



UNIVERSIDADE DA CORUÑA



Escola Politécnica Superior

Trabajo Fin de Máster

CURSO 2017/18

*VALIDACIÓN DEL SOFTWARE SIMULANCLA
MEDIANTE LA FABRICACIÓN Y SIMULACIÓN DE UN
PROTOTIPO DEL SISTEMA DE FONDEO*

Máster en Ingeniería Industrial

ALUMNA/O

Marta Frade Manso

TUTORAS/ES

Daniel Dopico Dopico

Emilio Sanjurjo Maroño

FECHA

JUNIO 2018

RESUMEN

En este momento de incertidumbre en el sector naval es cuando se debe poner más empeño en aportar estudios que contribuyan con iniciativas que permitan dar otra visión a los mercados. Es por ello que en este proyecto se propone alcanzar como objetivo la validación del software SIMULANCLA desarrollado por el laboratorio de ingeniería mecánica, LIM, de la UDC, que permitirá mejorar la maniobra de fondeo del buque sin la necesidad de una maqueta. Para lograr este objetivo, el estudio se estructura en tres partes a las que se añaden las conclusiones obtenidas y la bibliografía consultada. En la primera, se realiza el diseño de los modelos en CAD para construir un prototipo impreso en 3D que permita la fase experimental. En la segunda se desarrolla un sistema de control de la maniobra, medición de la fuerza de tiro (parámetro empleado para la validación) y tratamiento de esta señal medida. En la última parte, se aborda la simulación del modelo y se comparan los resultados extraídos con el estudio experimental.

RESUMO

Neste momento de incerteza no sector naval é cando debe poñerse máis empeño en aportar estudos que contribúan con iniciativas que permitan dar outra visión ós mercados. É por elo que neste proxecto se propón alcanzar como obxectivo a validación do software SIMULANCLA desenvolvido polo laboratorio de enxeñería mecánica, LIM, da UDC, que permitirá mellorar a manobra de fondeo do buque sen necesidade dunha maqueta. Para lograr este obxectivo, o estudo estrutúrase en tres partes ás que se engaden as conclusións obtidas e a bibliografía consultada. Na primeira, realízase o deseño dos modelos CAD para construír un prototipo impreso en 3D que permita a fase experimental. Na segunda desenvólvese un sistema de control da manobra, medición da forza de tiro (parámetro empregado para a validación) e tratamento desta sinal medida. Na última parte, abórdase a simulación do modelo e compáranse os resultados extraídos co estudio experimental.

ABSTRACT

At this time of uncertainty in the naval sector, it is when more effort must be put into providing studies that contribute with initiatives that give a different vision to the markets. For this reason, this project aims to achieve the validation of the SIMULANCLA software developed by the UDC's mechanical engineering laboratory, LIM, which will improve the ship's anchoring maneuver without the need for a prototype. To achieve this objective, the study is structured in three parts, to which the conclusions obtained and the bibliography consulted are added. First, the CAD models are designed to build a 3D printed prototype that allows the experimental phase. Secondly, a system is developed for controlling the maneuvers, measuring the tractive force (parameter used for validation) and processing this measured signal. Lastly, the simulation of the model is discussed and the results extracted are compared with the ones from the experimental study.



UNIVERSIDADE DA CORUÑA



Escola Politécnica Superior

**TRABAJO FIN DE MÁSTER
CURSO 2017/18**

*VALIDACIÓN DEL SOFTWARE SIMULANCLA
MEDIANTE LA FABRICACIÓN Y SIMULACIÓN DE UN
PROTOTIPO DEL SISTEMA DE FONDEO*

Máster en Ingeniería Industrial

Documento

MEMORIA TRABAJO FIN DE MÁSTER

AGRADECIMIENTOS

A Daniel Dopico Dopico, por la labor personal de dirección de este proyecto y su implicación, ayudando con su experiencia en la elaboración y desarrollo del mismo.

A Alberto Luaces Fernández por su gran apoyo y ayuda en la resolución de errores encontrados en el software.

A Emilio Sanjurjo Maroño por su constante e inestimable ayuda en la elaboración y corrección del proyecto, sobre todo en la parte de instrumentación.

A Ana Álvarez García por su ayuda, interés y explicaciones sobre el sistema de fondeo de buques.

A todos los componentes del Laboratorio de Ingeniería Mecánica, por el apoyo brindado.

Al Laboratorio de Ingeniería Mecánica de la Universidade da Coruña, por haberme permitido realizar el proyecto en sus instalaciones, y brindarme la oportunidad de trabajar con un grupo de profesionales.

A todos los que me habéis ayudado, muchas gracias.

Índice

Índice

Resumen	II
1 Introducción	1
1.1 Antecedentes.....	1
1.2 Motivación.....	2
2 Desarrollo de los modelos.....	3
2.1 Sistema de amarre y fondeo	3
2.2 Modelo inicial de la proa	4
2.3 Versiones iniciales	8
2.4 Escalado de los modelos	14
2.5 Eliminación de material de la proa para obtener un ahorro en costes	16
2.6 Eliminación extra de material y división de la proa en dos partes.....	16
2.7 Creación de una pieza extra para impresión transparente	17
2.8 Diseño del apoyo de la maqueta.....	20
3 Sistema de medición y control	22
3.1 Descripción global del experimento.....	22
3.2 Adquisición de datos	25
3.2.1 Teorema de Nyquist-Shannon	27
3.2.2 Diseño del amplificador.....	29
3.2.3 Diseño del filtro	31
3.2.4 Calibración experimental	36
3.3 Actuación	40
3.3.1 Funcionamiento del puente H	41
3.4 Control del experimento	47
3.4.1 Comunicación con el ordenador	48
4 Modelización del sistema	49
4.1 Modelización elegida para el sistema de fondeo	49
4.1.1 Definición del buque	51
4.1.2 Definición de sólidos del sistema de fondeo	52
4.1.3 Grados de libertad	55
4.1.4 El vector de variables.....	55
4.1.5 Fuerzas.....	55
5 Simulación de la maniobra en SIMULANCLA.....	60

5.1 Uso del software SIMULANCLA.....	60
5.2 Preparación de los modelos.....	68
5.3 Parámetros de la simulación.....	69
6 Resultados experimentales.....	78
7 Conclusiones.....	83
Bibliografía.....	84
Anexos.....	87
Anexo I: Plano diseño apoyo de la maqueta.....	87
Anexo II: Código Arduino.....	89

Índice de figuras

Figura 2-1. Barra de comandos del Rhinoceros	4
Figura 2-2. Modelo inicial de la proa en Rhinoceros.....	4
Figura 2-3. Fallo en el comando que deja un hueco en la proa	5
Figura 2-4. Comando "Unir" en Rhinoceros	5
Figura 2-5. Comando "_OffsetSrf" en Rhinoceros	6
Figura 2-6. Resultado final de la proa con el espesor en Rhinoceros.....	6
Figura 2-7. Bordas salientes en la proa del barco	7
Figura 2-8. Comando escalar en Rhinoceros	7
Figura 2-9. Modelo final de la proa sin bordes salientes.....	8
Figura 2-10. J-Swivel-B familia.....	8
Figura 2-11. J-Swivel-B familia prueba.....	8
Figura 2-12. Rosca Swivel	9
Figura 2-13. 5.6- End Shackle familia	9
Figura 2-14. 5.6-End Shackle_familia original	9
Figura 2-15. Pasador de la argolla	9
Figura 2-16. Grillete sin pasador	10
Figura 2-17. Grillete original	10
Figura 2-18. Pasador del grillete	10
Figura 2-19. Fabricacion cadena.asm	11
Figura 2-20. Fabricacion cadena parte 2.asm	11
Figura 2-21. Puntos de unión del montaje.....	11
Figura 2-22. Modelo de la caña original	12
Figura 2-23. Caña parte 1	12
Figura 2-24. Caña parte 2	12
Figura 2-25. Unión Uñas-Caña	12
Figura 2-26. Modelo original proa.....	13
Figura 2-27. Apoyo para la prolongación del escobén.....	13
Figura 2-28. Fases de diseño de la proa.....	14
Figura 2-29. Ventanas del escalado de piezas.....	14
Figura 2-30. Proa del barco con la geometría nueva.....	15
Figura 2-31. Panel para tapar la proa.....	15
Figura 2-32. Nueva versión de la proa (tercera versión).....	16
Figura 2-33. Cubierta, parte del banco del motor	17
Figura 2-34. Cubierta, parte de la regola.....	17
Figura 2-35. Cubierta, parte de la regola con agujero	18
Figura 2-36. Pieza para cubrir la regola	18

Figura 2-37. Nuevo modelo del casco.....	19
Figura 2-38. Nuevo modelo del grillete.....	19
Figura 2-39. Modelo de la caja.....	20
Figura 2-40. Modelo con las piezas de la proa montadas	20
Figura 2-41. Soporte con montaje final.....	21
Figura 3-1. Diagrama experimental.....	22
Figura 3-2. Conexiones Arduino.....	24
Figura 3-3. Diagrama de instrumentación	25
Figura 3-4. Funcionamiento muestreo x4	26
Figura 3-5. Resultados de la prueba de muestreo realizada	26
Figura 3-6. Onda senoide de 1 Hz.....	27
Figura 3-7. Muestreo de la señal de 1 Hz a 2 Hz	27
Figura 3-8. Señal de 1 Hz muestreada a 3 Hz.....	28
Figura 3-9. Señal de 1 Hz muestreada a 1'5 Hz.....	28
Figura 3-10. Señal reconstruida con frecuencia de muestreo de 1'5 Hz.....	28
Figura 3-11. Señal compuesta por ondas de 1 Hz, 2 Hz y 3 Hz	29
Figura 3-12. Señal compleja muestreada a 6 Hz.....	29
Figura 3-13. Esquema del amplificador de instrumentación	30
Figura 3-14. Gráfica de la ganancia del filtro Butterworth para distintos órdenes	31
Figura 3-15. Esquema de un filtro Sallen-Key	32
Figura 3-16. Esquema de un filtro Sallen-Key con flujo de corriente	32
Figura 3-17. Esquema explicativo del flujo de corrientes en el filtro	33
Figura 3-18. Esquema del filtro a construir.....	36
Figura 3-19. Resultados obtenidos en los distintos días.....	38
Figura 3-20. Resultados finales.....	39
Figura 3-21. Esquema puente H L298N.....	40
Figura 3-22. Montaje real.....	40
Figura 3-23. Esquema de funcionamiento del puente H.....	41
Figura 3-24. Explicación de los símbolos del esquema	42
Figura 3-25. Ejemplo 1 del funcionamiento del puente.....	43
Figura 3-26. Ejemplo 2 del funcionamiento del puente H	44
Figura 3-27. Ejemplo 3 del funcionamiento del puente H	45
Figura 3-28. Esquema de conexión de los diodos.....	45
Figura 3-29. Ejemplo para la justificación de los diodos	46
Figura 3-30. Funcionamiento del PWM	47
Figura 3-31. Código Matlab.....	48
Figura 4-1. Restricciones para los eslabones.....	52

Figura 4-2. Restricciones para el grillete y la caña	53
Figura 4-3. Restricciones para la caña y las uñas del ancla	54
Figura 4-4. Contacto normal.....	56
Figura 4-5. Contacto tangencial	56
Figura 4-6. Ángulo del ancla con la caña	59
Figura 5-1. Configuración de la simulación	61
Figura 5-2. Configuración del buque	62
Figura 5-3. Cálculo de las características del elemento	63
Figura 5-4. Configuración de la caña	64
Figura 5-5. Configuración del ancla.....	64
Figura 5-6. Configuración de la cadena.....	65
Figura 5-7. Configuración del grillete y el molinete	66
Figura 5-8. Fuerzas representadas en la simulación	67
Figura 5-9. Estructura de malla	67
Figura 5-10. Conversión modelo a formato .stl.....	68
Figura 5-11. Apertura del archivo .stl en Blender	68
Figura 5-12. Exportación del modelo a formato .obj	69
Figura 5-13. Botón de exportado a formato .obj	69
Figura 5-14. Parámetros de simulación - Configuración de la simulación.....	70
Figura 5-15. Parámetros de simulación - Configuración del buque	71
Figura 5-16. Obtención coordenadas en SolidEdge	72
Figura 5-17. Parámetros de simulación - Configuración de la caña.....	72
Figura 5-18. Parámetros de simulación - Configuración del ancla.....	73
Figura 5-19. Parámetros de simulación - Configuración de la cadena.....	74
Figura 5-20. Parámetros de simulación - Configuración del grillete y el molinete	74
Figura 5-21. Partes del montaje	76
Figura 5-22. Montaje final	77
Figura 6-1. Primer experimento.....	78
Figura 6-2. Segundo experimento.....	79
Figura 6-3. Comparación de velocidades entre experimentos.....	80
Figura 6-4. Comparativa entre Experimento 1 y 2	80
Figura 6-5. Valor del PWM en zonas de alto rozamiento.....	81

Índice de tablas

Tabla 1. Características Arduino Leonardo	23
Tabla 2. Características del grupo motor-reductora.....	23
Tabla 3. Balance de grados de libertad.....	55

Capítulo 1

Introducción

1 INTRODUCCIÓN

Las maniobras de fondeo en buques entrañan muchos riesgos, tanto en el propio proceso de fondeo sobre el fondo marino como en la recogida de las cadenas y las anclas. Estas obligatoriamente deben quedar perfectamente estibadas en el casco sin que se produzcan situaciones de inestabilidad o atascos de las mismas antes de llegar a una estiba satisfactoria. El diseño de la zona del casco sobre la que debe acomodarse el ancla en el proceso de estiba no es en absoluto trivial.

Hasta hace unos años, el estudio del sistema de fondeo de un buque se realizaba mediante el uso de maquetas, siendo costosas en tiempo y materiales. Además, limitan el estudio por los recursos necesarios y solamente ofrecen una aproximación general del comportamiento real. Con el uso del software SIMULANCLA, una novedosa tecnología que evita la necesidad de realizar maquetas del sistema de fondeo y todos los equipos que intervienen en esta maniobra, para el diseño de cascos de buques de grandes dimensiones, se busca simplificar este análisis, permitiendo un estudio más cercano a la realidad, fiable y con unos tiempos de ejecución más rápidos.

El software desarrollado durante los últimos siete años facilita a los ingenieros del astillero el diseño geométrico de la regola o nicho del buque, de tal forma que la maniobra para levar el ancla se lleve a cabo de manera satisfactoria, así como la elección de los eslabones que deben componer la cadena y todos los elementos mecánicos implicados en las maniobras.

Por lo tanto, con el fin de verificar su correcto funcionamiento, el objetivo del presente proyecto es la validación de este software para la simulación de maniobra de fondeo de buques. La validación objeto del proyecto consistirá en ensayar el mismo sistema que se va a simular y verificar que la fuerza de tiro medida a lo largo del ensayo coincida con la simulada.

Se partirá de los archivos CAD de un buque desarrollados en el proyecto de Diego Saya Carro, modificando los existentes y creando otros nuevos para permitir la construcción de una maqueta creada mediante la impresión 3D de dichos modelos. Además se preparará un sistema de control y medición para la obtención de datos reales del ensayo de la maqueta, que se contrastarán contra los obtenidos de SIMULANCLA al simular esta misma maqueta. Este proceso y sus resultados nos llevarán a la validación del software de simulación.

1.1 Antecedentes

Hasta hace poco, se fabricaba, para cada diseño nuevo de buque que había que desarrollar, una maqueta de madera que representaba la maniobra de fondeo a escala 1:10. Estas maquetas son costosas en tiempo y no proporcionan la fiabilidad requerida, puesto que ni la escala ni los materiales en que están hechas son los reales, de modo que permiten sólo una evaluación cualitativa y aproximada de la bondad de la solución de diseño propuesta.

La creación de una de estas maquetas supone aproximadamente 1500 horas de trabajo y la implicación de diversos departamentos.

La nueva tecnología para el diseño de todos los elementos que intervienen en el proceso de fondeo supone un cambio de paradigma con respecto a la metodología empleada hasta la fecha en los astilleros, basada en las citadas maquetas de madera a

escala del casco, cadena, ancla y de todos los sistemas que intervienen en el proceso de fondeo y leva de anclas

Hay diversos factores que se deben tomar en consideración para el diseño, como las distintas posiciones en las que se puede encontrar el ancla en el momento del izado y el posible balanceo y/o inclinación del buque durante el mismo.

Asimismo, hay que asegurarse de que no se bloquea el mecanismo cuando el contacto entre el ancla y el casco se produce en una posición desfavorable.

El diseño debe asegurar que no se produzca el bloqueo del ancla durante la maniobra y que ésta adopte la posición esperada en el casco del buque cuando esté completamente recogida.

El software desarrollado permite simular, con gran detalle, la dinámica de las maniobras de leva de anclas, basándose en las técnicas de simulación de sistemas multicuerpo.

Por lo tanto los resultados del programa permitirán saber a lo largo de la simulación, por un lado, las fuerzas de tiro del motor, fuerzas de contacto normal (la fuerza de reacción entre un par de elementos, por ejemplo, entre eslabones) y fuerzas de rozamiento que sufre el sistema de fondeo diseñado y, por otro lado, comprobar cómo se movería todo el sistema y ver por ejemplo los posibles atascos mencionados, todo ello con más fiabilidad que con una maqueta y ahorro en costes y tiempo.

1.2 Motivación

Como se dijo en el apartado anterior, el software SIMULANCLA va a permitir realizar una simulación completa y detallada, sin la necesidad de inversión en tiempo y coste de maquetas. Gracias al programa, será posible definir la maniobra con mayor similitud a la situación real.

El empleo de una maqueta, en vez de la simulación con SIMULANCLA, sólo permite una aproximación de la maniobra real, pues el empleo de otros materiales y una escala distintos a los reales, aportan una evaluación cualitativa que solamente nos puede acercar a una idea del comportamiento real de los elementos que intervienen en el proceso.

Sin embargo, el software SIMULANCLA permite realizar un estudio más completo, pudiendo analizar la dinámica de la maniobra, así como visualizar la maniobra, con la ventaja de poder conocer diferentes datos en tiempo de simulación, todo con las propiedades de los materiales reales y a escala real.

De este modo, será posible conocer con una mayor exactitud el sistema de fondeo de un buque, pudiendo comparar distintos modelos y comprobar que configuraciones se adaptan mejor a los requerimientos del equipo técnico. Todo esto sería muy difícil de materializar con el empleo de maquetas, pues requeriría de una inversión económica y horaria muy extensa.

Este software además ha reducido el plazo necesario desde que se dispone de la estructura de la zona de proa hasta obtener las formas definitivas de la regola, nicho, piezas fundidas necesarias, etc., así como la realización de modificaciones de manera fácil y rápida que permiten optimizar las maniobras.

Capítulo 2

Desarrollo de los modelos

2 DESARROLLO DE LOS MODELOS

Se ha decidido realizar la fabricación de la maqueta mediante impresión en 3D a partir de unos modelos de CAD (Carro 2016). En este capítulo detallaremos los pasos seguidos para el desarrollo de estos modelos, con una sección al final donde también se detallará diversas ofertas de impresión buscadas y la elección final.

El modelo se compone de las siguientes partes:

- Un modelo de la proa
- Un modelo de la cadena
- Un modelo del ancla

Otros datos de interés antes de empezar a explicar el desarrollo del modelo son:

- La escala con respecto del modelo original proporcionado es de 1:21 aproximadamente y
- los eslabones de la cadena se diseñaron de acuerdo a la *International Standard ISO 1704 (1973)* (International Organization for Standardization 1973). De esta forma las dimensiones están normalizadas y todas las medidas de los eslabones están parametrizadas respecto al diámetro de la cadena.

En el siguiente apartado se hará una breve introducción del sistema de fondeo

2.1 Sistema de amarre y fondeo

El sistema de fondeo de un buque es el encargado de inmovilizar el mismo y de separarlo de la acción de las corrientes y del viento mediante elementos capaces de fijarse en el fondo marino (Comas 1990). Estos elementos forman parte del buque y van unidos a él. Aunque puede incluir más elementos, las partes fundamentales de un sistema de fondeo son las siguientes:

- Anclas
- Cadenas
- Molinete
- Escobén
- Estopor
- Tubos de las cadenas
- Caja de cadenas

El elemento principal es el ancla, encargado de fondear el buque por la acción de su propio peso y por sus uñas. Funciona de manera que, si es sometida a una fuerza paralela al fondo del mar, esta tiende a clavarse en el mismo. Para ello será necesario que el buque conserve una pequeña velocidad. Están fabricadas en hierro o acero y se unen al buque mediante un cabo o cadena.

En cuanto a las cadenas, se ha ido generalizando el uso de cadenas de acero. En estas se pueden distinguir los siguientes elementos: grillete de ancla, eslabón final, eslabón grande, eslabón giratorio, eslabón ordinario y eslabón de unión. Haciendo referencia a los tipos de cadenas, pueden ser con conrete o sin conrete. El conrete es la pieza que se coloca en el interior del eslabón en sentido del eje menor para evitar su deformación cuando está trabajando a tracción. Además, los eslabones deben ser normalizados y estandarizados para una mayor facilidad de sustitución cuando sea necesario.

El molinete es el equipo que se encarga de elevar el ancla del buque. Éste gira sobre un eje horizontal provocando el paso de los eslabones hacia la caja de cadenas.

El escobén es un conducto de sección circular o elíptica que se abre entre la cubierta y la armadura de un buque para permitir el paso de las cadenas y alojar la caña del ancla en la posición de estiba.

El estopor es un elemento de acero colocado en la cubierta entre el molinete y el escobén. Este retiene la cadena evitando que la tensión actúe directamente sobre el molinete. Todos los estopores tienen dos partes principales: una parte fija con una ranura que permite el paso del eslabón en posición vertical y una parte móvil que trata de retener al eslabón que se encuentra en la ranura.

La caja de cadenas será el lugar del buque donde se estiban las cadenas de las anclas. Deben colocarse debajo del molinete y en las proximidades de su vertical. Las cadenas de cada ancla deben estibarse por separado, siendo necesaria la existencia de un mamparo que evite que se puedan mezclar. El fondo de la caja de cadenas debe hacerse de tal modo que permita el achique del agua y fango que se pueda ir depositando. Una vez introducido el sistema de fondeo, se pasará a explicar el desarrollo de los modelos CAD.

2.2 Modelo inicial de la proa

Partimos de un modelo inicial en Rhinoceros (McNeel, R. 2013) provisto por Diego Saya Carro (Carro 2016), que consiste en la proa del barco pero en chapa (sin espesor). El programa que vamos a utilizar para el modelado de las piezas va a ser Solid Edge ST8 (Siemens 2015). Por lo tanto el trabajo inicial consistió en, usando una versión de Rhinoceros provista por la Escuela Politécnica Superior de Ferrol, darle espesor a la pieza y redimensionar los bordes de la cubierta para evitar que sobresaliesen. Los comandos que a continuación se indican como `_comando` se escriben en la barra de comandos del Rhinoceros que se puede ver en la siguiente imagen.



Figura 2-1. Barra de comandos del Rhinoceros

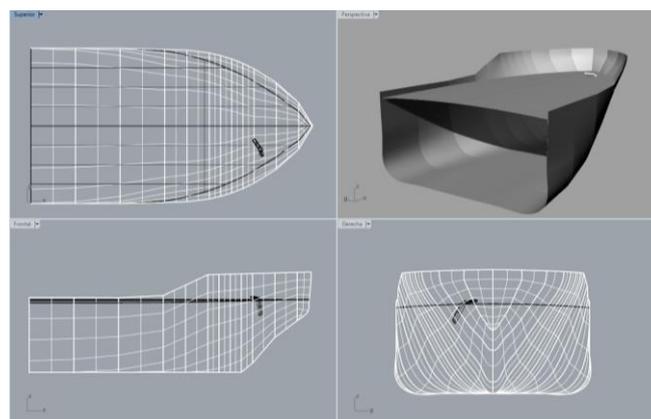


Figura 2-2. Modelo inicial de la proa en Rhinoceros

Primero se unen una serie de superficies entre sí para evitar puntos singulares y que nos dé resultados como el que vemos a continuación.

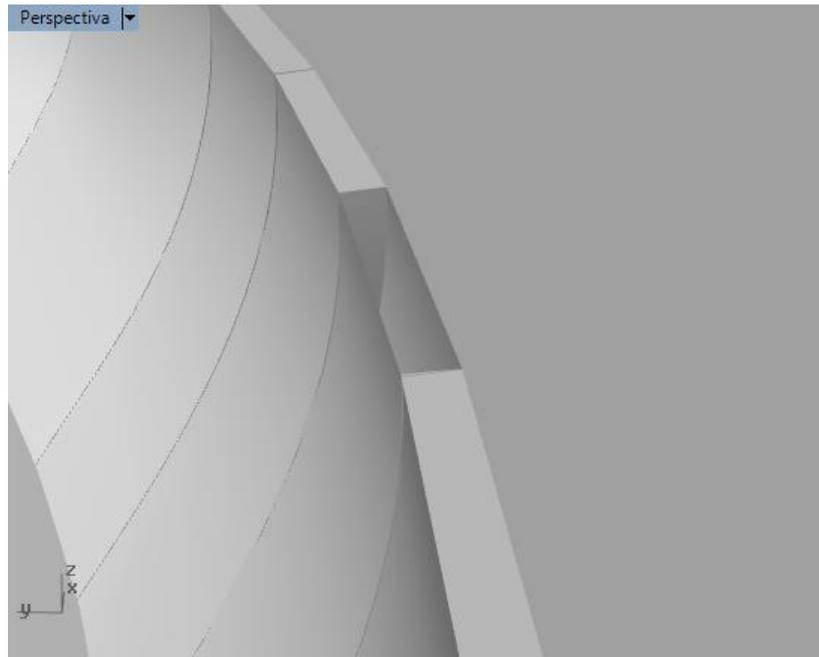


Figura 2-3. Fallo en el comando que deja un hueco en la proa

Para ello se empleará el comando “_join” en las siguientes superficies.

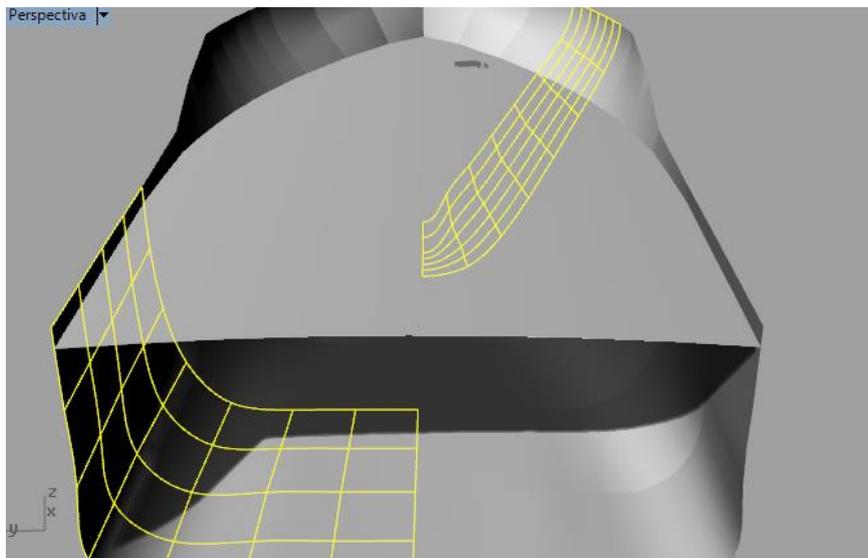


Figura 2-4. Comando "Unir" en Rhinoceros

Para la primera tarea que consiste en proporcionar espesor a la chapa se empleará el comando “_OffsetSrf” en las caras que se ven a continuación (extendemos hacia dentro para conservar las dimensiones exteriores de la pieza).

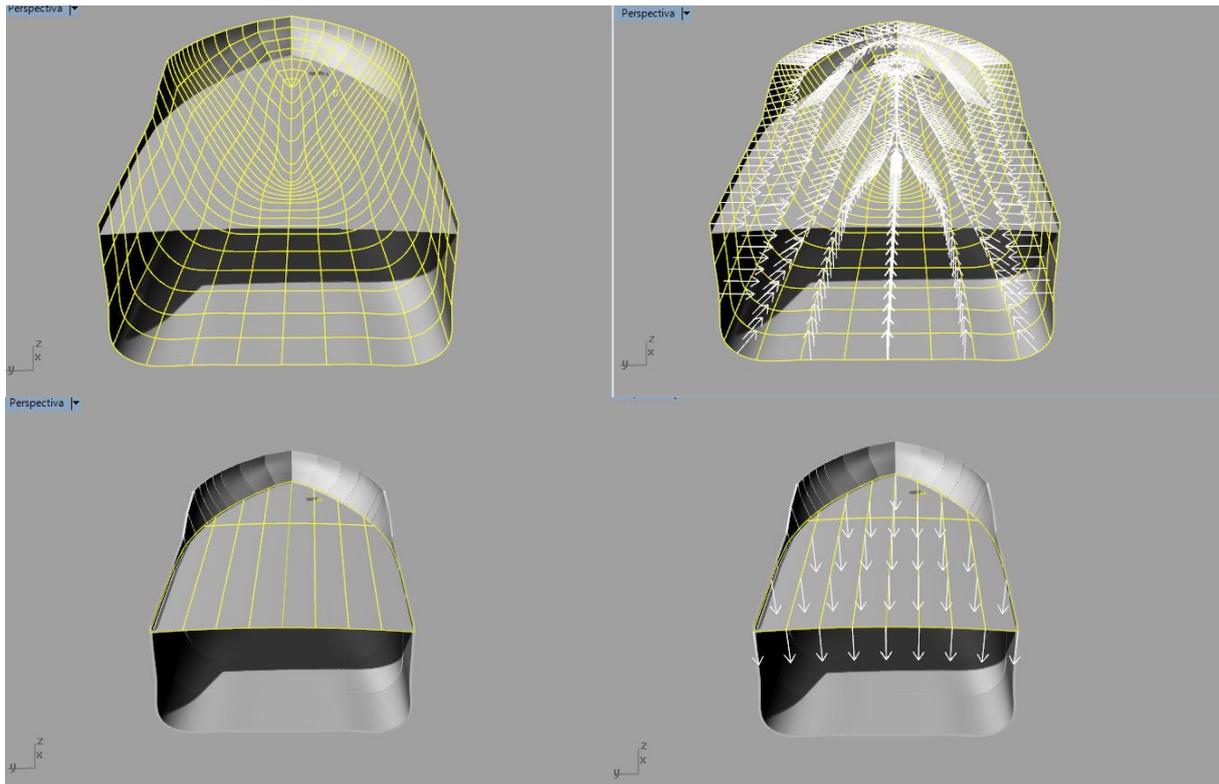


Figura 2-5. Comando "_OffsetSrf" en Rhinoceros

Obteniendo el siguiente resultado.

