

Proyecto Fin de Carrera Grado en Ingeniería Aeroespacial

Modelado y Control de un Vehículo Marino de Superficie no Tripulado

Autor: Javier Gallo Aguilar

Tutor: Ángel Rodríguez Castaño

Dpto. Control Automático
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2021



Proyecto Fin de Carrera
Grado en Ingeniería Aeroespacial

Modelado y Control de un Vehículo Marino de Superficie no Tripulado

Autor:

Javier Gallo Aguilar

Tutor:

Ángel Rodríguez Castaño

Profesor Titular

Dpto. Control Automático
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2021

Proyecto Fin de Carrera: Modelado y Control de un Vehículo Marino de Superficie no Tripulado

Autor: Javier Gallo Aguilar
Tutor: Ángel Rodríguez Castaño

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Agradecimientos

Con este proyecto pongo fin la etapa de grado, una época inolvidable, que ha supuesto un esfuerzo enorme y muchos sacrificios pero que ha merecido la pena a cada instante. Termino siendo una persona mucho más madura, culta y racional de lo que era cuando llegué el primer día a la Sevilla. Llegados a este punto, no es lo duro ni lo exigente del grado lo que me llevo, ni ya me pesan las interminables horas sentado en mi escritorio. Cuando pienso en mis años en Sevilla recuerdo a cada una de las personas que me han acompañado y los momentos felices que he vivido con ellos.

En primer lugar, a mis padres. A ellos quiero agradecerles todo lo que han tenido que hacer para que jamás me haya faltado nada, ese esfuerzo que hicieron para regalarme estos años en Sevilla y todo lo que me llevo de aquí. A mi padre, que siempre has confiado en mí y has sido un hombro de apoyo en los buenos momentos y en los malos. Desde pequeño me has inculcado esa curiosidad por aprender cosas nuevas y ese interés por esforzarme más allá de lo que incluso yo me veía capaz. Las únicas palabras que salen de mí ahora mismo son "Gracias Papá". Y a mi madre, ¿Qué decir de mi madre?, ¿Qué decir que no pueda expresarse con un "Te quiero"? Has sido siempre la voz de mi conciencia, la que has dado la cara por mí, la que has antepuesto mi bienestar al tuyo propio, la que sabes cómo me siento mejor que yo mismo, la persona a la que le debo todo. Siempre has sido y serás un referente en mi vida. Muchas gracias a los dos por ser tan pacientes conmigo y no haberos rendido nunca, aunque no lo haya puesto fácil.

No me olvido de mi hermana. María. Sé que tenemos nuestros más y nuestros menos, pero eres todo lo que un hermano podría querer. Hemos crecido juntos, nos hemos caído juntos, y nos hemos vuelto a levantar juntos. Siempre estás ahí para apoyarme cuando verdaderamente lo necesito. A mi abuelo también, por ser un referente en mi vida, por haberme enseñado a ser quien soy hoy, y por haber antepuesto a tu familia por delante de todo y permitirnos disfrutar de lo que tenemos hoy. Sé que sin ti hoy no estaría donde estoy ahora. No quiero olvidar por supuesto al resto de mi familia, que siempre me han apoyado cuando me más me ha hecho falta.

A mis amigos, en especial a Jose Luis, que han estado conmigo incluso después de venirme a Sevilla y que no han tenido reparo en quedar conmigo para relajarme cuando la carrera sacaba lo peor de mí. Pero hablando de amigos, no puedo sino dar mil gracias a toda la gente que he conocido aquí en Sevilla. A Miguel, que fue mi primer amigo en la Escuela y en el Colegio. Has sido mi compañero desde primero, a ti te debo la gran trayectoria académica que llevo. Me has ayudado innumerables veces, y no ha sido menos con este proyecto. Desde luego, si algo me llevo de aquí, es a un gran amigo y hermano. A Mendiola, que ha estado ahí sopesándome cuando más insoportable me ponía. Desde luego otro gran amigo que del que tengo el honor de presumir. Cómo olvidar las tardes interminables en la sala de estudio del Colegio estudiando entre los tres hasta que acabábamos quemados. Vosotros habéis sido un pilar indispensable tanto en mi desarrollo académico como personal. También debo dar las gracias a toda la gente que ha estado ahí tanto de la Escuela como del Colegio, a Drolu, a Razquin, a Agustín, a Chechu, a Linde, a Dani, a Jacobo, a Víctor, a Javi, a Ale. De verdad os estoy muy agradecido, aunque a veces me cueste expresarlo. No me olvido de Ana, que no me ha aguantado poco, y por haberse contenido y no ponerse tiquismiquis al ver mis dibujos hechos a mano.

Y por supuesto a ti, Carmen. Nunca podré estar lo suficientemente agradecido por poder pasar mi tiempo a tu lado. Siempre estás ahí para escucharme, para darme ánimos, para hacerme recuperar las fuerzas, para sacar lo mejor de mí. Gracias por haberte sentado pacientemente a mi lado escuchando quejarme de lo difícil que es la carrera, por haberme consolado en mis momentos más duros, por ser la mejor novia del mundo. Eres una persona muy especial para mí y no podría estar más orgulloso de estar contigo.

Javier Gallo Aguilar
Sevilla, 2021

Resumen

La tecnología de modelado hidrodinámico y sistemas de control de vehículos marinos ha progresado enormemente en estos años recientes. En especial, los vehículos de superficie no tripulados (USV's) han encontrado un número de campos en los que su aplicación es de gran utilidad, desde la monitorización del agua hasta la recogida de residuos y el estudio de fondos marinos.

Surge entonces la idea de realizar una herramienta informática que, sin exigir una excesiva cantidad de datos y parámetros, y pensada para uso ingenieril de análisis, sea capaz de simular el comportamiento de un vehículo controlado de forma autónoma en un entorno marino gobernado por unas leyes de control.

Entonces, este proyecto consiste en la creación de dicha herramienta. Para ello el proyecto se divide en diferentes capítulos, cada uno con un objetivo distinto, con el fin de organizar de forma coherente esta memoria.

Lo primero sería realizar un estudio analítico de las actuaciones del vehículo, junto a las fuerzas que intervienen en su movimiento. Es por tanto objetivo de este capítulo establecer las ecuaciones y relaciones entre parámetros que rigen el comportamiento del USV.

Después, se pasaría a construir un modelo dinámico sencillo del vehículo, pero lo suficientemente completo como para que se aproxime a la realidad y a las físicas que participan en la simulación. Para ello se necesitará determinar valores apropiados para los parámetros que definen el problema.

Una vez logrado esto, se comenzarán a crear una serie de controladores funcionales que sean capaces de dirigir el USV en el entorno de acuerdo a las leyes de control que se establezcan previamente.

Finalmente, con el modelo y controlador ya terminados, se probará su alcance y su fiabilidad para distintas configuraciones de actuadores y distintas misiones, comparando los resultados obtenidos para hallar la configuración óptima. Acto seguido, a la vista de los resultados, se emitirán unas conclusiones finales sobre el proyecto.

Abstract

The technology of hydrodynamics modeling and marine craft motion control systems has progressed greatly in recent years. In particular, unmanned surface vehicles (USV's) have found a number of fields in which their application is of great utility, such as water monitoring, waste collection or seabed study.

The idea is to create an informatic tool that, without demanding an excessive quantity of data and parameters, and designed for an engineer use, will be able to simulate the behaviour of an autonomous vehicle in a marine environment ruled by a set of control laws.

So, this project consists in the creation of that app. For this, the project is divided into different chapters, each with a different objective, in order to organize this report in a coherent way.

First, we will carry out an analytic study about the performances of the vehicle, together with the forces that take part in its movement. Therefore, the objective of this chapter is to establish the equations and relationships between parameters that govern the USV's behaviour.

Afterwards, we would go on to build a simple but complete dynamic model so that it approximates reality and the physics that participate in the simulation. In order to achieve that, it will be needed to establish proper values for the parameters that define the problem.

Once this is achieved, we will begin to create a series of functional controllers that will be able to lead the USV on the environment in accordance with the control laws previously established.

Finally, with the model and controller already finished, we will test its scope and reliability for different actuators configurations and missions, comparing the results obtained in order to find the optimal configuration. Then, in view of the results, we will issue final conclusions about the project.

Índice

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Notación</i>	IX
1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	3
1.3 Metodología	3
1.4 Estructura de la memoria	4
2 Análisis del problema	5
2.1 Matriz de coeficientes de empuje	5
2.1.1 Actuación de los propulsores (T-rpm)	5
2.1.2 Actuación del timón (F- δ)	7
2.2 Matriz de configuración de empuje	7
2.2.1 Matriz de configuración extendida	8
2.3 Optimización mediante programación cuadrática	9
2.3.1 Funcionamiento de quadprog	9
2.3.2 Uso para control allocation	9
2.3.3 Solución para $\alpha = \text{cte}$	10
2.3.4 Solución para $\alpha \neq \text{cte}$	10
2.4 Estudio de distintos USV's	12
3 Modelado de vehículos marinos de superficie	15
3.1 Modelado en Simulink	15
3.2 Navegador	16
3.2.1 Matriz de configuración	17
3.2.2 Fuerzas externas	17
3.2.3 Vector de estado	17
3.2.4 Modelado en Matlab	18
3.3 Configuraciones de USV's	19
3.3.1 1º Configuración: 2 propulsores longitudinales	19
3.3.2 2º Configuración: 2 propulsores longitudinales y 1 transversal	19
3.3.3 3º Configuración: 2 propulsores azimutales	20
3.3.4 4º Configuración: 2 propulsores azimutales y 1 transversal	20
4 Diseño y desarrollo de controladores	23
4.1 Evolución del diseño	23
4.1.1 Modelo cinemático	23
4.1.2 Modelo dinámico Nº1	24
4.1.3 Modelo dinámico Nº2	24

4.1.4	Modelo dinámico N°3	25
4.1.5	Modelo dinámico N°4	26
4.1.6	Modelo dinámico N°5	27
4.2	Resultados de pruebas	28
4.2.1	Simulación N°1	28
4.2.2	Simulación N°2	30
4.2.3	Simulación N°3	32
4.2.4	Simulación N°4	34
5	Conclusiones y desarrollos futuros	37
5.1	Principales obstáculos	37
5.2	Puntos a mejorar	37
	Anexo: Matlab	39
	<i>Índice de Figuras</i>	43
	<i>Índice de Tablas</i>	45
	<i>Índice de Códigos</i>	47
	<i>Bibliografía</i>	49

Notación

GDL	Grados de libertad
QP	Programación cuadrática
USV	Vehículo de superficie no tripulado
PID	Controlador proporcional-integrador-derivativo
T	Empuje
V	Velocidad
m	Masa
I_b	Momento de inercia
n	Revoluciones por minuto
t	Tiempo
K_{ij}	Matriz de coeficientes de empuje
T_{ij}	Matriz de configuración de empuje
u_i	Variables actuadoras
ρ	Densidad
D	Diámetro
K_T	Coefficiente de empuje
t_p	Coefficiente de reducción de empuje
J_0	Parámetro de avance
α_i	Ángulo del propulsor azimutal i
δ_R	Ángulo del timón de dirección
F_X	Fuerza en el eje X
F_Y	Fuerza en el eje Y
M	Momento en el eje Z
L	Sustentación
C_L	Coefficiente de sustentación
t_i	Vector de configuración de empuje
$G_{b \rightarrow g}$	Matriz de giro
c_T	Coefficiente de resistencia traslacional
c_R	Coefficiente de resistencia rotacional
χ	Ángulo de resbalamiento
f	Vector de elementos
h	Vector de holguras
W	Matriz de pesos de actuaciones
Q	Matriz de pesos de holguras
τ	Vector de fuerzas
X	Vector de estado
\dot{X}	Derivada del vector de estado
sen	Función seno
cos	Función coseno
tan	Función tangente
arctan	Función arcotangente

atan2	Función arcotangente en cuatro cuadrantes
x	Posición en el eje x
y	Posición en el eje y
θ	Orientación
v_x	Velocidad en el eje x
v_y	Velocidad en el eje y
w	Velocidad angular
a_x	Aceleración en el eje x
a_y	Aceleración en el eje y
α	Aceleración angular
x_0	Posición inicial en el eje x
y_0	Posición inicial en el eje y
θ_0	Orientación inicial
v_{x0}	Velocidad inicial en el eje x
v_{y0}	Velocidad inicial en el eje y
w_0	Velocidad angular inicial
x_f	Posición objetivo en el eje x
y_f	Posición objetivo en el eje y
θ_f	Orientación objetivo
F_{max}	Fuerza de saturación
F_p	Vector fuerza de perturbaciones
R_s	Vector ruido de sensores
P	Proporcional
I	Integral
D	Derivativo
N	Coefficiente de filtro

1 Introducción

Los Vehículos de Superficie no Tripulados, o USV para abreviar, representan una tecnología que ha ido creciendo a lo largo de estos años, junto con el empleo de otros robots autónomos como los androides o los drones, con el objetivo de optimizar y facilitar una gran cantidad de tareas que a día de hoy se han venido haciendo por la mano de un hombre. Estos robots encuentran lugar en una amplia gama de sectores en los cuales ya apenas se vislumbra un futuro sin la utilización de estos.

1.1 Motivación

Como se ha comentado, el empleo de USV's en numerosos ámbitos de la ciencia y la investigación ha crecido exponencialmente en la última década. Su uso oscila entre aplicaciones tan variadas como:

- La monitorización de la calidad del agua, la cual es un gran problema medioambiental y uno de los mayores retos de la humanidad. Muchos países del mundo han sido confrontados por la polución del agua debido a fugas de aceite, residuos de plástico y actividades humanas.



Figura 1.1 Monitorización del agua.

- El mapeado de fondos marinos, tanto para finalidades científicas como para estudio del terreno en búsqueda de la prevención de desastres naturales. Los sensores de estos robots pueden recoger datos de lugares muy oscuros y difíciles de alcanzar durante un largo período de tiempo.

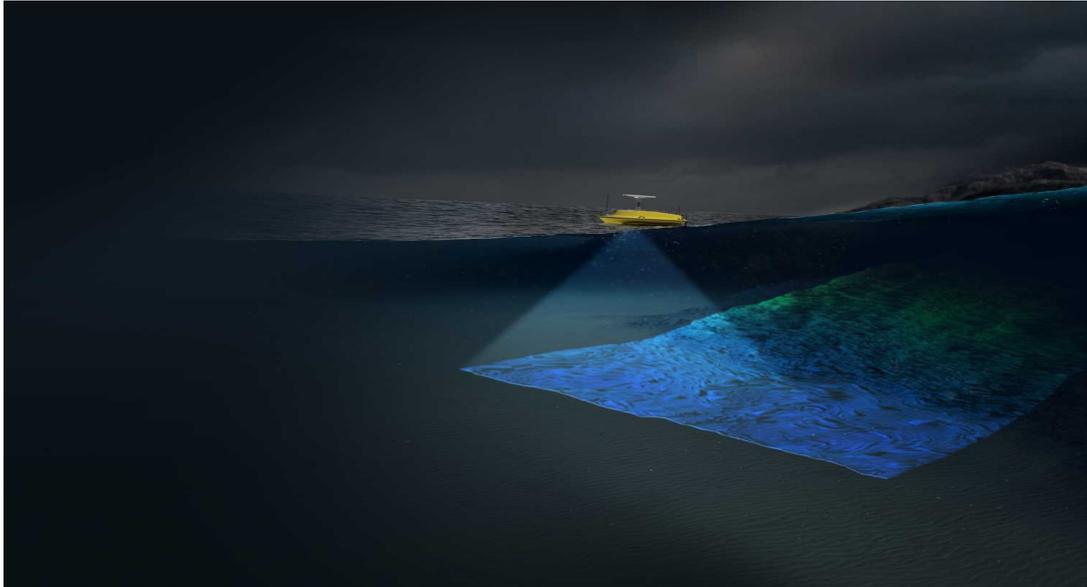


Figura 1.2 Mapeado de fondos marinos.

- La monitorización de la vida marina, la cual es crucial al entender fenómenos complejos que pueden ser de un valor incalculable para prevenir problemas medioambientales.



Figura 1.3 Monitorización de la vida marina.

