



**Conocimiento y transferencia de
tecnología sobre vehículos aéreos y
acuáticos para el desarrollo transfronterizo
de ciencias marinas y pesqueras**
(POCTEP 0622-KTTSEADRONES-5-E)

Informe técnico

**Informe sobre el desarrollo de sensores en sistemas fijos
(boyas) y ROVs para su uso en estuarios y explotaciones
piscícolas**

Producto final – Acción 2.3A (adenda) | 28/02/2022

Nombre del Proyecto: KTTSeaDrones - Conocimiento y transferencia de tecnología sobre vehículos aéreos y acuáticos para el desarrollo transfronterizo de ciencias marinas y pesqueras

Código del proyecto: POCTEP 0622-KTTSEADRONES-5-E

Datos del proyecto

Localización	España y Portugal		
Fecha de inicio	01/10/2017		
Fecha de finalización	31/12/2021	Prórroga hasta	30/10/2022
Financiación total	€ 1.120.864,43		
Contribución UE	€ 840.648,32		
Cofinanciación UE (%)	75%		

Datos del beneficiario principal (BP)

Beneficiario principal	Universidad de Huelva
Persona de contacto	Vicerrector de Investigación y Transferencia
Dirección	C/ Cantero Cuadrado 6, 21004 Huelva
Teléfono	+34 959 218013
E-mail	kttseadrones@gmail.com
Página WEB (proyecto)	https://kttseadrones.wixsite.com/kttseadrones

Datos del documento

Nombre	Informe sobre el desarrollo de sensores en sistemas fijos (boyas) y ROVs para su uso en estuarios y explotaciones piscícolas
Acción	Acción 2.3A. Desarrollo de un sistema fijo (boya) y un vehículo operado remotamente (ROV) con la capacidad de recoger, transmitir y almacenar información sobre las condiciones físico-químicas en estuarios y en balsas de engorde en instalaciones de producción acuícola
Autores	Alejandro Garrocho Cruz (UHU) Fernando Gómez Bravo (UHU) Juan Carlos Gutiérrez Estrada (UHU) Inmaculada Pulido Calvo (UHU) Jairo Castro Gutiérrez (UHU) Samuel López Domínguez (UHU) Antonio Peregrín Rubio (UHU)
Equipo técnico	Juan Antonio Gómez Galán (UHU) Raúl Jiménez Naharro (UHU) Rafael López de Ahumada (UHU) Manuel Sánchez Raya (UHU) Manuel Pedro Carrasco (UHU)

Javier Aroba Páez (UHU)
Francisco José Moreno Velo (UHU)
José Manuel Martín Ramos (UHU)
Ana María Roldán Ruiz (UHU)
Francisco Alfredo Márquez Hernández (UHU)
Antonio Ángel Márquez Hernández (UHU)
Ignacio de la Rosa Lucas (UHU)
Fecha
28/02/2022
Versión (nº)
1.0

Las opiniones presentadas en este informe comprometen únicamente a los beneficiarios del proyecto y, en consecuencia, no representan en ningún caso la opinión oficial de los órganos de gestión del POCTEP.

ÍNDICE

RESUMEN	4
1. INTRODUCCIÓN	4
2. CONTROL AUTOMÁTICO	5
2.1. FUNCIONES CALLBACK	6
2.2. ALGORITMO DE CONTROL	7
2.3. RESULTADOS Y CONCLUSIONES	9
3. ESTIMACIÓN DE ABUNDANCIA DE PECES	11
3.1. INTRODUCCIÓN	11
3.2. PROCESAMIENTO PREVIO	13
3.3. MATRIZ FRONTERA	15
3.4. MÉTODO DE AGRUPAMIENTO (CLUSTERING)	16
3.5. RESULTADOS Y CONCLUSIONES	19
4. BIBLIOGRAFÍA	22

Acción 2.3 Integración de las distintas herramientas y sensores, en función de las distintas misiones para que se diseñará la boya y el ROV. Ensayo en estuarios y piscifactorías

RESUMEN

En este informe se ha desarrollado una descripción cronológica y detallada de los procesos realizados desde julio a diciembre de 2021 para la consecución de la acción 2.3 (“Integración de las distintas herramientas y sensores, en función de las distintas misiones para que se diseñará la boya y el ROV. Ensayo en estuarios y piscifactorías”), correspondiente a la UHU. Permitiendo hacer un seguimiento del progreso y conocer cuáles han sido nuestras tareas, ensayos en estuarios y piscifactorías y objetivos conseguidos.

1. INTRODUCCIÓN

En el informe técnico anterior se describían los resultados alcanzados hasta junio de 2021, mostrándose los desarrollos realizados en lo concerniente al control manual y la representación de datos, tanto de los sonares como de los datos telemétricos del ROV. En los apartados finales, de dicho documento se realizaba una breve descripción del planteamiento que el equipo de investigación proponía para abordar los problemas de control de orientación (*heading*) y de desplazamiento. Estos últimos resultados se enmarcaban en el objetivo final propuesto al inicio del proyecto: Dotar de cierto nivel de autonomía al SibiuPro para transformarlo de un ROV convencional a un ROV Híbrido (HROV) es decir, un vehículo que mantiene al operador a cargo del control de alto nivel y a la vez introduce la capacidad del vehículo de realizar maniobras de forma autónoma, supeditadas a las decisiones que tome el operador en función de los objetivos de la operación a realizar.

La prueba anterior al periodo en el que se hace referencia en este informe fue llevada a cabo en Salinas del Astur, donde se llevaron optimizados los algoritmos de control manual y de representación de datos, obteniendo unos resultados completamente satisfactorios. Mientras que, por otro lado, se realizaron pruebas de control automático donde el sistema de localización fue demasiado impreciso, provocando una ejecución con errores mayores a los deseados de los movimientos automatizados del ROV. Por este motivo, se tomó la decisión de dejar apartado la experimentación para mejorar dicha localización, centrándonos en la automatización sobre valores telemétricos que sí eran más precisos como la altura o el ángulo de cabeceo (*heading*).

Tras conseguir los objetivos con el sistema automático de control, se presentó el reto de filtrar y procesar los datos del SONAR Ping360 con el objetivo de poder detectar automáticamente objetos de interés, en nuestro caso, peces. Con este algoritmo creado mediante métodos de agrupamientos (clustering) ha sido posible detectar, cuantificar y localizar dichos objetos en tiempo real y de forma automática, rápida, eficaz y para grandes cantidades de peces.