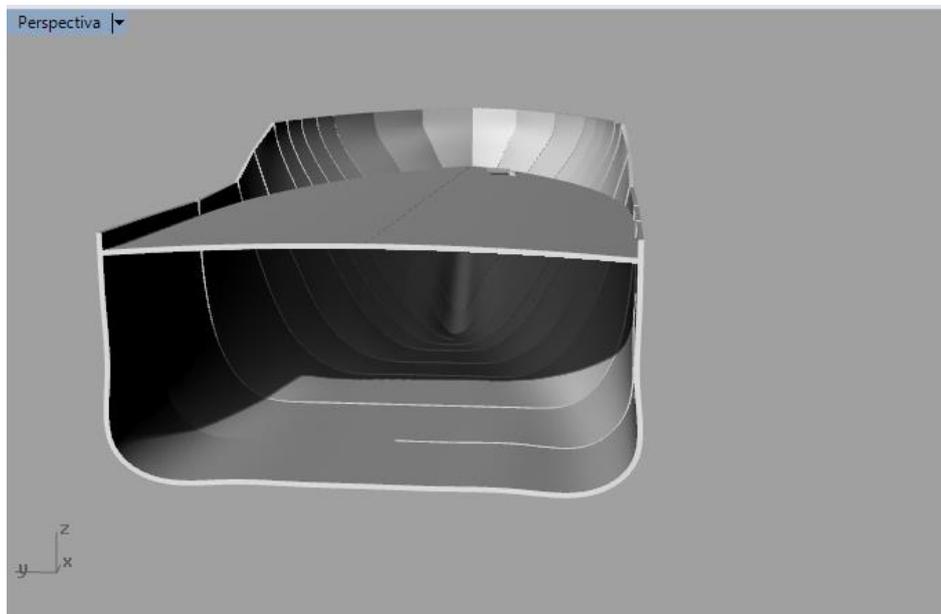


Figura 2-6. Resultado final de la proa con el espesor en Rhinoceros

En la Figura 2-7 se puede observar como sobresalen los bordes de la cubierta a través del casco.

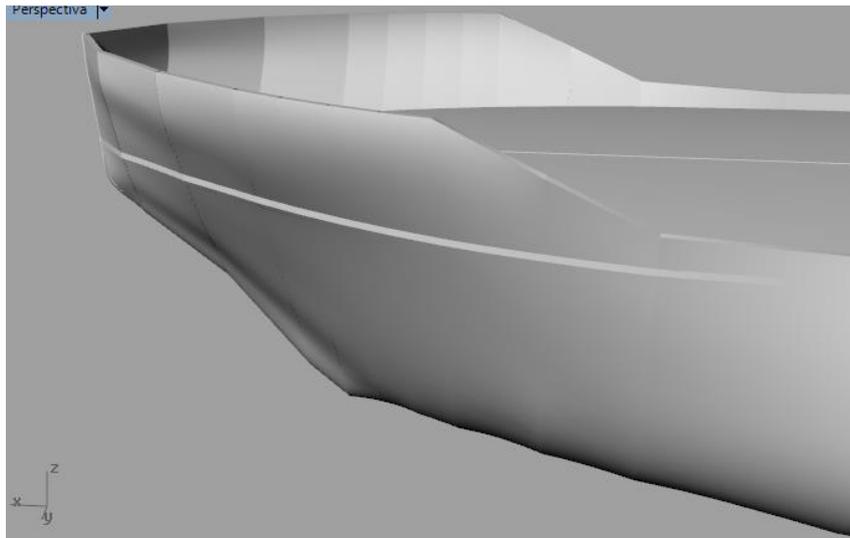


Figura 2-7. Bordes salientes en la proa del barco

Para solucionar esto se empleó el comando de escalado no uniforme, “Transformar→Escalar→Escala no uniforme”.

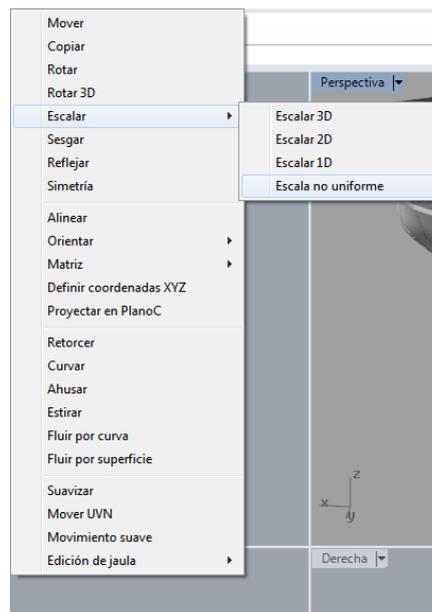


Figura 2-8. Comando escalar en Rhinoceros

Y tras diversas pruebas se consiguió el resultado que puede observarse en la Figura 2-9.

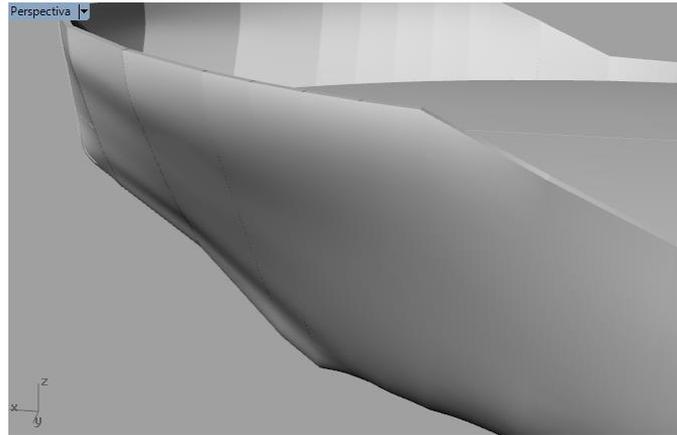


Figura 2-9. Modelo final de la proa sin bordes salientes

Este modelo se exportó a formato .stl compatible con otros programas de CAD, en nuestro caso, se utilizó Solid Edge ST8.

2.3 Versiones iniciales

En primer lugar se creó la cadena del ancla a partir de los siguientes archivos desarrollados y proporcionados por el tutor Daniel Dopico Dopico (Dopico, Simulancla 2018):

- 5.5-Kenter Link_familia
- 5.2-Enlarged Link_2.8
- 5.1-Common Link_familia
- J-Swivel-A_familia
- J-Swivel-B_familia_prueba
- 5.3-End Link_2.8
- 5.6-End Shackle_familia

Los modelos proporcionados fueron modificados para introducir nuevos sistemas de coordenadas para una mayor facilidad de ensamblaje entre las piezas. Además al archivo “J-Swivel-B_familia_prueba” se le realizaron cambios con respecto a su versión original “J-Swivel-B_familia”. Estos cambios se pueden apreciar en las siguientes figuras.

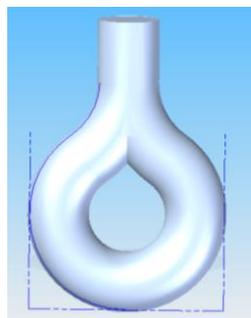


Figura 2-11. J-Swivel-B familia prueba

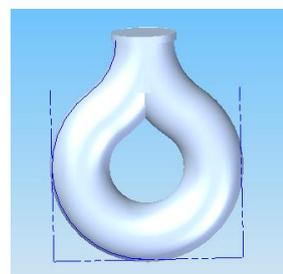


Figura 2-10. J-Swivel-B familia

Como se puede observar en las anteriores imágenes los cambios consistieron en la eliminación del “tope” de la pieza original y la extensión de esta parte para poder ser ensamblada con la pieza anterior llamada “J-Swivel-A_familia”. Para evitar que la pieza ensamblada se saliese de su posición se creó la pieza “Rosca_Swivel” que puede verse en la Figura 2-12.

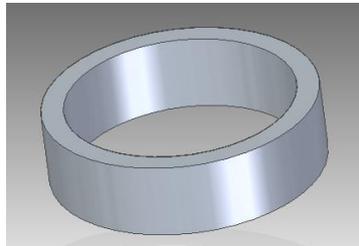


Figura 2-12. Rosca Swivel

Otra de las piezas modificadas con el propósito de que sean ensamblables en el montaje real fue “5.6-End Shackle_familia”.

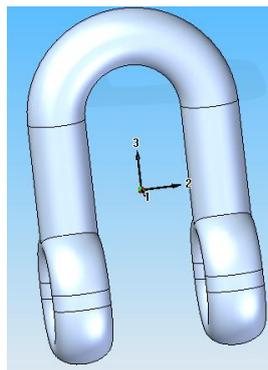


Figura 2-13. 5.6- End Shackle familia

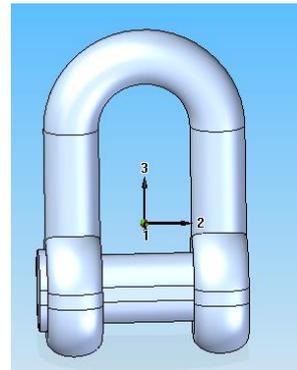


Figura 2-14. 5.6- End Shackle_familia original

Como se puede observar en las anteriores figuras, se eliminó el pasador al grillete para que este pudiera ser ensamblado en el montaje real y se creó una pieza nueva, que se puede ver en la Figura 2-15, para imprimirla como por separado y así poder realizar la unión de la cadena con el ancla.

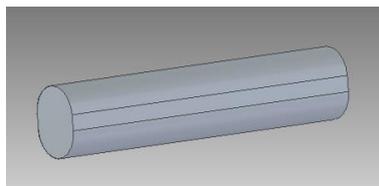


Figura 2-15. Pasador de la argolla

La última pieza que se modificó fue el grillete que funciona como nexo de unión entre la cadena y el ancla.

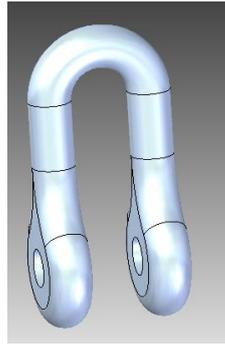


Figura 2-16.
Grillete sin
pasador

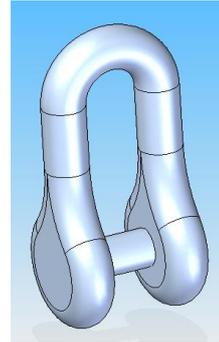


Figura 2-17.
Grillete original

Como en el caso anterior se eliminó el material del pasador y se crearon los agujeros necesarios para su ensamblaje. El pasador se creó en un archivo independiente para permitir unir el grillete al ancla (ver Figura 2-18).

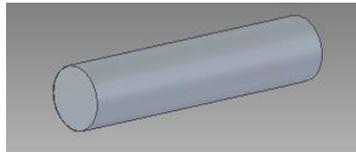


Figura 2-18. Pasador
del grillete

Una vez clarificados los cambios con respecto a los modelos originales, procederemos a detallar el orden de los eslabones que componen la cadena; se empezará desde la parte que conecta al motor. El orden y la cantidad de eslabones se detallan a continuación:

- Un 5.5-Kenter Link_familia (elemento nº 1)
- Un 5.1-Common Link_familia (elemento nº 2)
- Dos 5.2-Enlarged Link_2.8 (elementos nº 3-4)
- Un 5.1-Common Link_familia (elemento nº 5)
- Un 5.5-Kenter Link_familia (elemento nº 6)
- Veinticinco 5.1-Common Link_familia (elementos nº 7-32)
- Un J-Swivel-A_familia (elemento nº 33)
- Un J-Swivel-B_familia_prueba (elemento nº 34)
- Un 5.2-Enlarged Link_2.8 (elemento nº 35)
- Un 5.3-End Link_2.8 (elemento nº 36)
- Un 5.6-End Shackle_familia (elemento nº 37)

Para que los modelos de los eslabones al ser impresos en la impresora 3D no salgan pegados y puedan tener movimiento relativo entre ellos se les puso, durante la creación del archivo de ensamblaje, una distancia entre ellos de 1,10 mm. Para ello se fue a los archivos de los eslabones y se modificó la posición del sistema de coordenadas de forma que en el ensamblaje la distancia entre los eslabones fuese la detallada anteriormente.

Además para permitir que el modelo de la cadena, una vez impresa, presentase un movimiento como la real y que los elementos no estuviesen soldados, se efectuó la impresión de la cadena en dos partes. Para ello se hizo un primer ensamblaje, llamado

“Fabricacion cadena.asm” que comprendía el montaje hasta el “J-Swivel-A_familia”, y un segundo ensamblaje, llamado “Fabricacion cadena parte 2.asm” que contiene desde la “J-Swivel-B_familia_prueba” hasta la “5.6-End Shackle_familia”. Los archivos de ensamblaje creados se pueden observar en las figuras Figura 2-19 y Figura 2-20.

Las dos partes se unirán entre sí, en el montaje real, ensamblando el J-Swivel-A y el J-Swivel-B y manteniendo la unión en su lugar con la pieza de la rosca elaborada anteriormente para este fin.

El aspecto de estos dos ensamblajes puede verse en las figuras Figura 2-19 y Figura 2-20.

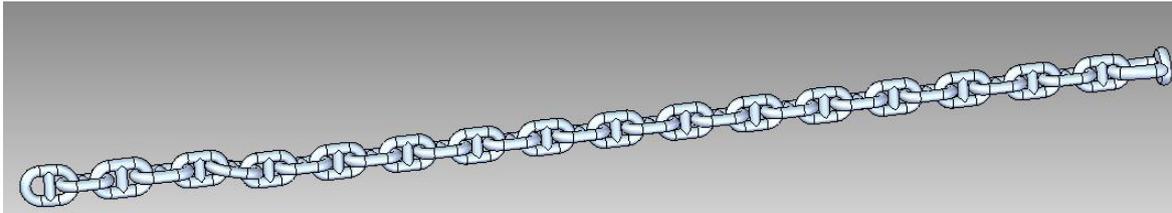


Figura 2-19. Fabricacion cadena.asm

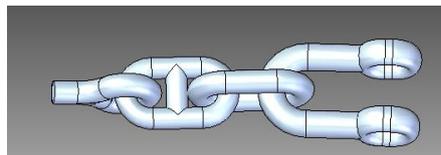


Figura 2-20. Fabricacion cadena parte 2.asm

En la siguiente imagen se puede ver como quedarían los puntos de ensamblaje entre las dos partes de la cadena y el ancla.

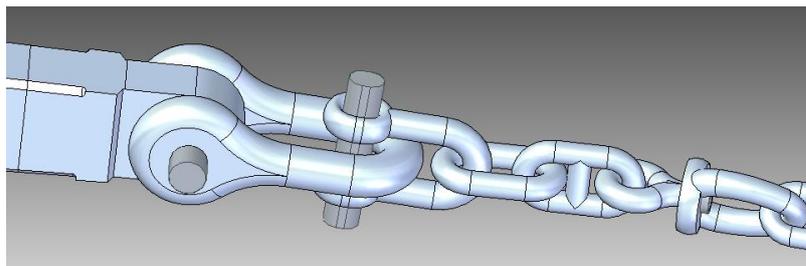


Figura 2-21. Puntos de unión del montaje

A continuación se verán los cambios realizados al modelo original de la caña. En la siguiente imagen puede verse el modelo original.

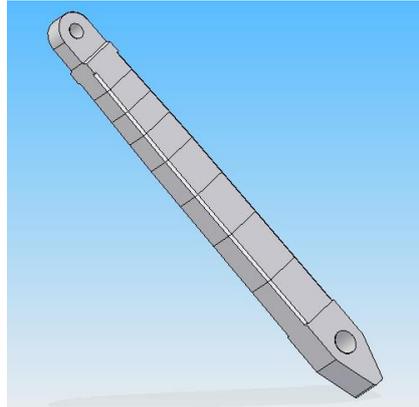


Figura 2-22. Modelo de la caña original

En las figuras anteriores puede observarse que se tiene el mismo problema que en los otros casos para poder realizar el montaje de la maqueta una vez impresa y que esta permita el movimiento relativo de rotación en el eje. Para evitar que salga soldada con el ancla, la parte que se une a esta la tendremos que dividir en dos piezas. El resultado se muestra en las figuras Figura 2-23 y Figura 2-24

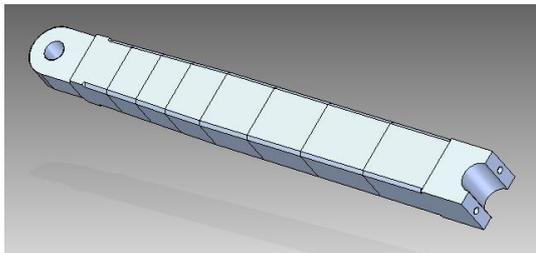


Figura 2-23. Caña parte 1

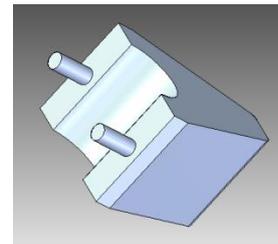


Figura 2-24. Caña parte 2

Estas partes se unirán alrededor de un cilindro que tiene las uñas del ancla y cuyo resultado se puede observar en la imagen a continuación:

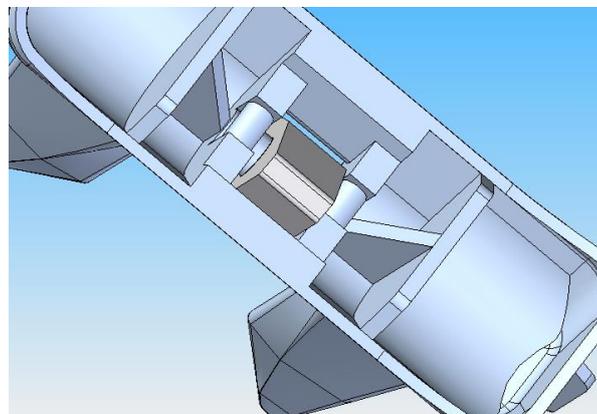


Figura 2-25. Unión Uñas-Caña

Por último se observa cómo se ha modificado el modelo de la proa inicial que ya se definió en el apartado uno de este capítulo.

Una vez exportada a .stl e importada en Solid Edge la pieza tiene el siguiente aspecto.

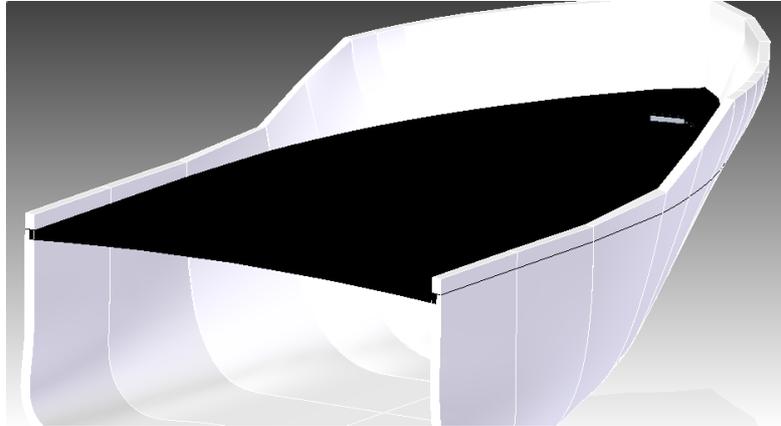


Figura 2-26. Modelo original proa

Sobre este modelo lo que se hizo fue crear una base para colocar el motor de forma que quede perpendicular a la dirección del apoyo para la salida de la cadena de la regala y unos apoyos para esta zona, que se muestra en la Figura 2-27.

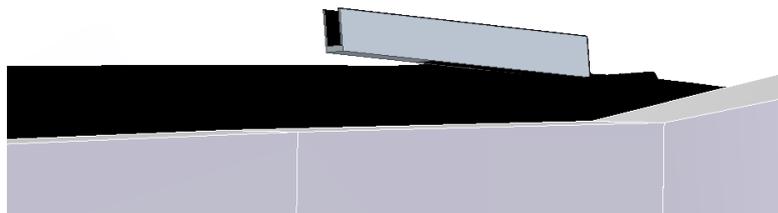


Figura 2-27. Apoyo para la prolongación del escobén

Por lo tanto el modelo, una vez aplicados los cambios antes mencionados y después de una serie de iteraciones, queda como se ve en la Figura 2-28

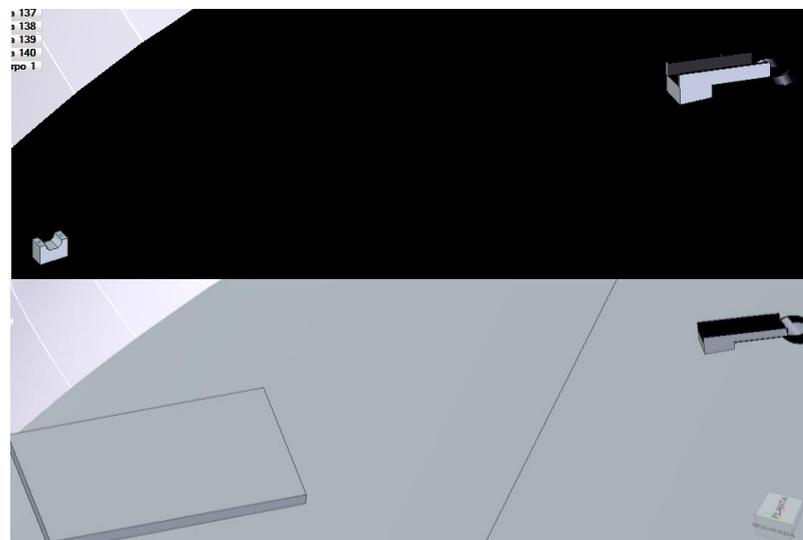




Figura 2-28. Fases de diseño de la proa

Una vez definido todo el modelo, el siguiente paso fue exportarlo a .stl y comprobar que todo encajase. Después de esta comprobación se solicitaron los pedidos a varias empresas de impresión 3D que se eligieron después de hacer un cribado inicial, de entre varias, por su fiabilidad.

El primer problema encontrado al recibir las respuestas de las distintas empresas, fue que la maqueta era demasiado grande para una impresora 3D convencional e imprimirla en una industrial salía muy costoso además de que el número de empresas se acortaba bastante por lo que hubo que escalarla para hacerla más pequeña. Esta segunda versión del modelo se discutirá en el siguiente apartado.

2.4 Escalado de los modelos

El límite inferior de tamaño vino dado por los eslabones, ya que hay un tamaño mínimo por debajo del cual no pueden imprimir las impresoras 3D convencionales por resultar la pieza demasiado pequeña. Por lo tanto la escala por la que nos decantamos al final fue de aproximadamente 9,5:10. Con esta escala procedimos a escalar todos los archivos del modelo creando piezas nuevas en Solid Edge y usando el comando “Insertar copia de pieza” y poniendo la escala descrita:

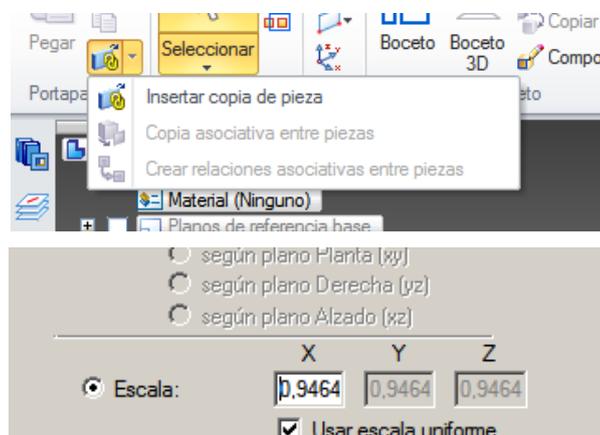


Figura 2-29. Ventanas del escalado de piezas

Además de esto, con la intención de que en el futuro la maqueta pudiese sumergirse en agua, se creó un archivo nuevo donde se dibujó una placa y se añadió a la geometría de la proa una estructura de forma que después se le pudiese insertar una placa. A continuación se pueden observar ambos cambios.

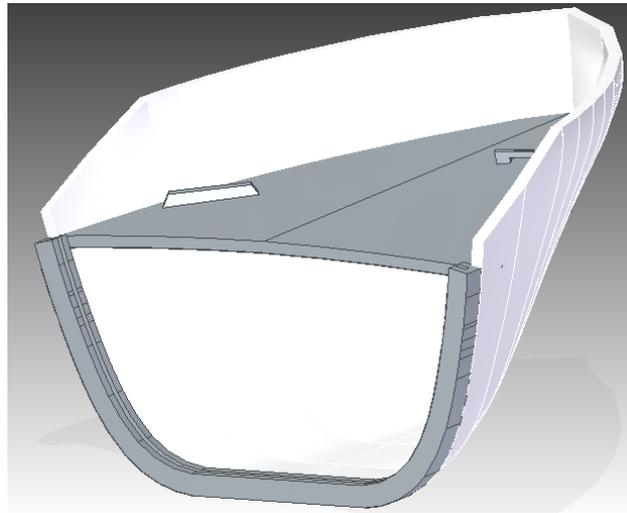


Figura 2-30. Proa del barco con la geometría nueva

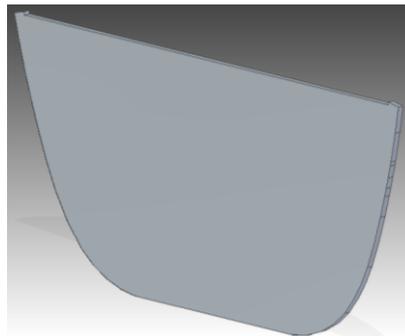


Figura 2-31. Panel para tapar la proa

Una vez terminado con esto se volvió a exportar todo a .stl y se envió a las distintas empresas.

Con el fin de minimizar los gastos de envío, dado que la variación de precios no resultaba muy significativa, se decidió escoger fabricar todo en la empresa en la que por término medio saliese más barata ya que en calidad y material todas tenían ofertas similares.

El siguiente problema que surgió fue que había algunas empresas que no imprimían la proa o la cadena, por lo que en principio esas se descartaron. Las restantes sí ofrecían la impresión de todos los componentes pero el precio de la proa era desorbitado, por lo que se decidió hacer cambios con el fin de disminuir la cantidad de material. Estos cambios se explicarán en el siguiente apartado.

2.5 Eliminación de material de la proa para obtener un ahorro en costes

Como primera reducción se decidió eliminar la nueva geometría que se había incorporado a la proa y por lo tanto también la placa ya que éstas dos piezas eran bastante material y no estaban relacionadas con el fin principal de este proyecto.

La siguiente reducción consistió en eliminar todo el casco, menos una parte de la regola, y dejar solo la cubierta. Pero este cambio requirió de la creación desde cero de esta parte del casco ya que Solid Edge no reconocía el casco como un sólido y por tanto no podíamos realizar operaciones de vaciado sobre él. Además se transformó la cubierta inclinada por la línea de crujía en totalmente plana por consejo del fabricante de las piezas, ya que fabricarla con aquella inclinación consumía más material de relleno, haciendo la pieza más cara.

Por lo tanto, y dado que para la validación del modelo no influye la inclinación de la cubierta ya que al simulador le daremos los CAD actualizados con todos nuestros cambios y sólo interesa la maniobra de subida de la cadena, se decidió aplicar este cambio. Esto implicó crear también la cubierta desde cero. El resultado final puede observarse en la siguiente imagen:

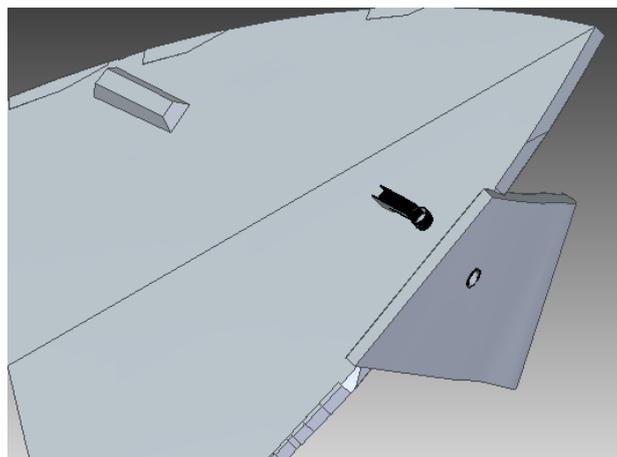


Figura 2-32. Nueva versión de la proa (tercera versión)

Se vuelve una vez más a enviar las nuevas versiones a las empresas de impresión 3D. Los nuevos cambios están descritos en la siguiente sección.

2.6 Eliminación extra de material y división de la proa en dos partes

Los nuevos presupuestos que llegan siguen siendo muy caros pero sirven para cribar el número de empresas restantes hasta quedarnos con una. Se decidió eliminar más material de cubierta, del casco y dividir la impresión, siguiendo los consejos de la empresa de impresión 3D elegida, reduciendo también el espesor de las piezas. El resultado es el siguiente:

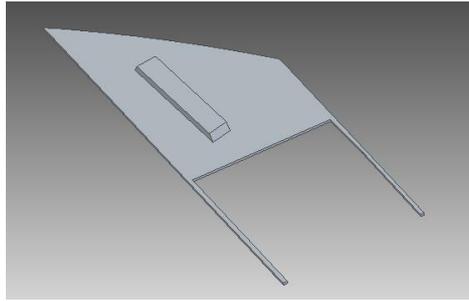


Figura 2-33. Cubierta, parte del banco del motor

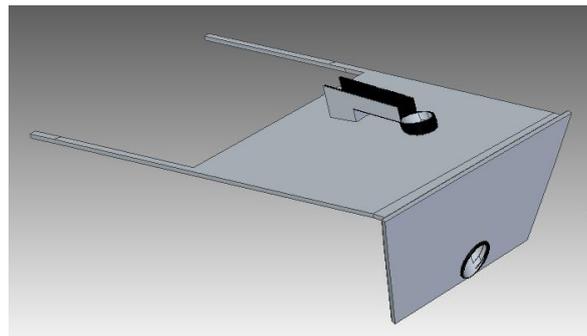


Figura 2-34. Cubierta, parte de la regala

Los listones presentes en las dos piezas sirven para guardar la distancia y posición relativa entre las piezas.

Aparte de estos cambios, como se quiere poder ver el desplazamiento de la cadena por el interior de la regala, se pide a la empresa que está pieza la imprima con un material transparente que resulta ser un poco más caro.

Una vez se recibió el nuevo presupuesto observamos que con el nuevo material transparente el precio era desorbitado. Por lo tanto se propusieron nuevos cambios que se detallan a continuación.

2.7 Creación de una pieza extra para impresión transparente

Lo que se decidió fue sólo hacer transparente una parte de la regala y practicar una abertura en la cubierta para poder ver a través de ella. La razón por la que no se deja descubierta también la parte del agujero de la regala es evitar que la cadena, al ser recogida, se salga de su camino ya que en las simulaciones previas ésta roza con la parte superior de la regala. La pieza queda de la siguiente forma:

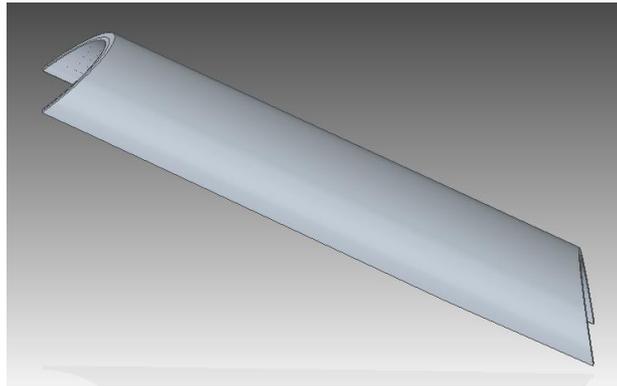


Figura 2-35. Cubierta, parte de la regala con agujero

Otra consecuencia de este cambio es la obtención de una nueva pieza para imprimir, que se puede observar en la Figura 2-36.

