

Trabajo Fin de Máster

Máster Universitario en Ingeniería Aeronáutica

Modelado y control de una plataforma de estabilización inercial

Autor: José María Bascón Mallado

Tutores: Francisco Gordillo Álvarez

Antonio de la Cruz González Fernández

Dep. de Ingeniería de Sistemas y Automática

Dep. de Física Aplicada III

Escuela Técnica Superior de Ingeniería

Sevilla, 2019



Trabajo Fin de Máster
Máster Universitario en Ingeniería Aeronáutica

Modelado y control de una plataforma de estabilización inercial

Autor:

José María Bascón Mallado

Tutores:

Francisco Gordillo Álvarez

Antonio de la Cruz González Fernández

Dep. de Ingeniería de Sistemas y Automática

Dep. de Física Aplica III

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2019

Trabajo Fin de Máster: Modelado y control de una plataforma de estabilización inercial

Autor: José María Bascón Mallado

Tutores: Francisco Gordillo Álvarez
Antonio de la Cruz González
Fernández

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2019

El Secretario del Tribunal

A mi familia.

A mis amigos

*A todos los que han contribuido a
mi formación.*

Agradecimientos

Son muchas las personas a las que debo de dedicar este esfuerzo, en el ámbito académico, profesional y personal.

En primer lugar, me gustaría destacar el papel de tres personas fundamentales en este proyecto por su tiempo y dedicación. Quiero darle las gracias a Joaquín Bernal por ofrecerme la oportunidad de realizar un proyecto con aplicación en el ámbito profesional y participar de forma activa a lo largo de su desarrollo, contribuyendo a formarme en aspectos que debe cubrir un ingeniero más allá del ámbito técnico. Como no, debo darle las gracias a Francisco Gordillo por aceptar y dirigir gran parte de este trabajo sin que estuviera siquiera en sus planes. Gracias Paco. Por último, y fundamental en este proyecto, debo darle las gracias a Antonio González, por su ayuda y su tiempo. Gracias a todos.

Debo darle las gracias a Julia Batista y Rafael Berjillos por confiar en mí para el desarrollo de esta tarea. He trabajado día a día con Julia, por ello me gustaría destacar su brillantez como persona e ingeniera. Gracias Julia por prestarme ayuda en cada problema que ha ido surgiendo, involucrarme de forma completa en el proyecto, contribuir a mi formación, y como no, por los momentos de risas. Deseo y estoy seguro que llegarás muy lejos.

Durante esta etapa he conocido también a otro profesional excepcional, Carlos Domínguez (2se: Space Submicron Electronic-<http://www.2se.es/>). Gracias Carlos por tu tiempo, ayuda y aporte de nuevas ideas.

Deseo también darle las gracias a un amigo, Álvaro Sosa, por animarme a continuar mi formación tras el grado. Gracias amigo, llevabas razón en eso de “dos años no son nada”, hoy me alegro de haberte escuchado.

No se me puede olvidar María, para mí Mery. Gracias por apoyarme en cada decisión, ayudarme cada día y desear siempre lo mejor para mí. Gracias por todo.

Como no, debo destacar el papel durante todos estos años de mi familia. Gracias por animarme a seguir formándome, por confiar en mí desde el primer momento, y a pesar de los altibajos vividos durante estos años, siempre darme ese empujón que he necesitado para continuar hacia adelante y no rendirme.

A todos los que habéis contribuido de una forma u otra a mi formación, gracias.

Aquí, todos, tenéis un amigo.

José María Bascón Mallado

Mayo de 2019

Resumen

El trabajo presente surge como respuesta a una de las tareas propuestas en el proyecto SADSIM, enmarcado dentro del programa FEDER INNTERCONECTA en la convocatoria de 2016. SADSIM (Sistema Aéreo de Detección, Identificación y Seguimiento Marítimo) está constituido por un consorcio de empresas (TTI, Ingenostrum y Aeorum), que de forma conjunta han desarrollado un sistema destinado a grandes buques mercantes y de pesca. Dicho sistema, instalado en vehículos aéreos no tripulados, proporciona información de utilidad para los buques, ahorrándole costes de operación, aumentado su seguridad y facilitando las operaciones.

El proyecto está dividido en cuatro actividades, los cuales están compuestas por distintas tareas a desarrollar. Dentro de la segunda de ellas se encuentra la tarea “*A2.3 Diseño de sistema de algoritmos y control predictivo*”, la cual tiene como objetivo el desarrollo de una plataforma estabilizada sobre la cubierta de un buque para el aterrizaje de UAV.

Para el desarrollo de dicha tarea, en primer lugar, se ha llevado a cabo una investigación de los sistemas desarrollados con funciones similares y su aplicación. Conocida la solución tecnológicamente viable, se ha obtenido el sistema mecánico que modela el comportamiento de la plataforma, estudiando la cinemática que lo modela y las estrategias de control aplicables, optando por una solución de compromiso entre dificultad y prestaciones. Además, con vistas a una implementación real, se ha implementado un modo de ahorro energético. Todo ello se ha realizado usando la herramienta Matlab & Simulink.

Para modelar la respuesta del barco ante la excitación marina, se ha hecho uso de la tarea previamente desarrollada, introduciendo de forma resumida el modelado en este documento.

Por último, se han llevado a cabo distintas simulaciones para comprobar el funcionamiento del sistema, y mediante la definición de parámetros de interés, se han estudiado los límites operativos del sistema.

Abstract

This project develops an SADSIM project's activity which is part of FEDER INNTERCONNECTA 2016 program. SADSIM (Maritime Aerial System of Detection, Tracking and Recognition) it is comprised from a consortium of companies (TTI, Ingenostrum and Aeorum) which have development a system for merchant ships and fishing vessels. This system, installed on some UAV, can send useful information to these ships, saving operation costs and increasing the security.

The project made up four activities. In the second activity, the task "*A2.3: Predictive control and algorithms system design*" can be found, whose objective is to develop a stabilized platform above a ship deck for UAV landing.

Firstly, a study of similar systems and its use has been carried out, being Stewart Platform the answer. This platform, which consists of a mechanical block, a control block and a kinematics block, has been modelled in Matlab & Simulink. PID control has been chosen due to different reasons. Moreover, for a real implementation, an energy savings mode has been designed.

Ship movement has been obtained from a previous task, and in this document only has been introduced of a summarized form.

Finally, different simulations have been run to test system performance and to get the operation range.

Índice

Agradecimientos	i
Resumen	iii
Abstract	v
1 Introducción	1
2 La Plataforma Stewart	7
2.1 <i>Cinemática inversa</i>	8
2.2 <i>Cinemática directa</i>	9
2.3 <i>Cinemática en Matlab & Simulink</i>	12
2.4 <i>Configuración geométrica y dinámica de la plataforma</i>	13
3 El Sistema de Control	17
3.1 <i>El controlador PID</i>	17
3.2 <i>Linealización del modelo</i>	18
3.3 <i>Diseño frecuencial del controlador</i>	22
4 Definición de las Condiciones Operativas	29
4.1 <i>Respuesta de un barco a la excitación marina</i>	29
4.1.1 <i>Introducción al modelado del movimiento marino</i>	29
4.1.2 <i>Modelado del movimiento del barco</i>	33
4.2 <i>Límites operativos de la plataforma</i>	38
4.3 <i>Actuaciones del UAV</i>	39
5 Resultados	41
5.1 <i>Simulación 1</i>	45
5.2 <i>Simulación 2</i>	46
5.3 <i>Simulación 3</i>	48
5.4 <i>Simulación 4</i>	49
5.5 <i>Análisis de resultados</i>	50
5.6 <i>Simulaciones tridimensionales</i>	53
6 Conclusiones y Trabajos Futuros	55
Anexos	57
<i>Anexo A</i>	57
<i>Escala Douglas</i>	57
<i>Escala Beaufort</i>	59
<i>Anexo B</i>	61
<i>Script Cinemática directa</i>	61
<i>Script de configuración</i>	63
Bibliografía	67

Índice de Figuras

Figura 1. Accidentes de barcos 1990-2012 [1]	2
Figura 2. Porcentaje de accidentes para aeronaves tripulada según la fase del vuelo [2]	3
Figura 3. Despegue UAV con sistema de catapulta [3]	3
Figura 4. Plataforma de aterrizaje SADSIM [4]	4
Figura 5. UAV sin tren de aterrizaje	4
Figura 6. Net Landing [5]	5
Figura 7. Plataforma Stewart para simulador [6]	5
Figura 8. Configuración geométrica Plataforma Stewart [7]	7
Figura 9. Sistema de acelerómetros y giróscopos en la base de la plataforma [4]	8
Figura 10. Ejes locales [8]	9
Figura 11. Nomenclatura cinemática directa	10
Figura 12. Implementación cinemática inversa en Simulink	12
Figura 13. Bloque <i>Body Sensor</i> en Simulink	12
Figura 14. Configuración del bloque <i>Body Sensor</i>	13
Figura 15. Geometría del actuador lineal [10]	13
Figura 16. Modelo Simulink de la Plataforma Stewart	15
Figura 17. Modelo Simulink del actuador	15
Figura 18. Representación 3D Plataforma Stewart y pista de aterrizaje	16
Figura 19. Bode PID ideal (azul) frente a PID real (rojo)	18
Figura 20. Configuración de la plataforma en el punto de equilibrio	19
Figura 21. Señal de control U_0 (Fuerza) en el punto de linealización	19
Figura 22. Estabilización de la plataforma en torno al punto de linealización	20
Figura 23. Introducción de U_0 en el modelo Simulink	20
Figura 24. Polos y ceros del sistema linealizado	21
Figura 25. Respuesta de un sistema según sus polos [12]	21
Figura 26. Simulación de la respuesta del sistema ante escalón durante 10 horas	22
Figura 27. Simulación de la respuesta del sistema ante escalón en hora 2 durante 7 horas	22
Figura 28. Diagrama Bucle cerrado vs Bucle abierto [13]	23
Figura 29. Diagrama de Bode de sistema estable	23
Figura 30. Bode del modelo linealizado sin controlador	24
Figura 31. Bode del sistema (en bucle abierto) con controlador PID	25
Figura 32. Bode del sistema en bucle cerrado con PID	26

Figura 33. Respuesta del sistema linealizado ante escalón.	26
Figura 34. Respuesta ante escalón del sistema no lineal	27
Figura 35. Diagrama de desarrollo del proyecto	30
Figura 36. Espectro de ola según velocidad del viento [15]	31
Figura 37. Modelado del comportamiento del mar a partir de la distribución espectral [14]	31
Figura 38. Rango de aplicación de la función de Pierson-Moskowitz [14]	31
Figura 39. Espectro JONSWAP vs Pierson-Moskowitz [15]	32
Figura 40. Simulación movimiento marino según función JONSWAP	32
Figura 41. Grados de libertad para barco	33
Figura 42. Esquema del comportamiento de la embarcación a la excitación marina [14]	34
Figura 43. Función de transferencia que modela la respuesta del barco [14]	35
Figura 44. Dirección del frente de olas [14]	35
Figura 45. Características geométricas barco [18]	36
Figura 46. RAOs para barco tipo F	37
Figura 47. Bloques Simulink para introducir el movimiento marino	38
Figura 48. Altura H de la ola [14]	38
Figura 49. Actuador electrohidráulico linear y sus especificaciones (EXLAR®) [19]	39
Figura 50. RPAS Tucan [3]	40
Figura 51. Modelo Simulink del sistema completo	42
Figura 52. Diagrama para la simulación del movimiento del sistema	44
Figura 53. Movimiento marino. Simulación 1	45
Figura 54. Errores y fuerzas. Simulación 1	45
Figura 55. x, y, zcdg plataforma. Simulación 1	46
Figura 56. Movimiento marino. Simulación 2	46
Figura 57. Errores y fuerzas. Simulación 2	47
Figura 58. x, y, zcdg plataforma. Simulación 2	47
Figura 59. Movimiento marino. Simulación 3	48
Figura 60. Errores y fuerzas. Simulación 3	48
Figura 61. x, y, zcdg plataforma. Simulación 3	49
Figura 62. Movimiento marino. Simulación 4	49
Figura 63. Errores y fuerzas. Simulación 4	50
Figura 64. x, y, zcdg plataforma. Simulación 4	50
Figura 65. Periodos típicos de olas según velocidad del viento y altura de ola [6]	51
Figura 66. Excitación sinusoidal en grados de libertad z y cabeceo	51
Figura 67. Errores y fuerzas ante excitación sinusoidal	52
Figura 68. Excitación aleatoria sinusoidal	52
Figura 69. Errores y fuerzas ante excitación sinusoidal aleatoria	53

Índice de Tablas

Tabla 1. Parámetros geométricos y dinámicos de la plataforma	14
Tabla 2. Polos y ceros del sistema linealizado	21
Tabla 3. Polos y Ceros PID	24
Tabla 4. Grados de libertad barco	33
Tabla 5. Características barcos estudiados en [17]	35
Tabla 6. Escala Douglas	57
Tabla 7. Clasificación de la longitud de ola	58
Tabla 8. Clasificación de la altura de ola	58
Tabla 9. Escala Douglas	59

1 INTRODUCCIÓN

El fracaso es una gran oportunidad para empezar de nuevo con más inteligencia.

Henry Ford

Poco tiempo después del primer vuelo de los hermanos Wright (1903-Kitty Hawk, Carolina del Norte), se realizó el primer despegue experimental desde la cubierta de un barco de la US Navy (USS Birmingham, 14 de noviembre de 1910). Dos meses más tarde, se logró el aterrizaje en un buque, esta vez en el USS Pennsylvania. En ambas ocasiones los barcos estaban atracados en puerto. Fue el 9 de mayo de 1912 la primera vez que se consiguió el despegue desde un buque en marcha, en esta ocasión desde la cubierta del HMS Hibernia de la Royal Navy.

Desde el nacimiento de la aviación es de interés operar una aeronave desde la cubierta de un barco. Aunque hasta comienzos del nuevo milenio dicho interés radicaba en el sector militar, la inclusión de las aeronaves no tripuladas (UAV) en el mercado civil y el bajo coste de las mismas, hace que sea posible desarrollar aplicaciones marítimas en las que es necesario despegar y aterrizar desde cubiertas.

En el tráfico marítimo la seguridad es un elemento clave, tanto en lo que respecta a la protección de la tripulación/pasajeros, como en lo referente a la eficiencia económica derivada de una mejor protección de la carga transportada y del menor consumo de combustible que se logra con una mayor eficiencia en las rutas trazadas.

En un contexto como el actual, en el que el tráfico marítimo de mercancías está incrementándose continuamente, de forma pareja al tamaño de los grandes buques transportadores de contenedores, es cada vez mayor la apuesta de los grandes armadores por la búsqueda de soluciones tecnológicas que garanticen la seguridad en los grandes buques y la eficiencia económica de esta actividad.

A pesar de los sofisticados equipos de ayuda a la navegación y de detección de otros buques y elementos extraños dentro de la trayectoria, al cabo del año se producen una importante cantidad de accidentes por colisión entre grandes buques y embarcaciones más pequeñas o por dificultad de maniobrabilidad en puerto, que ocasionan pérdida de vidas humanas, buques y mercancías y daños ecológicos. Muchos de estos accidentes podrían ser evitados si los buques hubiesen tenido unas mejores condiciones de maniobrabilidad apoyadas por sistemas de seguridad con una relación coste-eficiencia adecuada. Ver Figura 1.

La demanda de nuevos productos que apoyen las actividades de control y seguridad que desempeñan las entidades portuarias, las fuerzas de seguridad del Estado o las empresas privadas de seguridad marítima, ha incentivado el desarrollo de un **sistema aéreo de detección, seguimiento e identificación marítima** basado en el uso de RPAS, por sus siglas SADSIM.

El mercado objetivo será el sector del transporte marítimo, y más específicamente el vinculado a los grandes cargueros transportadores de contenedores y los grandes buques pesqueros, sectores que tienen actualmente un importante potencial de crecimiento y que presentan además una fuerte tendencia a la aplicación de innovaciones tecnológicas.

El proyecto ha sido desarrollado por un consorcio de tres empresas: TTI, Aeorum e Ingenostrum. Además, ha contado con la colaboración del grupo DIANA de la Universidad de Málaga, DIESIA de la Universidad de Huelva, CATEC y Tecnalía.

Table 1. Casualty Records, Time Period 1990-2012.

No of serious accidents per ship type		
Generic Ship Type	Number	
General Cargo	4,114	37.9%
Bulk Carriers	1,951	18.0%
Passenger Ro-Ro Cargo	955	8.8%
Fishing	610	5.6%
Passenger ²	389	3.6%
Reefer	303	2.8%
Cruise	241	2.2%
Ro-Ro Cargo	230	2.1%
Car Carriers	227	2.1%
LPG	183	1.7%
LNG	28	0.3%
Cellular Containerships	1,235	11.4%
Large Crude oil ³	375	3.5%
Total	10,841	100%

Figura 1. Accidentes de barcos 1990-2012 [1]

Tiene como objetivo “el diseño y desarrollo de un prototipo de Sistema Aéreo de Detección, Seguimiento e Identificación Marítima (SADSIM) que permita validar la integración de los sistemas de visión artificial, de comunicación y control, en los RPAS, así como dar solución a los problemas actuales que presenta el aterrizaje sobre superficies móviles”.

Las innovaciones que presenta SADSIM son:

- a) Un sistema de visión artificial de altas prestaciones, que identifique de forma precisa objetos en entornos difíciles y prediga la trayectoria del mismo.
- b) Una plataforma autonivelada de aterrizaje sobre la superficie del barco, que proporcione una referencia estable al UAV en su aproximación a la cubierta del barco en las maniobras de aterrizaje.
- c) Una red de comunicaciones que permita el envío de datos y vídeo tanto entre UAV-UAV como entre UAV-Estación Base, ofreciendo soluciones en el caso de que no haya línea de vista entre los nodos a comunicar o en los que el vehículo esté fuera del rango de cobertura de la estación de control.
- d) Un sistema de control y de algoritmos adaptativos que permitan controlar los drones haciendo uso del sistema de visión artificial, reprogramando actuaciones y leyes de control de vuelo en función de la detección de objetos y las necesidades para las operaciones de aproximación y aterrizaje.

Las tareas y actividades que componen el proyecto son:

- A.1 Sistema de Visión Artificial
- A.2 Plataforma estabilizada sobre superficie marina
- A.3 Sistema de comunicación
- A.4 Desarrollo de la unidad lógica de control

Dentro de la actividad “A.2 Plataforma estabilizada sobre superficie marina”, está enmarcada la tarea “A.2.3 Diseño de sistema de algoritmos y control predictivo”, objeto de este documento. La tarea está directamente relacionada con la “A.2.2 Estudio y desarrollo predictivo de inestabilidad de plataforma marina”, en la cual se desarrolla el modelo del movimiento del barco según el estado del mar. Dicho modelo se usa como entrada al sistema desarrollado.

La necesidad de disponer de una plataforma que se estabilice, es decir, mantenga su posición horizontal en la cubierta de un barco está motivada por la criticidad de las maniobras de aterrizaje y despegue en cualquier aeronave, en especial, desde superficies en movimiento. Aproximadamente, el 50% de los accidentes en aeronaves tripuladas¹ ocurren durante las fases de aproximación o aterrizaje según estadísticas de Airbus [2]. Ver Figura 2.

Aunque los datos no son directamente extrapolables a UAV, la criticidad se mantendrá en dichas fases al ser las más complejas. La complejidad está justificada porque durante estas fases se incrementa de forma sustancial la

¹ No se dispone de base de datos sobre accidentes de aeronaves no tripuladas. Esto justifica el uso de este dato para argumentar la criticidad de dichas fases.

carga de trabajo para el piloto, pues es necesario cambiar la configuración de la aeronave, hay una comunicación continua con los servicios ATC, y en muchos casos hay que añadir el estrés que supone operar con malas condiciones meteorológicas y de visibilidad.

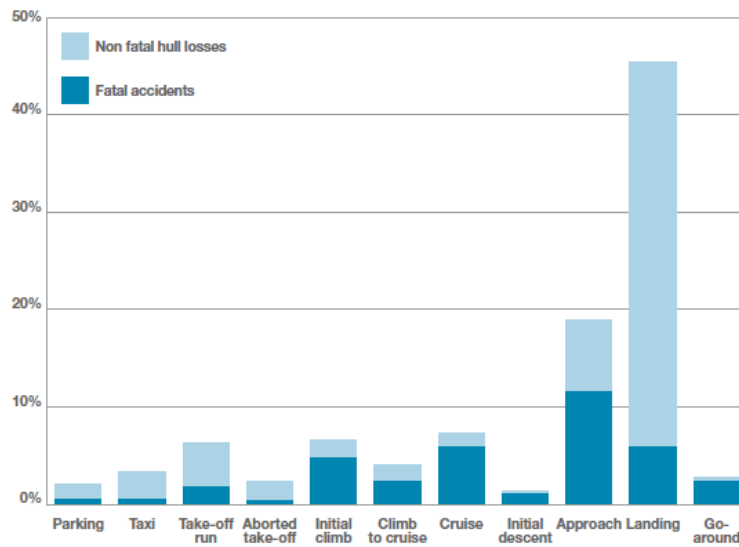


Figura 2. Porcentaje de accidentes para aeronaves tripulada según la fase del vuelo [2]

Para el caso de operaciones en buques, además hay que tener en cuenta la corta distancia que tiene la pista y el movimiento continuo de la misma debido al movimiento del mar.

Aunque el despegue desde la cubierta sí que es crítico para el caso de aeronaves tripuladas debido a la corta longitud de la pista, para el caso de aeronaves no tripuladas, hasta el momento, no es un problema, ya que generalmente necesitan poco recorrido de despegue y un gran porcentaje de ellas con catapultadas para iniciar el vuelo (Figura 3). En cambio, la operación de aterrizaje sí que mantendrá la criticidad o incluso la aumentará, ya que o bien estará controlada por un piloto remoto el cual no dispone de la conciencia situacional de un piloto en una aeronave tripulada, o bien la aeronave embarcará un sistema que realice el aterrizaje de forma autónoma.



Figura 3. Despegue UAV con sistema de catapulta [3]

Para el caso de aterrizajes de forma manual, el piloto necesitará disponer de algún sistema que le proporcione indicaciones de derrota, existiendo actualmente dos alternativas, el sistema OLS (Optical Landing System) y el sistema JPALS (Joint Precision Approach and Landing System). En SADSIM, la operación de aterrizaje se llevará a cabo de forma autónoma, y aunque el sistema JPALS podría ser una solución viable, se decidió disponer una marca QR en la pista-plataforma de aterrizaje-para que este la identifique e inicie las maniobras.

Una forma de disminuir la criticidad de esta operación es, por ejemplo, el diseño de una superficie de aterrizaje móvil, es decir, una superficie de aterrizaje con $n \in [1,6]$ grados de libertad capaz de compensar el movimiento del barco. La viabilidad tecnológica de esta solución depende del tamaño de la misma, ya que necesita actuadores que la muevan de forma rápida y controlada. Para aeronaves de pequeño tamaño, como son los UAV objeto de

este proyecto, es tecnológicamente viable el diseño, pues necesitarán poca longitud de aterrizaje, ya que dispondrá de una red al final de la misma que realizará la desaceleración final del vehículo. En la Figura 4 se presenta de forma gráfica el concepto que se va a desarrollar.

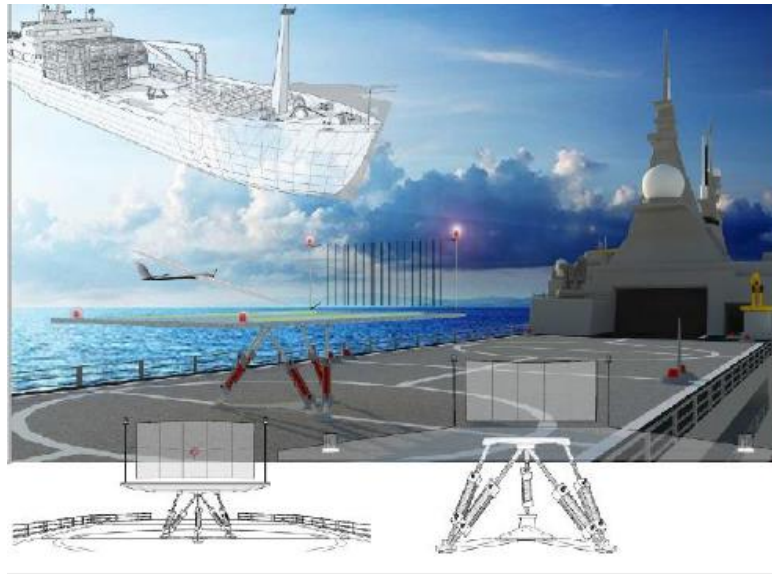


Figura 4. Plataforma de aterrizaje SADSIM [4]

Es importante de cara al proyecto conocer las opciones disponibles para el aterrizaje de UAV, en concreto, los de pequeño y mediano tamaño. Usan dos formas de aterrizaje principalmente:

- **Sobre la parte inferior del fuselaje.** La parte inferior del fuselaje dispone de un material protector que evita que se dañe el material del fuselaje durante el rozamiento con la superficie de aterrizaje.



Figura 5. UAV sin tren de aterrizaje

- **Net Landing.** Se dispone una red que absorbe la energía de la aeronave. El diseño y material de la malla debe ser tal que evite dañar la aeronave.



