

Universidad
Politécnica
de Cartagena



industriales
etsii UPCT

Modelado, Simulación y Control de un Vehículo Submarino Manipulado de forma Remota (ROV)

Titulación: Ingeniería Industrial
Intensificación: Sistemas eléctricos
Alumno/a: Javier de la Red Calvo
Director/a/s: José Luis Aguirre Martínez

Cartagena, 18 de Noviembre de 2015

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



Proyecto Fin de Carrera

Modelado, Simulación y Control de un Vehículo
Submarino Manipulado de forma Remota (ROV)



TITULACIÓN: Ingeniería Industrial

AUTOR: Javier de la Red Calvo

DIRECTOR: José Luis Aguirre Martínez

Diciembre de 2014

Índice

1. Introducción.....	1
1.1. Antecedentes.....	1
1.2. Objetivos.....	1
1.3. Herramientas utilizadas	2
2. Vehículos Marinos.....	3
2.1. Definición	3
2.2. Clasificación de los Vehículos Marinos	3
2.2.1. Vehículos Marinos sobre el agua	3
2.2.2. Vehículos submarinos	4
2.2.3. ROVs	4
2.2.4. ROVs y AUVs Comerciales	5
2.2.5. Aplicaciones de los robots submarinos	9
2.3. Particularidades de los vehículos marinos.....	9
2.3.1. Flotación	9
2.3.2. Amortiguación hidrodinámica.....	11
2.3.3. Masa añadida	12
2.3.4. Propulsión.....	12
3. Diseño.....	13
3.1. Modelos	13
3.1.1. Modelo 1	14
3.1.2. Modelo 2.....	14
3.1.3. Modelo 3.....	16
3.1.4. Modelo 4.....	18
3.2. Posición de las hélices	20
3.3. Pesos auxiliares	22
4. Modelo matemático	24
4.1. Sistemas de referencia	24
4.2. Ecuaciones dinámicas.....	25
4.2.1. Matriz de masas	25
4.2.3. Matriz de Coriolis.....	29
4.2.3. Matriz de resistencia hidrodinámica.....	31

4.2.4. Vector de flotación	32
4.2.5. Vector de fuerzas de propulsión	34
4.3. Resolución de las ecuaciones	43
5. Control.....	46
5.1. Control manual	46
5.2. Control de trayectoria	47
6. Simulación.....	51
6.1. Entrada de datos.....	51
6.2. Cálculos	52
6.3. Almacenamiento de datos.....	53
6.4. Visualización	53
6.5. Proceso de iteración.....	55
7. Resultados.....	57
7.1. Ensayo 1: Control manual	57
7.1.1. Trayectoria.....	58
7.1.2. Orientación	59
7.1.3. Velocidades	60
7.1.4. Fuerzas.....	61
7.2. Ensayo 2: Control por trayectoria.....	62
7.2.1. Trayectoria.....	63
7.1.2. Orientación	65
7.1.3. Velocidades	65
7.3. Ensayo 3: Movimiento con orientación.....	67
7.2.1. Trayectoria.....	68
7.1.2. Orientación	70
7.1.3. Velocidades	71
8. Conclusiones.....	74
8.1. Trabajo realizado	74
8.2. Mejoras y trabajos futuros	75
A. Anexo 1: Instalación de bibliotecas e inclusión de directorios en C++.....	77
Eigen.....	77
OpenSceneGraph.....	78
Archivos OBJ	82

Referencias 84

1. Introducción

Los ROV (Remotely Operated Vehicle) son vehículos controlados de forma remota por un humano. Su relevancia ha crecido durante la última década debido al aumento de su uso militar (drones), por otra parte, también han surgido grandes innovaciones que permiten su uso en el campo científico y empresarial.

El caso de estudio en este proyecto es el de un ROV submarino, o ROUV (Remotely Operated Underwater Vehicle). Este tipo de vehículo destaca por su alta maniobrabilidad en aguas profundas y su uso está muy extendido en la exploración y en el mantenimiento de estructuras submarinas.

1.1. Antecedentes

El proyecto que nos ocupa, se inició a partir de los estudios realizados sobre AUVs (Vehículos Autónomos Submarinos) en un proyecto de estudio del grupo de investigación “Diseño, mantenimiento y disponibilidad de sistemas mecánicos” en colaboración con la empresa Navantia.

Durante los últimos años se han modelado en el Departamento de Ingeniería Mecánica algunos vehículos autónomos, sin embargo, ahora se pretende construir un prototipo con el que se pueda comprobar la validez y cercanía de estos modelos con la realidad. Con esta aspiración, se decidió crear un vehículo submarino, de diseño sencillo, capaz de realizar maniobras complejas, movimientos y giros en las tres dimensiones espaciales, y con cabida para modificaciones, tanto del sistema de control, como de la configuración mecánica. La realización de una simulación previa del comportamiento del prototipo nos permitirá comprobar su comportamiento ante diferentes configuraciones de diseño, y así poder construir el que se considere óptimo.

1.2. Objetivos

La meta de este proyecto consiste en realizar el diseño conceptual y el modelo matemático de un ROV submarino. También se pretende programar el control manual, interrelacionando la acción de los propulsores con los movimientos del vehículo. Se diseñará e implementará un sistema de seguimiento autónomo de trayectorias y se programará una salida para visualizar el comportamiento del ROV en un contexto operacional simulado.

Para conseguir estos objetivos ha sido necesario seguir una serie de pasos que defino a continuación y se verán con más detalle en capítulos posteriores:

- Evaluación de distintos modelos físicos y la elección del más conveniente, así como su diseño en una herramienta de software de modelado 3D.

- Estudio de las ecuaciones cinemáticas y dinámicas del sólido rígido, así como las fuerzas hidrodinámicas que sufre el vehículo en función de su forma y su velocidad y aceleración en cada instante.
- Implementación de este modelo matemático en el programa MATLAB incluyendo un control manual de los propulsores para la comprobación de las ecuaciones anteriores.
- Transformación del código a un lenguaje de programación más adecuado para la representación visual e introducción del control manual por trayectoria.

1.3. Herramientas utilizadas

Para la realización de este proyecto se han utilizado varias herramientas informáticas, cuyo uso ha sido fundamental para alcanzar los objetivos fijados. A continuación se indican cuáles son estas herramientas, la razón de su elección y el uso que se ha dado de las mismas.

- **Rhinoceros 5:** Esta herramienta de software permite modelar en tres dimensiones y se usa fundamentalmente para el diseño industrial y naval. Se ha elegido este programa frente a otras opciones de modelado 3D debido a su facilidad para importar y exportar archivos en una gran variedad de formatos, en concreto, en nuestro caso nos ha permitido exportar los modelos en formato OBJ, compatible con la biblioteca OSG en C++. Con este programa hemos realizado principalmente los diferentes modelos del cuerpo del ROV, de los cuales hemos podido obtener propiedades básicas, momentos de inercia, centro de gravedad, volumen, etc. También se han realizados modelos de las hélices y de otros objetos introducidos en la simulación de C++ (flechas, terreno, rocas...). Para facilitar la modificación del modelo del cuerpo del ROV, en caso de que sea necesario, también se han creado en esta herramienta dos programas en RhinoScript en los que se crea el modelo a partir de dimensiones modificables en el código.
- **Matlab 2013:** Este programa ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M), especialmente indicado para el cálculo y manipulación de matrices. Esta capacidad para realizar operaciones con matrices es la razón fundamental por la que utilizamos este programa para comprobar inicialmente el modelo matemático, ya que este se basa en una ecuación matricial. En este software hemos implantado las ecuaciones del modelo matemático y hemos comprobado que estas producen resultados acordes a los que cabe esperar del comportamiento de un ROV. Una vez comprobadas las ecuaciones se intentó realizar una simulación visual del comportamiento del vehículo, sin embargo, las limitaciones del lenguaje de programación impiden conseguir un resultado adecuado.

- **C++:** El lenguaje de programación C++ tiene la característica de poder ser usado en una gran variedad de aplicaciones. Aunque está orientado a la programación de sistemas, se puede utilizar en muchos otros contextos. Para suplir las deficiencias de la programación en MATLAB, se decidió utilizar este programa para crear la simulación visual del ROV, ya que contamos con la biblioteca OpenSceneGraph (OSG), especializada en este tipo de aplicaciones. Se traspasó a C++ el código ya creado en MATLAB y se añadieron nuevas propiedades al código, como el control de trayectoria.

Referencias: [1], [2], [3]

2. Vehículos Marinos

2.1. Definición

Un vehículo marino es todo aquel capaz de navegar sobre o bajo el agua, están diseñados para el desplazamiento por las diferentes masas de agua que podemos encontrar en nuestro planeta, ríos, lagos y mares, y su uso se puede remontar hasta la época prehistórica en asentamientos de Indonesia y Australia.

Los vehículos marinos han tenido una gran importancia histórica por el descubrimiento de nuevos territorios y el comercio entre países. En la actualidad siguen siendo de gran importancia, para el transporte transcontinental y para la realización de actividades marinas como la pesca. Por otro lado, se abren nuevas posibilidades gracias al desarrollo de los vehículos autónomos, ideales para la exploración, que son capaces de realizar mapas del lecho marino para la instalación y mantenimiento de estructuras submarinas, así como el estudio científico de la composición del agua a diferentes profundidades.

Referencias: [4]

2.2. Clasificación de los Vehículos Marinos

2.2.1. Vehículos Marinos sobre el agua

Los vehículos marinos superficiales son aquellos que se mueven siempre sobre la superficie de agua, es decir, en ningún momento se encuentran totalmente sumergidos. Existe una gran cantidad de vehículos de este tipo, con variados diseños dependientes de su función.

Los vehículos marinos a los que más estamos acostumbrados son barcos y naves, pero también existen otros menos convencionales como las motos de agua o hidroaviones. Su diseño depende de la función que vayan a realizar, normalmente la cantidad de personas o carga que vaya a contener. Para el transporte de grandes cargas a largas

distancias se crean grandes y alargadas naves como los transatlánticos, para el transporte de personas y pequeñas cargas, por el contrario, se usan barcos más pequeños.

2.2.2. Vehículos submarinos

Los vehículos submarinos que pueden moverse, o se mueven exclusivamente, por debajo del agua, es decir, se sumergen por completo dentro del agua. El ejemplo más común de este tipo de vehículos es el submarino, es tan prominente que da nombre al conjunto.

Los submarinos son vehículos de gran tamaño capaces de sumergirse a grandes profundidades, gracias a un casco capaz de aguantar la presión, durante largos periodos de tiempo. La potencia para la propulsión se obtiene en grandes submarinos con energía nuclear, aunque submarinos más pequeños utilizan una propulsión diésel-eléctrica. Los submarinos son utilizados sobre todo en actividades militares, tomaron parte en las dos guerras mundiales, se utilizaron para cruzar áreas de mar controladas gracias a su difícil detección por parte del enemigo.

Pero, aparte de los submarinos convencionales tripulados, también existen otros vehículos que son capaces de sumergirse y realizar actividades bajo el agua, estos son los vehículos submarinos autónomos (AUV), y los vehículos submarinos operados de forma remota (ROV). Estos vehículos son muy diferentes a los submarinos comunes ya que, al no necesitar ser controlados por un operario dentro del vehículo, se diseñan con un tamaño mucho menor, lo que permite conseguir una gran maniobrabilidad. Este tipo de vehículos son útiles en la exploración del fondo marino y el mantenimiento de estructuras submarinas, como las dedicadas a la extracción de hidrocarburos.

Referencias: [5]

2.2.3. ROVs

El proyecto que nos ocupa se centra en un vehículo submarino operado de forma remota, ROUV o más comúnmente ROV, así que creo necesario explicar más detalladamente las características de este tipo de vehículos.

El acrónimo ROV representa la expresión inglesa Remotely Operated Vehicle, que en español se puede traducir como vehículo operado de forma remota, aunque normalmente cuando se habla de ROVs nos estamos refiriendo a vehículos que trabajan bajo el agua, así, aunque el termino correcto sería ROUV, Remotely Operated Underwater Vehicle, generalmente se usa simplemente ROV.

Los ROVs son vehículos submarinos muy comunes en las industrias que trabajan con estructuras submarinas. Su característica principal es estar controlados por un operario fuera del vehículo, en un barco o en tierra, lo que le concede gran maniobrabilidad.

Un cable con flotabilidad nula está unido físicamente al ROV para transmitir la información que manda el operario. Este cable contiene varias líneas por las que se trasfiere la potencia eléctrica necesaria para propulsar al ROV y para hacer funcionar los diferentes componentes, se envían las señales de datos por parte del operario y se reciben las señales de video que obtiene el ROV en caso de tener una o varias cámaras incorporadas.

Dentro del casco del ROV se encuentran varios componentes que permiten medir con gran exactitud la posición y velocidad en la que se encuentra el vehículo, estos pueden ser: magnetómetros, sensores de presión, GPS, sistemas de navegación inercial.... También existen otros instrumentos de medida que pueden ser útiles, como sonar, medidores de temperatura, penetración de luz y claridad del agua.

La mayoría de ROVs llevan incorporado una cámara de video y linternas para poder recibir imágenes del entorno. Además, para interactuar con el medio, algunos ROVs están equipados con brazos que permiten realizar operaciones bajo el agua. La complejidad de los ROVs puede variar enormemente dependiendo de las tareas que se quieran realizar con ellos.

Dependiendo de la capacidad de operación que tenga el vehículo, los ROVs se pueden clasificar en tres subconjuntos:

- 1.- ROVs de observación: Son capaces de trasladarse hasta un lugar concreto bajo el agua y tomar imágenes de su entorno. Se pueden utilizar para inspeccionar e identificar problemas en estructuras submarinas.
- 2.- ROVs de observación con capacidad de carga: Pueden transportar pequeñas cargas, como herramientas.
- 3.- ROVs de trabajo: Se usan en el mantenimiento y reparación de estructuras submarinas.

Los avances tecnológicos en el área de la electrónica han acelerado el desarrollo de la industria de los ROVs con lo que a día de hoy existen un gran número de modelos de ROV diferentes, capaces de realizar muy diversas tareas. La mayoría de los ROVs actuales están dedicados a las industrias del petróleo y del gas, donde son necesarios para la construcción, mantenimiento y reparación de líneas de tuberías, sin embargo, se está extendiendo su uso a otras aplicaciones relacionadas con la investigación, la arqueología y el uso militar.

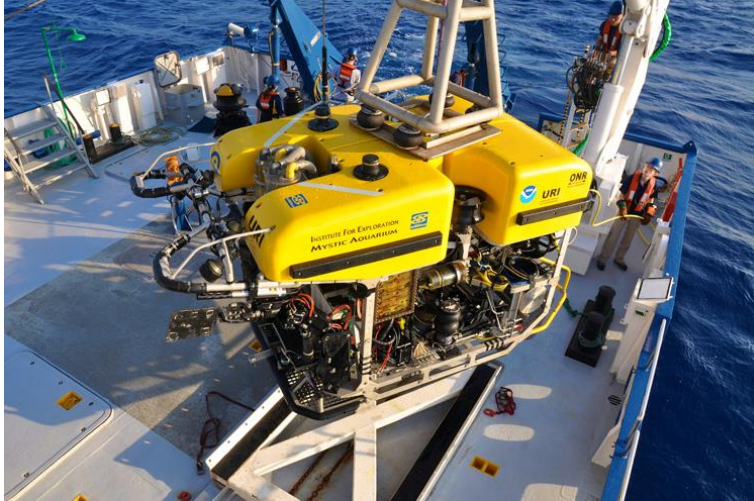
Referencias: [6], [7], [8], [9]

2.2.4. ROVs y AUVs Comerciales

El hecho de que cada uno de los vehículos submarinos que se construyen en la actualidad, se diseñan con un propósito concreto, comporta la existencia de una gran variedad de modelos. La tarea para la que esté diseñado el vehículo determinará la carga útil, el número y potencia de los motores de las hélices y las herramientas que portan.

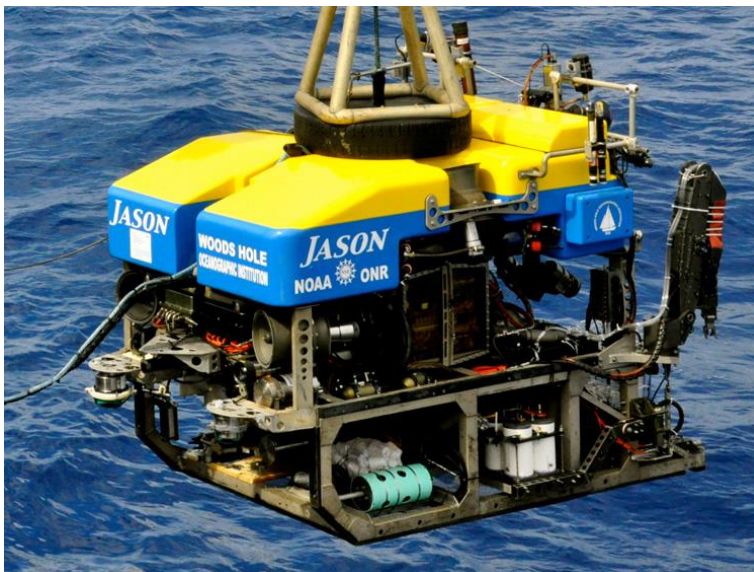
A continuación se presentan seis diferentes modelos de vehículos submarinos, con sus características principales.

○ **HERCULES**



- **Tipo:** ROV
- **Sensores:** Cámaras, sensor de sonido, cámara de alta definición, sensor de presión, sensor de temperatura, medidor de concentración de oxígeno y medidor de salinidad.
- **Herramientas:** Dos brazos articulados para la manipulación de objetos.
- **Utilidades:** Arqueología, biología y geología.
- **Número de propulsores:** Seis.
- **Máxima profundidad:** Cuatro kilómetros.

○ **JASON**



- **Tipo:** ROV

- **Sensores:** Cámara en blanco y negro, cámara de alta definición y sensor de temperatura.
- **Herramientas:** “Catadores” de agua, brazo manipulador para tomar muestras del terreno y rocas y bomba de succión para recolectar pequeños animales.
- **Utilidades:** Exploración e investigación científica.
- **Número de propulsores:** Seis.
- **Máxima profundidad:** Seis kilómetros y medio.

○ **ROPOS**



- **Tipo:** ROV
- **Sensores:** Dos cámaras de video digitales, sonar y sistema de telemetría
- **Herramientas:** Dos brazos manipuladores, bomba de succión y cortadora.
- **Utilidades:** Exploración.
- **Velocidad máxima:** 1.3 m/s.
- **Máxima profundidad:** Cinco kilómetros.

○ **HUGIN**



- **Tipo:** AUV
- **Sensores:** Cámara fija, sonar y sensor de turbulencia, sensor de metano, compás y magnetómetro.
- **Utilidades:** Biología, geología y defensa.
- **Velocidad máxima:** 2 m/s.
- **Número de propulsores:** Uno

- **Máxima profundidad:** Cinco kilómetros.

- **PLUTO PLUS**



- **Tipo:** AUV
- **Sensores:** Sonar, compás y magnetómetro.
- **Utilidades:** Localización y destrucción de minas submarinas.
- **Velocidad máxima:** 1.5 m/s.
- **Número de propulsores:** Tres.
- **Máxima profundidad:** Cinco kilómetros.

- **SEAOTTER**



- **Tipo:** ROV
- **Sensores:** Sonar, GPS, compás, sensor de presión y sensor de temperatura.
- **Herramientas:** Brazo con garra.
- **Utilidades:** Inspección de construcciones sumergidas
- **Número de propulsores:** Cuatro.
- **Máxima profundidad:** Quinientos metros.

Referencias: [10], [11], [12], [13]

2.2.5. Aplicaciones de los robots submarinos

En el pasado reciente la mayoría de los robots submarinos se desarrollaron con el objetivo de realizar trabajos relacionados con dos aplicaciones, el tendido de cables para comunicaciones y las plataformas oceánicas de extracción de petróleo y gas. En la actualidad, sin embargo, existen otras aplicaciones que aprovechan las particulares características de los robots submarinos.

Los usos más habituales en la actualidad de los robots submarinos son los siguientes:

- Investigación oceánica en las ramas de la biología y la geología
- Arqueología.
- Aplicación militar, centrada en la vigilancia, localización y neutralización o recuperación de minas y armamento en el fondo marino.
- Localización y rescate de barcos y aviones hundidos
- Inspección y reparación de buques
- Inspección y reparación de construcciones con estructura sumergida (presas puentes, etc.)

Referencias: [14], [15]

2.3. Particularidades de los vehículos marinos

Todos los vehículos marinos comparten ciertas particularidades que los diferencian de los vehículos terrestres, debidas obviamente al diferente medio en el que trabajan. A continuación voy a desarrollar las diferencias producidas por el medio que considero más importantes: flotación, amortiguación hidrodinámica, masa añadida y propulsión, aunque hay algunas más, como la presión, en especial para vehículos submarinos y las perturbaciones ambientales (oleaje, viento, corrientes...).

2.3.1. Flotación

El hecho de que los vehículos marinos estén parcial o totalmente sumergidos en una masa de agua produce una fuerza ascendente que se debe equilibrar con el peso del vehículo para mantener su posición en el eje vertical. A su vez, para favorecer la

estabilidad, y para que no se creen momentos no deseados, el centro de gravedad debe situarse en la vertical con el centro de carena (centro geométrico de la parte sumergida del vehículo) y por debajo de este. Esta es la razón por la que los vehículos marinos se

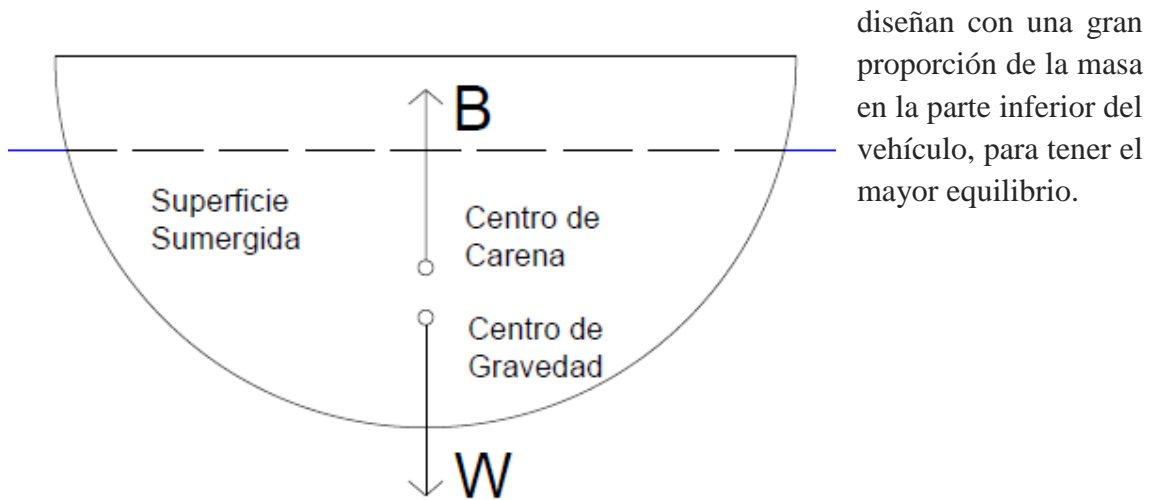


FIGURA 1: FUERZAS DE FLOTACIÓN

En la figura 1 se pueden distinguir las fuerzas de flotación (B) y la fuerza peso (W) alineadas según la vertical, por lo tanto sin crear momento. Además, al estar el centro de gravedad por debajo del centro de carena, pequeñas variaciones en la inclinación del vehículo producirán un momento corrector que tiende el vehículo a su posición inicial. La magnitud de ese momento corrector solo depende de la suma de las fuerzas de gravedad y flotación (B+W) y la distancia entre ellas en el plano horizontal. La ecuación que define este momento es la siguiente:

$$RM = \frac{1}{2} \cdot d \cdot (B + W)$$

Si, por el contrario, el centro de gravedad se encontrara por encima del centro de carena, el cuerpo se encontraría en un equilibrio inestable, cualquier pequeña perturbación provocaría un momento en la dirección opuesta a la de equilibrio, este efecto se retroalimentaría positivamente, hasta el momento del vuelco del vehículo, cuando se alcanzaría una nueva posición de equilibrio, ahora sí, con el centro de gravedad por debajo del centro de carena.

La fuerza de flotación se calcula aplicando el principio de Arquímedes: “Un cuerpo total o parcialmente sumergido en un fluido en reposo, recibe un empuje de abajo hacia arriba igual al peso del volumen del fluido que desaloja”. Esto quiere decir que si, por ejemplo, nos encontramos con un vehículo parcialmente sumergido en una masa de

agua, el empuje recibido por este cuerpo será en sentido ascendente y con una magnitud igual al volumen sumergido por la densidad del agua que lo rodea. Expresado matemáticamente:

$$B = -\rho_{agua} \cdot V_{sum}$$

El símbolo negativo es debido a la orientación de los ejes de referencia, el eje z tiene el sentido orientado según la fuerza de la gravedad, hacia abajo, por lo tanto, la fuerza de flotación, de sentido ascendente, tiene signo negativo.

Los submarinos son capaces de variar su flotabilidad para realizar las maniobras de sumersión y emersión. Mediante el uso de cámaras que pueden almacenar tanto agua como aire con tanques de aire comprimido. De esta forma se puede manipular el peso del submarino para que sea mayor o menor que la fuerza de flotación.

Para vehículos submarinos más pequeños, el movimiento según el eje z se suele realizar con la ayuda de propulsores, y su flotabilidad es ligeramente positiva ($B > W$) para que en caso de fallo el vehículo flote a la superficie.

Referencias: [16], [17]

2.3.2. Amortiguación hidrodinámica

La amortiguación hidrodinámica se puede definir como toda fuerza contraria al movimiento que se produce por efecto del movimiento del vehículo en el agua. Hay dos fenómenos que pueden producir amortiguación hidrodinámica, la fuerza de arrastre y la fricción superficial. Ambas fuerzas son proporcionales a la velocidad, de forma lineal en flujo laminar y de forma cuadrática en flujo turbulento.

En la mayoría de los casos tratamos con flujos turbulentos y una fricción superficial despreciable frente a la fuerza de arrastre. Para calcular esta fuerza podemos recurrir a la ecuación de arrastre:

$$F_D = \frac{1}{2} \cdot \rho_{agua} \cdot v^2 \cdot C_D \cdot A$$

A es el área del cuerpo sumergida en el fluido, perpendicular a la dirección de movimiento y C_D es el coeficiente de arrastre del cuerpo que dependerá de su forma.

Para minimizar esta fuerza en los vehículos marinos se intenta reducir los términos C_D , para ello se proporciona de una forma más pequeña y aerodinámica a la parte sumergida del vehículo que se enfrenta al flujo de agua. Por este principio se diseñan los barcos con formas alargadas y redondeadas en la parte sumergida, y con forma de torpedo los submarinos. Este no es el caso de los ROV, ya que en ellos se busca la maniobrabilidad en las tres direcciones espaciales y su velocidad no suele ser muy elevada por lo que su efecto, aunque importante, no condiciona la forma del vehículo.

Podemos encontrar el equivalente de esta fuerza para los vehículos terrestres en la amortiguación aerodinámica, aunque menor, debido a la baja densidad del aire, su efecto es importante a grandes velocidades, se tiene en cuenta para el diseño de automóviles, pero es especialmente llamativo en trenes de alta velocidad donde el diseño de la parte frontal es esencial para alcanzar una mayor velocidad punta. Por supuesto este efecto en vehículos aéreos, como aviones y naves espaciales, tiene gran importancia y condiciona su forma como consecuencia.