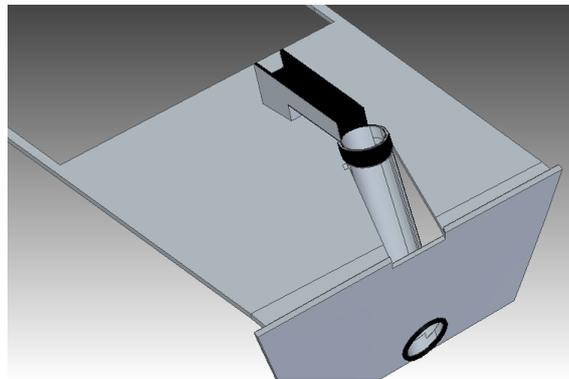


Figura 2-36. Pieza para cubrir la regala

Las piezas se envían de nuevo para presupuestar y la respuesta que llega está dentro de un rango aceptable de precios, por lo que se ultiman detalles y se envían las piezas para realizar la impresión.

Sobre este modelo final se hicieron dos modificaciones extra. La primera fue sobre el casco cuyo nuevo modelo se puede ver en la Figura 2-37 y la segunda sobre el grillete (ver Figura 2-38).

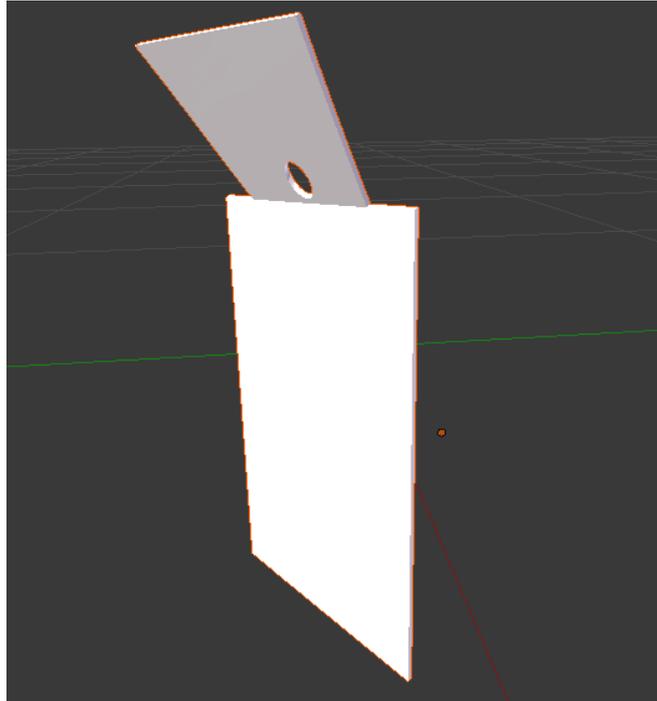


Figura 2-37. Nuevo modelo del casco

La razón por la que se añadió esa sección vertical es para representar la parte de la caja donde se apoya toda la maqueta y sobre la que podría contactar el ancla, ya que de no tener esto en cuenta el programa consideraría que no hay ninguna superficie en esa zona, lo que provocaría que los resultados difiriesen de la realidad.

También se modificó el modelo del grillete ya que al mandar a imprimir el modelo a la empresa hubo una equivocación en el envío de los elementos de la cadena y ancla, enviándose unas versiones anteriores a distinta escala que el casco del buque. El nuevo modelo puede observarse en la Figura 2-38.

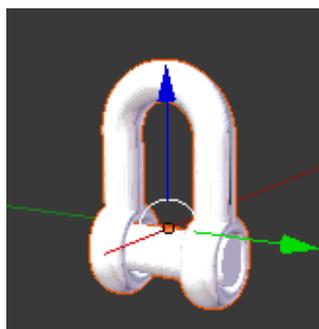


Figura 2-38. Nuevo modelo del grillete

2.8 Diseño del apoyo de la maqueta

Dado que hubo que eliminar gran parte del casco y de la cubierta, para apoyar las dos piezas de la proa resultantes se decidió construir una caja de madera. Para ello se hizo un modelo de CAD de la caja que se puede ver en la Figura 2-39.

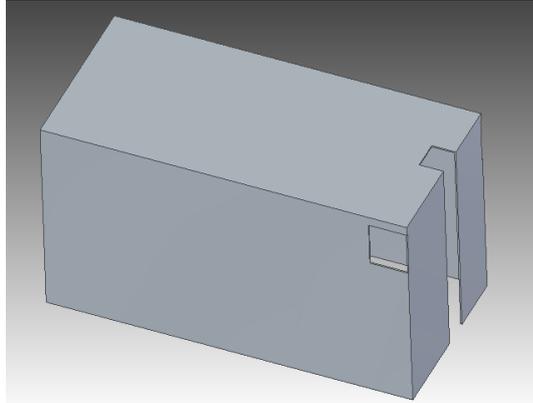


Figura 2-39. Modelo de la caja

En la siguiente imagen se puede observar como quedarían las piezas de la proa montadas. Además se dibuja como línea de referencia la línea de crujía.

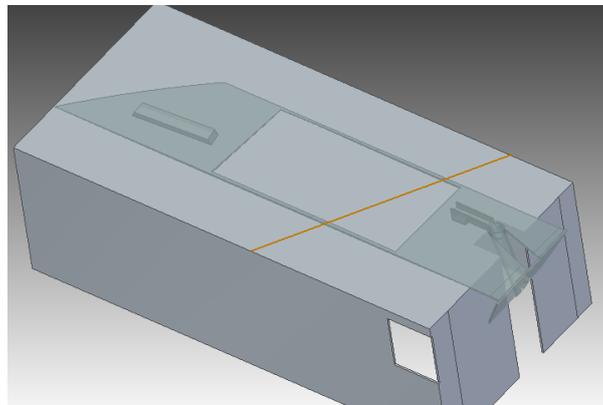


Figura 2-40. Modelo con las piezas de la proa montadas

A partir del modelo de esta caja se obtienen unos planos que se pueden ver en el anexo.

Finalmente de entre varios presupuestos se eligió un carpintero de la provincia de la Coruña y se le llevaron los planos, obteniendo el resultado que se puede observar en la Figura 2-41.



Figura 2-41. Soporte con montaje final

Capítulo 3

Sistema de medición y control

3 SISTEMA DE MEDICIÓN Y CONTROL

3.1 Descripción global del experimento

Se empezará este capítulo mostrando el esquema experimental para el ensayo de la maqueta:

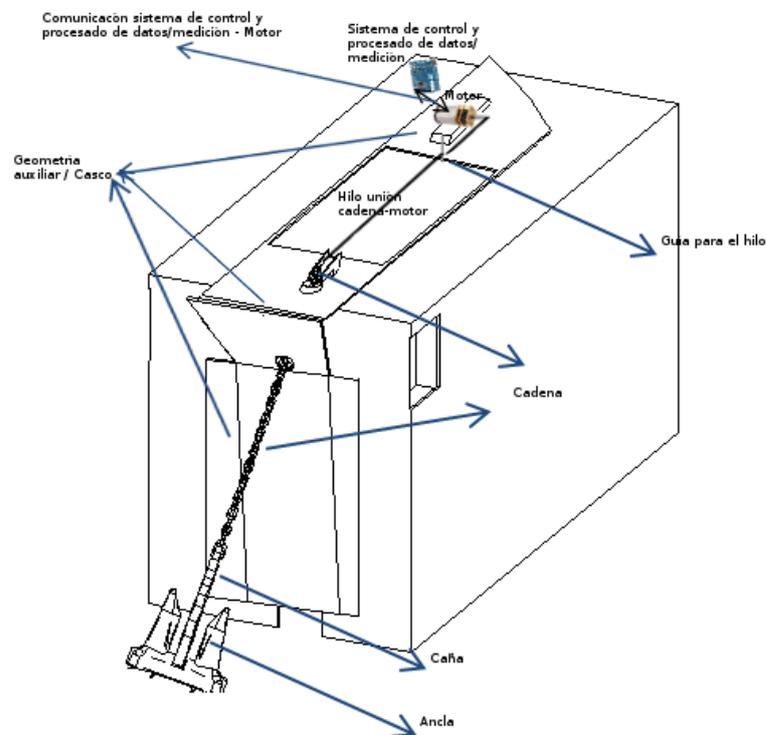


Figura 3-1. Diagrama experimental

Para el controlador se decidió usar un Arduino Leonardo (Arduino 2017), una placa de desarrollo basada en el controlador ATmega32u4, ya que ya se disponía de uno en el laboratorio, además de tenerse experiencia y conocimiento previo de uso y programación de los Arduinos. Posee las siguientes características:

Microcontrolador	ATmega32u4
Voltaje de funcionamiento	5V
Voltaje de alimentación recomendado	7-12V
Límites de voltaje de alimentación	6-20V
Nº pines digitales de E/S	20
Canales PWM	7
Canales de entrada analógicos	12
Corriente continua para Pin E/S	40 mA
Corriente continua para Pin 3.3V	50 mA
Velocidad de reloj	16 MHz
Longitud	68.6 mm
Ancho	53.3 mm
Peso	20 g

Tabla 1. Características Arduino Leonardo

Los datos del conjunto motor-reductora (Pololu 2017) son:

Tipo motor	Motor de corriente continua con escobillas
Voltaje de alimentación	6V
Relación transmisión	298:1
Parámetros con el motor sin carga	45 RPM, 20 mA
Parámetros con el motor a máximo par	2.0 kg-cm, 0.36 A

Tabla 2. Características del grupo motor-reductora

La Figura 3-2 muestra las conexiones del controlador. Sólo se detallan los pines empleados.

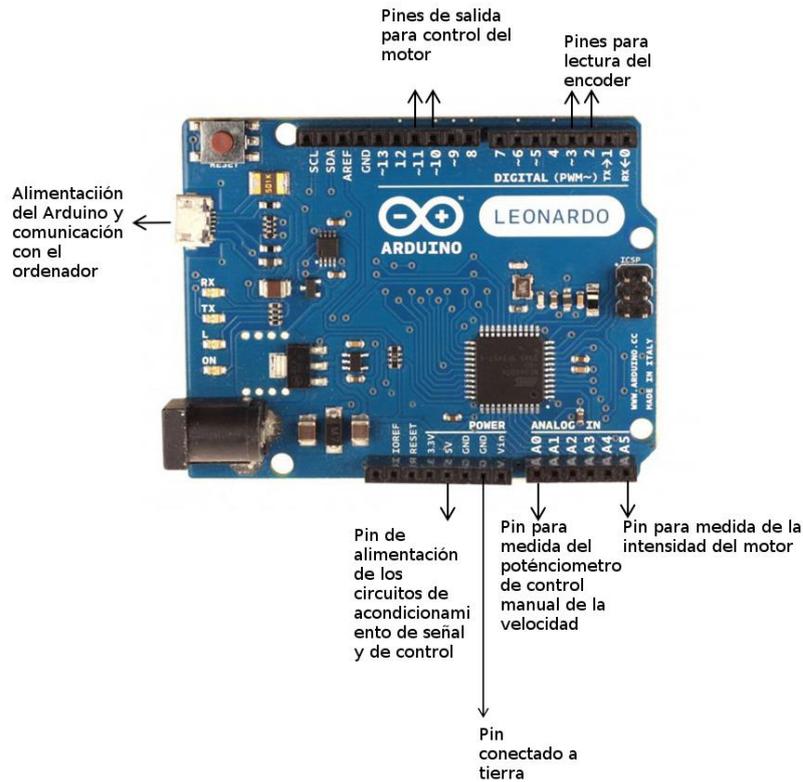


Figura 3-2. Conexiones Arduino

En la Figura 3-3 se muestra un esquema que detalla la parte de instrumentación del proyecto. El objetivo de este circuito es el de controlar la velocidad y medir la fuerza de tiro. Para el control de velocidad se emplea un encoder y para determinar la fuerza se mide la corriente que pasa por el motor. A pesar de que se quiere medir la intensidad que pasa a través del motor, el Arduino sólo mide tensiones, por lo tanto se tendrá que transformar este valor de intensidad en un valor de tensión conocido. Para ello pondremos una resistencia a la salida del motor de valor conocido. Se explicará más acerca de esta resistencia en el apartado 3.2.2. Esta señal será amplificada y filtrada antes de llegar al Arduino, que la enviará al ordenador para que sea registrada.

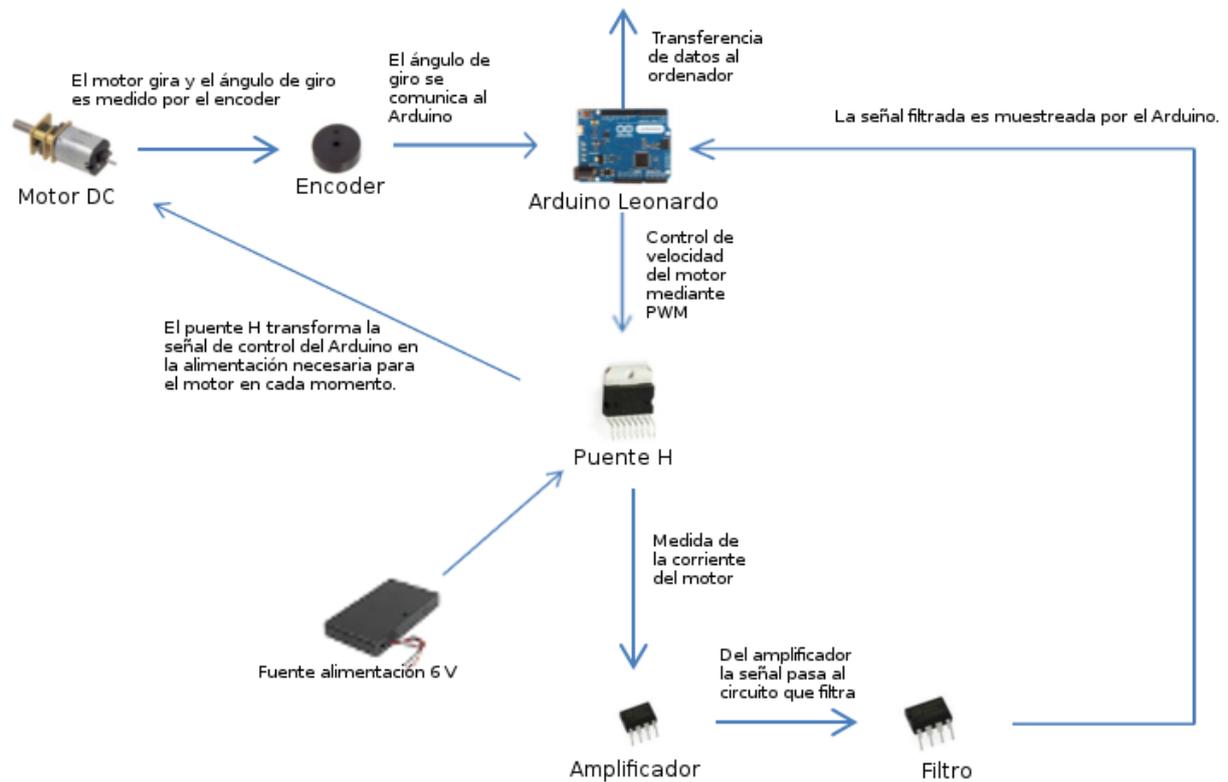


Figura 3-3. Diagrama de instrumentación

3.2 Adquisición de datos

Se divide en dos partes: la primera consiste en medir la velocidad de giro del motor y la segunda en medir la fuerza de tiro.

Para medir la velocidad se emplea un encoder rotatorio, un sensor electromecánico que convierte la posición angular de un eje a una señal analógica o digital. Las vueltas medidas por este sensor en combinación con el tiempo medido por el Arduino, permite obtener la velocidad de giro.

El encoder usado en este proyecto es un encoder incremental, llamado así porque no proporciona una posición absoluta, sino que proporciona incrementos angulares desde una posición de referencia que se fija al principio de una maniobra. Dichos incrementos angulares se codifican mediante dos señales cuadradas con un desfase de 90° entre ellas (ver Figura 3-4).

El encoder usado en este trabajo (Polulu 2017) tiene una resolución de 3 pulsos por revolución en cada una de las señales, que resulta en una resolución final de 12 pulsos por revolución si se muestrea a 4x, es decir, verificando las rampas de subida y de bajada de las dos señales proporcionadas por el encoder. Para esta tarea se ha empleado la biblioteca "Encoder" del Arduino obtenida de la página del Arduino <http://playground.arduino.cc/Main/RotaryEncoders>. Aplicando la relación de la reductora, 298:1, nos da una resolución de 3576 pulsos por vuelta a la salida de la reductora.

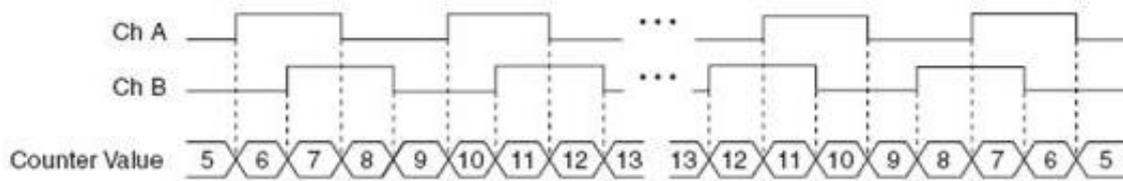


Figura 3-4. Funcionamiento muestreo x4

(Fuente: <https://www.motioncontroltips.com/faq-what-do-x1-x2-and-x4-position-encoding-mean-for-incremental-encoders>)

La fuerza de tiro se estimará a través de la corriente que consume el motor. Como el sistema a estudiar es bastante rígido, se prevén variaciones bastante rápidas de la fuerza, lo que implica que la frecuencia de muestreo también ha de ser alta. Para ver la máxima frecuencia de muestreo que es capaz de alcanzar el sistema empleado en este trabajo, se hacen diversas medidas con el Arduino, con muestras de 1000 datos, en las que se lleva a cabo la ejecución del programa que va a ser usado en el experimento, y se recogen los valores de tiempo en Matlab (Mathworks 2014), proceso que puede observarse en la Figura 3-5. Después se calcula la diferencia entre los sucesivos tiempos leídos y se toma el valor máximo de estos. Una vez conocida la frecuencia de muestreo que permite nuestro sistema, podemos determinar la máxima frecuencia contenida en la señal muestreada que se puede medir aplicando el teorema de Nyquist-Shannon (Olshausen 2000), que se explica en sección 3.2.1. Esta información se empleará para diseñar el un filtro que elimine las frecuencias por encima de la máxima, para evitar el aliasing.

Con estos experimentos, se ha comprobado que la frecuencia máxima a la que se puede muestrear ha resultado ser de 1400Hz. Sabiendo este valor y según el teorema de muestreo de Nyquist–Shannon (Olshausen 2000), la frecuencia más alta contenida en la señal analógica debe ser la mitad de la frecuencia de muestreo, en nuestro caso 700 Hz, si se quiere obtener la información de esta sin aliasing (señales falsas que no se pueden distinguir de la verdadera ni eliminar con un posterior tratamiento de señales, provocando que obtengamos datos erróneos).

T (microsegundos)	T _{i+1} - T _i (microsegundos)	Máx (microsegundos)	Máx (segundos)	Frecuencia (Hz)
28956200	632	700	0,0007	1428,57143
28956832	584			
28957416	584			
28958000	584			
28958584	576			
28959160	576			
28959736	592			
28960328	580			
28960908	584			
28961492	584			
28962076	584			
28962660	592			
28963252	572			
28963824	592			
28964416	580			
28964996	584			
28965580	588			
28966168	576			
28966744	588			
28967332	572			

Figura 3-5. Resultados de la prueba de muestreo realizada

3.2.1 Teorema de Nyquist-Shannon

Este teorema indica que la tasa de muestreo mínima para evitar aliasing, debe de ser al menos el doble de la frecuencia más alta contenida en la señal:

$$f_s \geq 2f_c \quad (3.1)$$

donde f_s es la frecuencia de muestreo y f_c es la frecuencia más alta contenida en la señal. Lo que se define como frecuencia de Nyquist es la mínima frecuencia de muestreo f_s que cumple la condición en (3.1). Frecuencias menores producen el llamado *aliasing*.

Considérese una señal compuesta de una sola onda senoide de 1 Hz:

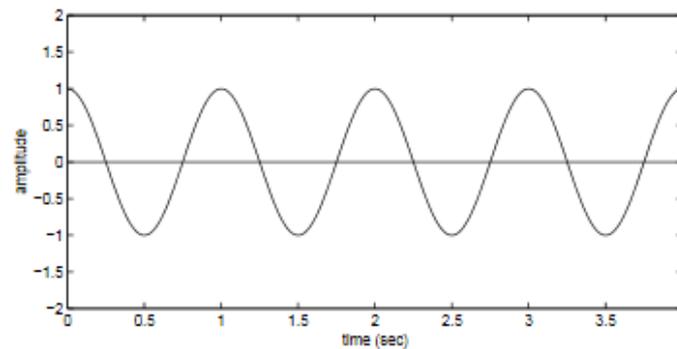


Figura 3-6. Onda senoide de 1 Hz

(Fuente: <http://redwood.berkeley.edu/bruno/npb261/aliasing.pdf>)

Si muestreamos esta señal a 2 Hz, como indica el teorema de Nyquist, se observa que es suficiente para capturar todos los picos y valles de la onda como se observa en la Figura 3-7:

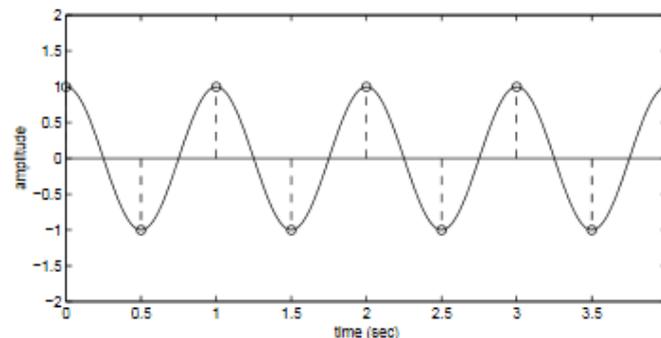


Figura 3-7. Muestreo de la señal de 1 Hz a 2 Hz

(Fuente: <http://redwood.berkeley.edu/bruno/npb261/aliasing.pdf>)

Si muestreamos esta señal a una frecuencia aún mayor, por ejemplo 3 Hz, entonces se obtienen más que suficientes puntos para capturar las variaciones en la señal, como se observa en la Figura 3-8:

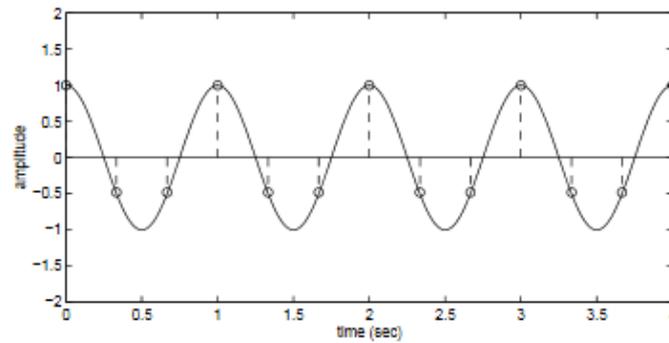


Figura 3-8. Señal de 1 Hz muestreada a 3 Hz

(Fuente: <http://redwood.berkeley.edu/bruno/npb261/aliasing.pdf>)

Pero si se muestrea a una frecuencia menor que la frecuencia de Nyquist (2 Hz en nuestro ejemplo), como, por ejemplo, 1'5 Hz, no habría suficientes puntos para capturar todos los picos y valles:

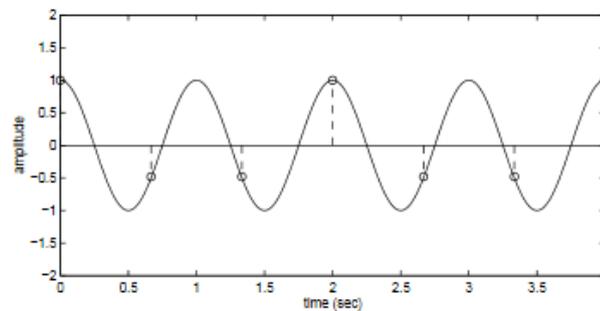


Figura 3-9. Señal de 1 Hz muestreada a 1'5 Hz

(Fuente: <http://redwood.berkeley.edu/bruno/npb261/aliasing.pdf>)

Lo que provocaría que estuviésemos obteniendo la información errónea de la señal. La persona que obtuviese estas medidas, si no tuviera ningún conocimiento previo de cómo era la señal original, creería que la señal podría tener la siguiente forma:

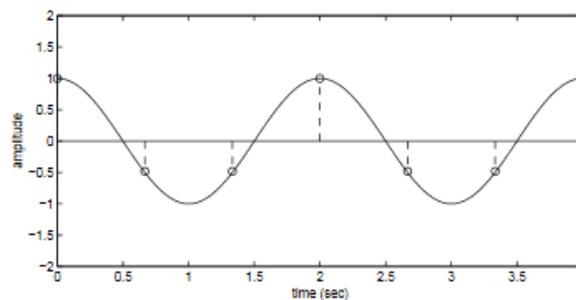


Figura 3-10. Señal reconstruida con frecuencia de muestreo de 1'5 Hz

(Fuente: <http://redwood.berkeley.edu/bruno/npb261/aliasing.pdf>)

Esta señal es un "alias" de la señal original, obtenida por muestrear por debajo de la frecuencia de Nyquist, y por lo tanto sin la información suficiente para reconstruir adecuadamente la señal original.

Como ejemplo final se verá este teorema aplicado a una señal compleja, formada por múltiples frecuencias, como es el caso en este proyecto. Por el teorema de Fourier se sabe que cualquier señal continua puede ser descompuesta en suma de senos y cosenos de distintas frecuencias. Por ejemplo, la siguiente onda está compuesta sumando ondas de 1 Hz, 2 Hz y 3 Hz:

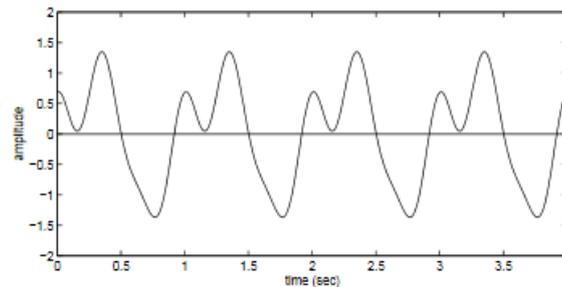


Figura 3-11. Señal compuesta por ondas de 1 Hz, 2 Hz y 3 Hz

(Fuente: <http://redwood.berkeley.edu/bruno/npb261/aliasing.pdf>)

De acuerdo con el teorema de Nyquist, la frecuencia de muestreo debe ser el doble de la frecuencia más alta que compone la señal. Por lo tanto, en este caso, f_s debería ser 6 Hz, ya que la frecuencia más alta es 3 Hz.

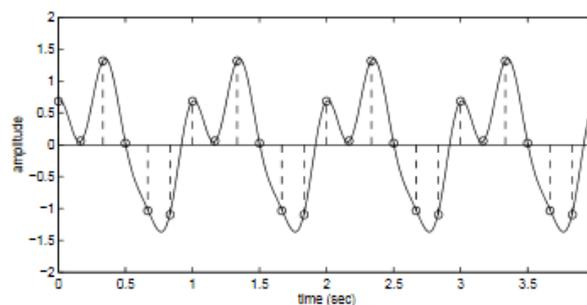


Figura 3-12. Señal compleja muestreada a 6 Hz

(Fuente: <http://redwood.berkeley.edu/bruno/npb261/aliasing.pdf>)

Como se ve en la anterior figura esta frecuencia es suficiente para obtener los puntos suficientes para una perfecta reconstrucción de la señal.

En nuestro caso, como la frecuencia más alta de muestreo ha resultado ser de 1400 Hz, la frecuencia más alta de nuestra señal medida será 700 Hz. Para asegurar que esto sea así se construirá un filtro con frecuencia de corte de 700 Hz, que se explicará en el apartado 3.2.3, evitando de esta forma que los alias de frecuencias superiores a esta causen que muestreemos información incorrecta.

3.2.2 Diseño del amplificador

El valor de la resistencia a colocar en serie con el motor ha de ser bajo para no limitar la potencia del mismo. Sin embargo, un valor bajo de resistencia produce un valor bajo de tensión. Dado que la resolución de medida de señales analógicas del microcontrolador es limitada (en este caso es de 10 bits, con lo que se obtienen 1024 valores ente 0 y 5V), es necesario amplificar dicha señal para mejorar la precisión de la medida.

3.2.2.1 Valor de la resistencia a la salida del motor

Para elegir el valor de la resistencia hay que tener en cuenta dos aspectos. El primero, como se ha mencionado anteriormente, es que si el valor de la resistencia es muy alto, le

quitamos potencia al motor ya que parte de la energía se iría en superar esta resistencia que le hemos puesto, y por otro, si es muy pequeña, la amplificación que haría falta hacer sería muy grande, perdiendo ancho de banda a la salida del posterior filtro y la medida podría ser menos precisa. Otra consideración es la potencia que aguanta la resistencia en sí. Las resistencias de las que se dispone son de 0,25 W.

Por lo tanto, teniendo todo esto en cuenta y que en el peor de los casos el valor máximo de corriente, mirando la hoja de especificación del fabricante, es de 360 mA (la corriente con el motor bloqueado) se procede a realizar los siguientes cálculos:

$$P = \frac{1}{4} W$$

$$I = 360 mA$$

$$P = I \cdot V = I^2 \cdot R$$

$$R = \frac{P}{I^2} = \frac{0,25}{0.360^2} = 1,92 \Omega$$

El valor máximo que soportarían estas resistencias con esta corriente sería entonces de 1,92 Ω . Teniendo en cuenta las consideraciones de que el valor no puede ser muy alto ni muy bajo y aproximando el valor a uno comercial, se decide poner una resistencia de 1 Ω . En la siguiente expresión vemos que cumple el requerimiento de no superar los 0,25 W:

$$P = I^2 \cdot R = 0,360^2 \cdot 1 = 0,1296 W < 0,25 W$$

Una vez obtenido el valor de la resistencia, calculamos el valor de tensión máxima que tenemos a la entrada del amplificador:

$$V = I \cdot R = 0,360 \cdot 1 = 0,360 V$$

Por lo tanto, nos interesará amplificar este valor unas 10 veces para que quede un valor de unos 3,6 V que está en un rango intermedio entre 0 V y 5 V (el valor máximo soportado por el Arduino), teniendo en cuenta lo indicado acerca de la precisión de conversión del Arduino antes de iniciar este apartado.