Figura 6. Net Landing [5]

Respecto a la viabilidad técnica de la propuesta, existe una solución tecnológica que lleva años funcionando en distintas aplicaciones, siendo la más extendida en simuladores de vuelo para entrenamiento de pilotos, aunque actualmente se emplea ampliamente en la industria de los simuladores de automoción y los videojuegos. Esta solución se conoce como **Plataforma Stewart**.



Figura 7. Plataforma Stewart para simulador [6]

Esta será la solución que se desarrolle. Para ello, la plataforma, el sistema de control y el sistema que mide las aceleraciones del barco se modela en Matlab&Simulink, y las simulaciones también se llevarán a cabo en este software. La estrategia de control desarrollada se basa en controladores PID por razones que se justifican en este documento.

Por último, es necesario tener en cuenta el estado del mar para el cual operará de forma satisfactoria el sistema, es decir, el tamaño de ola que corresponde a cada estado. Los estados del mar están clasificados según las escalas Douglas y Beaufort, expuestas en el Anexo B. Como se ha referido al principio de esta introducción, el movimiento del barco está proporcionado por la tarea A2.3.

2 LA PLATAFORMA STEWART

Los científicos estudian el mundo tal y como es; los ingenieros crean el mundo que nunca ha sido.

Theodore Von Karman

Enmarcada dentro de la tipología de robots paralelos, dispone de seis actuadores prismáticos, normalmente hidráulicos, eléctricos o electrohidráulicos, fijados a pares a tres posiciones de la base y dispuestos a pares a otros tres puntos de la plataforma móvil. Esta disposición permite que la plataforma móvil tenga seis grados de libertad (dentro de las limitaciones físicas de diseño y configuración de la plataforma), tres traslaciones (x , y , z) y tres rotaciones (pitch, yaw, roll). En general, la plataforma móvil se modela de forma triangular y se rota 60 grados respecto a la base, lo que permite que todos los actuadores sean equidistantes entre sí y puedan moverse de forma independiente de las demás.

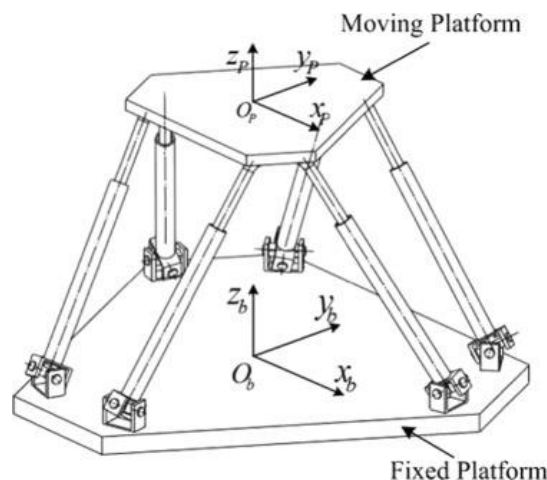


Figura 8. Configuración geométrica Plataforma Stewart [7]

Aunque ingenieros e investigadores han estudiado diferentes variantes de la Plataforma Stewart, esta es la que tiene mayor rango de movimientos y capacidad de posicionamiento preciso. Además, debido al diseño, permite soportar cargas elevadas y los actuadores solo soportan fuerzas de compresión, evitando la aparición de fuerzas de flexión que sí aparecen en otros diseños.

La cinemática del robot estudia el movimiento del mismo con respecto a un sistema de referencia. Así, la cinemática se interesa por la descripción analítica del movimiento espacial del robot como una función del tiempo y , en particular, por las relaciones entre la posición y la orientación del extremo final del robot con los valores que toman las coordenadas articulares.

Existen dos problemas fundamentales a resolver en la cinemática del robot, el primero de ellos se conoce como el problema cinemático directo, y el segundo se conoce como problema cinemático inverso.

El problema cinemático directo consiste en determinar la orientación y posición del extremo final del robot, con respecto a un sistema de coordenadas que se toma como referencia, conocidos los valores de las articulaciones y los parámetros geométricos del robot.

El cinemático inverso resuelve la configuración de debe adoptar el robot para una posición y orientación del extremo conocidas.

La cinemática inversa resulta más adecuada para el caso de la Plataforma Stewart, ya que:

- Es fácil calcular los ángulos a partir de la posición y orientación de la plataforma móvil (que en este caso es el actuador final o efector).
- El funcionamiento de la plataforma se basa en seguir una trayectoria especificada. Esta trayectoria puede venir, por ejemplo, de las medidas de una IMU (Inertial Measurement Unit) dispuesta en la base de la plataforma. El dispositivo medirá las aceleraciones y rotaciones que sufre la base de la plataforma (coincide con la cubierta del barco, la cual se modela como un sólido rígido para la aplicación de interés). La integración de estas aceleraciones y rotaciones proporcionarán las traslaciones y rotaciones que se producen en la cubierta. Multiplicando dichos valores por una ganancia de valor “-1” se genera el movimiento que tiene que llevar a cabo la plataforma móvil para compensar el movimiento de la cubierta, y que está se mantenga en posición horizontal en todas las condiciones de operación definidas.

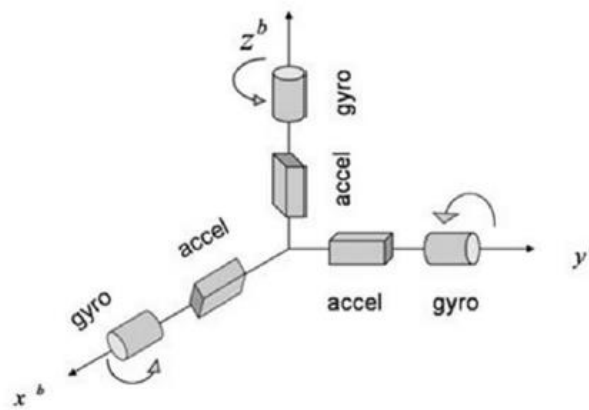


Figura 9. Sistema de acelerómetros y giróscopos en la base de la plataforma [4]

2.1 Cinemática inversa

Para calcular la longitud que debe tener cada actuador en cada instante de tiempo/simulación es necesario conocer el giro y los desplazamientos que tiene un punto de la cubierta (punto de instalación de la IMU). La longitud de cada actuador se puede calcular a partir de la ecuación (1).

$$l_i = |\underline{R} \cdot p_i - b_i| - l_n \quad (1)$$

Donde \underline{R} denota la matriz de rotación, p_i las coordenadas de los puntos a los que se fijan los actuadores de la plataforma móvil referidos a unos ejes locales, b_i la posición de los puntos de la base (en los que están anclados dichos actuadores) respecto a ejes locales, y l_n la longitud nominal de cada actuador.

$$R = \begin{bmatrix} R_{xx} & R_{xy} & R_{xz} \\ R_{yx} & R_{yy} & R_{yz} \\ R_{zx} & R_{zy} & R_{zz} \end{bmatrix} \quad (2)$$

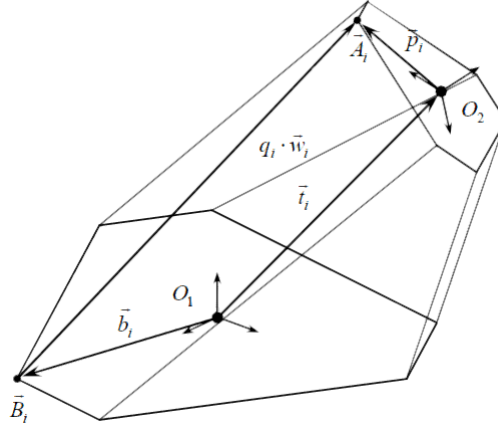


Figura 10. Ejes locales [8]

2.2 Cinemática directa

Es de interés conocer la posición que tendrá en cada momento el plano que forman los puntos a los que están fijados los actuadores en la plataforma móvil para ser capaz de simular el movimiento de la misma. Para lograr dicho objetivo se asumen las siguientes hipótesis:

- 1) Posición conocida del centro de gravedad (x_g, y_g, z_g) en cada instante de tiempo.
- 2) Se conoce la longitud de todos los actuadores en cada instante de tiempo.
- 3) Se conocen las coordenadas de los puntos de la base y la plataforma móvil en ejes locales.

Atendiendo a [9], se tienen aquí dos sólidos o sistemas de referencia. El sólido fijo “1” es la plataforma inferior, y el sólido móvil “2” la superior.

A cada sólido se le asigna un sistema de referencia. La base $\{\vec{i}_1, \vec{j}_1, \vec{k}_1\}$ es la base fija asociada al sólido 1, y la base $\{\vec{i}_2, \vec{j}_2, \vec{k}_2\}$ es la “base ligada” unida al sólido 2.

Cualquier punto y cualquier vector puede expresarse en cualquiera de las dos bases:

$$\overrightarrow{OP} = x_p \vec{i}_1 + y_p \vec{j}_1 + z_p \vec{k}_1 = X_p \vec{i}_2 + Y_p \vec{j}_2 + Z_p \vec{k}_2 \quad (3)$$

Las relaciones entre las componentes en ambas bases se obtienen proyectando:

$$x_p = \overrightarrow{OP} \cdot \vec{i}_1 = X_p \vec{i}_2 \cdot \vec{i}_1 + Y_p \vec{j}_2 \cdot \vec{i}_1 + Z_p \vec{k}_2 \cdot \vec{i}_1 \quad (4)$$

De igual forma se obtienen para las otras dos. Esto se puede escribir en forma matricial:

$$\begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} = \begin{pmatrix} R_{xx} & R_{xy} & R_{xz} \\ R_{yx} & R_{yy} & R_{yz} \\ R_{zx} & R_{zy} & R_{zz} \end{pmatrix} \begin{pmatrix} X_p \\ Y_p \\ Z_p \end{pmatrix} \quad (5)$$

donde

$$\bar{R} = \begin{pmatrix} R_{xx} & R_{xy} & R_{xz} \\ R_{yx} & R_{yy} & R_{yz} \\ R_{zx} & R_{zy} & R_{zz} \end{pmatrix} \quad (6)$$

es la matriz de rotación, que lleva de la posición en el sistema ligado a la posición en el sistema fijo. Dicho de otra manera, es la rotación que hay que efectuar para llevar al sólido 2 desde una posición alineada con el 1 a su orientación final. Los elementos de esta matriz son, por tanto, los cosenos directores:

$$R_{xx} = \vec{i}_2 \cdot \vec{i}_1 \quad R_{xy} = \vec{j}_2 \cdot \vec{i}_1 \quad R_{xz} = \vec{k}_2 \cdot \vec{i}_1 \quad (7)$$

$$R_{yx} = \vec{i}_2 \cdot \vec{j}_1 \quad R_{yy} = \vec{j}_2 \cdot \vec{j}_1 \quad R_{yz} = \vec{k}_2 \cdot \vec{j}_1 \quad (8)$$

$$R_{zx} = \vec{i}_2 \cdot \vec{k}_1 \quad R_{zy} = \vec{j}_2 \cdot \vec{k}_1 \quad R_{zz} = \vec{k}_2 \cdot \vec{k}_1 \quad (9)$$

o, dicho de otra forma, las filas son los vectores de la base fija (en la base ligada) y las columnas los vectores de la base ligada (en la base fija).

$$\bar{R} = \begin{pmatrix} \leftarrow \vec{i}_1 & \rightarrow \\ \leftarrow \vec{j}_1 & \rightarrow \\ \leftarrow \vec{k}_1 & \rightarrow \end{pmatrix} = \begin{pmatrix} \uparrow & \uparrow & \uparrow \\ \vec{i}_2 & \vec{j}_2 & \vec{k}_2 \\ \downarrow & \downarrow & \downarrow \end{pmatrix} \quad (10)$$

En el problema que se pretende resolver, se tiene en la plataforma inferior seis puntos de anclaje, de posiciones conocidas en la base fija:

$$\overrightarrow{OA} = x_A \vec{i}_1 + y_A \vec{j}_1 \quad (11)$$

(este sería uno de ellos). Tiene dos componentes porque son supuestas en un disco plano centrado en O.

En la plataforma móvil se tienen otros 6 puntos de anclaje con posiciones dadas en el sistema ligado:

$$\overrightarrow{GB} = X_B \vec{i}_2 + Y_B \vec{j}_2 \quad (12)$$

también con dos componentes porque está en disco plano centrado en G, que a su vez tiene la posición:

$$\overrightarrow{OG} = x \vec{i}_1 + y \vec{j}_1 + z \vec{k}_1 \quad (13)$$

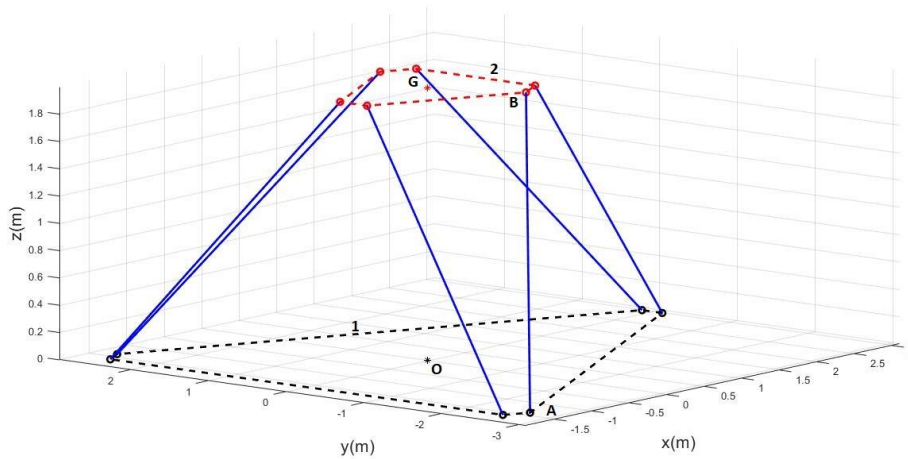


Figura 11. Nomenclatura cinemática directa

Hay que destacar que X_B y Y_B son conocidos, esto es, se trata de la posición de un punto de la plataforma móvil respecto a la propia plataforma. Nuestro objetivo es determinar $\{\vec{i}_2, \vec{j}_2, \vec{k}_2\}$ respecto a la base 1 o equivalentemente la matriz de rotación.

La distancia entre A y B es la longitud de una varilla, que es conocida. Al cuadrado:

$$\ell_{AB}^2 = \overrightarrow{AB} \cdot \overrightarrow{AB} \quad (14)$$

siendo la posición relativa,

$$\overline{AB} = \overline{AO} + \overline{OG} + \overline{GB} = (x - x_A)\vec{l}_1 + (y - y_A)\vec{j}_1 + z\vec{k}_1 + X_B\vec{l}_2 + Y_B\vec{j}_2 \quad (15)$$

Al multiplicar este vector por sí mismo resulta,

$$\begin{aligned} \ell_{AB}^2 &= (x - x_A)^2 + (y - y_A)^2 + z^2 + X_B^2 + Y_B^2 + \\ &+ 2X_B(x - x_A)\vec{l}_2 \cdot \vec{l}_1 + 2X_B(y - y_A)\vec{l}_2 \cdot \vec{j}_1 + 2X_B z \vec{l}_2 \cdot \vec{k}_1 + \\ &+ 2Y_B(x - x_A)\vec{j}_2 \cdot \vec{l}_1 + 2Y_B(y - y_A)\vec{j}_2 \cdot \vec{j}_1 + 2Y_B z \vec{j}_2 \cdot \vec{k}_1 \end{aligned} \quad (16)$$

o, en términos de la matriz de rotación,

$$\begin{aligned} &(2X_B(x - x_A))R_{xx} + (2X_B(y - y_A))R_{yx} + (2X_B z)R_{zx} + \\ &+ (2Y_B(x - x_A))R_{xy} + (2Y_B(y - y_A))R_{yy} + (2Y_B z)R_{zy} = \\ &= \ell_{AB}^2 - ((x - x_A)^2 + (y - y_A)^2 + z^2 + X_B^2 + Y_B^2) \end{aligned} \quad (17)$$

Esto es una ecuación lineal con 6 incógnitas:

$$R_{xx}, R_{xy}, R_{yx}, R_{yy}, R_{zx}, R_{zy}$$

donde todos los coeficientes son conocidos y tiene como parámetros la posición (x, y, z) de G y la longitud de la varilla (las otras coordenadas son constantes definidas por las dimensiones de las piezas).

Como se tienen 6 varillas correspondientes a sendos pares de puntos A y B, nos queda un sistema lineal de 6 ecuaciones con 6 incógnitas, cuya solución es inmediata.

Una vez que se tienen esas incógnitas, ya se conocen dos vectores de la base móvil:

$$\vec{l}_2 = R_{xx}\vec{l}_1 + R_{yx}\vec{j}_1 + R_{zx}\vec{k}_1 \quad (18)$$

$$\vec{j}_2 = R_{xy}\vec{l}_1 + R_{yy}\vec{j}_1 + R_{zy}\vec{k}_1 \quad (19)$$

El tercer vector de la base se obtiene con la condición de que forman un triedro dextrógiro:

$$\vec{k}_2 = \vec{l}_2 \times \vec{j}_2 \quad (20)$$

La posición de un punto cualquiera de la plataforma móvil, no necesariamente del mismo plano, es:

$$\overline{OP} = \overline{OG} + \overline{GP} = x\vec{l}_1 + y\vec{j}_1 + z\vec{k}_1 + X_P\vec{l}_2 + Y_P\vec{j}_2 + Z_P\vec{k}_2 \quad (21)$$

Si lo que se desea es la ecuación del plano es, en forma vectorial:

$$\vec{k}_2 \cdot \overline{GP} = 0 \quad \Rightarrow \quad R_{xz}(x_P - x) + R_{yz}(y_P - y) + R_{zz}(z_P - z) = 0 \quad (22)$$

Matlab & Simulink permiten la obtención de la matriz de rotación en cada instante de tiempo. Si se supone conocida la misma, el cálculo de la posición de cada punto de la plataforma móvil se llevaría a cabo como se indica a continuación.

La longitud de cada pata, siguiendo la nomenclatura expuesta en los párrafos anteriores, se calcularía como se indica en la ecuación (23).

$$L = |R \cdot \overline{GB} + \overline{OG} - \overline{OA}| \quad (23)$$

La posición de cada punto de la plataforma móvil se obtendría según (24).

$$\overline{GP} = \overline{OG} + \overline{GB} \cdot R \quad (24)$$

Siendo \overline{GP} la posición de un punto de la plataforma móvil respecto a los ejes ligados a la base.

2.3 Cinemática en Matlab & Simulink

Es necesario llevar a cabo la implementación de la cinemática expuesta en los apartados anteriores, ya que por un lado la inversa calcula la longitud que debe tener el actuador en cada instante para conseguir posicionar la plataforma en la posición deseada, y la directa nos permite calcular la posición de la misma a partir de la matriz de rotación y así simular el movimiento.

Atendiendo a la inversa, esta está implementada mediante bloques Simulink directamente.

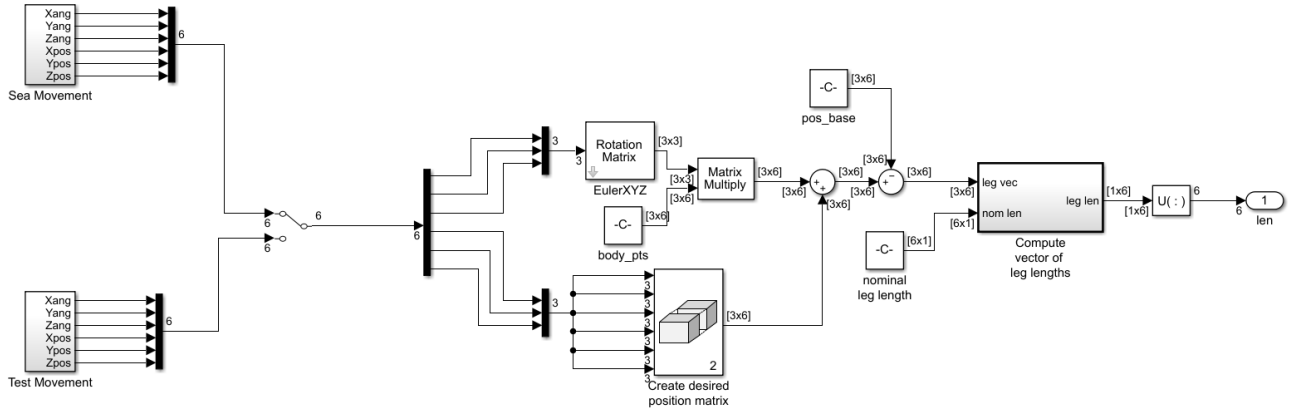


Figura 12. Implementación cinemática inversa en Simulink

Ese bloque proporciona el cambio de longitud que debe tener cada actuador respecto a la longitud nominal. Implementa la ecuación (1).

Respecto a la cinemática directa, se ha implementado en un script Matlab “*Mov_TopPlate.m*.” Dicho script puede consultarse en el Anexo C. En el mismo se usa información de la matriz de rotación obtenida en cada paso de simulación usando el bloque *Body Sensor* activando la opción *Rotation Matrix [3x3]*. Ver Figura 13 y Figura 14.

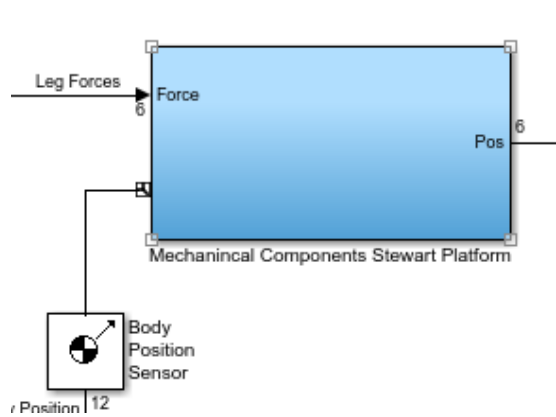


Figura 13. Bloque *Body Sensor* en Simulink

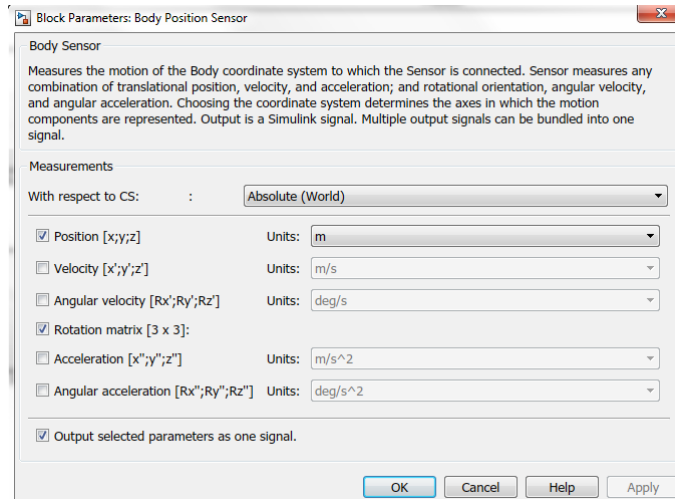


Figura 14. Configuración del bloque *Body Sensor*

La ejecución del script necesita un procesamiento previo de los valores que provienen de la simulación, contenidos en la variable “cdgpos”. Se lleva a cabo ejecutando el archivo *R_matrix.m*. En el mismo se calcula, primero la longitud total de cada actuador en cada instante de simulación, y después la posición de los puntos a los que están anclados dichos actuadores en la plataforma móvil, ecuación (24).

2.4 Configuración geométrica y dinámica de la plataforma

Atendiendo a los parámetros expuestos en la Figura 15, los parámetros que configuran la geometría y la dinámica que tendrá la plataforma se presentan en la Tabla 1. Dichos parámetros condicionan la elección de los parámetros de control, en el caso aquí expuesto las ganancias de los controladores PID que controlarán el movimiento de la misma.

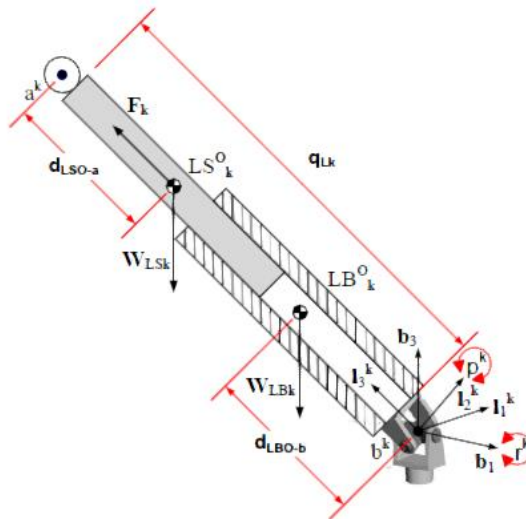


Figura 15. Geometría del actuador lineal [10]

	Parámetro	Valor	Unidad
Aceleración Gravedad	g	9.81	$\frac{m}{s^2}$
Geometría	d_{LB^o-b}	1.1827	m
	d_{LS^o-a}	1.1827	m
Radio base	r_B	3	m
Radio plataforma móvil	r_S	1	m
Espesor plataforma móvil	l_S	0.05	m
Material	ρ	7747	$\frac{kg}{m^3}$
Radio cilindro superior	r_{LS}	0.03	m
Longitud cilindro superior	l_{LS}	2.3654	m
Radio exterior cilindro inferior	r_{LBout}	0.05	m
Radio interior cilindro inferior	r_{LBin}	0.03	m
Longitud cilindro inferior	l_{LB}	2.3654	m
Masa plataforma móvil	m_S	1217	kg
Masa cilindro superior	m_{LS}	52	kg
Masa cilindro inferior	m_{LB}	92	kg

Tabla 1. Parámetros geométricos y dinámicos de la plataforma

El sistema completo (plataforma, actuadores, IMU y sistema de control) se ha modelado en Matlab & Simulink, como se refirió en la introducción. En este apartado se expone la parte correspondiente a la plataforma y los actuadores.