- Otra serie de estudios relacionados con la investigación de las propiedades del agua, detección de residuos o perturbaciones y control del ecosistema marino con el fin de preservar el estado natural del bioma.

Sin embargo, los USV's comerciales son caros y no están completamente creados con hardware de código abierto, haciendo difícil comprarlos y modificar sus diseños y programación para adaptarse a los distintos entornos.

La gran aplicabilidad de estos robots autónomos junto con las facilidades que supone diseñar un modelo de código abierto han sido la fuente de inspiración que ha llevado al emprendimiento de este proyecto.

1.2 Objetivos

El objetivo de este proyecto es, por tanto, diseñar un modelo de USV completamente autónomo, de pequeño tamaño, bajo coste y código abierto que se adapte a las condiciones ambientales y las necesidades específicas para el tipo de misión que se le quiera asignar. El alcance de este estudio no será precisamente crear el propio vehículo, sino modelar una simulación que nos muestre de la forma más exacta posible su comportamiento una vez ya creado y puesto en funcionamiento.



Figura 1.4 Descripción de un USV de pequeño tamaño.

Más concretamente, el objeto principal de esta memoria va a ser el programar un simulador del USV, prestando especial interés en la configuración de propulsores que permiten su movimiento. Esto es debido a que el principal control que se quiere tener sobre el vehículo es el posicionamiento dinámico de éste entre una serie de puntos definidos previamente y orientado en cada uno de la manera deseada.

1.3 Metodología

Como se acaba de mencionar, el objetivo es diseñar un modelo de USV que tenga como función el posicionamiento dinámico de este. El posicionamiento dinámico se caracteriza por darle prioridad a la orientación del vehículo antes que a la traslación de este y por la baja velocidad de su movimiento, lo que permite el desplazamiento lateral para optimizar el consumo de los propulsores.

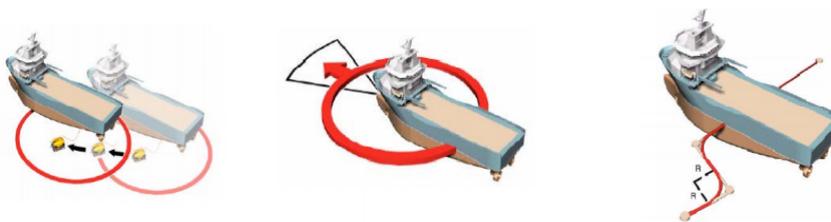


Figura 1.5 Posicionamiento dinámico de USV's.

Este diseño se va a llevar a cabo usando la herramienta de Matlab, Simulink. Primeramente se va a realizar un estudio del problema dinámico para establecer las relaciones entre los parámetros del problema. A continuación se van a modelar las físicas que rigen el comportamiento del vehículo y al propio vehículo en sí mismo, para finalmente diseñar un controlador que gobierne el movimiento de este y le permita alcanzar su objetivo.

De los resultados obtenidos de la simulación se tomarán las conclusiones oportunas con el fin de hallar las configuraciones óptimas y tener en consideración los posibles puntos que podrían mejorar su rendimiento.

1.4 Estructura de la memoria

La memoria que se prelude está estructurada en 5 diferentes capítulos con el objetivo de organizar mejor la información y dividirla en sus respectivos campos de aplicación. Los capítulos que presenta son los siguientes:

- **Introducción:** capítulo que se acaba de mostrar. Tiene como finalidad mostrar las motivaciones que han llevado a cabo la realización de este proyecto, junto con los objetivos del mismo y la metodología empleada para realizarla y estructurarla.
- **Análisis del problema:** segundo capítulo de la memoria que tiene como objetivo presentar todas las ecuaciones y relaciones matemáticas que rigen el comportamiento del problema en cuestión. En él se establecerán gran parte de los parámetros de los distintos componentes que participan en el mismo.
- **Modelado de USV's:** tercer capítulo en el que se diseñarán distintas configuraciones de actuaciones de vehículos en Simulink y Matlab con el fin de obtener una amplia abanico de posibilidades entre las que poder comparar a la hora de simular sus comportamientos para escoger el que optimice el tiempo de la misión y el consumo de sus actuadores.
- **Diseño de controladores:** cuarto capítulo en el que se desarrollarán un número de distintos controladores para regir el movimiento del vehículo, basados en distintos parámetros geométricos de la misión. Se realizará también un estudio comparativo para determinar cual cumple el objetivo preestablecido de la manera más eficaz posible.
- **Conclusiones:** capítulo final de la memoria en la que se establecerán las conclusiones finales extraídas de los resultados proporcionados, y se propondrán posibles mejoras al diseño para futuros trabajos.

2 Análisis del problema

En este capítulo se va a introducir el estudio analítico de las actuaciones del barco, así como de la cinemática y la hidrodinámica asociadas al problema. Se ha mostrado especial interés por el modelado matemático de los distintos actuadores que pueden incorporarse en la embarcación, así como en la optimización a la hora del empleo de sistemas sobreactuados con un número redundante de controladores, para lo se ha estudiado el potencial de la programación cuadrática.

2.1 Matriz de coeficientes de empuje

Para hallar la relación entre las actuaciones del barco (revoluciones por minuto de los motores y deflexión de los timones) y las fuerzas y momentos que estas generan, se debe realizar un estudio hidrodinámico de los actuadores.

El estudio y los resultados del mismo se presentan a continuación:

2.1.1 Actuación de los propulsores (T-rpm)

Para calcular los valores de los coeficientes de empuje asociados a los propulsores se debe realizar el análisis dinámico que relaciona las revoluciones por segundo de este con el empuje que proporciona. La ecuación que regula este comportamiento es la siguiente:

$$T = (1 - t_p)\rho D^4 K_T (J_0)n|n| \quad (2.1)$$

donde ρ es la densidad del agua, D es el diámetro del motor y t_p es el coeficiente de reducción de empuje. Al situarse los propulsores en la parte trasera del barco se genera un efecto de succión que aumenta la presión en el casco y que se traduce en un aumento de la resistencia. El empuje necesario para contrarrestar esta penalización δT es la que define: $t_p = (T - \delta T)/T$.

El parámetro de avance J_0 viene dado por la expresión:

$$J_0 = V/(nD) \quad (2.2)$$

En la ecuación 2.1, K_T es el coeficiente de empuje en función del parámetro de avance J_0 , que es modelado como una función lineal dada por:

$$K_T = \alpha_1 + \alpha_2 J_0 \quad (2.3)$$

donde α_1 y α_2 son dos constantes que pueden obtenerse tanto de forma analítica como de forma experimental. Con el fin de simplificar el proceso se tomarán como referencia las siguientes gráficas obtenidas experimentalmente [2] para un modelo de barco con geometría y características similares al empleado en la simulación.

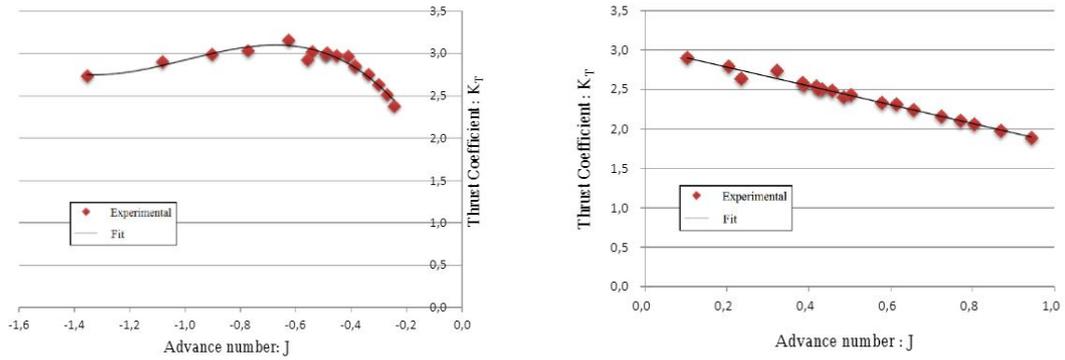


Figura 2.2 Aceleración en el sentido de la marcha y contrario.

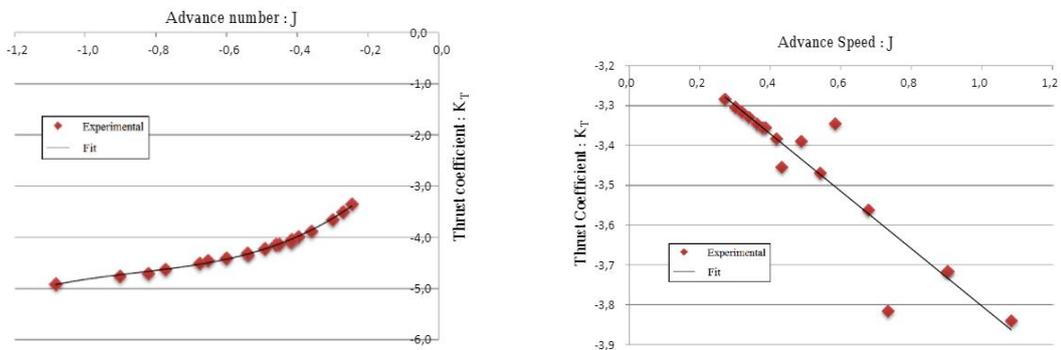


Figura 2.4 Freno en sentido contrario a la marcha.

Como se puede apreciar, la relación entre estos dos parámetros depende tanto de si el USV está desplazándose hacia delante o hacia atrás, como de si está acelerando o frenando. Sin embargo, se va a simplificar aún más el modelo unificando los cuatro casos como uno solo y expresando el empuje como:

$$T = cVn + d|n|n \tag{2.4}$$

reduciendo así de cinco parámetros a considerar ($\rho, D, \alpha_1, \alpha_2, t_p$) a dos parámetros (c, d), que se podrán obtener también tanto de forma analítica como de forma experimental, pero que tomaremos de una gráfica elaborada para un modelo de barco similar al empleado.

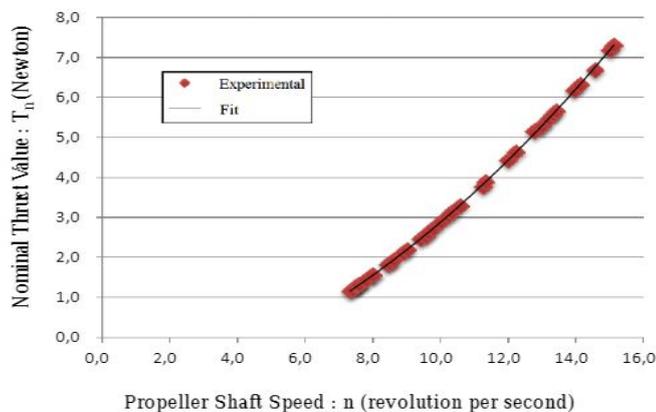


Figura 2.5 Relación Empuje - Revoluciones por segundo.

De esta forma se termina estableciendo una relación aproximada de linealidad entre las revoluciones por minuto los propulsores y las fuerzas que estos generan. Esta relación vendrá definida por la constante k .

2.1.2 Acuación del timón (F- δ)

Al igual que con los propulsores, se va a presentar el modelado matemático que se usará para relacionar el ángulo de deflexión del timón con las fuerzas longitudinales, transversales y momento de guiñada que este genera. Las ecuaciones que regulan este comportamiento son las siguientes:

$$\begin{cases} F_X = |F_N \sin \delta_R| \\ F_Y = a_y F_N \cos \delta_R \\ M_z = a_z F_N \cos \delta_R \end{cases} \quad (2.5)$$

donde δ_R es el ángulo del timón, normalmente considerado pequeño ($\delta_R \ll 1$), F_N es la fuerza normal al timón, a_y es el coeficiente de influencia del casco sobre la fuerza F_Y del timón, y a_z es el coeficiente de influencia del casco sobre el momento M_z del timón.

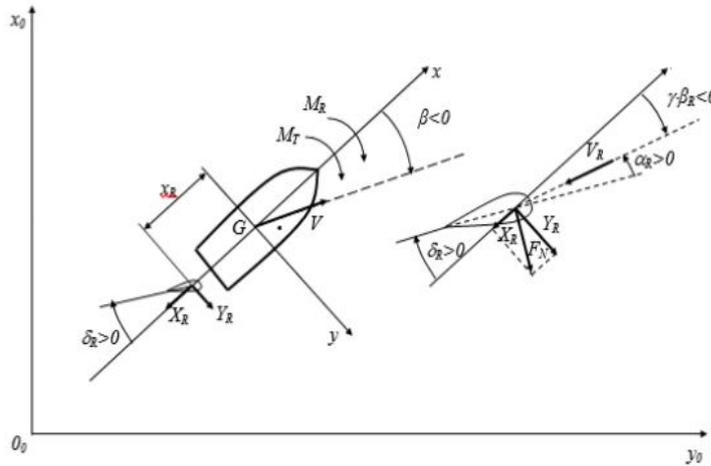


Figura 2.6 Modelado del rudder.

Los coeficientes se pueden determinar de forma experimental [10], pero puesto que en la mayoría de configuraciones se va a considerar $\delta_R \ll 1$, resulta evidente considerar despreciable la fuerza longitudinal al timón (*Drag*), por lo que únicamente se requiere obtener la fuerza normal (*Lift*) para establecer la solución del problema.

$$L = 1/2 \rho V^2 A_R C_L \quad (2.6)$$

siendo A_R el área en planta del timón y C_L el coeficiente de sustentación del perfil seleccionado, el cual se calculará mediante un programa de fluidodinámica.

2.2 Matriz de configuración de empuje

Por otro lado, es de interés hallar la relación entre las fuerzas que generan cada uno de los actuadores del barco y las fuerzas resultantes sobre el centro de gravedad del mismo, ya que serán las que rijan su movimiento modelado como el de una partícula puntual.

Para ello es necesario realizar un estudio geométrico del vehículo, determinar sus dimensiones, su centro de gravedad y la posición relativa de los actuadores respecto a este. Primeramente, se va a definir los distintos tipos de actuadores con los que se van a trabajar:

- **Propulsor longitudinal:** están montados normalmente en la popa del casco de la embarcación y producen una fuerza en la dirección del eje x . Es el tipo de propulsor más empleado.
- **Propulsor transversal:** suelen posicionarse en el casco del barco dentro de un tubo transversal, proporcionando empuje en la dirección del eje y . Solo son efectivos a bajas velocidades, lo que los hace ideales para posicionamiento dinámico.
- **Propulsor azimutal:** propulsores que pueden girar un ángulo α en el eje z y producir una fuerza de dos componentes en el plano horizontal. Son atractivos para posicionamiento dinámico y resultan en problemas de control sobreactuados que pueden ser optimizados en función del consumo o de la posibilidad de fallo.
- **Timón de dirección:** suele estar localizado en la popa de la embarcación y producen una fuerza en la dirección del eje y que induce un momento de guiñada. Son el principal mecanismo de dirección para embarcaciones tradicionales.

Tabla 2.1 Actuadores y fuerzas que generan.

Actuador	u	α	f^T
Propulsor longitudinal	rpm	-	$[F_x, 0, 0]$
Propulsor transversal	rpm	-	$[0, F_y, 0]$
Propulsor azimutal	rpm	ángulo	$[F_x \cos \alpha, F_x \sin \alpha, 0]$
Timón de profundidad	ángulo	-	$[0, F_y, 0]$

Se puede apreciar que cada actuador influye de manera distinta en la fuerza resultante total, teniendo así asignado cada uno su *Vector de configuración de empuje* [4] en función de su posición respecto al centro de gravedad del USV, resultando así en una matriz de dimensiones $3 \times n$, siendo n el número de actuadores que tenga el vehículo.

$$\underbrace{t_l = \begin{Bmatrix} 1 \\ 0 \\ -l_y \end{Bmatrix}}_{\text{Propulsor longitudinal}} \quad
 \underbrace{t_t = \begin{Bmatrix} 0 \\ 1 \\ l_x \end{Bmatrix}}_{\text{Propulsor transversal}} \quad
 \underbrace{t_a = \begin{Bmatrix} \cos \alpha \\ \sin \alpha \\ l_x \sin \alpha - l_y \cos \alpha \end{Bmatrix}}_{\text{Propulsor azimutal}} \quad
 \underbrace{t_f = \begin{Bmatrix} 0 \\ 1 \\ l_x \end{Bmatrix}}_{\text{Timón de dirección}} \quad (2.7)$$

2.2.1 Matriz de configuración extendida

Trabajar con propulsores azimutales puede ser especialmente complejo a la hora de su inserción en la matriz de configuración de empuje ($T = T(\alpha)$) ya que requeriría de resolver un problema de optimización no lineal. Por esta razón conviene eliminar en ángulo de la matriz y asignarle dos actuaciones independientes (u_x y u_y) que vendrán definidas por las siguientes relaciones [1]:

$$t_{ae} = \begin{Bmatrix} 1 & 0 \\ 0 & 1 \\ -l_y & l_x \end{Bmatrix} \quad (2.8)$$

$$u = \sqrt{u_x^2 + u_y^2} \quad (2.9)$$

$$\alpha = \text{atan2}(u_y, u_x) \quad (2.10)$$

Esta es la que se conoce como *Matriz de configuración extendida*, T_e , la cual nos permite evitar tener que resolver el problema de optimización no lineal, convirtiéndolo en uno que sí lo es.