Para ello disponemos de un amplificador de instrumentación "AD623AN". Mirando su hoja de características obtenemos la siguiente ecuación para calcular R_G (Analog devices 2017) en función de la ganancia que queremos:

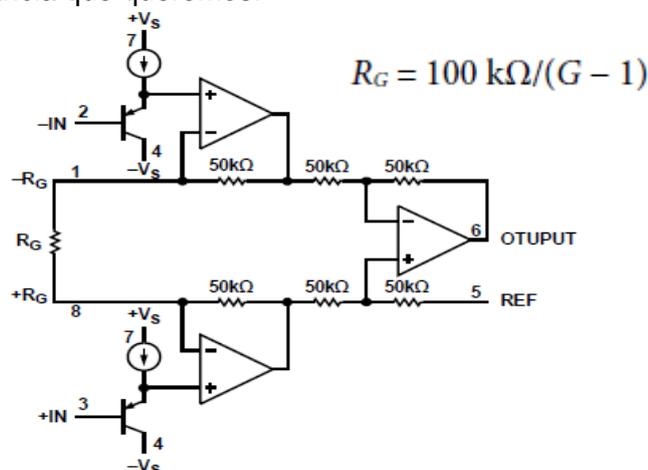


Figura 3-13. Esquema del amplificador de instrumentación

(Fuente: <http://www.analog.com/media/en/technical-documentation/data-sheets/AD623.pdf>)

Por lo tanto para una ganancia de 10 obtenemos:

$$R_G = \frac{100}{10 - 1} = \frac{100}{9} = 11,11 \text{ k}\Omega$$

Aproximamos este valor para obtener uno comercial a 10 kΩ. Nos quedaría por tanto una ganancia final de:

$$G = 1 + \frac{100}{R_G} = 1 + \frac{100}{10} = 11$$

La tensión final sería entonces:

$$V = 0,360 \cdot 11 = 3,96 \text{ V}$$

Que es menor que 5 voltios. Además, este rango de amplificación permite aprovechar casi la totalidad del rango de medida del microcontrolador sin llegar en ningún caso a saturar.

3.2.3 Diseño del filtro

El filtro que se ha decidido construir después del amplificador es un filtro Sallen-Key (de segundo orden) Butterworth ya que la respuesta es prácticamente plana desde 0 hasta la frecuencia de corte, y desciende bruscamente para frecuencias por encima de las de corte (Bruce y Mancini 2001):

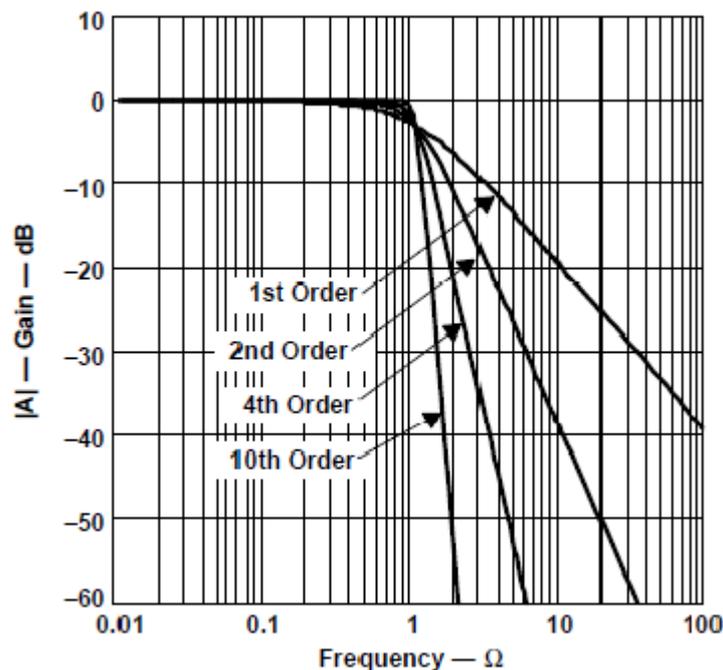


Figura 3-14. Gráfica de la ganancia del filtro Butterworth para distintos órdenes

(Fuente: <https://www.changpuak.ch/electronics/downloads/sloa088.pdf>)

El filtro Sallen-Key tiene el siguiente esquema:

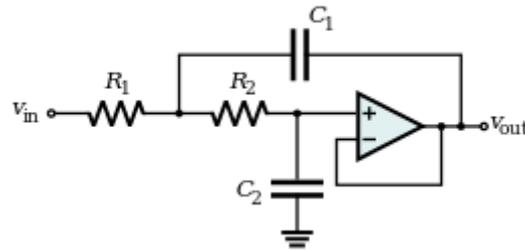


Figura 3-15. Esquema de un filtro Sallen-Key

(Fuente: <https://www.i-ciencias.com/pregunta/59881/como-reconocer-el-tipo-de-filtro-analogico-dado-su-topologia>)

El objetivo es obtener una función de transferencia del siguiente aspecto (Bruce y Mancini 2001):

$$H(s) = \frac{\omega_0^2}{s^2 + 2\alpha s + \omega_0^2} \quad (3.2)$$

Siendo:

$$\begin{aligned} \omega_0 &= 2 \cdot \pi \cdot f_0 \\ 2\alpha &= 2\xi\omega_0 = \frac{\omega_0}{Q} \end{aligned}$$

Siendo f_0 la frecuencia de resonancia y Q el factor de calidad que, como se puede deducir de la expresión superior, es igual a:

$$Q = \frac{1}{2 \cdot \xi} \quad (3.3)$$

Siendo ξ el amortiguamiento del sistema.

Se puede deducir de esta expresión que a mayor factor de calidad menor amortiguamiento y viceversa.

Una vez explicado todo lo anterior se procederá a la demostración y cálculo del filtro. Partiendo del esquema en la Figura 3-16 se obtienen las ecuaciones en (3.4):

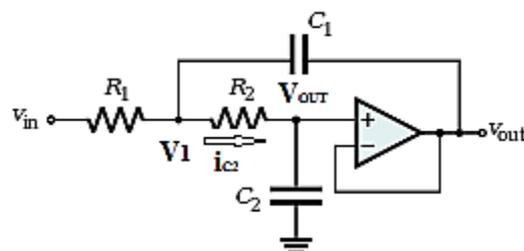


Figura 3-16. Esquema de un filtro Sallen-Key con flujo de corriente

(Fuente: elaboración propia a partir del esquema base de <https://www.i-ciencias.com/pregunta/59881/como-reconocer-el-tipo-de-filtro-analogico-dado-su-topologia>)

$$\begin{aligned}
 V_1 &= i_{C2} \cdot R_2 + V_{OUT} \\
 V_{OUT} &= i_{C2} \cdot \frac{1}{C_2 \cdot s} \\
 i_{C2} &= V_{OUT} \cdot C_2 \cdot s \\
 V_1 &= V_{OUT} \cdot C_2 \cdot s \cdot R_2 + V_{OUT} \\
 V_1 &= V_{OUT} \cdot (R_2 \cdot C_2 \cdot s + 1)
 \end{aligned}
 \tag{3.4}$$

Haciendo suma de corrientes en V_1 :

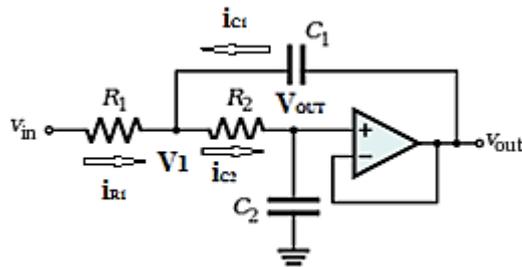


Figura 3-17. Esquema explicativo del flujo de corrientes en el filtro

(Fuente: elaboración propia a partir del esquema base de <https://www.i-ciencias.com/pregunta/59881/como-reconocer-el-tipo-de-filtro-analogico-dado-su-topologia>)

$$\begin{aligned}
 i_{C2} &= i_{R1} + i_{C1} \\
 i_{C2} &= \frac{V_{OUT}}{1} \\
 &\quad \frac{1}{s \cdot C_2} \\
 i_{R1} &= \frac{V_{in} - V_1}{R_1} \\
 i_{C1} &= \frac{V_{OUT} - V_1}{1} \\
 &\quad \frac{1}{s \cdot C_1} \\
 \frac{V_{OUT}}{1} &= \frac{V_{in} - V_1}{R_1} + \frac{V_{OUT} - V_1}{1} \\
 &\quad \frac{1}{s \cdot C_2} \quad \frac{1}{s \cdot C_1} \\
 R_1 \cdot s \cdot C_2 \cdot V_{OUT} &= V_{in} - V_1 + R_1 \cdot s \cdot C_1 (V_{OUT} - V_1)
 \end{aligned}
 \tag{3.5}$$

Sustituyendo V_1 por el valor obtenido en (3.4):

$$R_1 \cdot s \cdot C_2 \cdot V_{OUT} = V_{in} - V_{OUT} \cdot (s \cdot R_2 \cdot C_2 + 1) + R_1 \cdot s \cdot C_1 V_{OUT} - s \cdot R_1 \cdot C_1 \cdot V_{OUT} \cdot (s \cdot R_2 \cdot C_2 + 1) \tag{3.6}$$

Despejando V_{OUT} :

$$\begin{aligned}
s \cdot R_1 \cdot C_2 \cdot V_{OUT} \cdot (R_2 \cdot C_2 \cdot s + 1) - s \cdot R_1 \cdot C_1 \cdot V_{OUT} + s \cdot R_1 \cdot C_1 \cdot (R_2 \cdot C_2 \cdot s + 1) \cdot V_{OUT} &= V_{in} \\
V_{OUT} \cdot [R_1 \cdot C_2 \cdot s + R_2 \cdot C_2 \cdot s + 1 - s \cdot R_1 \cdot C_1 + s \cdot R_1 \cdot C_1 \cdot (R_2 \cdot C_2 \cdot s + 1)] &= V_{in} \\
V_{OUT} \cdot (R_1 \cdot C_2 \cdot s + R_2 \cdot C_2 \cdot s + 1 - s \cdot R_1 \cdot C_1 + s^2 \cdot R_1 \cdot R_2 \cdot C_1 \cdot C_2 + s \cdot R_1 \cdot C_1) &= V_{in} \quad (3.7) \\
V_{OUT} \cdot [s^2 \cdot R_1 \cdot R_2 \cdot C_1 \cdot C_2 + s \cdot (R_1 \cdot C_2 + R_2 \cdot C_2) + 1] &= V_{in} \\
\frac{V_{OUT}}{V_{in}} = H(s) = G &= \frac{1}{s^2 \cdot R_1 \cdot R_2 \cdot C_1 \cdot C_2 + s \cdot (R_1 \cdot C_2 + R_2 \cdot C_2) + 1}
\end{aligned}$$

Dividiendo arriba y abajo por $\frac{1}{R_1 \cdot R_2 \cdot C_1 \cdot C_2}$:

$$G = \frac{1}{s^2 + s \cdot \left(\frac{1}{R_2 \cdot C_1} + \frac{1}{R_1 \cdot C_1} \right) + \frac{1}{R_1 \cdot R_2 \cdot C_1 \cdot C_2}} \quad (3.8)$$

Comparando esta ecuación con la original en (3.2) e igualando términos obtenemos:

$$\begin{aligned}
\omega_0^2 &= \frac{1}{R_1 \cdot R_2 \cdot C_1 \cdot C_2} \\
\omega_0 &= \frac{1}{\sqrt{R_1 \cdot R_2 \cdot C_1 \cdot C_2}} \\
\omega_0 &= 2 \cdot \pi \cdot f_c \quad (3.9) \\
f_c &= \frac{1}{2 \cdot \pi \cdot \sqrt{R_1 \cdot R_2 \cdot C_1 \cdot C_2}} \\
2\alpha = \frac{\omega_0}{Q} &= \frac{1}{R_2 \cdot C_1} + \frac{1}{R_1 \cdot C_1} = \frac{R_1 + R_2}{R_1 \cdot R_2 \cdot C_1}
\end{aligned}$$

Por lo tanto Q será igual a:

$$Q = \frac{\omega_0 \cdot R_1 \cdot R_2 \cdot C_1}{(R_1 + R_2)} \quad (3.10)$$

Sustituyendo el valor de ω_0 :

$$Q = \frac{R_1 \cdot R_2 \cdot C_1}{(R_1 + R_2) \cdot \sqrt{R_1 \cdot R_2 \cdot C_1 \cdot C_2}} \quad (3.11)$$

Multiplicando arriba y abajo por $C_2 \cdot \sqrt{R_1 \cdot R_2 \cdot C_1 \cdot C_2}$ nos queda:

$$Q = \frac{\sqrt{R_1 \cdot R_2 \cdot C_1 \cdot C_2}}{C_2 \cdot (R_1 + R_2)} \quad (3.12)$$

Para el diseño de nuestro filtro realizamos la siguiente simplificación:

$$\begin{aligned} R_1 &= m \cdot R \\ R_2 &= R \\ C_1 &= n \cdot C \\ C_2 &= C \end{aligned} \quad (3.13)$$

Siendo m y n unas constantes elegidas por nosotros.

Nos quedarán por tanto las siguientes expresiones:

$$\begin{aligned} f_c &= \frac{1}{2 \cdot \pi \cdot R \cdot C \cdot \sqrt{m \cdot n}} \\ Q &= \frac{\sqrt{m \cdot n}}{m + 1} \end{aligned} \quad (3.14)$$

Elegimos los valores de m 1 y n 2 para que la Q se igual a:

$$Q = \frac{\sqrt{2}}{2} = \frac{1}{\sqrt{2}}$$

ya que tiene que tener este valor para ser un filtro Butterworth y tener así el tipo de respuesta descrita al principio de este apartado.

Los condensadores que teníamos disponibles eran de 330 nF por lo que emplearemos este valor para C y el valor de la frecuencia de corte requerida es 700 Hz como se explicó en el apartado 3.2.1. Por lo tanto despejando el valor de R de la expresión anterior nos queda:

$$R = \frac{1}{2 \cdot \pi \cdot 700 \cdot 330 \cdot 10^{-9} \cdot \sqrt{2}} = 487,18 \Omega$$

Se aproxima este valor a 488 Ω , que se obtiene sumando dos resistencias en serie, una de 470 Ω y otra de 18 Ω . Quedará por tanto una frecuencia de corte de 698,83 Hz, valor muy aproximado a los 700 Hz que buscamos, y dado que es incluso un valor menor, está en el lado de la seguridad, lo que asegura que no haya efectos de aliasing de frecuencias superiores.

Finalmente nuestro filtro quedará:

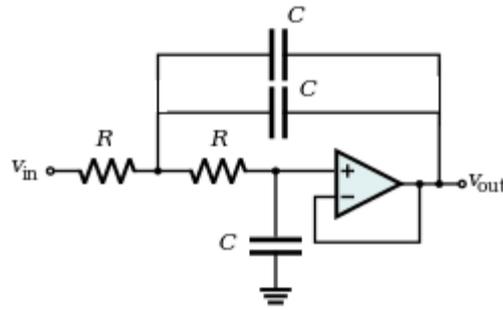


Figura 3-18. Esquema del filtro a construir

(Fuente: elaboración propia a partir del esquema base de <https://www.i-ciencias.com/pregunta/59881/como-reconocer-el-tipo-de-filtro-analogico-dado-su-topologia>)

Con C igual a 330 nF y R igual a 488 Ω .

La señal amplificada y filtrada es digitalizada por el Arduino, que la representa con un valor entre 0 y 1023. Falta pues relacionar este valor con la fuerza de tiro que sufre la cadena.

3.2.4 Calibración experimental

Para poder usar la corriente del motor para medir la fuerza de tiro, primero habrá que determinar la relación que existe entre dicha intensidad y el par que ejerce el motor, que es el que se transmite a la cadena. La ecuación del motor eléctrico (Wildi 2005) que muestra la relación entre el par y la intensidad puede verse en (3.15):

$$\tau = k_t I \quad (3.15)$$

Siendo

k_t una constante del motor eléctrico

I la intensidad que pasa a través del motor

τ es el par que ejerce el motor

El método experimental empleado para sacar la recta que relaciona el par y la tensión es el siguiente: se coloca el peso que se le va a aplicar al motor de forma perpendicular a este, con el motor apoyado en una mesa, para así obtener un valor puro de par de fuerza (obtenido al multiplicar el peso del objeto por el radio de la polea que lleva el motor) y se mide la intensidad que pasa a través del motor. Se eligieron los siguientes tres pesos para la calibración:

- 65 gramos
- 101 gramos
- 123 gramos

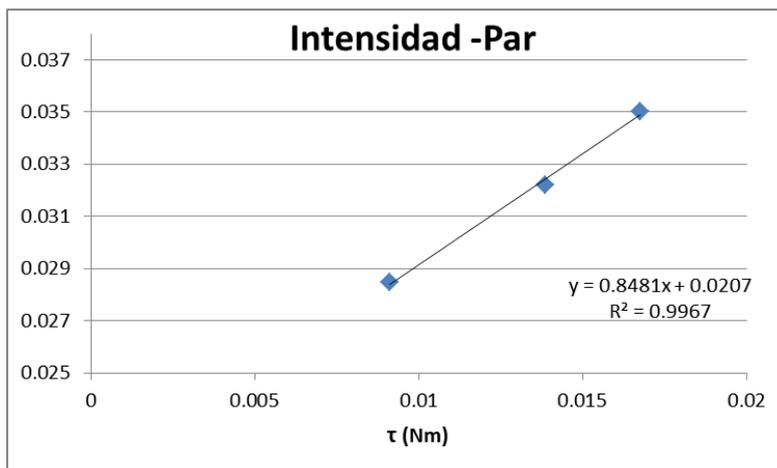
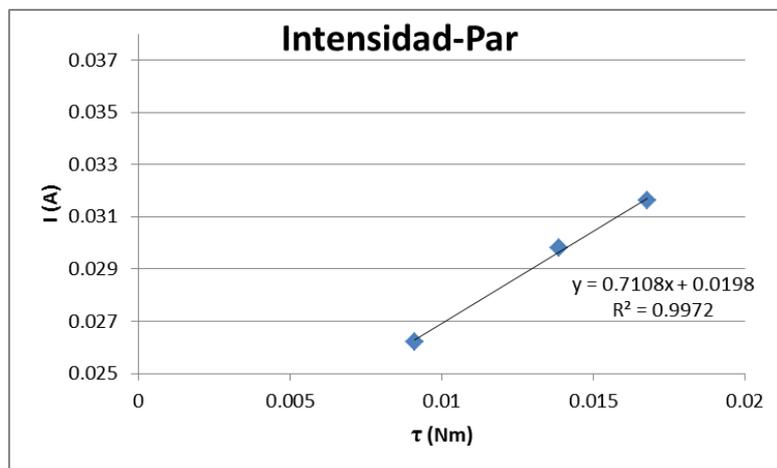
porque cubren todo el rango de fuerzas dinámicas que se espera ocurran durante la maniobra y la ecuación no va a ser del todo una recta por efecto de la fuerza de rozamiento explicada anteriormente. El valor de par se obtendrá multiplicando la masa aplicada en el experimento por el radio de la polea (13.2 mm) y la aceleración de la gravedad.

El experimento se va a realizar a velocidad constante de 1 rad/s para así tener en cuenta las posibles variaciones de la fuerza de rozamiento con la velocidad ya que, aunque

en la ecuación (3.15) no aparece la velocidad, en la realidad el rozamiento en los engranajes y otros elementos de transmisión del motor sí se ven afectados por la velocidad. Al medir a esta velocidad, que es la misma que se empleará en la simulación, ya se tiene en cuenta estas variaciones cuando se calcule el valor de k_t , que ahora no sólo tendrá en cuenta las características del motor eléctrico, sino también la de rozamiento de los elementos de transmisión.

Para mantener la velocidad del experimento constante durante la maniobra, se emplea un controlador que monitoriza la velocidad de giro del motor a través del encoder. Dicho encoder proporciona valores de posición angular y a partir de esos valores se obtiene la velocidad angular por derivación. Sin embargo, la velocidad angular así obtenida resulta ser bastante ruidosa debido a la combinación de la baja velocidad del experimento y la poca resolución del encoder, con lo que la intensidad (y la fuerza de tiro) también presentan fluctuaciones. Este problema ha sido paliado promediando 50 valores de la parte central del experimento. Además este experimento se ha repetido un total de 5 veces en días distintos, y también se ha obtenido el resultado promedio de todos los experimentos.

A continuación se muestran las ecuaciones intensidad-par obtenidos en los distintos días:



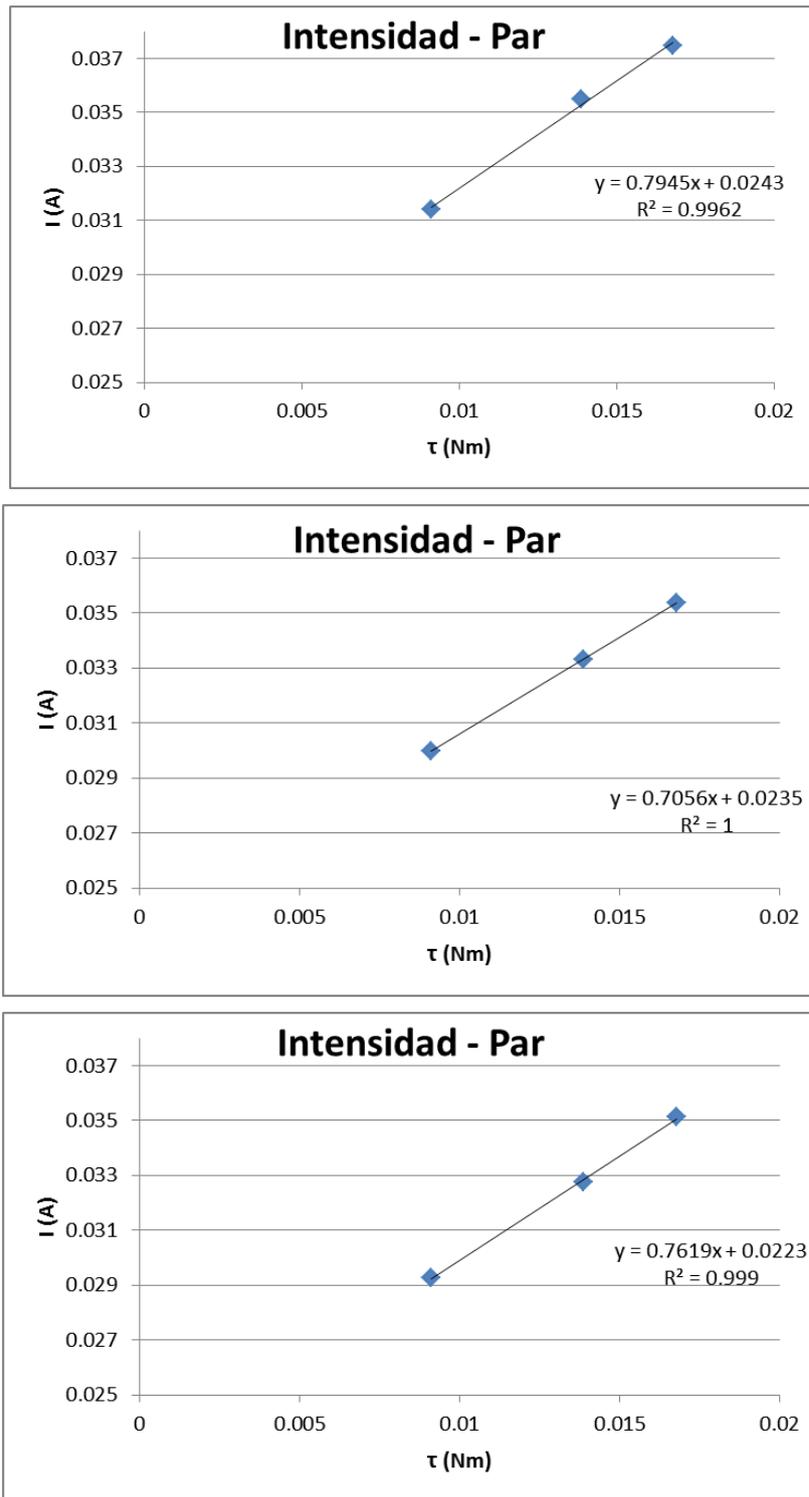


Figura 3-19. Resultados obtenidos en los distintos días

Por último con los cinco valores para cada peso de los distintos días, se realizó una media final y se obtuvo así un valor final para la relación entre par e intensidad. Esto puede verse en la siguiente imagen:

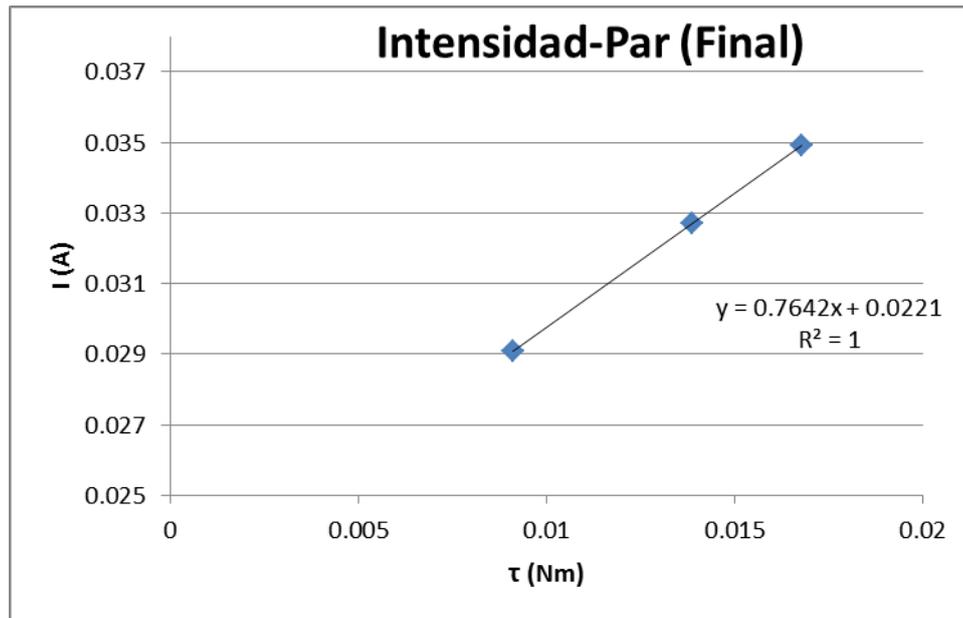


Figura 3-20. Resultados finales

El procesado del valor del Arduino para obtener los anteriores resultados es como sigue: del Arduino llega un valor que puede ir del 0 a 1023. Este valor se transforma en uno de tensión real dándole al 0 el valor de 0 V y al 1023 el de 5 V (la tensión de referencia mencionada antes). Este valor de tensión se convierte en uno de corriente mediante la ley de Ohm:

$$V = I \cdot R$$

$$I = \frac{V}{R} \quad (3.16)$$

Siendo R el valor de la resistencia calculado en el apartado 3.2.2.1. Por último este valor se divide por la ganancia del amplificador calculada en 3.2.2. Esto nos lleva al valor de intensidad que se ve representado en el eje y. Para los valores del par, representados en el eje x, se multiplica el valor de la masa (en kg) por el radio de la polea (en metros) y por la aceleración de la gravedad.

De la ecuación obtenida en la Figura 3-20

$$y = 0.7642x + 0,0221 \quad (3.17)$$

La x representa el par y la y la intensidad. Para más comodidad, ya que los valores que vamos a obtener son intensidades, despejaremos la x. También cambiaremos el valor de x por el de par (τ) y el de y por el de intensidad (I)

$$\tau = \frac{I - 0.0221}{0.7642} \quad (3.18)$$

3.3 Actuación

En este apartado se explicará los componentes empleados para la actuación y control del experimento. Se ha elegido la utilización del puente H L298N (Sparkfun Electronics 2017), cuyo esquema se puede ver en la Figura 3-21, ya que permite invertir el sentido de giro de motor y aislar la parte de potencia (la fuente de alimentación del motor) de la de control (las señales PWM enviadas por el Arduino. Este tipo de señal se explica en el apartado 3.4):

PIN CONNECTIONS (top view)

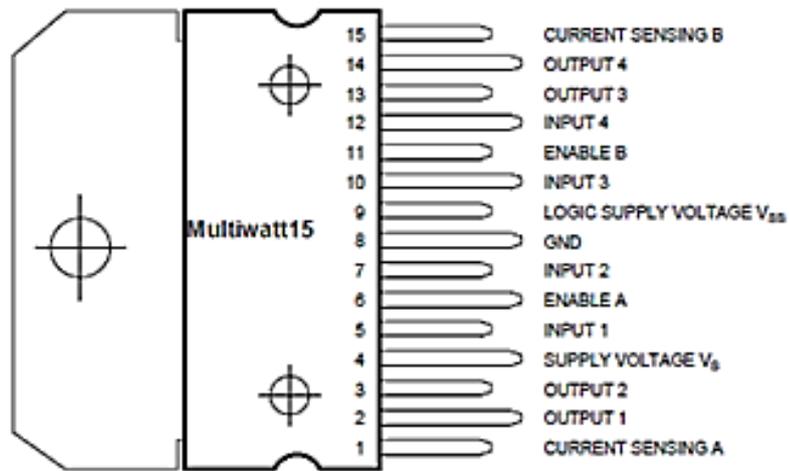


Figura 3-21. Esquema puente H L298N

(Fuente: https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf)

Este componente incluye dos puentes H, pero en este caso sólo se emplea uno, por lo que los pines del 10 al 15, que corresponden al segundo puente H, no se utilizan.

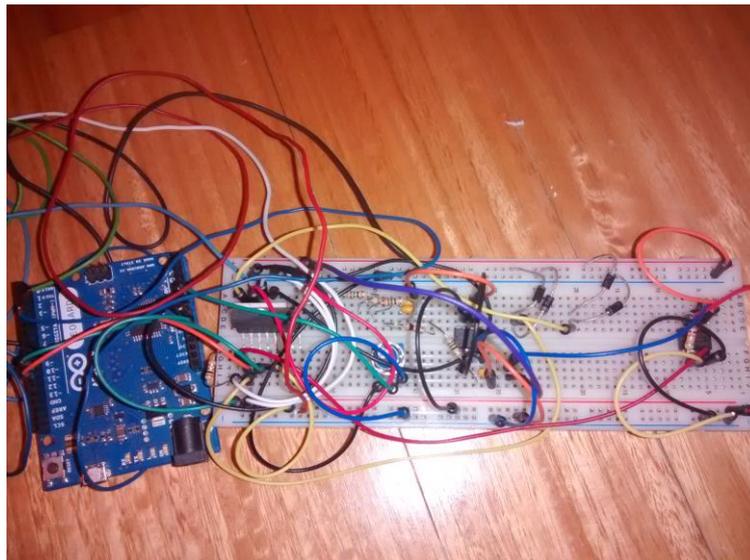


Figura 3-22. Montaje real

Por el pin 1 de este elemento, llamado "Current Sensing A" circula toda la corriente que pasa por el motor, por lo que es el lugar adecuado para colocar la resistencia para la medida de la intensidad del motor, como se ha explicado en el apartado 3.2.2.1.

El pin 2 y el pin 3 son las salidas del puente H a las que se conectarán los bornes del motor.

La alimentación del motor se conecta al pin 4 que en nuestro caso será una batería de pilas en serie de 6 voltios.