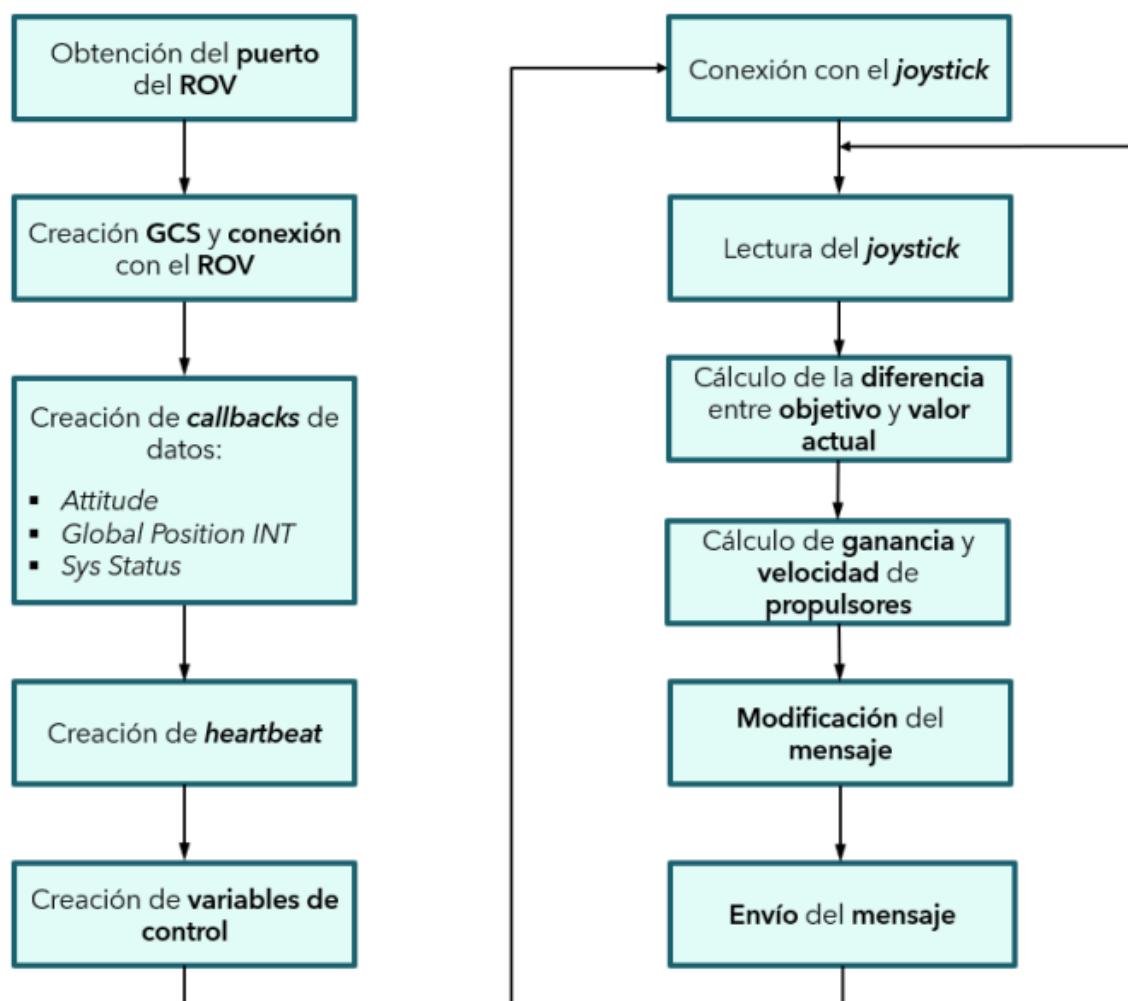
En este documento, se van a tratar los avances realizados siguiendo un orden cronológico, donde comenzaremos hablando del algoritmo de control, para luego pasar a explicar el algoritmo de estimación de la abundancia de peces en diferentes entornos.

2. CONTROL AUTOMÁTICO

La ejecución de este algoritmo permite realizar un control automático de profundidad y orientación (*heading*) de un vehículo submarino operado remotamente (ROV) de pequeño tamaño, utilizando los valores proporcionados por los sensores con los que éste ha sido dotado. En concreto, el control de profundidad se realiza utilizando las lecturas entregadas por un sensor ‘eco-sonda’ y el *heading* mediante los valores proporcionados por una Unidad de Medida Inercial (IMU).

La comunicación entre la CPU externa y el sistema de conducción del ROV está soportada por el protocolo Mavlink, implementado sobre TCP/IP, utilizando como protocolos de nivel físico y enlace el estándar Ethernet. La programación del sistema completo se ha realizado generando scripts y funciones de Matlab.

A continuación, se muestra el diagrama de bloques que representa el código creado:



Ambos controladores funcionan siguiendo la misma filosofía y se han implementado según se describe en el diagrama de flujo mostrado en la figura anterior. En el inicio del algoritmo se identifica el puerto TCP de comunicación del controlador del vehículo que permite establecer la transferencia de datos. A partir de esta información es posible definir interrupciones disparadas mediante temporizadores utilizando funciones ‘callback’ para la petición periódica de datos telemétricos. Posteriormente se implementa el algoritmo de control, el cual comienza calculando el error (la diferencia entre la variable de referencia y la última variable almacenada). El algoritmo de control específico tiene en cuenta este error para generar las consignas de control que harán que el vehículo alcance la orientación y profundidad establecidas por la referencia. Pueden implementarse distintas estrategias, desde controladores lineales (P, PID, realimentación por variables de estado etc...) a otras más sofisticadas basadas en control no lineal o control inteligente. En cualquiera de los casos, el controlador proporcionará la velocidad de los propulsores y la ganancia general del ROV para llevar a éste hasta la configuración de destino deseada.

En las siguientes secciones se describen los principales módulos desarrollados para implementar el sistema de control automático de orientación y profundidad.

2.1. FUNCIONES CALLBACK

Una vez obtenido automáticamente el puerto del ROV (aleatorio tras cada puesta en marcha) a través de comandos de Matlab, con los cuales se escribe un mensaje por consola que captura un paquete con unas características determinadas gracias al analizador de protocolos Wireshark, se configura el PC como estación de tierra y se conecta con el ROV.

Finalizado estos primeros pasos, nos centramos a continuación en los datos que vamos a requerir del ROV. Para ello, haremos uso de funciones callbacks, que son funciones que se ejecutan en respuesta a alguna acción predefinida por el usuario. En este caso, se crearán tantos temporizadores como funciones para que la recepción de datos se produzca de forma constante y con la misma frecuencia para cada función. De esta forma, se realizarán las solicitudes de los mensajes ATTITUDE, GLOBAL_POSITION_INT y SYS_STATUS cada 2, 0.5 y 60 segundos, respectivamente:

ATTITUDE

ATTITUDE (#30)

[Message] The attitude in the aeronautical frame (right-handed, Z-down, X-front, Y-right).

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
roll	float	rad	Roll angle (-pi..+pi)
pitch	float	rad	Pitch angle (-pi..+pi)
yaw	float	rad	Yaw angle (-pi..+pi)
rollspeed	float	rad/s	Roll angular speed
pitchspeed	float	rad/s	Pitch angular speed
yawspeed	float	rad/s	Yaw angular speed

Gracias a la información que proporciona podremos realizar posteriormente un estudio del comportamiento de las variables de los sensores iniciales en determinadas pruebas y movimientos.