2.3 Optimización mediante programación cuadrática

Para vehículos marinos, donde la matriz de configuración T no es cuadrada ($n \geq 3$) y hay más controladores que grados de libertad, es posible encontrar una distribución óptima de las fuerzas de control f para cada GDL usando un método específico.

El caso más simple de asignación de control, o *control allocation*, es en el que solo existen fuerzas producidas por propulsores fijos en combinación con timones de dirección, y en el que no existen ningún tipo de restricción tanto en vector de elementos f_i , α_i y u_i como en sus derivadas con el tiempo.

Sin embargo, en sistemas reales es importante minimizar el consumo de energía teniendo en cuenta factores como la saturación de los actuadores, desgaste por uso, sectores prohibidos o sobrecarga del sistema. En general, esto conlleva un problema de optimización restringido, para cuya resolución emplearemos la función de Matlab "*quadprog*".

2.3.1 Funcionamiento de *quadprog*

Esta función de Matlab permite resolver problemas de optimización convexos mediante programación cuadrática. Pero antes de utilizarlo, se debe comprender en que consiste la programación cuadrática (o QP) y como nos permite resolver problemas de optimización. *Quadprog* resuelve problemas con la siguiente estructura:

$$\begin{aligned} \min_x \quad & \frac{1}{2}x^T Hx + f^T x & (2.11) \\ \text{s.a.} \quad & lb \leq x \leq ub \\ & A_{eq}x = b_{eq} \\ & Ax \leq b \end{aligned}$$

donde x se corresponde con el vector de elementos, H con el término objetivo cuadrático, f^T con el término objetivo lineal, lb y ub con los límites superiores e inferiores respectivamente, A_{eq} y b_{eq} con las restricciones de igualdad lineales, y A y b con las restricciones de desigualdad lineales.

Nótese que la nomenclatura que se acaba de emplear no es la que se usará durante el resto del desarrollo del problema, sino la estándar en este tipo de problemas de QP .

2.3.2 Uso para control allocation

En paralelo a lo mencionado en el apartado anterior, el problema de optimización a resolver que se trata en este documento viene definido por las siguientes expresiones y restricciones [6]:

$$\begin{aligned} \min_J \quad & \{J = f^T W f + h^T Q h\} & (2.12) \\ \text{s.a.} \quad & T f = \tau + h \\ & f_{min} \leq f \leq f_{max} \end{aligned}$$

donde:

- **f**: es el vector de elementos, que incluye las fuerzas generadas por los actuadores.
- **h**: es el vector de holguras, que representan lo lejos que nos quedamos de las actuaciones deseadas.
- **W**: es la matriz de pesos asociada a las actuaciones.
- **Q**: es la matriz de pesos asociados a las holguras.
- **τ** : es el vector de fuerzas deseadas proporcionadas por el controlador ($\tau = T K u$).
- **f_{max} y f_{min}** : son los vectores de saturación superior e inferior de las actuaciones.

Tanto la matriz W como Q deben ser definidas positivas. En el caso de las actuaciones para vehículos marítimos, se penalizan a todos los motores por igual, mientras que la deflexión del timón se penalizan mucho menos al no consumir apenas energía.

Las holguras se penalizan de distinta manera, ya que existe una mayor libertad a la hora de decidir que variables queremos que sean más precisas. Lo ideal es que todas sus componentes sean cero ($Tf = \tau$), pero para problemas de posicionamiento dinámico es de mayor interés favorecer primero que la orientación de la embarcación sea la deseada, relegando a segundo lugar la velocidad a la que se llega al destino.

Código 2.1 Ejemplo del uso de quadprog.

```
T = [ 1   1   0;
      0   0   1;
      -ly1 -ly2 lx3];
W = diag([1 1 1]);
Q = 1e4*diag([1 10 100]);

F = [tau(1); tau(2); tau(3)];
H = blkdiag(W,Q);
f = [];

A = [];
b = [];
Aeq = blkdiag([T,-eye(3)]);
beq = F;
lb = [u1_min u2_min u3_min 2*Fx_min 2*Fy_min 2*M_min];
ub = [u1_max u2_max u3_max 2*Fx_max 2*Fy_max 2*M_max];
xo = zeros(6,1);
opts = optimset('Algorithm','active-set','Display','off','LargeScale','off');
[x,fval,exitflag,output,lambda] = quadprog(H,f,A,b,Aeq,beq,lb,ub,xo,opts);
```

Para conseguir el resultado óptimo de la forma más precisa posible, y asegurarnos de que las holguras son próximas a cero, definimos las matrices de pesos de forma que $Q \gg W > 0$. Por otro lado, el término lineal que controla el producto cruzado de actuaciones y holguras se considera nulo por motivos de simplificación de cálculo.

2.3.3 Solución para $\alpha = cte$

Este es el caso más simple de control allocation ya que todos los propulsores son fijos y, por tanto, el problema de optimización es puramente lineal. También será el método por el que se optará a la hora de modelar el vehículo, ya que acarrea varias simplificaciones de cálculo numérico. De esta manera se establecerá el ángulo deseado de cada uno de los propulsores, y se supondrá que el propulsor es capaz de alcanzarlo independiente de su orientación actual.

La forma de resolverlo en Matlab sería similar al ejemplo mostrado en el apartado anterior solo que, al considerarse α como una constante, podemos introducirla en la matriz de configuración de empuje de forma que $T = T(\alpha) = cte$.

2.3.4 Solución para $\alpha \neq cte$

En un sistema real, los actuadores no modifican su valor instantáneamente, sino que requieren de un tiempo de transición. Durante este margen, la embarcación sigue moviéndose y recibiendo señales externas de los sensores, por lo que se debe tener en cuenta a la hora de encontrar la solución óptima para alcanzar las fuerzas deseadas.

Es de especial interés el caso de los propulsores azimutales, donde hay que tener en cuenta que estos solo pueden orientarse dentro de un rango de ángulos con el objetivo de evitar que el fluido sucio que atraviesa uno de los propulsores entre en otro, pudiendo así estropear este último. Las direcciones que quedan fuera de este rango se conocen como la "zona prohibida".

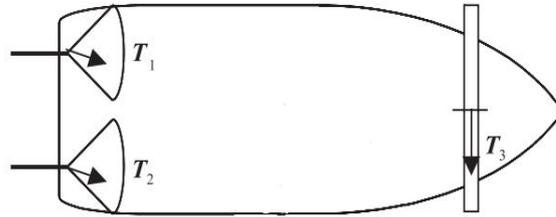


Figura 2.7 Embarcación con propulsores azimutales y transversal.

Se puede observar en la figura 2.7 que si los propulsores estuvieran orientados verticalmente, el flujo sucio que saldría del motor 1 entraría en el motor 2, lo cual no es deseado. En sistemas reales el óptimo puede involucrar orientar el propulsor en sentido opuesto ya que, como vimos en el apartado 2.1.1, estos generan una mayor fuerza con rpm's positivas que negativas. Esto provoca que para orientarlos en sentido opuesto se tengan que apagar momentáneamente mientras cruzan por la zona prohibida.

Claramente, mantener los motores apagados supone dejar el barco sin control momentáneamente, algo que es totalmente indeseado, por lo que se debe tener en cuenta a la hora de la optimización. Es por ello que se debe analizar el óptimo en todas las configuraciones posibles de orientación de los propulsores, lo cual supone un incremento del tiempo de cálculo así como de la complejidad.

Código 2.2 Pequeño extracto del código para Thruster Allocation del ADAM M1.

```

if (alpha(1)<(pi/2)) || (alpha(1)>(3*pi/2))
    if (alpha(2)<(pi/2)) || (alpha(2)>(3*pi/2)) % Conf. 1 P1-F, P2-F
        fsol(2) = u(2,1); % Se toma el ángulo del Prop. 1
        fsol(4) = u(4,1); % Se toma el ángulo del Prop. 2
    else % Conf. 4 P1-F, P2-B
        fsol(2) = u(2,4); % Se toma el ángulo del Prop. 1
        fsol(4) = u(4,4); % Se toma el ángulo del Prop. 2
    end
end
else
    if (alpha(2)<(pi/2)) || (alpha(2)>(3*pi/2)) % Conf. 3 P1-B, P2-F
        fsol(2) = u(2,3); % Se toma el ángulo del Prop. 1
        fsol(4) = u(4,3); % Se toma el ángulo del Prop. 2
    else % Conf. 4 P1-B, P2-B
        fsol(2) = u(2,2); % Se toma el ángulo del Prop. 1
        fsol(4) = u(4,2); % Se toma el ángulo del Prop. 2
    end
end
end
end

```

Para resolver este problema en Matlab se debe utilizar la matriz de configuración extendida T_e que se vio en el apartado 2.2.1, y calcular las rpm's y la orientación de cada motor azimutal (u_i y α_i) a posteriori, usando las expresiones en 2.9 y 2.10.

2.4 Estudio de distintos USV's

En este apartado se van a presentar una serie de USV's obtenidos de la literatura, y que se han tomado como referencia para el modelado y desarrollo de este proyecto. Algunos se ajustan más que otros a las características de vehículo que se quiere diseñar, pero se considerarán igualmente para con el fin de ampliar el rango de posibilidades disponibles.

SMARTBoat 3:

Tamaño: 28 x 24 x 28 cm (ancho x largo x alto).

Peso: 3 kg.

Velocidad máxima: 40 cm/s.

Autonomía: 30 min.

Motor: PROPDRIVEv2 2826 1000KV Brushless Outrunner.

Configuración: 2 motores longitudinales equidistados respecto al eje de simetría longitudinal.

Matriz T y K: Desconocidas.

Fuente: HardwareX 6 (2019) e00076. [5]

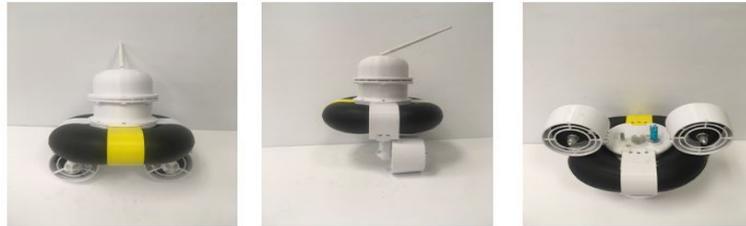


Figura 2.8 SMARTBoat 3.

IEEE Xplore:

Tamaño: 29 x 33 x 21 cm (ancho x largo x alto).

Peso: 1.3 kg.

Velocidad máxima: 60 cm/s.

Autonomía: 90 min.

Motor: Disposable Brushless Motors.

Configuración: 2 motores longitudinales equidistados respecto al eje de simetría longitudinal.

Matriz T: Desconocida.

$$\text{Matriz } K: n = \begin{cases} \frac{-(v/\sqrt{k})^2 \pm 2x^2}{2\sqrt{2}x} & (x < -\frac{v}{\sqrt{2k}}) \\ \frac{(v/\sqrt{k})^2 \pm 2x^2}{2\sqrt{2}x} & (x > \frac{v}{\sqrt{2k}}) \\ \frac{\pm x + \sqrt{(v/\sqrt{k})^2 - x^2}}{\sqrt{2}} & c.c. \end{cases}$$

Fuente: IEEE Xplore (2016). [9]

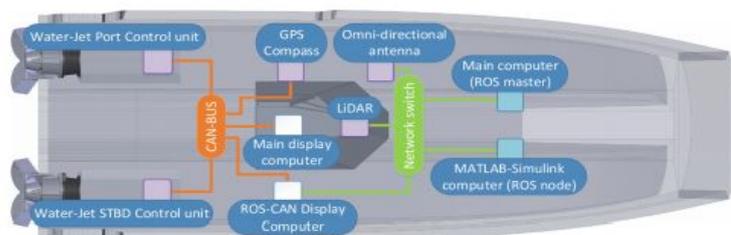


Figura 2.9 IEEE Xplore.

DOI 10.1002:

Tamaño: 150 cm de largo x 90 cm de ancho.

Peso: 20 kg.

Autonomía: 5 horas.

Motor: Brushless Motor.

Configuración: 1 motor longitudinal y 2 aletas equidistadas.

Matriz T y K: Desconocidas.

Fuente: Journal Field of Robotics (2010). [7]

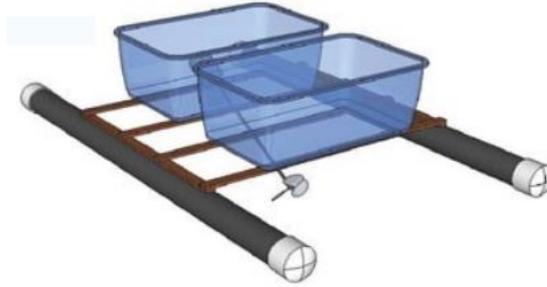


Figura 2.10 DOI 16.1002.

WQMS:

Tamaño: 80 x 180 x 60 cm (ancho x largo x alto).

Peso: 5 kg.

Empuje máximo: 10 - 20 N.

Motor: DC Brushless Motor.

Configuración: 2 motores longitudinales esquinados al eje de simetría.

Matriz T y K: Desconocidas.

Fuente: Water 2020, 12, 681. [3]

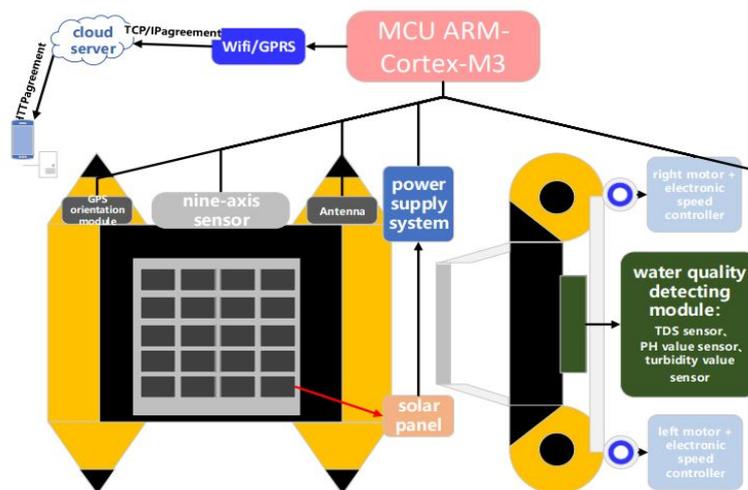


Figura 2.11 WQMS.

ADAM M1:

Tamaño: 82 m de largo x 18 m de ancho.

Peso: 3600 toneladas.

Configuración: 2 propulsores azimutales y 3 transversales.

$$\text{Matriz } T: \begin{bmatrix} \cos \alpha_1 & \cos \alpha_2 & 0 & 0 & 0 \\ \sin \alpha_1 & \sin \alpha_2 & 1 & 1 & 1 \\ -38.6 \sin \alpha_1 - 4.5 \cos \alpha_1 & -38.6 \sin \alpha_2 + 4.5 \cos \alpha_2 & 28.2 & 30.9 & 33.6 \end{bmatrix}$$

Matriz K: Matriz unidad (se modela más tarde).