Los pines 5 y 7 son las entradas del puente H. Con estas dos señales se controlará tanto la velocidad como la dirección del motor.

El pin 6 se conecta a los 5 V del Arduino, ya que sin él no puede activarse el puente H.

El pin 8 es la tierra que se conecta borne negativo de la batería.

Por último, el pin 9 alimenta la circuitería lógica del puente H L298N.

BLOCK DIAGRAM

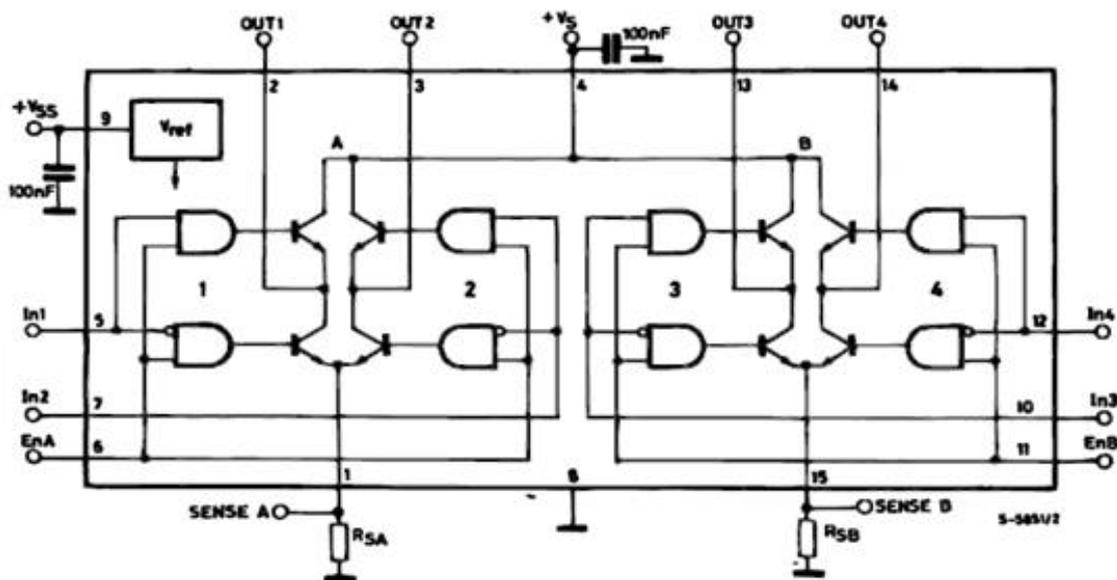


Figura 3-23. Esquema de funcionamiento del puente H

(Fuente: https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf)

3.3.1 Funcionamiento del puente H

En la Figura 3-23 se puede ver cómo está conectado internamente el L298N. Para entender su funcionamiento vamos a explicar primero el significado de unos símbolos:

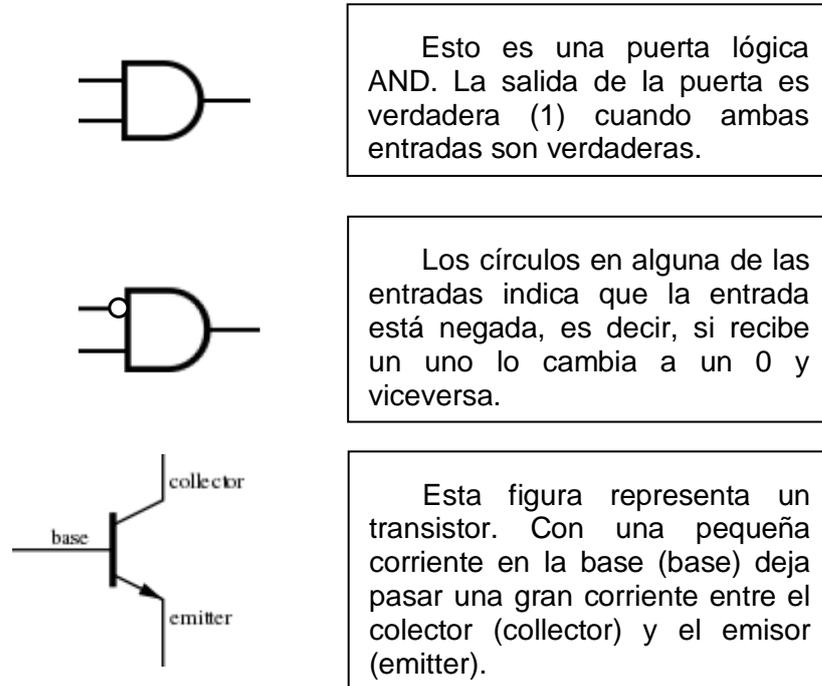


Figura 3-24. Explicación de los símbolos del esquema

(Fuente: <https://sub.allaboutcircuits.com/images/04114.png>, https://en.wikipedia.org/wiki/AND_gate y <https://en.wikipedia.org/wiki/Transistor>)

A continuación, explicamos el funcionamiento del puente H con una serie de ejemplos para facilitar la comprensión del mismo.

Supongamos que I_{n1} vale 1 (valor lógico positivo, 5V), I_{n2} vale 0 (valor lógico negativo, 0V) y E_{nA} vale 1 (valor lógico positivo, 5V). Los valores de I_{n1} los pondremos en verde, los valores de I_{n2} en azul y los valores de E_{nA} en rojo. Entonces:

BLOCK DIAGRAM

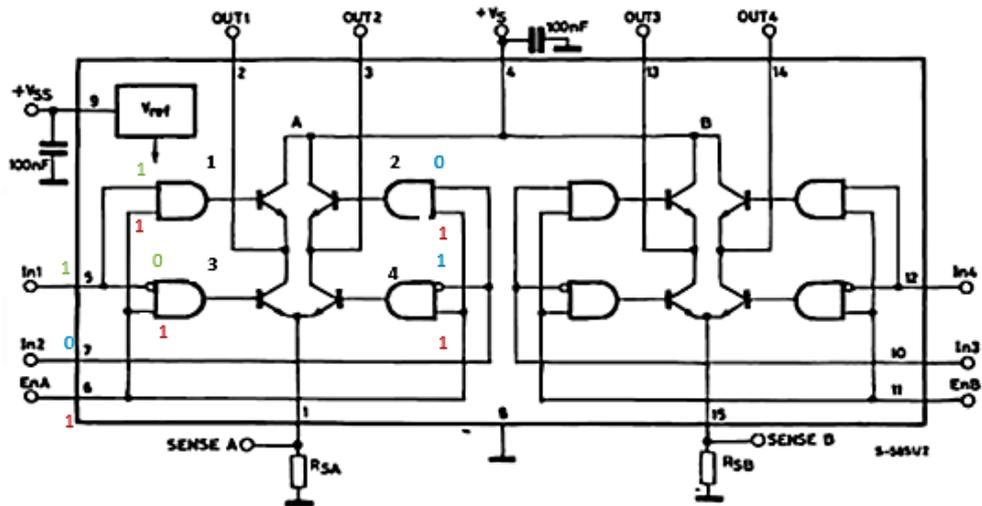


Figura 3-25. Ejemplo 1 del funcionamiento del puente

(Fuente: elaboración propia con esquema base extraído de https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf)

Se puede observar en el esquema superior que con estos valores que la puerta AND “1” y “4” estarían abiertas y dejarían pasar voltaje a los transistores para que estos dejaran pasar a su vez corriente entre el emisor y el colector.

Se aprovecha para explicar aquí que según la tensión que pase por la base pasa más o menos corriente entre el emisor y el colector, hasta un máximo corriente limitado por los 6 voltios (alimentación del motor) y la resistencia de la carga. De esta forma es cómo se consigue regular la velocidad del motor.

En estas circunstancias, la corriente que activa el motor entra en el puente H a través del pin 4, sale hacia el motor a través del pin OUT1, que en esta configuración es el pin positivo del motor, vuelve al puente H a través del pin OUT2, y se va a tierra a través del pin 1. En el pin 1 es donde está colocada la resistencia calculada en el apartado 3.2.2.1.

Si se quiere invertir el sentido de funcionamiento del motor, se cambia la configuración de la siguiente manera: se ajusta el valor de In1 a 0, e In2 a 1, obteniendo el resultado que se muestra en la siguiente figura:

BLOCK DIAGRAM

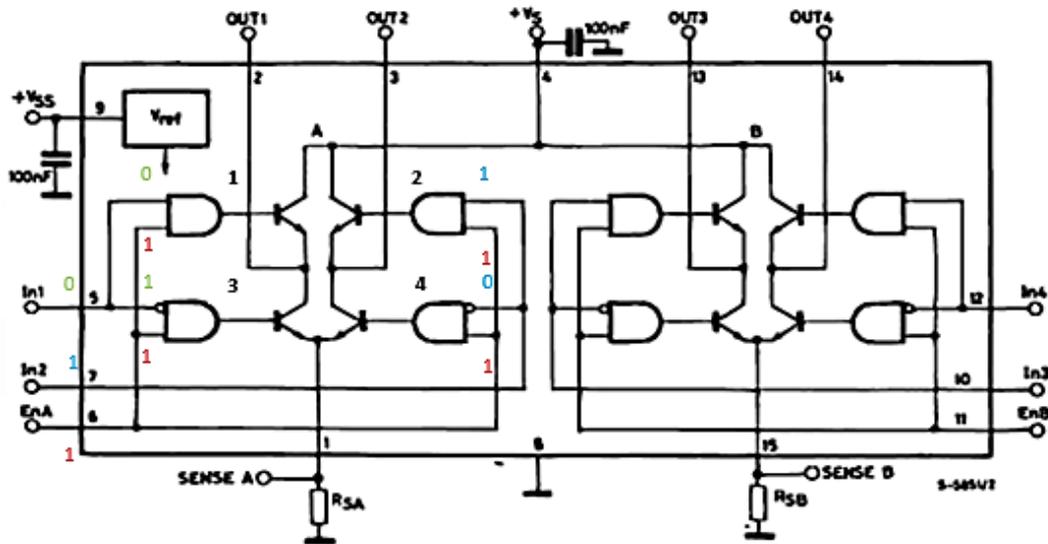


Figura 3-26. Ejemplo 2 del funcionamiento del puente H

(Fuente: elaboración propia con esquema base extraído de https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf)

Se ve como en este caso las puertas lógicas AND que están abiertas son la “2” y la “3”. En este caso por lo tanto entraría la corriente al circuito por 4, entra al motor por OUT2 y sale del motor por OUT1, al contrario que en la configuración anterior. De este modo se consigue invertir el sentido de giro del motor.

Estas dos son las configuraciones que hemos usado, y en ambas se requiere que el valor del pin EnA tenga un valor lógico de 1, por lo que siempre se deja conectado a 5V.

Cogiendo e ejemplo anterior y cambiando el valor de EnA a “0”.

BLOCK DIAGRAM

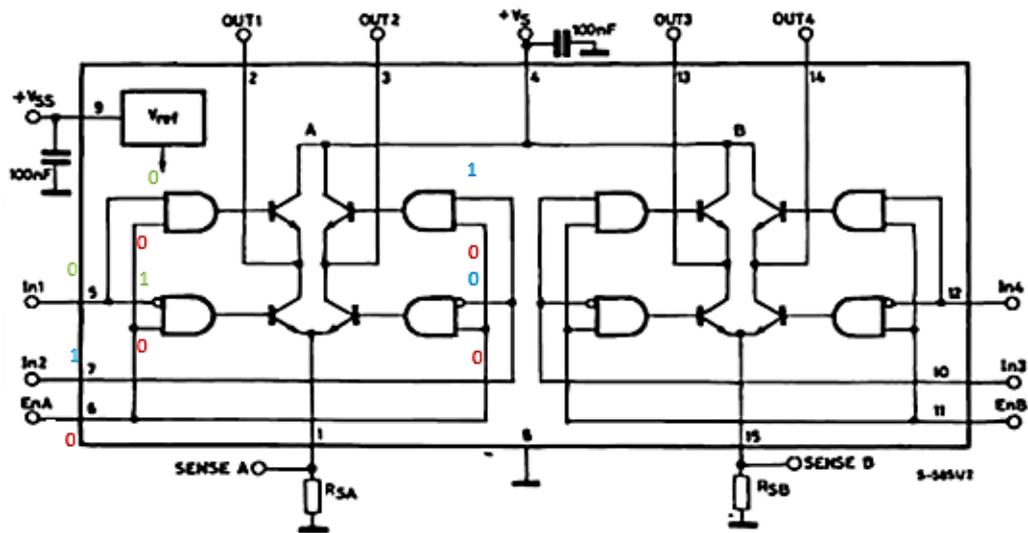


Figura 3-27. Ejemplo 3 del funcionamiento del puente H

(Fuente: elaboración propia con esauema base extraído de https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf)

Como se puede observar la condición de que las dos entradas sean "1" no se cumple en ningún caso ya que EnA siempre es "0". Por eso la necesidad que tenga un valor lógico positivo (la entrada esté a "1").

En la Figura 3-28 se muestra como irían conectados los diodos con respecto al motor:

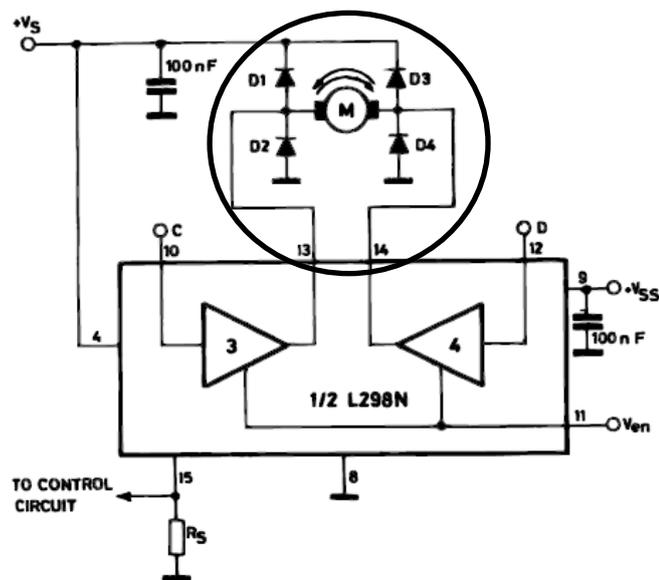


Figura 3-28. Esquema de conexión de los diodos

(Fuente: elaboración propia con esquema base extraído de https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf)

Estos diodos colocados en antiparalelo con los transistores son necesarios siempre que el puente H alimenta una carga inductiva (como el caso de un motor eléctrico). La función de estos diodos es permitir el paso de la corriente en sentido contrario al habitual cada vez que se corta la corriente de alimentación al motor, de forma que se evita dañar los transistores (Galeano, Quintero y Lezama 2016). El funcionamiento se explica en la siguiente figura.

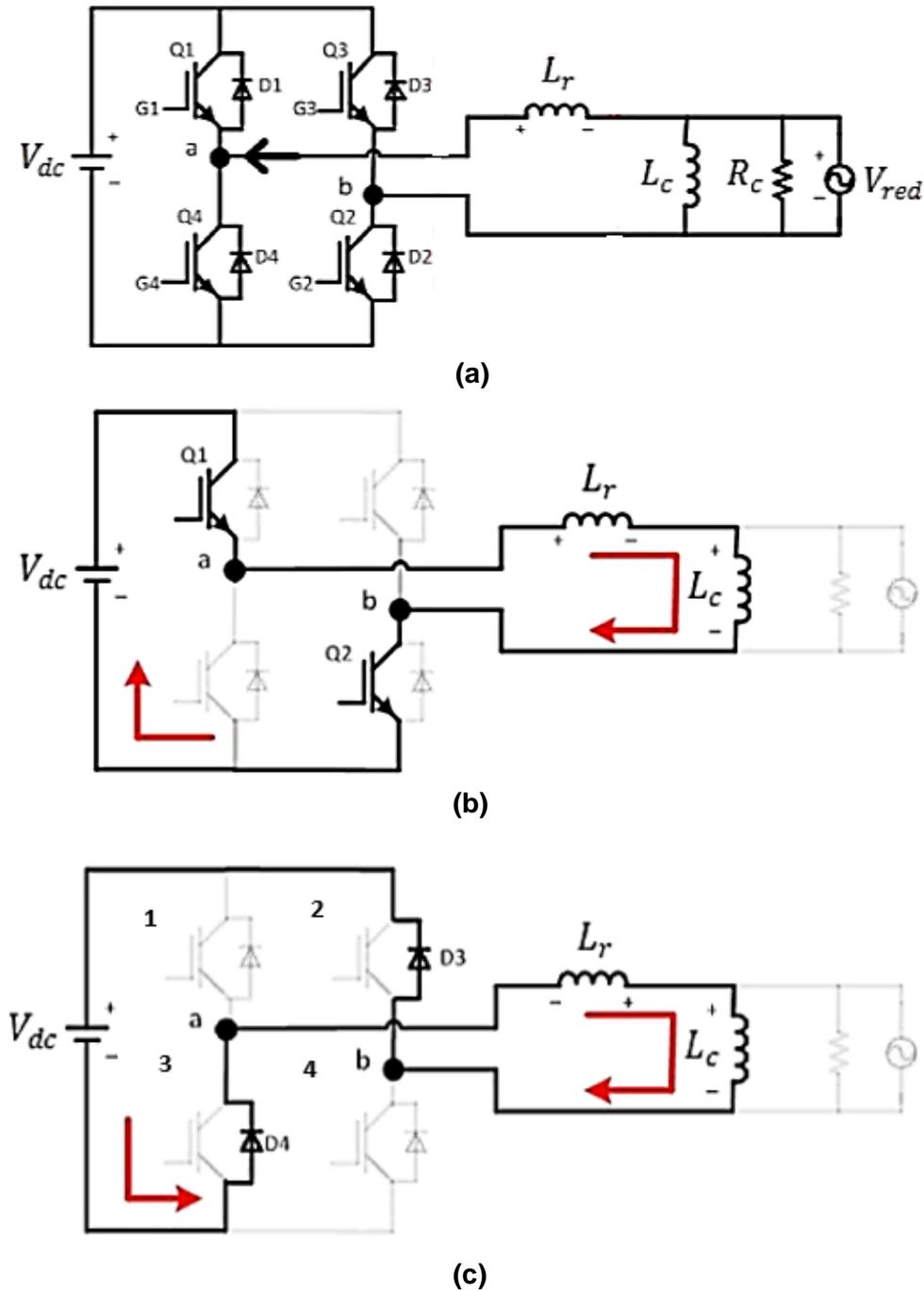


Figura 3-29. Ejemplo para la justificación de los diodos

(Fuente: <http://www.scielo.cl/pdf/formuniv/v9n1/art13.pdf>)

En la Figura 3-29(b) se observa cómo se conectan los transistores superior izquierdo e inferior derecho y la corriente va del punto a al b pasando por estos. Cuando la fuente de alimentación para de suministrar corriente en la Figura 3-29(c) y la bobina devuelve parte de la corriente a la batería esta pasa por los diodos, quedando los transistores protegidos.

3.4 Control del experimento

En este apartado se explica el algoritmo programado en el Arduino para controlar el experimento. Este control se realiza por medio de una señal PWM. Las siglas PWM significan modulación por ancho de pulso (pulse-width modulation para sus siglas en inglés) y lo que permite es emular una señal analógica a partir de una señal digital (que solo tiene la posibilidad de tomar su valor máximo o mínimo). Esta modulación se consigue variando el llamado ciclo de trabajo (el porcentaje del tiempo que la señal permanece en su valor alto en cada período) (Arduino 2017).

Si la frecuencia de esta señal es alta en comparación con el tiempo de respuesta del sistema que controla, se puede conseguir un control tan fino como con una señal analógica. El sistema de modulación por ancho de pulso (PWM) presenta una ventaja adicional, ya que hace trabajar a los transistores del puente H en corte y en saturación, minimizando el tiempo que están en la zona activa. La ventaja de esto está en que los transistores disipan mucha menos energía tanto en la zona de corte como en la de saturación, mientras que en la zona activa disipan mucha más energía, produciendo pérdidas de rendimiento y calentamiento adicional del puente H, lo que exigiría emplear elementos diseñados para más potencia.

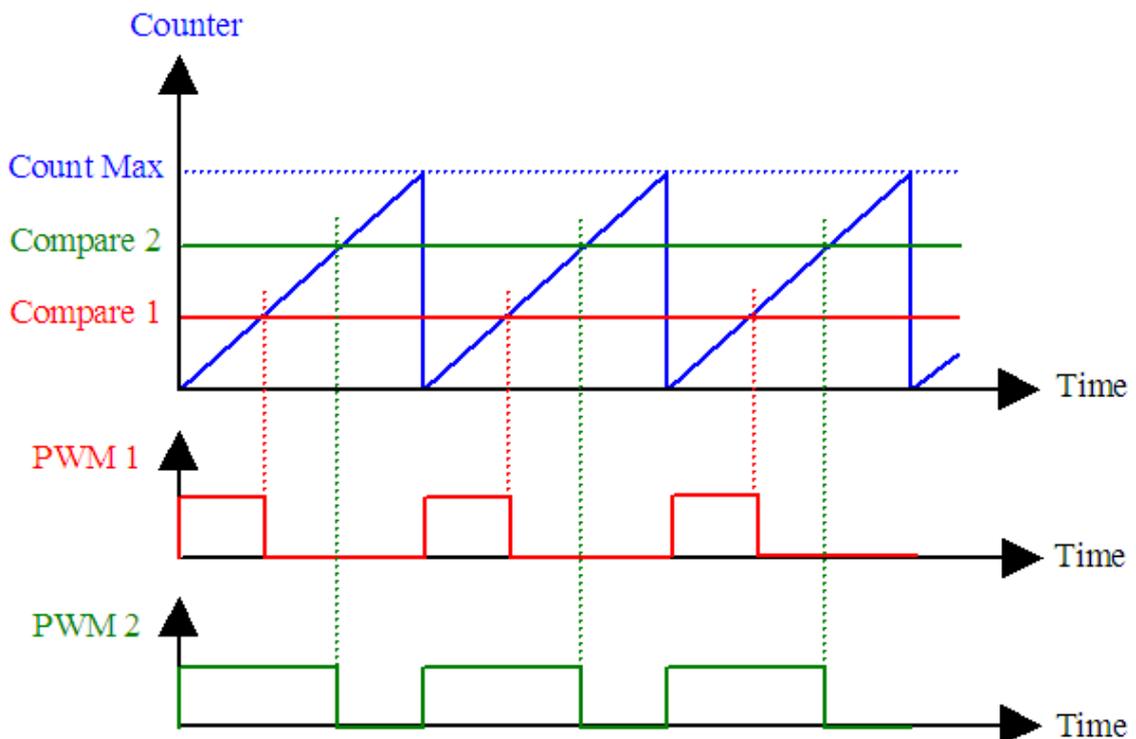


Figura 3-30. Funcionamiento del PWM

(Fuente: <http://r6500.blogspot.com/2014/12/fast-pwm-on-arduino-leonardo.html>)

El ciclo de carga se controla comparando el valor de un contador que se incrementa con el tiempo (señal azul) con el valor del ciclo de carga deseado. Cuando el contador está por

debajo del valor prefijado (señal roja/verde), la señal PWM está en su fase activa y los transistores dejan pasar la corriente de alimentación hacia el motor, mientras que cuando se supera dicho valor, la señal del PWM se pone a 0 V y los transistores se cierran. Cuanto mayor es el ciclo de carga fijado, más porcentaje de tiempo permanece la señal en su estado alto.

Así pues, lo que hace el código es, según el valor de velocidad de giro del motor medida, calcular un valor de PWM que regule la velocidad real para que coincida con la deseada.

Ver Anexo II: Código Arduino para ver el código implementado.

3.4.1 Comunicación con el ordenador

El Arduino envía los datos por USB a través de un puerto serie virtual, y luego se recuperan en el PC con un script de Matlab.

```

arduino=serial('COM8','BaudRate',115200);           % con esta funcion definimos que puerto tiene que leer y a que tasa

fopen(arduino);                                     % abrimos la conexión que definimos antes
data=ones(40000,15);                                % creamos las matrices para guardar los datos
DATA=char(data);
x=size(DATA);
x=x(1,1);                                           % creamos los índices para movernos por las matrices
i=1;
j=1;
while i <= x                                       % en este bucle vamos guardando los datos que leemos del arduino en la matriz DATA
    a= fscanf(arduino,'%s');
    z= size(a);
    z= z(2);
    while j <= z
        DATA(i,j)= a(j);
        j=j+1;
    end
    i=i+1;
    j=1;
    clear a;
end

fclose(arduino);                                   % cerramos la conexión del arduino

```

Figura 3-31. Código Matlab

Capítulo 4
Modelización del
sistema

4 MODELIZACIÓN DEL SISTEMA

Un sistema multicuerpo o multibody system es la modelización física del conjunto de sólidos de un mecanismo y las relaciones y restricciones que existen entre ellos. A través de los sistemas multicuerpo podemos estudiar la cinemática, la dinámica inversa o la dinámica directa del mecanismo. En el caso del programa Simulancla es un problema de dinámica directa, que permite conocer la posición, velocidad y aceleración del sistema conocidas las propiedades de los sólidos y las fuerzas que actúan sobre el sistema.

El primer paso es la modelización del sistema multicuerpo. Para la aplicación de la modelización multicuerpo es precisa la parametrización de las posiciones de los sólidos mediante unas coordenadas apropiadas y la definición de las ecuaciones de restricción que ligan estas coordenadas.

El siguiente paso es la resolución de la cinemática del sistema en la posición inicial, obteniendo unas posiciones y velocidades iniciales que cumplan las ecuaciones de restricción. A continuación se lleva a cabo un tercer paso es opcional: el planteamiento de las ecuaciones de la estática del sistema y la resolución de la posición de equilibrio inicial del mismo. Finalmente, el último paso es el planteamiento y resolución de las ecuaciones del movimiento o ecuaciones dinámicas del sistema en cada paso de tiempo. Las entradas al problema dinámico son las fuerzas actuantes sobre el sistema y dichas fuerzas son tremendamente complejas de obtener, resultado de la evaluación de complejos modelos de fuerzas.

A partir de la solución multicuerpo, será posible la obtención de todos aquellos resultados que nos permitan analizar la dinámica de la maniobra de fondeo.

4.1 Modelización elegida para el sistema de fondeo

En este caso el sistema de coordenadas usado es un sistema de coordenadas mixtas. Este sistema emplea dos tipos de coordenadas, las naturales y las relativas. Las primeras se emplearán para modelizar sólidos, y las segundas para actuadores (motores), definición de posiciones grados de libertad e introducción de ciertos tipos de fuerzas (Cuadrado 1999).

- **Las coordenadas naturales**

Las coordenadas naturales son básicamente las coordenadas cartesianas de puntos y vectores unitarios unidos a los sólidos del mecanismo. Esas coordenadas sitúan cada sólido del mecanismo. Para cada sólido 3D se definen un mínimo de 12 coordenadas en total, así pues, dado que el número de grados de libertad de un sólido libre es 6, estas coordenadas son dependientes por propia definición.

Aportan la ventaja de una definición sencilla y sistemática, con ecuaciones de restricción igualmente simples, ya que son lineales o cuadráticas, pero tienen el inconveniente de que, para obtener una modelización correcta, precisan de cierta experiencia.

- **Las coordenadas relativas**

Las coordenadas relativas se diferencian de las naturales en que, en lugar de referir los sólidos a un origen común a todos, se refieren cada uno de ellos a su antecesor en la cadena cinemática.

Cuando se modelizan sistemas de cadena abierta exclusivamente en coordenadas relativas, se obtiene una cinemática sin restricciones ya que el número de coordenadas relativas necesarias es igual al número de grados de libertad del sistema.

Cuando se modelizan sistemas de cadena cerrada exclusivamente con coordenadas relativas, éstas son redundantes, pero se obtiene un reducido número de ecuaciones de restricción; las necesarias para imponer las condiciones de cierre de lazo.

Cuando se emplean las coordenadas relativas conjuntamente con otros sistemas de coordenadas, éstas son especialmente apropiadas para la definición de grados de libertad, actuadores o fuerzas asociadas a pares cinemáticos. El primer inconveniente cuando únicamente estas coordenadas para definir el sistema, es la necesidad de situar todos los elementos anteriores de la cadena para poder conocer la posición de un sólido, por las razones anteriormente descritas.

El segundo es que las ecuaciones contienen términos trigonométricos bastante complejos, lo cual aumenta considerablemente el coste computacional de la resolución del problema.

- **Las coordenadas mixtas**

Como ya se indicó al principio de este apartado este sistema emplea dos tipos de coordenadas, las naturales y las relativas. Utilizar dos sistemas de coordenadas diferentes nos permite aprovechar las ventajas de ambos. Por un lado, se contará con la sencillez de las coordenadas naturales para la definición de las posiciones de los sólidos de forma independiente al resto de sólidos de la cadena cinemática, y por otro, se tendrá una facilidad de definición para los grados de libertad, actuadores y fuerzas gracias a las coordenadas relativas.

Se tendrá que trabajar, entonces, con los puntos y vectores de las coordenadas naturales y con los ángulos y distancias de las coordenadas relativas.

Una vez explicado las coordenadas que se van usar explicaremos como se emplearán en la modelización.

Para modelizar cualquier sólido se necesitan cuatro entidades (12 coordenadas generalizadas), ya sean puntos o vectores, definiendo siempre al menos un punto y al menos un vector. Esto permitirá representar el sólido en cuestión en un espacio tridimensional y orientarlo, ya que se construirá, para cada sólido, un sistema de referencia formado por un punto y tres vectores linealmente independientes.

Se puede demostrar que, definiendo los sólidos de esta manera, la matriz de masas y el vector de fuerzas generalizadas gravitatorias son constantes y no aparecerán en ellos términos de fuerzas de inercia dependientes de la velocidad o fuerzas dependientes de la velocidad respectivamente (Cuadrado 1999). Esto permite calcularla solamente en el instante inicial y prescindir de un cálculo continuo.

El software SIMULANCLA emplea la biblioteca de sistemas multicuerpo MBSLIM (Multibody Systems en Laboratorio de Ingeniería Mecánica) para llevar a cabo la tarea de modelización del sistema de fondeo, planteamiento y resolución de las ecuaciones del movimiento.

Simulancla emplea las siguientes restricciones entre las coordenadas naturales para definir un sólido rígido (Dopico, González, y otros 2014):

1. **Distancia:** para puntos se impone que la distancia entre dos puntos sea constante. En el caso de vectores, la norma del vector ha de ser la unidad. Es una sola ecuación.

$$(\mathbf{p}_i - \mathbf{p}_j)^T (\mathbf{p}_i - \mathbf{p}_j) - l_{ij}^2 = 0 \quad (4.1)$$

$$\mathbf{v}_i^T \mathbf{v}_i - 1 = 0 \quad (4.2)$$

2. **Producto escalar constante:** el ángulo entre dos vectores se mantendrá constante. Si el sólido se define con más de un punto se tomarán como vectores los que unen esos puntos. Es una sola ecuación.

$$\mathbf{v}_i^T \mathbf{v}_j - k_k = 0 \quad (4.3)$$

$$(\mathbf{p}_i - \mathbf{p}_j)^T \mathbf{v}_k - k_l = 0 \quad (4.4)$$

$$(\mathbf{p}_i - \mathbf{p}_j)^T (\mathbf{p}_i - \mathbf{p}_j) - k_m = 0 \quad (4.5)$$

3. **Combinación lineal:** los elementos de modelización del sólido forman un subespacio. Los puntos y vectores definidos a mayores para facilitar la definición del modelo pertenecerán a ese subespacio (serán linealmente dependientes) y además su posición en coordenadas locales permanecerá constante el tiempo, es decir los valores de los parámetros de combinación lineal serán constantes.