GLOBAL POSITION INT

GLOBAL_POSITION_INT (#33)

[Message] The filtered global position (e.g. fused GPS and accelerometers). The position is in GPS-frame (right-handed, Z-up). It is designed as scaled integer message since the resolution of float is not sufficient.

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
lat	int32_t	degE7	Latitude, expressed
lon	int32_t	degE7	Longitude, expressed
alt	int32_t	mm	Altitude (MSL). Note that virtually all GPS modules provide both WGS84 and MSL.
relative_alt	int32_t	mm	Altitude above ground
vx	int16_t	cm/s	Ground X Speed (Latitude, positive north)
vy	int16_t	cm/s	Ground Y Speed (Longitude, positive east)
vz	int16_t	cm/s	Ground Z Speed (Altitude, positive down)
hdg	uint16_t	cdeg	Vehicle heading (yaw angle), 0.0..359.99 degrees. If unknown, set to: UINT16_MAX

En este mensaje se encuentran las variables a controlar, altitud (alt) y ángulo de cabecero del vehículo (hdg), es por ello por lo que realizamos la pedida de datos con mayor frecuencia.

SYS_STATUS

Field Name	Type	Units	Values	Description
onboard_control_sensors_present	uint32_t		MAV_SYS_STATUS_SENSOR	Bitmap showing which onboard controllers and sensors are present. Value of 0: not present. Value of 1: present.
onboard_control_sensors_enabled	uint32_t		MAV_SYS_STATUS_SENSOR	Bitmap showing which onboard controllers and sensors are enabled: Value of 0: not enabled. Value of 1: enabled.
onboard_control_sensors_health	uint32_t		MAV_SYS_STATUS_SENSOR	Bitmap showing which onboard controllers and sensors have an error (or are operational). Value of 0: error. Value of 1: healthy.
load	uint16_t	d%		Maximum usage in percent of the mainloop time. Values: [0-1000] - should always be below 1000
voltage_battery	uint16_t	mV		Battery voltage, UINT16_MAX: Voltage not sent by autopilot
current_battery	int16_t	cA		Battery current, -1: Current not sent by autopilot
battery_remaining	int8_t	%		Battery energy remaining, -1: Battery remaining energy not sent by autopilot
drop_rate_comm	uint16_t	c%		Communication drop rate, (UART, I2C, SPI, CAN), dropped packets on all links (packets that were corrupted on reception on the MAV)
errors_comm	uint16_t			Communication errors (UART, I2C, SPI, CAN), dropped packets on all links (packets that were corrupted on reception on the MAV)
errors_count1	uint16_t			Autopilot-specific errors
errors_count2	uint16_t			Autopilot-specific errors
errors_count3	uint16_t			Autopilot-specific errors
errors_count4	uint16_t			Autopilot-specific errors
onboard_control_sensors_present_extended **	uint32_t		MAV_SYS_STATUS_SENSOR_EXTENDED	Bitmap showing which onboard controllers and sensors are present. Value of 0: not present. Value of 1: present.
onboard_control_sensors_enabled_extended **	uint32_t		MAV_SYS_STATUS_SENSOR_EXTENDED	Bitmap showing which onboard controllers and sensors are enabled: Value of 0: not enabled. Value of 1: enabled.
onboard_control_sensors_health_extended **	uint32_t		MAV_SYS_STATUS_SENSOR_EXTENDED	Bitmap showing which onboard controllers and sensors have an error (or are operational). Value of 0: error. Value of 1: healthy.

Este mensaje solo se requiere para avisar al usuario a través de un mensaje en el caso de que la batería del ROV se encuentre demasiado descargada.

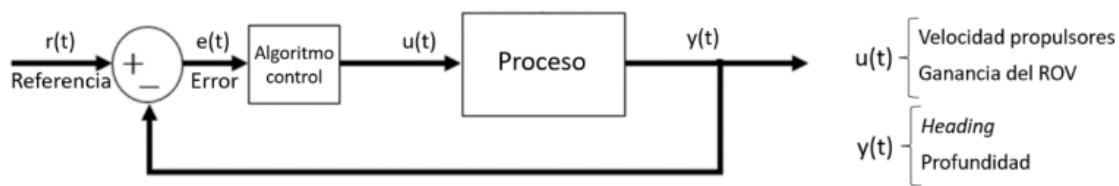
La función Heartbeat, que también se trata de una función callback con temporizador, está explicada más detalladamente en el anterior informe técnico.

2.2. ALGORITMO DE CONTROL

A continuación, se definen las variables que funcionan como referencia en el sistema de control (ángulo de cabeceo y altura que queremos que el ROV alcance), cuyo error comparado con el valor real de dichos datos consistirán en la entrada de nuestro algoritmo de control. También es necesario definir las variables de salida del bloque de control, que consisten en la ganancia general del ROV y en la velocidad de los propulsores.

Para la definición de la referencia que ha de suministrársele al controlador, se ha decidido usar algunos botones del joystick para poder modificar el ángulo de cabeceo que queremos alcanzar en tiempo real. Es por este motivo por el cual aparece el módulo de joystick en el diagrama de flujo.

Posteriormente, se realiza el proceso de control, el cual comienza calculando la diferencia entre la variable de referencia y la última variable almacenada. Dependiendo de este resultado, el algoritmo proporciona la velocidad de los propulsores y una ganancia general del ROV para llevar a éste a la posición de destino de forma óptima.



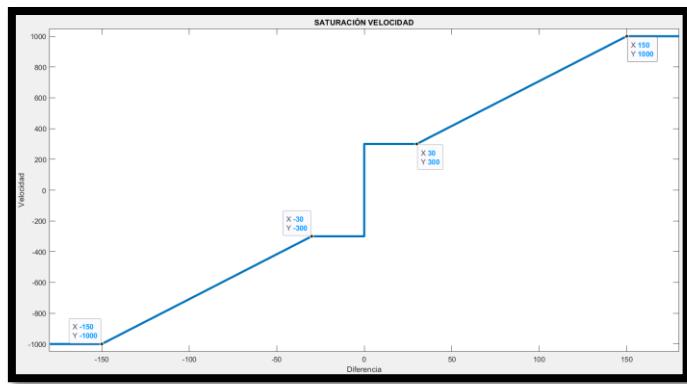
Durante la experimentación con el ROV ‘SibiuPro’, el diseño del sistema de control desarrollado ha sido calibrado heurísticamente hasta conseguir un resultado adecuado para las funciones requeridas.

Se ha desarrollado un algoritmo de control proporcional siguiendo una estrategia de acuerdo con la gráfica que veremos a continuación.

Las salidas de nuestro algoritmo se corresponden con una velocidad de los motores y una ganancia.

Para la ganancia, se ha decidido tras la experimentación que, dependiendo de si la diferencia es mayor o menor a un valor predefinido, esta variable va a tomar valores entre 0.2 y 0.4.