Referencias: [16], [18]

2.3.3. Masa añadida

El concepto de masa añadida es usualmente malentendido como una cantidad finita de fluido que arrastra el vehículo en su movimiento, formando el conjunto del vehículo y el fluido un nuevo sistema con una masa e inercia suma de las de los cuerpos anteriores. La masa añadida se debe entender como fuerzas y momentos inducidos por la presión debido al movimiento armónico inducido por el fluido que es proporcional a la velocidad del vehículo. Consecuentemente, las fuerzas y las aceleraciones de masa añadida se opondrán al movimiento al estar 180° fuera de fase.

Referencias: [16]

2.3.4. Propulsión

La propulsión de los vehículos marinos es muy variada, y ha ido cambiando a lo largo de la historia. Hasta que no se consiguió el perfeccionamiento del motor a vapor por James Watt, la propulsión de vehículos marinos se realizaba, o bien con trabajo humano, barcos a remos, o mediante el uso de las corrientes de viento, barcos a vela. La invención del barco a vapor supuso una revolución y dio salida a una carrera tecnológica para crear barcos más grandes y con motores más potentes.

La manera más utilizada de transformar la potencia de los motores en movimiento del vehículo es el uso de propulsores; estos son hélices que giran unitariamente al eje del motor y, debido a su perfil alar, crean una diferencia de presión entre la parte posterior y anterior, empujando el agua del entorno hacia atrás, por la conservación de la cantidad de movimiento el vehículo se mueve hacia delante.

Existen otros sistemas de propulsión menos utilizados, las ruedas de paletas se utilizaron desde la invención del barco de vapor, a finales del siglo XVIII, hasta principios del siglo XX. Otros sistemas de propulsión más modernos los ventiladores de aire, pero se utilizan únicamente para vehículos especiales, como en los aerodeslizadores o los hidrodslizadores.

Referencias: [19]

3. Diseño

Para el diseño del ROV se han valorado varios modelos distintos, y entre ellos se ha elegido el que se adapta mejor a nuestras necesidades, las cuales son: alta maniobrabilidad, pequeño tamaño, capacidad de carga de la instrumentación y fácil fabricación.

En este proyecto no se va a profundizar en el diseño de detalle, debido a la limitación del tiempo disponible para la realización del mismo. El diseño de detalle del ROV, así como el proceso de construcción del mismo, puede ser por lo tanto considerado línea futura de desarrollo. Además, gracias a la versatilidad de los programas desarrollados, se pueden introducir cambios sustanciales en los parámetros de diseño (materiales utilizados, distancia del centro de carena a los motores, tamaño del casco...).

Las bases iniciales del diseño fueron las de un ROV en forma de cubo, con seis brazos donde se colocarían las hélices que permitieran el desplazamiento y el giro en las tres dimensiones espaciales.

Toda la instrumentación se ubicará dentro del cubo, siendo este una caja estanca con la posibilidad de abrirse para acceder a dicha instrumentación. Los brazos estarán unidos a la caja, ya sea por tornillos, soldadura u otro tipo de unión, y debe ser posible la conexión del cuerpo del ROV con los motores a través de los brazos. Para la sujeción de los motores es posible que sea necesario además otro conjunto de brazos adicionales. También será necesaria la conexión del ROV con el exterior a través de un cable de flotabilidad nula.

En el diseño general solo contemplaremos el diseño de la caja y los brazos de las hélices, ya que son los que nos definen los parámetros que serán necesarios más tarde en el modelado del comportamiento del ROV (peso, volumen, distancia del centro de carena a los motores, perfil hidrodinámico...).

Se decidió utilizar un cubo como el cuerpo del ROV por su capacidad para generar un espacio donde colocar los componentes electrónicos, aunque esta forma tiene menor integridad estructural que otras formas simples como el cilindro o la esfera, esta solución permite crear un espacio accesible y resistente. Esta forma de cubo puede adquirirse con una cámara estanca que sea ligera, modular y resistente a las condiciones ambientales a las que va a estar sometida.

3.1. Modelos

A continuación expondré los tres modelos contemplados durante el proyecto y que fueron diseñados con el programa Rhinoceros.

3.1.1. Modelo 1

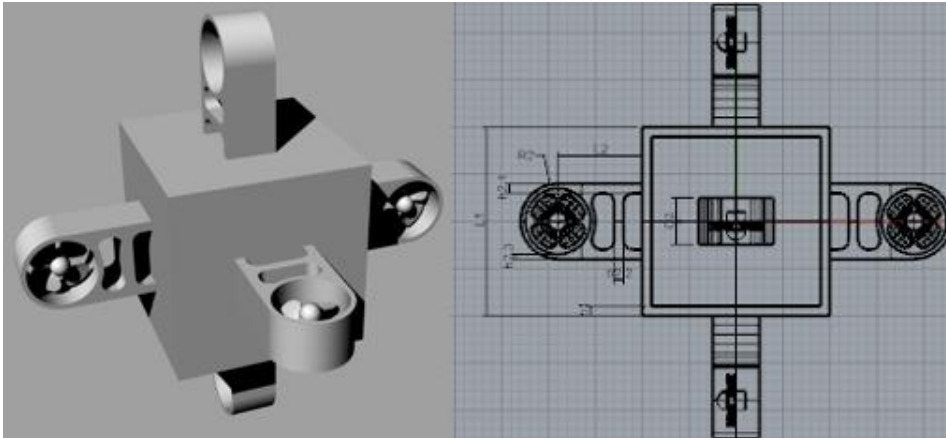


FIGURA 2: VISTA TRIDIMENSIONAL Y FRONTAL DEL MODELO 1

Inicialmente se propuso un modelo con una caja estanca de 200 mm de lado y 10 mm de grosor, para los brazos se eligió un diseño complejo con una estructura rígida y pesada, por ello, se decidió cambiar la forma de los brazos con un diseño más sencillo. Sin embargo, el resto de parámetros del modelo, el tamaño de la caja, la distancia entre ejes y la posición de las hélices se mantienen en los modelos posteriores.

3.1.2. Modelo 2

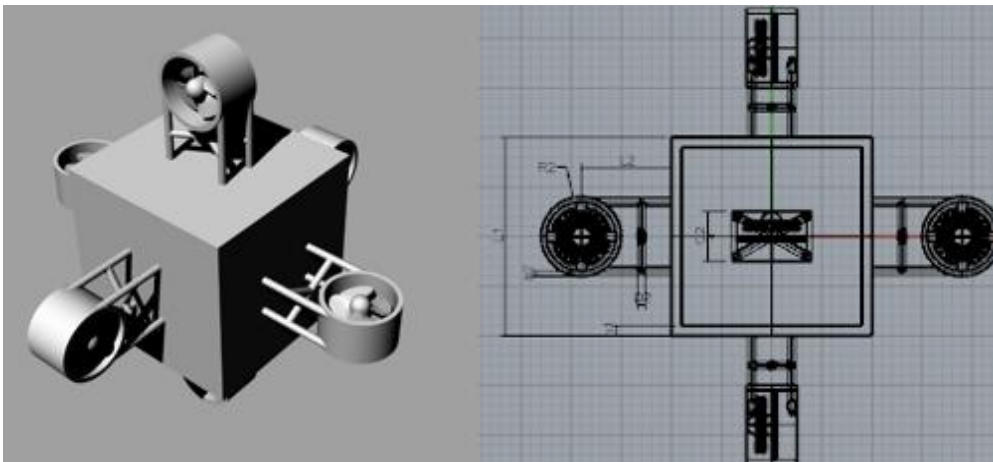


FIGURA 3: VISTA TRIDIMENSIONAL Y FRONTAL DEL MODELO 2

Para aligerar el peso de los brazos se propuso un diseño con varillas de aluminio, manteniendo la rigidez del conjunto y reduciendo el peso. En este caso nos encontramos con el problema de como introducir los cables que conectan los motores. Las varillas son demasiado finas para poder pasar un cable por dentro de ellas, y que el cable pase por el exterior es una no es una opción muy segura, aparte de ser antiestético. Por estas razones se decidió descartar este modelo.

Sin embargo, se decidió calcular los parámetros físicos de este modelo para comenzar con la programación. En la siguiente tabla se presentan los datos geométricos y físicos necesarios para el cálculo de los parámetros.

Lado Caja (mm)	200
Espesor Caja (mm)	10
Densidad Caja (kg/m³)	1900
Diámetro varillas (mm)	8
Diámetro Exterior Cilindro Hueco (mm)	80
Altura Cilindro Hueco (mm)	50
Espesor Cilindro Hueco (mm)	5
Densidad brazos (kg/m³)	2500
Masas auxiliares (kg)	2.5
Distancia Eje Hélice a Centro de Carena (mm)	190

TABLA 1: DATOS DEL MODELO 2

Las masas auxiliares que se refieren en la tabla corresponden a las masas tanto de los componentes electrónicos que va a llevar instalado el ROV, como pequeños pesos colocados dentro de la caja del ROV que cumplen varios propósitos que explicaremos más adelante.

Para calcular el empuje, un dato esencial, es necesario conocer la densidad del agua en el medio en el que va a trabajar el vehículo. Aunque normalmente la densidad del agua se toma como 1000 kg/m³, al trabajar en agua salada, que es donde se desarrollan casi por completo todas las actividades que requieren el uso de ROVs, la densidad utilizada para los cálculos ha sido de 1030 kg/m³, ligeramente superior a la normal.

Con los datos anteriores se consiguen los parámetros físicos del modelo. Estos están presentados en la siguiente tabla.

Masa (kg)	8.804
Empuje (N)	86.72
Peso (N)	86.36
Centro de Gravedad	
CDGx (mm)	0.09676
CDGy (mm)	0.5806
CDGz (mm)	23.30

Momentos de Inercia Principales	
I_x (kg · m²)	0.1075
I_y (kg · m²)	0.1075
I_z (kg · m²)	0.09144

TABLA 2: RESULTADOS DEL MODELO 2

Estos datos se han calculado a través de una aplicación de Rhinoceros llamada RhinoScript que permite dibujar y realizar cálculos de forma automatizada, esto significa que se pueden modificar fácilmente los parámetros físicos a partir de los datos geométricos.

En la tabla se puede observar que el empuje es ligeramente superior al peso, esto ejerce una fuerza hacia arriba de 0.36 N o 0.4% del peso. Esta semejanza entre las fuerzas no es casual, evidentemente se ha diseñado el modelo e introducido las masas auxiliares necesarias para que esto se cumpla.

Otro dato importante que se aprecia en esta tabla es el centro de gravedad, el cual se encuentra 23.3 mm por debajo del centro de carena, el centro geométrico. Esto se ha conseguido colocando los pesos añadidos en la cara inferior de la caja, a 80 mm del centro de carena. La posición del centro de gravedad según los ejes X e Y es muy pequeña, inferior a un milímetro, por lo que podemos considerar al centro de gravedad en la vertical del centro de carena, lo que mantiene al vehículo recto de forma natural.

Los momentos de inercia principales son muy similares debido al carácter simétrico del vehículo, la inercia según el eje Z es algo inferior debido a que los pesos añadidos se han colocado centrados en ese eje y por lo tanto no crean momento según esa componente. No se han incluido los productos de inercia cruzados (I_{xy} , I_{xz} , I_{yz}) ya que al ser la geometría prácticamente simétrica, estos son casi nulos, y no se han tenido en cuenta a la hora de hacer los cálculos.

3.1.3. Modelo 3

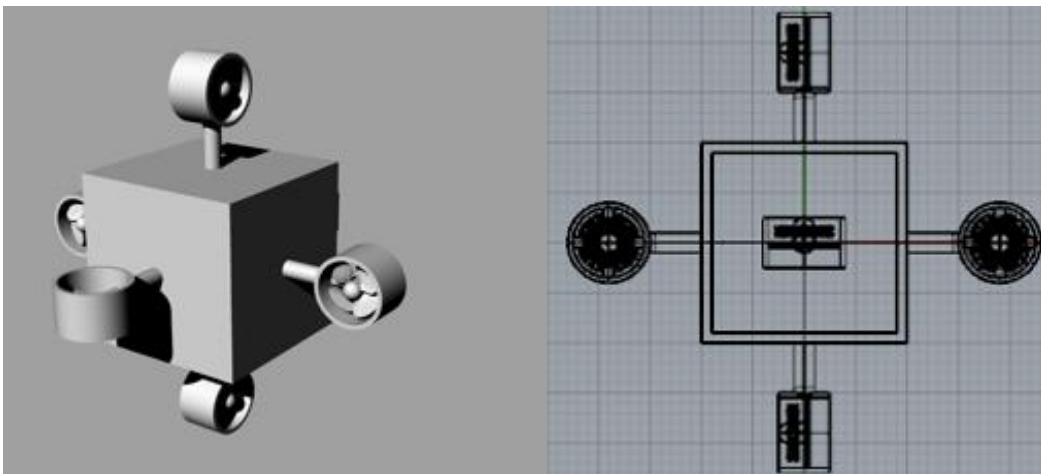


FIGURA 4: VISTA TRIDIMENSIONAL Y FRONTAL DEL MODELO 3

Para solucionar el problema de la conexión de los cables de los motores, se optó finalmente por un único cilindro hueco en cada brazo que conectara la caja con el difusor que cubre la hélice. De esta forma se reduce aún más el peso y la rigidez de los brazos, se llegó a la conclusión de que no era necesario proporcionar tanta rigidez a los brazos ya que no van a estar sometidos a grandes fuerzas. Otra ventaja de este modelo es que reduce la resistencia hidrodinámica, ya que el área del perfil perpendicular al flujo se reduce considerablemente.

Las dimensiones de este modelo son muy similares al anterior, pero en la siguiente tabla se pueden apreciar algunas diferencias.

Lado Caja (mm)	200
Espesor Caja (mm)	10
Densidad Caja (kg/m³)	1900
Diámetro varilla (mm)	20
Diámetro Exterior Cilindro Hueco (mm)	80
Altura Cilindro Hueco (mm)	50
Espesor Cilindro Hueco (mm)	5
Densidad brazos (kg/m³)	2500
Masas auxiliares (kg)	2.5
Distancia Eje Hélice a Centro de Carena (mm)	190

TABLA 3: DATOS DEL MODELO 3

Con estos datos, y a partir del programa de Rhinoceros se pueden calcular los parámetros fundamentales del modelo, utilizando como en el modelo anterior una densidad del agua de 1030 kg/m³.

Masa (kg)	8.715
Empuje (N)	86.37
Peso (N)	85.49
Centro de Gravedad	
CDGx (mm)	0.9775
CDGy (mm)	0.9774
CDGz (mm)	23.047
Momentos de Inercia Principales	
Ix (kg · m²)	0.1058
Iy (kg · m²)	0.1058

Iz (kg · m²)	0.08984
--------------------------------	---------

TABLA 4: RESULTADOS DEL MODELO 3

Si comparamos estos datos con la tabla equivalente del modelo anterior, observamos que se reduce ligeramente el empuje, al reducirse el volumen de los brazos, pero se reduce más aún el peso, al ser el material de los brazos más denso que el agua. Esto implica que la flotabilidad no solo sigue siendo positiva, sino que aumenta respecto al modelo anterior. En concreto, la fuerza de la flotabilidad, el empuje menos el peso, es ahora de 0.88 N y en relación al peso de 1.03%. Este dato es más del doble que en el modelo anterior pero aún es bastante bajo, por lo que no consideramos necesario aumentar los pesos añadidos.

El centro de gravedad se mantiene prácticamente en la vertical del centro de carena y por debajo de este, por lo que no producirá problemas de estabilidad. Los momentos de inercia, por su parte, se reducen ligeramente respecto al modelo 2, como es natural al reducirse el peso.

3.1.4. Modelo 4

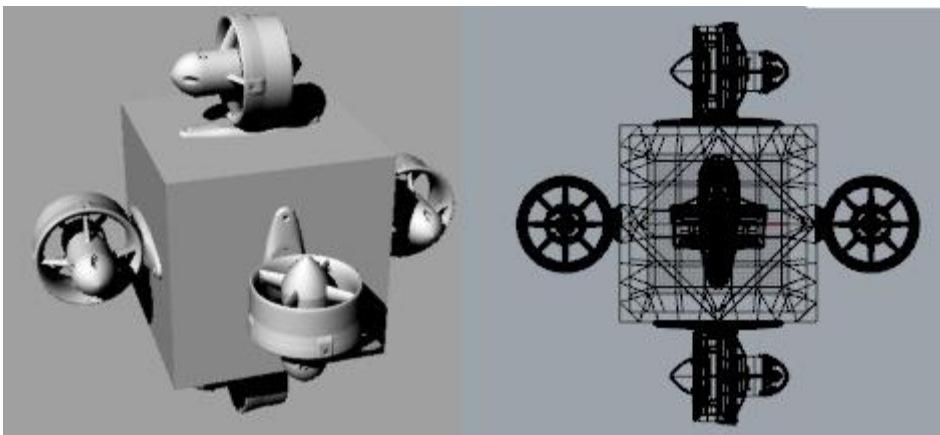


FIGURA 5: VISTA TRIDIMENSIONAL Y FRONTAL DEL MODELO 4

Durante el desarrollo del proyecto se decidió utilizar un modelo específico de hélice-motor, concretamente el modelo T-100 diseñado por Blue Robotics. Por ello, en este modelo se ha incluido el diseño 3D de los motores así como de las sujeciones, ambos proporcionados por el fabricante. Se puede observar en este modelo las hélices son más grandes, de 80 mm a 100 mm, y están menos separadas del cuerpo, la distancia de los ejes de las hélices al centro de carena pasa de 190 mm a 155 mm. Esta reducción de distancia supone generar un menor momento durante los movimientos de giro, pero se estima que no será un problema ya que en las simulaciones el vehículo se comporta de manera similar con estos cambios.

Las dimensiones de este modelo son muy similares al anterior, pero en la siguiente tabla se pueden apreciar algunas diferencias.

Lado Caja (mm)	200
Espesor Caja (mm)	10
Densidad Caja (kg/m³)	1900
Peso propulsor (kg)	0.378
Diámetro Exterior propulsor (mm)	100
Anchura Cilindro Hueco (mm)	45
Espesor Cilindro Hueco (mm)	10
Densidad propulsor (kg/m³)	1845
Masas auxiliares (kg)	3
Distancia Eje Hélice a Centro de Carena (mm)	155

TABLA 5: DATOS DEL MODELO 4

Con estos datos, y a partir del programa de Rhinoceros se pueden calcular los parámetros fundamentales del modelo, utilizando como en el modelo anterior una densidad del agua de 1030 kg/m³.

Masa (kg)	9.387
Empuje (N)	93.25
Peso (N)	92.09
Centro de Gravedad	
CDGx (mm)	1.478
CDGy (mm)	1.478
CDGz (mm)	28.76
Momentos de Inercia Principales	
Ix (kg · m²)	0.10212
Iy (kg · m²)	0.10212
Iz (kg · m²)	0.07782

TABLA 6: RESULTADOS DEL MODELO 4

Si comparamos estos datos con la tabla equivalente del modelo anterior, observamos que aumenta tanto el empuje como el peso, al aumentar el volumen de los brazos, ha aumentado el empuje, y hemos tenido que aumentar el peso de las masas auxiliares para equilibrar las fuerzas. Esto implica que la flotabilidad sigue siendo positiva, con una pequeña diferencia entre ambas. En concreto, la fuerza de la flotabilidad, el empuje menos el peso, es ahora de 1.17 N y en relación al peso de 1.27%. Este dato está dentro de los límites normales para este tipo de vehículos.

El centro de gravedad se mantiene prácticamente en la vertical del centro de carena y por debajo de este, por lo que no producirá problemas de estabilidad, sin embargo, para obtener un equilibrio perfecto se pueden distribuir los pesos añadidos en diferentes puntos del ROV para compensar este efecto. Los momentos de inercia, por su parte, se reducen ligeramente respecto al modelo 3, debido a dos efectos opuestos, la reducción de la distancia entre los ejes de las hélices y el centro de carena disminuye los momentos de inercia, mientras que el aumento de los pesos auxiliares incrementa los momentos de inercia en las direcciones X e Y, ya que los pesos se han supuesto inicialmente colocados en la línea del eje Z.

Este modelo es la mejor aproximación que tenemos actualmente de cómo va a ser el vehículo una vez fabricado. Desafortunadamente, no podemos incluir este modelo 3D en la visualización de la simulación, ya que, la inclusión de los modelos 3D de los propulsores T-100 supone una carga, en términos de procesamiento de información, inasumible para nuestro programa de C++ (El archivo .obj del modelo pesa 27 MB comprimiéndolo lo máximo posible). Como solución, se ha decidido utilizar en la simulación un modelo 3D simplificado basado en el modelo 4, que podemos ver en la siguiente figura.

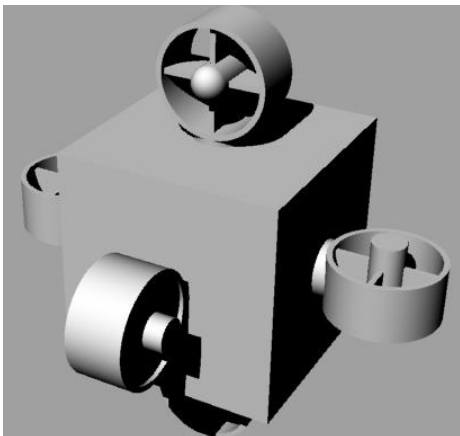


FIGURA 6: MODELO 4 SIMPLIFICADO

Todos los datos necesarios para calcular el comportamiento del vehículo que se incluirán en el programa serán los calculados anteriormente para el modelo 4, es decir, con las propiedades físicas de los propulsores reales.

Referencias: [20]

3.2. Posición de las hélices

Un factor muy importante a la hora de diseñar un ROV es tener claro cómo se va a mover y que propulsores han de actuar en cada caso. En nuestro modelo disponemos de seis hélices, distribuidas en paralelo a las caras del cubo que hace de cuerpo del ROV.

Para poder definir con mayor claridad la posición de estas hélices es necesario introducir un sistema de coordenadas cartesianas local, solidario al vehículo. Este sistema de coordenadas tiene como origen el centro de carena, es decir, el centro del cubo, y los ejes X, Y, Z perpendiculares a las caras del mismo. La dirección del eje X será la dirección de avance, con sentido positivo en la dirección de la cara anterior del cubo, en lenguaje de navegación, la proa. La dirección del eje Y será la dirección de deriva, con sentido positivo en la cara derecha si se mira hacia proa, el lado de estribor. Por último, la dirección del eje Z es la dirección vertical o de arfada, con sentido en la dirección de la cara inferior del cubo. Este último eje puede producir confusión, ya que no es muy usual que el eje Z apunte hacia abajo, suele apuntar hacia arriba, pero en la práctica es muy útil tener este sistema de coordenadas ya que es muy similar al sistema global, el cual mide la profundidad positiva desde la superficie del mar, por lo que el eje Z apunta hacia abajo.

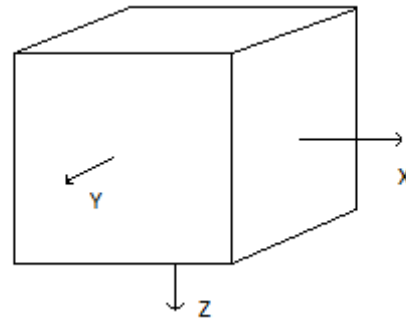


FIGURA 7: COORDENADAS LOCALES

Una vez definido el sistema de coordenadas, podemos concretar mejor donde queremos nuestras hélices. Queremos tener dos hélices para el movimiento en cada dirección y que a su vez estas hélices puedan producir un giro en otra dirección.

Comenzamos con las hélices de avance, que llamaremos hélice 1 y hélice 2. Estas hélices deben colocarse paralelas al eje X y, para que puedan producir momento, deben encontrarse en los ejes Y o Z. Ciertamente estas hélices podrían colocarse en un punto intermedio y no en uno de los ejes, pero para no producir momentos innecesarios y para simplificar un poco el modelo vamos a posicionar las hélices en puntos de los ejes. Si colocamos las hélices en el eje Y, producirán momentos según el eje Z, y podrán modificar el ángulo de guiñada, por otro lado, si las colocamos en el eje Z, producirán momentos según el eje Y, y modificarían el ángulo de cabeceo. Se decidió finalmente colocar las hélices 1 y 2 en el eje Y para controlar con ellas tanto el avance como la velocidad de guiñada, que son los movimientos más importantes del ROV. La posición de estas hélices es por lo tanto, según el sistema de coordenadas local y en milímetros, el siguiente:

Hélice 1:	Posición (0, 155, 0)	Dirección (1, 0, 0)
Hélice 2:	Posición (0, -155, 0)	Dirección (1, 0, 0)

Las siguientes hélices son las hélices de arfada, que controlan el movimiento según el eje Z y que son las encargadas de vencer la fuerza de flotación, las llamaremos hélice 3 y hélice 4. Estas hélices deben colocarse paralelas al eje Z y, para que puedan producir momento, deben encontrarse en los ejes X o Y. Puesto que ya hemos colocado las hélices 1 y 2 en el eje Y, tiene sentido colocar estas en el eje X, para conservar la simetría. Al colocar las hélices en el eje X, producirán momentos según el eje Y, y

modificarán el ángulo de cabeceo. La posición de estas hélices es por lo tanto, según el sistema de coordenadas local y en milímetros, el siguiente:

Hélice 3: Posición (155, 0, 0) Dirección (0, 0, 1)

Hélice 4: Posición (-155, 0, 0) Dirección (0, 0, 1)

Es importante recordar que estas hélices ejercen una fuerza sobre el vehículo hacia abajo, en el sentido del eje Z, como hemos visto antes, por ello la dirección es positiva.

Por último tenemos las hélices 5 y 6, que serán las encargadas de permitir el movimiento de deriva, según el eje Y. Estas hélices se colocarán en el eje Z, ya que es el único libre que queda y en el único en el que pueden proporcionar momentos según el eje X, y así controlar la velocidad de balanceo. La posición de estas hélices es por lo tanto, según el sistema de coordenadas local y en milímetros, el siguiente:

Hélice 5: Posición (0, 0, 155) Dirección (0, 1, 0)

Hélice 6: Posición (0, 0, -155) Dirección (0, 1, 0)

Referencias: [16]

3.3. Pesos auxiliares

Los pesos auxiliares son pequeñas piezas con masa que se pueden distribuir en diferentes posiciones dentro del ROV. Los motivos para colocar estos pesos son los siguientes:

Alinear el centro de carena con el centro de gravedad: Un ajuste fino del centro de gravedad permite que el ROV se mantenga recto de forma natural, sin necesidad del uso de los propulsores.

Modificar la flotabilidad: Es muy conveniente que el ROV mantenga una flotabilidad positiva, que la fuerza de empuje sea mayor que el peso, sin embargo, esta debe ser lo más pequeña posible, para así, se mantenga su posición según la vertical sin necesidad de un excesivo uso de los propulsores.

Bajar el centro de gravedad: Para proporcionar de estabilidad al vehículo marino, el centro de gravedad debe estar por debajo del centro de carena, al colocar pesos por debajo del centro de carena, en la cara inferior de la caja, se desplaza el centro de gravedad. Cuanto mayor sea la distancia vertical entre el centro de gravedad y el centro de carena mayor serán los momentos correctores, por lo tanto mayor será la estabilidad, sin embargo, si los momentos correctores son muy grandes, estos impedirán por completo giros en los sentidos X e Y (balanceo y cabeceo) ya que el momento producido por los propulsores no será capaz de contrarrestar el momento corrector si se quiere alcanzar una posición inclinada.

Permitir la instalación de nuevos componentes: Si se quiere ir desarrollando poco a poco las utilidades del ROV, no es adecuado diseñar el ROV para que funcione

únicamente en las condiciones actuales, si se quisiera añadir un componente nuevo al ROV, y no existieran estos pesos, el peso de este componente provocaría una flotabilidad negativa, lo que sería muy perjudicial en caso de fallo.

