando una alineación precisa con el tubo.

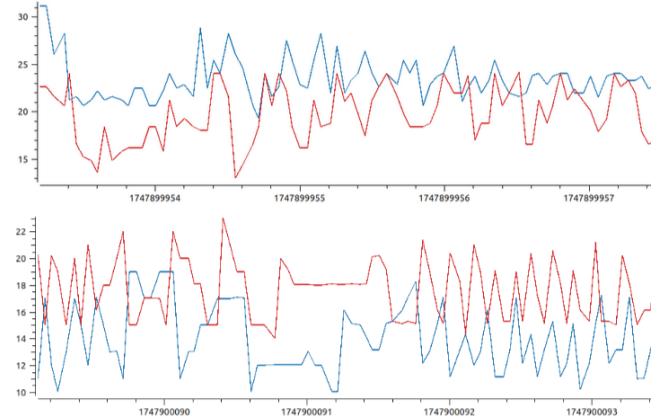


Figura 6: Evolución temporal del diámetro aparente en los extremos del tubo. En la gráfica superior, se observa una orientación con yaw positivo y en la gráfica inferior, se muestra una orientación con yaw negativo. El extremo derecho está representado en azul y el del extremo izquierdo en rojo.

4. Arquitectura de control

La implementación del sistema de control y percepción se realiza utilizando ROS Noetic, que facilita la integración modular de sensores, algoritmos y actuadores. ROS provee una infraestructura robusta basada en nodos y tópicos que permite el intercambio eficiente de mensajes entre los distintos componentes del sistema, favoreciendo la escalabilidad y el desarrollo distribuido.

En esta sección se describe la arquitectura de control diseñada para el robot BlueROV, detallando los principales nodos implementados, su función y cómo se comunican entre sí para lograr la percepción, navegación y manipulación autónoma del tubo. Esta arquitectura se ilustra en la Figura 7, donde se muestran los nodos principales y los tópicos utilizados para la comunicación entre ellos.

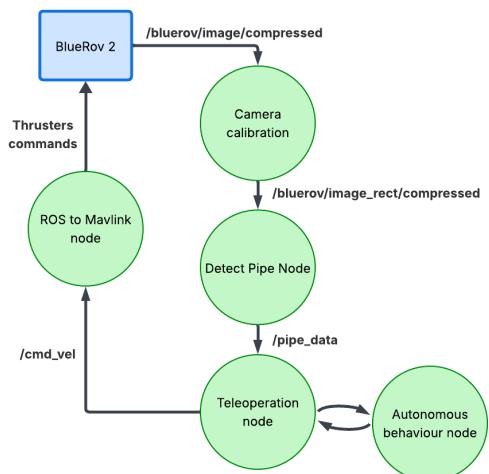


Figura 7: Diagrama de arquitectura ROS.

El sistema empieza con la captura de imágenes por la cámara del BlueROV, que se transmiten en formato comprimido por el tópico /bluerov/image/compressed al no-

do de calibración. Este nodo se encarga de corregir las distorsiones propias del lente y otros efectos ópticos para generar una imagen rectificada, que es publicada en el tópico `/bluerov/image_rect/compressed`.

A continuación, el nodo de detección del tubo (Detect Pipe Node) recibe la imagen rectificada, ejecuta el modelo de segmentación y extrae características relevantes del objeto. Estos datos se publican en el tópico `/pipe_data`, que utiliza un mensaje personalizado (*custom message*) diseñado específicamente para este sistema. El mensaje tiene la siguiente estructura.

```
pipe_data:
  center_px: geometry_msgs/Point
  imin: geometry_msgs/Point
  imax: geometry_msgs/Point
  g1: geometry_msgs/Point
  g2: geometry_msgs/Point
  center_error: geometry_msgs/Vector3
  pipe_prop: float32
  extr: geometry_msgs/Point
  distance: float32
```

En este mensaje, `center_px` indica el centroide del tubo en píxeles, `imin` e `imax` representan los ejes de mínima y máxima inercia, `g1` y `g2` son los puntos de agarre, `center_error` es el desplazamiento del centroide del tubo al centro, `pipe_prop` la proporción ancho/largo del tubo, `extr` los diámetros en los extremos, y `distance` el diámetro real estimado en metros.