La velocidad va a depender del producto entre el error calculado por el factor proporcional K_p . Esta velocidad que se pasa al ROV debe tener un valor entre ± 1000 para los movimientos de cabeceo, guiñada y alabeo y de 0 a 1000 para el empuje. Por ello, es necesario saturar este valor con un límite tanto superior como inferior, ya que por debajo de 300 la velocidad de los motores es insignificante:

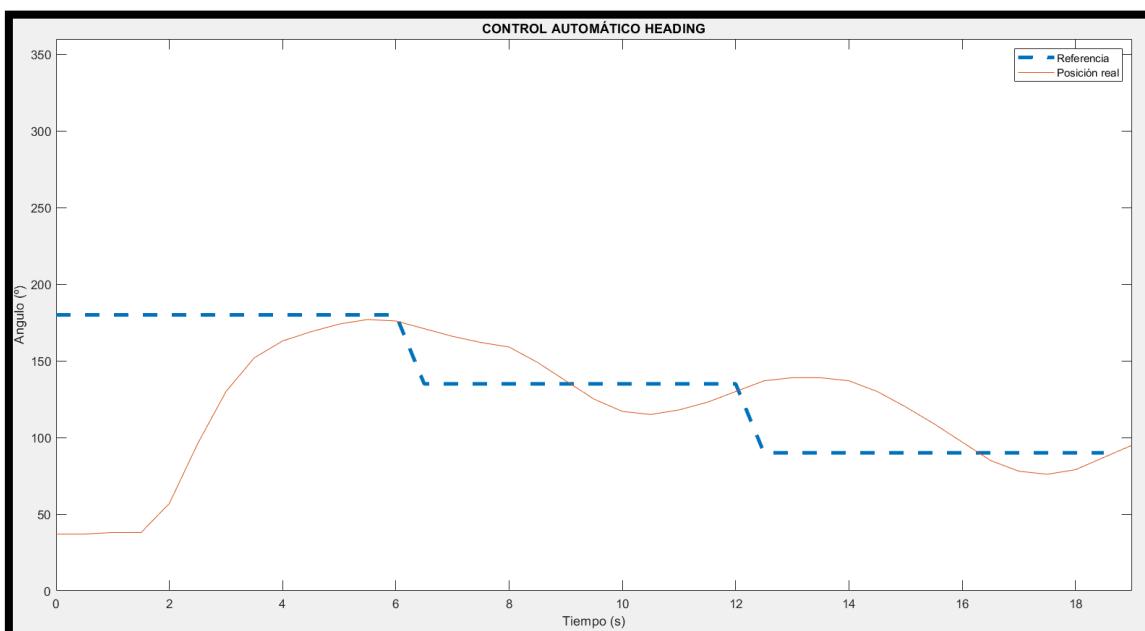


Para el caso del control de la altura, se ha seguido la misma filosofía, tomando valores en el empuje de entre 350 y 650 (siendo 500 movimiento nulo), dependiendo de si está por encima o por debajo de la altura deseada.

2.3. RESULTADOS Y CONCLUSIONES

La gráfica que se muestra a continuación representa uno de los resultados obtenidos durante la ejecución del algoritmo con el SibiuPro en la piscina situada en el patio del edificio de la Escuela Técnica Superior de Ingeniería de la Universidad de Huelva.

El objetivo de la prueba consistía en mantener el ángulo de referencia definido por el usuario, el cual podía modificarse durante la ejecución, con un rango permitido de $\pm 5^\circ$ alrededor de dicho ángulo:



Por un lado, ha sido representado con una línea discontinua azul la referencia del ángulo de cabeceo que queremos alcanzar y, por otro lado, con línea naranja y continua la posición real que nos proporcionaba los sensores del ROV. Durante los primeros segundos el ROV no se encuentra activo (armado), una vez se arma comienza su movimiento, provocando así un seguimiento de la señal de referencia.

Podemos observar que es posible optimizar la precisión, corrigiendo las sobreoscilaciones, y la velocidad de reacción. Pudiendo llevar a cabo una nueva estrategia para la obtención de las salidas que mejore estos resultados o añadiendo factores integrales o derivativos en el controlador, siempre y cuando no afecte a la velocidad de ejecución. Para el futuro, el desarrollo del controlador será una de las líneas de investigación más importantes.

3. ESTIMACIÓN DE ABUNDANCIA DE PECES

3.1. INTRODUCCIÓN

Una vez obtenidos unos resultados productivos en el ámbito de la creación de autonomía del ROV, y a la espera de recibir información sobre el incremento de precisión del sistema de localización, se pensó en poder realizar la estimación de la cantidad de peces que se encuentran en el entorno del ROV gracias a los datos del sonar Ping360.

Durante este apartado, se realizará una descripción de cómo se ha usado el MSIS (Mechanical Scanned Imaging Sonar) Ping360 para conocer el entorno durante una navegación subacuática. Como ya se ha explicado en documentos anteriores, este SONAR se encuentra montado sobre el ROV y contiene un motor que lo gira en incrementos de 0.9 grados y, gracias a ello, puede generar una imagen circular de los alrededores de la sonda con un alcance máximo de 50 metros. El resultado es un gráfico visto desde la planta, similar a lo que podría ver en un radar meteorológico en las noticias locales o en un escáner láser en un robot autónomo.

Los sonares de barrido se definen por tener un haz acústico en forma de “abanico” con un haz vertical ancho y un haz horizontal estrecho para obtener secciones transversales acústicas del entorno, para luego mostrar las secciones transversales como una sola imagen.

Cuando el sonido se transmite bajo el agua, se produce una reflexión y dispersión hasta que el agua absorbe la energía de vibración. La reverberación representa la energía que se refleja en la fuente de sonido. Para cada toma de datos en cada ángulo, el dispositivo va a proporcionar 1200 datos equidistantes sobre la intensidad del rebote en la misma dirección, partiendo desde el propio sonar hasta la distancia deseada.

Esto proporciona una matriz de datos de 1200x400 por cada vuelta completa donde cada uno de los datos hace referencia a una intensidad (desde 0 hasta 255) en un punto concreto del entorno.

subacuáticos se realiza un análisis de los datos filtrados mediante técnicas de agrupación o clustering, basadas en la asociación de elementos vecinos colindantes entre sí. Estas técnicas proporcionan un conjunto de agrupaciones de puntos detectados (clusters) y el centroide de cada una de ellas. La clasificación de cada cluster como objeto o como pez se hace en función de los siguientes factores::

- Cantidad de datos del componente.
- Distancia del centroide del componente respecto del sonar.
- Diferencia entre el ángulo mínimo y máximo del componente.

Para poder operar matemáticamente y representar las matrices de datos se hace uso de las funciones generales de Matlab, mientras que para la identificación sobre los datos procesados se utiliza la Toolbox de procesamiento de imágenes que este software dispone.