En el modelo que hemos elegido, el modelo 3, hemos utilizado un peso de 2 kg que hemos situado en la cara inferior del ROV, 80 mm por debajo del centro de carena para bajar la posición del centro de gravedad y para reducir la diferencia entre la fuerza de empuje y el peso. No ha sido necesario colocar pesos fuera de la vertical del centro de carena ya que al ser el modelo muy simétrico, no es necesario ajustar la posición del centro de gravedad respecto a los ejes X e Y.

Referencias: [16], [17], [18]

4. Modelo matemático

Para poder simular el comportamiento que va a tener el ROV dentro del agua, es necesario realizar un análisis matemático basado en el comportamiento estático y dinámico de los vehículos marinos.

4.1. Sistemas de referencia

Anteriormente se ha definido el sistema de coordenadas cartesianas locales solidario al vehículo, con el eje X apuntando hacia delante, el eje Y hacia la derecha y el eje Z hacia abajo, a este sistema lo denominamos {B}. También existe otro sistema de coordenadas cartesianas globales, fijo a la tierra, al que denominamos {W} y en el que el eje X apunta hacia el norte, el eje Y hacia el este y el eje Z hacia abajo, aunque los ejes X e Y se puede definir en otras direcciones si la situación lo requiere.

Para transformar un vector de un sistema de coordenadas a otro es necesario utilizar una matriz de rotación. Así, un vector en coordenadas locales (P^B) que se desee expresar en el sistema de coordenadas global (P^W) ha de multiplicarse por la matriz de rotación que se define como:

$$R = \begin{bmatrix} \cos\theta \cdot \cos\psi & \sin\phi \cdot \sin\theta \cdot \cos\psi - \cos\phi \cdot \sin\psi & \cos\phi \cdot \sin\theta \cdot \cos\psi + \sin\phi \cdot \sin\psi \\ \cos\theta \cdot \sin\psi & \sin\phi \cdot \sin\theta \cdot \sin\psi + \cos\phi \cdot \cos\psi & \cos\phi \cdot \sin\theta \cdot \sin\psi - \sin\phi \cdot \cos\psi \\ -\sin\theta & \sin\phi \cdot \cos\theta & \cos\phi \cdot \cos\theta \end{bmatrix}$$

$$P^W = R \cdot P^B$$

Si se quiere realizar el proceso inverso basta con calcular la inversa de la matriz de rotación, que coincide con la transpuesta.

$$P^B = R^T \cdot P^W$$

Existe un problema con este sistema cuando $\theta = \pm \pi/2$, ya que al alcanzar estos valores se producen singularidades en la matriz de rotación, sin embargo, un ángulo de cabeceo de 90° es muy difícil de alcanzar por parte del ROV y no tiene ninguna utilidad práctica tomar esa posición. Aun así, en el programa en C++ se ha incluido una excepción en el sistema de ecuaciones para que no se alcancen estos valores y por tanto no se produzcan singularidades.

Con este método se pueden transformar los vectores de velocidad de coordenadas locales a globales y viceversa, como se puede ver a continuación.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} R & 0 \\ 0 & W \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix}$$

Siendo W la matriz de rotación correspondiente para las velocidades angulares.

$$W = \begin{bmatrix} 1 & \sin\phi \cdot \tan\theta & \cos\phi \cdot \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi \cdot \sec\theta & \cos\phi \cdot \sec\theta \end{bmatrix}$$

Referencias: [16], [21]

4.2. Ecuaciones dinámicas

Todas las fuerzas que influyen en el comportamiento dinámico del ROV se pueden expresar en una única ecuación matricial.

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

Esta ecuación matricial consta de los siguientes términos:

- M: Matriz de masas
- C(v): Matriz de Coriolis en función de la velocidad
- D(v): Matriz de resistencia hidrodinámica
- g(η): Vector de flotación función de la posición
- τ: Vector de fuerzas de propulsión
- v y \dot{v} : Vectores de velocidad y aceleración del vehículo

En los siguientes apartados explicaremos en detalle qué significa cada uno de estos términos y cuál es su magnitud en el caso de nuestro vehículo.

Referencias: [16]

4.2.1. Matriz de masas

La matriz de masas (M) está formada por la matriz de la masa y la inercia del vehículo (M_{RB}), así como de la matriz de la masa virtual (M_A).

$$M = M_{RB} + M_A$$

La matriz M_{RB} tiene en cuenta únicamente los efectos inerciales que produce el sólido rígido, pero también hay que tener en cuenta en los cálculos la masa del fluido que

arrastra consigo el vehículo al moverse y que adquiere también cierta inercia al desplazarse. El efecto de este fenómeno suele tener el mismo orden de magnitud que el de la propia masa del vehículo, aunque algo inferior.

La matriz M_{RB} tiene la siguiente estructura.

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & mz_G & -my_G \\ 0 & m & 0 & -mz_G & 0 & mx_G \\ 0 & 0 & m & my_G & -mx_G & 0 \\ 0 & -mz_G & my_G & I_x & I_{xy} & I_{xz} \\ mz_G & 0 & -mx_G & I_{xy} & I_y & I_{yz} \\ -my_G & mx_G & 0 & I_{xz} & I_{yz} & I_z \end{bmatrix}$$

En nuestro caso, el vehículo es simétrico, por lo que los productos de inercia cruzados (I_{xy} , I_{xz} , I_{yz}) son nulos. La posición del centro de gravedad según X e Y también se podrían despreciar, al ser estos casi nulos, y al ser el objetivo del diseño que sean nulos, pero se ha decidido no despreciar estos datos en el programa de C++ para dotar de mayor generalidad al modelo, ya que incluirlos no conlleva mayor complejidad en el cálculo.

Si introducimos los datos que se obtuvieron anteriormente en la tabla 4, y despreciamos los productos de inercia y la posición del centro de gravedad según X e Y, ya que son despreciables, obtenemos la siguiente matriz.

$$M_{RB} = \begin{bmatrix} 9.387 & 0 & 0 & 0 & 0.27 & 0 \\ 0 & 9.387 & 0 & -0.27 & 0 & 0 \\ 0 & 0 & 9.387 & 0 & 0 & 0 \\ 0 & -0.27 & 0 & 0.1021 & 0 & 0 \\ 0.27 & 0 & 0 & 0 & 0.1021 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.07782 \end{bmatrix}$$

La matriz M_A se compone con una serie de coeficientes que dependen de la masa de fluido que arrastra el vehículo en su movimiento debido a su forma y que, al estar el vehículo completamente sumergido, podemos considerar constantes.

La matriz de masa virtual tiene la siguiente forma.

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

Al tener el vehículo tres planos de simetría se puede despreciar la contribución en el comportamiento del vehículo de los elementos fuera de la diagonal de la matriz por lo que se puede simplificar enormemente.

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

Hallar los miembros de la diagonal no es tarea fácil, ya que se suelen obtener de manera experimental, y nosotros no podemos realizar estas pruebas todavía ya que el modelo no se ha construido aún. Sin embargo, podemos realizar algunas aproximaciones sobre la forma del modelo que nos permiten calcular estos términos basándonos en experimentos realizados sobre sólidos con formas simples.

Así, suponiendo que nuestro vehículo tiene forma de elipsoide, y teniendo en cuenta la simetría en las tres direcciones espaciales, podemos calcular los coeficientes de la siguiente forma.

La aproximación de un vehículo con forma de cubo a un elipsoide puede parecer algo burda, pero se aproxima bastante en los ejes Y e Z, de los cuales extrapolamos los resultados para el eje X. En cualquier caso, el mejor método de cálculo de estas componentes es de manera experimental, lo cual se podrá hacer una vez construido el modelo.

Hemos escogido la distancia del cuerpo del ROV entre los extremos de dos brazos opuestos como la dimensión del elipsoide en la dirección de avance, por lo que el semieje principal del elipsoide será de 0.2 metros. Para el semieje secundario se ha escogido la dimensión de la caja, de lado L=0.2 m por lo que el semieje tiene una dimensión de 0.1 metros.

$$a = 0.2 \text{ m} \quad b = 0.1 \text{ m}$$

$$Y_{\dot{v}} = Z_{\dot{w}} = \frac{\beta_0}{2 - \beta_0} \cdot m$$

$$M_{\dot{q}} = N_{\dot{r}} = -\frac{1}{5} \cdot \frac{(b^2 - a^2)^2 \cdot (\alpha_0 - \beta_0)}{2 \cdot (b^2 - a^2) + (b^2 + a^2) \cdot (\beta_0 - \alpha_0)} \cdot m$$

Siendo α_0 y β_0

$$\alpha_0 = \frac{2 \cdot (1 - e^2)}{e^3} \cdot \left(\frac{1}{2} \cdot \ln \left(\frac{1 + e}{1 - e} \right) - e \right) = 0.4624$$

$$\beta_0 = \frac{1}{e^2} - \frac{1 - e^2}{2 \cdot e^3} \cdot \ln \left(\frac{1 + e}{1 - e} \right) = 0.7688$$

Con

$$e = 1 - \left(\frac{b}{a} \right)^2 = 0.75$$

Introduciendo los datos, obtenemos los siguientes resultados.

$$Y_{\dot{v}} = Z_{\dot{w}} = -5.861 \text{ kg}$$

$$M_{\dot{q}} = N_{\dot{r}} = -0.01170 \text{ kg} \cdot \text{m}^2$$

Debido al carácter simétrico del vehículo, podemos inducir los valores de los componentes $X_{\dot{u}}$ y $K_{\dot{p}}$ ya que para estos términos no es válida la aproximación del elipsoide.

$$X_{\dot{u}} = Y_{\dot{v}} = Z_{\dot{w}} = -5.861 \text{ kg}$$

$$K_{\dot{p}} = M_{\dot{q}} = N_{\dot{r}} = -0.01170 \text{ kg} \cdot \text{m}^2$$

Los resultados obtenidos son coherentes, ya que suponen entre el 10% y el 70% de la magnitud de la masa del vehículo, lo que coincide con otros estudios realizados en vehículos autónomos.

Introduciendo los datos de los parámetros en la Matriz de masa virtual, esta quedaría de la siguiente forma.

$$M_A = \begin{bmatrix} 5.861 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5.861 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5.861 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01170 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01170 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01170 \end{bmatrix}$$

Ahora si sumamos las matrices M_{RB} y M_A obtenemos la matriz de masas M

$$M = \begin{bmatrix} 15.248 & 0 & 0 & 0 & 0.27 & 0 \\ 0 & 15.248 & 0 & -0.27 & 0 & 0 \\ 0 & 0 & 15.248 & 0 & 0 & 0 \\ 0 & -0.27 & 0 & 0.1138 & 0 & 0 \\ 0.27 & 0 & 0 & 0 & 0.1138 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8952 \end{bmatrix}$$

Referencias: [16], [22], [23], [24], [25], [26]

4.2.3. Matriz de Coriolis

La matriz de Coriolis permite tener en cuenta los efectos producidos por las fuerzas centrípetas y la fuerza de Coriolis.

La matriz de Coriolis para un sólido rígido que se mueve a través de un fluido ideal tiene la siguiente forma:

$$C_A(v) = \begin{bmatrix} 0_{3 \times 3} & -S(A_{11} \cdot v_1 + A_{12} \cdot v_2) \\ -S(A_{11} \cdot v_1 + A_{12} \cdot v_2) & -S(A_{21} \cdot v_1 + A_{22} \cdot v_2) \end{bmatrix}$$

Siendo las matrices A submatrices 3×3 de la matriz de masas:

$$M = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

Siendo los vectores v_1 y v_2 los vectores 3×1 de posición lineal y angular:

$$v = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

Y finalmente siendo S la transformada de un vector lineal tal que:

$$a = (a_1, a_2, a_3)$$

$$S(a) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

En este caso obtendremos una matriz dependiente de la velocidad en cada instante.

A continuación, vamos a resolver la matriz de forma genérica, suponiendo una matriz de masas y unas velocidades dadas:

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} & m_{36} \\ m_{41} & m_{42} & m_{43} & m_{44} & m_{45} & m_{46} \\ m_{51} & m_{52} & m_{53} & m_{54} & m_{55} & m_{56} \\ m_{61} & m_{62} & m_{63} & m_{64} & m_{65} & m_{66} \end{bmatrix}$$

$$v = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix}$$

$$C_A(v) = \begin{bmatrix} 0 & 0 & 0 & 0 & -a_3 & a_2 \\ 0 & 0 & 0 & a_3 & 0 & -a_1 \\ 0 & 0 & 0 & -a_2 & a_1 & 0 \\ 0 & -a_3 & a_2 & 0 & -b_3 & b_2 \\ a_3 & 0 & -a_1 & b_3 & 0 & -b_1 \\ -a_2 & a_1 & 0 & -b_2 & b_1 & 0 \end{bmatrix}$$

Siendo:

$$a_1 = m_{11} \cdot u + m_{12} \cdot v + m_{13} \cdot w + m_{14} \cdot p + m_{15} \cdot q + m_{16} \cdot r$$

$$a_2 = m_{21} \cdot u + m_{22} \cdot v + m_{23} \cdot w + m_{24} \cdot p + m_{25} \cdot q + m_{26} \cdot r$$

$$a_3 = m_{31} \cdot u + m_{32} \cdot v + m_{33} \cdot w + m_{34} \cdot p + m_{35} \cdot q + m_{36} \cdot r$$

$$b_1 = m_{41} \cdot u + m_{42} \cdot v + m_{43} \cdot w + m_{44} \cdot p + m_{45} \cdot q + m_{46} \cdot r$$

$$b_2 = m_{51} \cdot u + m_{52} \cdot v + m_{53} \cdot w + m_{54} \cdot p + m_{55} \cdot q + m_{56} \cdot r$$

$$b_3 = m_{61} \cdot u + m_{62} \cdot v + m_{63} \cdot w + m_{64} \cdot p + m_{65} \cdot q + m_{66} \cdot r$$

Referencias: [16]

4.2.3. Matriz de resistencia hidrodinámica

La resistencia hidrodinámica de los vehículos submarinos suele ser no lineal y acoplada, y nuestro caso no es diferente. Al tratarse de un vehículo de forma cúbica, los términos no acoplados son iguales en las tres direcciones espaciales, sin embargo, también existen términos cruzados, ya que, la cara que se enfrenta al flujo, cambia de forma drástica si la dirección del vehículo tiene varias componentes en las coordenadas locales, es decir, no se desplaza de forma recta sino en diagonal, desde el punto de vista del vehículo.

Se ha investigado arduamente tanto por mi parte como por la de mi director de proyecto para encontrar algún método para el cálculo teórico de términos acoplados para una forma cúbica, desafortunadamente, no ha sido posible encontrar referencias en este aspecto. Se ha decidido finalmente, ignorar estos términos, suponiendo que todos los movimientos que realiza el ROV tienen una dirección predominante que coincide con una de las coordenadas locales. Al despreciar los términos acoplados, en la matriz de resistencia hidrodinámica solo quedan los términos de la diagonal.

Por otra parte, la baja velocidad nos permite despreciar los términos de tercer orden, aquellos que multiplican a la velocidad al cubo, ya que este término se hace muy pequeño. Por ello, solo vamos a considerar los términos lineales y cuadráticos en nuestra matriz. Una vez despreciado los términos cruzados y los términos de tercer orden o superior, la matriz de resistencia hidrodinámica tiene la siguiente forma.

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

La principal causa de resistencia hidrodinámica es el derramamiento de vórtice, “Vortex Shedding”, el cual tiene únicamente componente cuadrática, pero también consideraremos cierta componente lineal, basada en una fracción de la componente cuadrática, para simular el comportamiento a velocidades muy pequeñas.

Para hallar las componentes cuadráticas, necesitamos conocer primero el coeficiente de resistencia según el área perpendicular al flujo, debido a la simetría del vehículo, esta es la misma según los tres ejes de coordenadas por lo que las componentes de la matriz serán las mismas.

Podemos simplificar nuestro vehículo como un cubo, ya que los brazos son despreciables, y no ofrecen casi resistencia. En el caso de un flujo perpendicular a una de las caras del cubo, este flujo se enfrenta a una superficie de forma cuadrada, cuyo coeficiente de resistencia hidrodinámica es igual a 1.

Con este dato se pueden obtener los parámetros cuadráticos de la matriz, a partir de la siguiente fórmula.

$$X_{u|u} = Y_{v|v} = Z_{w|w} = \frac{1}{2} \cdot \rho \cdot C_D \cdot A$$

Siendo ρ la densidad del agua, C_D el coeficiente de resistencia que hemos calculado antes y A el área de la superficie del vehículo que se enfrenta al flujo.

$$X_{u|u} = Y_{v|v} = Z_{w|w} = \frac{1}{2} \cdot 1030 \cdot 1 \cdot 0.2^2 = 20.6 \text{ kg/m}$$

El resultado obtenido tiene signo positivo, lo cual es lógico ya que la resistencia hidrodinámica es una fuerza que se opone al movimiento y al introducir este término en la matriz $D(v)$ se vuelve negativo, al estar multiplicado por el valor absoluto de la velocidad y otra vez por la velocidad, la fuerza correspondiente tiene signo contrario al de la velocidad, como cabía de esperar.

Los términos correspondientes al movimiento angular son mucho menores que los lineales, se pueden aproximar con la siguiente fórmula:

$$K_{p|p} = M_{q|q} = N_{r|r} = \frac{1}{2} \cdot \rho \cdot C_D \cdot A \cdot R$$

$$K_{p|p} = M_{q|q} = N_{r|r} = \frac{1}{2} \cdot 1030 \cdot 1 \cdot 0.2^2 \cdot 0.155 = 3.19 \text{ kg/m}$$

Estos parámetros son probablemente algo menores que los reales, ya que no hemos tenido en cuenta la influencia de los brazos y hemos considerado el cuerpo como un cubo. Sería interesante por lo tanto realizar pruebas experimentales que determinen estos parámetros de manera más precisa una vez construido el modelo.

Referencias: [16], [21], [22], [24], [25]

4.2.4. Vector de flotación

Existen dos fuerzas que intervienen en la flotación del vehículo, la fuerza de la gravedad (W), que actúa sobre el centro de gravedad, y la fuerza de flotación (B), que actúa sobre el centro de carena. La magnitud de estas fuerzas es fácil de calcular, y su dirección y sentido en coordenadas locales dependerá de la posición angular del vehículo.

Para calcular la fuerza gravitatoria solo es necesario conocer la masa del vehículo y la aceleración de la gravedad.

$$W = m \cdot g = 8.387 \cdot 9.81 = 92.09 \text{ N}$$

Para calcular la fuerza de flotación es necesario conocer el volumen de agua que desplaza el vehículo, así como la densidad del agua y la aceleración de la gravedad. El dato del volumen ya fue calculado en el programa de Rhinoceros.

$$B = -\rho_{agua} \cdot V \cdot g = -1030 \cdot 9.2317 \cdot 10^{-3} \cdot 9.81 = -93.25 \text{ N}$$

El sentido negativo de la fuerza de flotación es debido a que el agua empuja el vehículo hacia arriba, dirección contraria al eje Z según nuestro sistema de coordenadas globales.

Ahora bien, sabemos la dirección de estas fuerzas en coordenadas globales, expresado como vectores sería:

$$f_W = \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix} \quad f_B = \begin{bmatrix} 0 \\ 0 \\ B \end{bmatrix}$$

Como consideramos el centro de carena como el origen de coordenadas de nuestro sistema local, la fuerza de flotación no produce momentos, pero si lo hará la fuerza gravitatoria, por lo tanto, teniendo en cuenta los puntos de aplicación de las fuerza podemos calcular el vector de fuerzas y momentos de flotación en coordenadas globales.

Para calcular las fuerzas solo es necesario realizar la suma de la fuerza gravitatoria y la fuerza de flotación:

$$g_1^W = f_W + f_B = \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ B \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ W + B \end{bmatrix}$$

Para calcular los momentos, la ecuación necesaria es la siguiente:

$$g_2^W = r_G \times f_W = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ x_G & y_G & z_G \\ 0 & 0 & W \end{vmatrix} = \begin{bmatrix} -y_G \cdot W \\ x_G \cdot W \\ 0 \end{bmatrix}$$

Así, el vector de flotación, compuesto por los vectores g_1 y g_2 es el siguiente:

$$g^w = \begin{bmatrix} 0 \\ 0 \\ W + B \\ -y_G \cdot W \\ x_G \cdot W \\ 0 \end{bmatrix}$$

Para obtener ahora el vector en coordenadas locales necesitamos realizar la transformación de las fuerzas de gravedad y flotación multiplicándolas por la transpuesta de la matriz de rotación.

$$g_1^B = (f_W + f_B)^B = R^T \cdot (f_W + f_B)^W = \begin{bmatrix} -(W + B) \cdot \text{sen}(\theta) \\ (W + B) \cdot \text{cos}(\theta) \cdot \text{sen}(\phi) \\ (W + B) \cdot \text{cos}(\theta) \cdot \text{cos}(\phi) \end{bmatrix}$$

$$g_2^B = r_G \times (f_W)^B = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ x_G & y_G & z_G \\ -W \cdot \text{sen}(\theta) & W \cdot \text{cos}(\theta) \cdot \text{sen}(\phi) & W \cdot \text{cos}(\theta) \cdot \text{cos}(\phi) \end{vmatrix}$$

$$g_2^B = \begin{bmatrix} y_G \cdot W \cdot \text{cos}(\theta) \cdot \text{cos}(\phi) - z_G \cdot W \cdot \text{cos}(\theta) \cdot \text{sen}(\phi) \\ -z_G \cdot W \cdot \text{sen}(\theta) - x_G \cdot W \cdot \text{cos}(\theta) \cdot \text{cos}(\phi) \\ x_G \cdot W \cdot \text{cos}(\theta) \cdot \text{sen}(\phi) + y_G \cdot W \cdot \text{sen}(\theta) \end{bmatrix}$$

Realizando los cálculos pertinentes se obtiene el siguiente vector de flotación, ahora sí en coordenadas locales.

$$g^B = \begin{bmatrix} -(W + B) \cdot \text{sen}(\theta) \\ (W + B) \cdot \text{cos}(\theta) \cdot \text{sen}(\phi) \\ (W + B) \cdot \text{cos}(\theta) \cdot \text{cos}(\phi) \\ y_G \cdot W \cdot \text{cos}(\theta) \cdot \text{cos}(\phi) - z_G \cdot W \cdot \text{cos}(\theta) \cdot \text{sen}(\phi) \\ -z_G \cdot W \cdot \text{sen}(\theta) - x_G \cdot W \cdot \text{cos}(\theta) \cdot \text{cos}(\phi) \\ x_G \cdot W \cdot \text{cos}(\theta) \cdot \text{sen}(\phi) + y_G \cdot W \cdot \text{sen}(\theta) \end{bmatrix}$$

Esta matriz depende, como se puede observar, de la posición angular del vehículo, por lo que no es constante, y será necesario conocer en cada instante la posición del ROV para realizar el control del mismo.

Referencias: [16], [27]

4.2.5. Vector de fuerzas de propulsión

4.2.5.1. Cálculo de las fuerzas de propulsión en estado estacionario.

En el modelo inicial del vehículo, la fuerza T que ejerce cada hélice depende cuadráticamente con la velocidad de giro de las mismas según la siguiente ecuación.

$$T_i = T_{nn} \cdot |w_i| \cdot w_i$$

Donde hemos definido T_{nn} como el coeficiente de transformación de las hélices. Este coeficiente depende de la forma de las hélices, su tamaño, el pitch y el número de hojas, y tiene como unidades el kilogramo por metro, al estar expresada la velocidad angular en radianes por segundo y la fuerza en Newtons.

En el caso del control directo de los motores, se les aplica un cierto voltaje que se transforma en velocidad de giro para las hélices, por lo que también podríamos obtener la fuerza a partir del voltaje, sin embargo, en nuestra simulación, para el control directo introduciremos las velocidades de giro deseadas en cada motor para controlar el ROV.

Vamos a suponer que todas las hélices son iguales, y que actúan de igual forma en sentido positivo y negativo, normalmente cuando las hélices giran en reverso producen una fuerza menor. Con estas simplificaciones el coeficiente T_{nn} se mantiene constante e igual para las seis hélices.

Para hallar el valor de T_{nn} hemos recurrido a la página de características de un propulsor comercial, concretamente del propulsor T-100 perteneciente a la compañía Blue Robotics, del cual obtenemos una fuerza de 21.4 Newtons para la velocidad máxima, que es aproximadamente de 3800 rpm. Con estos datos podemos calcular el factor T_{nn} .

$$T_{nn} = \frac{F}{\omega^2} = \frac{21.4}{\left(3800 \cdot \frac{2\pi}{60}\right)^2} = 1.35 \cdot 10^{-4} \text{ N/s}^2$$

4.2.5.2. Cálculo de las fuerzas de propulsión con flujo cuasi-estacionario.

En el modelo anterior, hemos tenido en cuenta únicamente la velocidad angular de la hélice para determinar la fuerza propulsora. Este modelo es muy sencillo, y se puede implementar fácilmente en nuestro programa, sin embargo, existen otros modelos más complejos, que tienen en cuenta el flujo incidente en la hélice, y que se asemejan con mayor precisión al comportamiento real de la hélice.

La ecuación general de este modelo es la siguiente:

$$T = 2 \cdot \rho \cdot A_p (k_1 \cdot u + k_2 \cdot D \cdot \omega) (k_1 \cdot u + k_2 \cdot D \cdot \omega - u)$$

Donde ρ es la densidad del agua, en agua salada es de 1030 kg/m^3 , A_p es el área del flujo al pasar por la hélice, en nuestro caso, es de 0.03142 m^2 , k_1 y k_2 son constantes que dependen de la forma específica de la hélices y D el diámetro característico de la hélice 76 mm.

Como podemos observar en la ecuación, ahora tenemos dos variables, la velocidad angular de la hélice (ω) y la velocidad del flujo incidente (u).