Fuente: Asociación de Investigación y Cooperación Industrial de Andalucía. [6]

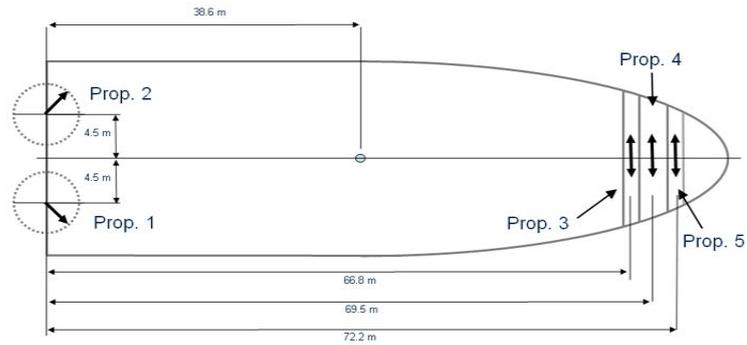


Figura 2.12 ADAM M1.

3 Modelado de vehículos marinos de superficie

Una vez estudiado analíticamente el problema, el primer paso a dar es modelar las físicas y el comportamiento del vehículo marítimo. En el sistema real, este capítulo representaría la configuración de la embarcación, así como las señales externas obtenidas de sus sensores y las perturbaciones a las que es sometida por culpa del viento y el oleaje, entre otros factores. Es por ello que aunque este bloque no vaya integrado en el USV como tal, resulta de crucial importancia diseñarlo de forma apropiada al ser esencial para la correcta simulación del sistema. Cabe destacar también que el modelado se realizará en el plano horizontal 2D, considerando despreciables los desplazamientos verticales y los giros de cabeceo y alabeo.

3.1 Modelado en Simulink

La simulación del sistema real se va a modelar empleando el entorno de Matlab, *Simulink*. Se ha dividido esta en bloques o "módulos", los cuales se pasarán a explicar en más detalle a continuación, junto con un ejemplo completo de una de las configuraciones escogidas:

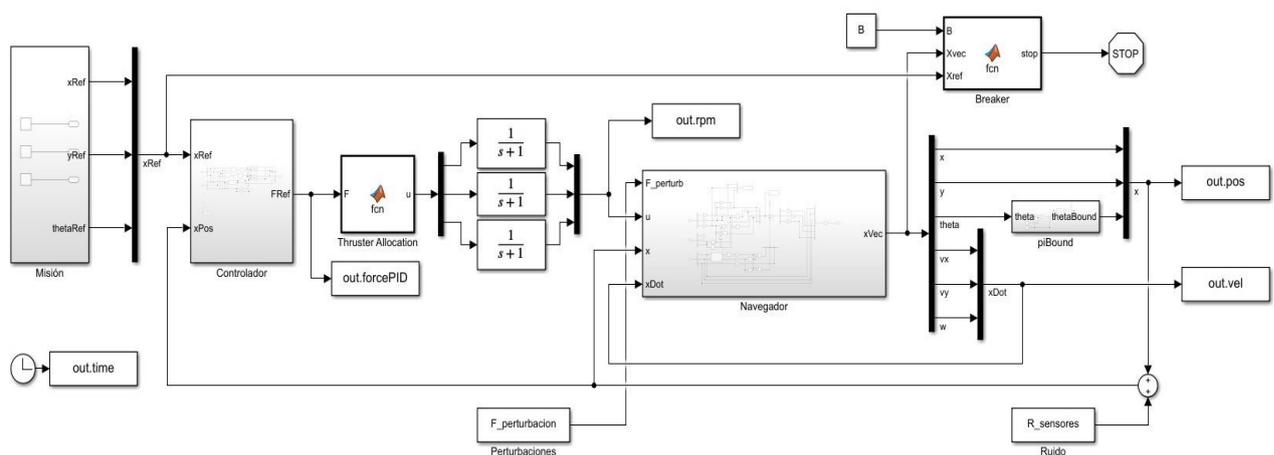


Figura 3.1 Esquema del modelado en Simulink.

Sus componentes son:

- **Misión:** es el módulo al que se le asigna el objetivo en posición y orientación.
- **Controlador:** es el encargado de conseguir que el barco llegue al destino proporcionado por el bloque de misión. Su funcionamiento se estudiará con más detalle en el capítulo 4.

- **Thruster Allocation:** es el módulo encargado de optimizar el problema de actuaciones que ha sido foco de atención en el capítulo 2. Tiene como entrada las fuerzas deseadas proporcionadas por el controlador y como salida las actuaciones requeridas, resultado de la optimización.
- **Funciones de transferencia:** encargadas de simular la variación con el tiempo de los actuadores de forma que esta no ocurra de manera instantánea
- **Navegador:** será el centro de estudio de este capítulo y tiene como función simular las físicas que rigen el movimiento del barco, incluyendo la configuración de actuadores, las fuerzas de resistencia hidrodinámicas y las de perturbación.
- **piBound:** es un módulo que se repite diversas veces a lo largo del programa y que se encarga de mantener todos los ángulos dentro del rango $[\pi, -\pi)$ para que los cálculos sean consistentes.
- **Factores externos:** los componen las fuerzas de perturbación y los ruidos de los sensores, simulando así posibles situaciones que se pueden encontrar en misiones reales.
- **Breaker:** tiene como función detener el programa cuando se alcance el objetivo con un margen de error establecido y una velocidad lineal y angular nula.
- **Reloj:** registra el tiempo de la simulación.

El esquema que se muestra en la figura 3.1 es para un modelo de embarcación que presenta dos propulsores longitudinales a ambos lados de la popa y uno transversal en el casco de proa, y es el que se usará como ejemplo durante este capítulo, previo a presentar las distintas configuraciones que se han desarrollado en este estudio.

3.2 Navegador

El módulo de navegación toma como entrada las actuaciones, posiciones y velocidades a tiempo real, y devuelve como salida el vector de estado que define la posición y la velocidad actualizadas con el tiempo. A continuación se pasará a explicar las distintas partes que componen el módulo de navegación modelado en Simulink representado en la figura 3.2:

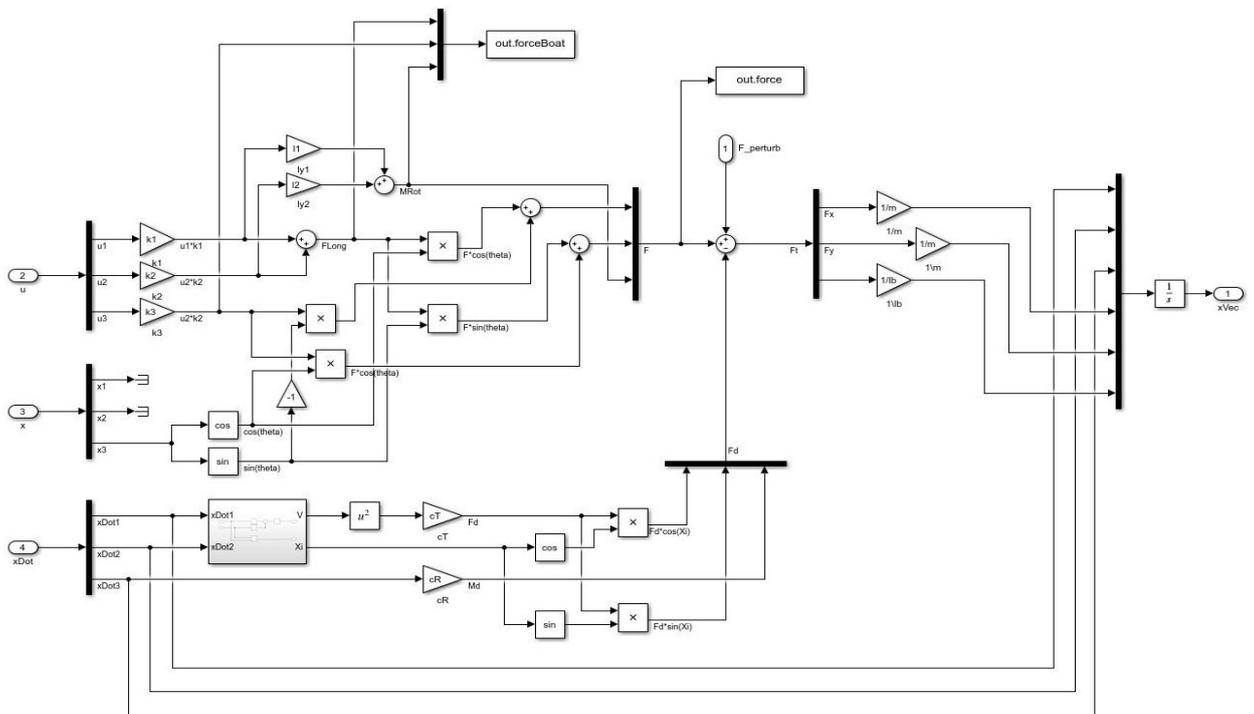


Figura 3.2 Módulo de Navegación en Simulink.

3.2.1 Matriz de configuración

Constituye el cuerpo principal del navegador, y funciona numéricamente de manera inversa al módulo de Thruster Allocation. Su trabajo es tomar las rpm's y ángulos de deflexión de timones y propulsores azimutales para calcular las fuerzas y momento resultantes sobre el centro de gravedad del barco.

Es necesario también introducir el ángulo de orientación del barco $\theta(t)$ para hacer el giro de ejes que nos permitirá pasar de ejes barco (eje x en la dirección popa-proa y eje y en la dirección estribor-babor) a ejes globales (ejes x e y asociados a la horizontal y la vertical del plano respectivamente)

$$G_{b \rightarrow g} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

Los parámetros del barco que interesan en este apartado son los coeficientes de empuje k_i obtenidos del estudio analítico en 2.1 y las posiciones geométricas de los actuadores respecto al centro de gravedad. En el esquema 3.2 viene representado en la esquina superior izquierda.

3.2.2 Fuerzas externas

Es igual de necesario modelar las fuerzas externas que actúan sobre la embarcación, de las que se han considerado las de resistencia hidrodinámica y las de perturbación por corriente y oleaje. Las fuerzas de perturbación vienen sumadas directamente a las fuerzas ejercidas por el barco en ejes globales, mientras que las fuerzas de resistencia requieren de un modelado ligeramente más elaborado.

Las fuerzas de resistencia hidrodinámicas se oponen siempre al movimiento y se ejercen en la dirección de este. Puesto que el USV puede tener un vector velocidad que no coincida en orientación con la del mismo, debemos definir este nuevo ángulo ($\chi \neq \theta$) para obtener la dirección de las fuerzas.

$$V = \sqrt{v_x^2 + v_y^2} \quad (3.2)$$

$$\chi = \text{atan2}(v_y, v_x) \quad (3.3)$$

La resistencia al movimiento lineal se ha definido como un valor proporcional al cuadrado de la velocidad mediante un coeficiente de resistencia lineal c_T , mientras que la resistencia al giro se ha definido como proporcional a la velocidad de giro mediante el coeficiente de resistencia angular c_R .

Esta sección viene representado en el esquema 3.2 en la parte inferior.

3.2.3 Vector de estado

Una vez obtenidas todas las fuerzas resultantes que actúan sobre la embarcación, ya es posible integrar estas para obtener el *vector de estado*, para lo que son imprescindibles los parámetros de la masa m y la inercia I_b . Con esto se obtienen las aceleraciones lineales y angular que, con las velocidades que se tenían ya definidas, podemos crear la *derivada del vector estado* \dot{X} .

Integrando este vector y añadiendo las condiciones iniciales de nuestra simulación (X_0) se obtiene finalmente el vector X que será el que contenga las posiciones y velocidades tanto lineales como angulares. Este apartado viene representado en la zona derecha del esquema 3.2.

$$\dot{X} = \begin{bmatrix} v_x \\ v_y \\ \dot{\theta} \\ a_x \\ a_y \\ \vdots \\ \dot{\theta} \end{bmatrix} \xrightarrow[\frac{1}{s} + c.c.]{\text{---}} X = \begin{bmatrix} x \\ y \\ \theta \\ v_x \\ v_y \\ \vdots \\ w = \dot{\theta} \end{bmatrix} \quad (3.4)$$

3.2.4 Modelado en Matlab

El código que se muestra a continuación es otra forma de modelar el bloque de navegación, pero esta vez en Matlab, de manera que queda más recogida y compactada la información, aunque no sea tan visual.

Código 3.1 Módulo de Navegación en Matlab.

```

% Fuerzas del barco
tau = [u(1) u(2) u(3)]';
T = [1 1 0;
     0 0 1;
     -ly1 -ly2 lx3];
K = [k1 0 0;
     0 k2 0;
     0 0 k3];
Fb = T*K*tau;

% Matriz de giro
G = [cos(x(3)) -sin(x(3)) 0;
     sin(x(3)) cos(x(3)) 0;
     0 0 1];
Fg = G*Fb;

% Fuerzas de Resistencia
V = sqrt(xDot(1)^2+xDot(2)^2);
Xi = atan2(xDot(2),xDot(1));

Fdx = V^2*cT*cos(Xi);
Fdy = V^2*cT*sin(Xi);
Md = xDot(3)*cR;
Fd = [Fdx Fdy Md]';

% Fuerzas totales
F = Fg-Fd+Fp;
for i=1:2
    if abs(F(i))<0.05
        F(i)=0;
    end
end

% Vector xVecDot
xVecDot(1) = xDot(1);
xVecDot(2) = xDot(2);
xVecDot(3) = xDot(3);
xVecDot(4) = F(1)/m;
xVecDot(5) = F(2)/m;
xVecDot(6) = F(3)/Ib;

% Resultados
forceBoat = Fb;
force = F;
xVecDot = xVecDot(:);

```

Se observa que la base es exactamente la misma, aunque se simplifica el cálculo de las fuerzas resultantes sobre el centro de gravedad del barco al poder emplear directamente la matriz de configuración sobre las actuaciones.

3.3 Configuraciones de USV's

Para finalizar este capítulo se van a presentar las distintas configuraciones de USV's con las que se van a trabajar en el capítulo 4, junto con los valores numéricos de los parámetros definidos anteriormente. Se han estudiado previamente una amplia gama de posibilidades, desde los más sencillos a los más complejos.

Algunas simples, como la típica configuración que se encuentra en numerosas embarcaciones consistente en un motor longitudinal y un timón, se han descartado debido a que presenta un acoplamiento en la generación de las fuerzas, dependiendo la fuerza directamente de la orientación del barco, y siendo el giro del mismo dependiente de su velocidad, lo que supone una gran complicación a la hora de diseñar un controlador que proporcione soluciones estables.

Otras más complejas, como la del ADAM M1, incluyen una gran complejidad numérica en el apartado de thruster allocation debido a la gran redundancia de sus actuadores, por lo que también se dejará fuera de este estudio, aunque se proporcionarán algunos ejemplos para comprender el alcance de su diseño.

Concretamente se van a tomar 4 configuraciones distintas de USV's, las cuales se van a presentar a continuación:

3.3.1 1ª Configuración: 2 propulsores longitudinales

Esta es otra de las configuraciones más básicas que suelen encontrarse en embarcaciones y vehículos de superficie. Dispone de dos propulsores longitudinales que se encuentran a una distancia l_y , cada uno del centro de gravedad en el eje y .

Tabla 3.1 Parámetros de la 1ª Configuración.

	m	I_b	F_{max}	l_{y1}	l_{y2}	k_1	k_2	c_T	c_R
Valor	10	5	25	-0.1	0.1	0.0125	0.0125	0.25	0.75

Supone un problema a la hora de controlarlo el hecho de que no dispone de ninguna forma de ejercer una fuerza en el eje y del barco, por lo que la trayectoria y la orientación de este van ligados, incrementando de esta manera la complejidad.

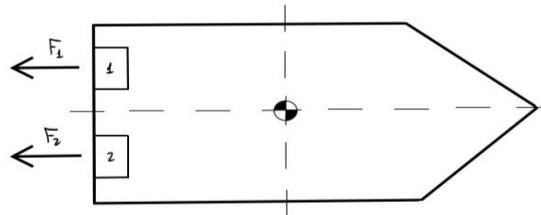


Figura 3.3 Configuración de actuadores N°1.