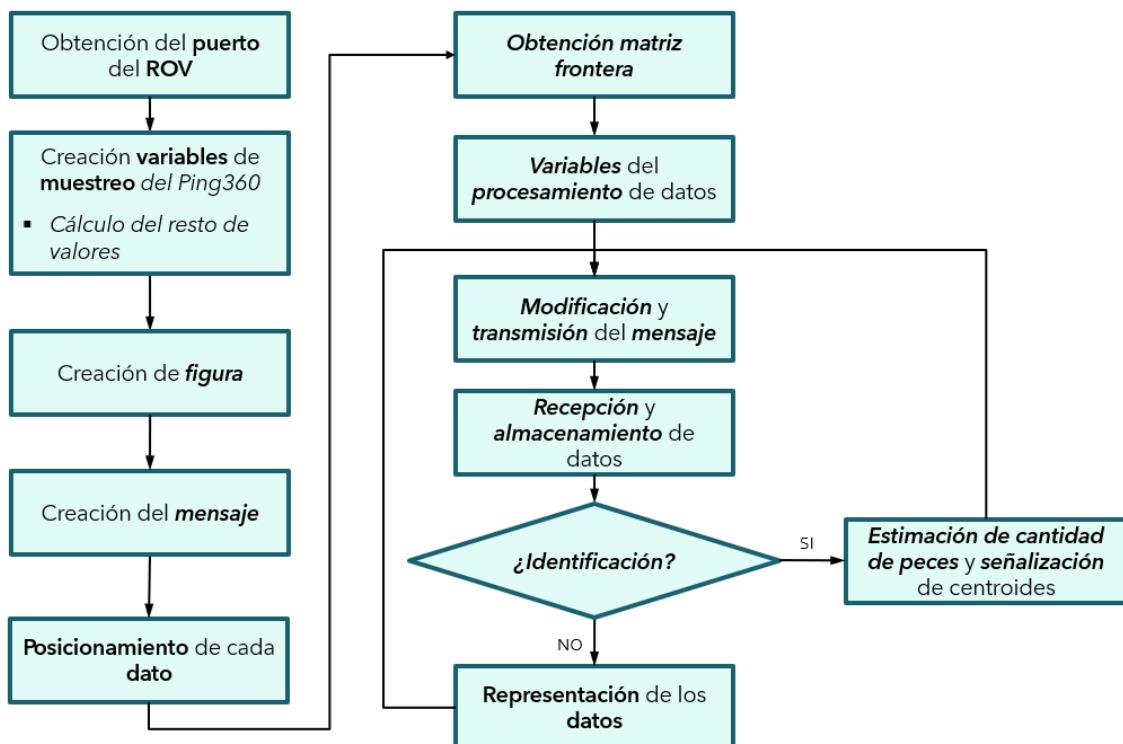
Por lo tanto, este algoritmo trata de estimar la cantidad de peces en balsas de plantas de acuicultura semi-intensivas (esteros mejorados) a partir de los datos obtenidos de un MSIS (*Mechanical Scanned Imaging Sonar*), concretamente el que se encuentra instalado sobre el

ROV SibiuPro, el SONAR Ping360. El uso de este tipo de dispositivo se debe al alto grado de turbidez del agua en este tipo de instalaciones acuícolas, lo que imposibilita la posibilidad de hacer uso de la cámara.

El sistema permite incluir la lectura del posicionamiento del ROV, estimada a través de cualquier sistema de posicionamiento para vehículos submarinos de los que existen en el mercado. Gracias a este procedimiento, es posible posicionar local o globalmente los elementos (peces u obstáculos de interés) identificados con el sistema propuesto.

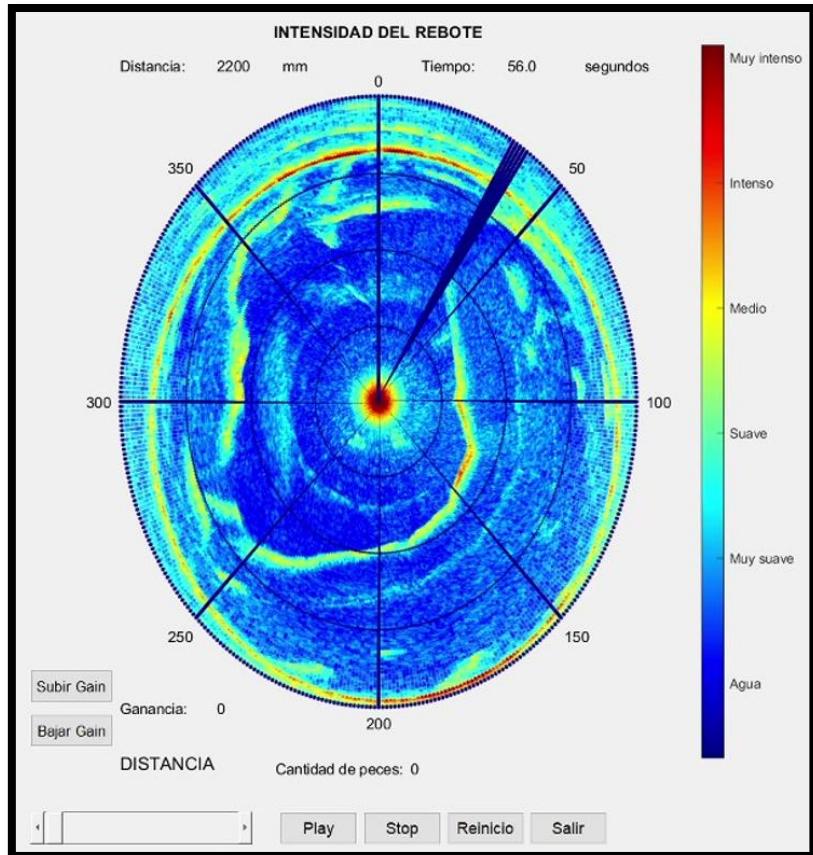
Para llevar a cabo el proceso de estimación de abundancia ha sido necesario desarrollar diversas funciones que realicen las acciones de solicitud de información al Ping360, tal y como se detalló en el anterior informe técnico. A partir de dicha información será posible obtener y procesar los datos en crudo ('raw') con las características deseadas por el usuario, y representar la información buscada en tiempo real.

En la siguiente figura, se muestra el diagrama de bloques del algoritmo completo:



Para la primera parte de este diagrama, hasta la creación de la matriz frontera, se han aprovechado los códigos de representación presentados en el anterior informe, aunque adaptando y optimizando el código y su funcionamiento a las nuevas funciones creadas.

En primer lugar, se han diseñado funciones para crear el mensaje enviado al Ping360, y generar la interfaz gráfica donde se representa la información procesada además de algunos datos más, como la distancia a la que estamos muestreando, el tiempo y la ganancia. Además el interfaz permite interactuar cambiando la ganancia, la distancia, parar, cerrar o reiniciar el muestreo, tal y como vemos en la siguiente figura:

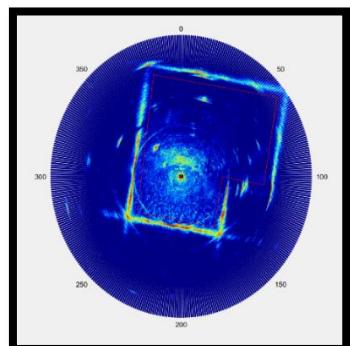


Por otro lado, tenemos las funciones encargadas representar los datos, de crear la matriz pared, dependiendo de si ésta es poligonal o circular, y de realizar el procesamiento, filtrado e identificación de los datos.