La ecuación anterior también se puede agrupar de la siguiente forma:

$$T = 2 \cdot \rho \cdot A_p (c_1 \cdot u^2 + c_2 \cdot u \cdot D \cdot \omega + c_3 \cdot D^2 \cdot \omega^2)$$

Con: $c_1 = k_1^2 - k_1$; $c_2 = 2 \cdot k_2 \cdot k_1 - k_2$; $c_3 = k_2^2$

Para calcular las constantes k_1 y k_2 nos podemos servir de los ensayos realizados en el propulsor T-100 de Blue Robotics, puesto que ya hemos decidido que este va a ser el propulsor a utilizar. En el siguiente gráfico se muestran los resultados obtenidos en el ensayo:

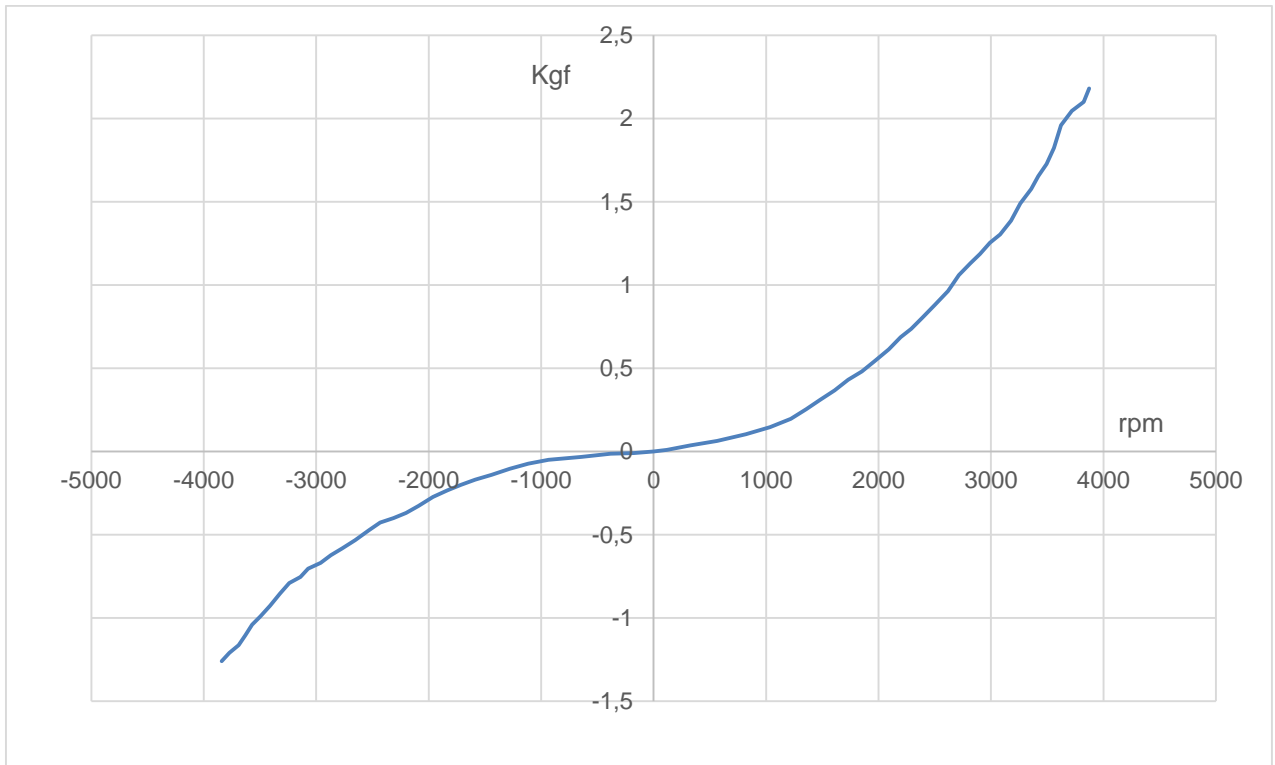


FIGURA 8: ENSAYO PROPULSOR T-100

En la gráfica se puede observar la fuerza otorgada por el propulsor en función de la velocidad angular. Este ensayo se ha realizado en estado estacionario, es decir, sin flujo incidente. Esto significa, que si queremos adaptar los valores k_1 y k_2 a la curva anterior, solo podremos calcular el valor de k_2 , ya que k_1 va siempre ligado a la velocidad incidente, y en este ensayo ese término se hace cero.

Otra particularidad a destacar, es la diferencia en la fuerza aportada dependiendo del sentido de giro, siendo significativamente menor en reversa. Se deberán por tanto formular dos ecuaciones, una para cada sentido de la velocidad de giro.

Primeramente vamos a calcular el valor de k_2 para el sentido positivo de la velocidad angular. Con la velocidad del flujo incidente nula, se eliminan muchos términos de la ecuación, que queda de la siguiente forma:

$$T = 2 \cdot \rho \cdot A_p (k_2 \cdot D \cdot w)(k_2 \cdot D \cdot w) = 2 \cdot \rho \cdot A_p \cdot k_2^2 \cdot D^2 \cdot w^2$$

Si calculamos la curva $T = C \cdot w^2$ que pasa por los puntos de la gráfica anterior, obtenemos:

$$C = 1.42889 \cdot 10^{-7} \frac{kgf}{rpm^2} = 1.278 \cdot 10^{-4} N/s^2$$

Comparando este resultado con la ecuación general podemos calcular k_2 :

$$C = 2 \cdot \rho \cdot A_p \cdot k_2^2 \cdot D^2$$

$$k_2 = \left(\frac{C}{2 \cdot \rho \cdot A_p \cdot D^2} \right)^{\frac{1}{2}} = 0.01849 \text{ s}^{-1}$$

De la misma forma se puede calcular C' para velocidades angulares negativas:

$$C' = -9.1181 \cdot 10^{-8} \frac{\text{kgf}}{\text{rpm}^2} = -8.157 \cdot 10^{-5} \text{ N/s}^2$$

El símbolo negativo de esta constante indica el sentido negativo de la fuerza. Utilizaremos el valor absoluto para calcular k'₂ ya que, de lo contrario, nos encontramos con resultados imaginarios. Cambiar el signo a la constante no supone un problema si tenemos claro cuál es el sentido de la fuerza en función del sentido de giro de las hélices.

Comparando este resultado con la ecuación general podemos calcular k'₂:

$$C' = 2 \cdot \rho \cdot A_p \cdot k'^2_2 \cdot D^2$$

$$k'^2_2 = \left(\frac{C'}{2 \cdot \rho \cdot A_p \cdot D^2} \right)^{\frac{1}{2}} = 0.01477 \text{ s}^{-1}$$

No tenemos ningún dato experimental que nos ayude a calcular el valor de k₁, sin embargo, hemos encontrado ecuaciones correspondientes a diferentes propulsores como la siguiente:

Dynamic Thrust Equation

F = thrust (N), d = prop diam. (in.), RPM = prop rotations/min., pitch = prop pitch (in.), V₀ = propeller forward airspeed (m/s)

Expanded Form:

$$F = 1.225 \frac{\pi(0.0254 \cdot d)^2}{4} \left[\left(\text{RPM}_{prop} \cdot 0.0254 \cdot \text{pitch} \cdot \frac{1 \text{min}}{60 \text{sec}} \right)^2 - \left(\text{RPM}_{prop} \cdot 0.0254 \cdot \text{pitch} \cdot \frac{1 \text{min}}{60 \text{sec}} \right) V_0 \right] \left(\frac{d}{3.29546 \cdot \text{pitch}} \right)^{1.5}$$

Simplified Form:

$$F = 4.392399 \times 10^{-8} \cdot \text{RPM} \frac{d^{3.5}}{\sqrt{\text{pitch}}} (4.23333 \times 10^{-4} \cdot \text{RPM} \cdot \text{pitch} - V_0)$$

Gabriel Staples, 2013. <http://electricrcaircraftguy.blogspot.com/>

FIGURA 9: ECUACIONES DE PROPULSIÓN

En esta ecuación, el pitch se refiere al avance que realiza una hélice al completar una revolución, dependiendo esta del diámetro y el ángulo de la hélice, en nuestro caso el pitch es de 1.9 pulgadas.

Utilizando la forma simplificada, podemos obtener la relación entre los términos relacionados con la velocidad de giro al cuadrado y los términos relacionados con la velocidad incidente por la velocidad de giro. Si comparamos esta relación con la que existe en nuestra ecuación agrupada por términos tenemos:

$$\frac{c_3 \cdot D^2}{c_2 \cdot D} = \frac{k_2 \cdot D}{2 \cdot k_1 - 1} \approx -7.6808 \cdot 10^{-3}$$

Sustituyendo k_2 obtenemos una aproximación de la constante k_1 para cada sentido de la marcha:

$$k_1 = 0.41 \text{ s}^{-1} ; k'_1 = 0.43 \text{ s}^{-1}$$

Ahora podemos sustituir todos estos coeficientes en la ecuación inicial:

$$T = 0.00012781 \cdot w^2 - 0.016645 \cdot u \cdot w - 15.639 \cdot u^2$$

$$T' = 0.000081557 \cdot w^2 - 0.010622 \cdot u \cdot w - 15.835 \cdot u^2$$

La velocidad angular de cada hélice vendrá dada por el ancho de pulso en el controlador, como podemos observar en la siguiente gráfica.

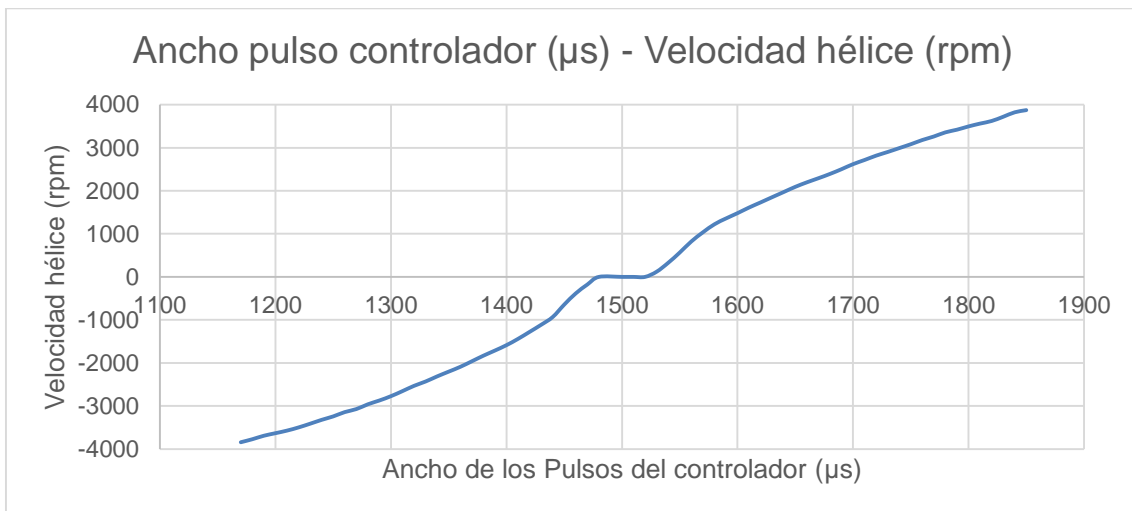


FIGURA 10: RELACIÓN VELOCIDAD-ANCHO DE PULSO PROPULSOR T-100

En nuestro programa utilizaremos los puntos de esta curva para transformar el ancho de pulso en velocidad de giro por interpolación.

4.2.5.3. Variaciones en la velocidad de giro.

Para simular el comportamiento de los propulsores, ya que estos no alcanzan las velocidades que les indicamos de forma instantánea, se utilizó inicialmente una curva hipotética de arranque que tiende a la velocidad máxima de los motores. La ecuación que caracteriza a esta curva es la siguiente:

$$w_i = w_{i-1} + (w_o - w_{i-1}) \cdot h$$

En esta ecuación w_i es la velocidad angular en el instante i , w_{i-1} es la velocidad angular en el instante anterior, w_o es la velocidad angular marcada por el controlador, a la que tiende la curva y h es el periodo entre un instante y el siguiente, también llamado paso, que utilizaremos en nuestro proceso iterativo.

La curva de arranque tiene una forma como la que podemos observar en la siguiente gráfica, en la que se arranca un propulsor desde una posición estacionaria, hasta una velocidad de dos mil revoluciones por minuto.

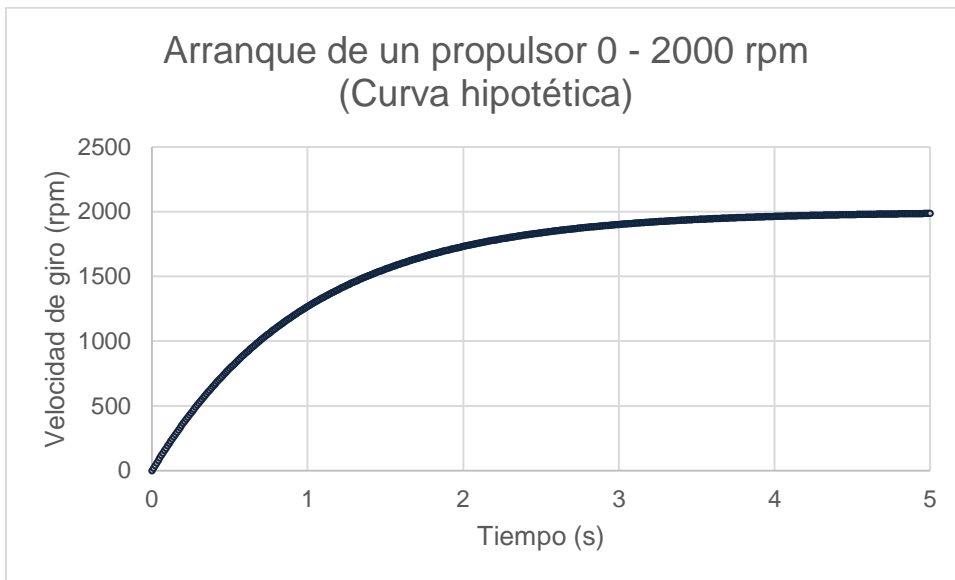


FIGURA 11: REPRESENTACIÓN DEL MODELO DE ARRANQUE DE LOS MOTORES DESDE PARADA HASTA 2000 RPM SEGÚN NUESTRA ECUACIÓN HIPOTÉTICA.

Se puede observar en esta gráfica como el arranque tarda unos tres segundos, en este tiempo la diferencia entre la velocidad de giro real y la marcada es menor del 5%.

Durante el progreso de la realización de este proyecto, sin embargo, se decidió utilizar una curva de arranque más realista, basada en los parámetros eléctricos de los motores. Esta curva se caracteriza con las siguientes ecuaciones:

$$V_r = k_b \cdot N_{prop}$$

$$I_i = \frac{V_i - V_r - L_i \cdot dI_i}{R_i}$$

$$M_m = k_m \cdot I_i$$

$$M_c = T_{nn} \cdot N_{prop} \cdot |N_{prop}|$$

$$N_{prop(i)} = N_{prop(i-1)} + \frac{\left(M_m - M_c - D_v \cdot N_{prop(i-1)} - D_c \cdot \text{sign}(N_{prop(i-1)}) \right)}{J_c + J_m}$$

El significado de cada símbolo se puede consultar en la siguiente tabla:

Vr	Tensión del rotor
Vi	Tensión de alimentación
Kb	Constante de velocidad
Nprop	Velocidad de giro (rad/s)
Ii	Corriente del inducido
Jc	Momento de Inercia de las hélices
Jm	Momento de Inercia del rotor
Ri	Resistencia interna
Li	Inductancia del motor
Dv	Par fricción viscosa
Dc	Par fricción de Coulomb
Mm	Par de la carga
Mc	Par de las hélices
Tnn	Coefficiente de fuerza de las hélices

Las constantes utilizadas para la resolución de las ecuaciones se obtuvieron a partir de los datos de un motor real proporcionados por el fabricante. El motor utilizado ha sido el modelo 82 890 002 del fabricante Crouzet Automation. Las constantes obtenidas fueron las siguientes:

Fabricante	Modelo	Tipo	Potencia (w)	Jm (Kgm ²)	Dv	Dc	Ri	Li
CROUZET AUTOMATION	83 890 002	M	95	7,95E-05	0	0,025	1,4	4,62E-03

Si resolvemos estas ecuaciones para el caso del arranque de un motor desde parada obtenemos la siguiente curva:

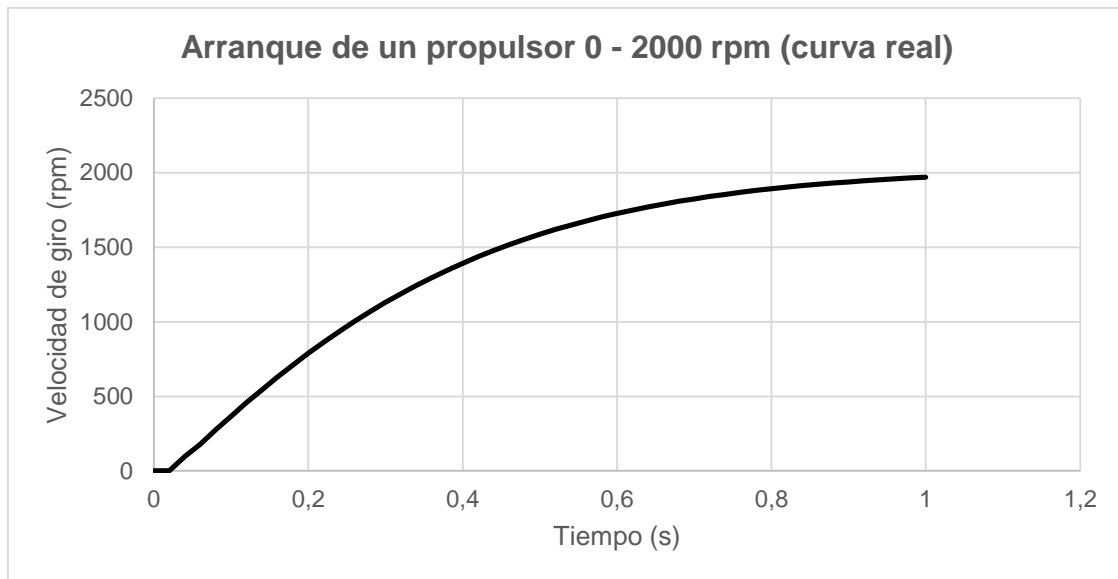


FIGURA 12: REPRESENTACIÓN DEL MODELO DE ARRANQUE DE LOS MOTORES DESDE PARADA HASTA 2000 RPM SEGÚN EL MODELO ELÉCTRICO DEL MOTOR.

Podemos observar que esta curva es muy parecida a la obtenida anteriormente en cuanto a forma, pero es mucho más rápida, ya que alcanza el máximo en apenas un segundo, mientras que en el modelo anterior tardaba más de tres segundos.

Se consideró la posibilidad de utilizar en nuestro programa informático la curva hipotética, ya que es mucho más simple de implementar, y se pueden modificar ciertos parámetros para asemejarla a la curva real de forma cuantitativa. Sin embargo, la posibilidad de generar cualquier tipo de curva a partir únicamente de los datos eléctricos del motor, ofrece una mayor versatilidad en caso de que se quiera cambiar el modelo del motor.

Finalmente se decidió implementar únicamente las curvas del propulsor real T-100, ya que los motores no son de corriente continua, no se puede adaptar la curva eléctrica a los nuevos motores. Este estudio, sin embargo, puede ser útil en caso de que se decida cambiar los motores actuales por unos de corriente continua.

4.2.5.4. Vectorización.

Las fuerzas de propulsión son las ejercidas por el vehículo a través de las hélices que giran conectadas a los motores. Por lo tanto, este vector dependerá de la posición, orientación y fuerza ejercida por estas hélices.

El vector τ se puede calcular a partir de la matriz de orientación y posición L y el vector de fuerzas de los propulsores U .

La matriz L está formada por seis filas, para introducir el vector unitario de orientación (u) y el vector de posición (r) de cada propulsor, y seis columnas, tantas como el número de propulsores. De forma general una matriz L con para un número N

de propulsores tiene la siguiente forma. Donde u_i y r_i son vectores columna de dimensión tres.

$$\begin{bmatrix} u_1 & u_2 & \dots & u_N \\ r_1 & r_2 & \dots & r_N \end{bmatrix}$$

Para nuestro caso la matriz L sería la siguiente.

$$L = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & l & -l & 0 & 0 \\ l & -l & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & l & -l \end{bmatrix}$$

Donde l es la distancia de los propulsores al centro de carena (l= 190 mm)

Esta matriz L se puede dividir en dos. Por un lado la matriz L_1 con los vectores de orientación y por otro la matriz L_2 con los vectores de posición.

$$L_1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$L_2 = \begin{bmatrix} 0 & 0 & l & -l & 0 & 0 \\ l & -l & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & l & -l \end{bmatrix}$$

El vector de fuerzas de propulsión es un vector columna de dimensión igual al número de propulsores. De forma general un vector U con para un número N de propulsores tiene la siguiente forma. Donde T_i son vectores columna de dimensión tres.

$$U = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_N \end{bmatrix}$$

Para calcular las fuerzas y momentos de propulsión necesitamos utilizar las matrices L_1 y L_2 así como el vector T.

Las fuerzas se calculan fácilmente multiplicando el vector L_1 por el vector T.

$$\tau_1 = L_1 \cdot U = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix} = \begin{bmatrix} T_1 + T_2 \\ T_5 + T_6 \\ T_3 + T_4 \end{bmatrix}$$

Para calcular los momentos es necesario realizar el producto vectorial de cada vector posición por su vector unitario de orientación y por la fuerza de dicho propulsor.

$$\tau_{2,i} = r_i \times u_i \cdot T_i$$

El sumatorio de estos vectores produce el vector τ_2 .

$$\tau_2 = \sum_{i=1}^6 \tau_{2,i} = \begin{bmatrix} l \cdot (T_6 - T_5) \\ l \cdot (T_4 - T_3) \\ l \cdot (T_2 - T_1) \end{bmatrix}$$

Uniendo los vectores τ_1 y τ_2 obtenemos el vector de fuerzas de propulsión.

$$\tau = \begin{bmatrix} T_1 + T_2 \\ T_5 + T_6 \\ T_3 + T_4 \\ l \cdot (T_6 - T_5) \\ l \cdot (T_4 - T_3) \\ l \cdot (T_2 - T_1) \end{bmatrix}$$

Referencias: [16], [20], [27], [28], [29], [30]

4.3. Resolución de las ecuaciones

Para la resolución de las ecuaciones necesitamos despejar el vector de aceleraciones lineales y angulares, ya que tanto la velocidad como la posición en cada instante son conocidas y con la aceleración se pueden calcular estos valores en el instante siguiente, así de esta forma se puede simular el comportamiento del ROV para unos valores dados de velocidad angular de los propulsores.

Podemos comparar las ecuaciones dinámicas del vehículo con las fuerzas y momentos externos para calcular las fuerzas y momentos reales que sufre el vehículo en cada dirección, a partir de estas, se puede hallar la aceleración dividiendo las fuerzas por la matriz de masas.

Las ecuaciones para calcular las fuerzas y momentos son las siguientes:

$$F_X = -(B - W) \cdot \sin\theta + X_{u|u|} \cdot u|u| + X_u \cdot u \\ + m (w \cdot q - v \cdot r - x_G \cdot (q^2 + r^2) + y_G \cdot p \cdot q + z_G \cdot p \cdot r) + T_1 + T_2$$

$$F_Y = (B - W) \cdot \cos\theta \cdot \sin\phi + Y_{v|v|} \cdot v|v| + Y_v \cdot v \\ - m (u \cdot r - w \cdot p - y_G \cdot (p^2 + r^2) + x_G \cdot p \cdot q + z_G \cdot q \cdot r) + T_5 + T_6$$

$$F_z = (B - W) \cdot \cos\theta \cdot \cos\phi + Z_{w|w|} \cdot w|w| + Z_w \cdot w \\ - m(v \cdot p - u \cdot q - z_G \cdot (p^2 + q^2) + x_G \cdot p \cdot r + y_G \cdot q \cdot r) + T_3 + T_4$$

$$M'_K = y_G \cdot W \cdot \cos\theta \cdot \cos\phi - z_G \cdot W \cdot \cos\theta \cdot \sin\phi + K_{p|p|} \cdot p|p| + K_p \cdot p \\ + (I_Y - I_Z) \cdot q \cdot r + m \cdot y_G \cdot (u \cdot q - v \cdot p) + m \cdot z_G \cdot (u \cdot r - w \cdot p) \\ + l(T_6 - T_5)$$

$$M'_M = -x_G \cdot W \cdot \cos\theta \cdot \cos\phi - z_G \cdot W \cdot \sin\theta + M_{q|q|} \cdot q|q| + M_q \cdot q + (I_Z - I_X) \cdot r \\ \cdot p + m \cdot z_G \cdot (v \cdot r - w \cdot q) + m \cdot x_G \cdot (v \cdot p - u \cdot q) + l(T_4 - T_3)$$

$$M'_N = x_G \cdot W \cdot \cos\theta \cdot \sin\phi + y_G \cdot W \cdot \sin\phi + N_{r|r|} \cdot r|r| + N_r \cdot r + (I_X - I_Y) \cdot p \cdot q \\ + m \cdot y_G \cdot (w \cdot q - v \cdot r) + m \cdot x_G \cdot (w \cdot p - u \cdot r) + l(T_2 - T_1)$$

Juntas forman el vector de fuerzas:

$$V'_F = (F_X, F_Y, F_X, M'_K, M'_M, M'_N)$$

Este método no despeja por completo la aceleración ya que los momentos según X e Y dependen de las aceleraciones lineales según Y e X respectivamente, sin embargo, este problema se puede solucionar despreciando en un principio estos términos, y una vez calculada las aceleraciones lineales según X e Y se calculan de nuevo estos momentos con los nuevos datos de aceleraciones obtenidos.

$$\dot{u} = (M_{masa}^{-1} \cdot V'_F)_1 \\ \dot{v} = (M_{masa}^{-1} \cdot V'_F)_2 \\ M_K = M'_K - m \cdot z_G \cdot \dot{v} \\ M_M = M'_M - m \cdot z_G \cdot \dot{u} \\ V_F = (F_X, F_Y, F_X, M_K, M_M, M_N)$$

Ahora se puede despejar el vector aceleración con el nuevo vector de fuerzas.

$$V_{acel} = M_{masa}^{-1} \cdot V_F$$

Para calcular la posición y la velocidad del instante siguiente, se recurre al método de Euler, es el método más simple para calcular la siguiente iteración, pero se ha demostrado que no es necesario recurrir a métodos más complicados como el método de Runge-Kutta ya que no suponen ninguna diferencia en los resultados obtenidos y únicamente aumentan el tiempo de computación.

$$V_{vel}^{i+1} = V_{vel}^i + h \cdot V_{acel}$$

Siendo h el tiempo de paso, la diferencia de tiempo entre dos iteraciones consecutivas, para conseguir la mayor precisión, el tiempo h debe ser lo menor posible, en el caso de nuestra simulación por ordenador hemos elegido 0.02 segundos.

Referencias: [16], [21], [24], [25], [26], [27], [31]

5. Control

Existen dos formas de control que hemos desarrollado en nuestra simulación. Primeramente se programa un control manual de la velocidad de giro de los motores, en el que se indica en cada momento la velocidad que estos deben adoptar y, una vez conseguido esto, se desarrolla un sistema de control más complejo, en el que en vez de controlar los motores, se definen puntos de una trayectoria a los que el ROV debe desplazarse dentro de un pequeño límite de error.

5.1. Control manual

Existen dos formas de control manual, por una parte, podemos introducir previamente en nuestra simulación la velocidad que queremos que adopten los motores en función del tiempo, este es un sistema de control muy sencillo que simplemente introduce los valores w_i indicados, y, a partir de las ecuaciones desarrolladas en el modelo matemático simula el comportamiento (posición y velocidad) del ROV.

Sin embargo, también podemos controlar el ROV en tiempo real, esto es, introducir los valores w_i según avanza la simulación, pudiendo así tomar decisiones al instante en caso de querer franquear un obstáculo o alcanzar una posición con precisión. Puesto que controlar los seis motores a la vez es complicado y no es nada práctico se decidió introducir una serie de comandos que adoptaran configuraciones de velocidades clásicas que permitieran controlar el ROV sin demasiada dificultad.

Estos comandos producen movimientos y giros en las tres dimensiones espaciales y en los dos sentidos, por lo tanto el número de comandos diferentes es doce. En nuestra simulación se ha decidido utilizar el teclado como mando controlador ya que la introducción de comandos a través de un mando externo es bastante compleja y no es el objetivo de nuestro proyecto.

A continuación se indican cada uno de los comandos y su relación con la velocidad de giro que adoptan los propulsores.