El modelo completo se ha descargado de la web de Matlab², el cual uso la Toolbox Simscape para el modelado de sistemas físicos en el entorno Simulink. Además, el modelo carga parámetros definidos en el Workspace de Matlab. Los parámetros se definen en el archivo “*StewartPlatformSetup.m*”, expuesto en el Anexo C.

El modelo mecánico de la plataforma se muestra en la Figura 16.

Los componentes que forman la plataforma son:

- Top Plate. Modela el comportamiento de la plataforma móvil. Los parámetros que definen su comportamiento dentro del script expuesto en el Anexo C son:
 - top_mass
 - top_inertia
- Leg. Modelan la respuesta de los actuadores que controlan el movimiento de la plataforma móvil. Los bloques de Simscape que lo componen se presentan en la Figura 17. Los parámetros que definen su geometría y dinámica en el script de configuración son:
 - lower_leg(i).origin
 - lower_leg(i).end_point
 - lower_leg(i).rotation
 - upper_leg(i).origin
 - upper_leg(i).end_point
 - upper_leg(i).rotation

² <https://es.mathworks.com/company/newsletters/articles/creating-a-stewart-platform-model-using-simmechanics.html>

- pos_top
 - pos_base
- Ground. Modela la base del sistema (cubierta del buque en este caso).

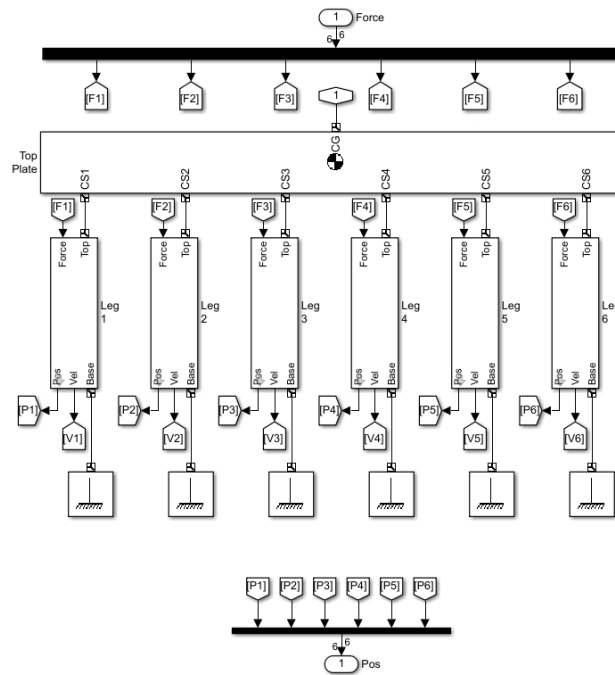


Figura 16. Modelo Simulink de la Plataforma Stewart

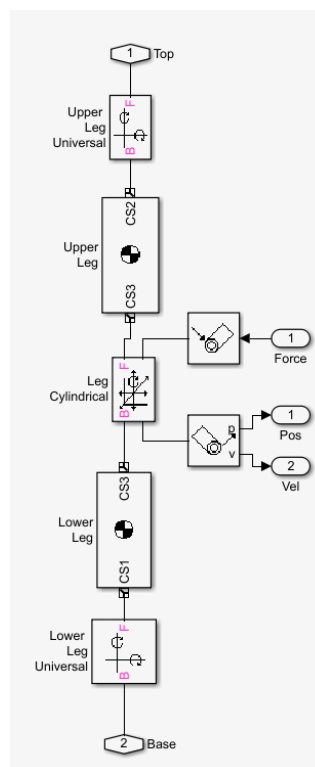


Figura 17. Modelo Simulink del actuador

A continuación, se muestra una representación tridimensional de la plataforma configurada. En ella se ha representado la pista de aterrizaje del UAV.

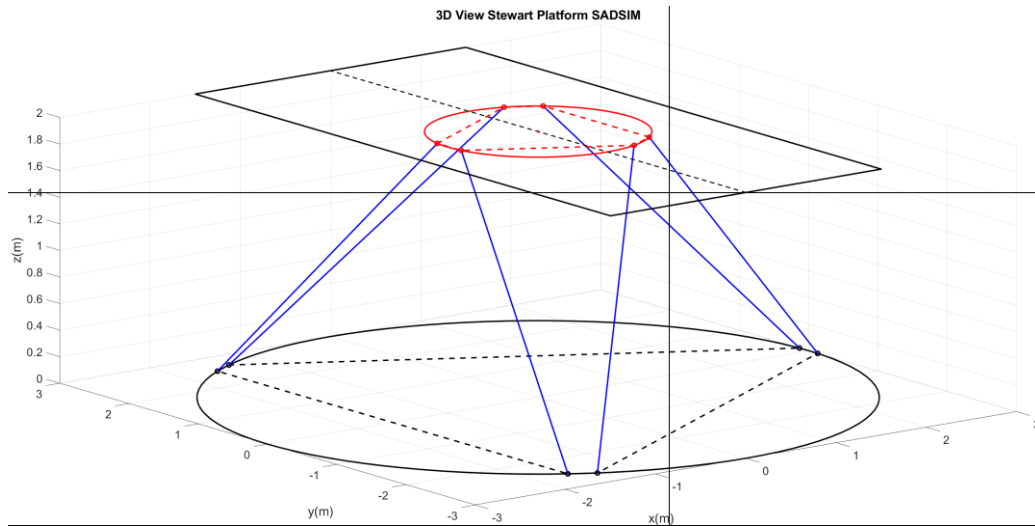


Figura 18. Representación 3D Plataforma Stewart y pista de aterrizaje

3 EL SISTEMA DE CONTROL

Es preferible obtener una respuesta razonablemente aproximada pero rápida que le indique si el diseño funciona o no, que invertir más tiempo y obtener el mismo resultado sólo que con más decimales.

Robert L Norton

Existen múltiples soluciones de control aplicables al sistema: controladores P, PI, PD o PID, control en variables de estado, sistemas de control robustos o sistemas de control predictivo. La elección de uno u otro depende de múltiples factores. El diseño de un sistema que permite un control más robusto aumenta de forma considerable la complejidad, y por tanto el tiempo y recursos necesarios para ello. Además, la aplicación a la que esté destinada el sistema establecerá las necesidades del mismo, es decir, exigirá que sea necesario usar unos modelos u otros para cumplir con los requisitos del sistema.

Aunque durante los estudios iniciales de esta tarea se valoraron diferentes opciones de control-control en variables de estado o control óptimo-, finalmente se ha optado por el diseño de un sistema de control basado en controladores PID.

Las razones que justifican esta elección son:

- 1) La experiencia en el uso de controladores PID, los cuales demuestran un buen funcionamiento en múltiples campos. Además, según bibliografía consultada la mayor parte de los diseños usan estos controladores.
- 2) Es un controlador simple de manera que, si con el PID se resuelve el problema de control, no es necesario acudir a soluciones más complejas. Además, el PID presenta buenas propiedades de robustez.
- 3) El controlador PID incluye por definición un término predictivo, término D. Dicho término anticipa el error futuro (depende del error actual y de cómo varía este). Esto responde al requisito de predictividad del proyecto.

Las características, el modelado y la sintonización del controlador se desarrollan en lo que sigue.

3.1 El controlador PID

El controlador PID aúna las ventajas que presentan los controladores P, PI y PD por separado, esto es:

- La parte integral añade un polo al origen, cambiando el tipo del sistema (mejora el error en régimen permanente).
- La parte derivativa mejora el transitorio reduciendo la sobreoscilación (SO).
- La parte proporcional controla el tiempo de subida (rapidez de la respuesta).

La ley de control que modela dicho controlador se muestra en (25):

$$\Delta u(t) = K_p (e(t) + T_d \frac{de(t)}{dt} + \frac{1}{T_i} \int_0^t e(\tau) d\tau) \quad (25)$$

La función de transferencia, que es como se presentará en Simulink, se muestra en (26):

$$C(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) = K_p + K_d s + \frac{K_i}{s} \quad (26)$$

Existen numerosos métodos para la sintonización de los controladores: por cancelación de polos, por tabla o mediante un diseño frecuencial, entre otros. En el desarrollo de este trabajo, debido a que no se conoce la función de transferencia del sistema, además de ser un sistema multivariable, se ha optado por un diseño frecuencial.

El controlador PID implementado en Matlab & Simulink incluye una variación en el término derivativo, incluyendo un polo más al sistema. Su función de transferencia se presenta en (27).

$$C(s) = K_p + \frac{K_i}{s} + K_d \frac{Ns}{s + N} \quad (27)$$

La introducción de dicho término queda justificada porque la ganancia en el diagrama de Bode del término derivativo crece de forma indefinida. Ese término, según el valor que tenga (en general $N \in [10^2, 10^3]$), hace que la misma se establezca en un valor constante a partir de una frecuencia. El efecto final que tiene es que el controlador tenga una respuesta más suave a frecuencias altas. Ver Figura 19.

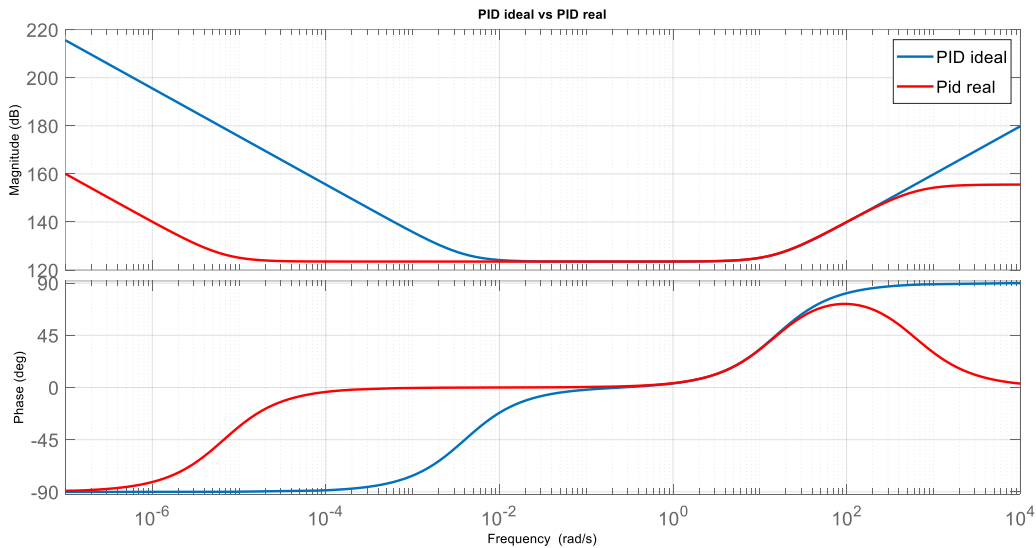


Figura 19. Bode PID ideal (azul) frente a PID real (rojo)

3.2 Linealización del modelo

El uso de herramientas y técnicas lineales para sintonizar el controlador hace que sea necesario linealizar el sistema respecto a un punto de operación. Esto asegurará que el controlador funcionará de forma adecuada en torno a dicho punto, empeorando su respuesta a medida que nos alejemos del mismo.

Linealizado el sistema respecto a un punto, la respuesta del sistema de control debe ser:

$$u = \left(K_p + \frac{K_i}{s} + K_d \frac{Ns}{s + N} \right) e(s) + U_0 \quad (28)$$

Para el diseño aquí tratado, el punto respecto al cual se ha linealizado ha sido aquel que mantiene el cdg de la plataforma en la posición (0, 0, 2), y esta se mantiene en posición horizontal. Para este caso, la longitud de los actuadores es de 3.1538 m y la fuerza que debe aplicar cada uno es de 4.0788×10^3 N. La señal de control, para este caso, se corresponde con el valor de la fuerza que debe aplicar el actuador, es decir, esa fuerza

es U_0 .

Para la obtención de esa fuerza, es decir, el valor de U_0 , existen dos posibilidades:

- 1) Configurar el modelo de la plataforma en Simulink de forma que nos proporcione la fuerza necesaria para mantenerla en la posición deseada.
- 2) A partir de unos valores de ganancia capaz de controlar el sistema, simular el tiempo suficiente para que el error en régimen permanente sea nulo y observar el valor que proporciona el PID para esa situación.

Ambas posibilidades, como es de esperar, se ha comprobado que convergen al mismo valor.

En la Figura 20 se presenta la plataforma configurada para obtener el valor de equilibrio.

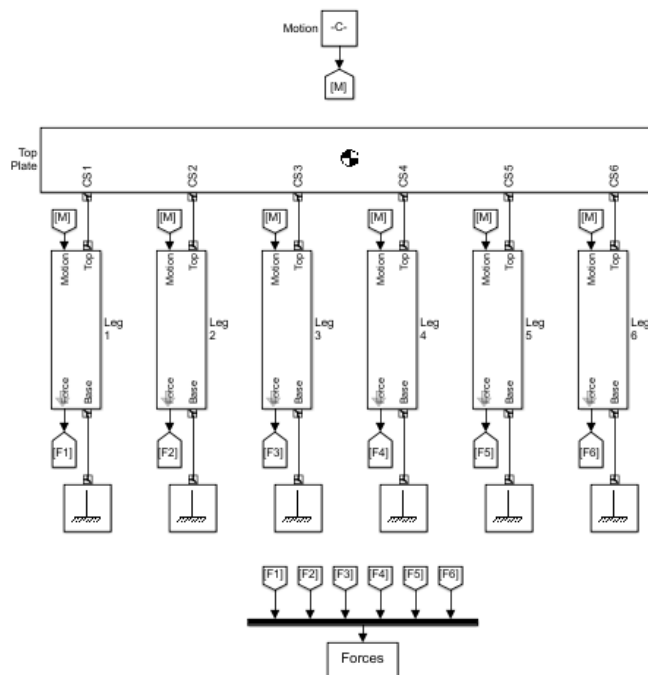


Figura 20. Configuración de la plataforma en el punto de equilibrio

Si se decide aplicar la segunda opción, los resultados que se obtienen se muestran en la Figura 21.

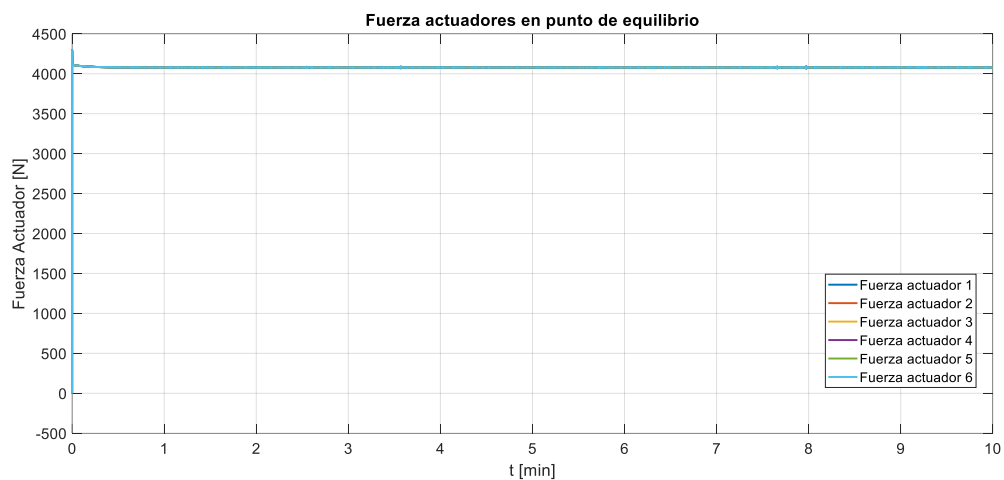


Figura 21. Señal de control U_0 (Fuerza) en el punto de linealización

Introduciendo este valor, el sistema es capaz de mantener la posición nominal con un error prácticamente nulo. Ver Figura 22.

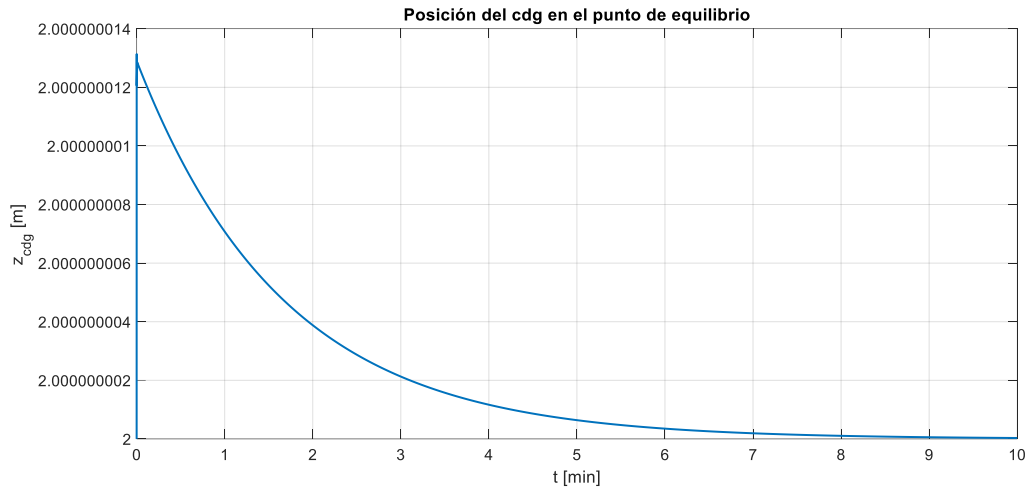


Figura 22. Estabilización de la plataforma en torno al punto de linealización

Este valor se introduce como una constante en el modelo Simulink.

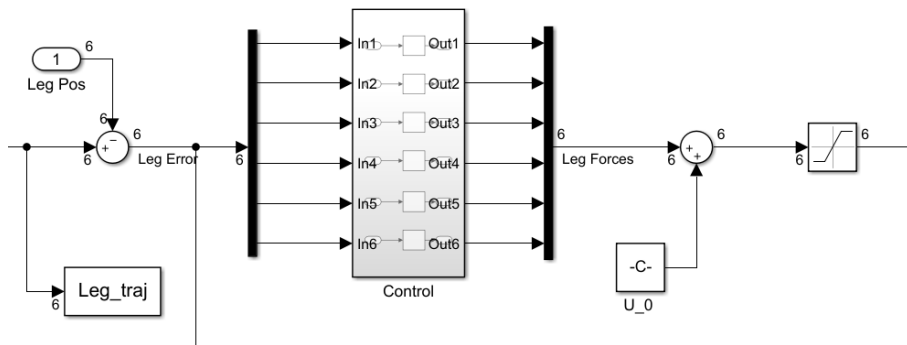


Figura 23. Introducción de U_0 en el modelo Simulink

Para linealizar modelos físicos desarrollados en Simulink mediante bloques Simscape, Matlab incluye el comando `linmod(File.mdl,[],U)` siendo U el punto en el que se linealiza el sistema.

Tras la linealización, es conveniente obtener los polos y ceros del sistema y así saber si es estable o inestable. El sistema, como ya se ha referido, es multivariable, tiene seis entradas y seis salidas, y cada entrada tiene relación con cada salida, es decir, existen términos cruzados. Todos los actuadores son iguales, por lo que solo es necesario sintonizar un PID. Siguiendo las recomendaciones expuesta en [11], los polos y ceros del sistema linealizado, Plataforma+Actuadores³, se muestran en la Figura 24.

³ Polos y ceros de la función de transferencia correspondiente a la entrada 1 y salida 1. Es decir, es la salida del sistema si se excita el mismo grado de libertad en la entrada.

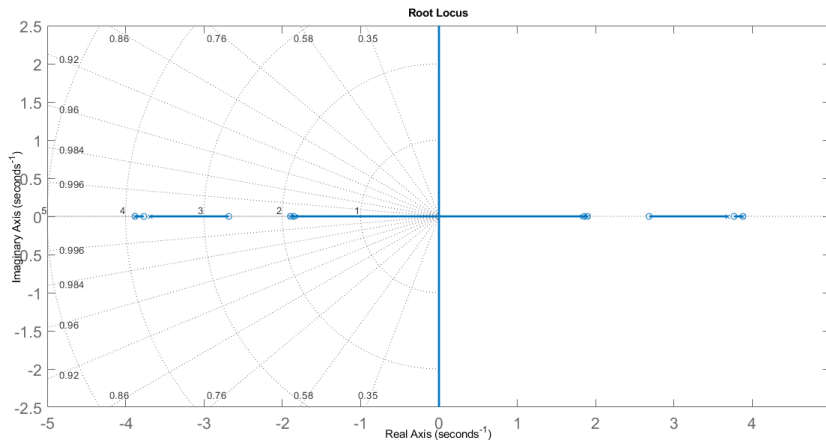


Figura 24. Polos y ceros del sistema linealizado

Polos del sistema linealizado	Ceros del sistema linealizado
$-3.8806 \pm 1e^{-7}i$	-3.8806
-3.6874	-3.7677
-1.8969	-2.6824
-1.8969	-1.8969
-1.8274	-1.8529
$3.8805 \pm 1e^{-7}i$	3.8806
3.6874	3.7677
$1.8969 \pm 1e^{-7}i$	2.6824
1.8274	1.8969

Tabla 2. Polos y ceros del sistema linealizado

El sistema tiene polos en el semiplano positivo, lo cual indica que es inestable –Ver Figura 25. Existen ceros muy cerca de los polos positivos, pero no son exactamente los mismos, por lo que no los cancelan. Sin embargo, la inestabilidad podría estar asociada a los errores de la linealización numérica a través del comando linmod. Para demostrar esto puede simularse durante un largo periodo de tiempo la respuesta del sistema ante una entrada en escalón. Si la respuesta del sistema tiende a $\pm\infty$, el sistema será inestable, si la salida se estabiliza en un valor finito, será estable.

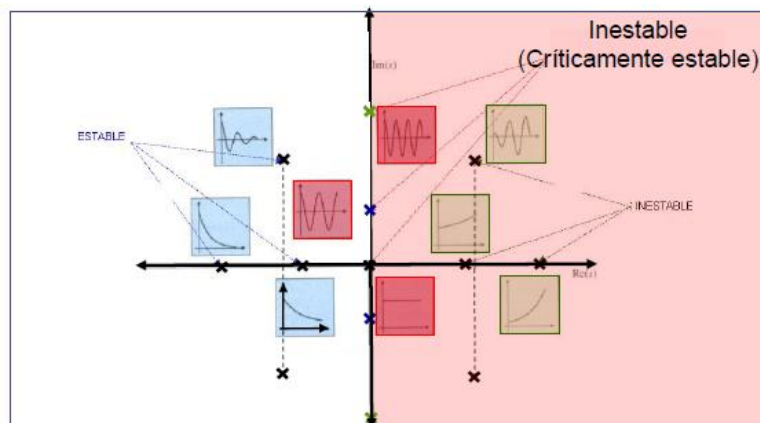


Figura 25. Respuesta de un sistema según sus polos [12]

Llevando a cabo estas simulaciones, en la Figura 26 y Figura 27 se demuestra como el sistema completo es capaz de mantener una posición tras una señal escalón. Se expone en primer lugar la respuesta de la plataforma, y en segundo el error, que es la diferencia entre la posición deseada del actuador y la que tiene realmente. Si el sistema fuera inestable, simulando durante suficiente tiempo se expondría dicha inestabilidad.