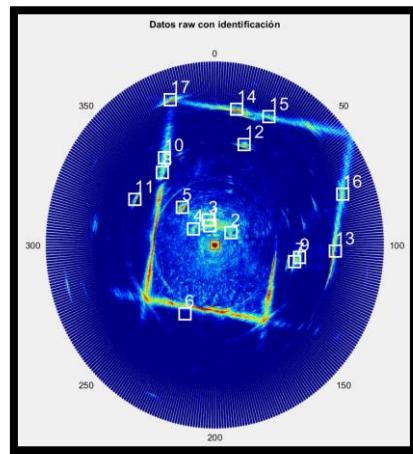
3.2. PROCESAMIENTO PREVIO

Sobre los datos en crudo del Ping360 serán realizados todos los procesos de forma sistemática hasta obtener una matriz donde solo se reflejen los datos que queremos destacar.

A continuación, se muestran los datos en crudo de una prueba realizada en la piscina del Club de Golf de Belavista con dos personas colocadas en puntos concretos y obstáculos delante del ROV.



Sobre estos datos, se ha realizado el proceso de identificación, obteniendo así una serie de detecciones erróneas debido en su mayor parte a las intensidades que hacen referencia a las paredes de la propia piscina.



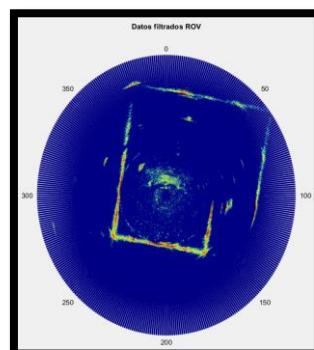
Para solucionar este problema, en primer lugar, es necesario realizar un filtrado general para aclarar los datos que hacen referencia a agua pero que, debido a su alto grado de turbidez, tienen una intensidad suficientemente alta como para provocar errores. Además, también se va a eliminar con el algoritmo el ruido que se encuentra en las inmediaciones del ROV. Realizando este proceso sobre los datos de la figura anterior se obtiene el siguiente conjunto de datos filtrados:

202	194	189	95	93	92	84	70
189	185	179	87	109	108	88	82
185	171	171	90	112	120	89	75
175	168	161	82	99	88	134	69
164	160	143	73	88	96	150	69
140	133	128	51	60	93	147	62
127	119	120	36	28	140	130	53
112	106	101	19	15	40	56	47



202	194	189	0	0	0	0	0
189	185	179	0	109	108	0	0
185	171	171	0	112	120	0	0
175	168	161	0	0	0	134	0
164	160	143	0	0	0	150	0
140	133	128	0	0	0	147	0
127	119	120	0	0	140	130	0
112	106	101	0	0	0	0	0

Como consecuencia de dicho procedimiento se consigue la siguiente figura donde se aprecia que se representan sin ruido los objetos reales dispuestos en la piscina.

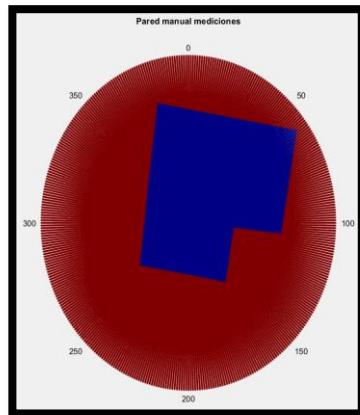


3.3. MATRIZ FRONTERA

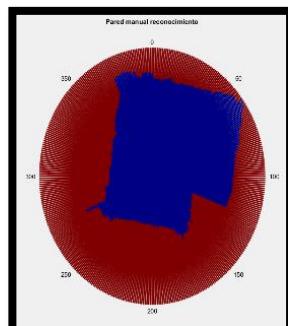
Para poder diferenciar los datos referidos a los límites del entorno de los objetos que se quieren identificar, se creará una matriz de datos de las mismas dimensiones que la matriz de datos inicial.

Esta matriz se crea con unas dimensiones similares a las de una vuelta completa del SONAR, 1200x400, donde todos los datos tienen dos posibles valores: '0' cuando forma parte del entorno, '255' cuando forma parte de la pared. De esta forma, la matriz se obtiene de la integración de dos métodos diferentes:

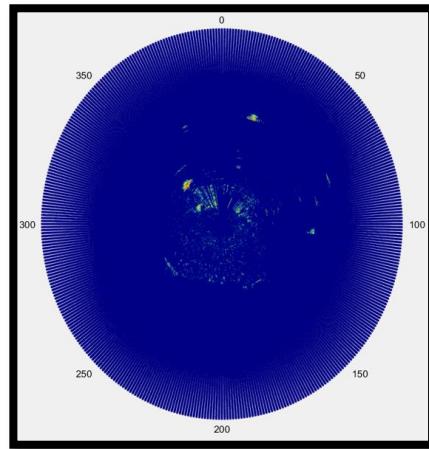
- Mediciones: se obtiene con ayuda de la información geométrica (definida por el usuario respecto del sistema de referencia local) de la zona de interés para realizar la representación del entorno. Han sido creadas funciones para crear esta matriz perímetro de forma poligonal (definiendo los vértices del mismo) o de forma circular (definiendo el centro y radio), por lo que para crear esta matriz es necesario hacer uso de los datos relacionados con la pose del ROV y las transformaciones explicadas en el anterior informe, para colocar los vértices o el centro respecto de la posición del vehículo.



- Reconocimiento: con el ROV parado y en el mismo lugar donde va a realizar la identificación, realiza una primera vuelta completa, determinando qué datos hacen referencia a los límites del entorno y creando la matriz a partir de esta información.



Una vez obtenida dicha matriz, ésta se integra con la matriz procesada para obtener un resultado donde queden únicamente los conjuntos de datos que se encuentren dentro de la piscina:



3.4. MÉTODO DE AGRUPAMIENTO (CLUSTERING)

Para obtener la información deseada de la cantidad y posicionamiento de los objetos a identificar, es necesario diferenciar los conjuntos que hacen referencia a dichos objetos del resto de componentes (ruido, paredes, suciedad, etc.) en la matriz de datos.

Por ello, ha sido llevado a cabo un estudio experimental de las características de los datos para cada una de los entornos donde las pruebas han sido realizadas, estudiando diferentes posibilidades para distinguir los componentes relevantes, obteniendo el mejor método para cada uno de ellos.

Las principales características obtenidas en la identificación son: el tamaño del componente, la distancia a la que se encuentra el centroide con respecto a la posición del ROV y la distribución e intensidad de sus datos.

En primer lugar, es necesario aclarar una serie de variables de las que dependerá el algoritmo. Tanto el procesamiento y filtrado como la detección del sistema completo depende de una serie de características que son:

- Límite intensidad agua: por debajo de este valor de intensidad será considerado intensidad nula y, por tanto, agua.
- Umbral de conectividad: cantidad mínima de datos para ser considerado componente.
- Conectividad: la conexión de los datos puede considerarse de dos formas en una matriz de datos de dos dimensiones.
- Distancia ruido: distancia que alcanza el ruido del SONAR.
- Límite cantidad de datos superior: límite máximo de la cantidad de datos para que sea considerado componente.