MOVIMIENTO	TECLA	W_1	W_2	W_3	W_4	W_5	W_6
AVANCE	W	2000	2000	1380	1380	0	0
RETROCESO	S	-2000	-2000	1380	1380	0	0
DERIVA DCHA	D	0	0	1380	1380	2000	2000
DERIVA IZQ	A	0	0	1380	1380	-2000	-2000
ARRIBA	R	0	0	0	0	0	0
ABAJO	F	0	0	2000	2000	0	0
GUIÑADA DCHA	Ñ	-2000	2000	1380	1380	0	0
GUIÑADA IZDA	K	2000	-2000	1380	1380	0	0
CABECEO ABAJO	O	0	0	2000	760	0	0
CABECEO ARRIBA	L	0	0	760	2000	0	0
BALANCEO DCHA	P	0	0	1380	1380	-2000	2000
BALANCEO IZDA	I	0	0	1380	1380	2000	-2000

TABLA 7: COMANDOS EN EL CONTROL MANUAL EN TIEMPO REAL

En esta tabla las velocidades angulares se expresan en revoluciones por minuto, siendo 2000 rpm las revoluciones máximas que soportan las hélices, 1380 rpm las revoluciones necesarias de las hélices 3 y 4 para contrarrestar el efecto de la flotación y mantenerse en una posición estática según el eje Z, debido a que la flotación del vehículo no es nula, sino ligeramente negativa por razones de seguridad.

5.2. Control de trayectoria

El control de trayectoria consiste en el control automático de los propulsores del ROV en función de la distancia y orientación del mismo con respecto a un punto marcado en la trayectoria que se ha definido previamente.

Para conseguir que el ROV tienda a este punto, es necesario crear un controlador que tenga en cuenta el error cometido por el ROV en cada momento, esto es, la distancia entre la posición del ROV y la posición del punto definido y la diferencia angular entre la orientación del ROV y la orientación determinada en el punto.

En nuestro caso implantaremos un controlador proporcional derivativo (PD), este tipo de controlador tiene en cuenta una acción proporcional al error de posición y angular que tiene el ROV en ese instante, así como una acción proporcional a la derivada del error, esto es, la diferencia entre el error que se comete en este instante con el cometido en el instante anterior.

Para simplificar el controlador, se desacoplan los movimientos en las tres direcciones espaciales. Los motores 1 y 2 se encargarán únicamente de alcanzar la posición establecida según el eje X y el giro según el eje Z, los motores 3 y 4 se encargarán de alcanzar la posición establecida según el eje Z y el giro según el eje Y los motores 5 y 6 se encargarán de alcanzar la posición establecida según el eje Y el giro según el eje X. De esta forma se pueden crear controladores para cada uno de los propulsores que solo dependen de dos variables.

Para calcular los errores debemos conocer la posición y orientación de nuestro ROV. Hay que tener en cuenta también, que el error de posición debe estar referido al sistema de coordenadas local, el solidario al cuerpo del ROV.

Los errores en el sistema global de coordenadas vendrían expresados en las siguientes matrices.

$$\xi_p = \begin{bmatrix} P_{i,x} - x \\ P_{i,y} - y \\ P_{i,z} - z \end{bmatrix}$$

$$\xi_\alpha = \begin{bmatrix} P_{i,\phi} - \phi \\ P_{i,\theta} - \theta \\ P_{i,\psi} - \psi \end{bmatrix}$$

Siendo x, y, z las coordenadas de posición del ROV, ϕ, θ, ψ los ángulos de orientación del ROV y los términos P los valores equivalentes para el punto P definido en la trayectoria.

Para obtener el error en coordenadas locales es necesario multiplicarlo por la matriz de rotación en el caso del error de posición, y por la matriz W en el caso del de orientación.

$$\xi_p^B = R^T \xi_p^W$$

$$\xi_\alpha^B = W^{-1} \xi_\alpha^W$$

Una vez hallados los errores en el sistema de coordenadas local, podremos introducirlos en nuestro controlador. Puesto que en nuestro caso hemos desacoplado el controlador, tenemos seis ecuaciones diferentes, una para cada propulsor, que dependen cada una de dos variables diferentes, tendrían la siguiente forma.

$$PWM_1 = K_{P,1,p} \left(\xi_{p,1}^B + T_{D,1,p} \cdot \frac{d\xi_{p,1}^B}{dt} \right) - K_{P,1,\alpha} \left(\xi_{\alpha,3}^B + T_{D,1,\alpha} \cdot \frac{d\xi_{\alpha,3}^B}{dt} \right)$$

$$PWM_2 = K_{P,2,p} \left(\xi_{p,1}^B + T_{D,2,p} \cdot \frac{d\xi_{p,1}^B}{dt} \right) + K_{P,2,\alpha} \left(\xi_{\alpha,3}^B + T_{D,2,\alpha} \cdot \frac{d\xi_{\alpha,3}^B}{dt} \right)$$

$$PWM_3 = K_{P,3,p} \left(\xi_{p,3}^B + T_{D,3,p} \cdot \frac{d\xi_{p,3}^B}{dt} \right) - K_{P,3,\alpha} \left(\xi_{\alpha,2}^B + T_{D,3,\alpha} \cdot \frac{d\xi_{\alpha,2}^B}{dt} \right)$$

$$PWM_4 = K_{P,4,p} \left(\xi_{p,3}^B + T_{D,4,p} \cdot \frac{d\xi_{p,3}^B}{dt} \right) + K_{P,4,\alpha} \left(\xi_{\alpha,2}^B + T_{D,4,\alpha} \cdot \frac{d\xi_{\alpha,2}^B}{dt} \right)$$

$$PWM_5 = K_{P,5,p} \left(\xi_{p,2}^B + T_{D,5,p} \cdot \frac{d\xi_{p,2}^B}{dt} \right) - K_{P,5,\alpha} \left(\xi_{\alpha,1}^B + T_{D,5,\alpha} \cdot \frac{d\xi_{\alpha,1}^B}{dt} \right)$$

$$PWM_6 = K_{P,6,p} \left(\xi_{p,2}^B + T_{D,6,p} \cdot \frac{d\xi_{p,2}^B}{dt} \right) + K_{P,6,\alpha} \left(\xi_{\alpha,1}^B + T_{D,6,\alpha} \cdot \frac{d\xi_{\alpha,1}^B}{dt} \right)$$

Los términos K_P en estas ecuaciones corresponden a las constantes del control proporcional y los términos T_D a las constantes del control derivativo. El termino PWM se refiere al ancho de pulso de la señal del controlador, que utilizamos para variar la velocidad de giro de cada una de las hélices.

Para calcular el error de velocidad se calcula la derivada del error con respecto al tiempo con aproximación por diferencias finitas.

$$\frac{d\xi}{dt} = \lim_{h \rightarrow 0} \frac{\xi_i - \xi_{i-1}}{h}$$

Donde ξ_i es el error en un instante, ξ_{i-1} el error en el instante anterior y h el periodo entre ambos instantes, cuando el paso h tiende a cero la aproximación se vuelve igualdad, como vamos a trabajar con periodos de paso muy pequeños ($h < 0.1$ s) se puede considerar correcta la aproximación.

$$\frac{d\xi}{dt} = \frac{\xi_i - \xi_{i-1}}{h}$$

Los valores de los términos K_P y los términos T_D se obtienen de forma experimental por el método de prueba y error, hasta encontrar unos términos para cada propulsor que consigan una solución de compromiso entre un movimiento rápido y sin producir subamortiguamiento. En la siguiente tabla se exponen los valores elegidos para cada término.

Nº Prop	$K_{P,p}$	$T_{D,p}$	$K_{P,a}$	$T_{D,a}$
1	1	0.4	1000	0.4
2	1	0.4	1000	0.4
3	2	0.4	1000	0.6
4	2	0.4	1000	0.6
5	1	0.4	1000	0.4
6	1	0.4	1000	0.4

TABLA 8: PARÁMETROS DEL SISTEMA DE CONTROL POR TRAYECTORIA

En esta tabla destaca el valor del término de control proporcional de posición para los propulsores 3 y 4, estos motores son los encargados del movimiento vertical, por lo que deben contrarrestar una fuerza constante de flotación que, en caso de elegir un valor más pequeño para este término, impediría alcanzar la posición deseada, ya que el ROV se mantendría por encima del punto, sin capacidad de alcanzar nunca su objetivo.

Existen problemas derivados de la decisión de elegir un valor tan alto para el control proporcional de posición, un término tan grande produce un gran subamortiguamiento y una amplia oscilación del ROV entorno a la posición objetivo, para solucionar esto, se decidió aumentar a su vez el término de control derivativo, esto logró reducir las oscilaciones y permitió al ROV alcanzar los puntos establecidos.

Otra solución propuesta para este problema fue la de incluir en el controlador un término constante que contrarrestara de manera continua la fuerza de flotación. Aunque la idea se probó en la simulación y produjo los resultados deseados, se decidió finalmente no incluir este término para dotar de más generalidad al controlador, ya que, en caso de una modificación de los parámetros que influyen en la flotación del vehículo, este controlador no sería válido para una situación diferente.

Para saber si el vehículo ha alcanzado un punto, es necesario definir unos límites según los cuales podamos concretar si se debe continuar al siguiente punto en la

trayectoria, o si todavía se debe ajustar más la posición o la orientación para que coincidan con la deseada.

Definimos el error de posición total como la distancia en términos absolutos entre la posición del ROV y la posición del punto objetivo.

$$\xi_{pT} = |\xi_p| = \sqrt{(P_{i,x} - x)^2 + (P_{i,y} - y)^2 + (P_{i,z} - z)^2}$$

El error de orientación total lo definimos de forma similar utilizando el vector de error de orientación.

$$\xi_{pT} = |\xi_p| = \sqrt{(P_{i,\phi} - \phi)^2 + (P_{i,\theta} - \theta)^2 + (P_{i,\psi} - \psi)^2}$$

En nuestra simulación hemos considerado que el ROV ha llegado al punto cuando la distancia absoluta es menor de 10 centímetros y el error en orientación es menor de 0.1 radianes. Cuando y solo cuando se cumplen estas dos condiciones, se considera que se ha alcanzado el punto y el proceso se repite con el siguiente punto de la trayectoria.

Todo este proceso se puede visualizar en el siguiente diagrama.

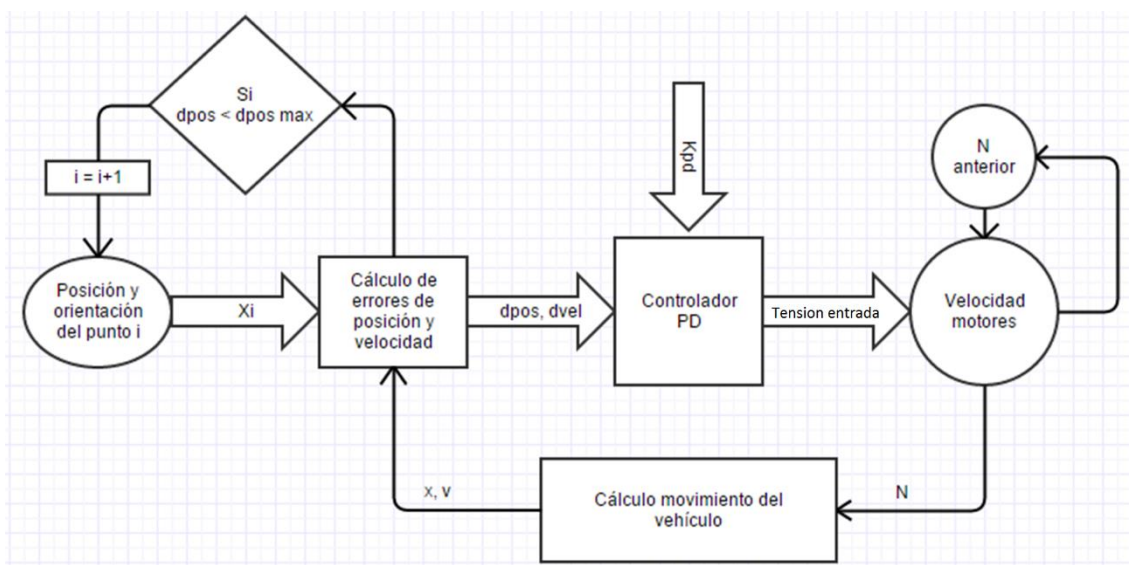


FIGURA 13: DIAGRAMA DE FLUJO DEL SISTEMA DE CÁLCULO DE LA VELOCIDAD DE LOS MOTORES EN CADA INSTANTE CON EL CONTROLADOR PROPORCIONAL DERIVATIVO.

El proceso de cálculo del control de trayectoria, así como los otros dos tipos de control se incluyen en el programa de C++ en los archivos de encabezado “Input.h”, “Input1.h”, “Input2.h” e “Input3.h”.

Referencias: [32]

6. Simulación

La simulación de este modelo se realizó primeramente en Matlab, y se comprobó que los datos obtenidos eran coherentes, para luego realizar la programación en un lenguaje más potente, C++, en el que la visualización de la simulación es mucho más detallada.

Nos vamos a centrar por tanto en la descripción del programa realizado en C++. Este lenguaje de programación tiene una biblioteca básica muy reducida, y para la realización de aplicaciones complejas es necesaria la instalación de bibliotecas externas. Para la realización de nuestro programa hemos necesitado la instalación de dos bibliotecas, Eigen para la realización de operaciones matemáticas complejas y OpenSceneGraph para la representación visual del ROV y de su entorno. En el anexo 1 se explica con detalle como instalar y hacer funcionar estas bibliotecas en un ordenador.

6.1. Entrada de datos

Este programa se ha diseñado de tal manera que se pueda utilizar no solo para este específico modelo de ROV, sino para todos los modelos que tengan cierta similitud con este; mismo número de hélices. La gran ventaja de este estilo de programación es el de poder visualizar el comportamiento del ROV para diferentes configuraciones, cambiando la distribución de los pesos añadidos, por ejemplo. Los datos que se han utilizado en esta simulación y que son obtenidos, en su mayor parte, de los cálculos realizados en capítulos anteriores, son los que se exponen en la siguiente tabla, pero como he explicado, estos se pueden cambiar al antojo del usuario.

Parámetro	Símbolo	Magnitud
Gravedad	g	9.81 m/s ²
Densidad agua	rho	1030 kg/m ³
Posición hélices	Hpos	$\begin{pmatrix} 0 & 0 & 0.155 & -0.155 & 0 & 0 \\ 0.155 & -0.155 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.155 & -0.155 \end{pmatrix} m$
Coordenadas CDG	(xG, yG, zG)	(0.00148, 0.00148, -0.02876) m
Peso	W	92.09 N
Empuje	B	93.25 N
Inercia	(Ix, Iy, Iz)	(0.10212, 0.10212, 0.07782) kg/m ²
Velocidad máxima de las hélices	mrxrpm	3800 rpm
Coefficiente de fuerza de las hélices	Tnn	6.5·10 ⁻⁵ N·s ²

TABLA 9: CARACTERÍSTICAS BÁSICAS DEL VEHÍCULO Y SU ENTORNO

Coeficientes hidrodinámicos		
Parámetro	Símbolo	Magnitud
Masa añadida según X	Xudot	-5.861 kg
Masa añadida según Y	Yudot	-5.861 kg
Masa añadida según Z	Zudot	-5.861 kg

Masa añadida según K	Kudot	-0.01170 kg/m ²
Masa añadida según M	Mudot	-0.01170 kg/m ²
Masa añadida según N	Nudot	-0.01170 kg/m ²
Amortiguamiento según X	(Xuu, Xu)	(-21 kg/m, -10 kg/s)
Amortiguamiento según Y	(Yuu, Yu)	(-21 kg/m, -10 kg/s)
Amortiguamiento según Z	(Zuu, Zu)	(-21 kg/m, -10 kg/s)
Amortiguamiento según K	(Kuu, Ku)	(-3.2 kg/m, -1.6 kg/s)
Amortiguamiento según M	(Muu, Mu)	(-3.2 kg/m, -1.6 kg/s)
Amortiguamiento según N	(Nuu, Nu)	(-3.2 kg/m, -1.6 kg/s)

TABLA 10: COEFICIENTES HIDRODINÁMICOS DEL VEHÍCULO

Estos parámetros se encuentran a nuestra disposición en el archivo de encabezado denominado “Datos.h”.

6.2. Cálculos

Los cálculos más importantes de la simulación son los que determinan la posición y orientación del vehículo y se basan en el modelo matemático del vehículo. Pero aparte de esto, también es necesario calcular otros términos que se detallan a continuación.

- **Matriz de rotación** en cada instante: Es necesaria para transformar vectores de coordenadas locales a coordenadas globales, sirve, entre otras cosas, para dibujar las hélices del vehículo en la posición correcta independientemente de la orientación del vehículo.
- **Velocidad de giro de los propulsores:** Para simular el comportamiento real de los propulsores se calcula la velocidad de giro a través de un sistema de ecuaciones en función de las características del propulsor y condiciones iniciales. También se comprueba que la velocidad de giro no supera la velocidad máxima permitida, y se acota en tal caso.
- **Fuerza ejercida por los propulsores:** Se calcula en cada instante la fuerza ejercida por los propulsores en función de la velocidad de giro de las hélices y la velocidad del flujo incidente. Para ello primero se transforma de revoluciones por minuto a radianes por segundo.
- **Momentos producidos por los propulsores:** Para conseguir un programa lo más versátil posible, que sea adaptable a modificaciones del modelo, la posición de las hélices son variables, lo que hace que el cálculo de los momentos que producen sea algo más complicada.
- **Posición de las hélices:** Para dotar de realismo a nuestra simulación, hemos calculado el movimiento de las mismas. De esta forma las hélices se mueven acorde a su velocidad de giro. Sin embargo, las altas velocidades que alcanzan estas hélices producen el efecto Aliasing, al ser mucho mayor que la velocidad de refresco, para evitarlo se ha reducido la velocidad de giro simulada de las hélices por un factor de 40.

Estos cálculos se encuentran a nuestra disposición en el archivo de encabezado denominado “Calculos.h”.

Referencias: [33]

6.3. Almacenamiento de datos

Para analizar los resultados de las simulaciones es conveniente almacenar los datos generados en un espacio de memoria preestablecido. Para guardar los datos con una organización correcta se han declarado una serie de matrices de dimensiones adecuadas para cada tipo de datos.

Los datos que varían con el tiempo necesitan grandes matrices con una fila por cada conjunto de datos en un instante de tiempo. Para definir estas matrices, es necesario, por tanto conocer el número de iteraciones que vamos a realizar en nuestra simulación. Siendo t_f el tiempo total de la simulación y siendo h el paso, periodo de tiempo entre iteraciones, tenemos que el número de iteraciones total, N , será igual al tiempo total dividido por el paso. En el caso de nuestras matrices variables con el tiempo, su número de filas será de $N+1$, ya que hay que añadir al número de iteraciones el estado inicial.

En la siguiente tabla se presentan algunas de las matrices más importantes con sus dimensiones correspondientes.

Matriz	Símbolo	Dimensiones
Posición y velocidad	M_{xv}	$N+1 \times 13$
Fuerzas	M_f	$N+1 \times 6$
Masas	M_{Mass}	6×6
Velocidad y aceleración	V_{der}	13×1
Rotación	Rot	3×3
Posición hélices	H_{pos}	3×6
Constantes control	KPD	6×4
Puntos trayectoria	$Tray$	$N_p^* \times 6$

TABLA 11: MATRICES UTILIZADAS Y SUS DIMENSIONES

* N_p es el número de puntos que utilizamos en el control de trayectoria por puntos

Todas estas matrices se han declarado en el archivo de encabezado denominado “Matrices.h” excepto la matriz de puntos de la trayectoria, que es declarada en el archivo “Trayectoria.h”.

Referencias: [33]

6.4. Visualización

La visualización de gráficos en 3D se ha realizado con la ayuda de la biblioteca de código abierto de OpenSceneGraph (OSG). Esto nos ha permitido introducir objetos en un entorno visual, cambiar la posición y orientación de dichos objetos en función de

los resultados de nuestra simulación y cambiar la posición del observador en dicho entorno.

Para la visualización del comportamiento del ROV es necesario introducir los modelos del cuerpo ROV y las hélices realizados en Rhinoceros, así mismo, también conviene importar un entorno marino, por el que se mueva el vehículo, para tener un punto de referencia. En el caso del control por trayectoria de puntos también se necesita un objeto con forma de flecha para indicar los puntos a los que debe desplazarse el vehículo.

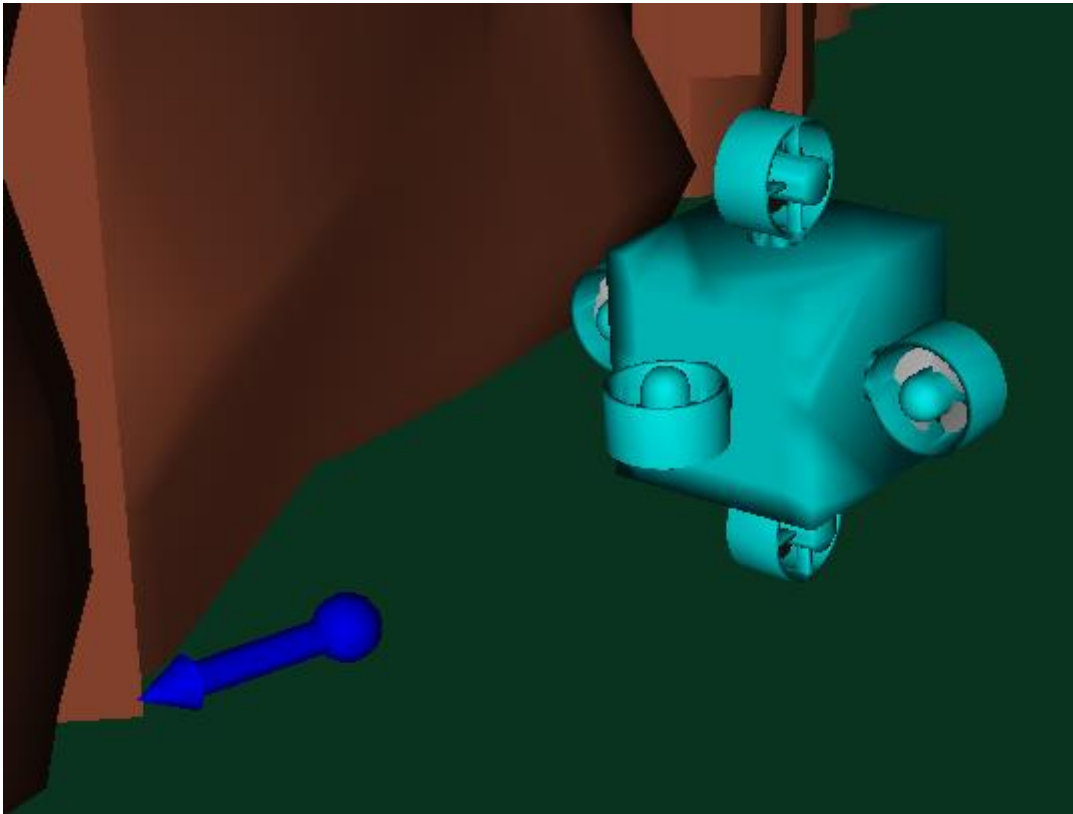


FIGURA 14: IMAGEN DE LA SIMULACIÓN OBTENIDA PARA EL CONTROL POR TRAYECTORIA CON CÁMARA FIJA.

Los modelos se han introducido en un formato de archivo OBJ, debido a su compatibilidad tanto con la biblioteca de C++ que realiza la visualización (OSG) como con el programa en el que hemos diseñado el vehículo (Rhinoceros), que es capaz de crear este tipo de archivos. El archivo OBJ correspondiente a las rocas que hemos añadido en el entorno ha sido obtenido en la página web TurboSquid.

Inicialmente se quiso asignar texturas a cada uno de los objetos, esto implicaría descargarse las texturas adecuadas para todos los objetos y escribir el código necesario para asignar cada textura a su objeto correspondiente. Esta tarea demostró ser más complicada de lo anticipado en un primer momento, debido a mi desconocimiento inicial sobre la biblioteca OSG y la poca documentación al respecto que pude encontrar, decidí desistir en esta tarea y sustituir las texturas por simples colores, esto hace que la

visualización no sea tan natural y realista como lo hubiera sido con las texturas. Implantar texturas es una tarea pendiente, que puede realizarse por otros alumnos que decidan continuar con este proyecto y quieran refinar mi simulación.

Se ha elegido dibujar los objetos de distintos colores, para dotar de algo de realismo a la simulación y para poder diferenciar mejor el vehículo de su entorno. Para conseguir implantar el color en los objetos se ha recurrido al archivo “CcolorVisitor.h” obtenido de la página web (<http://www.vis-sim.com/>) que permite cambiar el color de un nodo de forma simple. La introducción de este código permite asignar colores a cada nodo (objeto o conjunto de objetos) de manera sencilla.

Para producir el efecto de movimiento es necesario dibujar una y otra vez los objetos en posiciones ligeramente diferentes, en nuestro caso el tamaño de paso es de 0.02 segundos, por lo que se generaran cincuenta imágenes por segundo. Para ello es necesario posicionar y orientar los objetos móviles, el cuerpo del ROV y las hélices, en cada instante de tiempo.

La posición del observador en la simulación es muy importante para poder ver lo que está pasando, para ello, se han programado tres estilos de control de cámara diferentes:

- **Cámara fija:** La cámara está posicionada en unas coordenadas dadas y se orienta siempre hacia el centro del vehículo.
- **Cámara “on board”:** La cámara está montada sobre el vehículo, por lo que se mueve según este y está orientada en la dirección de avance, es decir, hacia proa.
- **Control de cámara dinámico:** Se puede indicar antes de la simulación la posición y orientación que se quiere que tenga la cámara en cada instante.

La programación de la visualización se ha realizado en los archivos de encabezado “Dibujo.h” y “Render.h” para implantar los objetos y “Camara.h” y “Camara1.h” para el control de la cámara.

Referencias: [34], [35], [36]

6.5. Proceso de iteración

Se recurre a la función principal, main(), para repetir los procesos de cálculo y visualización por cada iteración, así como para realizar el cálculo de parámetros que permanecerán constantes, como la matriz de masas o la posición relativa de las hélices.

Para que la visualización ocurra en tiempo real se ha introducido la función Sleep(t), donde t es el tiempo en milisegundos que debe esperar el programa entre una iteración y la siguiente una vez realizados los cálculos. Esta solución para la representación en tiempo real tiene el problema de que no tiene en cuenta el tiempo que tarda el programa en realizar los cálculos. Si los cálculos fueran instantáneos, el tiempo t en la función Sleep(t) sería igual al paso ($t = h$), pero esto no es así y hemos estimado que

por cada iteración el tiempo de cálculo es aproximadamente de 5 ms, en este caso el tiempo t deberá ser la diferencia entre el paso y el tiempo de cálculo.

$$t = h - t_c = 20 - 5 = 15 \text{ ms}$$

Aproximar el tiempo del cálculo no es una solución perfecta, porque, dependiendo del tipo de control elegido, el número de cálculos a realizar, y por tanto el tiempo de computación, varía de forma notable. En caso de elegir el control manual por teclado, este fallo se hace patente de forma clara, ya que, debido a la configuración del programa, mientras se está pulsando una tecla, el tiempo de computación es mayor que cuando no se pulsa ninguna. Así, nos podemos encontrar un caso en el que, al dejar de acelerar el vehículo en una dirección y dejar de pulsar la tecla correspondiente, el vehículo tenga la apariencia de acelerarse, cuando debería hacer lo contrario, lo que realmente sucede es que la disminución del tiempo de computación hace que el tiempo en la simulación vaya más rápido que el tiempo real, y dé la sensación de una mayor velocidad.

La solución ideal a este problema sería medir el tiempo de computación en cada iteración, lo cual es posible con la implantación de la biblioteca Boost. Desafortunadamente, no me ha sido posible implantar esta biblioteca en el programa por falta de tiempo.