Con esta restricción ocurre lo mismo que con la de producto escalar, si se modeliza con más de un punto se tomará como vector el que los une.

Se trata de una ecuación vectorial lo que equivale a tres ecuaciones.

$$\mathbf{p}_i - \mathbf{p}_j - \alpha_{k1} \cdot \mathbf{v}_l - \alpha_{k2} \cdot \mathbf{v}_m - \alpha_{k3} \cdot \mathbf{v}_n = 0 \quad (4.6)$$

$$\mathbf{v}_i - \alpha_{k1} \cdot \mathbf{v}_l - \alpha_{k2} \cdot \mathbf{v}_m - \alpha_{k3} \cdot \mathbf{v}_n = 0 \quad (4.7)$$

Donde \mathbf{p}_i y \mathbf{v}_i son el punto o vector extra que no pertenece a la definición del sólido.

Estas restricciones son incluidas automáticamente cuando se modeliza un sólido en MBSLIM. Ahora resta añadir las restricciones externas dependientes del mecanismo.

Para definir los GDL utilizaremos coordenadas relativas, creando así las variables adicionales que nos permitirán el guiado.

4.1.1 Definición del buque

El suelo se define con el punto \mathbf{p}_1 y los vectores \mathbf{v}_1 , \mathbf{v}_2 y \mathbf{v}_3 . Esto forma un sistema de coordenadas fijo al suelo del muelle, así que no es necesario incluir estas entidades en el vector de variables, pero sí precisaremos de ellas para la modelización gráfica.

4.1.2 Definición de sólidos del sistema de fondeo

Eslabones de la cadena.

Se describirán a continuación las variables y restricciones para dos eslabones consecutivos de la cadena.

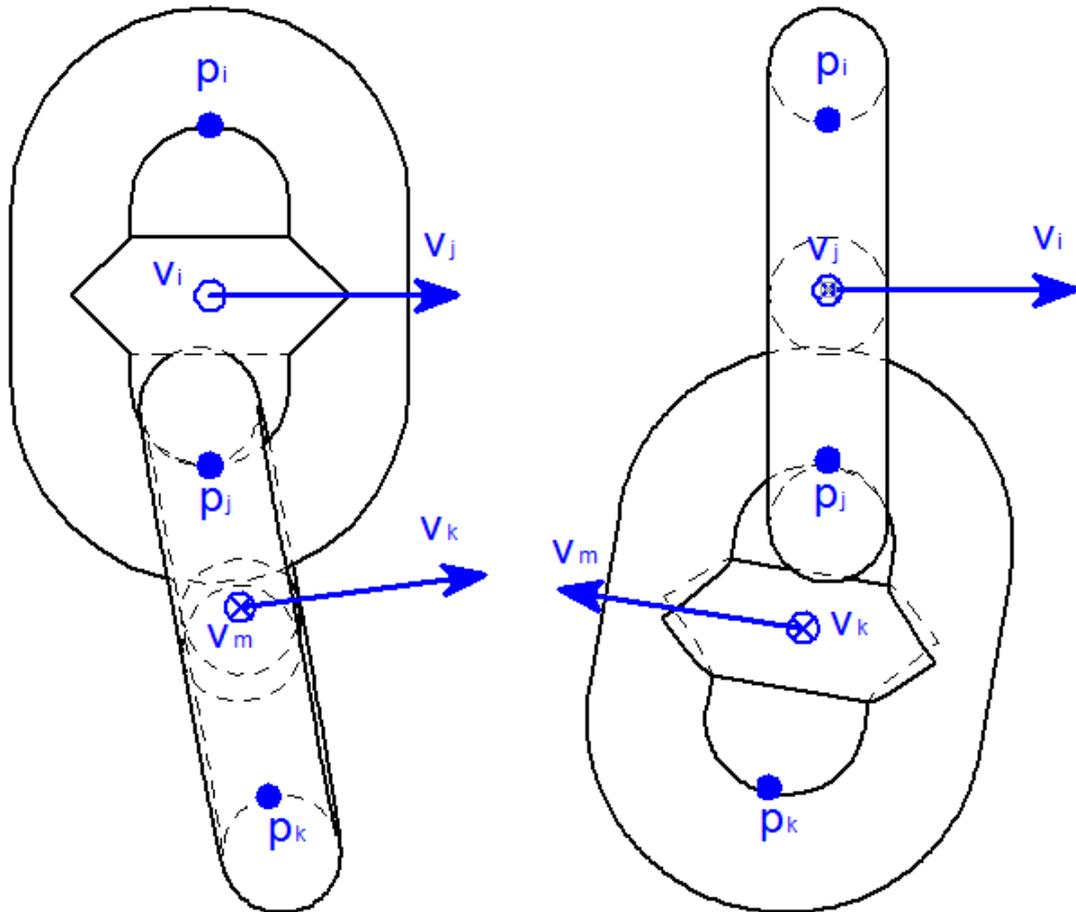


Figura 4-1. Restricciones para los eslabones

Variables de los elementos:

- Puntos: \mathbf{p}_i , \mathbf{p}_j , \mathbf{p}_k
- Vectores: \mathbf{v}_i , \mathbf{v}_j , \mathbf{v}_k , \mathbf{v}_m .

Elementos modelización:

Dos puntos y dos vectores por sólido. Hay un punto compartido entre eslabón y eslabón.

Restricciones adicionales:

$$\mathbf{v}_i^T \mathbf{v}_k = 0$$

$$\mathbf{v}_j^T \mathbf{v}_m = 0$$

Número de grados de libertad:

El número de grados de libertad entre sólidos, teniendo en cuenta las restricciones automáticas y adicionales es de 2 (dos puntos y dos vectores son 12 coordenadas, tenemos una restricción de distancia, dos de norma de vector unidad, tres de producto escalar constante y tres de combinación lineal, más las dos restricciones adicionales da $12-10=2$) entre eslabones de la cadena.

Grillete y caña

Se describirán a continuación las variables y restricciones para el grillete y la caña.

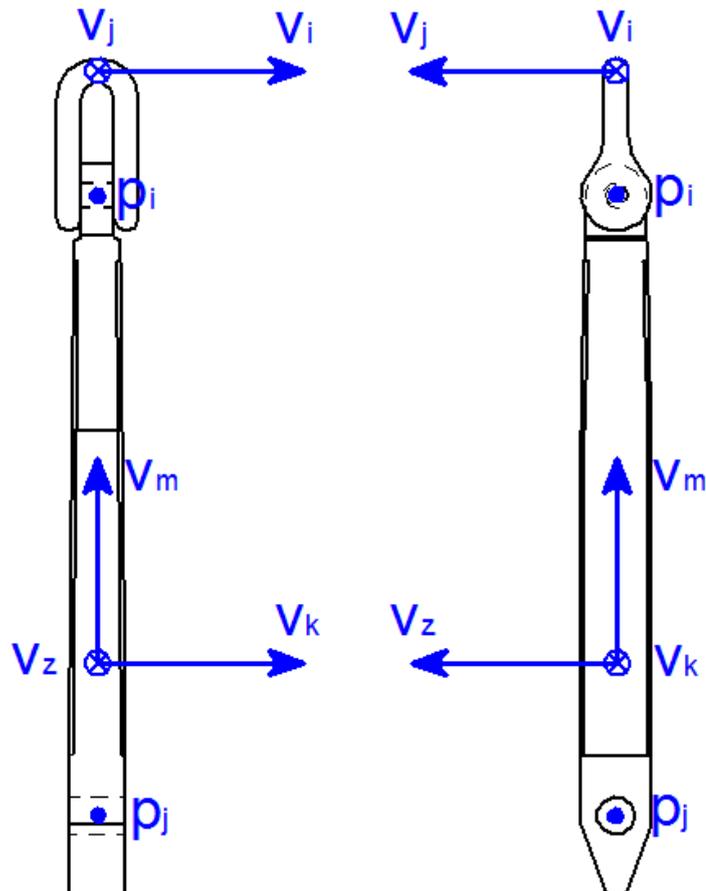


Figura 4-2. Restricciones para el grillete y la caña

Variables elemento:

- Puntos: p_i , p_j
- Vectores: v_i , v_j , v_k , v_m , v_z

Elementos modelización:

Un punto y dos vectores para el grillete y dos puntos y tres vectores para la caña.

Restricciones:

No son necesarias restricciones adicionales a las impuestas por la MBSLIM debido a que se comparte un punto y el vector del eje.

Número de grados de libertad:

Realizando el cómputo de las variables y restricciones impuestas, sale un grado de libertad entre el grillete y la caña.

Caña y ancla

Se describirán a continuación las variables y restricciones para la caña y el ancla.

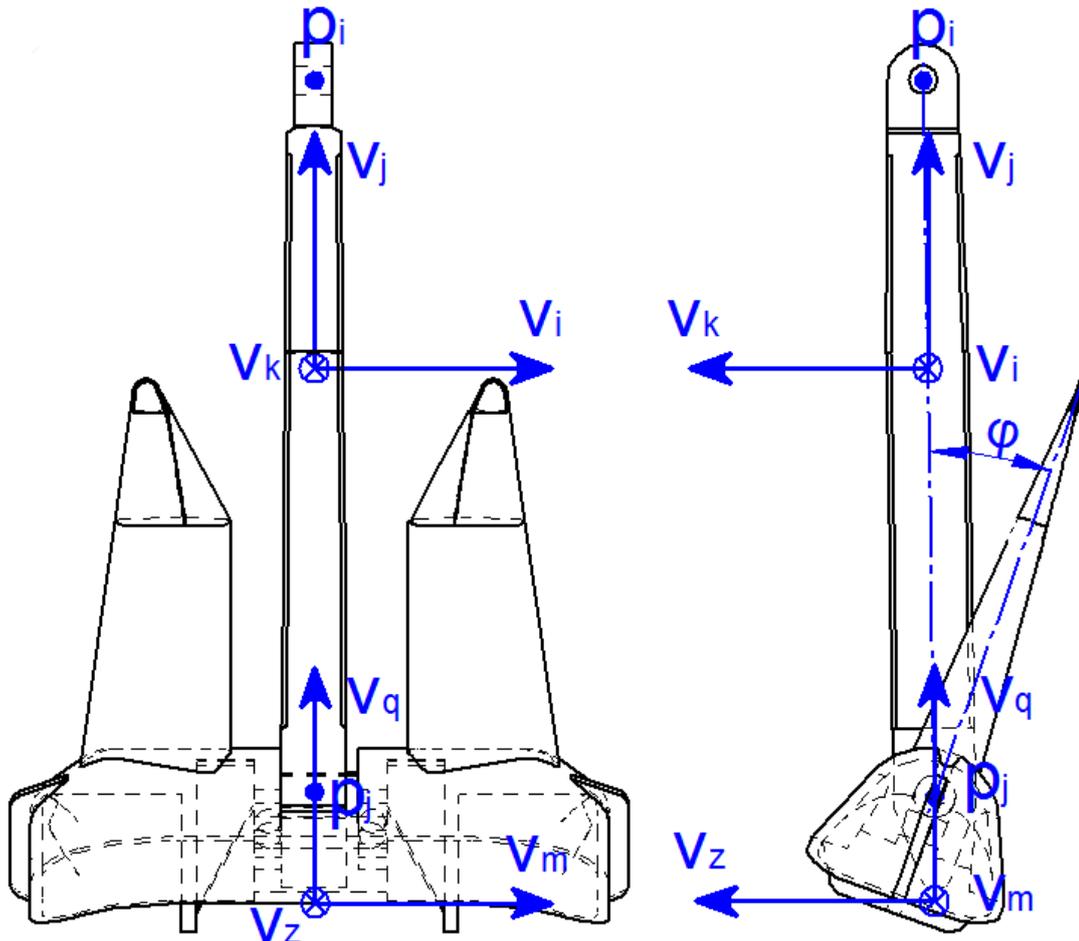


Figura 4-3. Restricciones para la caña y las uñas del ancla

Variables elemento:

- Puntos: p_i , p_j
- Vectores: v_i , v_j , v_k , v_m , v_q , v_z
- Ángulo ϕ

Elementos modelización:

Dos puntos y tres vectores para la caña y un punto y tres vectores para las uñas del ancla.

Restricciones:

No son necesarias restricciones adicionales a las impuestas por la MBSLIM debido a que se comparte un punto y el vector del eje.

Número de grados de libertad:

Realizando el cómputo de las variables y restricciones impuestas, sale un grado de libertad entre la caña y las uñas del ancla.

4.1.3 Grados de libertad

Haremos en primer lugar un balance de ecuaciones/variables para poder apreciar con mayor claridad los GDL.

Sólido	Grados de libertad relativos
Eslabón 1	3
Eslabón 2	2
...	...
Eslabón "i"	2
Caña	1
Ancla	1

Tabla 3. Balance de grados de libertad

4.1.4 El vector de variables

Este vector, \mathbf{q} , variará en cada paso integrable del tiempo, recogiendo el valor de cada variable. Tendrá la siguiente forma:

$$\mathbf{q} = \left[\mathbf{p}_{i \text{ con } i=2,3,..,20}, \mathbf{v}_{i \text{ con } i=4,5,..,16}, \varphi_{i \text{ con } i=1,2,..,5}, S_1, S_2, S_3 \right] \quad (4.8)$$

4.1.5 Fuerzas**Fuerzas normales de contacto**

El modelo de fuerzas de contacto normales empleado en este trabajo, es el modelo desarrollado por Flores en el 2011 en (Flores, y otros 2011). El modelo es muy similar al modelo de Hunt-Crossley, desarrollado originalmente en (Hunt y Crossley 1975). La expresión de la fuerza normal del modelo de Flores es la siguiente:

$$\mathbf{F}_n = k_n \delta^n \left(1 + \frac{8(1-\epsilon)}{5\epsilon} \frac{\dot{\delta}}{\dot{\delta}_0} \right) \mathbf{n} \quad (4.9)$$

donde k_n es la rigidez equivalente del contacto y depende de la forma y el material de los sólidos que colisionan, n es el exponente de Hertz, $\delta = R_{sph} - |\mathbf{p}_{center} - \mathbf{p}_{contact}|$ es la indentación, $\dot{\delta}$ es su derivada temporal, $\dot{\delta}_0$ es la velocidad normal relativa entre los

sólidos en colisión cuando el contacto es detectado, ϵ es el coeficiente de restitución y n es la dirección de la fuerza de contacto normal.

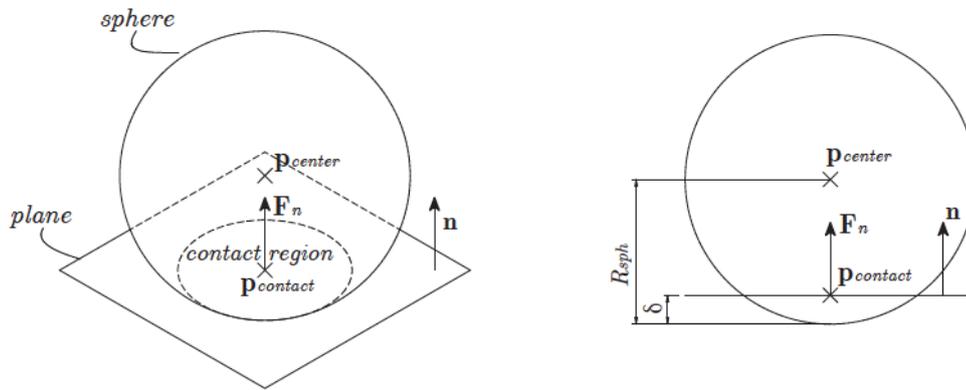


Figura 4-4. Contacto normal

El valor de k_n se puede calcular para dos sólidos en contacto con geometrías de esfera y plano respectivamente en los que la expresión para la rigidez puede ser expresada por

$$k_n = \frac{4}{3(\sigma_{sph} + \sigma_{pln})} \sqrt{R_{sph}} \quad (4.10)$$

siendo R_{sph} el radio de la esfera en contacto con el plano y los parámetros de sus respectivos materiales σ_{sph} y σ_{pln} vienen dados por

$$\sigma_{sph} = \frac{1-\nu_{sph}^2}{E_{sph}}; \quad \sigma_{pln} = \frac{1-\nu_{pln}^2}{E_{pln}} \quad (4.11)$$

donde ν y E son el cociente de Poisson y el módulo de Young o módulo elástico para cada uno de los dos materiales.

Fuerzas de rozamiento de contacto

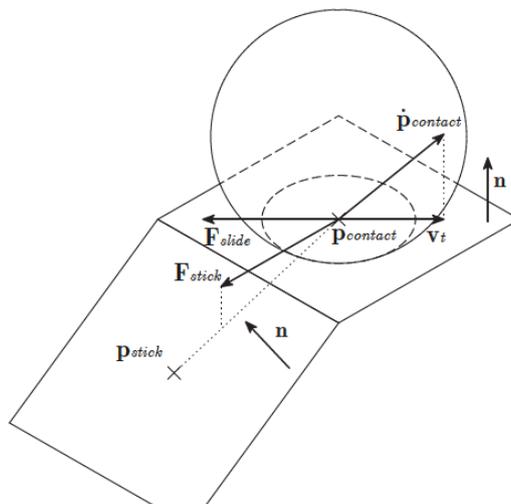


Figura 4-5. Contacto tangencial

El modelo de fuerzas de contacto tangenciales, o fuerzas de fricción, fue desarrollado en detalle en (Dopico, Luaces, y otros 2011) y está basado en la ley de Coulomb del rozamiento seco incluyendo adhesión y un término de disipación viscosa. La forma general de la fuerza es la siguiente,

$$\mathbf{F}_t = \kappa \mathbf{F}_{stick} + (1 - \kappa) \mathbf{F}_{slide} - \mu_{visc} \mathbf{v}_t \quad (4.12)$$

En la expresión previa, los primeros dos términos constituyen el rozamiento seco, mientras que el tercer término representa el rozamiento viscoso. Para tener una transición suave entre sticking (adhesión) y slipping (deslizamiento), la fuerza de fricción se divide en dos componentes acopladas mediante una función suave, siguiendo las ideas propuestas en (Gonthier, y otros 2004). El subíndice "t" indica aquí "tangencial".

En la ecuación (4.12), μ_{visc} es el coeficiente de rozamiento viscoso, \mathbf{F}_{stick} y \mathbf{F}_{slide} son las componentes de sticktion o adhesión y slipping o deslizamiento, es una función suave de la velocidad tangencial en el punto de contacto, \mathbf{v}_t que está definida en función del punto central de la región de contacto, $\mathbf{p}_{contact}$ y el vector normal al contacto, \mathbf{n} , como sigue:

$$\mathbf{v}_t = \dot{\mathbf{p}}_{contact} - \left(\mathbf{n}^T \dot{\mathbf{p}}_{contact} \right) \mathbf{n} \quad (4.13)$$

La citada función, tiene que cumplir las siguientes condiciones,

$$\kappa = \begin{cases} 0; & |\mathbf{v}_t| \gg v_{stick} \\ 1; & |\mathbf{v}_t| = 0 \end{cases} \quad (4.14)$$

donde v_{stick} es un parámetro del modelo que tiene en cuenta la velocidad de transición entre adhesión y deslizamiento. Una buena elección para esta función de transición fue proporcionada por (Gonthier, y otros 2004) y tiene la siguiente expresión:

$$\kappa = e^{-\left(\frac{|\mathbf{v}_t|}{v_{stick}}\right)^2} \quad (4.15)$$

La ecuación (4.12) muestra que la fuerza total está compuesta de tres componentes: la fuerza de rozamiento seco de deslizamiento cuando las velocidades son significativas, la fuerza de rozamiento seco adhesivo a bajas velocidades y la fuerza de rozamiento viscoso. La fuerza de rozamiento seco de deslizamiento viene dada por la expresión clásica de Coulomb, mientras que la fuerza de rozamiento seco adhesivo viene dada por las expresiones de los bristles, que son elementos viscoelásticos en forma de pelos o cerdas actuando entre los cuerpos en colisión.

Las expresiones de las fuerzas de deslizamiento y adhesión vienen dadas por las ecuaciones (4.16) y (4.17):

$$\mathbf{F}_{slide} = \begin{cases} 0; & |\mathbf{v}_t| = 0 \\ -\mu_{din} |\mathbf{F}_n| \frac{\mathbf{v}_t}{|\mathbf{v}_t|}; & |\mathbf{v}_t| > 0 \end{cases} \quad (4.16)$$

$$\mathbf{F}_{stick} = \begin{cases} 0; & s = 0 \\ \frac{f_{stick}^m}{s} (\mathbf{I}_3 - \mathbf{nn}^T) (\mathbf{p}_{contact} - \mathbf{p}_{stick}); & s > 0 \end{cases} \quad (4.17)$$

siendo μ_{din} el coeficiente de fricción dinámico, $s = |\mathbf{p}_{contact} - \mathbf{p}_{stick}|$ la deformación de los bristles, \mathbf{p}_{stick} el punto de adhesión, que inicialmente coincide con el centroide de la región de colisión en el instante en el que el contacto comenzó; \mathbf{I}_3 es la matriz de identidad de tamaño 3 x 3; f_{stick}^m es la función que representa el comportamiento de los bristles,

$$f_{stick}^m = -k_{stick} s - c_{stick} \dot{s} \quad (4.18)$$

siendo k_{stick} y c_{stick} los coeficientes de rigidez y amortiguamiento del modelo de adhesión. Sin embargo, hay un valor límite para la fuerza de adhesión:

$$|\mathbf{F}_{stick}| \leq \mu_{st} |\mathbf{F}_n| \quad (4.19)$$

En (4.19), μ_{st} es el coeficiente de fricción estático que en general es más elevado que el dinámico. Si se excede este límite y (4.19) no se cumple, se producen dos consecuencias: en primer lugar, la ecuación (4.18) ya no es válida para el comportamiento de los bristles y tiene que ser sustituida por la ecuación (4.20); en segundo lugar, el punto de adhesión debe ser actualizado con la expresión (4.21):

$$f_{stick}^m = \frac{-\mu_{st} |\mathbf{F}_n| s}{\left| (\mathbf{I}_3 - \mathbf{nn}^T) (\mathbf{p}_{contact} - \mathbf{p}_{stick}) \right|} \quad (4.20)$$

$$\mathbf{p}_{stick} = \mathbf{p}_{contact} - \left(\frac{\eta_{stick} \mu_{st} |\mathbf{F}_n|}{k_{stick}} \right) \frac{\mathbf{v}_t}{|\mathbf{v}_t|} \quad (4.21)$$

El coeficiente η_{stick} controla la deformación de los bristles cuando se alcanza la máxima fuerza. Físicamente el valor más razonable es $\eta_{stick} = 1$, pero pequeñas variaciones con $\eta_{stick} < 1$ pueden mejorar el comportamiento numérico del modelo.

Fuerza tope de ancla y caña

La naturaleza de esta fuerza tiene una forma similar a la de un muelle-amortiguador que actúa por tramos. El parámetro fundamental es el ángulo que forman las uñas del ancla con la caña. Este ángulo tiene una rotación máxima, para este caso en particular, de 30° respecto a la vertical para evitar que las uñas puedan oscilar libremente y provocar problemas en el proceso de estibado.

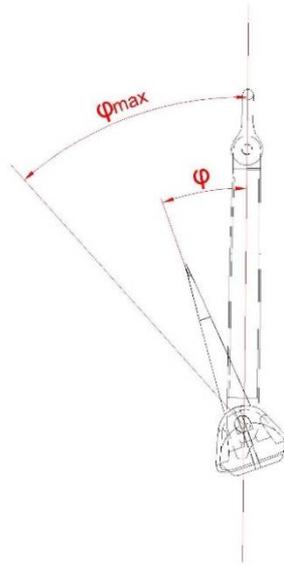


Figura 4-6. Ángulo del ancla con la caña

La expresión de la fuerza queda:

$$f_m^{lin} = \begin{cases} -k_m \cdot (\varphi - \varphi_{max}) - c_m \cdot \dot{\varphi}; & \text{si } \varphi > \varphi_{max} \\ -k_m \cdot (\varphi + \varphi_{max}) - c_m \cdot \dot{\varphi}; & \text{si } \varphi < -\varphi_{max} \\ 0 & \text{si } -\varphi_{max} < \varphi < \varphi_{max} \end{cases} \quad (4.22)$$

siendo k_m la constante elástica, c_m la constante de amortiguamiento, φ el ángulo, $\dot{\varphi}$ su derivada, φ_0 el ángulo natural y f_m la fuerza de ancla y caña.

Como se puede ver en (4.22), pueden presentarse tres casos. El primero es para $\varphi > \varphi_{m\acute{a}x}$, el segundo para $\varphi < -\varphi_{m\acute{a}x}$ y el tercero para $-\varphi_{m\acute{a}x} < \varphi < \varphi_{m\acute{a}x}$. Para este último caso la f_m será nula.

Capítulo 5
Simulación de la
maniobra en
SIMULANCLA

5 SIMULACIÓN DE LA MANIOBRA EN SIMULANCLA

En este capítulo explicaremos los pasos empleados para simular la maniobra de nuestra maqueta. De este modo se conocerá la dinámica del diseño del ancla cargado en SIMULANCLA. Los datos que nos proporcione el simulador se compararán con los obtenidos en la maniobra de la maqueta real, lo que nos permitirá sacar conclusiones sobre la validación del software. Pero antes se hará una breve introducción del programa (Dopico, Simulancla 2018), (Dopico, Software para el análisis computacional del diseño de sistemas de fondeo en buques de gran tamaño 2015).

5.1 Uso del software SIMULANCLA

El software SIMULANCLA permitirá realizar una simulación completa y detallada, sin la necesidad de fabricar maquetas para cada tipo de barco diseñado, y comprobar la maniobra en cuestión. Gracias al software, será posible definir la maniobra con una mayor similitud a la situación real. Además, también será posible visualizar la maniobra pudiendo conocer diferentes datos en tiempo real en cualquier momento.

El software SIMULANCLA permite analizar la dinámica de la maniobra basándose en la dinámica de sistemas multicuerpo, siendo ésta la disciplina en la que se han fundamentado los cálculos que se realizan para simular las maniobras de anclas.

Los métodos de la dinámica de sistemas multicuerpo son generales; no obstante, cada problema real tiene su problemática particular que puede convertirlo en un auténtico reto. En el caso de los sistemas multicuerpo resueltos por SIMULANCLA, una de las mayores dificultades estriba en los modelos de contacto establecidos entre todos los sólidos que intervienen en la simulación.

El programa SIMULANCLA consta de dos fases bien diferenciadas: la de especificación de diseño y la de simulación o visualización.

La primera fase consiste en definir paso a paso cada uno de los aspectos del sistema a simular, de una manera coherente y ordenada. Para ello, se establecen parámetros referentes al modo de funcionamiento del programa y otros parámetros físicos.

La segunda fase consiste en la visualización de la representación gráfica tridimensional del proceso de maniobra, en la que se muestra o bien la progresión de la simulación, o bien el movimiento y los valores de las fuerzas generadas durante éste en una simulación previa.

Especificar la posición individual de todos y cada uno de los elementos al inicio de la simulación sería prácticamente imposible para el usuario. Por lo tanto, solamente es necesario especificar de manera exacta los parámetros básicos de posición de la cadena y del ancla, y el resto se aproximan implícitamente a partir de éstos. Al iniciar las sesiones de simulación, es conveniente partir de una posición en la que los elementos del sistema estén en equilibrio estático, una opción que incorpora el programa y que permite calcular el estado de equilibrio mediante un proceso previo a la simulación.

Una vez obtenida la posición estática inicial, la simulación de la maniobra propiamente dicha da comienzo. La simulación termina cuando se alcance la condición de estiba del ancla, es decir, cuando la cadena quede bloqueada debido a que el ancla esté firmemente sujeta al casco del buque o a la regala. No obstante, sobre todo en los casos en los que el diseño no sea válido, puede interesar hacer simulaciones parciales más cortas y no se alcanzará dicha condición, por lo que el sistema puede no detenerse por sí mismo. Para este caso, se establece un valor máximo de la duración de la simulación, de tal manera

que ésta se detenga incondicionalmente si no se llegase a la condición de estiba en ese intervalo de tiempo. Dicho intervalo de tiempo se expresa en tiempo de simulación, no en tiempo real. El tiempo final de simulación podrá ser establecido en función del tiempo que no se deba exceder en la maniobra del buque de la simulación. Se puede ajustar la magnitud y dirección de la gravedad para posibilitar que se puedan emplear ficheros de geometría que no estén orientados con respecto al eje de coordenadas por defecto del programa, que asume que el eje “Z” está orientado verticalmente y hacia arriba. El último parámetro ajustable es el valor del coeficiente de fricción, μ , con el que se puede ajustar la magnitud de las fuerzas tangenciales de rozamiento que se deriven en los posibles contactos de las piezas entre sí. Esto se puede observar en la siguiente imagen:

Defina ficheros de datos de USUARIO y MANIOBRA:

Ruta fichero usuario/maniobra y ruta ficheros CAD

Nombre fichero usuario/maniobra:

Defina el modo de funcionamiento (SIMULACIÓN/REPRODUCCIÓN):

Simular

Reproducir

Simular con depuración (*sólo a petición del LIM)

Defina los parámetros básicos de la simulación:

Tiempo final de simulación (s):

Aceleración de la gravedad (m/s²): X Y Z

Iniciar simulación en posición de equilibrio

Coefficiente de fricción:

Figura 5-1. Configuración de la simulación

El primer paso para poder llevar a cabo la simulación, será desarrollar los ficheros CAD del buque que se vaya a estudiar. Se pueden diferenciar claramente dos partes en la geometría del buque, en primer lugar, se tendrán todos aquellos elementos del buque que van a intervenir directamente en la simulación (zonas de contacto del casco, escobén, cadena y ancla); por otra banda, se tendrán el resto de componentes del buque, es decir, lo que será la geometría auxiliar en términos de simulación, pues su presencia no va a variar los resultados de la simulación. De este modo, el programa necesitará que se le proporcionen los siguientes ficheros CAD:

- Fichero que contenga la geometría colisionable del casco.
- Fichero que contenga la geometría de la regola/escobén.
- Fichero que contenga la geometría auxiliar que se desea visualizar en la simulación pero que no participa en los cálculos dinámicos.

Los formatos de geometría que el programa es capaz de manejar son diversos, pero principalmente se dividen en dos categorías: de malla y de superficie.