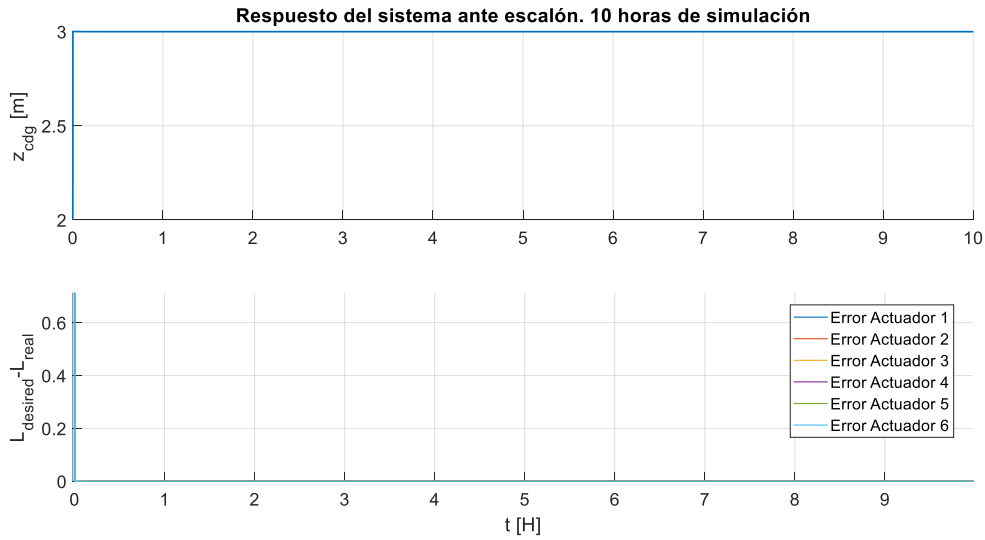


Figura 26. Simulación de la respuesta del sistema ante escalón durante 10 horas

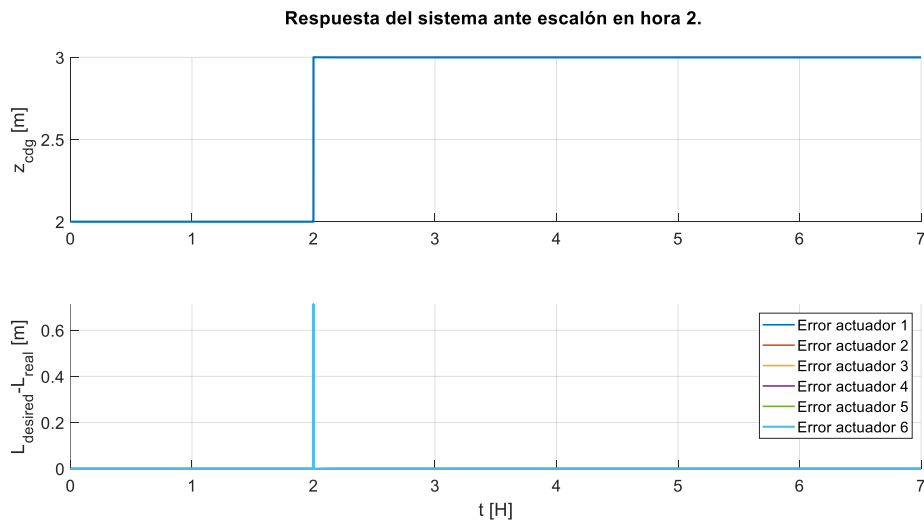


Figura 27. Simulación de la respuesta del sistema ante escalón en hora 2 durante 7 horas

3.3 Diseño frecuencial del controlador

El diseño frecuencial de controladores está basado en la respuesta del sistema (en este caso controlador + plataforma) cuando este es excitado a distintos valores de frecuencia. Según [13], el diseño de controladores en el dominio de la frecuencia se puede expresar como “encontrar un controlador $C(s)$ que modifique la forma de la respuesta frecuencial del sistema de manera que su respuesta en bucle cerrado cumpla con las especificaciones”.

Un sistema puede caracterizarse por su diagrama de Bode, en el cual se presenta el módulo y la fase en función de la frecuencia. El sistema trabajará en bucle cerrado, aunque el diagrama de Bode que lo caracteriza se estudia en bucle abierto, ya que la respuesta frecuencial en bucle abierto proporciona información importante de la respuesta temporal en bucle cerrado (estabilidad, sobreoscilación, tiempo de subida, etc).

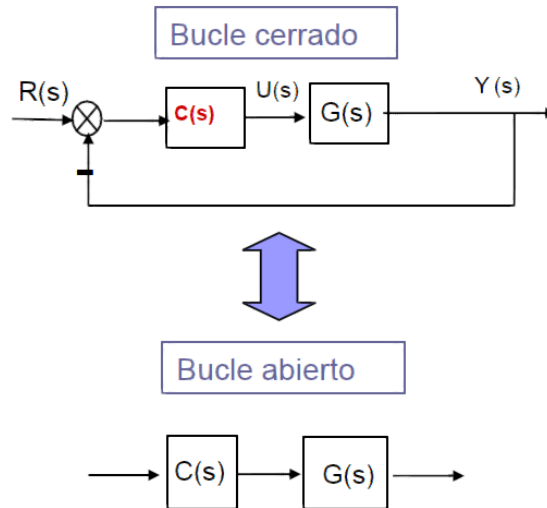


Figura 28. Diagrama Bucle cerrado vs Bucle abierto [13]

Para que un sistema sea estable, el margen de fase tiene que ser positivo, es decir, mayor que -180° , y el margen de ganancia también positivo.

Dada una función de transferencia $G(s)$, el módulo de la misma es:

$$|G(j\omega)|_{dB} = 20 \log_{10}(|G(j\omega)|) \quad (29)$$

La fase:

$$\angle G(j\omega) = \text{atan} \frac{\text{Im}(G(j\omega))}{\text{Re}(G(j\omega))} \quad (30)$$

El margen de ganancia, GM, es el valor que habría que sumar a la ganancia (en dB) para llegar a 0 dB cuando la fase es de -180° .

La frecuencia de corte indica la frecuencia a partir de la cual el sistema atenúa la señal de entrada. Antes de la misma el sistema amplifica. Ver Figura 29.

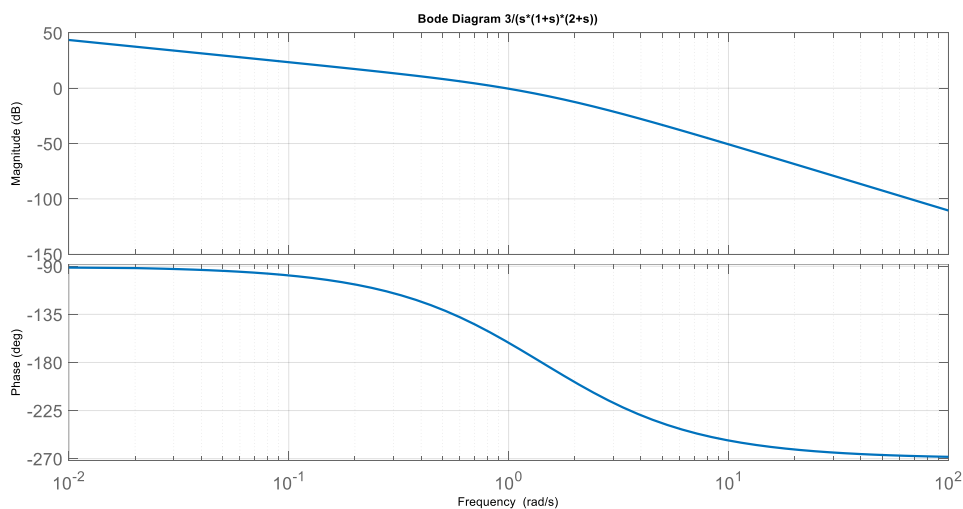


Figura 29. Diagrama de Bode de sistema estable

Para diseñar el controlador hay que tener en cuenta:

- Si el margen de fase disminuye la sobreoscilación aumenta:

$$M_f \downarrow \rightarrow SO \uparrow$$

- Si la frecuencia de corte, ω_c , aumenta el tiempo de subida disminuye, es decir, el sistema responde más rápido:

$$\omega_c \uparrow \rightarrow t_s \downarrow$$

El tiempo de subida se puede calcular de forma aproximada según (31):

$$t_s \approx \frac{\pi}{2 \omega_c} \quad (31)$$

Las especificaciones dependerán de la aplicación. Para el caso aquí de interés, es necesario que la respuesta de la plataforma sea rápida y la sobreoscilación lo más pequeña posible o nula. El objetivo es que vuelva a la posición horizontal tras una perturbación lo más rápido posible ya que en ella estará instalada la pista de aterrizaje del UAV.

Matlab dispone de la herramienta Control System Designer, la cual proporciona distintas posibilidades para el diseño de controladores. Esta utilidad permite introducir polos y ceros en el sistema, es decir, modelar el controlador, y visualizar el efecto del mismo en el diagrama de Bode, así como la respuesta ante escalón del sistema. Por tanto, el diseño del controlador se ha basado en el método del lugar de las raíces, que mediante la elección de los polos y ceros del PID (la herramienta permite hacerlo de forma interactiva), se modifica la respuesta del sistema. Además, puede obtenerse la respuesta del sistema ante una entrada en escalón. Con esta información, se ha buscado en primer lugar que el **margen de fase** en el Bode fuese **mayor que 0°**, lo que no asegura que el sistema será estable, y, como segundo objetivo, que la **sobreoscilación y el tiempo de establecimiento** sean **lo más pequeño posible**.

Atendiendo a lo expuesto en el apartado 3.1, el controlador que se va a implementar es un PID con filtro en el término derivativo [11], es decir con un polo más incluido en el término derivativo.

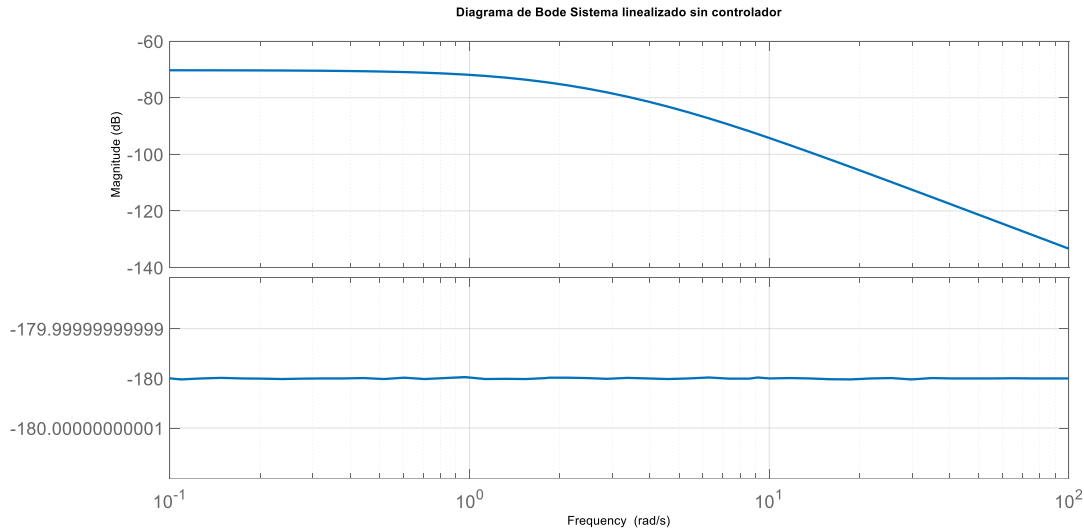


Figura 30. Bode del modelo linealizado sin controlador

Con los objetivos establecidos en los párrafos anteriores, los polos y ceros del controlador que proporcionan una respuesta adecuada son los expuestos en la Tabla 3.

Polos	Ceros
0 (Integrador)	-0.004
-600	-15

Tabla 3. Polos y Ceros PID

Un polo cercano a un cero implica la cancelación de sus efectos, por lo que respecto a los introducidos por el

controlador, el efecto introducido por el integrador será prácticamente nulo debido a que existe un cero muy cercano, -0.004.

Los parámetros que definen este controlador son:

$$K_p = 1.5 \times 10^6$$

$$K_i = 6 \times 10^3$$

$$K_d = 9.75 \times 10^4$$

$$N = 600$$

Para obtener dichos parámetros, a partir de la función de transferencia que tenemos del controlador, hay que ejecutar las siguientes líneas de código:

1. `[z,p,k] = zpkdata(cBiproperOpt) ;`
2. `A = -p{1,1}(2) ;`
3. `Ki = k*z{1,1}(1)*z{1,1}(2)/A ;`
4. `Kp = -(k*(z{1,1}(1) + z{1,1}(2)) + Ki)/A ;`
5. `Kd = (k - Kp)/A ;`

Cuando se usa la función `zpkdata()`, el argumento de la misma es el nombre que le hayamos dado al controlador al exportarlo desde la herramienta Control System Designer.

El diagrama de Bode, en lazo abierto, que ahora presenta el sistema es el expuesto en la Figura 31.

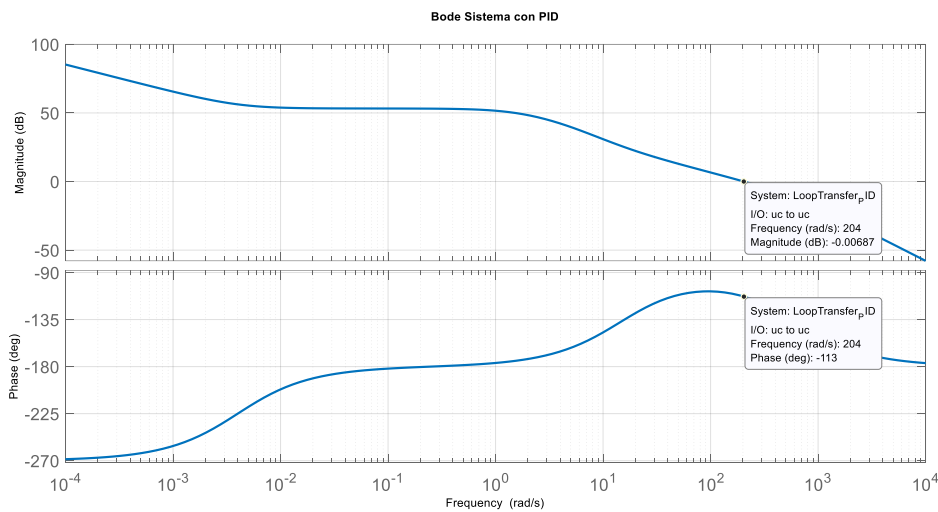


Figura 31. Bode del sistema (en bucle abierto) con controlador PID

A modo de aclaración sobre qué significa la respuesta del sistema en bucle cerrado, se expone a continuación su diagrama de Bode. Se observa como es sistema no atenúa ni amplifica la señal de entrada hasta una frecuencia de $204 \frac{rad}{s}$ aproximadamente. Para frecuencias superiores, se produce una atenuación de 40 dB por década.

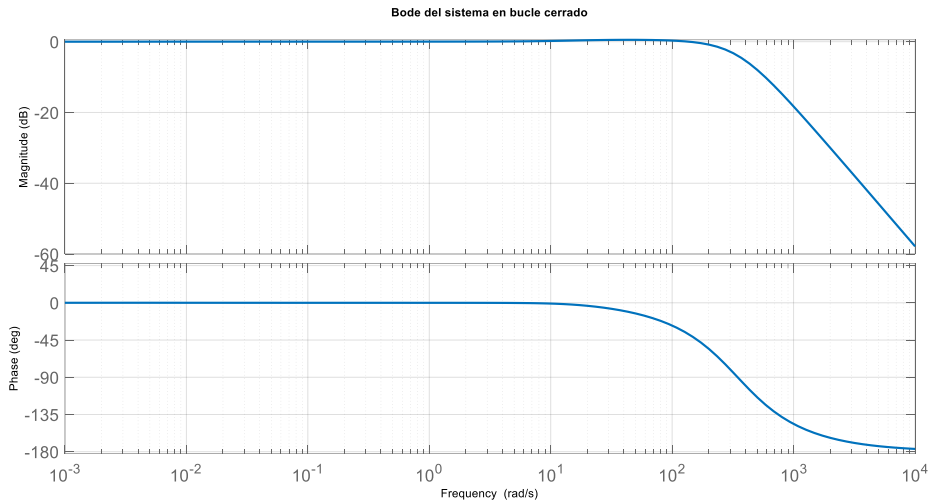


Figura 32. Bode del sistema en bucle cerrado con PID

El margen de fase es de 67° . La frecuencia de corte está en 204 rad/s. La respuesta del modelo linealizado ante escalón es la presentada en la Figura 33.

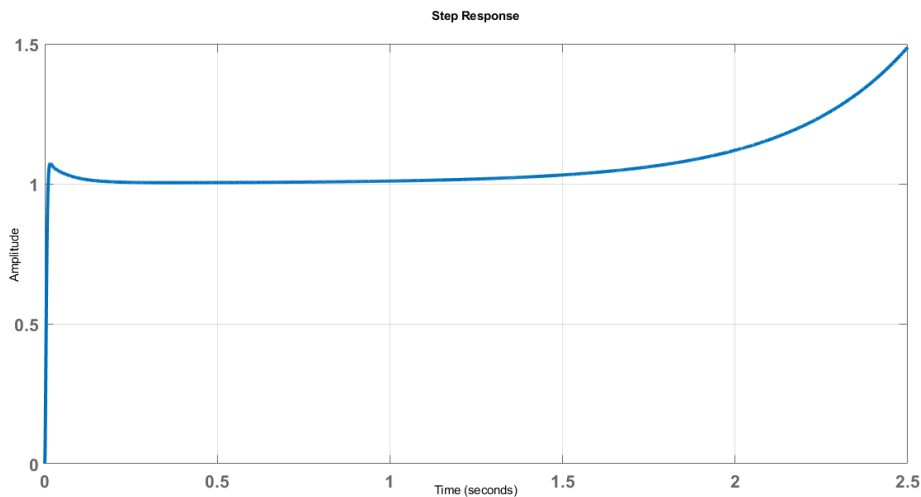


Figura 33. Respuesta del sistema linealizado ante escalón.

Se observa como la respuesta del **sistema linealizado** ante una señal escalón se vuelve inestable a partir del segundo 1. Este comportamiento era esperado, ya que existen polos en el semiplano derecho. Sin embargo, como se demuestra en el apartado 3.2, el sistema es estable. La inestabilidad es ficticia y debida a las aproximaciones de la linealización numérica del comando “*linmod*”.

Para caracterizar la respuesta del sistema, es necesario conocer el tiempo de subida y la sobreoscilación. Para ello se simula la respuesta del sistema real, es decir, el no linealizado, ante una entrada en escalón.

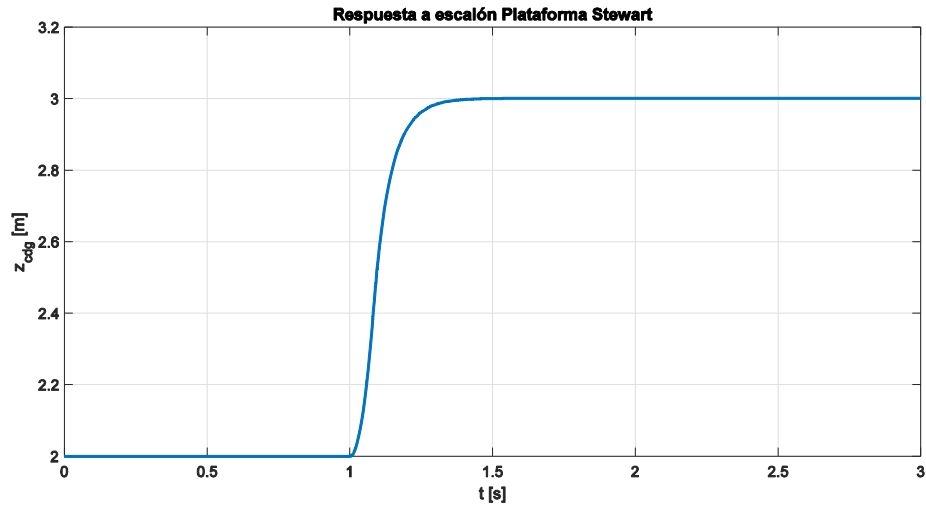


Figura 34. Respuesta ante escalón del sistema no lineal

Se observa que la respuesta ante una entrada en escalón no presenta sobreoscilación, y el tiempo de subida es aproximadamente 0.4 s.

$$SO = 0\%$$

$$t_e \approx 0.4 \text{ s}$$

En el capítulo 5 se analiza de forma exhaustiva el comportamiento del controlador, y se corrobora con escenarios realista la respuesta del mismo.

4 DEFINICIÓN DE LAS CONDICIONES OPERATIVAS

El mar dará a cada hombre una nueva esperanza, como el dormir le da sueños.

Cristóbal Colón

La definición de las condiciones operativas bajo las que funcionará la plataforma depende de la respuesta del barco a la excitación marina. Aunque **dicho estudio no es objeto de este trabajo**, es conveniente incluir una breve introducción al modelado de dicho comportamiento.

Por otro lado, conocida la respuesta del barco al estado del mar, la geometría de la plataforma limitará el tamaño de ola máximo que podrá compensar (el tamaño de ola está directamente relacionado con la amplitud de los movimientos del barco, aunque también influirá la dirección con la que estas llegan al buque). Por tanto, es necesario estudiar a partir de la respuesta del barco en distintas situaciones, donde se alcanzan los límites.

Por último, es de interés de cara a una implementación real tener en cuenta los límites operativos relacionados con el UAV que se vaya a operar. Se recuerda que el tamaño de ola, y el estado del mar, esta relacionado con la velocidad del viento, en consecuencia, es posible que haya situaciones en las que el UAV no pueda realizar la operación de aterrizaje por la componente de viento cruzado.

4.1 Respuesta de un barco a la excitación marina

Para llevar a cabo la simulación del movimiento del barco, es necesario previamente modelar el comportamiento del mar. Como ya se ha comentado en los párrafos introductorios de este apartado, no es objeto de este proyecto el estudio y desarrollo de esta tarea, ya que **fue desarrollada por otra de las empresas** que forman el consorcio SADSIM. Por lo tanto, aunque aquí se presente información acerca del modelado del movimiento marino y la respuesta del barco al mismo, la mayor parte del contenido se ha tomado del documento [14].

4.1.1 Introducción al modelado del movimiento marino

La empresa encargada del desarrollo del modelo de movimiento marino optó por una descripción estadístico-geométrica del estado del mar. Esta teoría se caracteriza por describir el comportamiento del oleaje según parámetros estadísticos, en el que el campo de alturas de las olas se considera una variable aleatoria de la posición sobre el plano horizontal y del tiempo, $h(X, t)$.

Los modelos estadísticos recurren a la posibilidad de descomponer la función $h(X, t)$ en una suma de funciones armónicas, senos y cosenos. Además, de cara a la implementación computacional se hace uso de la Transformada de Fourier (FFT).

La representación FFT del campo de alturas de ola $h(X, t)$ es una función de la posición en el plano horizontal, $X = (x, y)$, y puede representarse como la superposición de sinusoides de amplitud compleja y dependiente del tiempo:

$$h(X, t) = \sum_k \tilde{h}(k, t) \cdot e^{ikx} \quad (32)$$

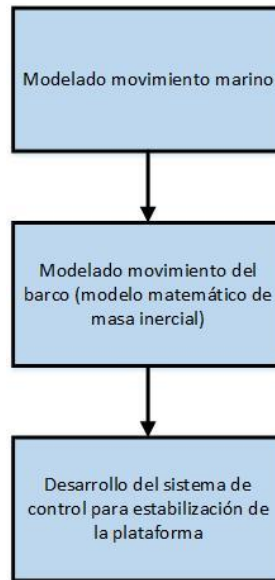


Figura 35. Diagrama de desarrollo del proyecto

La amplitud de las olas se puede modelar a partir una función de densidad de probabilidad $P_{\tilde{h}}(k)$, existiendo distintos modelos empíricos que estiman dicha distribución. Para este proyecto se optó por el uso el espectro ondulatorio de JONSWAP, el cual se basa en el espectro de Pierson-Moskowitz [15].

El espectro de Pierson-Moskowitz es un modelo empírico desarrollado en 1964, y una de las formas más simples de distribución espectral de olas. Supone que si el viento sopla de forma continua y estacionaria durante un largo periodo de tiempo y sobre una superficie marina suficientemente extensa, habrá un momento en el que las olas en dicha región alcancen un punto de equilibrio con el viento, es decir, alcancen su desarrollo máximo y ahí permanecerían. Es lo que se conoce como mar totalmente desarrollado.

Pierson y Moskowitz emplearon para la obtención de su expresión matemática una serie de medidas reales de olas en el Atlántico Norte durante 1964. Sin embargo, el modelo aplica a cualquier región marina donde pueda producirse un estado del mar totalmente desarrollado.

$$S_{PM}(\omega) = \frac{A}{\omega^5} e^{-\frac{B}{\omega^4}} \quad (33)$$

Con $A = \frac{4 \pi^3 H_s^2}{T_z^4}$ y $B = \frac{16 \pi^3}{T_z^4}$

Donde:

- S_{PM} = Función densidad espectral de potencia de Pierson Moskowitz
- ω = Frecuencia angular [rad/s]
- H_s = Altura de ola [m]
- T_z = Periodo medio de paso por cero [s]

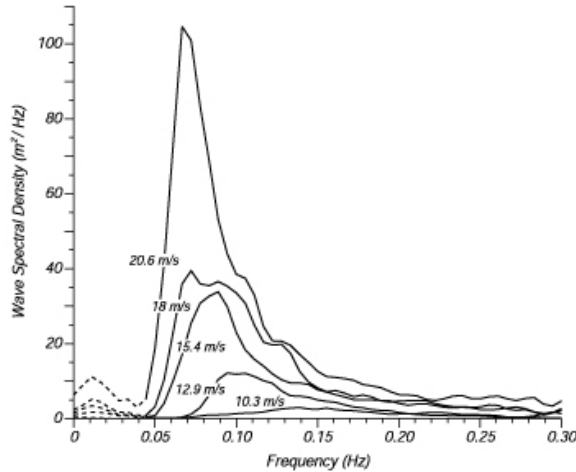


Figura 36. Espectro de ola según velocidad del viento [15]

Sin embargo, se puede demostrar que el estado del mar totalmente desarrollado, en el que se basan las hipótesis de Pierson-Moskowitz, nunca llega a producirse. Se observó que ocurrían fenómenos no lineales de interacción onda-onda incluso aunque el viento soplara de forma prolongada y en grandes extensiones del mar. Así, se concluyó que era necesario introducir un factor de corrección al espectro de Pierson-Moskowitz de manera que hubiera un mejor ajuste entre la función matemática y las medidas reales que se tomaron. A la corrección del espectro de Pierson-Moskowitz se le conoce como espectro JONSWAP (Joint North Sea Observation Project) [16].