La figura 15 muestra cómo se han implantado todas las funciones de las que hemos hablado anteriormente dentro de la función `main()`.

```
int main()
{
    helices();           //Posición y orientación de las hélices (Datos.h)
    matmasa();          //Calcula la matriz de masas
    matresh();          //Calcula la matriz de resistencia hidrodinámica
    matT100();          //Curva de los motores T-100
    valoresini();       //Posiciones y velocidades iniciales
    trayectoria();      //Puntos de la trayectoria y constantes CPD
    dibujo();           //Carga los objetos y ubica los objetos fijos (Terreno y rocas)

    for (int i=1; !viewer.done() && i<N ; i++) //Produce un ciclo hasta el final de la simulación (o pulsar esc)
    {
        input(i);       //Introduce el ancho de pulso del controlador
        T100();          //Obtiene las velocidades angulares a partir de la velocidad de giro
        calculos(i);    //Realiza el cálculo del modelo (Velocidades y fuerzas)
        camara(i);      //Indica los tipos de cámara utilizados en cada momento (Posición fija, On board...)
        render();        //Coloca el ROV y sus hélices en la ubicación correspondiente
        Sleep(15);      //Tiempo de computación aproximado (Tcomp = 5ms; Tsleep = h-Tcomp = 15ms)
                       //Utilizar librería boost para mayor precisión
    }
}
```

FIGURA 15: CÓDIGO DE LA FUNCIÓN PRINCIPAL MAIN() QUE AGRUPA EL RESTO DE FUNCIONES

Referencias: [33]

7. Resultados

Para comprobar la correcta resolución numérica de las ecuaciones del movimiento se ha decidido realizar tres ensayos, uno utilizando el control manual de los motores y dos con control de trayectoria. Para visualizar de forma sencilla el comportamiento mostrado por el vehículo, se han obtenido gráficas que representan las posiciones del vehículo durante el tiempo de simulación, así como la orientación, las velocidades y las fuerzas ejercidas sobre el vehículo en función del tiempo.

Se ha elegido realizar estos ejemplos en el programa de Matlab, ya que en él se pueden dibujar las gráficas con facilidad. Los resultados serán los mismos que los obtenidos en C++ debido al paralelismo entre ambos códigos.

7.1. Ensayo 1: Control manual

Para este ensayo se ha decidido asignar velocidades angulares de los diferentes propulsores en función del tiempo. El tiempo total de simulación es de veinte segundos y durante este tiempo manipulamos las velocidades angulares de los propulsores de tal forma que el vehículo se mueve en las tres direcciones espaciales y realiza un giro según psi (eje z). En la siguiente tabla se muestran cuáles son las velocidades angulares asignadas.

Tiempo(s)	N1 (rpm)	N2 (rpm)	N3 (rpm)	N4 (rpm)	N5 (rpm)	N6 (rpm)
$0 < t < 3$	2000	2000	1380	1380	0	0
$3 < t < 6$	0	0	1380	1380	2000	2000
$6 < t < 9$	0	0	2000	2000	0	0
$9 < t < 10$	-2000	2000	1380	1380	0	0
$10 < t < 13$	2000	2000	1380	1380	0	0
$13 < t < 20$	0	0	1380	1380	0	0

TABLA 12: COMANDOS DE VELOCIDAD DE LAS HÉLICES EN FUNCIÓN DEL TIEMPO

El vehículo parte de un estado de reposo en la posición del centro del sistema de coordenadas globales y con la orientación del mismo, por lo que las condiciones iniciales de posición y velocidad son las mostradas en la siguiente tabla.

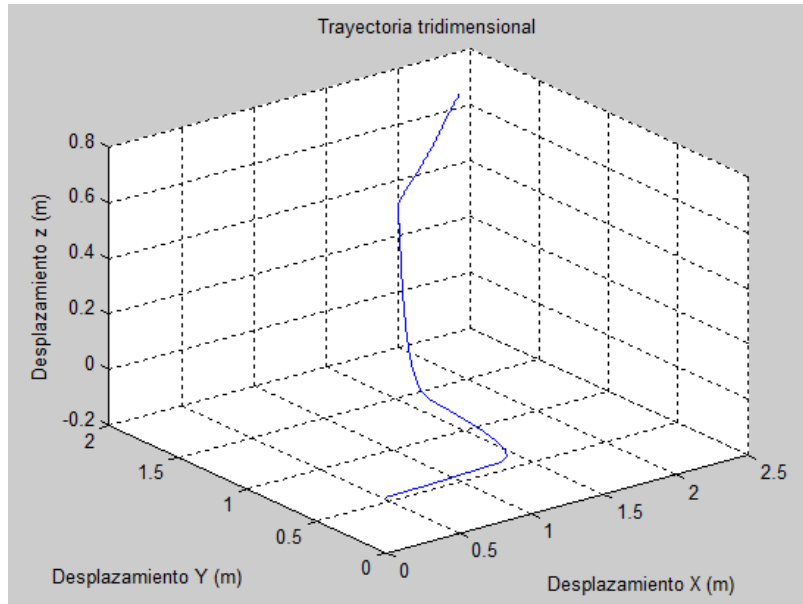
x	0	phi	0	u	0	p	0
y	0	theta	0	v	0	q	0
z	0	psi	0	w	0	r	0

TABLA 13: CONDICIONES INICIALES ENSAYO 1

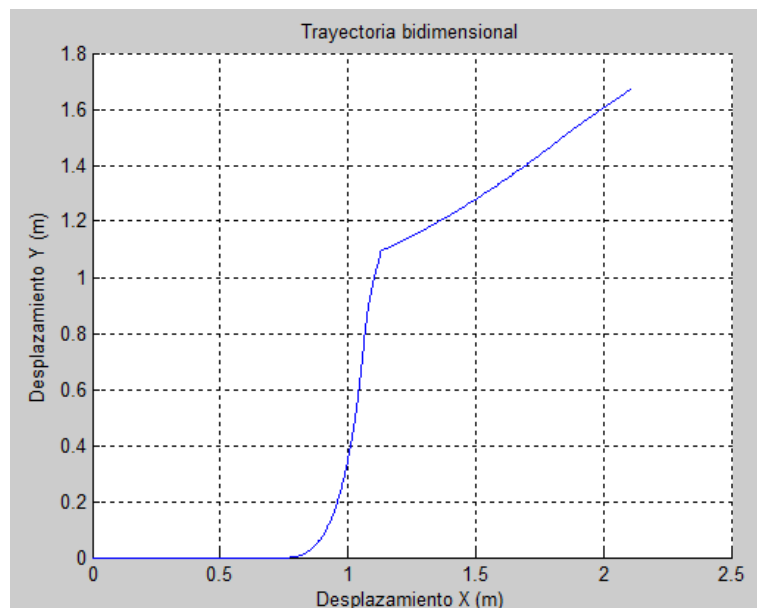
Con estos valores de entrada ya se puede realizar la simulación, y se obtienen las siguientes gráficas.

7.1.1. Trayectoria

La trayectoria seguida por el vehículo en tres dimensiones es difícil de mostrar con una imagen bidimensional, no obstante, se intentará mostrar las características de esta trayectoria en la siguiente serie de gráficas, una gráfica en la que se muestran las tres coordenadas y dos gráficas en las que se muestran únicamente dos coordenadas en cada una, las coordenadas X-Y e X-Z respectivamente.

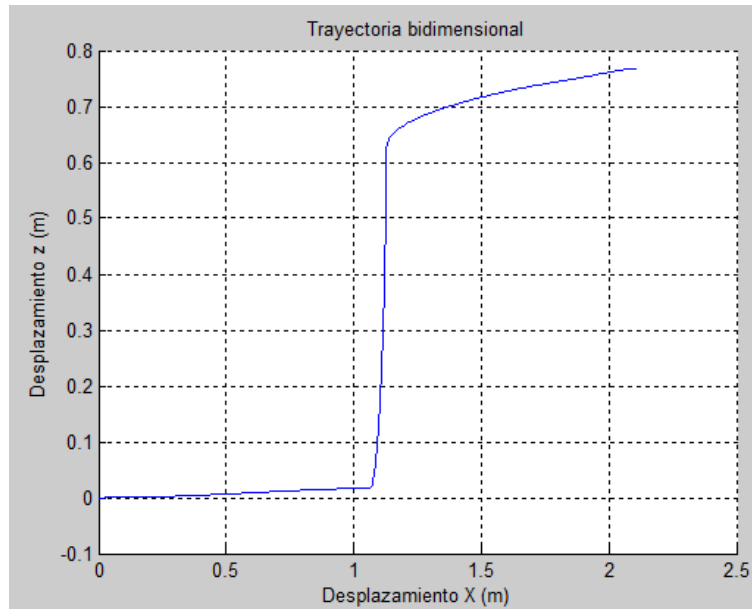


Esta gráfica muestra la trayectoria del vehículo según los tres ejes del sistema de coordenadas. Esta visualización puede parecer confusa, pero proporciona un esquema general del movimiento del vehículo, que se explicará en más claramente en las siguientes tablas dividiendo el movimiento según las coordenadas.



Para una mejor visualización del recorrido en el plano horizontal hemos obtenido la anterior gráfica según los ejes X e Y. Se puede observar que al empezar a

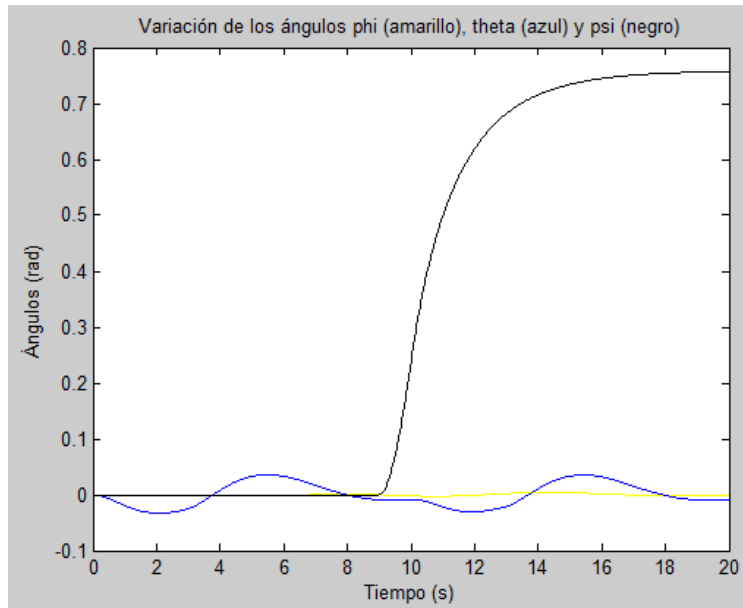
actuar los motores de deriva, en la dirección Y, los propulsores de avance se apagan, sin embargo, el vehículo se sigue moviendo durante un tiempo tras la parada, esto es debido a la inercia acumulada, este movimiento se disipa debido a la resistencia hidrodinámica. Al realizar el giro, el vehículo se coloca en un ángulo de aproximadamente 40° respecto al eje X.



En esta gráfica se representa la trayectoria según los ejes X y Z, y se puede observar claramente el descenso del vehículo producido por los motores de arfada. Cabe recordar que debido al sistema de coordenadas utilizado, el eje Z tiene el sentido de la fuerza de la gravedad, es decir, hacia abajo, por ello vemos el descenso del vehículo como un aumento en el valor positivo de la coordenada Z. Como en el caso anterior se pueden observar los efectos inerciales y de resistencia hidrodinámica tras el frenado de los motores de arfada.

7.1.2. Orientación

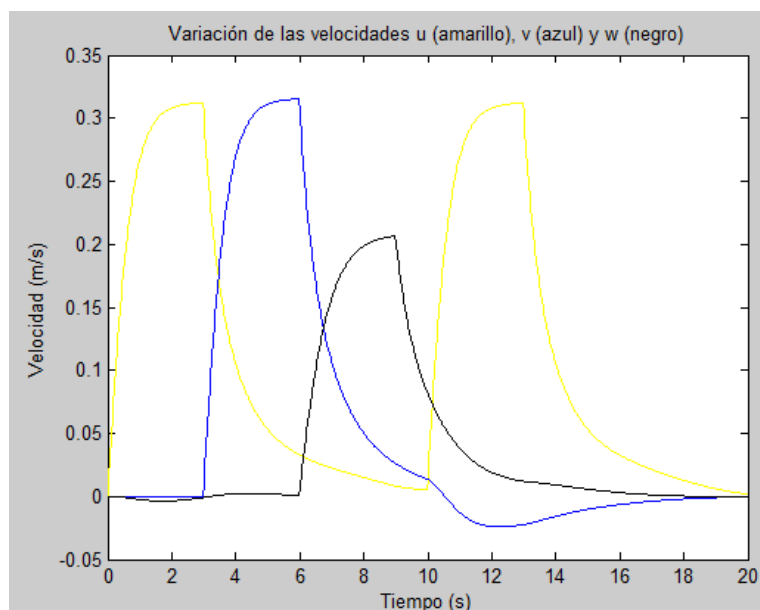
Para observar las orientaciones que ha adquirido el vehículo a lo largo de la simulación, en la siguiente gráfica mostramos el valor de los tres ángulos según las coordenadas cartesianas, phi, theta y psi, en función del tiempo.



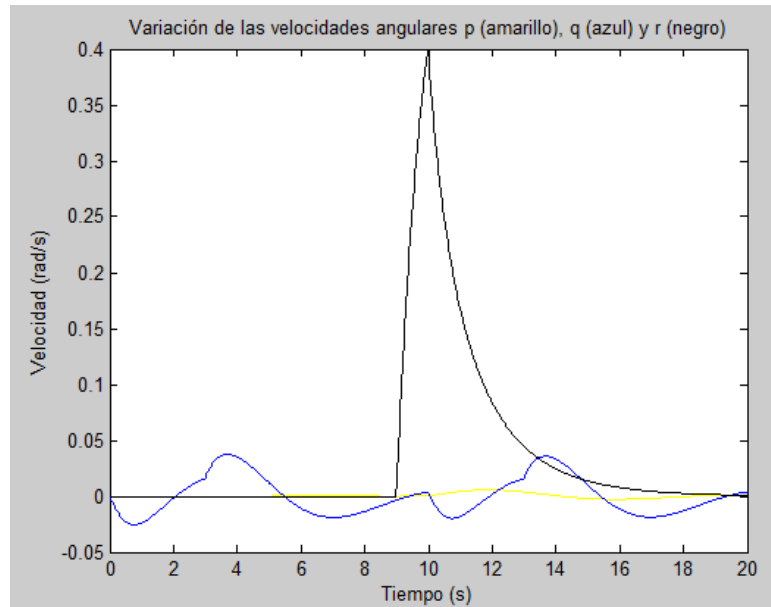
En esta gráfica se puede observar como en el segundo 9 de la simulación comienza a aumentar el ángulo de guiñada, al activarse los motores de avance en sentidos opuestos, hasta que alcanza un ángulo de 0.75 radianes, 43° . El ángulo phi se mantiene prácticamente constante, mientras que el ángulo de cabeceo, theta, sufre oscilaciones durante todo el ensayo, estas son debidas al hecho de que el centro de gravedad se sitúe por debajo del centro de carena, produciendo un par según el eje Y al producirse aceleraciones lineales en el eje X.

7.1.3. Velocidades

Las velocidades, tanto lineales como angulares, experimentadas por el vehículo se pueden observar en las siguientes gráficas, estas nos ofrecen información sobre la velocidad máxima y la amortiguación producida.



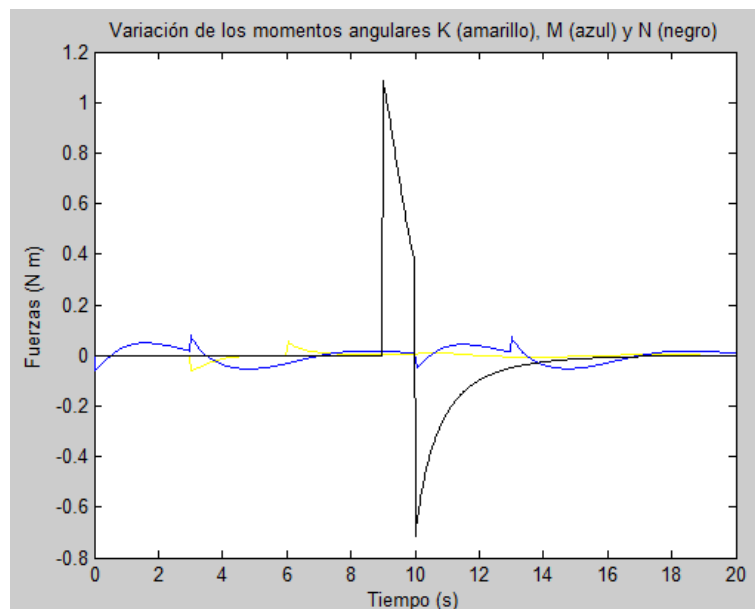
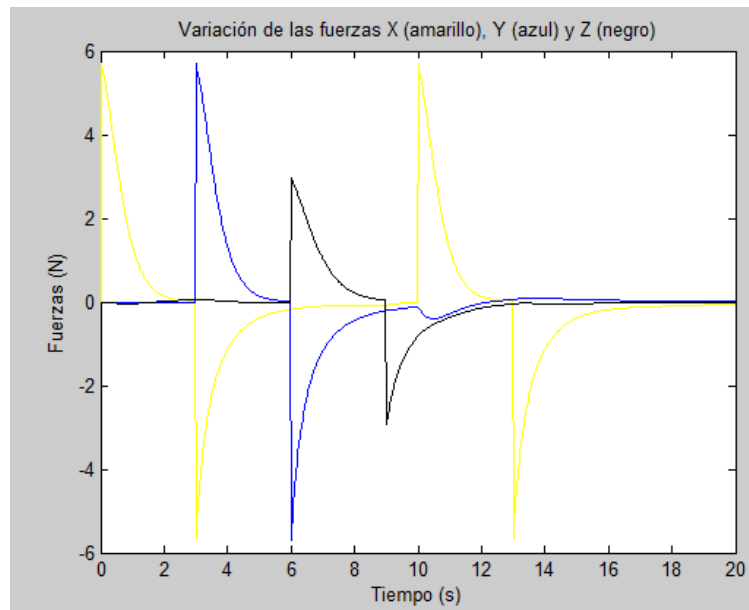
Las velocidades lineales máximas en las direcciones de avance y deriva son de 0.3 m/s, mientras que la velocidad máxima de arfada es de 0.2 m/s. Esto es debido a que los motores de arfada deben vencer a la fuerza de flotación durante el descenso, ya que nuestro vehículo tiene una flotabilidad ligeramente positiva. En este ensayo no se puede observar, pero es de suponer que la velocidad máxima de ascenso se superior a los 0.3 m/s, debido a la acción favorable de la flotabilidad.



En este gráfico se muestran las variaciones en las velocidades angulares, y, salvo algunas variaciones en las velocidades p y q, lo más destacable es el pico causado por el giro producido en el segundo 9, en el que se alcanza una velocidad de giro de 0.4 rad/s, no podemos considerar esta velocidad como la máxima, ya que el movimiento apenas se había amortiguado cuando se dejó de actuar sobre los motores que producían el giro. No es muy importante conocer esta velocidad máxima ya que en los movimientos de giro se busca más la precisión que la rapidez en la ejecución.

7.1.4. Fuerzas

A continuación se exponen las fuerzas y momentos que actúan sobre el vehículo durante la simulación. Estas fuerzas son debidas principalmente a una combinación de las fuerzas ejercidas por los propulsores y las fuerzas de resistencia hidrodinámica que experimenta el vehículo al adquirir una velocidad suficiente.



7.2. Ensayo 2: Control por trayectoria

Para este ensayo se han asignado una serie de puntos que conforman la trayectoria de un giro en el plano horizontal. El vehículo debe ir alcanzando estos puntos en orden, con la orientación indicada en cada uno de ellos, para ello se dispone del controlador proporcional derivativo que explicamos en el capítulo 5. El tiempo total de simulación es de sesenta segundos y durante este tiempo el controlador debe tomar las acciones necesarias que le permitan alcanzar los puntos objetivo. En la siguiente tabla se muestran cuáles son los puntos de la trayectoria asignados para este ensayo.

Punto	x	y	z	phi	Theta	psi
1	0	0	0	0	0	0
2	2200	0	0	0	0	0

3	2800	150	0	0	0	0.52
4	3200	500	0	0	0	1.05
5	3400	1000	0	0	0	1.57
6	3400	2700	0	0	0	1.57

TABLA 14: PUNTOS DE LA TRAYECTORIA DEL ENSAYO 2

El vehículo parte de un estado de reposo en la posición del centro del sistema de coordenadas globales y con la orientación del mismo, por lo que las condiciones iniciales de posición y velocidad son las mostradas en la siguiente tabla.

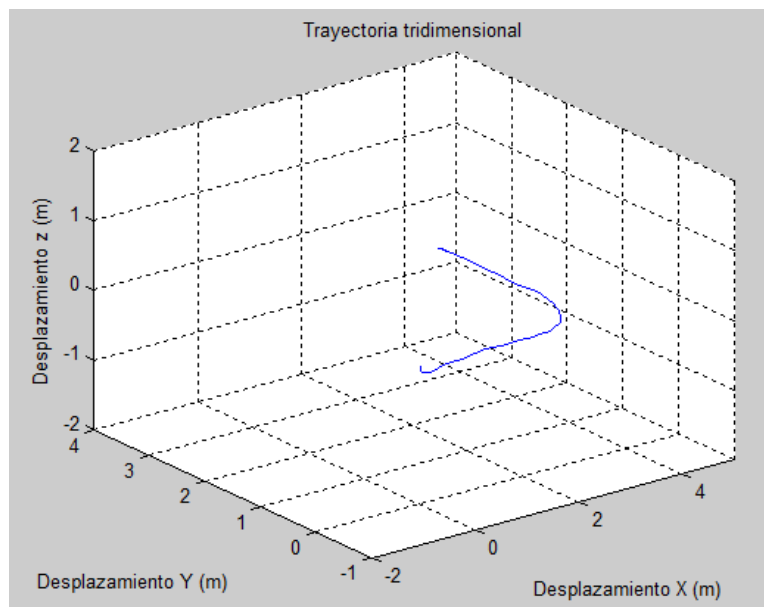
x	0	phi	0	u	0	p	0
y	0	theta	0	v	0	q	0
z	0	psi	0	w	0	r	0

TABLA 15: CONDICIONES INICIALES DEL ENSAYO 2

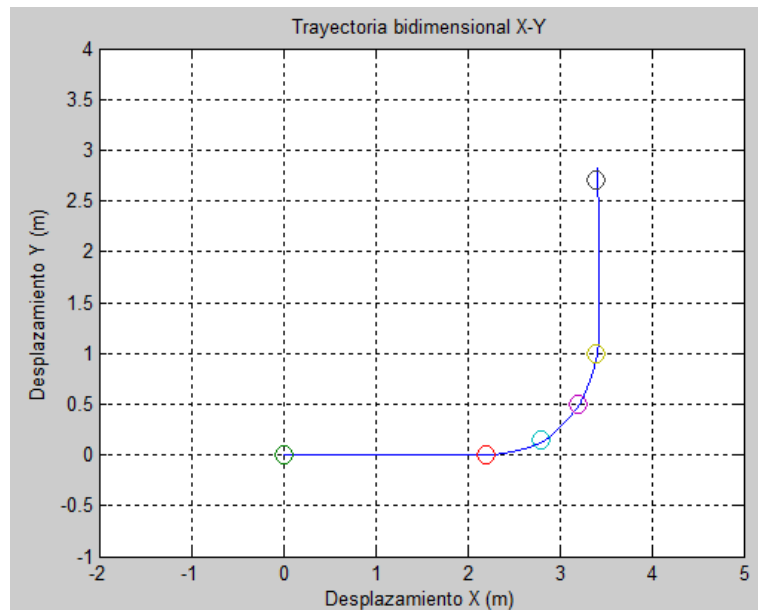
Con estos valores de entrada ya se puede realizar la simulación, y se obtienen las siguientes gráficas.

7.2.1. Trayectoria

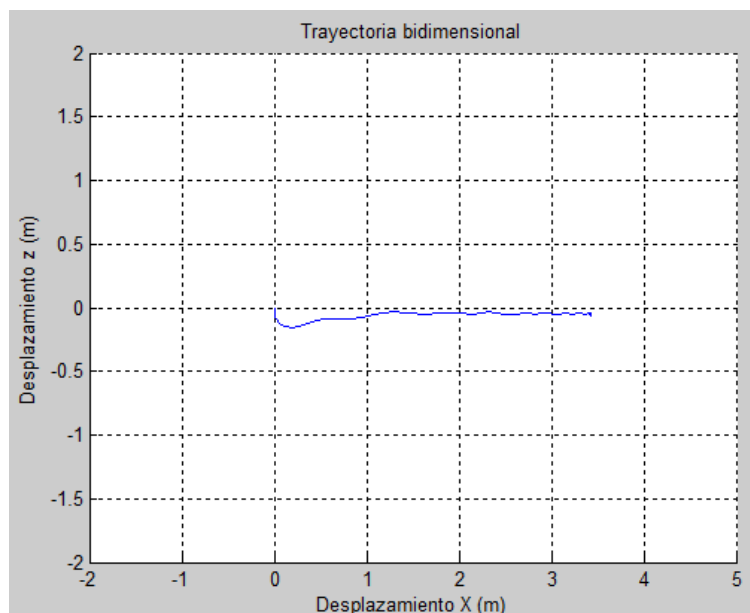
La trayectoria seguida por el vehículo en tres dimensiones es difícil de mostrar con una imagen bidimensional, no obstante, se intentará mostrar las características de esta trayectoria en la siguiente serie de gráficas, una gráfica en la que se muestran las tres coordenadas, otra gráficas en las que se muestran únicamente las coordenadas X-Y indicando los puntos objetivo y una última gráfica en el plano X-Z.



Esta gráfica muestra la trayectoria del vehículo según los tres ejes del sistema de coordenadas. No se producen apenas desplazamiento en el eje Z por lo que la mejor manera de visualizar esta trayectoria será según el plano X-Y que es donde pretendemos que se produzca el movimiento ya que todos los puntos de la trayectoria tienen la coordenada Z igual a 0.



Para una mejor visualización del recorrido en el plano horizontal hemos obtenido la anterior gráfica según los ejes X e Y. Podemos observar como el vehículo atraviesa todos los puntos con un margen error suficiente, cada círculo en el gráfico corresponde con el máximo error en distancia sobre la posición del punto, de diez centímetros.

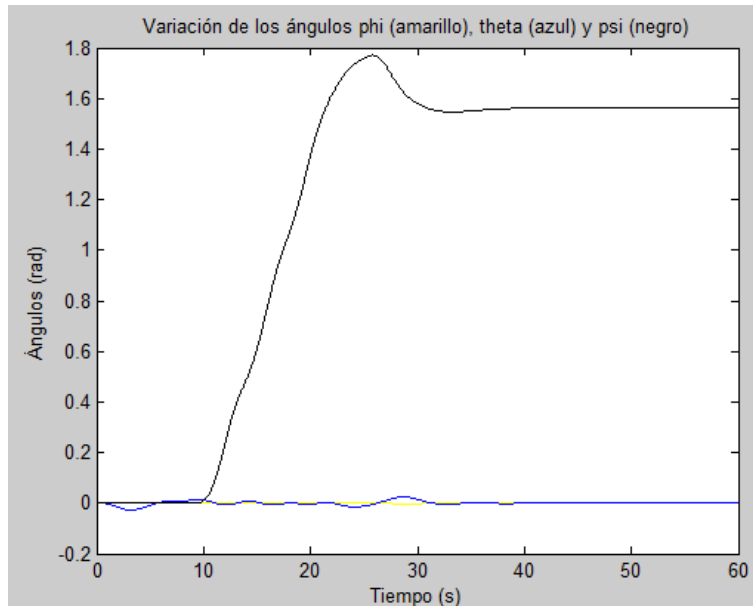


En esta gráfica se representa la trayectoria según los ejes X y Z, y se puede observar como al comienzo de la simulación, en $x=0$, se produce un ascenso significativo del vehículo, esto se debe al hecho de que los motores de arfada están parados inicialmente, y tardan cierto tiempo en arrancar, por lo que, durante ese tiempo, la fuerza de flotación actúa sin oposición. La fuerza de flotación es también la responsable de la ligera diferencia entre la trayectoria real y el plano horizontal $Z = 0$, al tratarse de una fuerza constante, el controlador no es capaz de contrarrestar por completo la fuerza de

flotación, y produce la ligera oscilación que se observa, intentando alcanzar la posición en z necesaria sin llegar a alcanzarla nunca.