Los ficheros de malla almacenan la representación geométrica de un objeto tridimensional como conjuntos de pequeñas facetas planas, poligonales, aunque

generalmente se trata de elementos triangulares. Este tipo de ficheros es sencillo de interpretar, y se asemeja mucho a la forma con la que el programa opera finalmente. No obstante, esta representación es una aproximación de la forma real del objeto representado, y de ello se deriva que su bondad dependa de la resolución y corrección del mallado con la que se generó. En las regiones de superficie más complejas, es necesario que los elementos de la malla estén dimensionados correctamente, puesto que influyen directamente en la calidad de los resultados obtenidos. El usuario debe garantizar que las mallas que introduce presenten una calidad mínima.

Los ficheros de superficie almacenan la geometría de cada uno de los objetos como la unión de varias superficies paramétricas. La principal ventaja de los formatos de superficie consiste en que su precisión no depende de la elección de un parámetro de resolución o tamaño de elemento. Obviamente, dichas superficies también constituyen una aproximación al objeto real, pero se definen mediante un conjunto muy reducido de parámetros, y son capaces de representar de manera exacta cierto tipo de superficies muy comunes.

En esta fase el usuario debe especificar tres puntos básicos para la situación inicial de la cadena: uno tangente al molinete, otro en el centro de entrada de la regala de cubierta y, por último, el del centro de la salida de la regala de forro. El punto de tiro tangente al molinete, es el punto en la circunferencia del molinete donde la cadena es enrollada por el sistema de recogida. Las coordenadas de entrada de la regala de cubierta definen el punto de entrada al escobén en la regala de cubierta, que debe estar centrado en la prolongación del cilindro del escobén y ligeramente por encima de la cubierta del buque, para que la cadena en la posición inicial no colisione con el buque. Este es uno de los puntos que se toma como aproximados, ya que la posición de la cadena, tomando las consideraciones de contacto y peso propio de la misma, es en principio desconocida. El ya descrito proceso de determinación del equilibrio inicial permite calcular la posición exacta antes de comenzar la simulación. Finalmente, el usuario debe indicar las coordenadas de salida de la regala de forro, es decir, el punto de salida del escobén en la regala de forro. Debe estar centrado en la prolongación del cilindro del escobén y ligeramente fuera de la regala de forro, para que la cadena en la posición inicial no colisione con el buque. A continuación se puede observar una imagen con estas coordenadas:

Introduzca los datos del BUQUE:

Fichero CAD casco:	casco003.obj				
Fichero CAD regolas/escoben:	regala_F105_MEF.obj				
Fichero CAD auxiliar (geom. no colisionable):	resto_casco2.obj				
Desplazamiento global de las regolas/escoben (m):					
X	0.00000	Y	0.00000	Z	0.00000
Rotación de las regolas/escoben alrededor del plano del casco (deg):					
Angle°	0.000				
Coordenadas globales del punto de tiro tangente al molinete (m):					
X	121.4700	Y	-1.43600	Z	11.93200
Coordenadas locales de entrada de la regala de cubierta (m):					
X	129.8610	Y	-0.91800	Z	12.67500
Coordenadas locales de salida de la regala de forro (m):					
X	131.5810	Y	-3.10600	Z	9.45500

Figura 5-2. Configuración del buque

El siguiente paso será proporcionar la geometría de los elementos móviles. Se trata de la caña del ancla, las uñas del ancla, los eslabones de la cadena, el grillete y, además, algunos datos necesarios del molinete. Estos objetos se clasifican en dos tipos principales: genéricos y paramétricos. Los objetos genéricos también se definen mediante ficheros de malla o superficie. Sin embargo, dado que son objetos móviles, es necesario conocer una serie de parámetros adicionales para introducirlos. Dichos parámetros son: la masa, el centro de gravedad, y el tensor de inercia de cada una de las piezas. La mayoría de los programas de CAD son capaces de suministrar estos datos, pero de no ser así, el programa es capaz de calcularlos por sí mismo a través de las mallas que representan a los objetos, como se puede ver en la Figura 5-3:

Los valores calculados automáticamente son

Masa: 2112.93 kg

Centro de masas:

4.14627e-07	-3.10966e-07	1.32356
-------------	--------------	---------

Momentos principales de inercia:

1857.63	1849.21	30.3644
---------	---------	---------

Productos de inercia:

-2.48939e-05	-2.48939e-05	0.00633999
--------------	--------------	------------

◆ Desea incorporar estos valores a la ficha?

Figura 5-3. Cálculo de las características del elemento

El primero de los objetos móviles a definir es la caña del ancla, como se puede ver en la Figura 5-4. Debe estar acondicionada de tal manera que una de las articulaciones coincida con el centro de su sistema de coordenadas local, en el que la geometría está definida. Se deben suministrar adicionalmente las coordenadas de la articulación del otro extremo, de tal manera que sus dimensiones queden implícitamente determinadas. La definición del ancla se completa con el siguiente formulario en aparecer, en el que se piden los datos de las uñas de la misma. La geometría tiene que venir dada mediante un fichero de CAD de malla o superficie, del cual se puedan extraer las propiedades físicas de masa, o bien introducirlas manualmente.

La geometría de las uñas, al igual que en el caso de la caña, tiene que estar expresada en un sistema de coordenadas establecido previamente. Es necesario determinar otra serie de parámetros adicionales para definir el movimiento entre estas dos piezas. El ángulo tope entre caña y ancla es el ángulo máximo de giro para el que el ancla hace tope con la caña, lo cual influye directamente en las posiciones de estiba que se adopten finalmente. El ángulo inicial entre caña y ancla es el ángulo inicial que forman ambas piezas antes de empezar la simulación. Dependiendo del ángulo elegido, las uñas caen hacia adelante o hacia atrás de la caña, por lo que se puede simular escenarios en los que una posición particular de inicio en la configuración del ancla pueda ser desfavorable. El diálogo correspondiente se puede observar en la Figura 5-5.

Introduzca los datos de la CAÑA del ancla:

Masa (kg):
2111.00 Fichero CAD caña: canha_MEF.obj

Centro de masas (m):
Xcdm 0.00000 Ycdm 0.00000 Zcdm -1.32401

Tensor de inercia en el CDM del sólido (kg·m²):
Ixx 5556.76000 Iyy 5548.34000 Izz 30.35400
Pxy 0.00000 Pxz 0.00000 Pyz 0.00000

Coordenadas locales de la articulación con el grillete de la cadena (m):
Xj 0.00000 Yj 0.00000 Zj 3.03400

Coordenadas locales de la articulación con el ancla (m):
Xj 0.00000 Yj 0.00000 Zj 0.00000

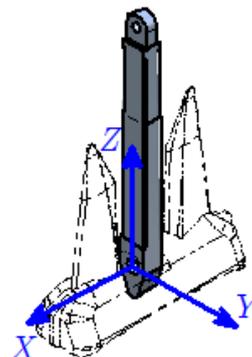


Figura 5-4. Configuración de la caña

Introduzca los datos del ANCLA:

Masa (kg):
4303.00 Fichero CAD ancla: ancla_MEF.obj

Centro de masas (m):
Xcdm 0.00000 Ycdm 0.00000 Zcdm -0.18159

Tensor de inercia en el CDM del sólido (kg·m²):
Ixx 3018.62000 Iyy 1199.11000 Izz 2098.40000
Pxy 0.00000 Pxz 0.00000 Pyz 0.00000

Coordenadas locales de la articulación con la caña (m):
Xj 0.00000 Yj 0.00000 Zj 0.00000

Ángulo tope entre ancla y caña (deg):
30.00

Ángulo inicial entre ancla y caña (deg):
0.00

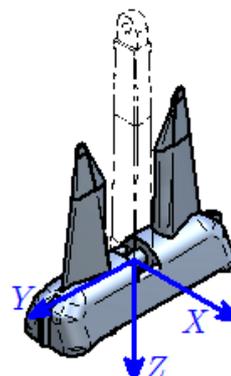


Figura 5-5. Configuración del ancla

Para el resto de elementos, tales como los eslabones o el molinete, simplemente habrá que configurar los distintos parámetros que se solicitan y el propio software los combina con el resto de la geometría proporcionada.

Los eslabones se agrupan en el conjunto de los objetos paramétricos anteriormente mencionados. Este tipo de elementos posee una geometría y características conocidas, por tratarse de elementos cuyas dimensiones están normalizadas según la *International Standard ISO 1704 (1973)* (International Organization for Standardization 1973). El parámetro que define a cada uno de los eslabones, a parte de su tipo, que determina su geometría, es el diámetro de la cadena. Todas las medidas de los eslabones están parametrizadas en función de este diámetro.

En cuanto al grillete, sí será necesario proporcionar el fichero CAD correspondiente, ya que va a influir en el comportamiento de la cadena. En el caso del molinete, se caracteriza mediante la fuerza máxima de tiro que puede desarrollar el motor, y la velocidad lineal de tiro del molinete. Ambos valores sirven para que el programa pueda simular el comportamiento del motor, es decir, tratar de alcanzar la velocidad de tiro especificada, pero sin superar nunca la fuerza máxima, que representa la potencia máxima que puede entregar el propio motor. A continuación se muestran imágenes de los distintos diálogos para meter los datos de la cadena, grillete y molinete:

Introduzca los datos de la CADENA:

Tipo y número de eslabones:

1	End Shackle	1
2	End Link	1
3	Enlarged Link	1
4	Common Link	50
5	Kenter Link	1
6	Common Link	1
7	Enlarged Link	1
8	Swivel	1
9	Enlarged Link	1
10	Common Link	1
11	Kenter Link	1
12	Common Link	0
13	Common Link	0
14	Common Link	0
15	Common Link	0

Diámetro de la cadena (mm):

58.0

Ángulo inicial en grados (deg) para los eslabones pivote (swivel link):

0

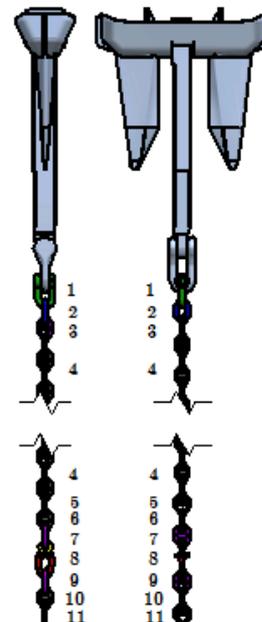


Figura 5-6. Configuración de la cadena

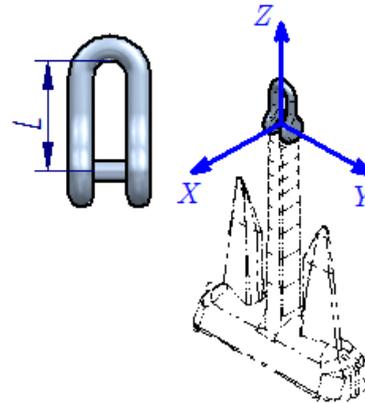
Introduzca los datos del GRILLETE de unión del ancla con la cadena:

Fichero CAD grillete: grillete_F105_MEF.obj

Tensor de inercia en el CDM del grillete ($\text{kg}\cdot\text{m}^2$):

Ixx 24.57070 Iyy 20.77530 Izz 6.09469

Masa (kg): 266.43 Altura del centro de masas (m): Zcdm 0.15217 Longitud interior del grillete, L (m): L 0.55000



Introduzca los datos del MOLINETE:

Fuerza máxima de tiro del molinete (N): 150000

Velocidad lineal de tiro del molinete (m/s): 0.20

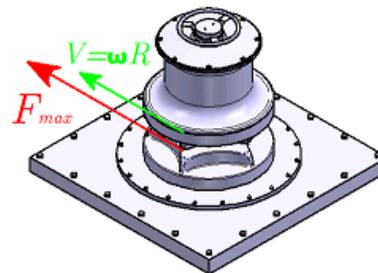


Figura 5-7. Configuración del grillete y el molinete

Una vez que están proporcionados los datos necesarios, el programa lee los parámetros y lanza la simulación. A medida que se computa el estado del sistema en cada instante de tiempo, se representan gráficamente los elementos del sistema de fondeo de tal manera que el usuario pueda observar si el sistema se ajusta a los requerimientos antes de que la simulación termine.

En el curso de la simulación, se representan gráficamente los valores de magnitudes físicas de interés: las fuerzas derivadas del contacto y la fuerza de tiro del motor de recogida. Las fuerzas de contacto normal se representan como flechas verdes, de longitud proporcional a su magnitud, y con origen en cada uno de los puntos de contacto. Se trata de la fuerza de reacción entre un par de elementos. Asociada a éstas se encuentran las indicaciones de fuerzas de rozamiento, que se representan como flechas rojas con el origen común a su fuerza de contacto asociada. La fuerza de tiro se representa con una flecha azul, y de manera similar, se indica su valor numérico. Por último, la pantalla de gráficos admite varias modalidades de visualización, como la vista en rejilla en la que se puede ver directamente la estructura de las mallas de todos los objetos. En las Figura 5-8 y Figura 5-9 se muestran estos detalles.

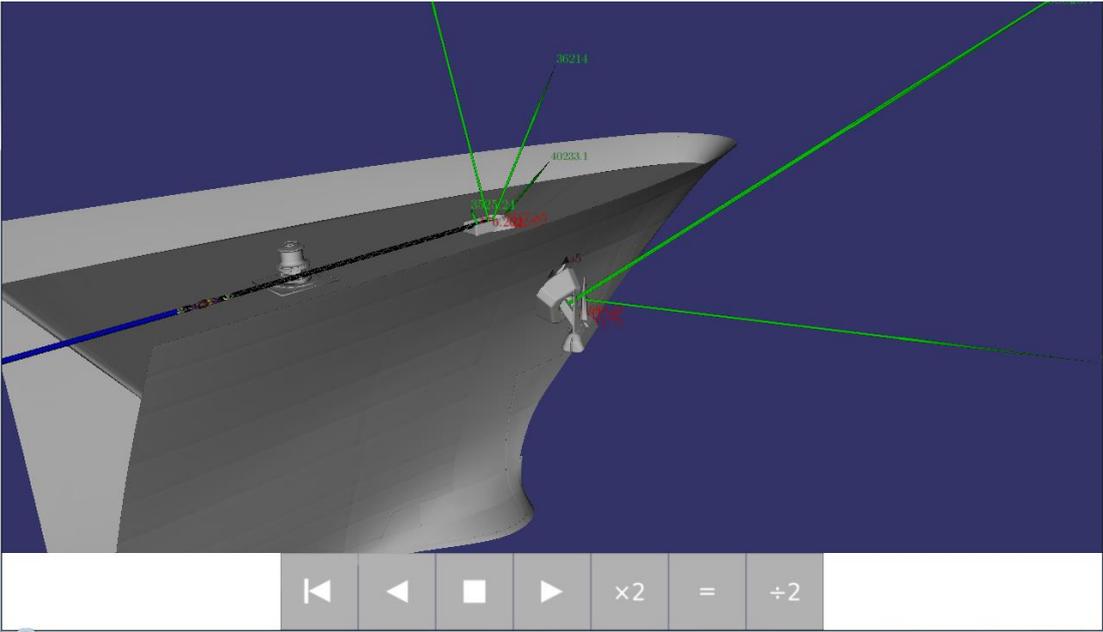


Figura 5-8. Fuerzas representadas en la simulación

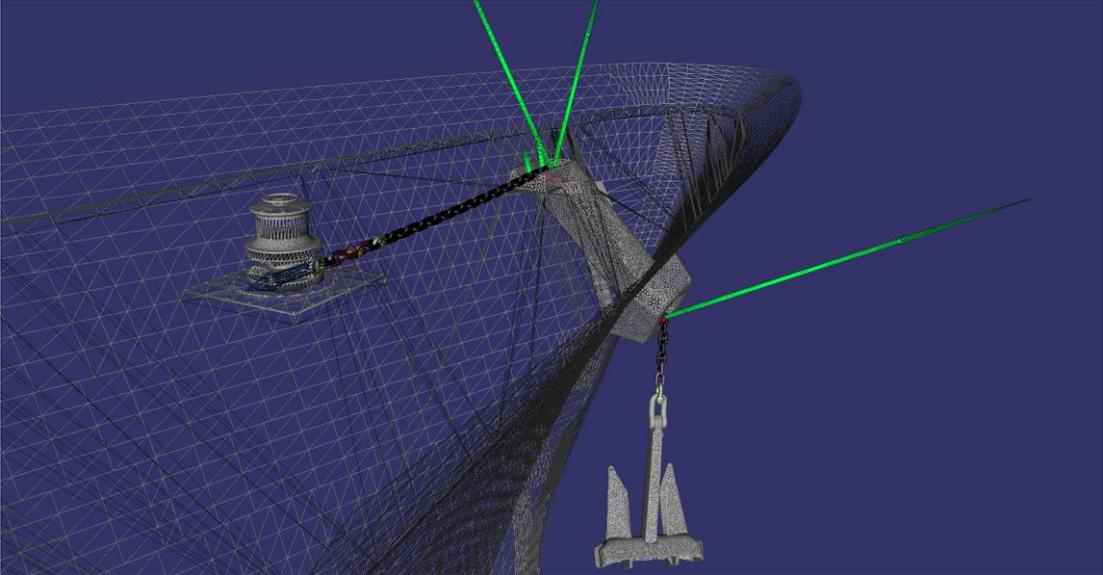


Figura 5-9. Estructura de malla

Antes de introducir los archivos y datos necesarios en SIMULANCLA, se prepararán los archivos CAD de los que partimos para que puedan ser leídos por el software. Esta preparación consistirá en pasar los modelos de formato .par a formato .obj mediante el empleo del software gratuito Blender (Blender Foundation 2017).

5.2 Preparación de los modelos

Para la preparación de los modelos, primero se guarda la pieza de formato .par a formato .stl.

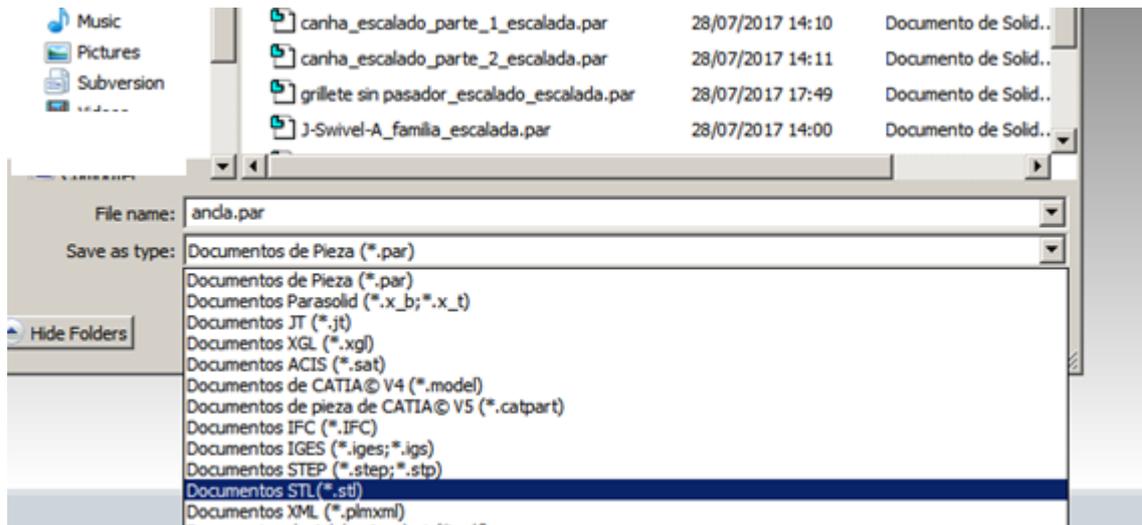


Figura 5-10. Conversión modelo a formato .stl

Cuando se tiene el archivo en el susodicho formato, con el programa Blender vamos a “File → Import → Stl”

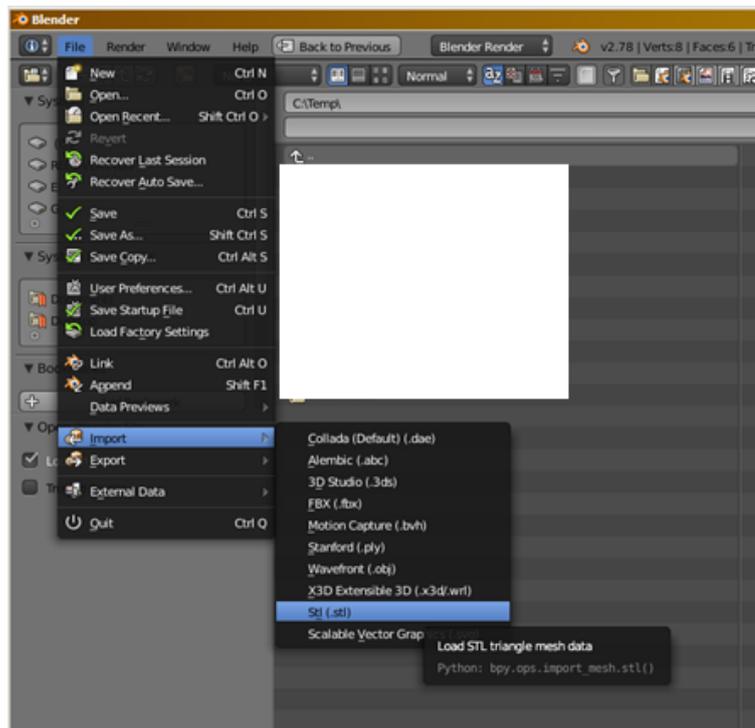


Figura 5-11. Apertura del archivo .stl en Blender

Una vez introducida la pieza que se quiere exportar a formato .obj, vamos a “File → Export → Wavefront”.

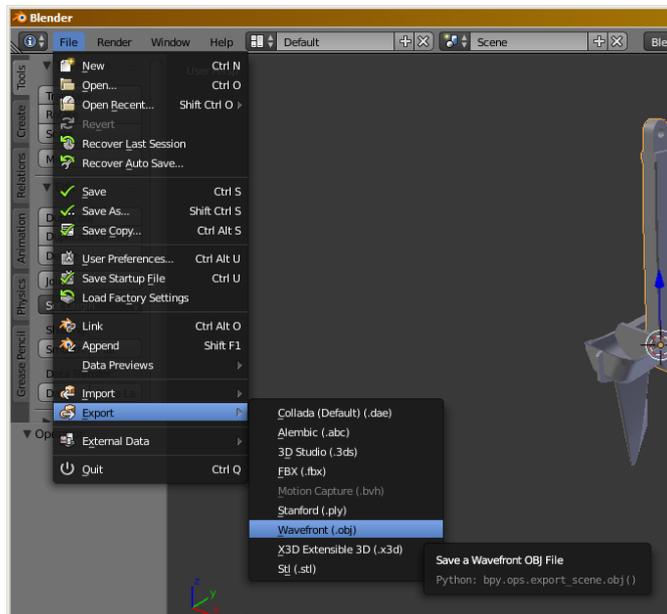


Figura 5-12. Exportación del modelo a formato .obj

Se selecciona la ruta donde deseáramos guardar el nuevo archivo y se pulsa sobre botón “Export OBJ”.

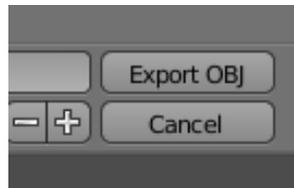


Figura 5-13. Botón de exportado a formato .obj

Una vez hecho el proceso anterior con todos los archivos necesarios, se procede a preparar la simulación en SIMULANCLA, introduciendo los parámetros necesarios con ayuda del asistente de configuración.

5.3 Parámetros de la simulación

Se realizarán dos simulaciones. Una con el modelo del escobén sin la tapa transparente y otra con el escobén con la tapa.

El valor de los parámetros introducidos para la simulación de este proyecto son los mismos para las dos simulaciones (sólo variará el modelo CAD del escobén). Por simplicidad sólo se explicará el valor de los parámetros introducidos cuando su valor sea distinto del que ofrece por defecto.

The screenshot shows a software window titled 'Asistente de SIMULANCLA'. It contains three main sections:

- Defina ficheros de datos de USUARIO y MANIOBRA:** Includes a file path selection button, a text input field for 'Nombre fichero usuario/maniobra' containing 'maniobram', and a button labeled 'Presione para cargar los datos de usuario'.
- Defina el modo de funcionamiento (SIMULACIÓN/REPRODUCCIÓN):** Features three radio buttons: 'Simular', 'Reproducir' (which is selected), and 'Simular con depuración (*sólo a petición del LIM)'. There is also a checkbox for 'Visualizar el océano'.
- Defina los parámetros básicos de la simulación:** Contains several input fields: 'Tiempo final de simulación (s): 2,00', 'Velocidad reproducción v/t.r.: 0,025', 'Aceleración de la gravedad (m/s²): X 0,00 Y 0,00 Z -9,81', 'Coeficiente de fricción: 0,36', and 'Densidad general (kg/m³): 1019,2'. A checkbox 'Iniciar simulación en posición de equilibrio' is checked.

At the bottom, there are three buttons: '< Back', 'Next >', and 'Cancel'.

Figura 5-14. Parámetros de simulación - Configuración de la simulación

- **Nombre fichero usuario/maniobra:** “maniobram”

Este nombre fue el elegido para guardar los parámetros de configuración de la simulación usada para la validación.

- **Coeficiente de fricción:** 0,36

Para medir el coeficiente de fricción se usó la superficie de una de las piezas y se midió el ángulo de inclinación en el que un objeto empezaba a resbalar. Con este ángulo se puede obtener el coeficiente de fricción como $\mu = \tan\alpha$, siendo α el ángulo medido. Los valores medidos se encontraron alrededor de 20 grados, por lo que se obtuvo un valor de coeficiente de fricción medio de 0,36.

- **Densidad general:** 1019,2

Este valor es la media de las densidades de cada pieza, obteniendo así la densidad del material usado para crear las piezas, teniendo en cuenta posibles irregularidades. Para hallar el valor de densidad de cada pieza primero se pesó en una báscula de precisión y después se dividió el valor por el volumen obtenido al consultar las propiedades de la pieza impresa en el software de CAD.

The screenshot shows a software window titled 'Asistente de SIMULANCLA'. The main heading is 'Introduzca los datos del BUQUE:'. Below this, there are three file selection fields: 'Fichero CAD casco:' with the path 'ras_Simulanda/Casco_Nuevo.obj', 'Fichero CAD regolas/escoben:' with the path 'la/Regola_Escoben_2_escalado.obj', and 'Fichero CAD auxiliar (geom. no colisionable):'. There are three sections of numerical input fields, each with X, Y, and Z coordinates: 1. 'Desplazamiento global de las regolas/escoben (m):' with values X: 0,0000, Y: 0,0000, Z: 0,0000. 2. 'Rotación de las regolas/escoben alrededor del plano del casco (deg):' with 'Angle°' set to 0,000. 3. 'Coordenadas globales del punto de tiro tangente al molinete (m):' with values X: 0,24772, Y: -0,00094, Z: 0,46816. Below these are two more sections of numerical input fields for local coordinates: 4. 'Coordenadas locales de entrada de la regola de cubierta (m):' with values X: 0,34410, Y: -0,13596, Z: 0,46049. 5. 'Coordenadas locales de salida de la regola de forro (m):' with values X: 0,36038, Y: -0,19818, Z: 0,34582. At the bottom right, there are three buttons: '< Back', 'Next >', and 'Cancel'.

Figura 5-15. Parámetros de simulación - Configuración del buque

- **Ficheros CAD**

Para los ficheros CAD que se pueden ver en la imagen anterior, se eligieron los modelos enviados a imprimir y cuya preparación se explicó en el apartado “5.2 Preparación de los modelos”. Aquí es donde varían las simulaciones entre sí, (cada simulación tendrá un fichero de CAD regola/escobén distinto)

- **Coordenadas**

Para obtener los diferentes valores de las distintas coordenadas que es necesario incluir en el programa, primero se midió la posición real de la maqueta, después se trasladó al modelo CAD y por último se utilizó la función de medida del Solid Edge para sacar un valor aproximado como se puede ver en la siguiente imagen.

CAPÍTULO 6: SIMULACIÓN DE LA MANIOBRA

MARTA FRADE MANSO

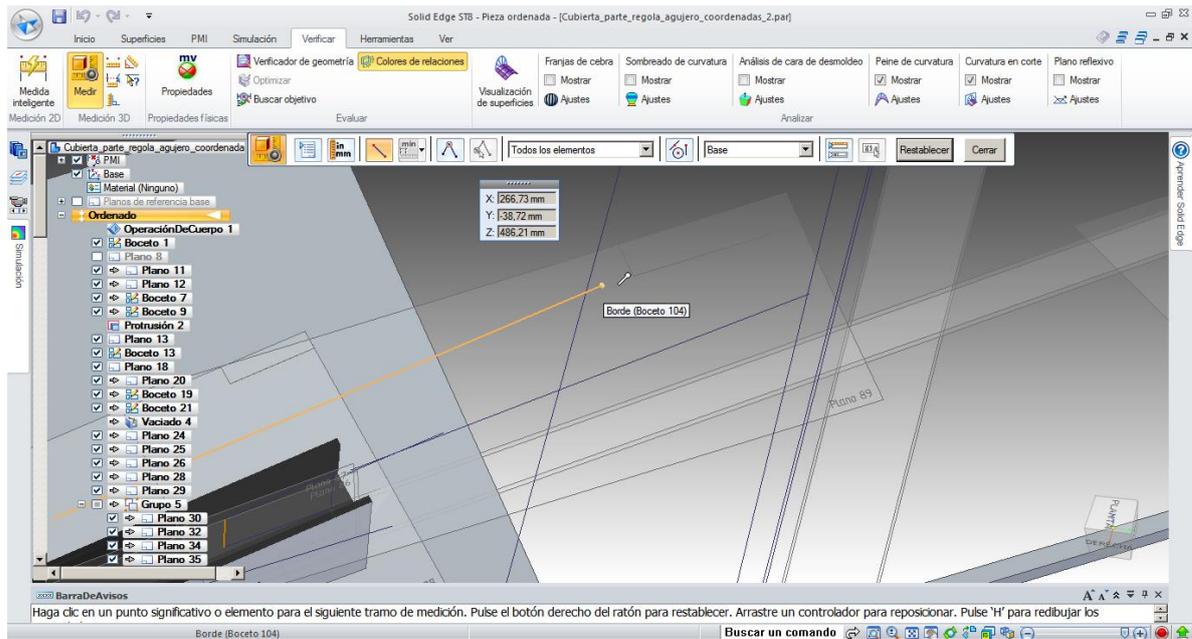


Figura 5-16. Obtención coordenadas en SolidEdge

Estos valores son proporcionados a SIMULANCLA y se observa la posición que muestra el programa, comprobando que coincida con la realidad. De no ser así se varían los valores hasta obtener la posición de la maqueta real. Este mismo método se emplea para sacar los valores de longitudes.