El proceso de identificación, una vez realizado el primer procesamiento, comienza binarizando la matriz obtenida sin establecer ningún orden de componentes y diferenciando estos mismos

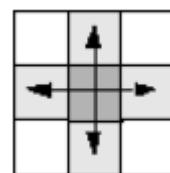
dependiendo de su tamaño (umbral de conectividad). De esta forma, el filtro elimina los componentes con un área menor que el parámetro establecido. A continuación, se muestra un ejemplo de cómo sería este proceso:

202	194	189	0	0	0	0
189	185	179	0	109	108	0
185	171	171	0	112	120	0
175	168	161	0	0	134	0
164	160	143	0	0	150	0
140	133	128	0	0	147	0
127	119	120	0	0	140	130
112	106	101	0	0	0	0

1	1	1	0	0	0	0
1	1	1	0	1	1	0
1	1	1	0	1	1	0
1	1	1	0	0	0	1
1	1	1	0	0	0	1
1	1	1	0	0	1	0
1	1	1	0	0	0	0

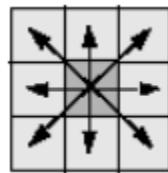
El etiquetado y enumeración de los componentes se realiza sobre una matriz binaria, especificando una conectividad, y se devuelve una matriz donde cada componente está identificado por un número (etiqueta) diferente. La conectividad de los datos se trata de una característica muy relevante, pues de esta variable dependerá directamente la cantidad de datos de los componentes. Esta conectividad para matrices de dos dimensiones puede tener dos valores:

- Conectado en 4: los datos se encuentran conectados porque sus lados están en contacto. Dos datos forman parte del mismo componente si ambos están conectados de forma vertical u horizontal.



1	1	1	0	0	0	0
1	1	1	0	2	2	0
1	1	1	0	2	2	0
1	1	1	0	0	0	3
1	1	1	0	0	0	3
1	1	1	0	0	0	3
1	1	1	0	0	3	3
1	1	1	0	0	0	0

- Conectado en 8: los datos se encuentran conectados porque sus lados o vértices están en contacto. Dos datos forman parte del mismo componente si ambos están conectados de forma vertical, horizontal o diagonal.



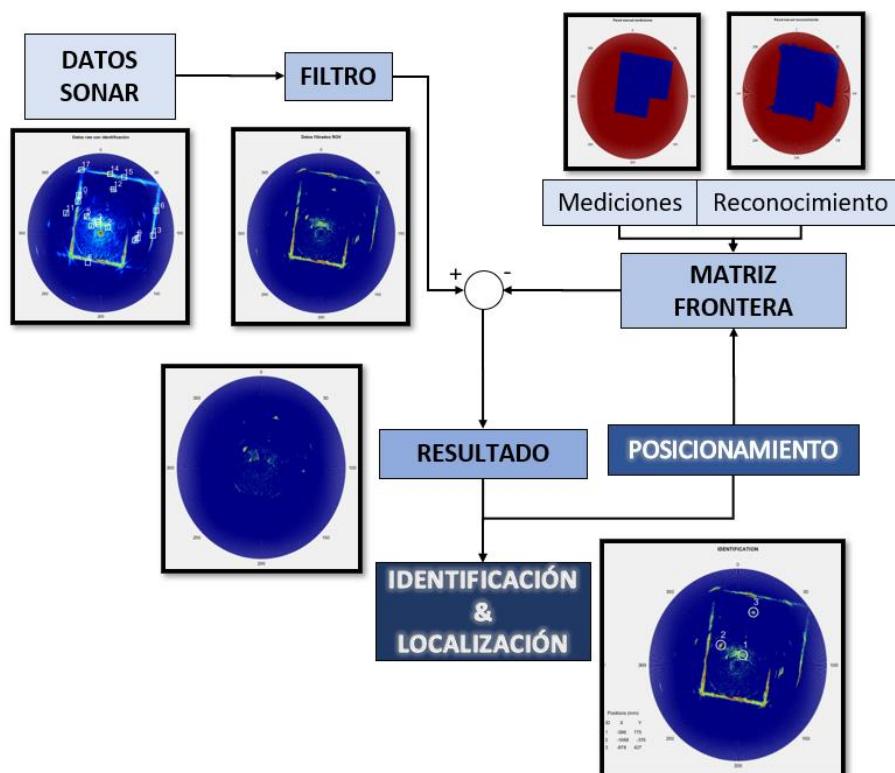
1	1	1	0	0	0	0	0
1	1	1	0	2	2	0	0
1	1	1	0	2	2	0	0
1	1	1	0	0	0	2	0
1	1	1	0	0	0	2	0
1	1	1	0	0	0	2	0
1	1	1	0	0	2	2	0
1	1	1	0	0	0	0	0

Una vez realizado este etiquetado, se puede conocer fácilmente la cantidad de componentes de la matriz y las celdas que corresponden a cada componente. Además, a través de otros comandos, es posible conocer muchas propiedades de cada conjunto de datos, donde serán necesarios el centroide y la cantidad de datos.

El centroide se especifica como una posición de la matriz en fila y columna, por lo que tenemos que pasar este valor a X-Y, formato con el que trabajamos. Sabiendo que con las filas obtenemos el ángulo y con las columnas sabemos la distancia (son 1200 datos equidistantes y especificamos al inicio la distancia de muestreo), se implementa una transformación de coordenadas polares a cartesianas.

Sabiendo la posición y tamaño de todos los componentes, podemos diferenciar los objetos relevantes. Para ello, el ruido siempre tiene el centroide más cercano al origen y tiene una gran cantidad de datos, detectándolo fácilmente y eliminándolo de la matriz final.

A través del siguiente diagrama podemos conocer como está relacionada cada matriz con el orden en el que se realizan los procesos del método propuesto: procesamiento, filtración y detección de objetos relevantes.

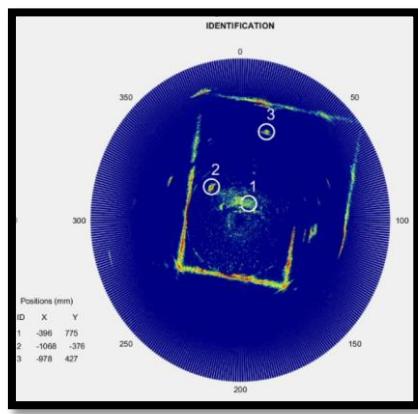


3.5. RESULTADOS Y CONCLUSIONES

Dentro de los resultados de este apartado hay que diferenciar entre los que se han desarrollado en el laboratorio, con datos almacenados de algunas pruebas, y los resultados obtenidos en pruebas en tiempo real.