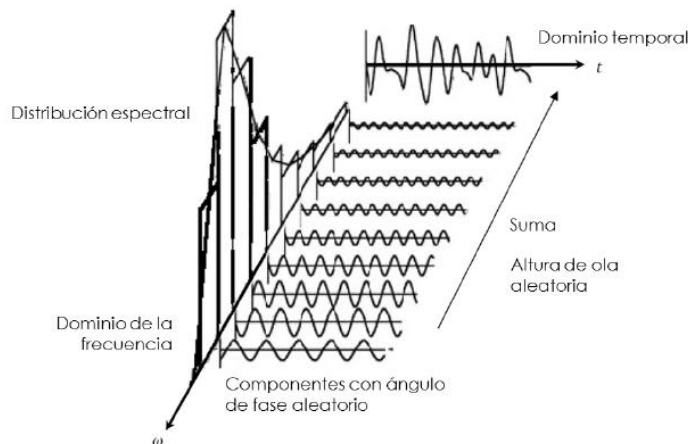


Figura 37. Modelado del comportamiento del mar a partir de la distribución espectral [14]

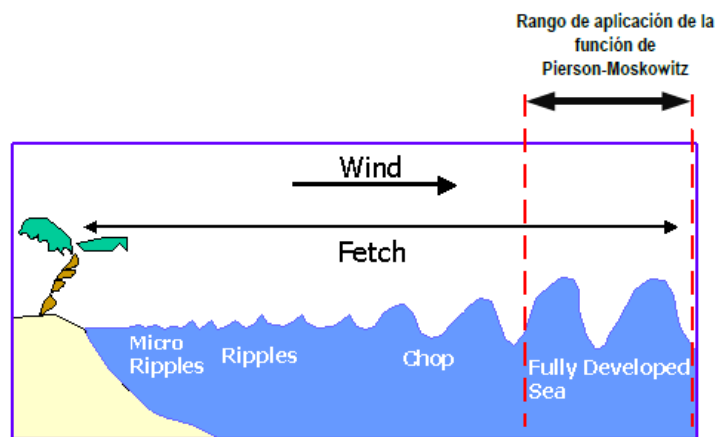


Figura 38. Rango de aplicación de la función de Pierson-Moskowitz [14]

La función densidad espectral de potencia para el espectro JONSWAP es:

$$S_{JS}(\omega) = nf \cdot S_{PM} \cdot \gamma^a \quad (34)$$

Con $a = e^{-\frac{1}{2}(\frac{\omega-\omega_m}{\sigma\omega_n})^2}$ y $\omega_m = \left(\frac{4}{5} B\right)^{\frac{1}{4}}$

Donde:

- S_{JS} = Densidad espectral de JONSWAP
- nf = Facto de normalización entre la función de JONSWAP y la de Pierson – Moskowitz
- γ = Parámetro de forma
- ω_m = Frecuencia angular modal
- σ = Parámetro (σ_a para $\omega < \omega_m$; σ_b para $\omega > \omega_m$)

Según bibliografía consultada:

- $\gamma = 3.3$
- $\sigma_a = 0.07$
- $\sigma_b = 0.09$
- $nf = 0.625$

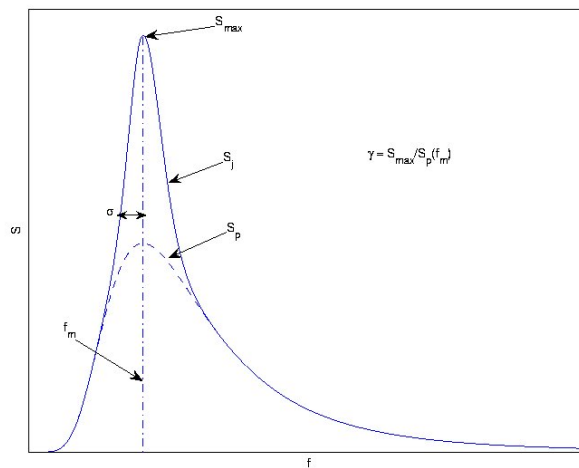


Figura 39. Espectro JONSWAP vs Pierson-Moskowitz [15]

Simulándose esta función espectral se tiene resultados como los mostrados en la Figura 40.

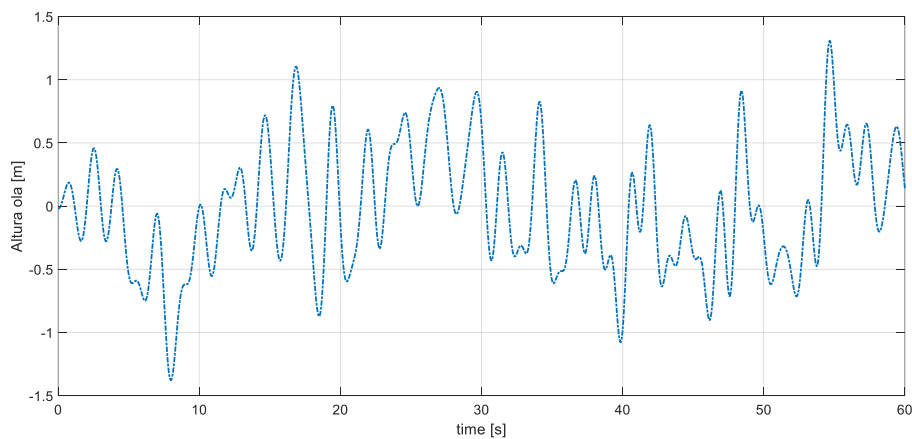


Figura 40. Simulación movimiento marino según función JONSWAP

4.1.2 Modelado del movimiento del barco

Antes de modelar el movimiento del barco, es necesario definir los grados de libertad del mismo. En general, se considerarán los seis grados de libertad habituales para un sólido libre, tres traslaciones y tres rotaciones según se expone en la Figura 41.

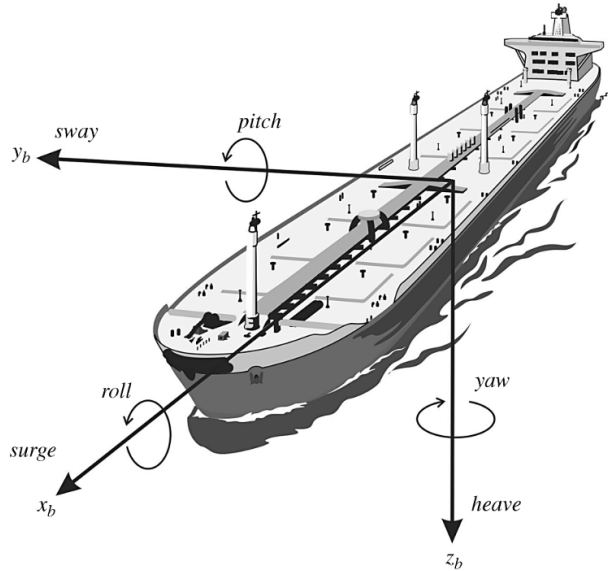


Figura 41. Grados de libertad para barco

Grados de libertad y movimientos asociados			
Descripción	Denominación en inglés	Denominación en español	Tipo de movimiento (uds.)
Movimiento vertical	Heave	Arfada	Traslación [m/s]
Movimiento lateral	Sway	Deriva	Traslación [m/s]
Movimiento longitudinal	Surge	Largada	Traslación [m/s]
Rotación alrededor de z_b	Yaw	Guiñada	Rotación [rad/s]
Rotación alrededor de x_b	Roll	Balanceo	Rotación [rad/s]
Rotación alrededor de y_b	Pitch	Cabeceo	Rotación [rad/s]

Tabla 4. Grados de libertad barco

En general, el movimiento de una embarcación es resultado de las fuerzas a las que se ve sometida. Las fuerzas pueden descomponerse en los grados de libertad definidos (para un sistema de referencia ligado al barco):

$$F_j = M_{ij} \frac{d^2 \eta_j}{dt^2} \text{ con } i, j = 1:6 \quad (35)$$

M_{ij} representa la matriz de masas, que corresponde con las masas equivalentes en cada grado de libertad traslacional y los momentos de inercia en los grados de libertad rotacionales.

$$M_{ij} = m, \text{ para } i = j = 1, 2, 3$$

$$M_{ij} = I_i, \text{ para } i = j = 4, 5, 6$$

$$M_{ij} = 0, \text{ para } i \neq j \text{ [16]}$$

F_j representa las fuerzas generalizadas que actúan en cada grado de libertad j . De forma general, estas fuerzas pueden clasificarse en dos grandes grupos: fuerzas gravitatorias y fuerzas hidrodinámicas. Haciendo un estudio más detallado de dichas fuerzas, se llega a la siguiente expresión:

$$F_j = F_j^{\text{masa hidrodinámica}} + F_j^{\text{memoria}} + F_j^{\text{viscosidad}} + F_j^{\text{restitución}} + F_j^{\text{difracción}} + F_j^{\text{Froude-krilov}}$$

La obtención de dichas fuerzas implica una complejidad alta, por lo que habitualmente se recurre a otras herramientas.

Considerese el caso de una ola armónica, la respuesta de la embarcación a dicha ola debe ser también armónica y de la misma frecuencia, si suponemos que el sistema-barco- es un LTI, aunque con un desfase. La respuesta además dependerá de las características físicas de la embarcación y de la dirección que forme la ola incidente con la misma. De forma general, una ola compuesta de una superposición de olas armónicas, provocaría también una superposición de armónicas en la respuesta de la embarcación.

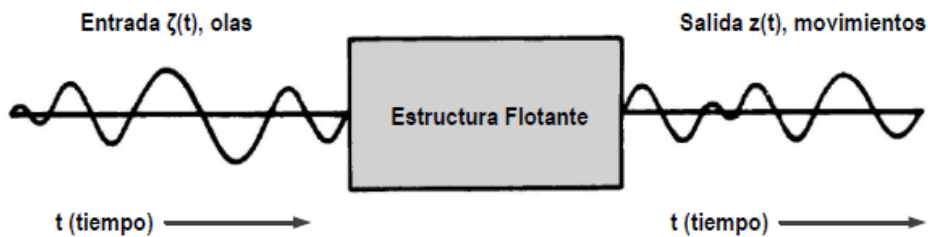


Figura 42. Esquema del comportamiento de la embarcación a la excitación marina [14]

En arquitectura naval la amplitud del movimiento (o respuesta) del buque para cada movimiento (roll, cabeceo, etc) usualmente se normaliza por la altura de la ola incidente y se denomina RAO –Response Amplitude Operator–Operador de Amplitud de Respuesta o Función de Transferencia. Esa transferencia tiene una amplitud y un desfase. Las RAOs solo pueden definirse cuando se considera que el movimiento del barco es lineal.

En las simulaciones se emplearán estos operadores para tratar el barco como una caja negra modelado por su función de transferencia.

Las funciones RAO dependen de la geometría de la embarcación y de su masa, pero también de la dirección de propagación de la ola (o frentes de olas) incidente en la misma. Incluyen de forma implícita todos los efectos comentados en el estudio analítico del movimiento. Por tanto, son funciones particulares y únicas de cada buque que caracterizan el movimiento del mismo ante la excitación marina.

$$RAO(\omega) = \frac{F_0}{C - (M + A(\omega)) \cdot \omega^2 + iB(\omega) \cdot \omega} \quad (36)$$

Donde:

- M es la masa estructural del barco.
- $A(\omega)$ es la inercia.
- $B(\omega)$ es el coeficiente de amortiguación lineal.
- C es el coeficiente de fuerza de restauración.

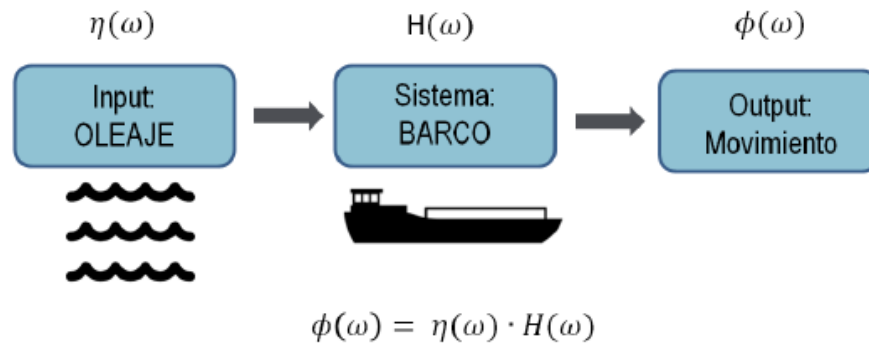


Figura 43. Función de transferencia que modela la respuesta del barco [14]

Las funciones se obtienen normalmente de forma experimental, mediante medidas y ensayos en un canal, o mediante software específico para cálculo numérico tales como Nemoh, DELFRAC o WAMIT. Este tipo de software necesita además como entrada la geometría de la embarcación y se emplea con regularidad en las fases de diseño preliminares de embarcaciones para modelar su comportamiento en la mar.

En el estudio realizado por la empresa encargada de realizar esta tarea, se han tomado las RAOs de [17]. En dicho documento se presenta las RAOs para distintos tipos de barcos y diferentes direcciones de frente de olas. En concreto, la respuesta aquí expuesta se corresponde a un buque tipo F con el frente de olas a 90° (olas laterales en la Figura 44).

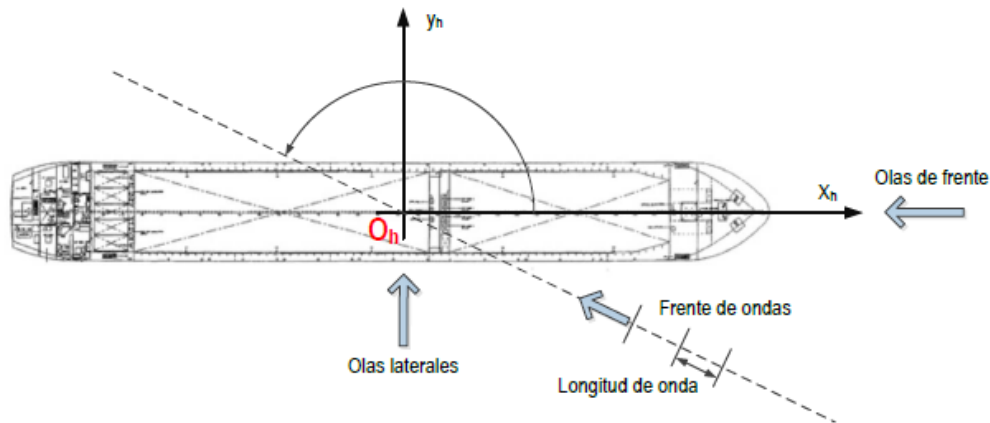


Figura 44. Dirección del frente de olas [14]

Carácterística	Tip D	Tipo F	Tipo G
Eslora [m]	291.13	190	180
Manga de flotación [m]	40,25	32	33
Calado [m]	15	11,60	11,60

Tabla 5. Caracterísitcas barcos estudiados en [17]

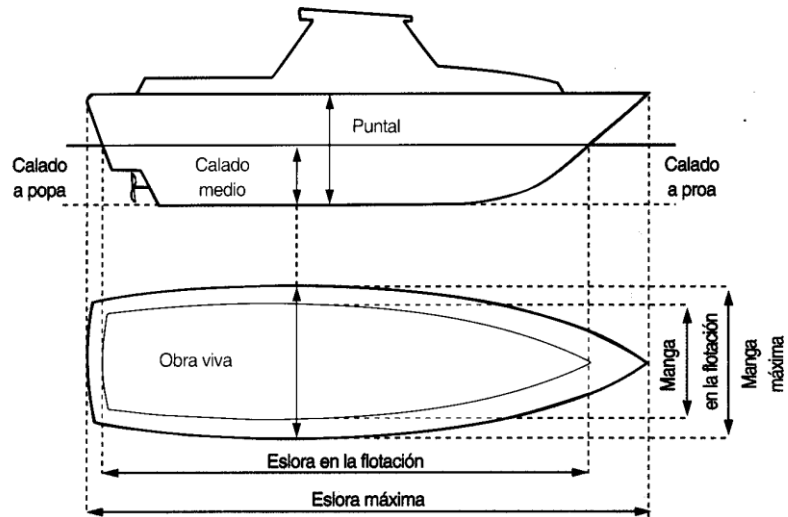


Figura 45. Características geométricas barco [18]

Las RAOs se han digitalizado con AutoCAD, y cargado los datos en Matlab a través de Excel. Los resultados se presentación a continuación.

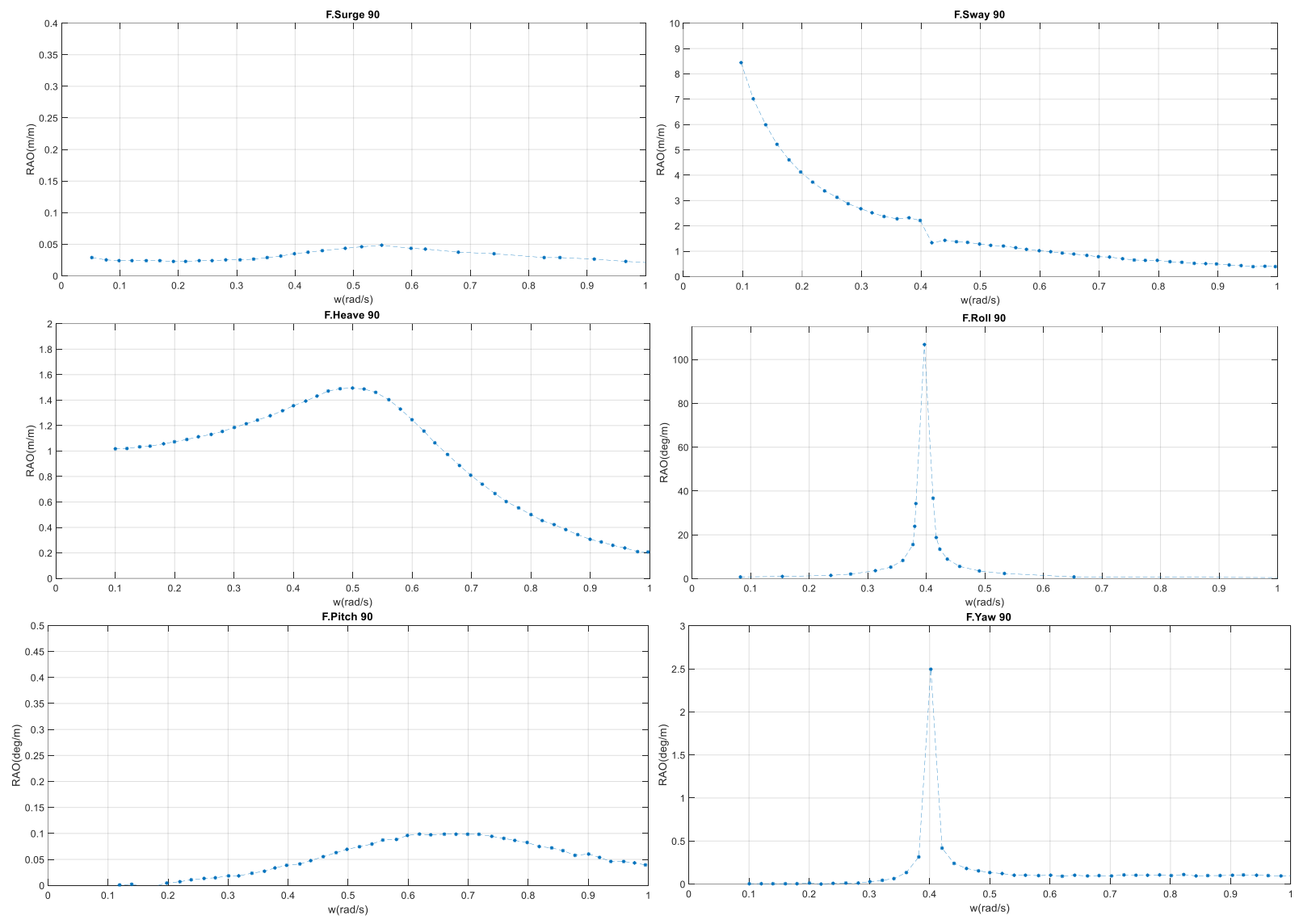


Figura 46. RAOs para barco tipo F

4.2 Límites operativos de la plataforma

Geoméricamente la plataforma no puede compensar cualquier movimiento. Las limitaciones vienen impuestas por la longitud máxima y mínima que pueden tener los actuadores de la plataforma, que para el caso de la plataforma aquí desarrollada es $L_{max}^{actuador} = 4.730 \text{ m}$, y $L_{min}^{actuador} = 2.365 \text{ m}$.

Definir un tamaño de ola para el cual el sistema operará de forma satisfactoria no es trivial, ya que es dependiente de la dirección en la que llegue la ola al barco, la frecuencia de las mismas y la respuesta propia del barco, como se ha demostrado. La nave es excitada en los seis grados de libertad cuando le llega una ola, por tanto, el movimiento de la cubierta, considerándola como un sólido rígido, es una combinación de las rotaciones y traslaciones de la misma. Habrá infinitas combinaciones de traslaciones y rotaciones que hagan que alguno o todos los actuadores de la plataforma alcance su longitud máxima.

En relación a lo expuesto hasta ahora, definir la situación más desfavorable en la que la plataforma sería capaz de proporcionar una respuesta adecuada requiere, de forma general, el conocimiento de:

- 1) El buque en el que se instalará la plataforma, en concreto, la respuesta del mismo a la excitación marina ante distintas situaciones.
- 2) Un estudio completo de todas las situaciones en las que el buque puede navegar, conociendo las condiciones más desfavorables del mar y la respuesta de este.

Atendiendo a los resultados expuestos en la tarea encargada de simular el comportamiento del barco, estos se proporcionan como entrada al sistema en Simulink. Para llevar a cabo esto es necesario ordenar los datos en función del tiempo mediante el comando “*timeseries*” de Matlab (Ver Anexo B>Script de configuración).

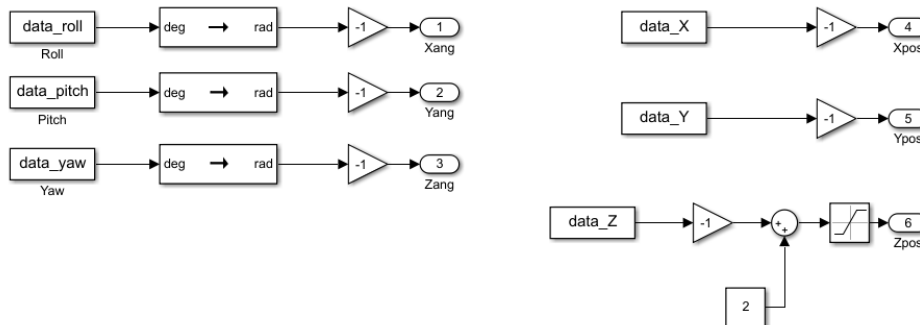


Figura 47. Bloques Simulink para introducir el movimiento marino

Se observa que la plataforma responde de forma adecuada para **olas de hasta 6 m** aproximadamente, midiendo la altura H según lo expuesto en la Figura 48. Este resultado queda justificado en el apartado 1.

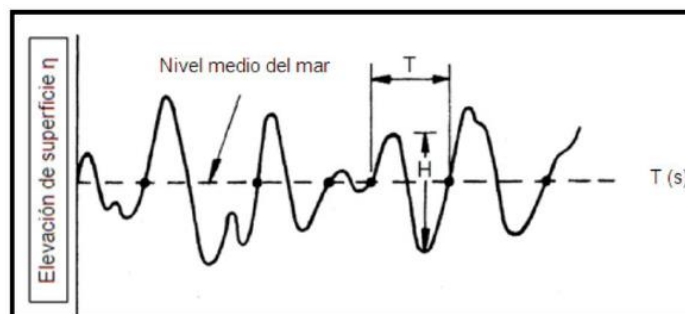


Figura 48. Altura H de la ola [14]

Si se el sistema opera en condiciones que supongan una mayor sollicitación, funcionará, pero habrá movimientos que no sea capaz de compensar, llegando a sus límites físicos y proporcionando un mensaje de advertencia “*Warning*” en la “*Command Window*” de Matlab.

A la vista de las escalas expuesta en el Anexo A, una altura de ola de 6 m se corresponde con **Grado 6** en la

escala Douglas y un número de Beaufort 8, correspondiéndose con un viento en el intervalo [34, 40]kt.

Es necesario valorar otros aspectos relacionados con las características de los actuadores que montará el sistema, en concreto la fuerza máxima y mínima que estos pueden aplicar, ya que esto limitará la respuesta del actuador y con ello la de la plataforma completa. Los actuadores más convenientes para esta aplicación son los electrohidráulicos, ya que pueden proporcionar una respuesta rápida (eléctrico) y de gran potencia (hidráulico). La rapidez con la que se estabiliza la plataforma depende de esta fuerza máxima, pues se observa que cuando esta aumenta la rapidez de la respuesta también lo hace. Hay que tener en cuenta que no se está teniendo en cuenta efectos dinámicos asociados a la masa y el fluido del actuador, si no que se está considerando ideal. Puede probarse que si la **fuerza máxima y mínima es de $5 \times 10^4 \text{ N}$** , el comportamiento del sistema es adecuado (respuesta rápida). La elección de este valor no es casual, ya que se ha llevado a cabo un estudio de las soluciones disponibles en el mercado para este tipo de aplicaciones, concretándose el rango de valores posibles para un actuador de este tipo. Por ejemplo, los actuadores *GSX Series Integrated Motor Actuator* de EXLAR®.