El nodo de teleoperación recibe comandos manuales de movimiento a través de un *joystick* y los publica en el tópico `cmd_vel` para controlar el robot en modo manual. Desde este nodo también se puede activar y desactivar el comportamiento autónomo, que envía sus propios comandos en el mismo tópico para la navegación y manipulación automática del tubo. De esta forma, el robot puede alternar entre control manual y autónomo utilizando el mismo canal de comunicación.

Para evitar conflictos, ambos modos de control coordinan el acceso al tópico `cmd_vel`, garantizando una transición fluida y segura entre ambos. Los comandos de velocidad y orientación generados por estos nodos se transmiten al nodo ROS to Mavlink, que traduce los comandos lineales y angulares contenidos en el mensaje `geometry_msgs/Twist` a señales PWM, las cuales accionan directamente los thrusters del BlueROV para realizar los movimientos físicos requeridos.

El sistema opera bajo un esquema cíclico en el que los datos de percepción se actualizan continuamente y se retroalimentan en tiempo real al módulo de control, garantizando así una respuesta dinámica y precisa del sistema durante su operación.

5. Resultados

Para validar el rendimiento del modelo de segmentación, se ha llevado a cabo una fase de evaluación utilizando datos sobre tres entornos distintos: el simulador, el tanque de pruebas del CIRTESU y el puerto de Castellón. El conjunto de datos empleado para el entrenamiento se compone en un 85 % de imágenes capturadas en el tanque del CIRTESU, un 15 % en el simulador, y no incluye imágenes del entorno del puerto.

Con el objetivo de analizar la capacidad de generalización del modelo, se realizarán pruebas independientes sobre conjuntos de test pertenecientes a cada uno de estos entornos, lo que permitirá evaluar su comportamiento bajo distintas condiciones.

Tabla 2: Resultados de segmentación del tubo en distintos entornos: simulador Stonefish, tanque de agua del CIRTESU y puerto de Castellón. Las métricas de evaluación mostradas en la tabla son Precisión, Recall, mAP@0.5 y mAP (mAP@0.5:0.95).

	Precisión	Recall	mAP@0.5	mAP
Simulador	0.971	0.895	0.973	0.797
Tanque	0.934	1.000	0.974	0.876
Puerto	0.980	0.983	0.995	0.804

Los resultados de la Tabla 2 muestran que el modelo de segmentación ofrece un rendimiento sólido y coherente en los tres entornos evaluados. Se observan valores elevados tanto en precisión como en mAP, y una buena capacidad de detección reflejada en el recall, con valores cercanos o iguales a 1. Además, el modelo demuestra una gran capacidad de generalización, obteniendo buenas métricas en todos los entornos, incluso en el del puerto, donde no se utilizaron imágenes durante el entrenamiento, lo que refleja su robustez y su adaptabilidad frente a distintas condiciones.

Para evaluar el desempeño del sistema de percepción durante la fase de aproximación al objetivo, se registraron los datos de odometría del robot. El objetivo de esta prueba es verificar la capacidad del sistema para converger hacia la posición deseada, condición necesaria para ejecutar con precisión la maniobra de agarre del tubo.