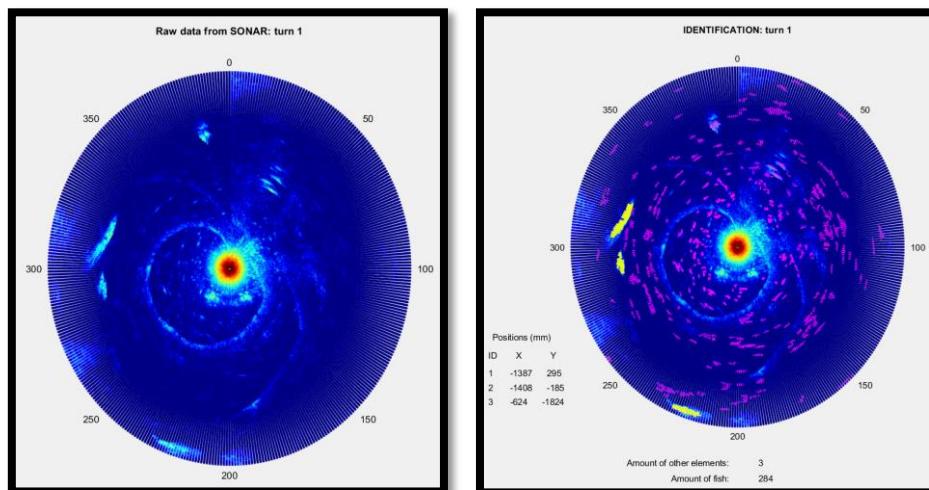
Identificación a datos almacenados

En primer lugar, se comenzó desarrollando el algoritmo de detección con pruebas realizadas en la piscina del Club de Golf de Bellavista, con personas y objetos colocados a conciencia en lugares medidos para compararlos con los valores obtenidos por SONAR. Los resultados obtenidos en este ámbito se muestran a continuación, donde las personas y el objeto son detectados y localizados en el sistema de referencia local:



Tras la toma de datos en el estero de Salinas del Astur, Punta Umbría, tuvimos que realizar un estudio mucho más minucioso sobre las características de los peces, aunque gracias a las grandes extensiones de agua no fue necesario el uso de matriz frontera.

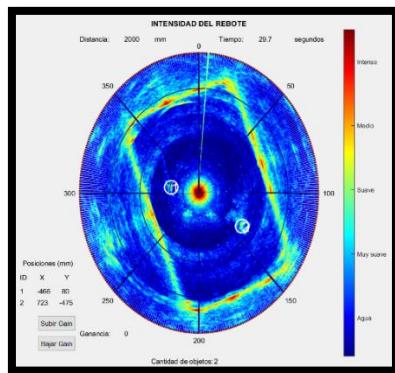
El resultado del conjunto de datos que se muestra ha detectado 284 peces (pintados en magenta) y 3 objetos (amarillo) que no se identifican con las propiedades de los componentes que representan a peces. Todos los conjuntos de datos están localizados, aunque solo se ha decidido mostrar en la parte inferior izquierda las posiciones de los objetos.



Gracias al uso de un SONAR de imagen multihaz se pudieron capturar imágenes a partir de las cuales se pudo contar, visualmente y uno por uno, una cantidad media entorno a los 100 peces en un sector de 130°. Por lo que, al estar realizando el muestreo con el sonar MSIS a lo largo de 360°, el resultado obtenido por el procedimiento propuesto es bastante coherente.

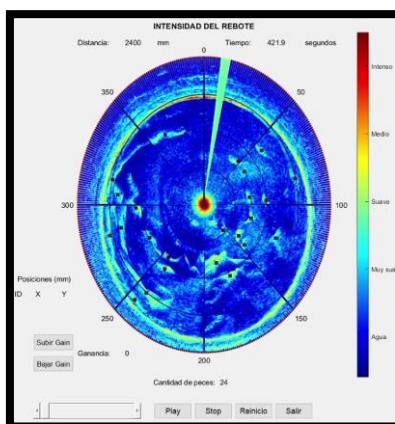
Identificación a tiempo real

Para la identificación a tiempo real, fue necesario optimizar la eficiencia del algoritmo de forma que la ejecución de la detección fuese lo más rápida posible, de manera que no se viese afectada la velocidad de muestreo del Ping360. Debido a esto, las primeras pruebas fueron realizadas en el patio de la ETSI, corrigiendo fallos y depurando el código hasta conseguir un resultado como el que se muestra en la siguiente figura:



En esta prueba, se colocó un objeto hundido en la piscina y otro en la superficie, detectando ambos perfectamente, llevando a cabo el algoritmo de detección cada 45°, lo cual no provocaba ningún retardo apreciable.

La última prueba consistió en realizar la detección a tiempo real en un entorno con peces. Con la ayuda de las instalaciones del IFAPA, pudimos muestrear datos en una primera prueba que nos sirvió para conocer las características del entorno y los peces. Una vez calibrado el algoritmo para las piscinas donde íbamos a llevar a cabo las pruebas de detección en tiempo real, volvimos a visitar las instalaciones y obtuvimos el resultado que se muestra en la siguiente figura:



Cabe destacar que en esta piscina había una cantidad aproximada de 25 peces, por lo que los resultados son totalmente satisfactorios, ya que el conteo se obtuvo un resultado de 24 peces.

Para el futuro, se ha comenzado a desarrollar un sistema de posicionamiento a través de los datos del SONAR, lo que nos permitiría poder realizar pruebas de detección dinámicas, así como movimientos de trayectorias predefinidas o de acercamiento a objetos, para lo que sería necesario trabajar paralelamente con el algoritmo de control.

4. BIBLIOGRAFÍA

- [1] Peter Corke Robotics, Vision and Control 123 FUNDAMENTAL ALGORITHMS IN MATLAB
- [2] Underwater Scan Matching using a Mechanical Scanned Imaging Sonar
- [3] Automatic Object Detection for AUV Navigation using Imaging Sonar within Confined Environments
- [4] Detecting small moving underwater objects using scanning sonar in waterside surveillance and complex security solutions
- [5] MSISpIC: A Probabilistic Scan Matching Algorithm Using a Mechanical Scanned Imaging Sonar
- [6] Multiple Obstacles Detection using Fuzzy Interface System for AUV Navigation in Natural Water
- [7] RBPF-MSIS: Toward Rao-Blackwellized Particle Filter SLAM for Autonomous Underwater Vehicle With Slow Mechanical Scanning Imaging Sonar
- [8] Underwater Loop-Closure Detection for Mechanical Scanning Imaging Sonar by Filtering the Similarity Matrix With Probability Hypothesis Density Filter
- [9] Fish abundance estimation with image sonar in semi-intensive aquaculture ponds
- [10] Estimación de la biomasa en acuicultura utilizando redes neuronales convolucionales
- [11] Localizando elementos de interés mediante vehículos operados remotamente para la explotación sostenible de granjas acuícolas.
- [12] Towards a multimodal interface for the specification of intervention tasks in underwater robotics



Interreg

España - Portugal

Fondo Europeo de Desarrollo Regional
Fundo Europeu de Desenvolvimento Regional