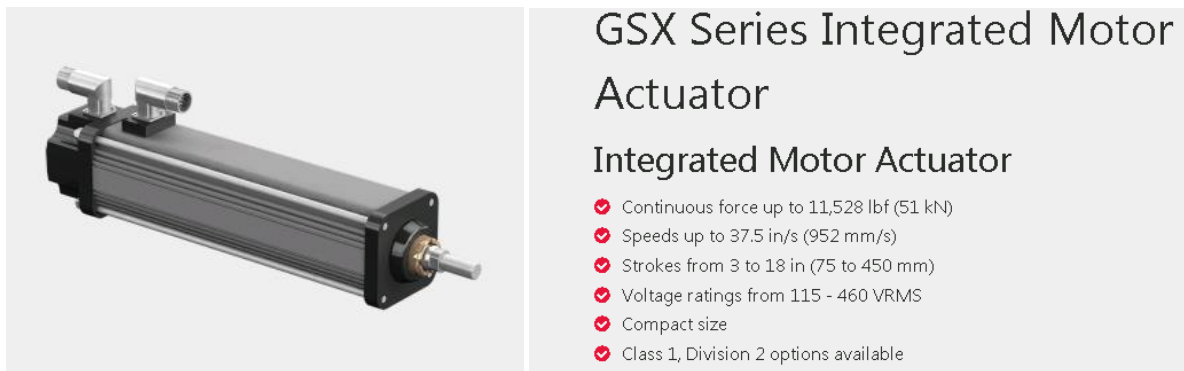


Figura 49. Actuador electrohidráulico lineal y sus especificaciones (EXLAR®) [19]

4.3 Actuaciones del UAV

Es conveniente valorar algunos aspectos relaciones con las aeronaves que operarán en este tipo de misiones. En primer lugar, según especificaciones de la memoria técnica del proyecto, los drones que llevarán a cabo los vuelos son aeronaves de ala fija, ya que, con la tecnología del momento, son las únicas que permiten cubrir el área necesaria.

En general, todas las aeronaves presentan límites operativos en el despegue y aterrizaje respecto a la componente máxima de viento cruzado y de cola que pueden soportar. El viento cruzado y de cola supone un peligro para las operaciones de aterrizaje y despegue, ya que por un lado del de cola hace que la distancia de aterrizaje y despegue sea mayor, pues la velocidad aerodinámica que ve la aeronave es menor, lo que implica que la longitud de pista necesaria para realizar la operación sea mayor (pudiendo ser mayor que la disponible), y el cruzado desvía la aeronave de su dirección de aterrizaje/despegue, exigiendo que el sistema o el piloto tenga que corregir constantemente dicha desviación (los vientos cruzados de gran magnitud pueden hacer que no sea posible el aterrizaje de la aeronave).

Para el tipo de aeronaves que aquí nos concierne, y teniendo en cuenta que al final de la pista está instalada una red, el viento de cola no supone un problema para la operación de aterrizaje.

En cuanto al viento cruzado, la sensibilidad de una aeronave al viento cruzado es inversamente proporcional a:

- La masa
- Velocidad de aterrizaje
- Carga alar:

$$n \approx L/W, \text{ si } \alpha \ll 1 \quad (37)$$

Para tener una primera estimación de la componente de viento cruzado máxima con la que podría operar un UAV, sería necesario conocer la curva de sustentación del perfil que monta dicho UAV, así como la superficie

alar. De esta forma se podría obtener la velocidad de aterrizaje del vehículo y su carga alar. Con esa información, y usando el MTOW, es posible obtener la componente de viento cruzado máxima que permitiría un aterrizaje seguro del RPAS.

La plataforma Stewart dispone de seis grados de libertad, por lo que puede rotarse la misma durante la operación de aterrizaje. La rotación máxima que permite la plataforma es dependiente de la longitud de los actuadores (a veces no se podrá orientar según la dirección del viento). Además, el sistema está enfocado a barcos para el transporte de mercancías, es decir, barcos de gran tamaño, por lo que no es viable que cambie el rumbo para orientar la pista en la dirección del viento, pudiéndose solo orientar la misma según los límites de la plataforma.

Ante la dificultad para encontrar datos sobre la componente de viento cruzado máximo que permite la operación de una aeronave tipo para esta misión, por ejemplo, el RPAS Tucán⁴, para aviones de pequeño tamaño de aviación general, **la componente máxima de viento cruzado que permite el aterrizaje de forma segura es de 15 kt**. No hay que perder de vista que esa componente solo garantiza que el aterrizaje será seguro, pero es posible que el piloto (ya sea de aeronave tripulada o RPAS) sea capaz de aterrizar con mayor componente de viento cruzado.



Figura 50. RPAS Tucan [3]

La operación de despegue para UAV de pequeño tamaño, como es el caso, no será crítica, ya que la aeronave será catapultada en con el viento de cara.

⁴ UAV propiedad de Everis. La aplicación principal de este RPAS son las misiones ISTAR (Inteligencia, Vigilancia, Adquisición de Objetivos y Reconocimiento).

RESULTADOS

Soy de las que piensan que la ciencia tiene una gran belleza. Un científico en su laboratorio no es sólo un técnico: también es un niño colocado ante fenómenos naturales que lo impresionan como un cuento de hadas.

Marie Curie

El modelado y simulación del sistema, como se refirió al principio de este trabajo, se ha llevado a cabo en Matlab & Simulink. El esquema expuesto en la Figura 51 está formado por los siguientes subsistemas:

- 1) **Mechanical components Stewart Platform**
- 2) **IMU+Control Block**
- 3) **Stand by Position+Control Block**

Aunque los subsistemas 1), 2) y 3) han sido explicados en los apartados 2 y 3, es necesario para la comprensión del sistema aclarar algunas características aún no expuestas:

- Para optimizar energéticamente el sistema, la plataforma solo compensará el movimiento marino cuando el UAV vaya a realizar la maniobra de aterrizaje. Para ello se ha implementado, mediante un bloque tipo switch, un modo de espera, el cual se desactiva cuando se produce la conmutación de dicho switch, simulando la señal de activación que recibiría de la aeronave indicándole que va aterrizar.
- Mientras el sistema no está funcionando, la plataforma se mantiene en su posición nominal, con el cdg en (0,0,2). Para simular esto se ha implementado el subsistema “*Stand by*”, el cual proporciona a los controladores dicha referencia⁵.

Para obtener el movimiento del barco, como ya se ha expuesto en el apartado 4, se toman los datos resultantes del trabajo desarrollado en otra tarea, en concreto, en la actividad “*A2.2: Desarrollo de modelos matemáticos de comportamiento de masas inerciales sometidas a excitación marina*”. En este trabajo se configuró un script, “*MovimientoBarco.m*”, que genera el movimiento del barco según los seis grados de libertad definidos, aunque de forma discreta. Mediante el bloque “*From Workspace*” (reconstruye la señal mediante interpolación polinómica) y el comando “*timeseries*” se introducen en la simulación. Esos datos simulan las señales de una IMU ya integradas instalada en la base de la plataforma, y como el movimiento de esta debe ser el contrario basta con multiplicar dicha señal por una ganancia negativa de valor -1. Ver Figura 47.

⁵ En un sistema real, dicha referencia no sería necesaria ya que el sistema se mantendría, sin señal de control, en la posición nominal debido a su configuración geométrica.

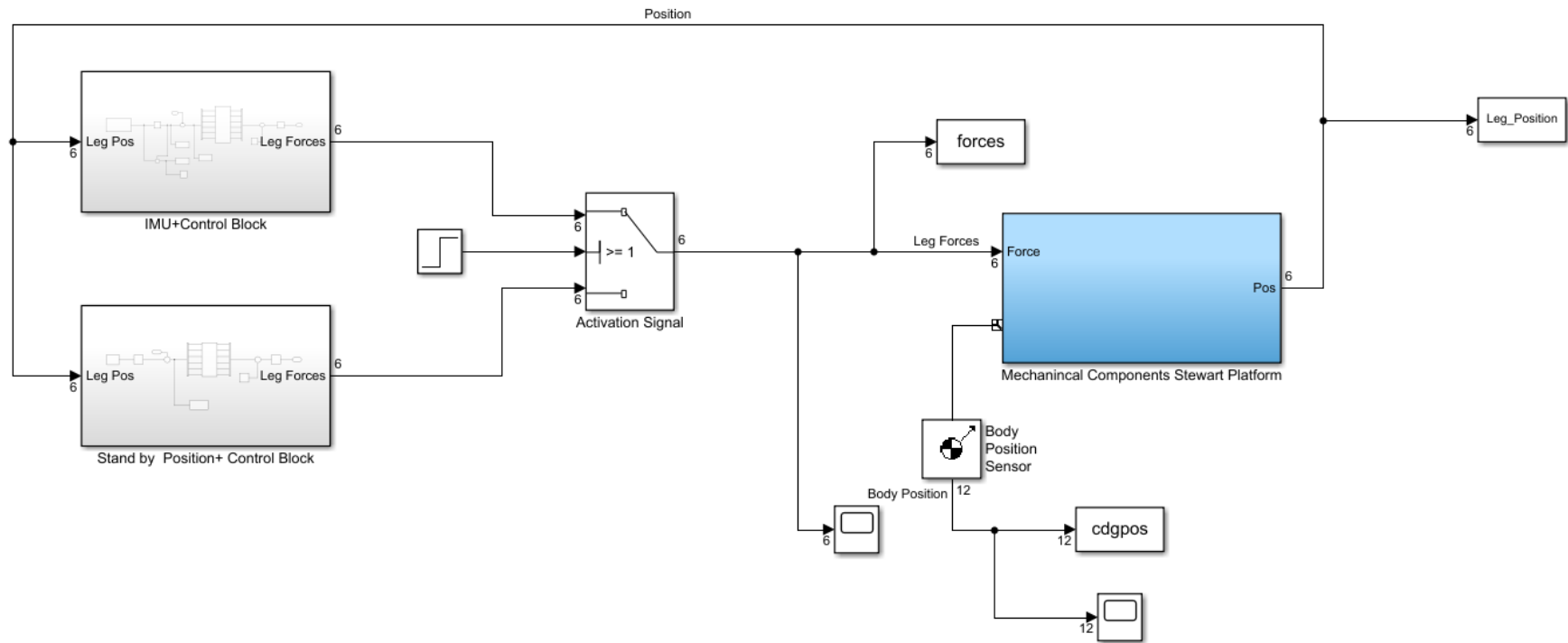
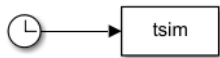


Figura 51. Modelo Simulink del sistema completo

Para llevar a cabo las simulaciones, se propone seguir el diagrama expuesto en la Figura 52. Respecto a dicho esquema:

- Si durante la simulación del movimiento de la plataforma aparece un mensaje “Warning” por la “*Command Window*” de Matlab, significa que algún actuador alcanza la **longitud máxima/mínima** posible.
- Para cambiar el tamaño de ola hay que eliminar el comentario de una de las líneas $\in [103,108]$ del Script de configuración del Anexo B. Esos archivos contienen la simulación del movimiento del barco para distintas situaciones.
- La pregunta “¿Variables cargadas?” se refiere a que es posible que al ejecutar la simulación en Simulink aparezca un error indicando que hay alguna variable que no existe. Normalmente, estas variables son, o bien los datos del movimiento del barco, o las ganancias del PID-cargadas al ejecutarse el archivo “*StewartPlatformSetup.m*”.

Todas las simulaciones tienen una duración de un minuto, aunque pueden realizarse simulaciones de mayor duración generando nuevos archivos del movimiento del barco.

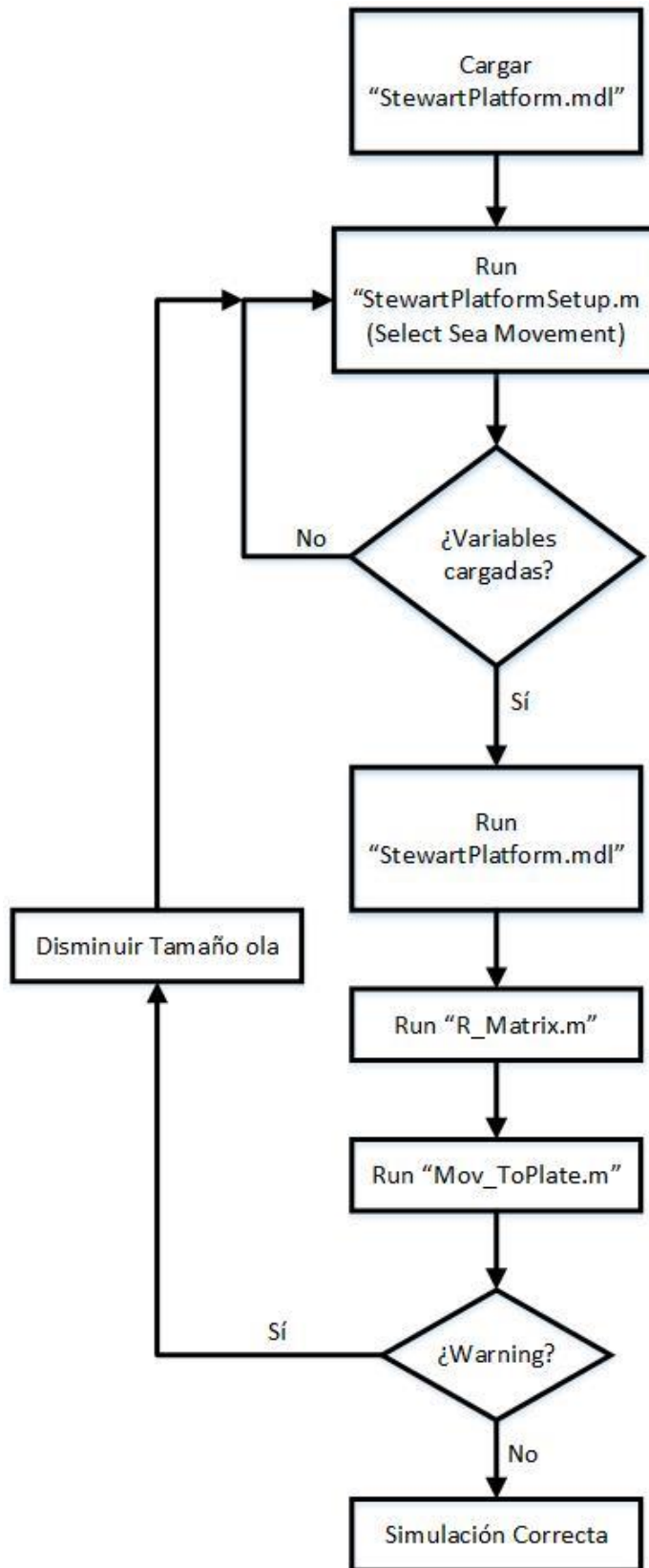


Figura 52. Diagrama para la simulación del movimiento del sistema

5.1 Simulación 1

La excitación marina a la que se ve sometida el barco se muestra en la Figura 53. Se observa como la altura máxima de ola es de 0.8m.

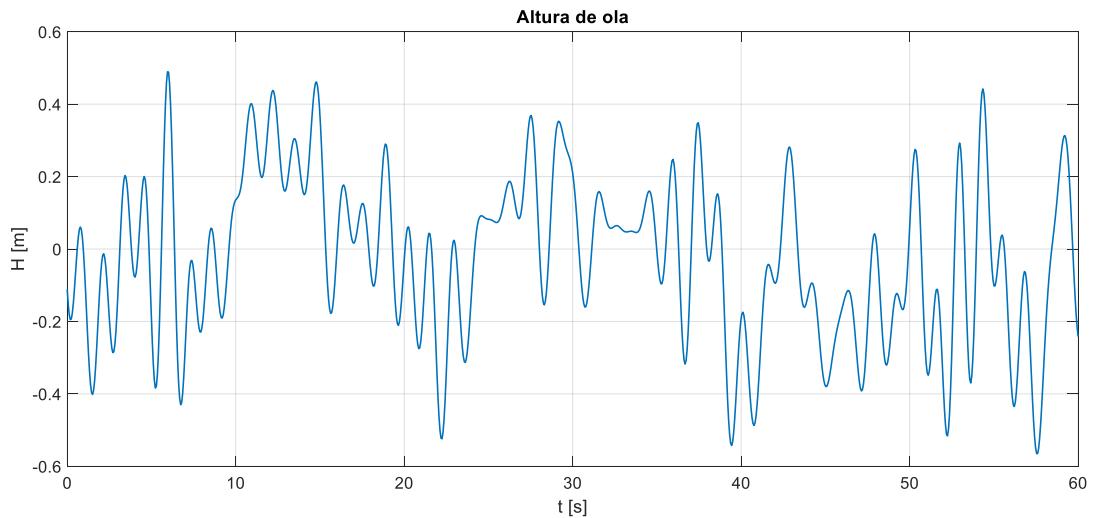


Figura 53. Movimiento marino. Simulación 1

La evolución del error de los actuadores, las fuerza de estos y el error debido a las limitaciones físicas de la plataforma se presenta en la Figura 54. Se percibe:

- $t \in [0, 5]$ s: El sistema no está compensando el movimiento de la cubierta, por lo que el error de seguimiento de los actuadores es grande y la fuerza es la necesaria para mantener la plataforma en posición horizontal.
- $t > 5$ s: El error de seguimiento es muy pequeño. No existe error por limitaciones físicas de la plataforma ya que esta es capaz de compensar dichos movimientos.

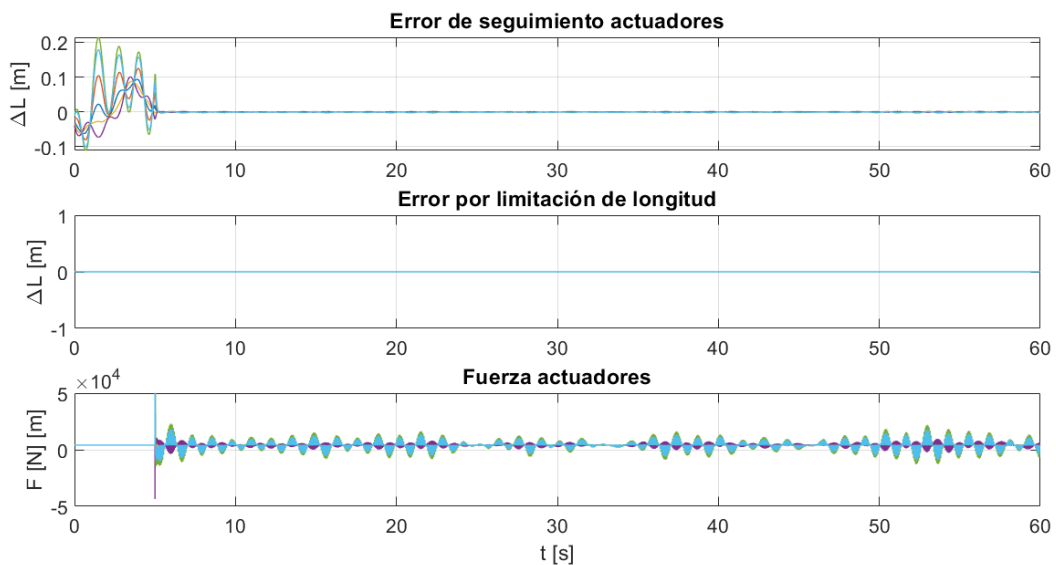


Figura 54. Errores y fuerzas. Simulación 1

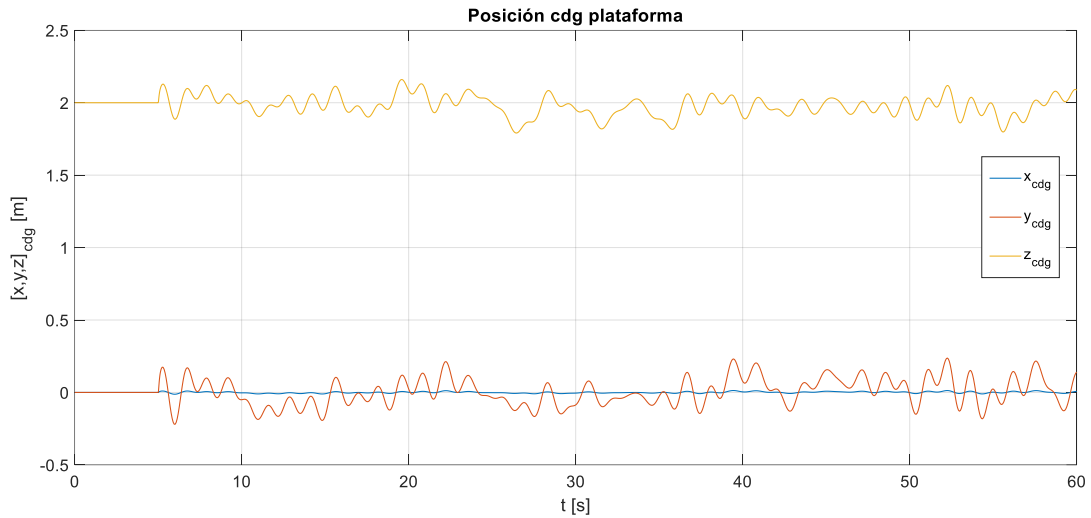


Figura 55. $(x, y, z)_{cdg}$ plataforma. Simulación 1

5.2 Simulación 2

Se empieza a aumentar la altura de ola (en este caso la máxima es de 4.7 m).

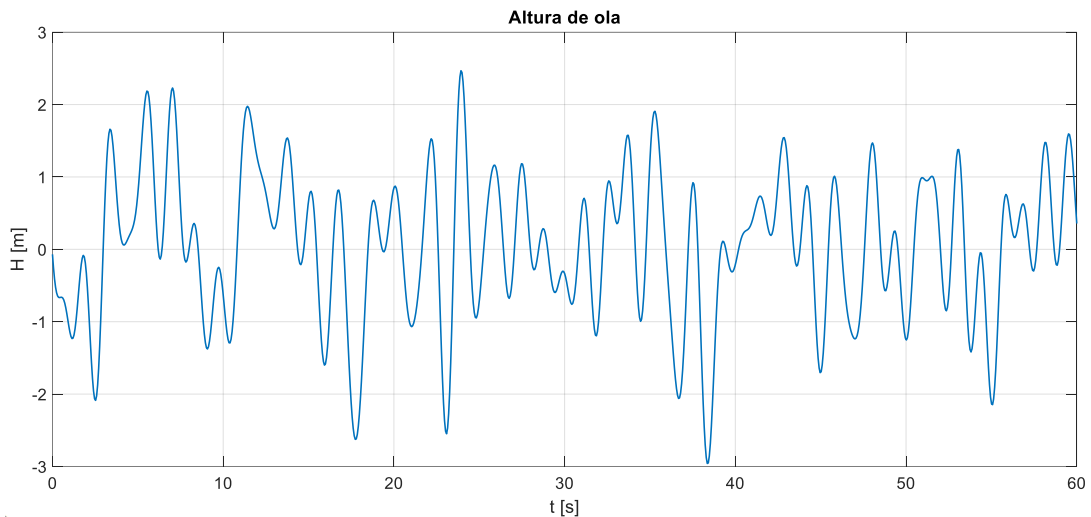


Figura 56. Movimiento marino. Simulación 2

De nuevo, durante los 5 primeros segundos de simulación el estado de la plataforma es “espera” y no está compensando el movimiento del barco, por lo que el error de los actuadores crece. A partir de dicho instante comienza a funcionar. Se observa que hay ciertos movimientos que hacen que físicamente la plataforma no sea capaz de compensarlo (esto explica los picos del gráfico “Error por limitación de longitud”) en la Figura 57.