3.3.2 2ª Configuración: 2 propulsores longitudinales y 1 transversal

Similar a la configuración anterior pero añadiéndole un tercer propulsor, esta vez transversal, en la proa del casco a una distancia l_x del centro de gravedad en el eje x .

Tabla 3.2 Parámetros de la 2ª Configuración.

	m	I_b	F_{max}	l_{y1}	l_{y2}	l_{x3}	k_1	k_2	k_3	c_T	c_R
Valor	10	5	10	-0.1	0.1	0.2	0.0125	0.0125	0.0125	0.25	0.75

Presenta una ventaja a la hora de su control debido a que en este caso las fuerzas en el eje x y en el eje y están desacopladas, permitiendo un movimiento puramente lateral sin la necesidad de orientar el barco en la dirección de la trayectoria.

Este desacoplamiento permitirá optimizar su rumbo de manera más sencilla y facilitar el desarrollo de un controlador que le permita funcionar de forma estable y precisa.

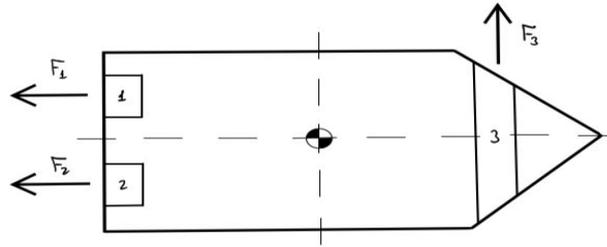


Figura 3.4 Configuración de actuadores N°2.

3.3.3 3ª Configuración: 2 propulsores azimutales

Esta tercera configuración coge los mismos parámetros que la primera, pero sustituyendo los dos propulsores longitudinales por dos propulsores azimutales, situados ambos en la popa del casco y a una distancia l_y del eje longitudinal del barco y l_x del transversal respecto a su centro de gravedad.

Tabla 3.3 Parámetros de la 3ª Configuración.

	m	I_b	F_{max}	l_{x1}	l_{y1}	l_{x2}	l_{y2}	k_1	k_2	c_T	c_R
Valor	10	5	15	-0.1	-0.1	-0.1	0.1	0.0125	0.0125	0.25	0.75

Este cambio permite una completa maniobrabilidad del mismo al poder orientar los propulsores en cualquier dirección, permitiendo así que se desplace en el sentido que se desee.

Aunque para casos reales se deberían acotar los rangos de ángulos admisibles de los propulsores, para este modelo simplificado se asumirá que estos son capaces de generar empuje en cualquier dirección que se desee, ignorando la zona prohibida.

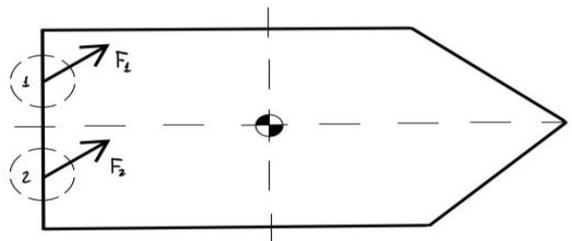


Figura 3.5 Configuración de actuadores N°3.

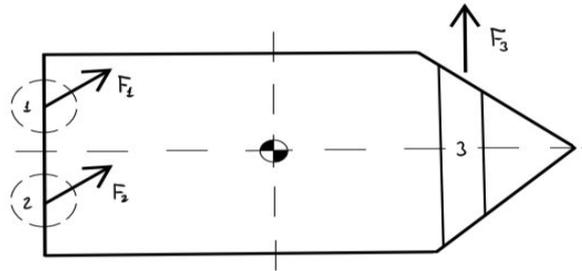
3.3.4 4ª Configuración: 2 propulsores azimutales y 1 transversal

Esta última presenta las ventajas de tanto la segunda configuración como de la tercera, poseyendo dos propulsores azimutales y un último propulsor transversal situados todos en posiciones similares a las de sus referencias homólogas.

Tabla 3.4 Parámetros de la 4ª Configuración.

	m	I_b	F_{max}	l_{x1}	l_{y1}	l_{x2}	l_{y2}	l_{x3}	k_1	k_2	k_3	c_T	c_R
Valor	10	5	10	-0.15	-0.1	-0.15	0.1	0.2	0.0125	0.0125	0.0125	0.25	0.75

Se obtiene así un sistema sobreactuado con redundancia de controladores, lo que implica una necesidad de optimizar correctamente su uso debido a la amplia gama de posibilidades de control, así como un aumento de la seguridad en caso de que halla un fallo de algún motor.

**Figura 3.6** Configuración de actuadores N°4.

4 Diseño y desarrollo de controladores

Llegados a este punto, lo que resta para terminar de modelar completamente el vehículo de superficie es diseñar un controlador que transforme los errores en posición y orientación, asignados por la misión, en las fuerzas necesarias que regirán el movimiento de este. La clave para la programación de un buen controlador es asegurar que opere el sistema de forma estable, que sea lo más eficiente y económico posible, y que sea capaz de reajustar la trayectoria del USV en caso de que fuerzas externas lo agiten y lo saquen de su rumbo. Este apartado es de los más simples a nivel conceptual, pero no por ello es menos complejo en lo que respecta a su implementación en Simulink debido a las consideraciones de carácter numéricas que esta acarrea.

4.1 Evolución del diseño

En primer lugar se va a mostrar el proceso mediante el cual se han ido diseñando distintos tipos de controladores en función de diversos parámetros y medidas de error.

Todos tienen como base el bloque *PID* de Simulink, entorno al cual se van a definir el resto de variables del problema para ajustarlo a las necesidades del posicionamiento y rumbo. Se presentarán todos los modelos considerados, aunque para las simulaciones solo se apliquen los que demuestren ser los óptimos.

4.1.1 Modelo cinemático

Este primer modelo constituye una prueba anterior a la construcción del resto del sistema, utilizando como modelo de barco el proporcionado en el artículo [8], el cual dispone de un propulsor longitudinal y un timón de dirección, y nos proporciona una relación directa entre la deflexión del timón con la velocidad de rotación ($\delta(t) - w(t)$) y las rpm's del motor con la velocidad del vehículo ($rpm(t) - V(t)$)

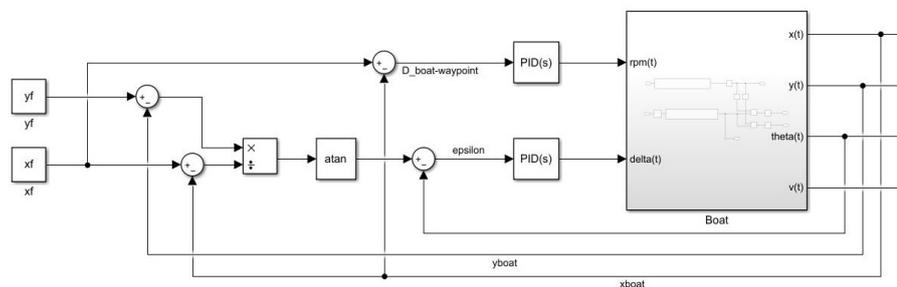


Figura 4.1 Modelo de controlador cinemático.

Se observa que la base de este controlador está en asociar la velocidad del USV con la distancia respecto al punto de destino, y forzar a que este se encuentre siempre orientado hacia el mismo. Los errores en ambos de estas magnitudes son los que definen las rpm's del motor y el ángulo de deflexión del timón.

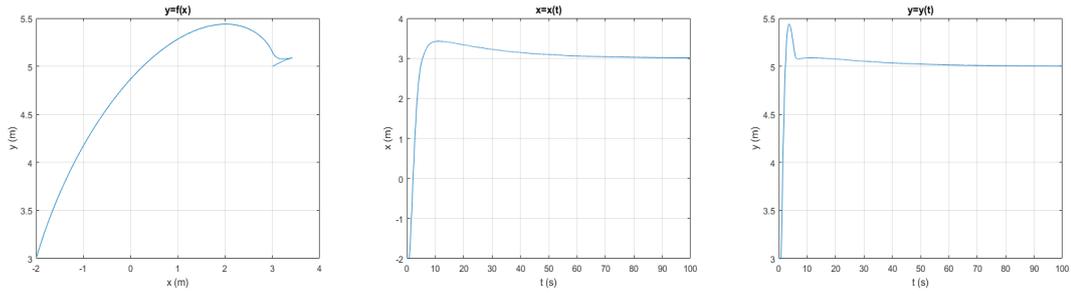


Figura 4.2 Trayectoria para una misión desde [0,0] hasta [3,5].

Este modelo demuestra alcanzar el objetivo de la misión de forma estable y rápida, por lo que se podrá emplear como base de algunos de los siguientes diseños. Su único inconveniente es no controlar la orientación final del vehículo.

4.1.2 Modelo dinámico N°1

El camino lógico a seguir sería modelar un controlador similar al anterior visto su buen funcionamiento, pero en este caso para un sistema dinámico real. El esquema para un vehículo cualquiera viene representado a continuación:

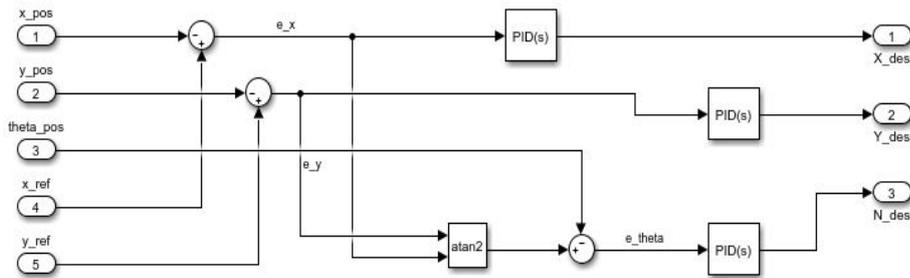


Figura 4.3 Modelo de controlador dinámico N°1.

Lamentablemente, la incorporación de este mismo en un sistema dinámico no tiene los mismos buenos resultados que en el sistema cinemático. En parte se puede deber a la necesidad de controlar más variables (como la saturación de los actuadores y la relación entre los distintos ángulos y ejes de referencia que se consideran en el problema), lo que lo descartará como posible controlador, aunque pueda ser sometido a ciertas modificaciones para mejorar su funcionamiento.

4.1.3 Modelo dinámico N°2

Para este segundo modelo de controlador se va a intentar abordar el problema desde otro ángulo. La base sobre el que se registrará el movimiento será el de desplazarse pegado a la línea que une el punto de partida y el objetivo.

Así, los dos errores que se van a controlar son la distancia del USV a la recta, d_e , y la diferencia entre la pendiente de la recta y el ángulo de orientación del mismo, δ . Estos parámetros vienen definidos por las expresiones mostradas a continuación:

$$d_e = \frac{Ax + By + C}{\sqrt{A^2 + B^2}} \tag{4.1}$$

$$\delta_e = \arctan\left(\frac{y_f - y_0}{x_f - x_0}\right) - \theta \tag{4.2}$$

donde A, B y C son los parámetros que definen la ecuación general de la recta que une los puntos inicial (x_0, y_0) y final (x_f, y_f) .

Código 4.1 Algoritmo de seguimiento de una recta.

```
% Parámetros de la recta
A = yf-y0;
B = x0-xf;
C = y0*xf-yf*x0;
omega = atan2((yf-y0), (xf-x0));

% Error en posición (distancia del barco a la recta)
D_e = (A*x+B*y+C)/sqrt(A^2+B^2);

% Error en ángulo
delta_e = omega-theta;
```

Así, se asignará el error en posición d_e a la fuerza longitudinal y transversal del vehículo, mientras que la orientación del mismo será gobernada por el error en ángulo δ_e , corregida dependiendo de la orientación del barco en respecto a la recta.

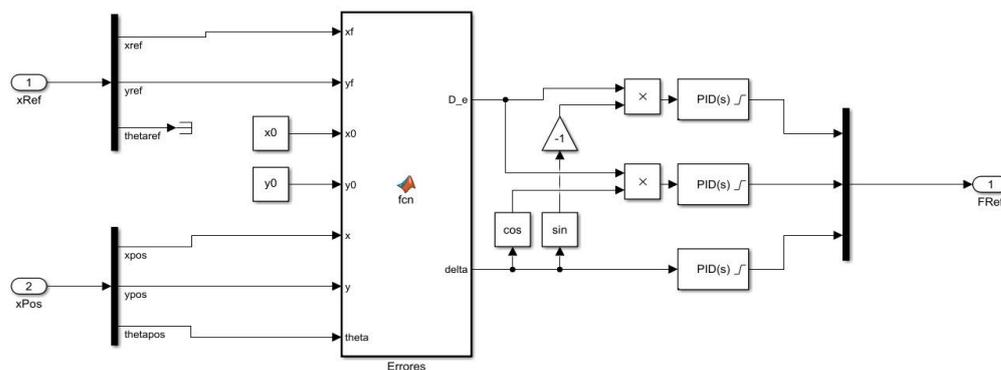


Figura 4.4 Modelo de controlador dinámico N°2.

Este controlador, aunque efectivo, presenta algunas dificultades en su diseño por trabajar con valores absolutos a la hora de determinar la distancia a la recta, por lo que solo será aplicable en algunos tipos de trayectoria.

4.1.4 Modelo dinámico N°3

El siguiente controlador va a tomar un enfoque similar al anterior, pero difiriendo ligeramente en concepto. La idea será controlar el movimiento del USV mediante errores en coordenadas polares, por lo que los parámetros a definir son las coordenadas polares tanto del vehículo (X_C) como del punto de destino (X_N):

$$R_N = \sqrt{(x_f - x_0)^2 + (y_f - y_0)^2} \quad (4.3)$$

$$\Omega_N = \arctan\left(\frac{y_f - y_0}{x_f - x_0}\right) \quad (4.4)$$

$$R_C = \sqrt{(x - x_0)^2 + (y - y_0)^2} \quad (4.5)$$

$$\Omega_C = \arctan\left(\frac{y - y_0}{x - x_0}\right) \quad (4.6)$$

Siendo los errores con los que se trabajan:

$$d_e = R_N - R_C \tag{4.7}$$

$$\delta_e = \Omega_N - \Omega_C \tag{4.8}$$

donde δ_e es la diferencia en la componente radial entre el USV y el punto de destino y d_e es la diferencia en la componente angular.

Código 4.2 Algoritmo de posicionamiento en polares.

```

% Coordenadas nominales
RN = sqrt((xf-x0)^2+(yf-y0)^2);
omegaN = atan2((yf-y0),(xf-x0));

% Coordenadas del USV
RC = sqrt((x-x0)^2+(y-y0)^2);
omegaC = atan2((y-y0),(x-x0));

% Errores polares
D_e = RN-RC;
delta_e = omegaN-omegaC;
    
```

Se asocia por tanto la coordenada radial a la fuerza longitudinal del barco, mientras que la coordenada angular viene controlada por la orientación de este. Deberán considerarse los distintos casos posibles de condiciones de cruce para evitar que surjan errores o inestabilidades.

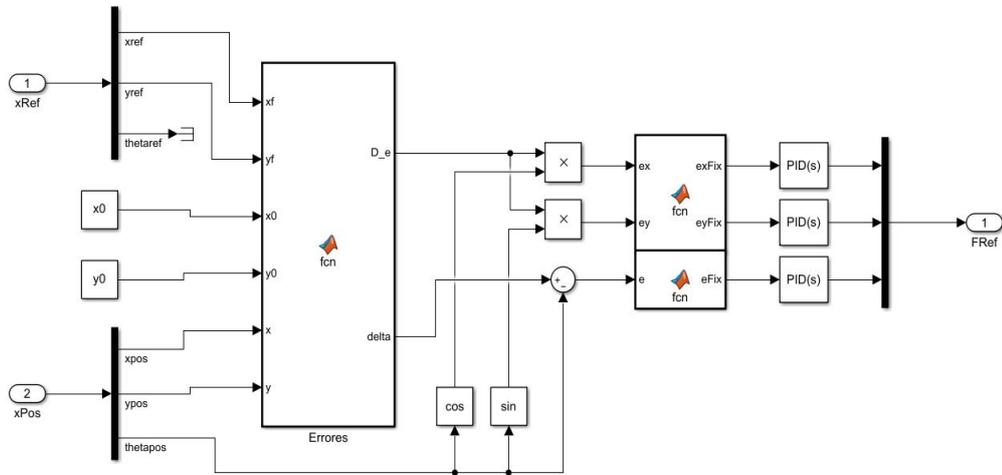


Figura 4.5 Modelo de controlador dinámico N°3.