7.1.2. Orientación

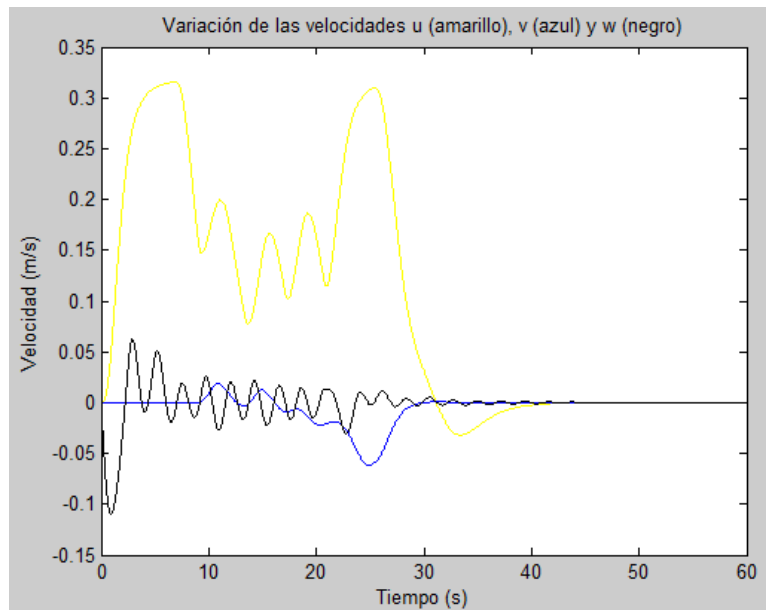
En la siguiente gráfica mostramos el valor de los tres ángulos según las coordenadas cartesianas, phi, theta y psi, en función del tiempo.



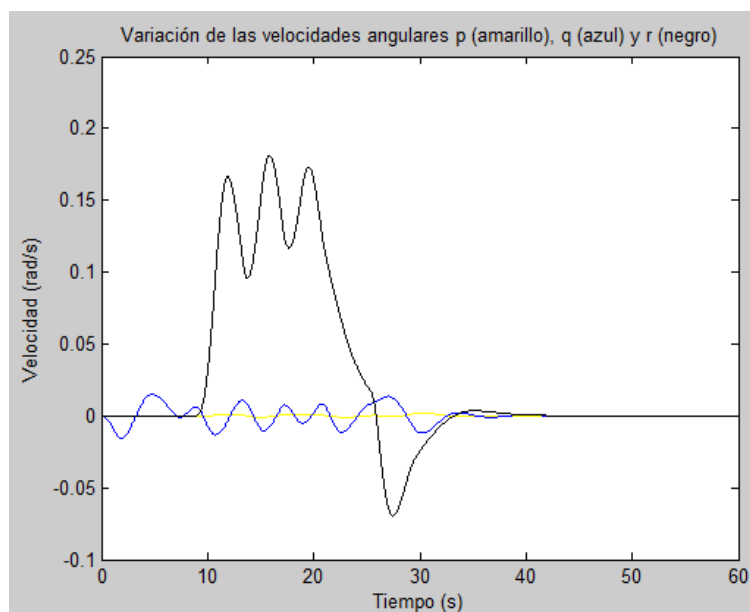
Vemos cómo se produce un giro de arfada aproximadamente constante, entre los tiempos 10 y 25 de la simulación. Una vez producido el giro necesario existe un pequeño sobregiro que se corrige posteriormente, estabilizándose finalmente el ángulo psi en 1.57 radianes, tal como se indicó en los dos últimos puntos de la trayectoria.

7.1.3. Velocidades

Las velocidades, tanto lineales como angulares, experimentadas por el vehículo se pueden observar en las siguientes gráficas, estas nos ofrecen información sobre la velocidad máxima y la amortiguación producida.

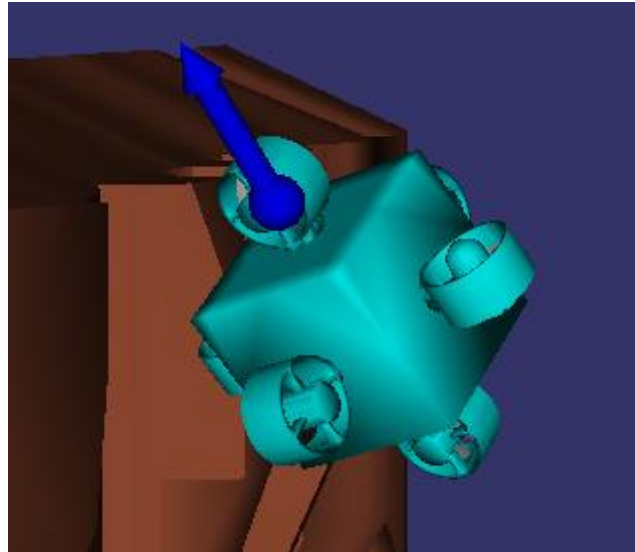


Las velocidad lineal máxima en la dirección de avance se encuentra, al igual que en el ensayo 1 en 0.3 m/s. Se puede observar una gran variabilidad en las tres velocidades lineales siendo la velocidad de avance la predominante, y actuando la velocidad de deriva en momentos puntuales. La velocidad de arfada muestra oscilaciones de pequeña amplitud pero de gran frecuencia, esto es debido a que en el diseño del controlador se eligieron valores para los parámetros proporcional y derivativo mayores en el caso de los motores de arfada que en el resto, esta medida se tomó para contrarrestar con más eficacia la fuerza de flotación. Aumentar estos parámetros produce una reducción en la distancia z entre el punto objetivo y el punto por el que pasa la trayectoria real, mientras que aumenta la amplitud y frecuencia de las oscilaciones. Finalmente se llegó a una solución de compromiso en la que se intentaron minimizar ambos efectos, pero no se lograron eliminar por completo.



En este gráfico se muestran las variaciones en las velocidades angulares en función del tiempo, lo más destacable es el triple pico causado por el giro según ψ , en el que se alcanza una velocidad media de 0.15 rad/s, la cual es inferior a la velocidad máxima, ya que no es necesaria una velocidad mayor para realizar esta maniobra. Las variaciones producidas en las velocidades p y q son muy inferiores a las de r , y pueden ser explicadas, como en el ensayo 1, por la diferencia de posición entre el centro de gravedad y el centro de carena.

7.3. Ensayo 3: Movimiento con orientación



Para este ensayo se intenta probar la capacidad del vehículo para alcanzar cierta posición angular y mantenerla durante la trayectoria. Para ello se han asignado una serie de puntos que conforman la trayectoria tridimensional y se ha elegido una cierta orientación del vehículo. Como en el ensayo anterior, el vehículo debe ir alcanzando estos puntos en orden con la orientación correspondiente. En la siguiente tabla se muestran cuáles son los puntos de la trayectoria asignados para este ensayo.

Punto	x	y	z	phi	theta	psi
1	0	0	0	0	0	0
2	500	100	-200	0.5	1	0.8
3	1000	200	-400	0.5	1	0.8
4	1500	200	-600	0.5	1	0.8
5	2000	100	-800	0.5	1	0.8
6	2500	0	-1000	0.5	1	0.8

TABLA 16: PUNTOS DE LA TRAYECTORIA DEL ENSAYO 3

El vehículo parte de un estado de reposo en la posición del centro del sistema de coordenadas globales y con la orientación del mismo, por lo que las condiciones iniciales de posición y velocidad son las mostradas en la siguiente tabla.

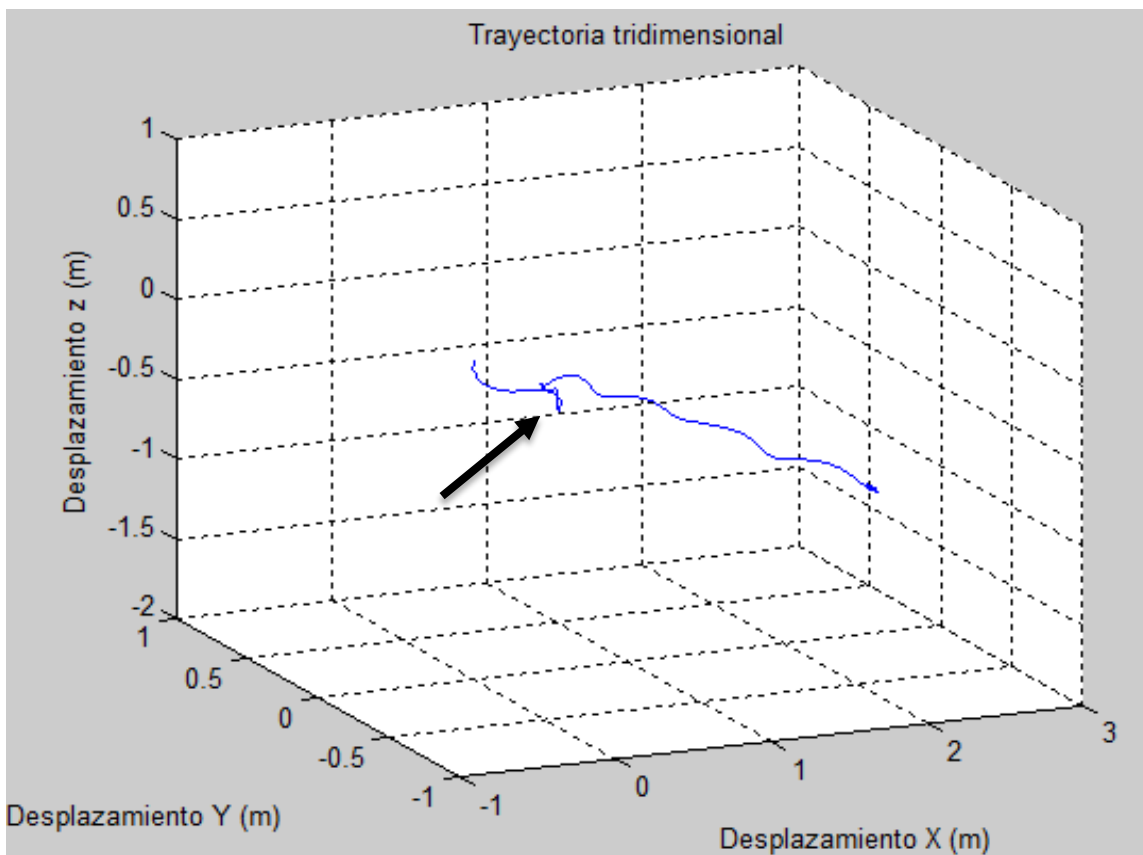
X	0	phi	0	u	0	p	0
----------	---	------------	---	----------	---	----------	---

Y	0	theta	0	v	0	q	0
Z	0	psi	0	w	0	r	0

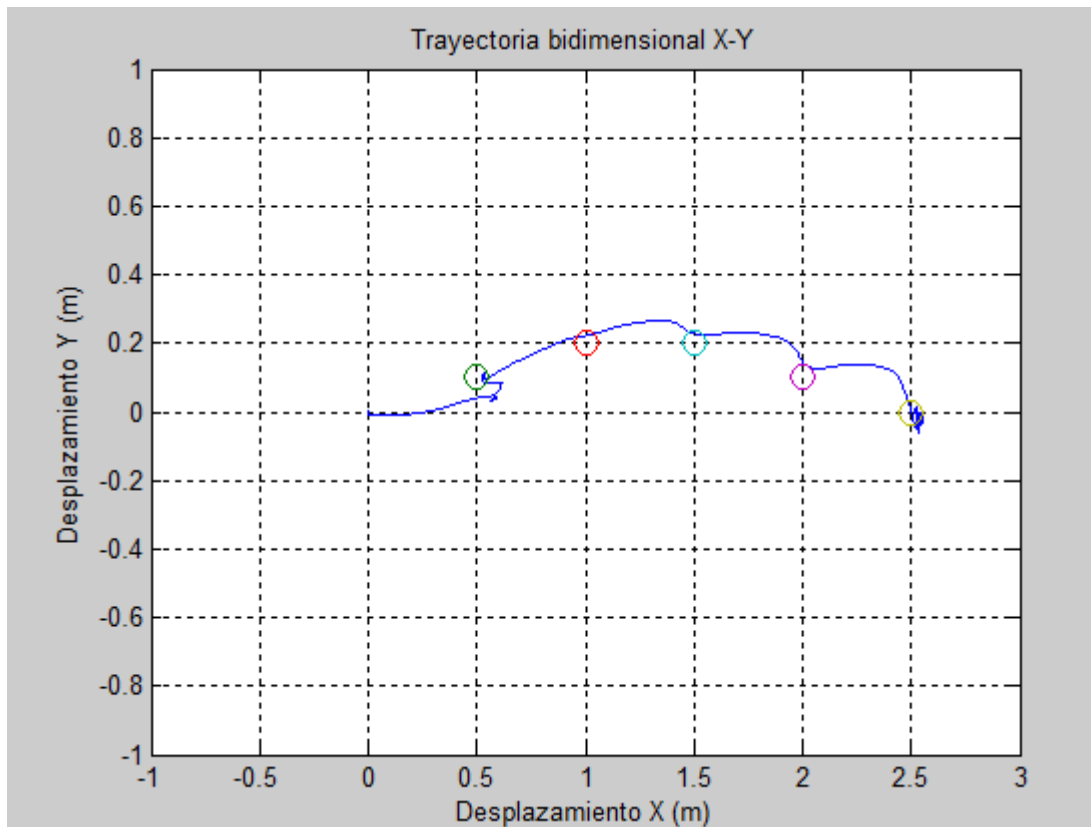
TABLA 17: CONDICIONES INICIALES DEL ENSAYO 3

Con estos valores de entrada ya se puede realizar la simulación, y se obtienen las siguientes gráficas.

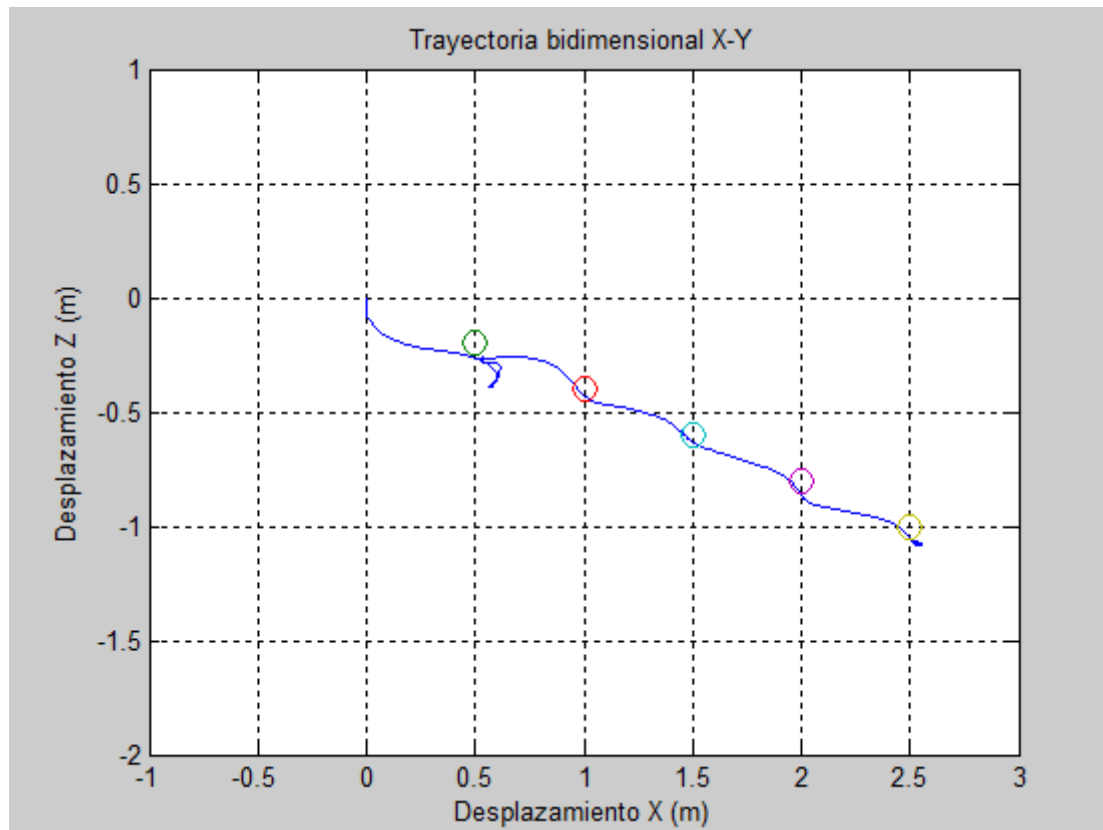
7.2.1. Trayectoria



Esta gráfica muestra la trayectoria del vehículo según los tres ejes del sistema de coordenadas. Se puede observar el avance en los ejes X y Z, el movimiento en el eje Y no se aprecia ya que es mucho menor. Al principio de la trayectoria, señalado con una flecha, aparece una pequeña vuelta realizada por el vehículo, esto indica que no se alcanzó la orientación deseada en un primer momento, y el vehículo tuvo que pararse unos segundos en la posición objetivo hasta que se orientó por completo. En el resto de puntos no existe este problema ya que, una vez alcanzada la orientación indicada, no tiene que cambiarse durante el resto de la trayectoria.



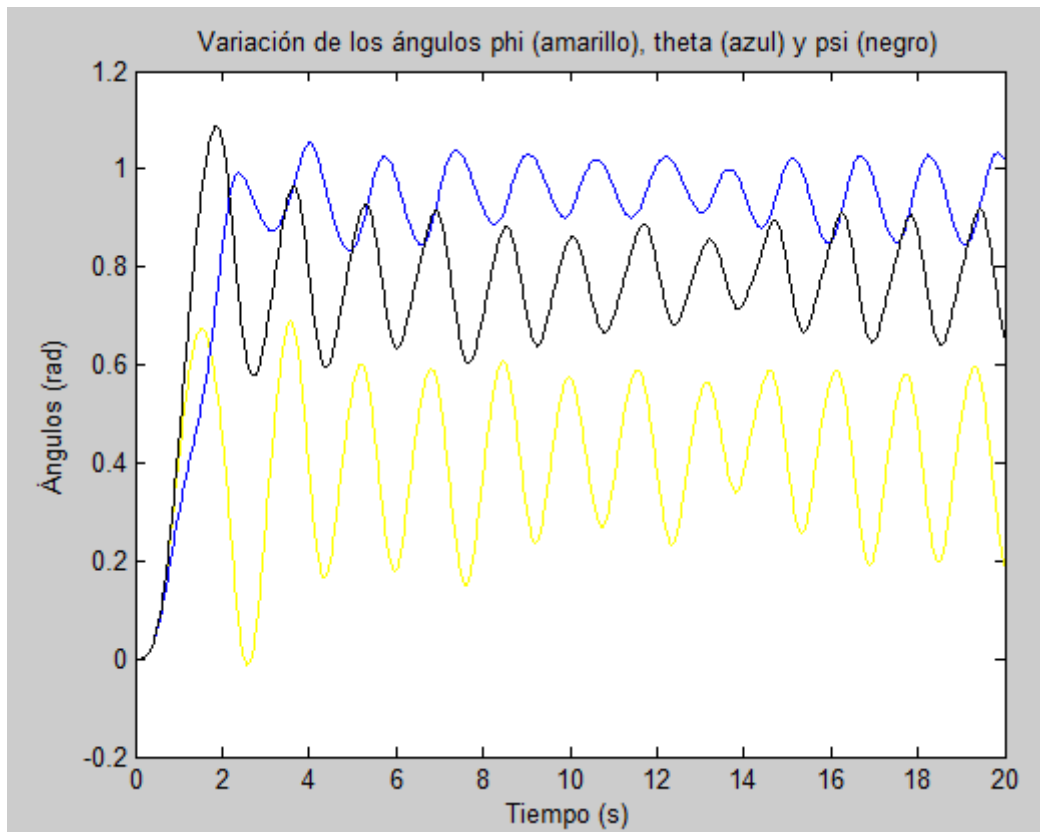
Para una mejor visualización del movimiento según Y hemos obtenido esta gráfica que muestra el plano XY. Podemos observar como el vehículo atraviesa todos los puntos con un margen error suficiente, cada círculo en el gráfico corresponde con el máximo error en distancia sobre la posición del punto, de diez centímetros.



En esta gráfica se representa la trayectoria en el plano XZ. Como se puede observar también en las dos gráficas anteriores, en este ensayo la trayectoria seguida no es tan lineal como en ensayos anteriores, a pesar de que la trayectoria es bastante simple. La culpa de esta trayectoria errática la tiene, como es de suponer, la particular orientación que hemos elegido, como veremos en el siguiente apartado.

7.1.2. Orientación

En la siguiente gráfica mostramos el valor de los tres ángulos según las coordenadas cartesianas, phi, theta y psi, en función del tiempo.

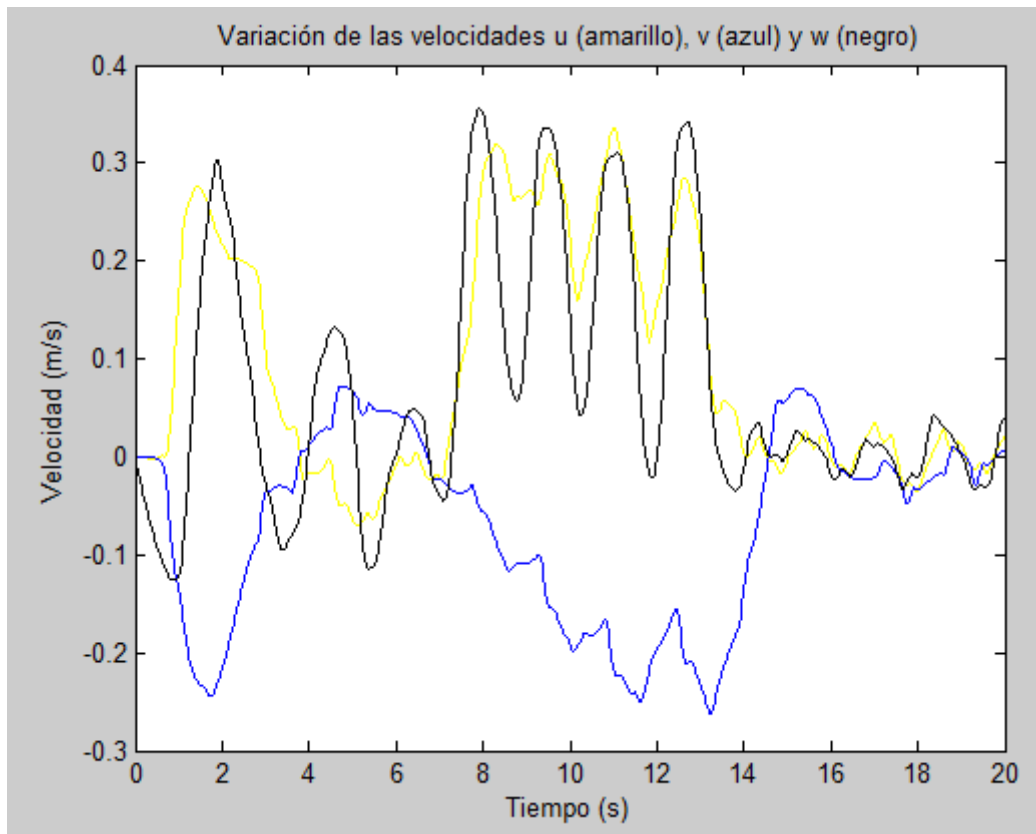


Las grandes oscilaciones producidas en los tres ángulos del sistema ofrecen la respuesta a la conducta errática del vehículo. Las hélices no son capaces de mantener la estabilidad en una posición tan intrincada. Las hélices tienen que realizar mayor fuerza para crear momento en esta posición, y a su vez generan fuerzas en direcciones no deseadas, que deben ser contrarrestadas por las otras hélices.

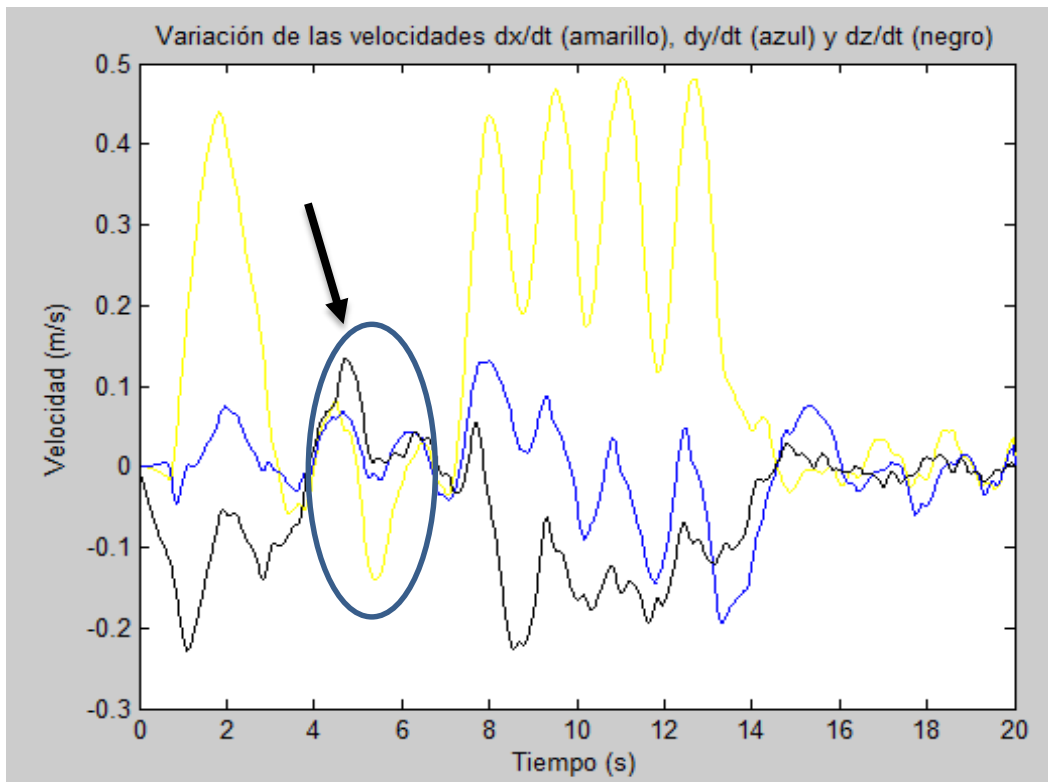
Con el sistema de control que hemos diseñado, desacoplado para las tres direcciones y los tres giros, cada par de hélices se encarga de una única dirección y un único giro, lo que impide que otras hélices colaboren en el movimiento, particularmente cuando, debido a la especial orientación, las hélices principales no tienen apenas capacidad de contribuir al movimiento. Por esta razón nuestro vehículo no es capaz de alcanzar ángulos de más de $\sim 80^\circ$ en phi y en theta. Los giros según psi no tienen esta restricción ya que el centro de gravedad se mantiene alineado, o casi alineado con el centro de carena, y no tiene que vencer importantes momentos.