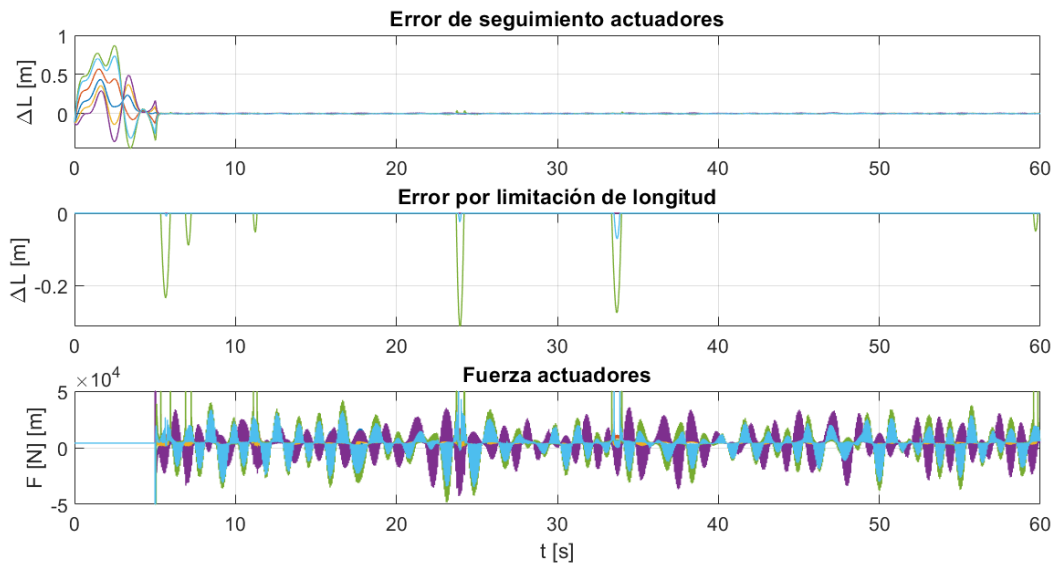


Figura 57. Errores y fuerzas. Simulación 2

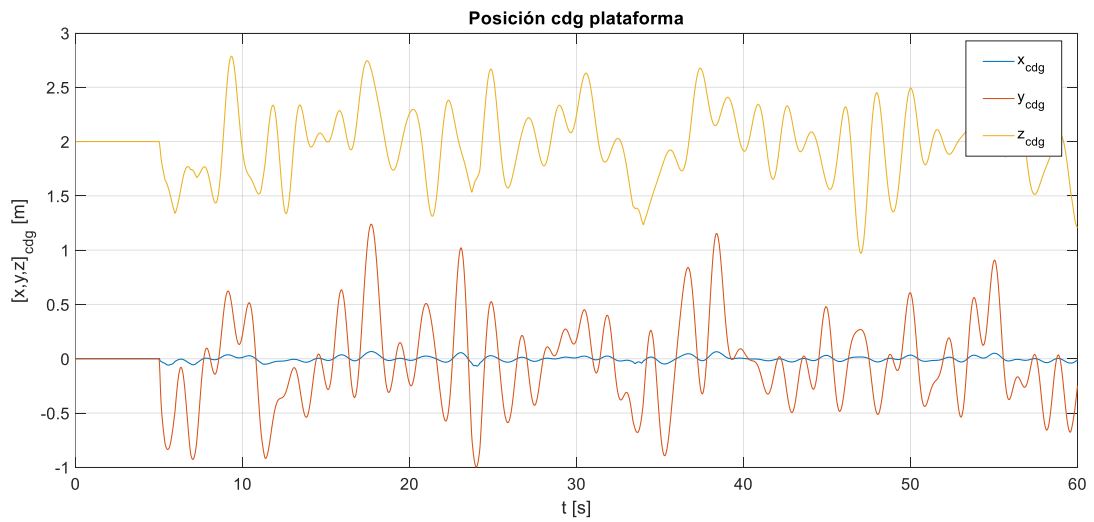


Figura 58. $(x, y, z)_{cdg}$ plataforma. Simulación 2

5.3 Simulación 3

La altura máxima de ola para esta simulación es de 5.7 m.

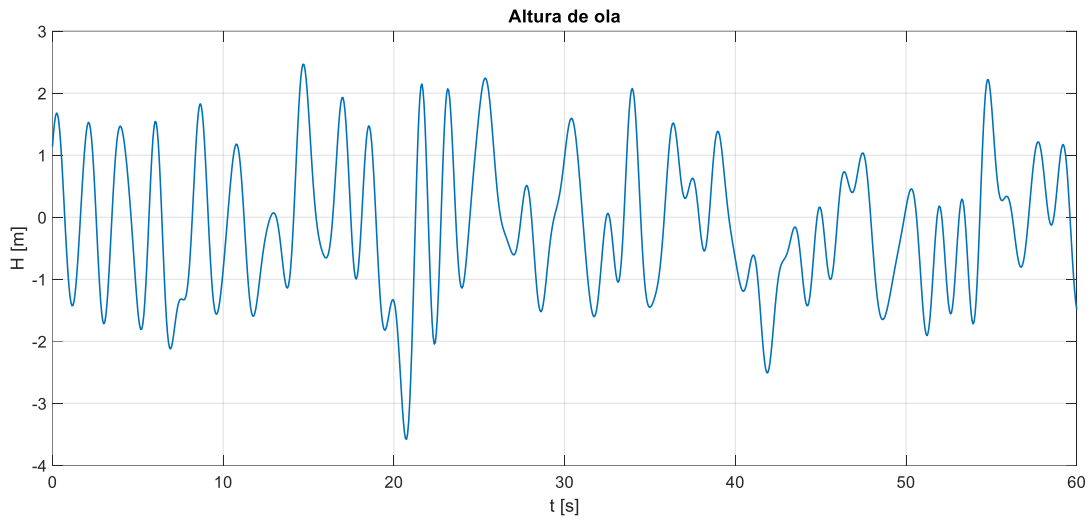


Figura 59. Movimiento marino. Simulación 3

En la Figura 60 se observa observa como empiezan a aumentar las situaciones de movimiento en las que la plataforma no puede compensar el movimiento del barco por la limitación en la longitud de los actuadores que la constituyen. Además, mientras que en la simulación anterior dicho error era del orden de 30 cm como máximo, en esta el error se máximo se duplica, llegando a valores de 60 cm. De nuevo, se ha impuesto un tiempo de espera de 5 segundos durante los cuales la plataforma no compensa el movimiento marino.

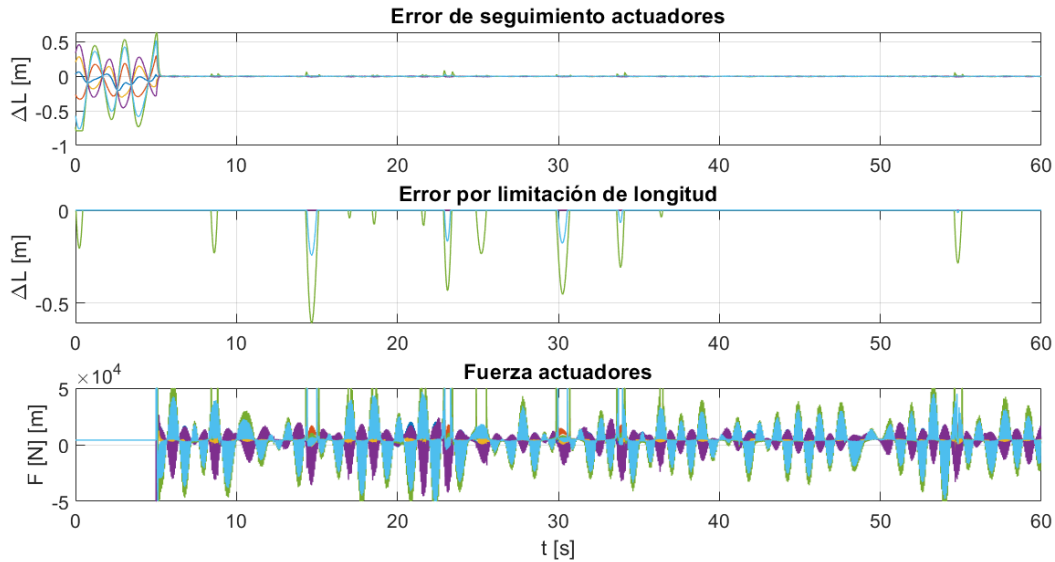


Figura 60. Errores y fuerzas. Simulación 3

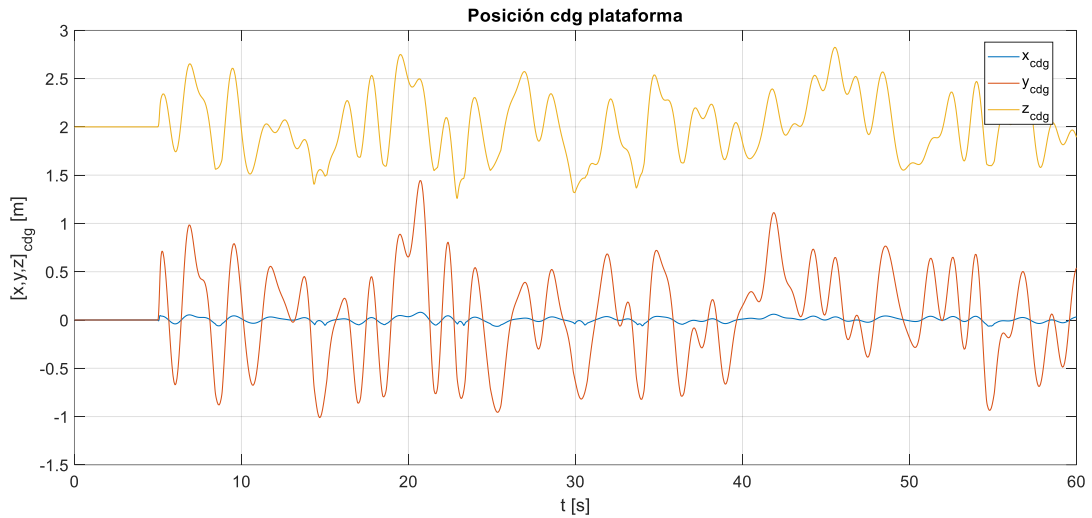


Figura 61. $(x, y, z)_{cdg}$ plataforma. Simulación 3

5.4 Simulación 4

Esta simulación tiene como objetivo demostrar que el sistema no funcionará de forma correcta para una altura de ola superior a 6 m (la altura máxima de ola es de 8 m).

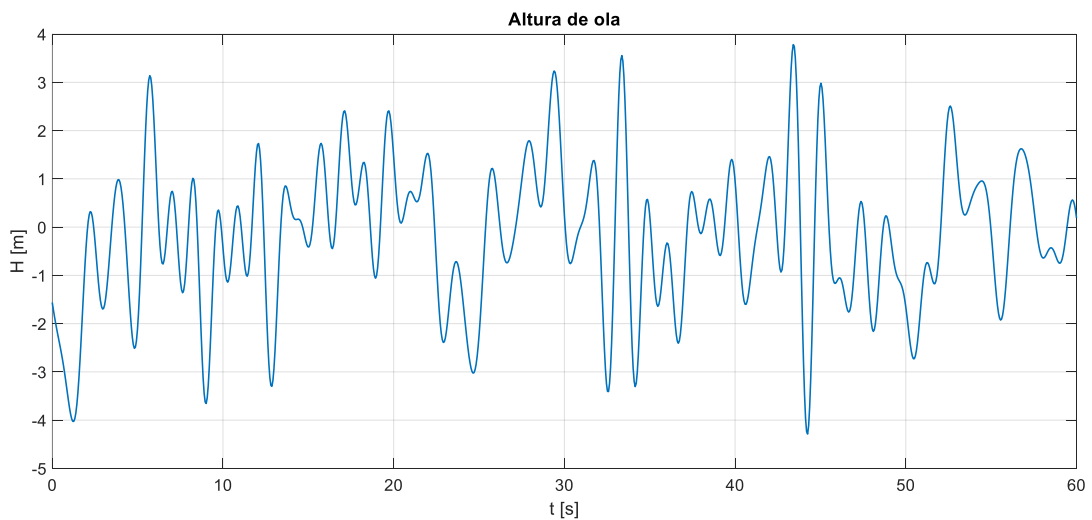


Figura 62. Movimiento marino. Simulación 4

Se manifiesta que los actuadores están saturados durante la mayor parte del tiempo de simulación, los errores de seguimiento son $\sim 1m$, y los errores por limitación de longitud siguen creciendo. Este comportamiento justifica que la plataforma no sea capaz de funcionar de forma correcta para una altura de ola mayor de 6 m (para olas incidiendo a 90° respecto a proa). Respecto al movimiento del cdg, en la posición del mismo se perciben grandes oscilaciones respecto a su punto nominal.

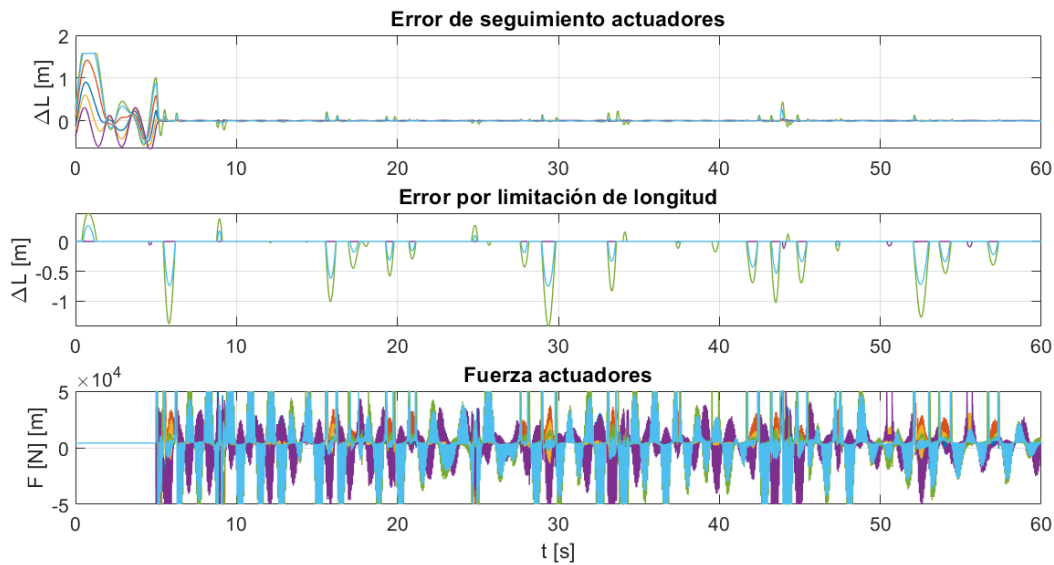


Figura 63. Errores y fuerzas. Simulación 4

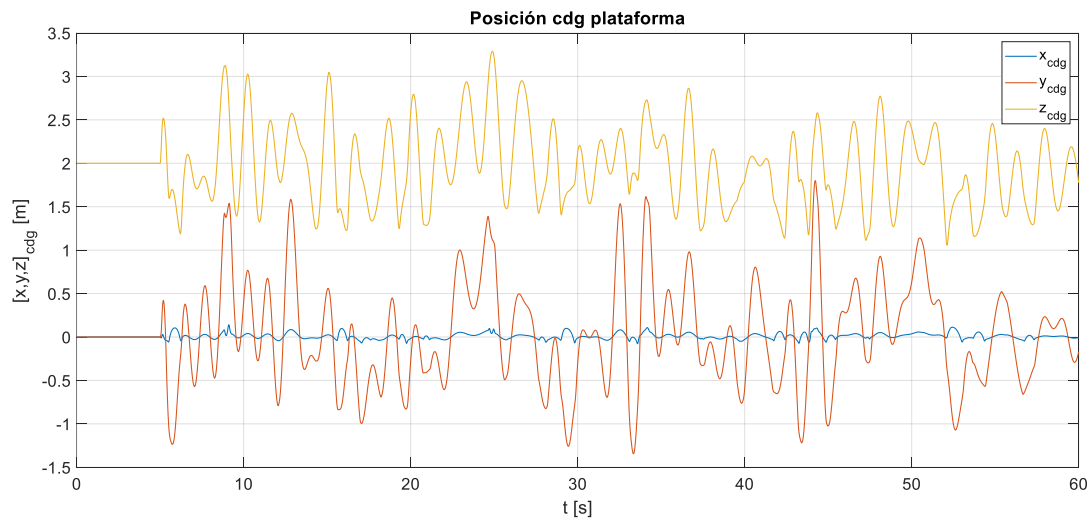


Figura 64. $(x, y, z)_{cdg}$ plataforma. Simulación 4

5.5 Análisis de resultados

Se observa que el modelo de movimiento marino presenta periodos de olas muy pequeños, lo cual es inusual para olas de tamaño comprendido entre 4 y 6 m. Según bibliografía consultada [6], los periodos típicos de las olas debidas al viento en mar completamente desarrollado se exponen en la Figura 65, y aunque no se esté simulando ese estado de mar (completamente desarrollado), porque como se ha indicado nunca llega a producirse, el periodo de las olas debe ser del orden de esos valores. Por ejemplo, en la Figura 53 se observa que la altura de ola es ~ 1 m, y estas presentan un periodo ~ 1.5 s, sin embargo atendiendo a la Figura 65, para olas de altura ~ 1 m, el periodo es de 5s aproximadamente. Otro ejemplo sería si atendemos a la Figura 66, en la que para una altura máxima de olas de 4.7 m, el periodo de ola es ~ 2.5 s, y si observamos lo expuesto en la Figura 65, el periodo debería ser ~ 8 s. Por tanto, se puede concluir de esos resultados que el archivo que genera el movimiento marino no lo hace de forma correcta.

Por otro lado, el sistema es capaz de compensar el movimiento a pesar de las variaciones bruscas. Los indicadores de esto son las gráficas tituladas como “Error de seguimiento de actuadores”, en las cuales se compara la longitud necesaria del actuador en cada instante de tiempo con la que logra, apreciándose que el error se mantiene en torno a 0, excepto en la última simulación por las cuestiones ya comentadas. Las gráficas tituladas como “Error por limitación de longitud” indican los movimientos para los cuales la longitud de los actuadores

no es suficiente, es decir, comparan la posición deseada con la máxima/mínima que podría tener el actuador; si esta está fuera de rango, aparece error. Según los resultados expuestos, se puede asegurar que para periodos normales de olas el sistema funcionará mejor, ya que la respuesta de los controladores no necesitará ser tan agresiva.

Conditions necessary for a fully developed sea at given wind speeds, and the parameters of the resulting waves					
Wind conditions			Wave size		
Wind speed in one direction	Fetch	Wind duration	Average height	Average wavelength	Average period and speed
19 km/h (12 mph)	19 km (12 mi)	2 hr	0.27 m (0.89 ft)	8.5 m (28 ft)	3.0 sec, 10.2 km/h (9.3 ft/sec)
37 km/h (23 mph)	139 km (86 mi)	10 hr	1.5 m (4.9 ft)	33.8 m (111 ft)	5.7 sec, 21.4 km/h (19.5 ft/sec)
56 km/h (35 mph)	518 km (322 mi)	23 hr	4.1 m (13 ft)	76.5 m (251 ft)	8.6 sec, 32.0 km/h (29.2 ft/sec)
74 km/h (46 mph)	1,313 km (816 mi)	42 hr	8.5 m (28 ft)	136 m (446 ft)	11.4 sec, 42.9 km/h (39.1 ft/sec)
92 km/h (57 mph)	2,627 km (1,632 mi)	69 hr	14.8 m (49 ft)	212.2 m (696 ft)	14.3 sec, 53.4 km/h (48.7 ft/sec)

Figura 65. Periodos típicos de olas según velocidad del viento y altura de ola [6]

Para verificar que el sistema funciona ante sollicitaciones más suaves, o ante “frentes de olas distintos de 90°”, se han realizado distintas simulaciones expuestas a continuación. Se asocia la siguiente notación a los ángulos de balanceo, cabeceo y guiñada:

- Balanceo (roll): ϕ
- Cabeceo (pitch): θ
- Guiñada (yaw): Ψ

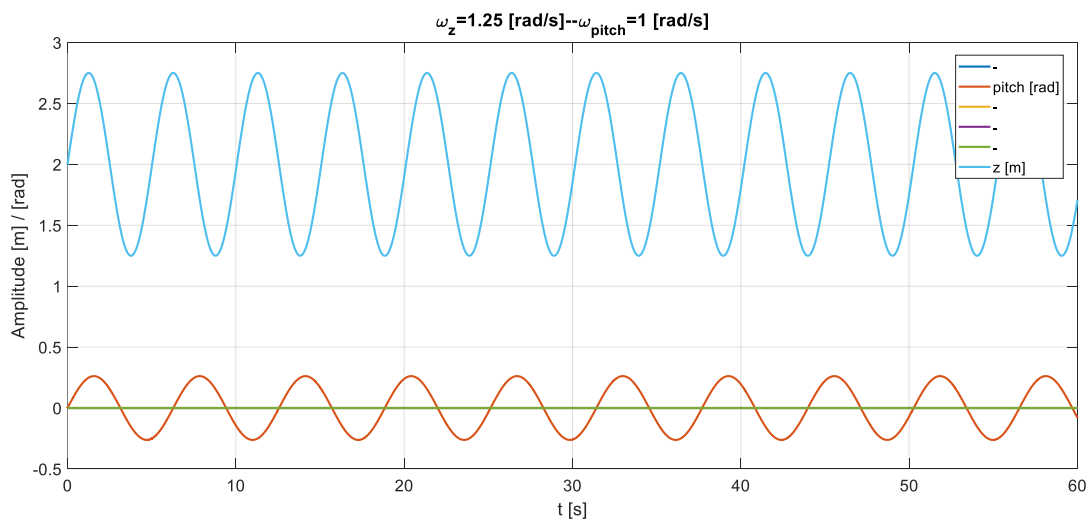


Figura 66. Excitación sinusoidal en grados de libertad z y cabeceo

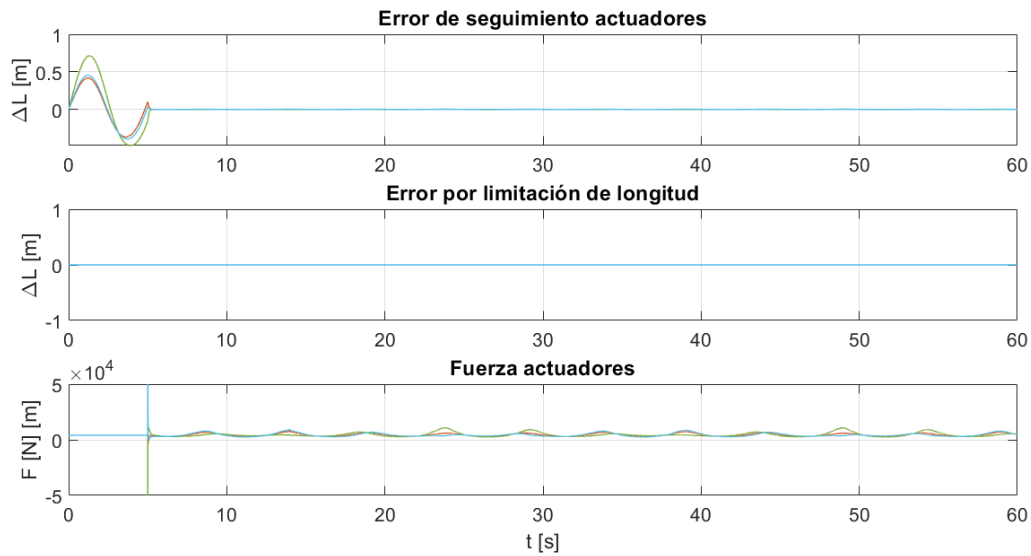


Figura 67. Errores y fuerzas ante excitación sinusoidal

En las simulaciones presentadas en la Figura 66 y Figura 67, se ha sometido la plataforma a una excitación sinusoidal en los grados de libertad z y θ , con una amplitud de 0,75m y 15 deg ,y una frecuencia de 1,25 rad/s y 1 rad/s respectivamente. Se percibe como el sistema responde de forma favorable, compensando el movimiento y haciendo que el error sea nulo. El error inicial se debe a que el sistema no funciona hasta que pasan los cinco primeros segundos.

Si ahora se somete a un movimiento sinusoidal aleatorio, puede comprobarse que el sistema logra compensarlo. Esta vez, el sistema no ha comenzado a funcionar hasta pasados 8 segundos de simulación.

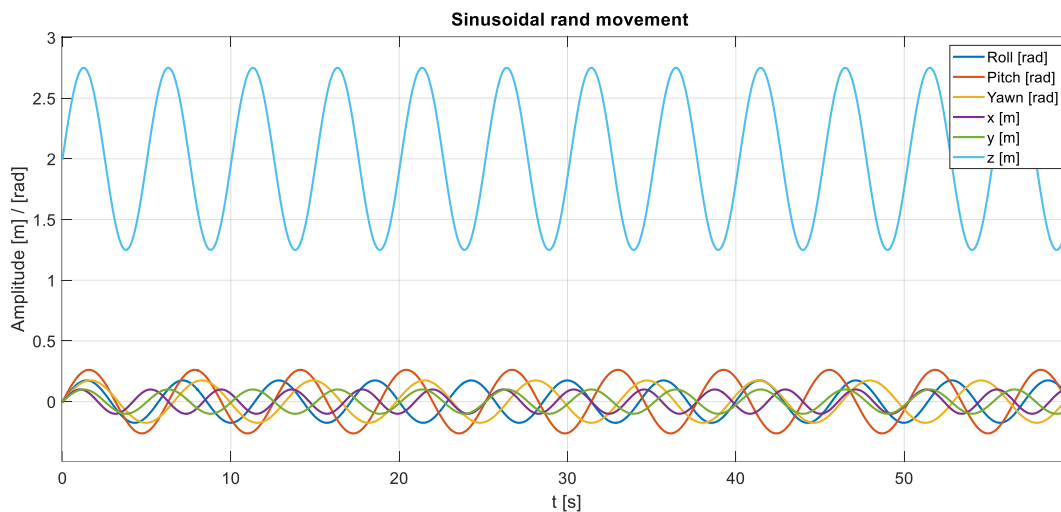


Figura 68. Excitación aleatoria sinusoidal

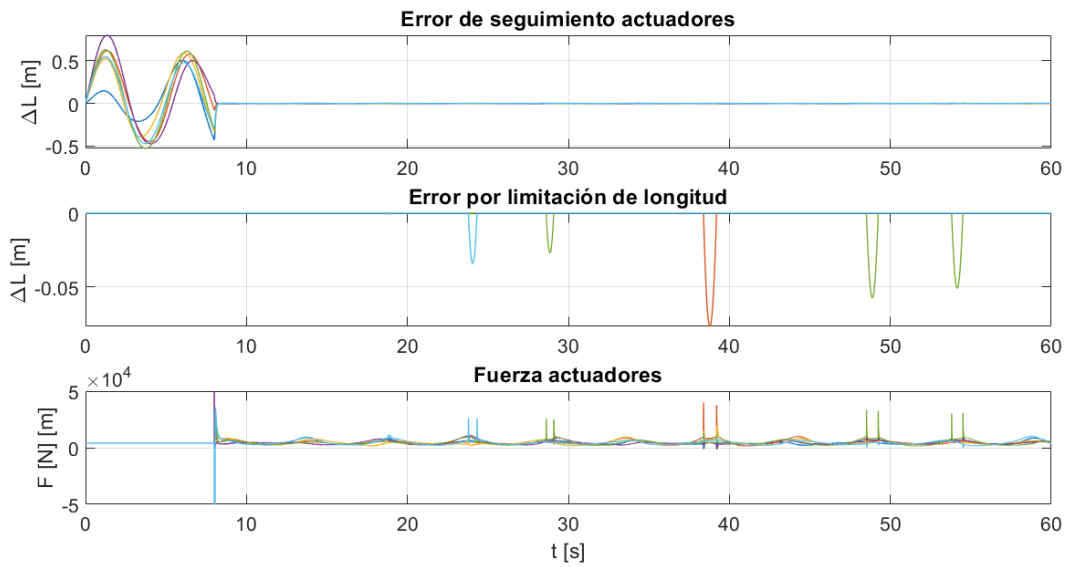


Figura 69. Errores y fuerzas ante excitación sinusoidal aleatoria

5.6 Simulaciones tridimensionales

Todas las simulaciones expuestas en los apartados anteriores pueden visualizarse tridimensionalmente ejecutando los pasos expuestos en la Figura 52. Algunas de estas han sido grabadas en formato mp4 con la herramienta “*Game DVR*” de Windows 10. Están incluídas en la carpeta “Simulaciones” que acompaña este documento.