Este controlador resulta menos efectivo que el anterior debido a la dificultad añadida de trabajar entre ejes cartesianos y polares. También resulta complejo corregir el rumbo del vehículo en función de los parámetros del problema (de lo que se encargan las funciones de Matlab de la derecha en la figura 4.5), por lo que no se considerará de gran utilidad ni se empleará en ningún caso.

4.1.5 Modelo dinámico N°4

Para este controlador se va a probar un sistema de guiado basado en dividir el posicionamiento del vehículo en distintas etapas separadas, de forma que cada una de ellas se pueda programar de forma sencilla. En este caso, y teniendo en mente la configuración de motores mas sencilla posible (2 motores longitudinales), se va a dividir en 3 etapas:

- **1º Fase:** esta primera fase constituye la orientación del USV de forma que la proa del barco apunte hacia el punto al cual se quiere viajar. Durante esta fase no se produce ningún desplazamiento.

$$\delta_{e1} = \arctan\left(\frac{y_f - y_0}{x_f - x_0}\right) - \theta \tag{4.9}$$

- **2º Fase:** a continuación, y una vez con el barco orientado al destino y sin velocidad angular, se avanza en línea recta empleando exclusivamente los propulsores longitudinales hasta detenerse en el punto proporcionado por la misión.

$$d_e = \sqrt{(x_f - x)^2 + (y_f - y)^2} \tag{4.10}$$

- **3º Fase:** finalmente, y sin moverse del sitio, se vuelve a orientar el USV en la dirección deseada antes de terminar la misión.

$$\delta_{e2} = \theta_f - \theta \tag{4.11}$$

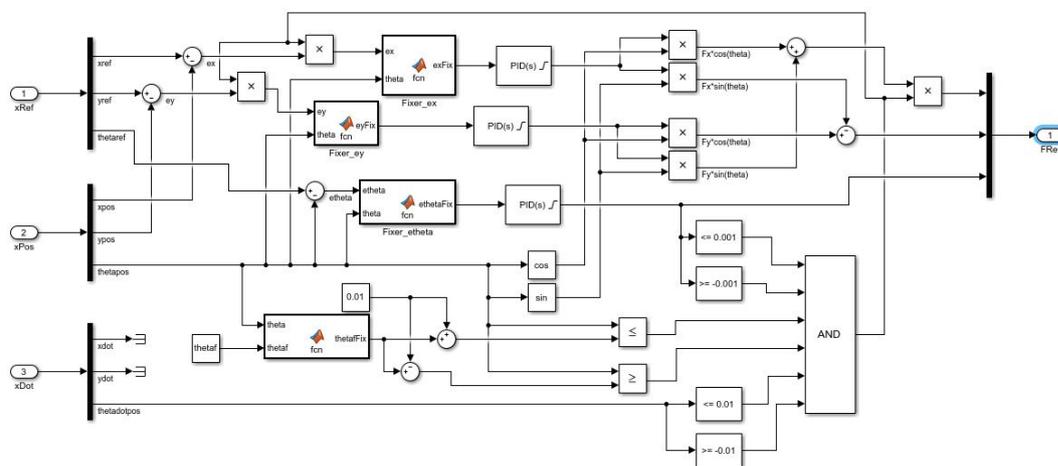


Figura 4.6 Modelo de controlador dinámico N°4.

Aunque en la figura 4.6 parece que el sistema es muy complejo, en realidad una gran fracción del espacio (parte inferior) está ocupado por el trozo de código encargado de comprobar las condiciones necesarias para pasar de una fase a otra (bloque AND).

Este controlador muestra resultados bastante positivos, aunque el tiempo que tarda en llevar a cabo la misión podría claramente reducirse ya que, al hacer cada maniobra por separado, gasta mucho más tiempo en total de lo que lo haría si estas estuvieran acopladas. Para una configuración con propulsor transversal, tanto la primera fase como la tercera podrían fusionarse como una sola.

4.1.6 Modelo dinámico N°5

En este último diseño de controlador se van a tomar como referencia los ejes globales, similar al modelo cinemático y primer modelo dinámico modelado, pero con algunas ligeras modificaciones. De primeras, se van a considerar por separado los errores de posición en el eje x (e_x) y en el eje y (e_y). También se va a tratar el error en orientación por su cuenta (e_θ)

Otro factor que se tendrá en cuenta será el paso de ejes locales (USV) a ejes globales, que se realizará directamente en el bloque del controlador para asociar directamente los errores globales con las actuaciones requeridas en ejes locales.

Al trabajar en ejes globales se deben tener en cuenta las posiciones y orientaciones relativas entre el barco y el objetivo, con la finalidad de que el controlador no interprete mal las referencias y el sistema se vuelva inestable.

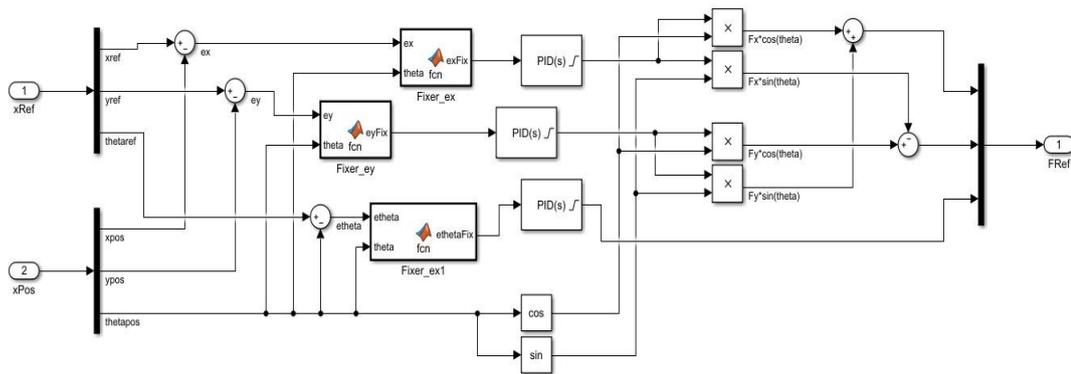


Figura 4.7 Modelo de controlador dinámico N°5.

Se aprecia en la figura 4.7 que este modelo parece más simple, aunque el código incluido en las funciones de Matlab que se muestran no es menos complejo que en cualquier de los otros casos, habiendo supuesto un gran desafío para hacer funcionar correctamente este controlador. También se observa en la zona derecha el cambio de ejes de locales a globales.

Este controlador es, con diferencia, el que ha mostrado mejores resultados en términos tanto de tiempo como de precisión, y es por ello que va a ser el que se incorpore en todas las configuraciones que se van a testear.

4.2 Resultados de pruebas

Finalmente se van a llevar a cabo las simulaciones sobre las distintas configuraciones de propulsores y modelos de USV que se han ido desarrollando a lo largo de este estudio. Para cada caso, se mostrarán los parámetros de entrada proporcionados a Matlab y la evolución de las variables de salida a lo largo del tiempo que dura la simulación.

Todas las simulaciones se han realizado con el último controlador mostrado en el apartado anterior (4.1.6), y se consideran finalizadas (empleando el módulo *Breaker*) cuando el USV se sitúa dentro de un radio de 0.25 metros del objetivo y las velocidades tanto lineales como angulares se encuentran por debajo de un margen de 10 centímetros y 0.01 radianes por segundo respectivamente.

Se pasan a presentar los resultados a continuación:

4.2.1 Simulación N°1

La misión llevada a cabo con la primera configuración de propulsores tiene las condiciones iniciales y finales mostradas en la siguiente tabla:

Tabla 4.1 Parámetros de la Simulación N°1.

	x_0	y_0	θ_0	v_{x0}	v_{y0}	w_0	x_f	y_f	θ_f
Valor	0	0	0	0	0	0	3	3	$\pi/2$

Y los valores de los parámetros de los controladores PID (*Proporcional, Integrador, Derivativo y Coeficiente de filtro*) son:

Tabla 4.2 Parámetros del controlador (S.1).

PID	F_x	F_y	M
P	133.634	133.634	24.292
I	2.781	2.781	1.952
D	1156.624	1156.624	67.568
N	355.736	355.736	286.291

Los resultados son los siguientes:

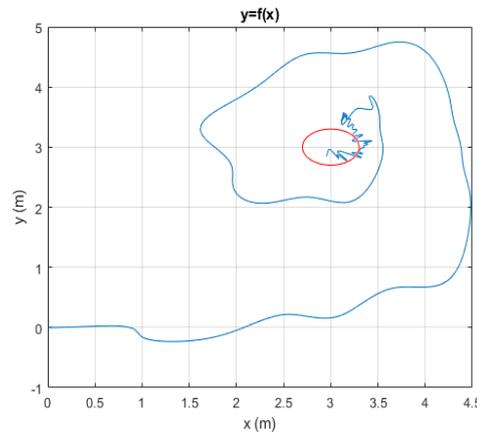


Figura 4.8 Trayectoria seguida en el plano xy (S.1).

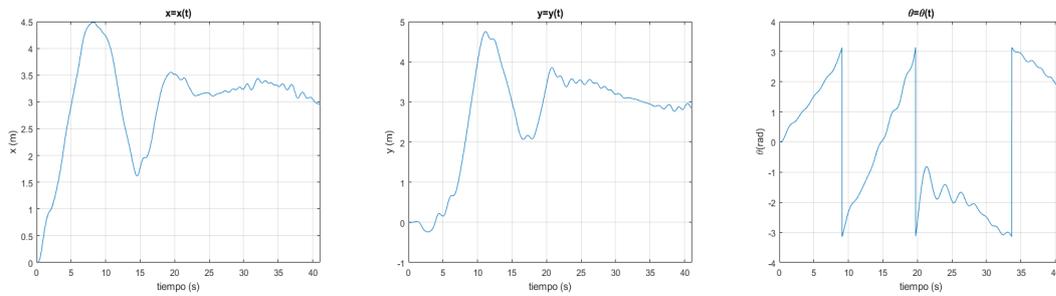


Figura 4.9 Evolución de la posición y la orientación (S.1).

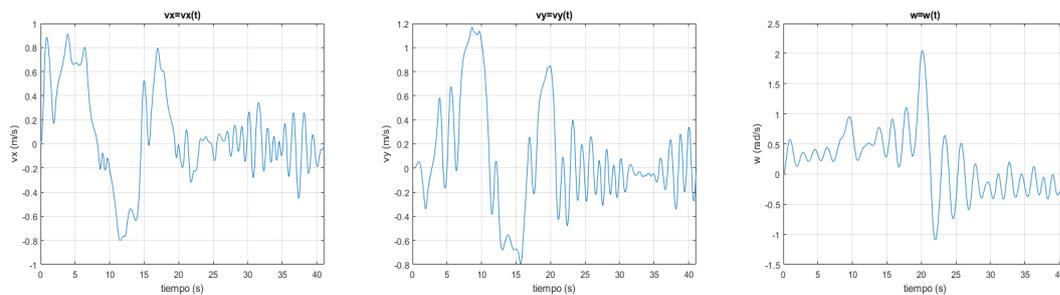


Figura 4.10 Evolución de las velocidades lineales y angular (S.1).

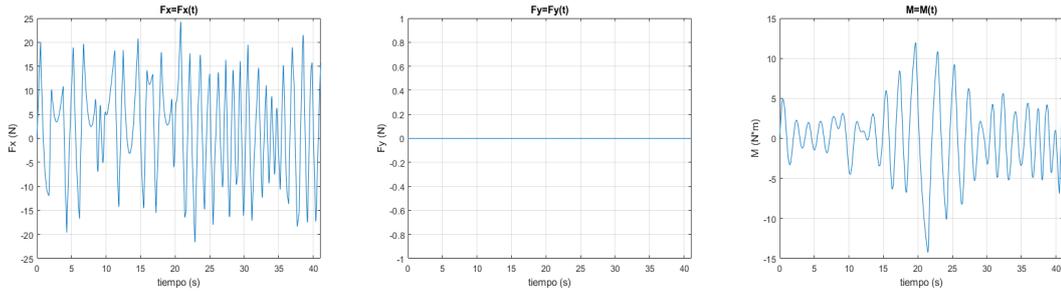


Figura 4.11 Evolución de las fuerzas y momento (S.1).

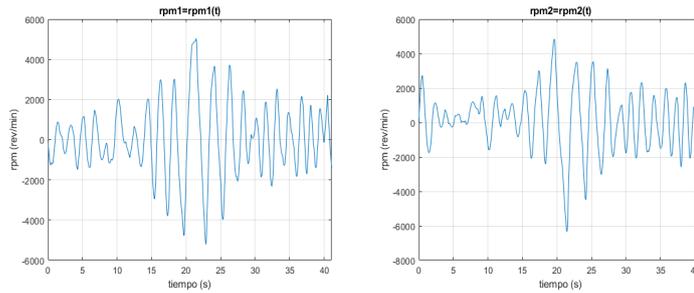


Figura 4.12 Evolución de las rpm's de los propulsores (S.1).

Se pueden apreciar en las gráficas que las fuerzas proporcionadas y, por tanto el movimiento, oscilan de manera muy agresiva, generando continuamente una gran cantidad de reajustes. Aunque el movimiento parezca un poco errático, alcanza el objetivo en un tiempo bastante bueno, siendo el único inconveniente que no posee mucha precisión a la hora de orientar el USV hacia su objetivo final.

4.2.2 Simulación N°2

La misión llevada a cabo con la segunda configuración de propulsores tiene las condiciones iniciales y finales mostradas en la siguiente tabla:

Tabla 4.3 Parámetros de la Simulación N°2.

	x_0	y_0	θ_0	v_{x0}	v_{y0}	w_0	x_f	y_f	θ_f
Valor	1	2	0	0	0	0	-5	5	$\pi/2$

Y los valores de los parámetros de los controladores PID son:

Tabla 4.4 Parámetros del controlador (S.2).

PID	F_x	F_y	M
P	0.328	0.803	0.187
I	6.965e-3	2.507e-2	3.880e-3
D	3.739	6.378	1.432
N	4.036	6.823	30.951

Los resultados son los siguientes:

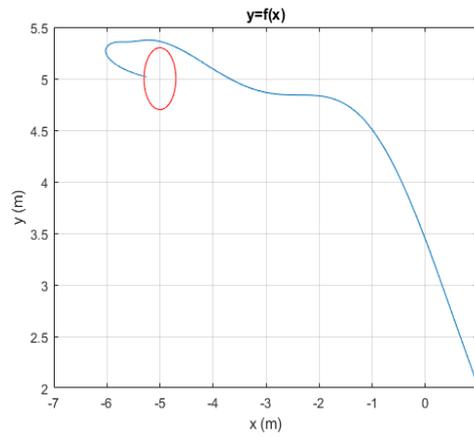


Figura 4.13 Trayectoria seguida en el plano xy (S.2).

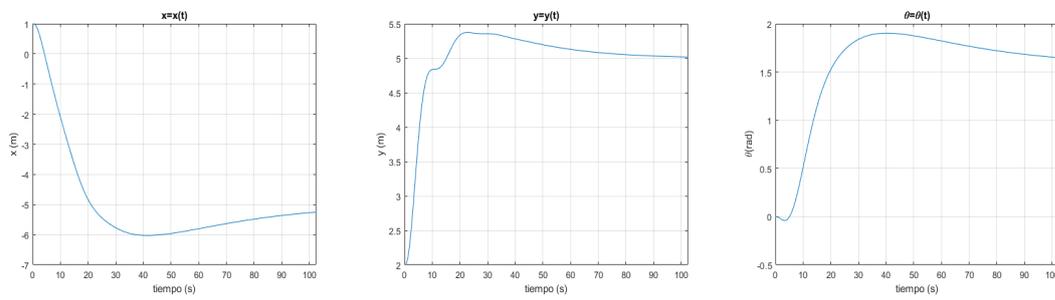


Figura 4.14 Evolución de la posición y la orientación (S.2).