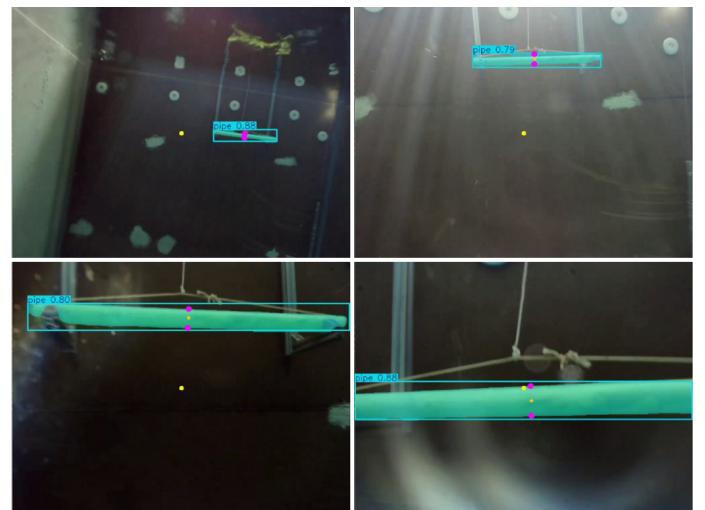


Figura 8: Evolución del estado del robot con respecto al tubo durante su aproximación.

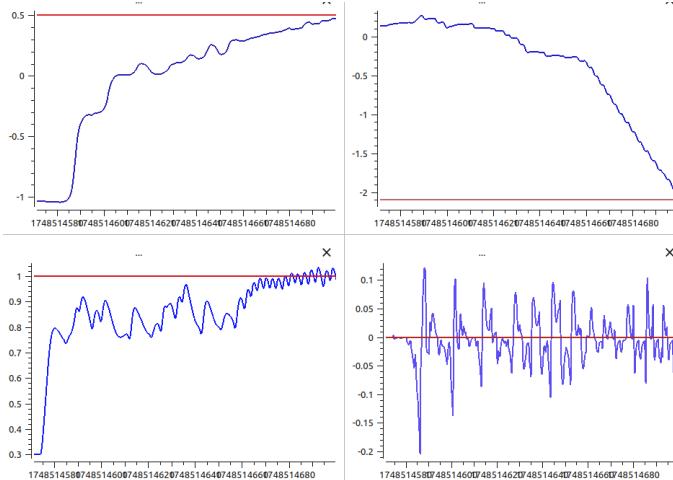


Figura 9: Evolución de la posición y orientación del robot durante la aproximación a la posición de agarre. El eje horizontal representa el número de iteraciones del sistema de control, mientras que el eje vertical muestra los valores de posición (en metros) y orientación (en radianes).

Para evaluar el rendimiento del sistema de movimiento, se registraron los valores de odometría del robot durante el proceso de aproximación al objetivo. El propósito de esta evaluación es verificar la capacidad del sistema para converger hacia la posición deseada, lo cual es fundamental para realizar correctamente la maniobra de agarre del tubo.

La Figura 9 presenta cuatro gráficas que ilustran la evolución de las variables de estado del robot durante el proceso de aproximación al objetivo. La primera gráfica, ubicada en la parte superior izquierda, muestra la componente de posición en el eje X; la segunda gráfica, en la parte superior derecha, corresponde a la componente de posición en el eje Y; la tercera gráfica, situada en la parte inferior izquierda, representa la componente de posición en el eje Z; y la cuarta gráfica, en la parte inferior derecha, muestra la evolución de la orientación en el eje yaw, es decir, la rotación del robot alrededor del eje vertical.

En cada una de estas gráficas, la línea azul indica los valores registrados por la odometría del robot durante su movimiento autónomo, mientras que la línea roja representa la posición u orientación objetivo final. La cercanía entre las curvas registradas y las de referencia muestra cómo las variables de estado del robot convergen progresivamente hacia los valores deseados. A lo largo del movimiento, el sistema de percepción y control realiza ajustes continuos para alinear la posición y orientación del robot con el objetivo. Esta convergencia final garantiza que el robot se sitúe en la ubicación adecuada para ejecutar de forma precisa la acción de agarre.