6 CONCLUSIONES Y TRABAJOS FUTUROS

La vista debe aprender de la razón.

Johannes Kepler

El trabajo desarrollado ha dado respuesta a la necesidad, dentro del proyecto SADSIM, al diseño de una plataforma de estabilización inercial para aterrizaje de UAV. Esta necesidad está justificada por dos razones:

- 1) El UAV llevará a cabo el aterrizaje de forma autónoma usando un código QR dispuesto al final de la plataforma. Para la detección de dicho código incorpora un sensor de visión artificial, y una vez detectado el mismo este no puede deformarse o moverse ya que produciría la pérdida de la marca. Los movimientos que se producen en la cubierta de un barco puede causar dicha deformación, por lo que es necesario disponer de un sistema que contrarreste ese movimiento, y mantenga el código-la plataforma- en su posición original, esto es, horizontal.
- 2) El UAV aterrizará sobre dicha plataforma, por lo que necesita que la misma se mantenga en posición horizontal, ya que los movimientos de la pista de aterrizaje podrían provocar que la aeronave sufriera un accidente.

Tras un estudio de la viabilidad tecnológica del proyecto, se llega a la conclusión de que ya existe un sistema que da respuesta a la problemática planteada, la **Plataforma Stewart**. Las aplicaciones de dicho sistema están orientadas a simuladores de vuelo para entrenamiento de pilotos, y recientemente simuladores de automoción, aunque es fácilmente adaptable a nuestras necesidades.

Como requisito del proyecto, el modelado y simulación del sistema se ha llevado a cabo en Matlab & Simulink. El modelado del sistema mecánico puede encontrarse en la web MathWorks, siendo configurable dicho sistema según la necesidad del proyectista. Para el caso aquí de interés, se mantienen los parámetros originales por la falta de especificaciones en el proyecto.

El modelo la plataforma consta de tres módulos:

- 1) Módulo mecánico.
- 2) Módulo de control.
- 3) Módulo de cinemática.

El proyecto se ha centrado en los módulos 2) y 3). Respecto al módulo de control se ha optado por el uso de controladores PID. Las razones que justifican esta decisión son:

- 1) La experiencia en el uso de controladores PID, los cuales demuestran un buen funcionamiento en múltiples campos. Además, según bibliografía consultada la mayor parte de los diseños usan estos controladores.
- 2) Aunque es posible acudir al diseño de otros tipos de controladores, el controlador PID es un controlador simple de manera que, si resuelve el problema de control, no es necesario acudir a soluciones más complejas. Además, el PID presenta buenas propiedades de robustez.

Para diseñar el controlador se opta por un diseño frecuencial, para el cual se usa la herramienta Control System Designer de Matlab. Previo al diseño del controlador, al usar técnicas de diseño lineales, es necesario linealizar el sistema. Matlab dispone del comando “*linmod*” para linealizar modelos físicos implementados en Simulink en torno a un punto, aunque este comando puede introducir polos y ceros ficticios en el modelo linealizado-

como ha ocurrido en este trabajo-, indicando que un sistema es inestable y realmente no lo es. El punto seleccionado para la linealización ha sido el $(0,0,2)_{xyz}$. Para verificar la estabilidad del sistema, se han realizado simulaciones de larga duración, demostrando que es capaz de mantener la posición indicada tras una entrada escalón. Respecto al diseño frecuencial, se han logrado calcular unas ganancias del controlador de forma que la **sobreoscilación** ante escalón es **nula**, y el **tiempo de establecimiento** sea de **0,4s** aproximadamente. Los objetivos del controlador son conseguir una respuesta lo más rápida posible y una sobreoscilación lo más pequeña posible, por lo que el diseño logra dichos objetivos.

Respecto al módulo de cinemática, está constituido por dos bloques:

- 1) Generador de trayectoria.
- 2) Cinemática inversa

El generador de trayectoria, es para este caso la IMU. Este sistema se ha modelado tomando como entrada el movimiento del barco del trabajo anteriormente desarrollado en el proyecto. Esta IMU irá instalada en la base de la plataforma, midiendo aceleraciones angulares y lineales. La integración de estas medidas, proporcionan las rotaciones y traslaciones que sufre la cubierta del buque, por lo que basta con multiplicar por una ganancia unitaria negativa para que el bloque de cinemática inversa calcule la longitud que debe tener cada actuador, y así compensar el movimiento.

El bloque de cinemática inversa calcula el cambio de longitud que debe tener cada actuador, respecto a su posición nominal, para que la plataforma se posicione.

Además, se han desarrollado las ecuaciones de la cinemática directa para el caso de que se tenga como dato la posición del centro de gravedad de la plataforma y la longitud de cada actuador en cada instante de simulación. Esta información puede conseguirse usando el bloque “*Body Sensor*” de Simulink, el cual también puede proporcionar la matriz de rotación. Con esta información se ha desarrollado un archivo que simula de forma tridimensional el movimiento de la plataforma.

Por último, se han establecido los límites operativos del sistema atendiendo al tamaño de ola máximo que podrá compensar. De este estudio se concluye:

- 1) El movimiento marino que sirve como entrada al sistema no es realista, ya que el periodo de olas que presenta es mucho mayor que los periodos típicos reales.
- 2) El sistema de control es capaz de responder a dicha excitación, por lo que para condiciones marinas menos exigentes puede asegurarse que funcionará correctamente.

Respecto al desarrollo de trabajos futuros, se proponen las siguientes líneas de investigación:

- Modelar mediante Catia una plataforma tipo Stewart y exportar el modelo al entorno Simulink. Según bibliografía consultada durante el desarrollo del proyecto, es posible desarrollar modelos CAD y exportarlos a Simulink.
- Obtención de las ecuaciones que modelan la dinámica y cinemática de la plataforma, para su linealización en torno a un punto. Esto podría usarse para comparar la linealización que realiza el comando “*linmod*” con la real, y comparar los resultados.
- Desarrollar estrategias de control más avanzadas para controlar el sistema. La incorporación de modelos de control predictivo podría ofrecer buenos resultados, además de permitir comparar esos resultados con los ofrecidos por controladores PID. También sería útil implementar un sistema de control robusto, y de nuevo, comparar los resultados con otros sistemas.
- Desarrollo e implementación de la plataforma física. Se propone el desarrollo, a escala, del sistema haciendo uso de técnicas de producción innovadoras (impresión 3D) y la implementación del sistema de control desarrollado, así como el testeo en condiciones reales de funcionamiento.
- Estudio de mercado para nuevas aplicaciones de la plataforma.

Anexo A

Escala Douglas

La Escala Douglas clasifica los estados de la mar en 10 grados usando como referencia el tamaño de las olas. Tiene dos códigos, uno para estimar el estado del mar y otro para describir la altura de las olas. Está aceptada internacionalmente.

La escala puede observarse en la Tabla 6.

Grado	Altura de las olas (m)	Descripción	Estado del mar
0	Sin olas	Mar en calma	La superficie del mar está lisa como un espejo.
1	0- 0.10	Mar rizada	El mar comienza a rizarse por partes.
2	0.10-0.5	Marejadilla	Se forman olas cortas pero bien marcadas. Empiezan a romper las crestas formando una espuma de aspecto vítreo.
3	0.5-1.25	Marejada	Se forman olas largas con cresta de espuma blanca bien caracterizadas. El mar de viento está bien definido y se distingue fácilmente del mar de fondo que pudiese existir.
4	1.25-2.5	Fuerte marejada	Se forman olas más largas, con crestas de espuma por todas partes.
5	2.5-4	Gruesa	Comienzan a formarse olas altas; las zonas de espuma blanca cubren una gran superficie.
6	4-6	Muy gruesa	El mar se alborota. La espuma blanca que se forma al romper las crestas comienza a disponerse en bandas en la dirección del viento.
7	6-9	Arbolada	Aumentan notablemente la altura y la longitud de las olas y de sus crestas. La espuma se dispone en bandas estrechas en la dirección del viento.
8	9-14	Montañosa	Se ven olas altas con largas crestas que caen como cascadas. Las grandes superficies cubiertas de espuma se disponen rápidamente en bandas blancas en la dirección del viento y el mar alrededor de ellas adquiere un aspecto blanquecino.
9	Más de 14	Enorme	Las olas se hacen tan altas que a veces los barcos desaparecen de la vista. El mar está cubierto de espuma blanca dispuesta en bandas en la dirección del viento. El aire está lleno de salpicaduras, por lo que la visibilidad de los objetos es imposible.

Tabla 6. Escala Douglas

El valor del parámetro Grado (D) depende de la raíz cuadrada de la altura media de ola, $D \sim \beta + \lambda \sqrt{H}$. Usando regresión lineal, se pueden calcular los coeficientes para los valores de altura pequeños ($\lambda_L = 2.3236$, $\beta_L =$

1.2551) y para los valores de altura elevados ($\lambda_H = 2.0872, \beta_H = 0.6091$). El Grado puede ser aproximado como la media entre la estimación baja y alta.

$$D \approx \frac{1}{2} [(\beta_L + \beta_H) + (\lambda_L \sqrt{H_L} + \lambda_H \sqrt{H_H})]$$

Longitud de ola (onda)	Ola corta (m)	Ola media (m)	Ola larga (m)
	$\lambda < 100$	$\lambda [100, 200]$	$\lambda \geq 201$

Tabla 7. Clasificación de la longitud de ola

Altura de la ola	Ola baja (m)	Ola moderada (m)	Ola alta (m)
	$H < 2$	$H [2,4]$	$H \geq 4.01$

Tabla 8. Clasificación de la altura de ola

Escala Beaufort

La escala de Beaufort es una medida empírica de la intensidad del viento basada principalmente en el estado del mar, de sus olas y la fuerza del viento.

La escala se muestra en la tabla expuesta a continuación. Ver Tabla 9.

NúmeroBeafort	Velocidad viento (kt – m/s)	Denominación	Efecto del viento en alta mar
0	[0, 0.2] – [<1]	Calma	Mar como un espejo.
1	[1, 3] – [0.3, 1.5]	Ventolina	Rizos como escama de pescado pero sin espuma.
2	[4, 6] – [1.6, 3.3]	Flojito (Brisa muy débil)	Pequeñas olas, crestas de apariencia vítrea, sin romperse.
3	[7, 10] – [3.4, 5.4]	Flojo (Brisa ligera)	Pequeñas olas, crestas rompientes, espuma de aspecto vítreo, asilados vellones de espuma.
4	[11, 16] – [5.5, 7.9]	Bonancible (Brisa moderada)	Pequeñas olas creciendo, cabrilleo numeroso y frecuente de las olas.
5	[17, 21] – [8, 10.7]	Fresquito (Brisa seca)	Olas medianas alargadas, cabrilleo (con salpicaduras).
6	[22, 27] – [10.8, 13.8]	Fresco	Se forman olas grandes, crestas de espuma blanca (salpicaduras frecuentes).
7	[28, 33] – [13.9, 17.1]	Frescachón	El mar crece. La espuma blanca que proviene de las olas es arrastrada por el viento.
8	[34, 40] – [17.2, 20.7]	Temporal	Olas de altura media y más alargadas. Del borde superior de sus crestas comienzan a destacarse torbellinos de salpicadura.
9	[41, 47] – [20.8, 24.4]	Temporal fuerte	Grandes olas, espesas estelas de espuma a lo largo del viento, las crestas de las olas se rompen en rollos, las salpicaduras pueden reducir la visibilidad.
10	[48, 55] – [24.5, 28.4]	Temporal duro	Olas muy grandes con largas crestas en penachos, la espuma se aglomera en grandes bancos y es llevada por el viento en espesas estelas blancas. En conjunto, la superficie está blanca, la visibilidad es reducida.
11	[56, 63] – [28.5, 32.6]	Temporal muy Duro	Olas de altura excepcional (pueden perderse de vista tras ellas barcos de tonelaje pequeño y medio). Mar cubierta de espuma, la visibilidad es reducida.
12	>64 - >32.7	Temporal huracanado	Aire lleno de espuma, salpicaduras, mar cubierto de espuma. Visibilidad muy reducida.

Tabla 9. Escala Douglas

Anexo B

Script Cinemática directa

```
1. clc;close all;
2. pos_top=pos_top(1:6,:);
3. pos_base=pos_base(1:6,:);
4.
5. %Calculo de la longitud de las patas
6. %-----1-----
   ---%
7. b=pos_base;
8. p=pos_top-[zeros(6,2) 2*ones(6,1)];
9. for i=1:length(cdgpos)
10.     cdg=cdgpos(i,1:3);
11.     RM=R(3*i-2:3*i,:);
12.     a=cdg;
13.     l=-b'+a'+RM*p';
14.     for j=1:length(l)
15.         L(i,j)=norm(l(:,j));
16.
17.     end
18.     if cdgpos(i,3)<1.5
19.         disp('Zcdg<1.5')
20.         disp(cdgpos(i,3))
21.     end
22. end
23.
24. %Comprobación de longitudes
25. longitud=L > 4.7038;
26. Lmax=L(longitud);
27.
28. if Lmax~=0
29.     disp('Warning: Operative Limits')
30. end
31.
32. %Representacion movimiento plataforma
33. figure('units','normalized','outerposition',[0 0 1 1])
34. for i=1:15:length(cdgpos)
35.     if i>length(cdgpos)
36.         i=length(cdgpos);
37.     end
38.
39.     postopsim=ones(6,1)*cdgpos(i,1:3)+(pos_top-
       [zeros(6,2),2*ones(6,1)])*R(3*i-2:3*i,:);
40.     postopsim=[postopsim;postopsim(1,:)];
41.     posbasim=[pos_base;pos_base(1,:)];
42.
```

```

43.     plot3(postopsim(:,1),postopsim(:,2),postopsim(:,3),'o--
        r','linewidth',2);
44.     xlim([-3 3])
45.     ylim([-3 3])
46.     zlim([0 4])
47.     hold on
48.     plot3(posbasim(:,1),posbasim(:,2),posbasim(:,3),'o--
        k','linewidth',2);
49.     xlim([-3 3])
50.     ylim([-3 3])
51.     zlim([0 4])
52.     legs_view_ant=[posbasim(1,:);postopsim(1,:)];
53.     for j=2:6
54.         legs_view=[legs_view_ant;posbasim(j,:);postopsim(j,:)];
55.         legs_view_ant=legs_view;
56.     end
57.     for k=1:6
58.         plot3(legs_view(2*k-1:2*k,1),legs_view(2*k-1:2*k,2),legs_view(2*k-
            1:2*k,3),'-b','linewidth',2);
59.         xlim([-3 3])
60.         ylim([-3 3])
61.         zlim([0 4])
62.     end
63.     grid on
64.     tiempo=tsim(i);
65.     time=num2str(tiempo,4);
66.     title(time)
67.     pause(0.001)
68.     hold off
69. %     if i==1
70. %         pause(8)
71. %     end
72. End

```

Script de configuración

```
1. clear all;close all;clc;
2.
3. %clc;clear all;close all;
4.
5. deg2rad = pi/180;
6. x_axis = [1 0 0];
7. y_axis = [0 1 0];
8. z_axis = [0 0 1];
9.
10.% Connection points on base and top plate w.r.t. World frame at the center
11.% of the base plate
12.pos_base = [];
13.pos_top = [];
14.alpha_b = 2.5*deg2rad; % +- offset angle from 120 degree spacing on base
15.alpha_t = 10*deg2rad; % +- offset angle from 120 degree spacing on top
16.height = 2.0; % 2 meter height in home configuration
17.radius_b = 3.0; % base radius in meters
18.radius_t = 1.0; % top radius in meters
19.for i = 1:3
20. % base points
21. angle_m_b = (2*pi/3)* (i-1) - alpha_b;
22. angle_p_b = (2*pi/3)* (i-1) + alpha_b;
23. pos_base(2*i-1,:) = radius_b* [cos(angle_m_b), sin(angle_m_b), 0.0];
24. pos_base(2*i,:) = radius_b* [cos(angle_p_b), sin(angle_p_b), 0.0];
25.
26. % top points
27. % Top points are 60 degrees offset
28. angle_m_t = (2*pi/3)* (i-1) - alpha_t+ 2*pi/6;
29. angle_p_t = (2*pi/3)* (i-1) + alpha_t+ 2*pi/6;
30. pos_top(2*i-1,:) = radius_t* [cos(angle_m_t), sin(angle_m_t), height];
31. pos_top(2*i,:) = radius_t* [cos(angle_p_t), sin(angle_p_t), height];
32.end
33.% figure
34.% plot(pos_top(:,1),pos_top(:,2),'*r');
35.% hold on
36.
37.% permute pos_top points so that legs are end points of base and top
    points
38.pos_top = [pos_top(6,:); pos_top(1:5,:)]; %6th point on top connects to
    1st on bottom
39.% plot(pos_top(:,1),pos_top(:,2),'ob');
40.
41.% Compute points w.r.t. to the body frame in a 3x6 matrix
42.body_pts = pos_top' - height*[zeros(2,6);ones(1,6)];
43.
44.% leg vectors
45.legs = pos_top - pos_base;
```

```

46.leg_length = [ ];
47.leg_vectors = [ ];
48.for i = 1:6
49. leg_length(i) = norm(legs(i,:));
50. leg_vectors(i,:) = legs(i,:) / leg_length(i);
51.end
52.
53.% Calculate revolute and cylindrical axes
54.for i = 1:6
55. rev1(i,:) = cross(leg_vectors(i,:), z_axis);
56. rev1(i,:) = rev1(i,:) / norm(rev1(i,:));
57. rev2(i,:) = - cross(rev1(i,:), leg_vectors(i,:));
58. rev2(i,:) = rev2(i,:) / norm(rev2(i,:));
59. cyll(i,:) = leg_vectors(i,:);
60. rev3(i,:) = rev1(i,:);
61. rev4(i,:) = rev2(i,:);
62.end
63.
64.
65.% Coordinate systems
66.lower_leg = struct('origin', [0 0 0], 'rotation', eye(3), 'end_point', [0
    0 0]);
67.upper_leg = struct('origin', [0 0 0], 'rotation', eye(3), 'end_point', [0
    0 0]);
68.
69.for i = 1:6
70. lower_leg(i).origin = pos_base(i,:) + (3/8)*legs(i,:);
71. lower_leg(i).end_point = pos_base(i,:) + (3/4)*legs(i,:);
72. lower_leg(i).rotation = [rev1(i,:)', rev2(i,:)', cyll(i,:)'];
73. upper_leg(i).origin = pos_base(i,:) + (1-3/8)*legs(i,:);
74. upper_leg(i).end_point = pos_base(i,:) + (1/4)*legs(i,:);
75. upper_leg(i).rotation = [rev1(i,:)', rev2(i,:)', cyll(i,:)'];
76.end
77.
78.% Inertia and mass calculation
79.top_thickness = 0.05;
80.base_thickness = 0.05;
81.inner_radius = 0.03;
82.outer_radius = 0.05;
83.density = 76e3/9.81; % steel density in Kg/m^3
84.
85.%leg inertia and mass
86.[lower_leg_mass, lower_leg_inertia] = inertiaCylinder(density, ...
87. 0.75*leg_length(1),outer_radius, inner_radius);
88.[upper_leg_mass, upper_leg_inertia] = inertiaCylinder(density, ...
89. 0.75*leg_length(1),inner_radius, 0);
90.
91.% top and base plate mass and inertia
92.[top_mass, top_inertia] = inertiaCylinder(density, ...

```

```

93.top_thickness, radius_t, 0);
94.[base_mass, base_inertia] = inertiaCylinder(density, ...
95.         base_thickness,radius_b, 0);
96.
97.% PID controller gains
98.load('PID_gains3')
99.
100. % Force Saturation
101. force_act_max=5e4;
102.
103. %load('ShipMov_H1');
104. %load('ShipMov_H2');
105. %load('ShipMov_H25');
106. load('ShipMov_H28');
107. %load('ShipMov_H3');
108. %load('ShipMov_H35');
109.
110. %Formato entrada modelo simulink
111. data_roll=timeseries(PHI,t);
112. data_pitch=timeseries(THETA,t);
113. data_yaw=timeseries(PHI,t);
114. data_X=timeseries(X,t);
115. data_Y=timeseries(Y,t);
116. data_Z=timeseries(Z1,t);

```


Bibliografía

- [1] K. B. & E. E. N. P. V. A. Papanikolaou, «Statistical analysis of ship accidents occurred in the period 1990-2012 and assessment of safety level of ship types».
- [2] Airbus, «www.airbus.com,» 2017. [En línea]. Available: <https://www.airbus.com/content/dam/corporate-topics/publications/safety-first/Airbus-Commercial-Aviation-Accidents-1958-2017.pdf>.
- [3] «Hispaviacion,» [En línea]. Available: <http://www.hispaviacion.es/everis-aeroespacial-defensa-seguridad-muestra-fidae-la-ultima-version-del-uav-tucan/>.
- [4] Consorcio SADSIM, «Memoria Técnica SADSIM».
- [5] Intelligence Resource Program, «FAS,» [En línea]. Available: <https://fas.org/irp/program/collect/pioneer.htm>.
- [6] Wikipedia, «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/Wind_wave.
- [7] R. K. P., «Researchgate,» Marzo 2014. [En línea]. Available: https://www.researchgate.net/publication/277020159_Variable_Gain_Super_Twisting_Controller_for_the_Position_Stabilization_of_Stewart_Platform. [Último acceso: Abril 2019].
- [8] L. J. Domagoj Jakobovic, «THE FORWARD AND INVERSE KINEMATICS PROBLEMS FOR STEWART PARALLEL MECHANISMS,» Enero 2002. [En línea]. Available: https://www.researchgate.net/publication/254096088_The_forward_and_inverse_kinematics_problems_for_stewart_parallel_mechanisms. [Último acceso: Abril 2019].
- [9] A. González, «Cinemática directa para la Plataforma Stewart».
- [10] L. A. S. Velásquez, Modeling and control techniques of autonomous helicopters for landing on moving platforms, Sevilla, 2016.
- [11] MathWorks, «MathWorks,» [En línea]. Available: <https://es.mathworks.com/help/physmod/sm/mech/ug/designing-and-improving-controllers.html>.
- [12] Dpto. Ingeniería de Sistemas y Automática (Escuela Técnica Superior de Ingeniería, US), «Control Automático».
- [13] Dpto de Ingeniería de Sistemas y Automática (ETSI, Universidad de Sevilla), «Diseño Frecuencial».
- [14] Ingenostrum, «Desarrollo de modelos matemáticos de comportamiento de masas inerciales sometidas a

excitación marina,» 2018.

- [15] R. Stewart, «wikiwaves,» 2012. [En línea]. Available: https://wikiwaves.org/Ocean-Wave_Spectra.
- [16] T. B. E. B. H. C. D. C. K. E. J. E. H. G. D. H. P. K. A. M. P. M. D. O. K. R. W. S. a. H. W. Hasselmann K., Measurements of wind-wave growth and swell decay during the Joint North Sea Wave Project (JONSWAP), Hamburgo, 1973.
- [17] G. y. T.P.Gourlay, «Comparison of open-source code Nemoh with Wamit for cargo ship motions in,» Octubre 2016.
- [18] estudiasonavegas, «estudiasonavegas,» [En línea]. Available: <https://www.estudiasonavegas.com/titulos-nauticos/patron-de-navegacion-basica/105-apuntes/apuntes-pnb>.
- [19] Exlar, «<https://exlar.com/>,» [En línea]. Available: <https://exlar.com/product/gsx-series/>.
- [20] L. B. D. Jakobović, «Forward Kinematics of a Stewart Platform Mechanism,» Zagreb, 2002.
- [21] Wikipedia, «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/Stewart_platform.