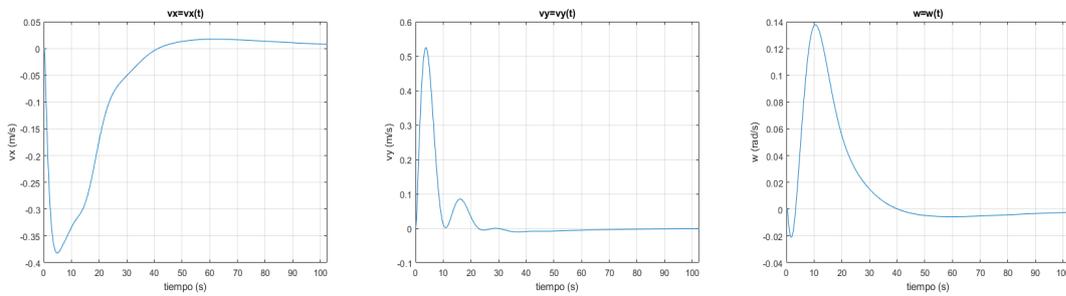


Figura 4.15 Evolución de las velocidades lineales y angular (S.2).

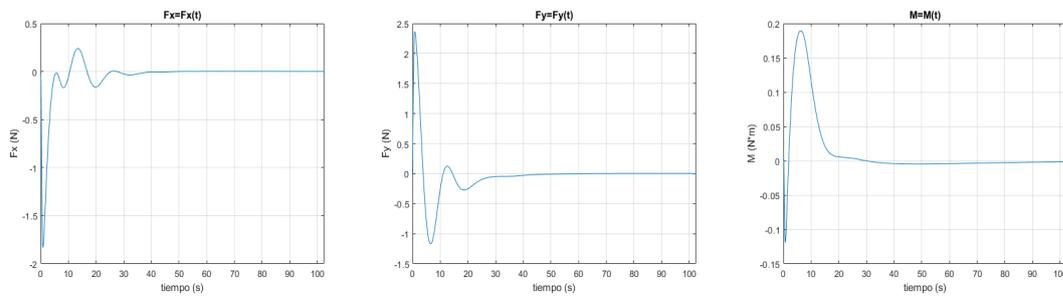


Figura 4.16 Evolución de las fuerzas y momento (S.2).

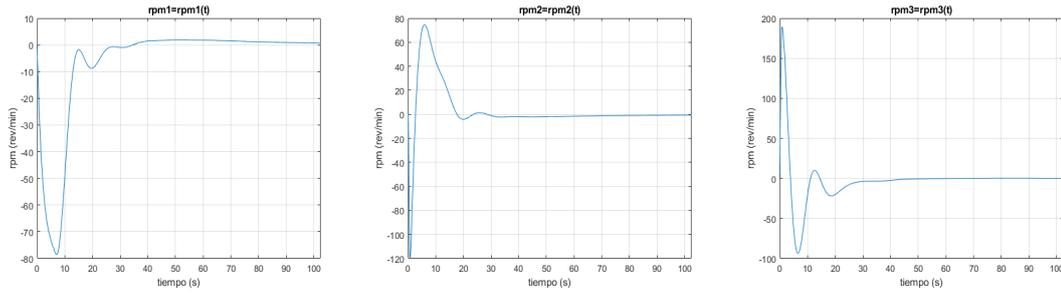


Figura 4.17 Evolución de las rpm's de los propulsores (S.2).

Se puede apreciar una trayectoria mucho más limpia y precisa que en caso anterior. Esto se debe principalmente a que el USV obtiene la capacidad de moverse en el eje x e y de forma independiente.

4.2.3 Simulación N°3

La misión llevada a cabo con la tercera configuración de propulsores tiene las condiciones iniciales y finales mostradas en la siguiente tabla:

Tabla 4.5 Parámetros de la Simulación N°3.

	x_0	y_0	θ_0	v_{x0}	v_{y0}	w_0	x_f	y_f	θ_f
Valor	0	0	$\pi/4$	0	-0.5	0.1	1	6	π

Y los valores de los parámetros de los controladores PID son:

Tabla 4.6 Parámetros del controlador (S.3).

PID	F_x	F_y	M
P	0.478	0.953	0.187
I	$7.975e-3$	$2.795e-2$	$3.182e-3$
D	3.124	3.378	1.432
N	1.036	5.792	25.711

Los resultados son los siguientes:

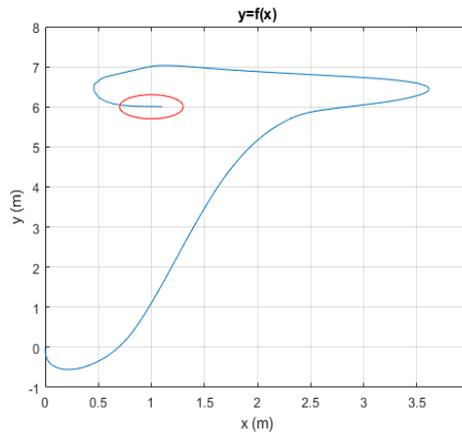


Figura 4.18 Trayectoria seguida en el plano xy (S.3).

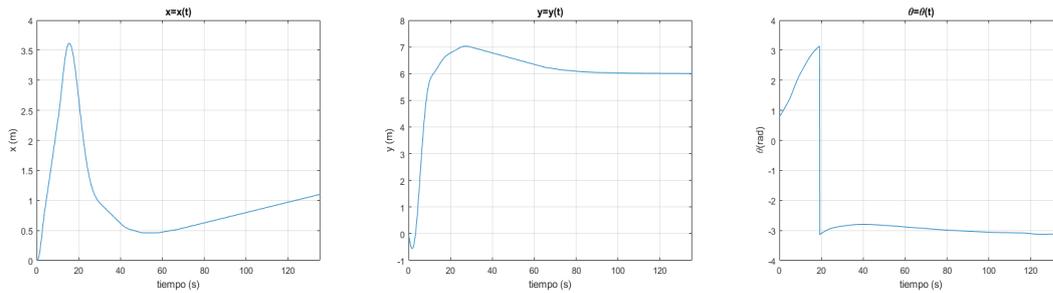


Figura 4.19 Evolución de la posición y la orientación (S.3).

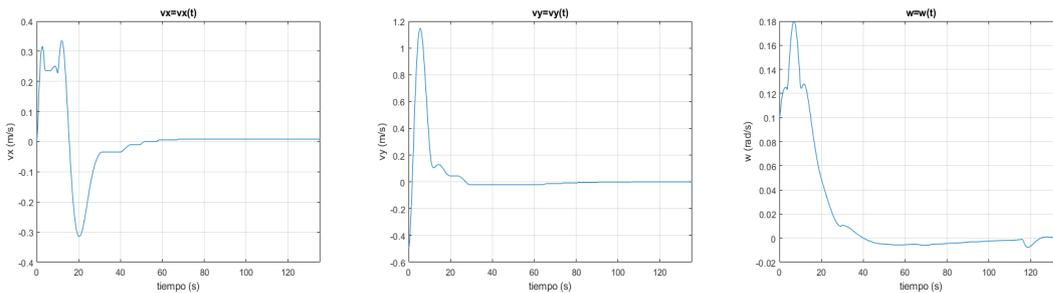


Figura 4.20 Evolución de las velocidades lineales y angular (S.3).

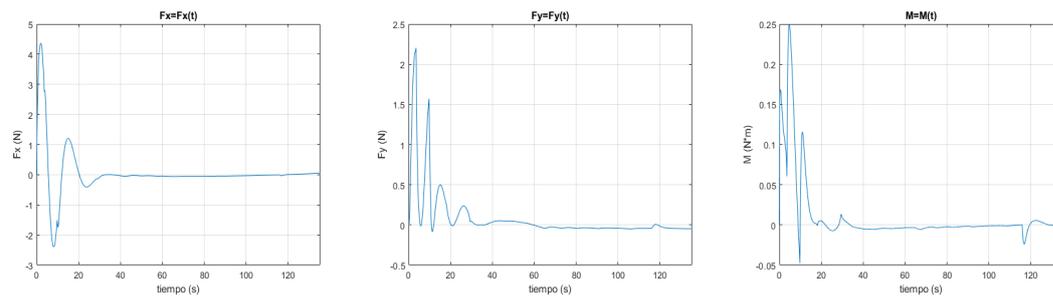


Figura 4.21 Evolución de las fuerzas y momento (S.3).

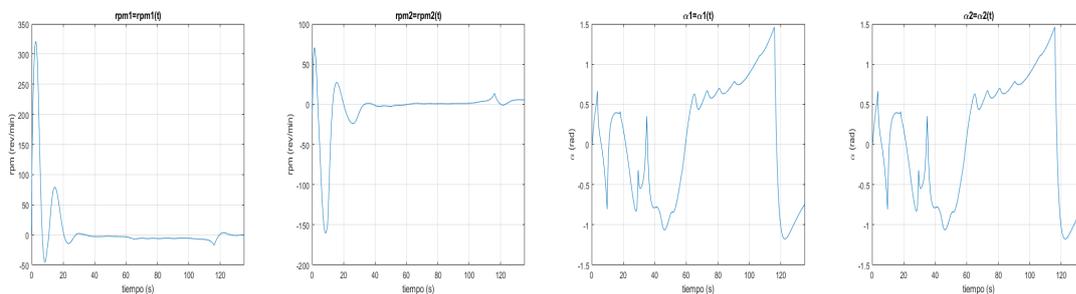


Figura 4.22 Evolución de las rpm's de los propulsores y orientaciones de los azimutales (S.3).

Aunque existe una sobreoscilación considerable en el eje x , esto puede deberse a la cercanía de la coordenada x_f respecto a la inicial x_0 , por lo que actuará de forma más rápida o lenta dependiendo del tipo de misión. En general, se observa un buen comportamiento y una habilidad para rotar el USV sin modificar el rumbo de este, lo que supone un gran alivio de tiempo.

4.2.4 Simulación N°4

La misión llevada a cabo con la cuarta configuración de propulsores tiene las condiciones iniciales y finales mostradas en la siguiente tabla:

Tabla 4.7 Parámetros de la Simulación N°4.

	x_0	y_0	θ_0	v_{x0}	v_{y0}	w_0	x_f	y_f	θ_f
Valor	1	2	0	0	0	0	-5	5	$\pi/2$

Y los valores de los parámetros de los controladores PID son:

Tabla 4.8 Parámetros del controlador (S.4).

PID	F_x	F_y	M
P	1.012	0.739	0.241
I	1.798e-2	2.657e-2	7.630e-3
D	3.739	6.122	1.232
N	11.637	8.841	41.959

Los resultados son los siguientes:

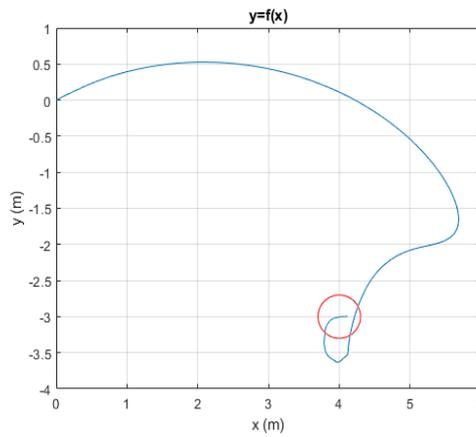


Figura 4.23 Trayectoria seguida en el plano xy (S.4).

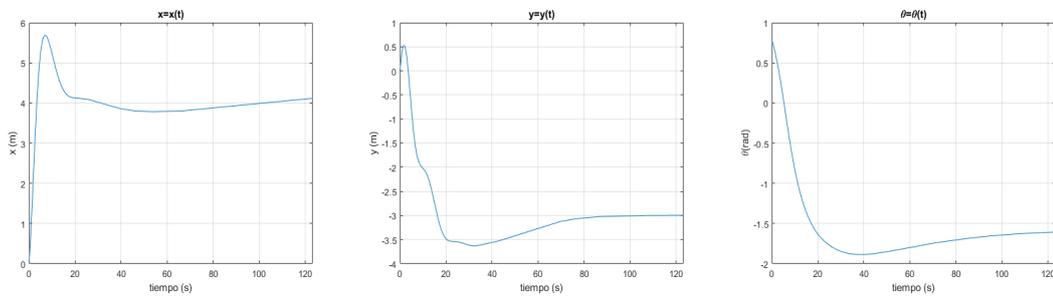


Figura 4.24 Evolución de la posición y la orientación (S.4).

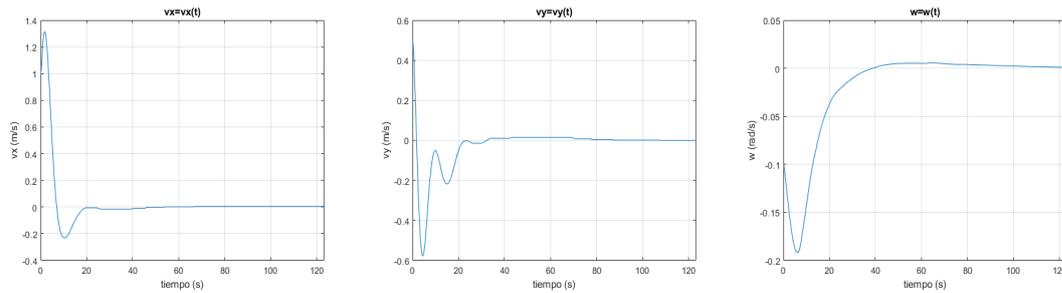


Figura 4.25 Evolución de las velocidades lineales y angular (S.4).

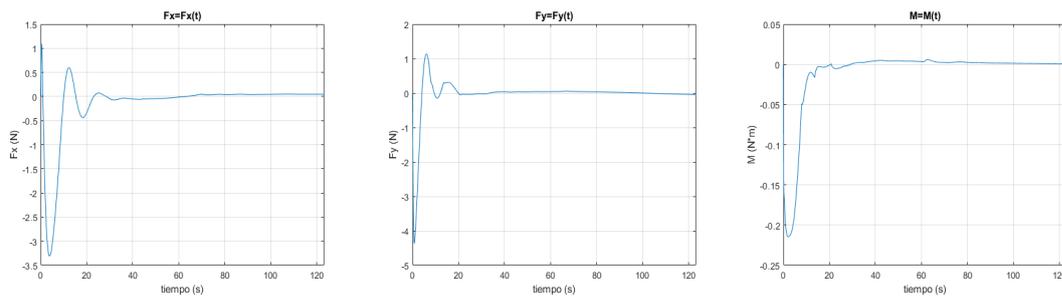


Figura 4.26 Evolución de las fuerzas y momento (S.4).

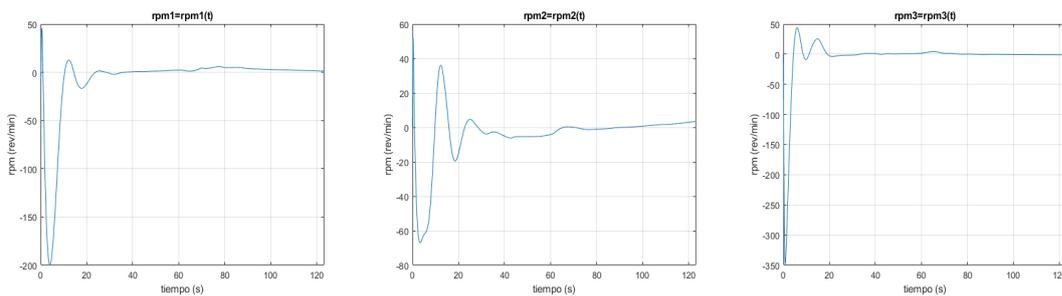


Figura 4.27 Evolución de las rpm's de los propulsores (S.4).

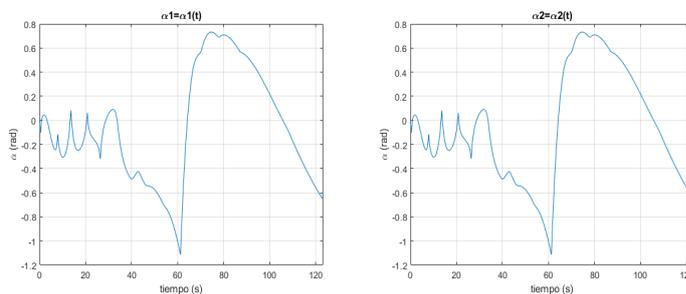


Figura 4.28 Evolución de las orientaciones de los propulsores azimutales (S.4).