6. Conclusiones

Este trabajo ha demostrado la viabilidad de utilizar un sistema de percepción basado únicamente en visión monocular para la detección, aproximación y agarre de tubos en entornos subacuáticos. A pesar de las limitaciones inherentes a la ausencia de sensores de profundidad, el sistema desarrollado ha sido capaz de estimar con precisión tanto la posición como la orientación del objeto, gracias a un modelo de segmentación

eficiente y a técnicas geométricas para el cálculo de puntos de agarre.

Los resultados experimentales, obtenidos en entornos simulados, de laboratorio y reales, confirman que el robot logra converger de forma estable y precisa hacia la posición deseada, realizando ajustes constantes durante el movimiento hasta alcanzar la configuración adecuada para ejecutar el agarre. Esta capacidad de alineación progresiva es fundamental para garantizar la fiabilidad del sistema en tareas de manipulación subacuática, incluso bajo condiciones de visibilidad limitadas o perturbaciones externas.

El enfoque propuesto representa una solución de bajo costo, escalable y adaptable a otros entornos o tipos de objetos. Como trabajo futuro, se plantea la mejora del sistema de control de movimiento mediante la incorporación de controladores más complejos. Además, se propone continuar el desarrollo del sistema de agarre, dotándolo de mayor robustez y adaptabilidad para enfrentar tareas de manipulación más complejas en condiciones subacuáticas variables. Finalmente, será fundamental validar todo el sistema en condiciones reales en el entorno del puerto, lo que permitirá evaluar su desempeño frente a factores ambientales no controlados, como corrientes y consolidar su aplicabilidad en escenarios operativos reales.

Agradecimientos

Este trabajo ha sido realizado gracias al proyecto COOPERAMOS (PID2020-115332RB-C31), financiado por el Ministerio de Ciencia e Innovación (MCIN/AEI/10.13039/501100011033).

Referencias

- Blue Robotics, 2016. Bluerov2: The world's most affordable high-performance rov. BlueROV2 datasheet, revised May 2025.
URL: <https://bluerobotics.com/store/rov/bluerov2/bluerov2/>
- Gonzalez, R. C., Wintz, P., 1987. Digital Image Processing, 2nd Edition. Addison-Wesley, Reading, Massachusetts.
- Jain, A. K., 1989. Fundamentals of Digital Image Processing. Prentice Hall, Englewood Cliffs, NJ.
- López-Barajas, S., Sanz, P. J., Marín-Prades, R., Echagüe, J., Realpe, S., 2025. Network congestion control algorithm for image transmission—hri and visual light communications of an autonomous underwater vehicle for intervention. Future Internet 17 (1).
URL: <https://www.mdpi.com/1999-5903/17/1/10>
DOI: 10.3390/fi17010010
- Open Source Robotics Foundation, 2020. Robot Operating System (ROS) – Noetic Nijmegen. <https://www.ros.org>, released May 23, 2020.
URL: <https://www.ros.org>
- OpenCV Team, 2024. Image moments - opencv documentation. https://docs.opencv.org/4.x/d49/tutorial_moments.html.
- Pi, R., Cieślak, P., Ridao, P., Sanz, P. J., 2021. Twinbot: Autonomous underwater cooperative transportation. IEEE Access 9, 37668–37684.
DOI: 10.1109/ACCESS.2021.3063669
- Raviv, D., Herman, M., 1993. Visual servoing from 2-d image cues. In: Aloimonos, Y. (Ed.), Active Perception. Lawrence Erlbaum Associates.
- Rocco, D. F., 2016. Plotjuggler: Real-time data visualization tool. <https://plotjuggler.io/>.
- Torralba, A., Isola, P., Freeman, W., 2024. Foundations of Computer Vision. Adaptive Computation and Machine Learning series. MIT Press.
URL: <https://mitpress.mit.edu/9780262048972-foundations-of-computer-vision/>
- Ultralytics, 2023. Yolov8: State-of-the-art real-time object detection and segmentation. <https://github.com/ultralytics/yolov8>.