7.1.3. Velocidades

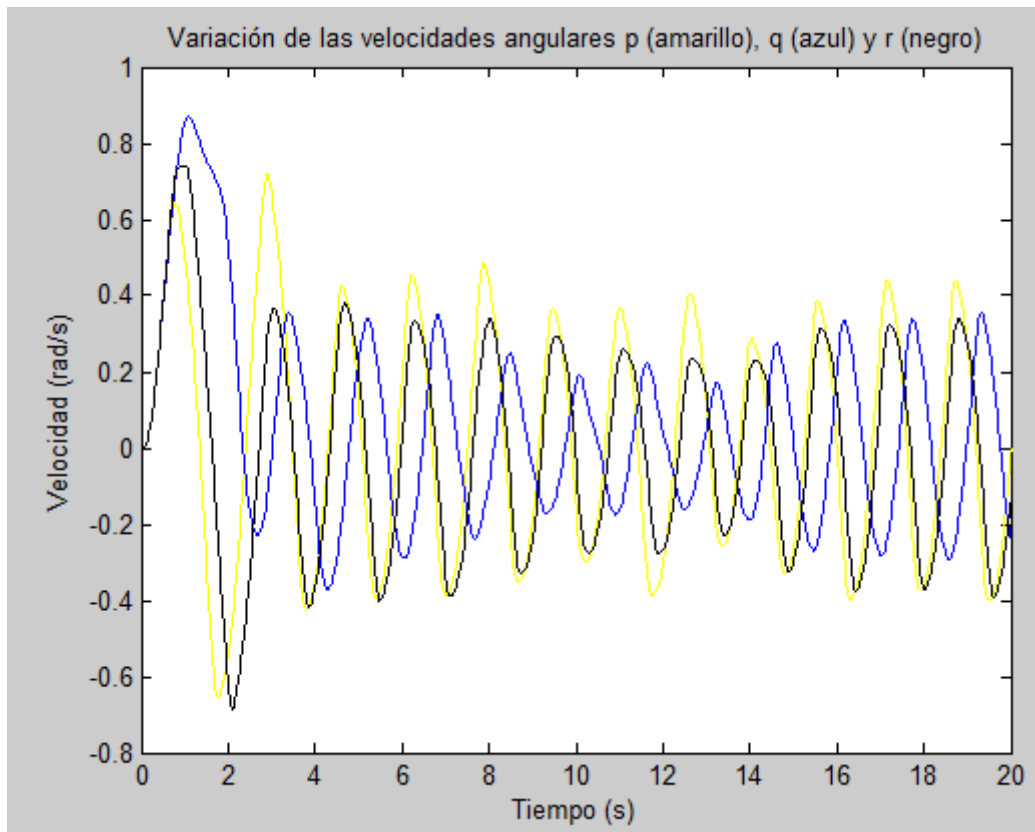
Las velocidades, tanto lineales como angulares, experimentadas por el vehículo se pueden observar en las siguientes gráficas.



Esta gráfica muestra la variación de las velocidades lineales en coordenadas locales. Es difícil sacar algo en claro de esta gráfica ya que la dirección real del movimiento depende de la orientación del vehículo. Para ello podemos recurrir a las velocidades lineales en coordenadas globales.



Esta gráfica de velocidades en coordenadas globales sigue siendo algo difícil de leer, debido a las grandes oscilaciones experimentadas en toda la trayectoria, pero se pueden sacar en claro varias conclusiones. La velocidad según X es más importante de las tres, y solo se reduce cuando se para el vehículo en el segundo punto, señalado con una flecha, y al final de la simulación. La velocidad según Y se mantiene entorno a cero, aunque con grandes oscilaciones. La velocidad según Z se mantiene negativa o nula excepto en el segundo punto.



De este gráfico no se puede obtener ninguna información, más que la constatación de las grandes oscilaciones producidas. Lo mismo ocurre con las gráficas de fuerzas y momentos.

8. Conclusiones

En este proyecto se ha realizado el diseño, el modelado, el control y la simulación de un vehículo submarino operado remotamente (ROV). Se ha creado un modelo en Rhinoceros del diseño del ROV, y un programa en C++, en el cual, se puede analizar y visualizar el comportamiento del vehículo diseñado aplicando sobre él un control, que puede ser manual o autónomo, a partir de la configuración de una trayectoria de puntos.

A partir de los resultados obtenidos de las simulaciones, se puede observar que el modelo de vehículo se comporta de una manera tolerable que se ajusta razonablemente a la realidad. Además, la acción de los sistemas de control desarrollados permiten guiar al vehículo con una estabilidad y precisión aceptable para este tipo de vehículos.

8.1. Trabajo realizado

Durante la realización del proyecto se han realizado gran cantidad de tareas con el objetivo de obtener el producto final. En la siguiente lista repasamos en que han consistido estos trabajos.

- Diseño de cuatro modelos diferentes del ROV, así como dos programas escritos en RhinoScript que permiten modificar las dimensiones básicas de los dos últimos modelos.
- Diseño de las hélices.
- Obtención del centro de gravedad, matriz de inercia, peso y volumen de los modelos de vehículo propuestos.
- Cálculo aproximado de los parámetros hidrodinámicos del modelo final de vehículo.
- Obtención y ajuste del modelo de comportamiento de los propulsores.
- Desarrollo de algoritmos para resolver las ecuaciones del comportamiento hidrodinámico del vehículo.
- Diseño del controlador proporcional derivativo, con la elección de los parámetros correspondientes.
- Programación en Matlab para comprobar las ecuaciones.
- Programación en C++ de todos los algoritmos y de la interfaz gráfica que nos permiten simular el comportamiento del ROV diseñado.

8.2. Mejoras y trabajos futuros

- **Diseño en detalle:** El diseño del ROV realizado en este proyecto se ha limitado a los aspectos generales del mismo, peso, dimensiones, posición de las hélices, etc. Estos datos son los necesarios para realizar el modelo matemático del vehículo, pero es necesario un diseño en detalle así como un diseño de conjunto del vehículo antes de realizar un prototipo del mismo.
- **Implementación de sensores y diseño del sistema electrónico:** Para poder realizar el control por trayectoria por puntos en el modelo real, es necesario conocer datos sobre la posición, orientación y velocidad del vehículo. Esto se puede alcanzar mediante sistemas de medición de inercia (IMU) y sistemas de posicionamiento global (GPS). Habría también que conectar estos dispositivos al cable conector para proporcionar potencia y transmitir los datos.
- **Fabricación:** El objetivo final es el de construir un prototipo funcional capaz de servir los propósitos para los que fue diseñado.

- **Obtención de parámetros reales:** Una vez construido el prototipo, sería positivo calcular los parámetros reales del vehículo (resistencia hidrodinámica, masa añadida, peso, empuje...) ya que los calculados en este proyecto son aproximaciones calculadas a partir de hipótesis que no se ajustan por completo al modelo real.
- **Control manual por Joystick:** Existen bibliotecas de código abierto que permiten realizar el control manual en tiempo real a través de un mando externo conectado al ordenador. Esto permitiría un control del vehículo más intuitivo y elegante que con el teclado.
- **Asignación de texturas:** Es preferible utilizar texturas en los diferentes objetos que forman parte de la simulación.
- **Cálculo del tiempo de computación:** Para mejorar la simulación en tiempo real es conveniente sustituir la función Sleep(t) por otra que tenga en cuenta el tiempo de computación. Para ello se recomiendan la biblioteca de Boost que es capaz de temporizar eventos.

A. Anexo 1: Instalación de bibliotecas e inclusión de directorios en C++

Para conseguir hacer funcionar el programa de C++ en un ordenador es necesario haber instalado previamente las bibliotecas que vamos a utilizar, **Eigen** y **OpenScenegraph**, así como tener en nuestro ordenador el archivo de cabecera **windows.h**.

Para la realización de este programa se ha utilizado el sistema operativo **Windows 8** y el programa de compilación **Microsoft Visual C++ 2010 Express**. Aunque el programa puede ser ejecutado desde cualquier tipo de ordenador, a continuación se va a explicar cómo instalar las bibliotecas necesarias para este sistema operativo y programa de compilación en concreto.

Eigen

La biblioteca Eigen es utilizada en nuestro programa para realizar cálculos con matrices. Se puede descargar la versión más reciente en la siguiente página web:

http://eigen.tuxfamily.org/index.php?title=Main_Page

La versión instalada en su momento fue Eigen 3.2.2 pero no debería importar que versión es utilizada.

Una vez descargado el archivo y descomprimido, se necesita vincular la biblioteca con el programa en el que se quiere utilizar. A continuación se detallan las instrucciones que se han de seguir para la instalación:

1. Abre el programa de C++ en el compilador.

Archivo > Abrir > Proyecto o solución... > Seleccionar el proyecto

-Se recomienda tener guardado el proyecto dentro de la carpeta del programa Visual Studio 2010 (Documents\Visual Studio 2010\Projects\)

2. Entra en propiedades del archivo.

Proyecto > Propiedades de (Nombre de archivo)

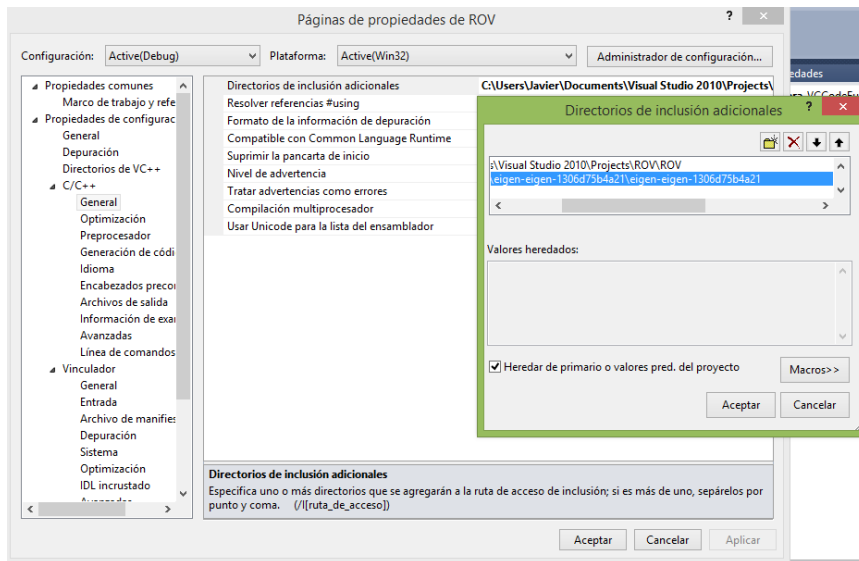
3. Edita los directorios de inclusión.

Propiedades de configuración > C/C++ > **General** > Directorios de inclusión adicionales

3. Introduce la dirección en la que has guardado la carpeta de Eigen.

Ej: C:\Users\<(Name)\Desktop\eigen-eigen-1306d75b4a21\eigen-eigen-1306d75b4a21

-Se ha de tener la precaución de escribir la dirección correctamente y que esta señale a las carpetas de Eigen que vamos a utilizar.



4. Compila el programa.

Ctrl+f5

Puede aparecer el siguiente error:

```
fatal error C1083: No se puede abrir el archivo incluir: 'Eigen/Dense': No such file or directory
```

Esto significa que la biblioteca no se ha añadido correctamente por lo que se ha de revisar el proceso realizado.

OpenSceneGraph

La biblioteca OpenSceneGraph (OSG) es utilizada en nuestro programa para realizar la visualización en tres dimensiones. Descargando los archivos binarios es la forma más sencilla de instalar estas bibliotecas y se pueden descargar en la siguiente página.

<http://openscenegraph.alphapixel.com/osg/downloads/free-openscenegraph-binary-downloads>

Se debe descargar tanto el archivo release como el debug, de la sección correspondiente (Visual Studio 10 y 32 o 64 bits dependiendo de la máquina en la que se quiera instalar).

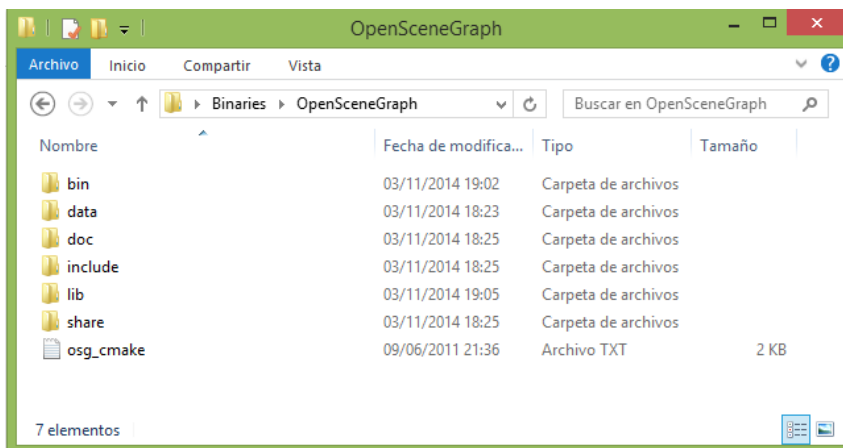
La versión instalada en su momento fue Eigen 2.8.5 pero no importa la versión utilizada.

Aparte de los archivos binarios, se deben instalar las dependencias, que se pueden encontrar en la siguiente página.

<http://openscenegraph.alphapixel.com/osg/downloads/openscenegraph-third-party-library-downloads>

Al igual que con los archivos binarios se descargan los archivos release y debug de la sección correspondiente.

Una vez descargados estos archivos, se descomprimen y se guardan todas las carpetas de cada uno de los archivos en **una sola carpeta**. El resultado debería ser parecido al que se muestra en la siguiente imagen.



Una vez hecho esto, se puede proceder a la instalación de las bibliotecas en el programa. Para ello se deben seguir los siguientes pasos:

1. Declarar variable de entorno.

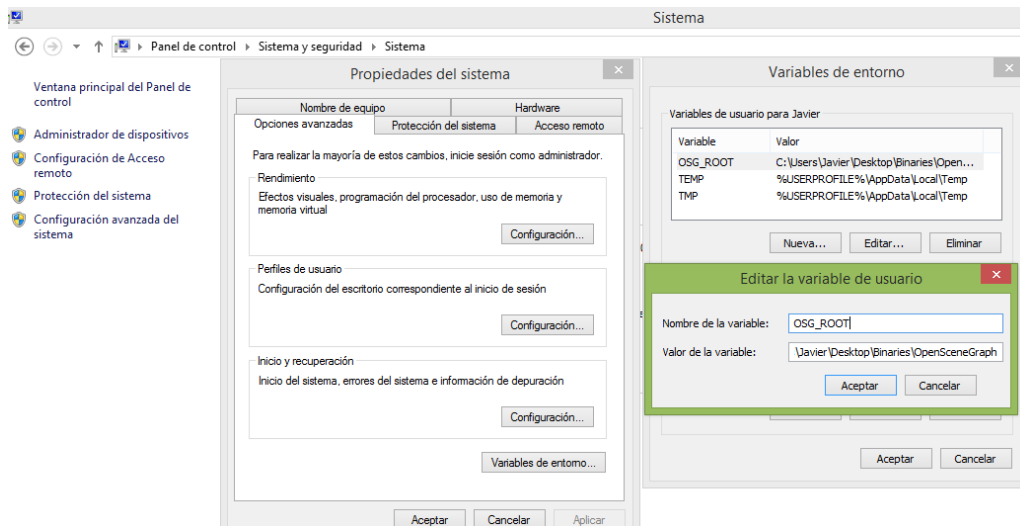
1.1. Entra en la configuración de las propiedades de entorno del ordenador.

Panel de control > Sistema y Seguridad > Sistema > Configuración avanzada del sistema > Variables de entorno...

1.2. Introduce variable.

Nombre de la variable: OSG_ROOT

Valor de la variable: Dirección de la carpeta donde se han guardado los archivos



2. Añadir directorios de inclusión.

2.1. Edita los directorios de inclusión.

Proyecto > Propiedades de (Nombre de archivo) > Propiedades de configuración > **C/C++** > **General** > Directorios de inclusión adicionales

2.2. Introduce la dirección de la carpeta include.

En nuestro caso, referenciándolo a la carpeta principal, sería:

`$(OSG_ROOT)/include`

3. Añadir directorios de bibliotecas adicionales.

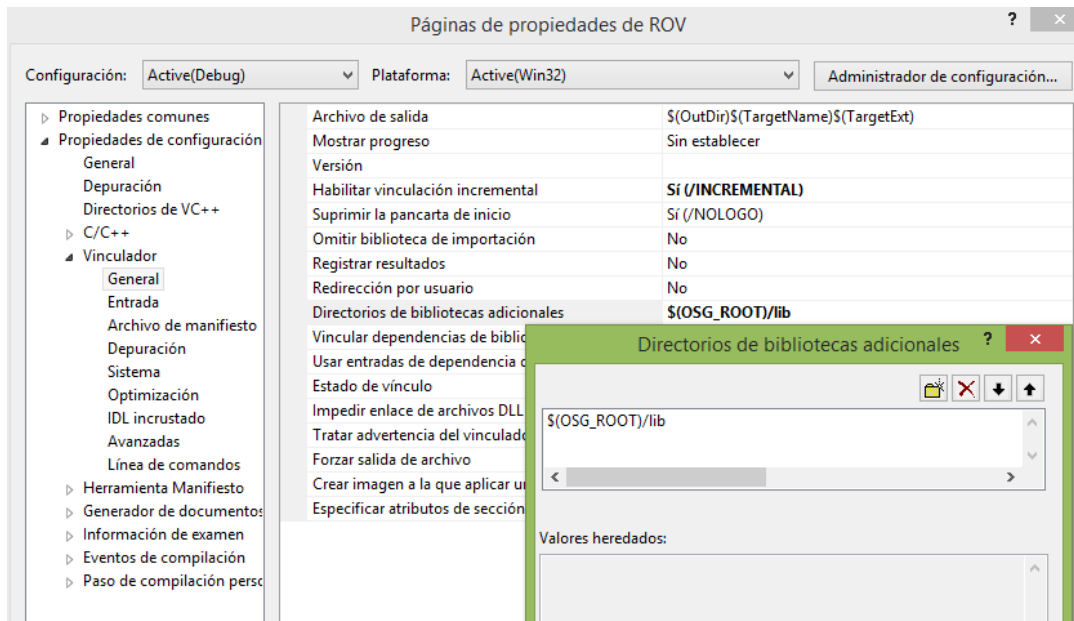
3.1. Edita los directorios de bibliotecas adicionales.

Proyecto > Propiedades de (Nombre de archivo) > Propiedades de configuración > **Vinculador** > **General** > Directorios de bibliotecas adicionales

3.2. Introduce la dirección de la carpeta lib.

En nuestro caso, referenciándolo a la carpeta principal, sería:

`$(OSG_ROOT)/lib`



4. Añadir dependencias adicionales.

4.1. Edita las dependencias adicionales.

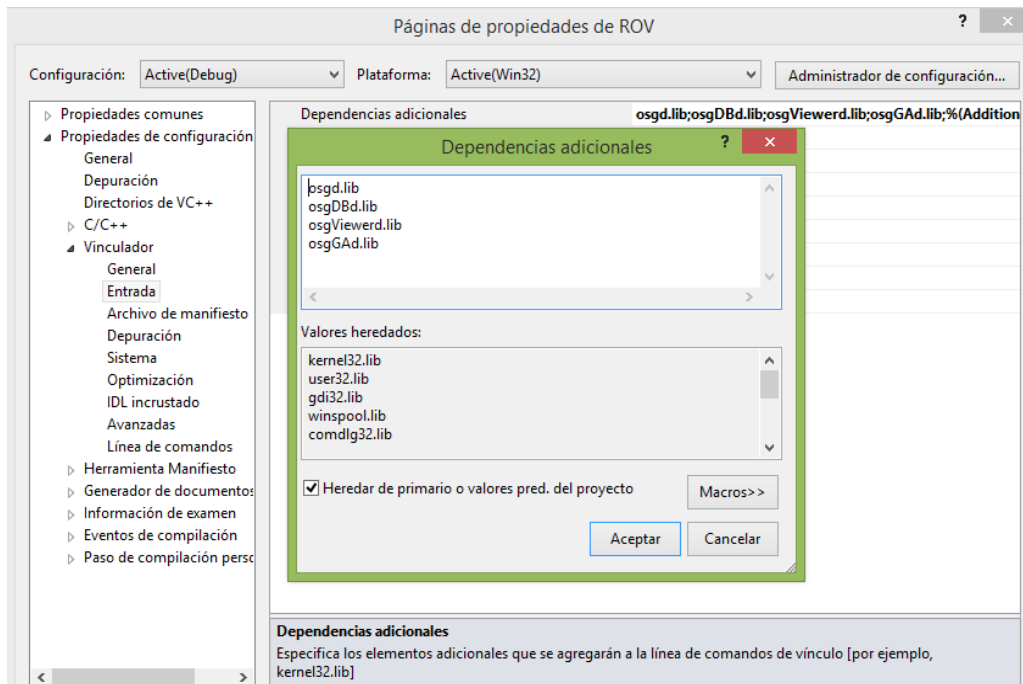
Proyecto > Propiedades de (Nombre de archivo) > Propiedades de configuración > **Vinculador** > **Entrada** > Dependencias adicionales

4.2. Introduce la dirección de las librerías que vamos a utilizar en nuestro programa.

En nuestro caso son:

osgd.lib; osgDBd.lib; osgViewerd.lib; osgGAd.lib

-La letra d al final del nombre de cada biblioteca indica que pertenecen a la configuración debug.



5. Compilar el programa.

Ctrl+f5

Este proceso es algo complejo y puede llevar a errores por lo que se recomienda seguir los pasos atentamente. Existe documentación adicional sobre como instalar librerías en las páginas oficiales de Eigen y OpenSceneGraph.

Eigen http://eigen.tuxfamily.org/index.php?title=Main_Page

OpenSceneGraph <http://www.openscenegraph.org/>

Adicionalmente se puede consultar los siguientes videos tutoriales en los que se explica cómo descargar e instalar la biblioteca OSG

<https://www.youtube.com/watch?v=115PAVCj2iY>

<https://www.youtube.com/watch?v=C4NiZoQvtfs>

Archivos OBJ

Los archivos OBJ incluidos en el trabajo (CuerpoROV.obj, Helice.obj, Flecha.obj, Terrain.obj y ROCK.obj) necesitan ser referenciados del mismo modo que las bibliotecas para poder ser utilizados en la simulación.

Para ello basta con añadir el directorio en el que estén guardados estos archivos con los demás directorios de inclusión. En nuestro caso están guardados dentro del propio programa.

1. Edita los directorios de inclusión.

Proyecto > Propiedades de (Nombre de archivo) > Propiedades de configuración > **C/C++** > **General** > Directorios de inclusión adicionales

2. Introduce la dirección de la carpeta donde se guardan los archivos OBJ

C:\Users\<Name>\Documents\Visual Studio 2010\Projects\ROV\ROV

Referencias: [37], [38]

Referencias

1. Rhinoceros 3D. Wikipedia. http://es.wikipedia.org/wiki/Rhinoceros_3D
2. MATLAB. Wikipedia. <http://es.wikipedia.org/wiki/MATLAB>
3. C++. Wikipedia. <http://en.wikipedia.org/wiki/C%2B%2B>
4. First mariners. Archaeology. <http://archive.archaeology.org/9805/newsbriefs/mariners.html>
5. Transporte Marítimo. Wikipedia. https://es.wikipedia.org/wiki/Transporte_mar%C3%ADtimo
6. Open ROV. Underwater exploration robots. <http://www.openrov.com>
7. ROV Exchange. <http://www.rovexchange.com>
8. QSTAR ROV Division. <http://www.rovs.eu/>
9. Homebuilt ROV's. <http://home.comcast.net/~homebuiltrovs/>
10. Ocean Explorer. NOAA. <http://oceanexplorer.noaa.gov/welcome.html>
11. Ropos. <http://www.ropos.com/>
12. Kongsberg Maritime. <http://www.km.kongsberg.com/>
13. IdRobotica. <http://www.idrobotica.com/>
14. SeaView Systems. FAQ. <http://www.seaviewsystems.com/questions/what-is-an-rov/>
15. ResearchSubmarine. http://www.researchsubmarine.com/sp_index1.htm
16. FOSSEN, T. (1994). *Guidance and Control of Ocean Vehicles*. Estados Unidos: Wiley.
17. Buoyancy. The Physics Hypertextbook. <http://physics.info/buoyancy/>
18. Drag. The Physics Hypertextbook. <http://physics.info/drag/>
19. Marine Propulsion. Wikipedia. https://en.wikipedia.org/wiki/Marine_propulsion

20. Blue Robotics. <https://www.bluerobotics.com/>
21. GONZALEZ, L. (2004) *Design, Modelling and Control of an Autonomous Underwater Vehicle*. Tesis. Australia: University of Western Australia.
22. ROSS, A., FOSSEN, T. y JOHANSEN T. (2004). “Identification of underwater vehicle hydrodynamic coefficients using free decay tests” en Conference on Control Applications in Marine Systems. Noruega: IFAC.
23. HYAKUDOME, T. (2011) “Design of Autonomous Underwater Vehicle” en International Journal of Advanced Robotic Systems. Japón: Japan Agency for Marine-Earth Science and Technology (JAMSTEC).
24. HALL, R. y ANSTEE, S. (2011) *Trim Calculation Methods for a Dynamical Model of the REMUS 100 Autonomous Underwater Vehicle*. Australia: Maritime Operations Division.
25. BUSCH, R. (2009) *Trim Calculation Methods for a Dynamical Model of the REMUS 100 Autonomous Underwater Vehicle*. Tesis. Sudáfrica: University of Stellenbosch.
26. FANDIÑO DE ANDRÉS, I. (2014) *Robótica Industrial*. España: Universidad Politécnica de Cartagena.
27. GARCÍA GARCÍA, J. (2013) *Desarrollo de una herramienta informática para la simulación dinámica de vehículos submarinos no tripulados*. Proyecto Fin de Carrera. España: Universidad Politécnica de Cartagena.
28. Propeller Static & Dynamic Thrust Calculation. ElectricRCaircraftGuy. <http://www.electricrcaircraftguy.com/2013/09/propeller-static-dynamic-thrust-equation.html>
29. KIM, J. (2008) *Underwater Vehicles. Thruster Modeling and Controller Design for Unmanned Underwater Vehicles (UUVs)*. Corea Del Sur: Seoul National University of Technology.
30. Crouzet Motors. Crouzet. <http://www.crouzet.es/productos/crouzet-motors/motores-de-corriente-continua/motores-de-corriente-continua.htm>
31. PALACIOS GARCÍA, A.C. (2007) *Prototipo de robot acuático explorador*. Tesis. México: Universidad Autónoma de Puebla.

32. Controlador de acción proporcional y derivativa. E-ducative. [http://e-ducativa.catedu.es/44700165/aula/archivos/repositorio//4750/4926/html/14_controlador de accin proporcional y derivativa pd.html](http://e-ducativa.catedu.es/44700165/aula/archivos/repositorio//4750/4926/html/14_controlador_de_accin_proporcional_y_derivativa_pd.html)
33. Eigen. http://eigen.tuxfamily.org/index.php?title=Main_Page
34. How to Change the Color of an osg::Node's Geometry. http://www.vis-sim.com/osg/code/osgcode_color1.htm
35. OpenSceneGraph. <http://www.openscenegraph.org/>
36. TurboSquid. <http://www.turbosquid.com/>
37. OSG installation. YouTube. <https://www.youtube.com/watch?v=115PAVCj2iY>
38. Open Scene Graph 3.0.1 in VS2010. YouTube. <https://www.youtube.com/watch?v=C4NiZoQvtfS>