Se aprecian unas sobreoscilaciones dentro de un rango admisible y un tiempo de duración de la misión similar a las demás configuraciones, sino ligeramente inferior. Aunque no parezca representar una gran mejora con respecto al resto de simulaciones, se debe recordar que gran parte del atractivo de este modelo es la redundancia de actuadores, que permitiría el completo funcionamiento del USV en caso de fallo o avería de cualquiera de sus motores.

5 Conclusiones y desarrollos futuros

Para finalizar este estudio, se va a realizar a modo de conclusión un breve análisis acerca de los resultados obtenidos en el capítulo 4, así como un énfasis en los focos de mayor problemática en el desarrollo de los modelos de USV's y posibles implementaciones para una futura ampliación de este proyecto, con el objetivo de optimizar su funcionamiento y considerar escenarios más realistas y cercanos a los usos que tienen en la actualidad.

5.1 Principales obstáculos

A lo largo del capítulo anterior se ha podido comprobar que, aunque no sin problemas, se ha podido diseñar un controlador no demasiado complejo que rijan el movimiento del vehículo de manera bastante efectiva. Gran parte de la complicación de este se ha hallado en asignar cada error correspondiente a una de las magnitudes que controlaban el USV, y tener en cuenta todas las consideraciones en relación a las posiciones relativas entre el barco y el destino.

La no linealidad del problema también ha supuesto un incremento de la dificultad al no hacer más complicado el escoger los parámetros adecuados de los PID's. Simulink posee una herramienta que permite tunear de manera semi-automática y sencilla los controladores, pero al trabajar con distintos ejes de referencia el sistema se vuelve rápidamente no lineal, por lo que el uso de esta herramienta se ha visto imposibilitado en numerosas ocasiones.

En un alto número de intentos, el resultado ha sido un movimiento errático del USV o directamente un control inestable, fruto de una combinación de parámetros que, sin ser tratados adecuadamente, provocaban el incorrecto funcionamiento del sistema. Otro problema muy común surgía a la hora de trabajar con los ángulos, para lo que ha hecho falta acotarlos dentro de un rango definido y hasta llegar a considerar, en algunos casos, todas las combinaciones posibles entre los cuatro cuadrantes en los que se podían encontrar.

Otro aspecto que también ha supuesto una inversión considerable de tiempo ha sido decidir los valores que tendrían las variables del problema. Dado que mucha de la dinámica de este se ha simplificado mediante relaciones lineales definidas por una constante, escoger correctamente esta constante era de vital importancia para asegurar que cada elemento tuviera la correspondiente repercusión en el problema que tendría realmente en un sistema físico.

5.2 Puntos a mejorar

En vista a futuro, se podrían considerar una serie de mejoras para la simulación que permitirían acercar más el sistema a su comportamiento en un entorno real.

La primera y más evidente sería modelar las fuerzas que intervienen en el problema de forma más precisa. Especialmente para los propulsores se podría llevar a cabo un estudio propulsivo de hélices, como el

desarrollado en la asignatura de *Propulsión* de 3º de GIA, para obtener una relación más acertada entre las rpm's y la fuerza generada. También se podría ahondar más en las fuerzas de resistencia hidrodinámicas, para cuyo modelado se puede tomar como base de estudio lo impartido en las asignaturas de *Mecánica de Fluidos II* y *Aerodinámica II* de 3º y 4º de GIA respectivamente.

Así mismo también se pueden incorporar fuerzas ambientales más realistas, simulando oleaje y corrientes, añadir ruido más realista de sensores e incluso tener en consideración un modelo tridimensional, donde se tengan en cuenta la traslación en el eje z y los momentos de cabeceo y alabeo, asumidos como despreciables en este estudio. También se puede añadir un sistema de detección de obstáculos y una modificación del controlador para que aprenda a esquivarlos.

Otro elemento a depurar es la saturación de los controladores tanto en revoluciones por minuto como en la máxima fuerza que pueden generar. Escoger valores óptimos en lo que respecta al tiempo que tardan los propulsores en acelerar o frenar, o los timones de dirección y propulsores azimutales en girar, mejorará desde luego las actuaciones proporcionando un resultado más cercano a su equivalente físico.

Por último, siempre está abierto el campo de mejora de los controladores, testeando y asignándoles nuevos parámetros que mejoren las actuaciones del vehículo dependiente de la misión que se le asigne. Introducir además conceptos como la "zona prohibida" comentada en el capítulo 2.3.4 o fuerzas externas adicionales como el efecto coriolis o la carga de pago puede ampliar el rango de funcionalidades en las que el vehículo puede operar.

Y por supuesto, siempre se puede ampliar el número de configuraciones a probar para expandir el estudio.

Anexo: Matlab

En este anexo se muestran los distintos archivos de Matlab que se han usado para la simulación de las distintas configuraciones de USV's:

Código 5.1 Ejecución de la simulación desde Matlab.

```
% Parametros
m=10;
Ib=5;
lx1=-0.15;
lx2=-0.1;
ly1=-0.15;
ly2=0.1;
lx3=0.2;
k1=0.0125;
k2=0.0125;
k3=0.0125;
cT=0.25;
cR=0.75;
param=[m Ib lx1 lx2 ly1 ly2 lx3 k1 k2 k3 cT cR]';
Fxmax=20;
Fymax=10;
Mmax=2;
F_perturbacion=[0 0 0]';
R_sensores=[0 0 0]';

% Misión
xf=4;
yf=3;
thetaf=-pi/2;

% Condiciones iniciales
x0=0;
y0=0;
theta0=pi/4;
vx0=1;
vy0=0.5;
w0=-0.1;
X0=[x0 y0 theta0 vx0 vy0 w0]';

% Activación del breaker
B=1;
```

```

% Corremos la simulación
sim('config_4_v1', 1000);

% Posiciones
x=ans.pos(:,1);
y=ans.pos(:,2);
theta=ans.pos(:,3);
% Velocidades
vx=ans.vel(:,1);
vy=ans.vel(:,2);
w=ans.vel(:,3);
% Fuerzas de los actuadores
FxB=ans.forceBoat(:,1);
FyB=ans.forceBoat(:,2);
MB=ans.forceBoat(:,3);
% Fuerzas en ejes globales
FxG=ans.force(:,1);
FyG=ans.force(:,2);
MG=ans.force(:,3);
% Fuerzas comandadas por el PID
FxC=ans.forcePID(:,1);
FyC=ans.forcePID(:,2);
MC=ans.forcePID(:,3);
% Actuaciones
rpm1=ans.rpm(:,1);
rpm2=ans.rpm(:,2);
rpm3=ans.rpm(:,3);
alpha1=ans.rpm(:,4);
alpha2=ans.rpm(:,5);
% Tiempo
t=ans.time;

% Ploteamos las gráficas
tiledlayout(3,3)
nexttile
plot(x,y)
title('y=f(x)')
xlabel('x (m)')
ylabel('y (m)')
viscircles([xf,yf],0.3,'LineWidth',0.5)
grid
nexttile
plot(t,x,t,y)
title('x=x(t) / y=y(t)')
xlabel('tiempo (s)')
ylabel('x (m) / y (m)')
xlim([0 t(end)])
legend('x', 'y')
grid
nexttile
plot(t,rpm1,t,rpm2,t,rpm3)
title('rpm=rpm(t)')
xlabel('tiempo (s)')
ylabel('rpm (rev/min)')
xlim([0 t(end)])
legend('rpm1', 'rpm2', 'rpm3')
grid

```

```

nexttile
plot(t,FxB,t,FyB)
title('Fx=Fx(t) / Fy=Fy(t)')
xlabel('tiempo (s)')
ylabel('Fx (N) / Fy (N)')
xlim([0 t(end)])
legend('Fx', 'Fy')
grid
nexttile
plot(t,MB)
title('M=M(t)')
xlabel('tiempo (s)')
ylabel('M (N*m)')
xlim([0 t(end)])
grid
nexttile
plot(t,alpha1,t,alpha2)
title('\alpha=\alpha(t)')
xlabel('tiempo (s)')
ylabel('\alpha (rad)')
xlim([0 t(end)])
legend('\alpha1', '\alpha2')
grid
nexttile
plot(t,vx,y,vy)
title('Vx=Vx(t) / Vy=Vy(t)')
xlabel('tiempo (s)')
ylabel('Vx (m/s) / Vy (m/s)')
xlim([0 t(end)])
grid
nexttile
plot(t,w)
title('w=w(t)')
xlabel('tiempo (s)')
ylabel('w(rad/s)')
xlim([0 t(end)])
grid
nexttile
plot(t,theta)
title('\theta=\theta(t)')
xlabel('tiempo (s)')
ylabel('\theta(rad)')
xlim([0 t(end)])
grid

```

Código 5.2 Función piBound.

```

function thetaB = piBound(theta)

thetaB=mod(theta+pi,2*pi)-pi;
if thetaB==-pi
    thetaB=pi;
end

```

Código 5.3 Función eFixer.

```
function eFix = eFixer(e, theta)

eFix=e;

if theta>pi/2 || theta<-pi/2
    eFix(1)=-e(1)
    eFix(2)=-e(2)
end

if e(3)>pi
    eFix(3)=e(3)-2*pi;
elseif e(3)<-pi
    eFix(3)=e(3)+2*pi;
end
```

Código 5.4 Función Breaker.

```
function stop = Breaker(B, Xvec, Xref)

stop=0;

if B==0
    return
end

dx=abs(Xvec(1)-Xref(1));
dy=abs(Xvec(2)-Xref(2));

De=sqrt(dx^2+dy^2);
Ve=sqrt(Xvec(4)^2+Xvec(5)^2);

Thetae=abs(Xref(3)-Xvec(3));
We=abs(Xvec(6));

if De<=0.25 && Ve<=0.01 && Thetae<=0.1 && We<=0.01
    stop=1;
end
```

Índice de Figuras

1.1	Monitorización del agua	1
1.2	Mapeado de fondos marinos	2
1.3	Monitorización de la vida marina	2
1.4	Descripción de un USV de pequeño tamaño	3
1.5	Posicionamiento dinámico de USV's	3
2.1	Aceleración en sentido contrario a la marcha	6
2.2	Aceleración en el sentido de la marcha y contrario	6
2.3	Freno en el sentido de la marcha	6
2.4	Freno en sentido contrario a la marcha	6
2.5	Relación Empuje - Revoluciones por segundo	6
2.6	Modelado del rudder	7
2.7	Embarcación con propulsores azimutales y transversal	11
2.8	SMARTBoat 3	12
2.9	IEEE Xplore	12
2.10	DOI 16.1002	13
2.11	WQMS	13
2.12	ADAM M1	14
3.1	Esquema del modelado en Simulink	15
3.2	Módulo de Navegación en Simulink	16
3.3	Configuración de actuadores N°1	19
3.4	Configuración de actuadores N°2	20
3.5	Configuración de actuadores N°3	20
3.6	Configuración de actuadores N°4	21
4.1	Modelo de controlador cinemático	23
4.2	Trayectoria para una misión desde [0,0] hasta [3,5]	24
4.3	Modelo de controlador dinámico N°1	24
4.4	Modelo de controlador dinámico N°2	25
4.5	Modelo de controlador dinámico N°3	26
4.6	Modelo de controlador dinámico N°4	27
4.7	Modelo de controlador dinámico N°5	28
4.8	Trayectoria seguida en el plano xy (S.1)	29
4.9	Evolución de la posición y la orientación (S.1)	29
4.10	Evolución de las velocidades lineales y angular (S.1)	29
4.11	Evolución de las fuerzas y momento (S.1)	30
4.12	Evolución de las rpm's de los propulsores (S.1)	30
4.13	Trayectoria seguida en el plano xy (S.2)	31
4.14	Evolución de la posición y la orientación (S.2)	31
4.15	Evolución de las velocidades lineales y angular (S.2)	31

4.16	Evolución de las fuerzas y momento (S.2)	31
4.17	Evolución de las rpm's de los propulsores (S.2)	32
4.18	Trayectoria seguida en el plano xy (S.3)	32
4.19	Evolución de la posición y la orientación (S.3)	33
4.20	Evolución de las velocidades lineales y angular (S.3)	33
4.21	Evolución de las fuerzas y momento (S.3)	33
4.22	Evolución de las rpm's de los propulsores y orientaciones de los azimutales (S.3)	33
4.23	Trayectoria seguida en el plano xy (S.4)	34
4.24	Evolución de la posición y la orientación (S.4)	34
4.25	Evolución de las velocidades lineales y angular (S.4)	35
4.26	Evolución de las fuerzas y momento (S.4)	35
4.27	Evolución de las rpm's de los propulsores (S.4)	35
4.28	Evolución de las orientaciones de los propulsores azimutales (S.4)	35

Índice de Tablas

2.1	Actuadores y fuerzas que generan	8
3.1	Parámetros de la 1º Configuración	19
3.2	Parámetros de la 2º Configuración	19
3.3	Parámetros de la 3º Configuración	20
3.4	Parámetros de la 4º Configuración	21
4.1	Parámetros de la Simulación Nº1	28
4.2	Parámetros del controlador (S.1)	29
4.3	Parámetros de la Simulación Nº2	30
4.4	Parámetros del controlador (S.2)	30
4.5	Parámetros de la Simulación Nº3	32
4.6	Parámetros del controlador (S.3)	32
4.7	Parámetros de la Simulación Nº4	34
4.8	Parámetros del controlador (S.4)	34

Índice de Códigos

2.1	Ejemplo del uso de quadprog	10
2.2	Pequeño extracto del código para Thruster Allocation del ADAM M1	11
3.1	Módulo de Navegación en Matlab	18
4.1	Algoritmo de seguimiento de una recta	25
4.2	Algoritmo de posicionamiento en polares	26
5.1	Ejecución de la simulación desde Matlab	39
5.2	Función piBound	41
5.3	Función eFixer	41
5.4	Función Breaker	42

Bibliografía

- [1] *Motion control systems*, ch. 12, pp. 343–415, John Wiley Sons, Ltd, 2011.
- [2] Bugra Alkan, *Thrust control design for unmanned marine vehicles*, Ph.D. thesis, 04 2012.
- [3] Huiru Cao, Zhongwei Guo, Shian Wang, Haixiu Cheng, and Choujun Zhan, *Intelligent wide-area water quality monitoring and analysis system exploiting unmanned surface vehicles and ensemble learning*, *Water* **12** (2020), no. 3.
- [4] Thor Fossen, Tor Johansen, and Perez Tristan, *A survey of control allocation methods for underwater vehicles*, 01 2009.
- [5] Wonse Jo, Yuta Hoashi, Lizbeth Leonor Paredes Aguilar, Mauricio Postigo-Malaga, José M. Garcia-Bravo, and Byung-Cheol Min, *A low-cost and small usv platform for water quality monitoring*, *HardwareX* **6** (2019), e00076.
- [6] Tor A. Johansen, Thomas P. Fuglseth, Petter Tøndel, and Thor I. Fossen, *Optimal constrained control allocation in marine surface vessels with rudders*, *Control Engineering Practice* **16** (2008), no. 4, 457–464, Special Section on Manoeuvring and Control of Marine Craft.
- [7] Pratap Tokekar, Deepak Bhadauria, Andrew Studenski, and Volkan Isler, *A robotic system for monitoring carp in minnesota lakes*, *J. Field Robotics* **27** (2010), 779–789.
- [8] Jose Villa Escusol, Jussi Aaltonen, and Kari Koskinen, *Path-following with lidar-based obstacle avoidance of an unmanned surface vehicle in harbor conditions*, *IEEE/ASME Transactions on Mechatronics PP* (2020), 1–1.
- [9] Atsushi Watanabe, Miwa Kuri, and Keiji Nagatani, *Field report: Autonomous lake bed depth mapping by a portable semi-submersible usv at mt. zao okama crater lake*, 2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), 2016, pp. 214–219.
- [10] Katarzyna Żelazny, *Approximate method of calculating forces on rudder during ship sailing on a shipping route*, *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation* **8** (2014), 459–464.