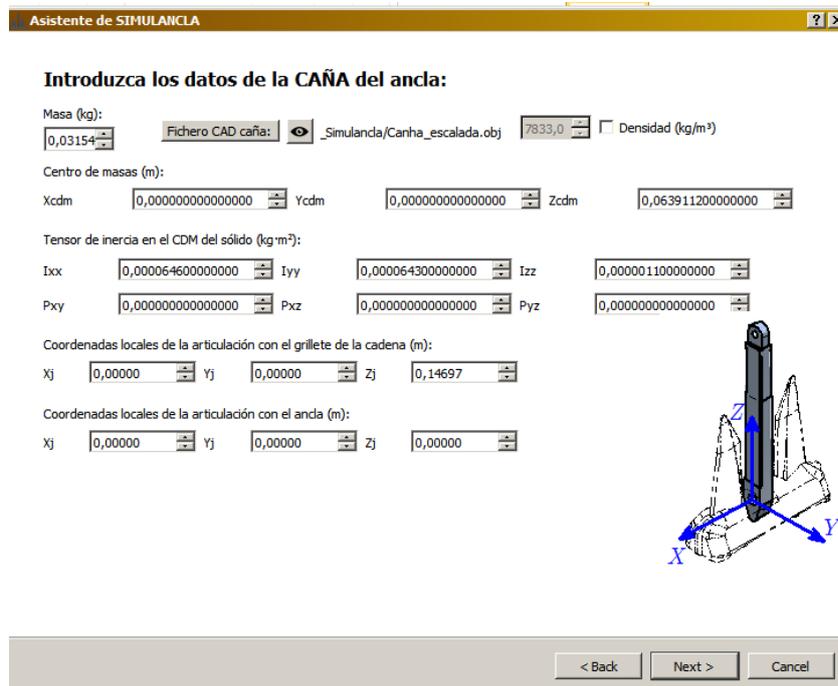


Figura 5-17. Parámetros de simulación - Configuración de la caña

- **Valores de masa, centro de gravedad y momentos de inercia**

Los valores de los centros de gravedad, masas y momentos de inercia los calcula el propio SIMULANCLA según lo explicado en el apartado 5.1.

- **Coordenadas de articulación**

El valor de estas coordenadas se obtiene con el mismo método explicado para la obtención de coordenadas en el punto de configuración del buque.

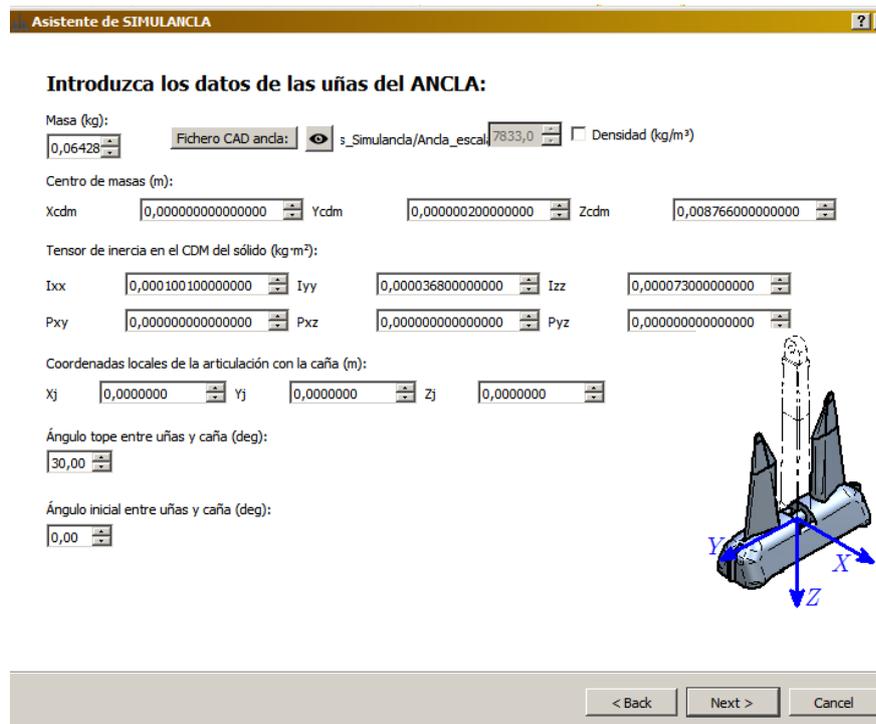


Figura 5-18. Parámetros de simulación - Configuración del ancla

- **Ángulo tope entre uñas y caña**

El valor de este parámetro es 30° ya que es el ángulo que tiene la maqueta construida.

- Para la obtención de las coordenadas se utiliza el mismo método explicado en puntos anteriores.

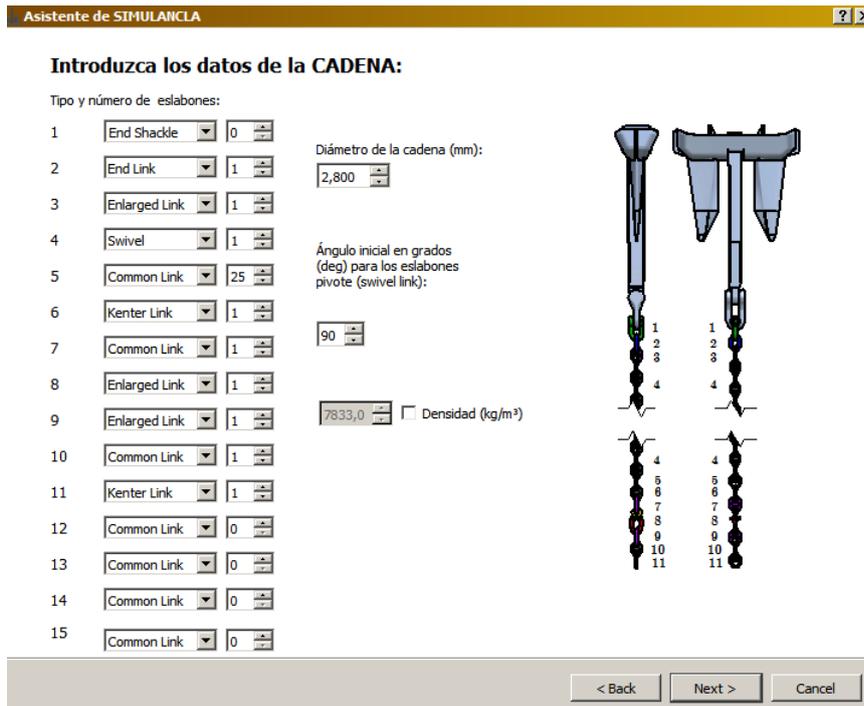


Figura 5-19. Parámetros de simulación - Configuración de la cadena

- Como ya se explicó en el apartado 5.1, los eslabones están parametrizados y tan sólo será necesario proporcionar su diámetro, tipo y número para cada uno en el orden que se puede ver en el dibujo de la Figura 5-19. Los valores introducidos de diámetro, tipo y número son aquellos de la maqueta real impresa.

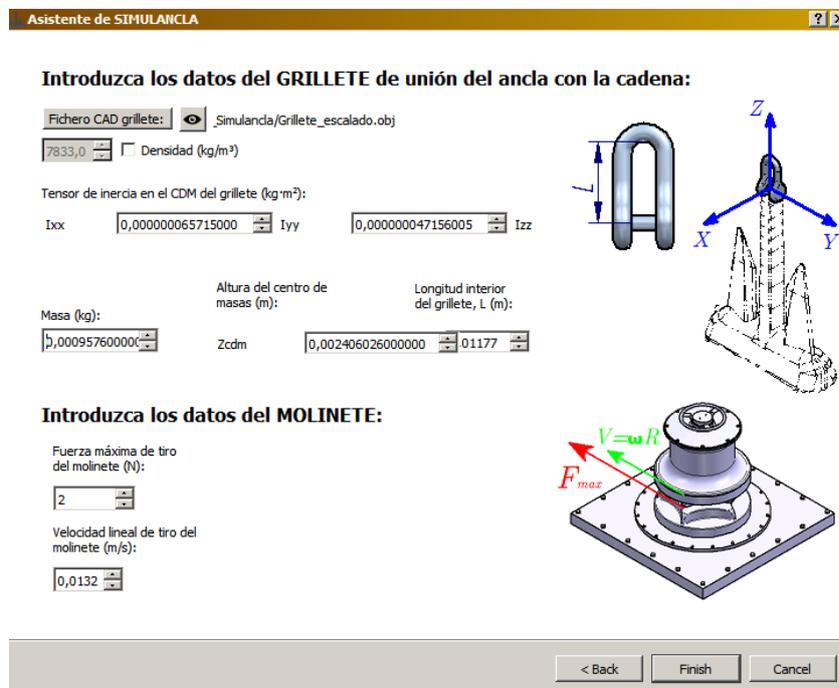


Figura 5-20. Parámetros de simulación - Configuración del grillete y el molinete

- **Fuerza máxima de tiro del molinete:** 1,5

Se ha elegido este valor ya que es el que se ha configurado como valor máximo de tiro en el Arduino para evitar que alguna pieza pueda romper.

- **Velocidad lineal de tiro del molinete:** 0.0132

Se ha elegido así ya que es la velocidad más pequeña que podemos controlar con el encoder que disponemos. De esta forma podemos controlar mejor la maniobra.

- Para la obtención de las coordenadas y longitudes se utiliza el mismo método explicado en puntos anteriores.

Como parte final del proceso de ensamblaje, se pulieron las diversas piezas para lograr un acabado que permitiese un giro sin resistencia ya que es así como se supone en la simulación y para lograr su buen encaje entre ellas. El resultado de todo este trabajo puede verse en las siguientes imágenes.



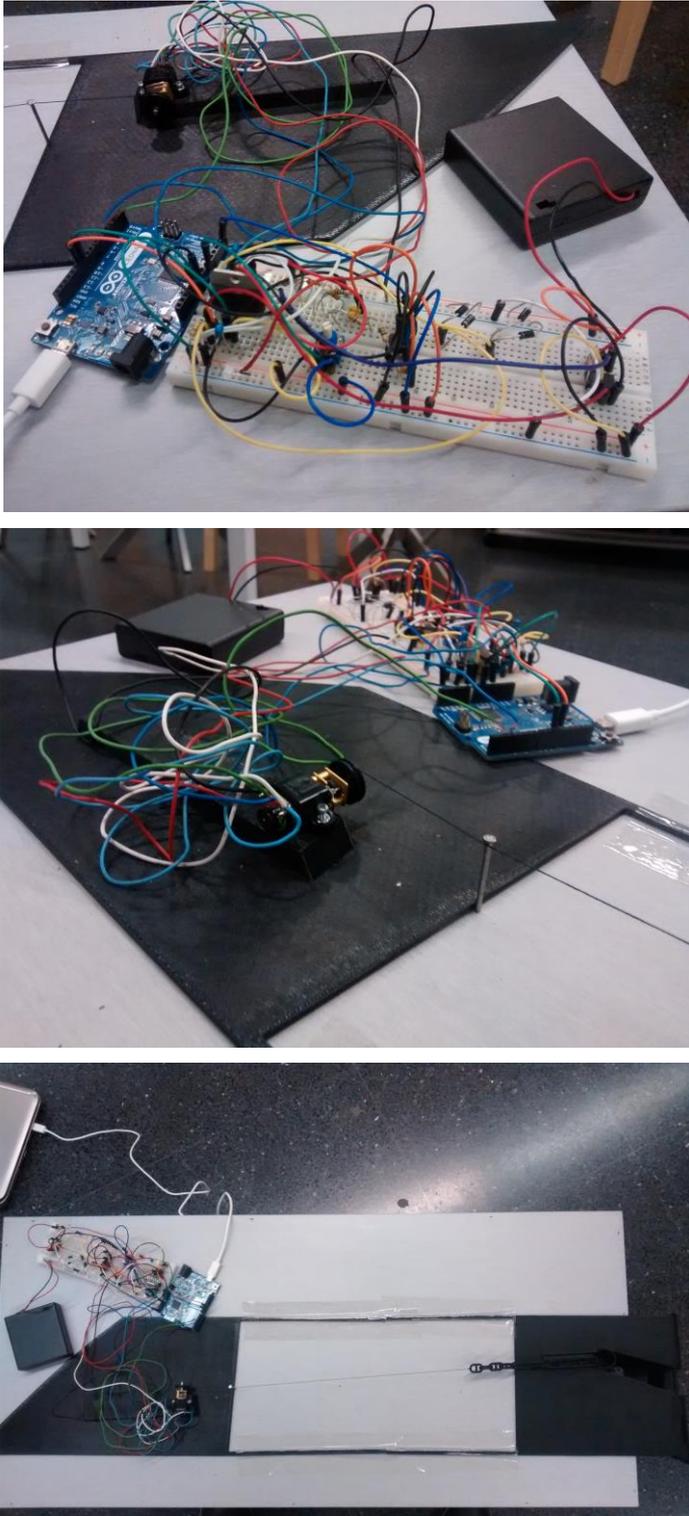


Figura 5-21. Partes del montaje

Obteniendo como resultado final general el que se muestra en la Figura 5-22.



Figura 5-22. Montaje final

Capítulo 6
Resultados
experimentales

6 RESULTADOS EXPERIMENTALES

En este capítulo se presentan los dos experimentos realizados de la maniobra de fondeo, empezando desde la posición inicial, que se sitúa a una distancia de la punta de las uñas del ancla al suelo de unos 165 mm, hasta la posición final, que se alcanza cuando el ancla queda estibada. Además, a los datos obtenidos se les aplicó una media móvil para contrarrestar los problemas de baja resolución del encoder y baja velocidad del experimento mencionados en el apartado 3.2.4.

En el primer experimento se puso la tapa transparente desarrollada para el escobén. Después de procesar los datos, empleando el método y la ecuación obtenida en el apartado 3.2.4, se obtiene la Figura 6-1. La fuerza de tiro resulta de dividir el valor de par por el radio de la polea.

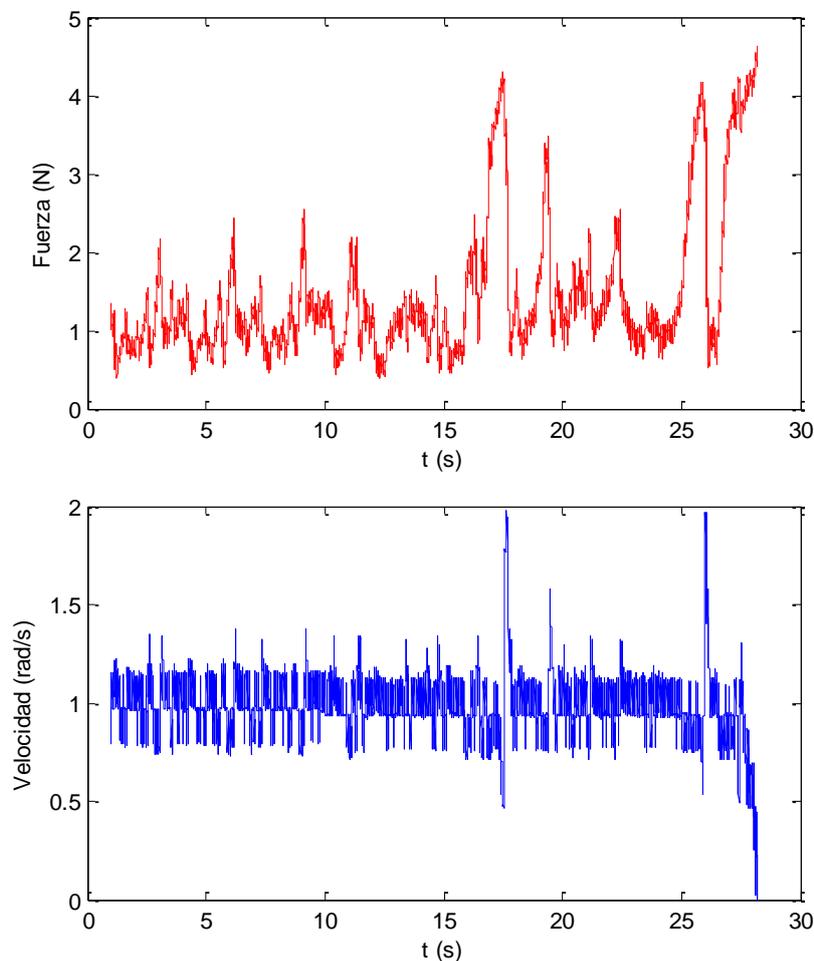


Figura 6-1. Primer experimento

En esta gráfica también se representa la velocidad, donde se comprueba que el controlador mantiene la velocidad constante alrededor de los 1 rad/s programados, actuando más intensamente en zonas con mayor rozamiento, como se puede observar alrededor del segundo 18, donde vemos que la velocidad baja y la fuerza sube para contrarrestar esta disminución, resultando después un pico de velocidad debido a la inercia. Cuando llega a la posición de estibado la velocidad se reduce a 0 rad/s.

En el segundo experimento no se puso la tapa transparente. Los resultados de fuerza de tiro junto a la velocidad se pueden ver en la Figura 6-2.

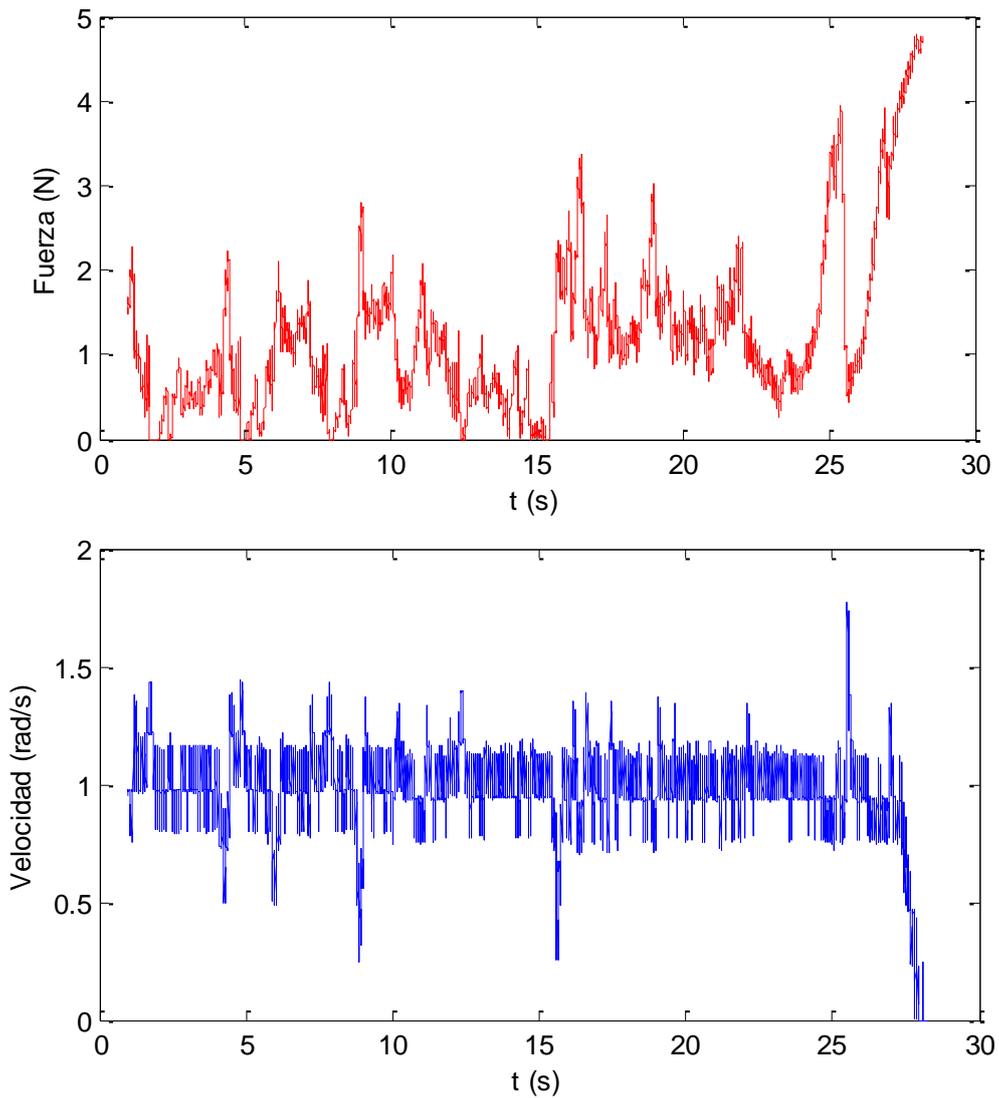


Figura 6-2. Segundo experimento

En la Figura 6 2 se puede apreciar como este experimento tiene menos dificultades para realizar la maniobra de fondeo, con fuerzas de tiro más bajas, lo que indica un menor rozamiento, seguramente debido a no tener el escobén totalmente cubierto, lo que lleva a que la cadena roce menos y tenga más libertad de maniobra.

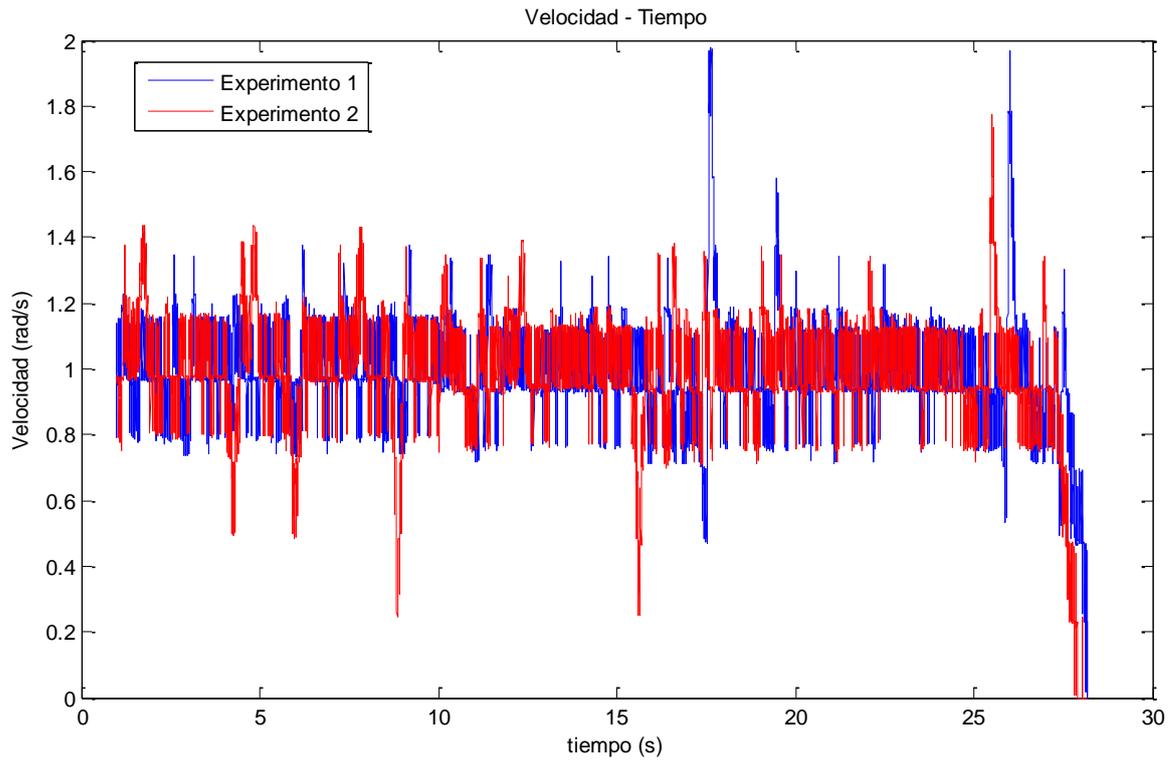


Figura 6-3. Comparación de velocidades entre experimentos

En la Figura 6-3 se puede apreciar como el experimento 1 es ligeramente más lento que el experimento 2, seguramente una combinación de la limitación de fuerza impuesta y mayor rozamiento en el experimento 1 que en el 2, lo que hace que al experimento 2 le lleve un poco más de tiempo superar la maniobra en el segundo 11 o 18.

La Figura 6-4 muestra los dos experimentos en un mismo gráfico, lo que facilita su comparativa.

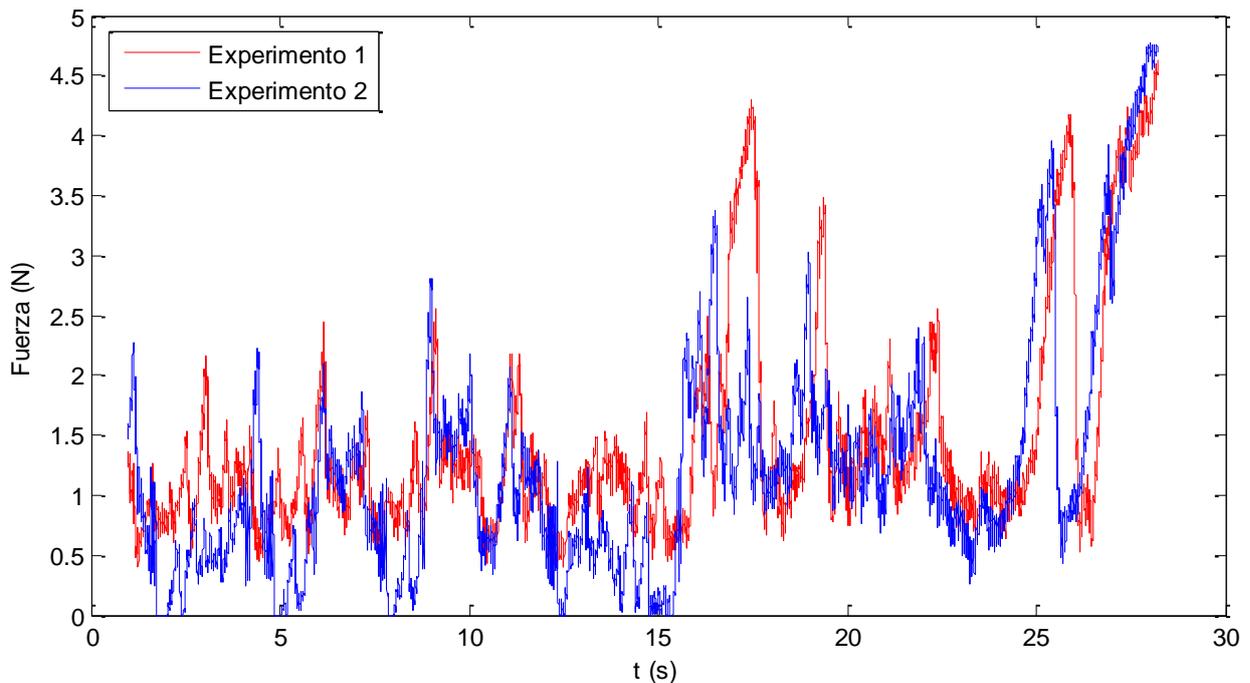


Figura 6-4. Comparativa entre Experimento 1 y 2

Se puede observar que los valores son similares (aunque cambian ya que uno tiene puesta la tapa al escobén y por lo tanto más rozamiento y, como se vio en la Figura 6-3, el experimento 1 es ligeramente más lento que el experimento 2, estando los valores desfasados). Los aumentos de la fuerza de tiro en zonas de atasco son causados por el controlador intentando que el motor siga girando con la velocidad indicada pero siendo incapaz de ello, lo que provoca que cada vez la tensión proporcionada al motor aumente hasta llegar a la máxima. También el hecho de que la velocidad se reduzca provoca que la fuerza contra electromotriz disminuya lo que causa otro aumento de intensidad.

Aunque el controlador tiene una condición para que se mantenga por debajo de los 1.5 N, como se puede apreciar en las zonas estables (sin atasco) de la Figura 6-4, en zonas con rozamiento suficientemente alto esto parece no cumplirse pero la razón se puede ver en el siguiente gráfico

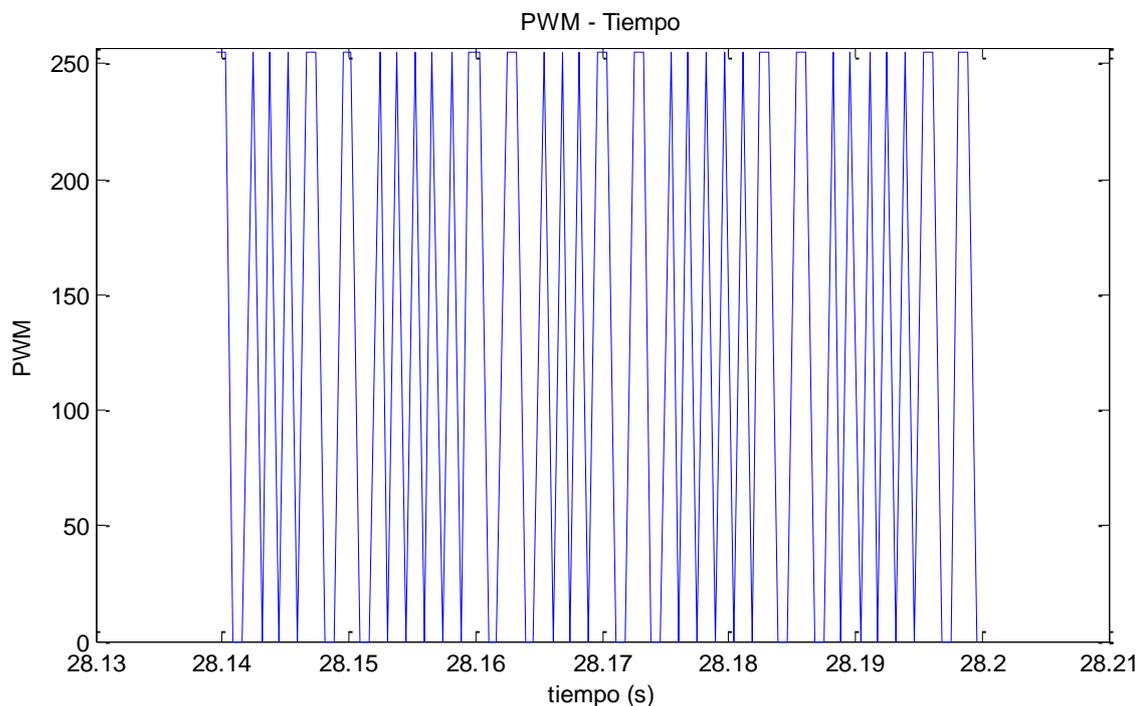


Figura 6-5. Valor del PWM en zonas de alto rozamiento

La Figura 6-5 representa cómo se comporta el PWM en zonas con alto rozamiento como, en este caso representa el valor hacia el final de la maniobra, cuando está alcanzando su posición de estibado. Cuando el valor de fuerza está por debajo de 1.5 N el controlador, como la diferencia con la velocidad deseada y la actual es grande, le da un valor alto de PWM. En este caso es tan grande la diferencia que el valor dado es el máximo, 255. Cuando el valor de fuerza supera los 1.5 N, el controlador deja de darle tensión y para ello le da el valor de PWM mínimo, 0. El valor variará entonces entre los valores máximo y mínimo, es decir, o adquiere el valor de 0 o el de 255, lo que provoca que el efecto fuese el de un PWM a la misma frecuencia a 127 de valor (mitad del máximo de potencia), y haciendo que el valor de la fuerza efectiva supere los 1.5 N, llegando a 4.8 N.

La solución a este problema se deja como trabajo futuro, con la implementación de un valor máximo de PWM variable, que venga dado por una intensidad máxima (la intensidad que equivale a los 1.5 N de fuerza, 0.0205 A) y la velocidad medida en el momento. Así, cuando la velocidad sea baja, el valor de PWM máximo permitido será pequeño (no 255) de forma que la fuerza no supere los 1.5 N. Cuando el valor de la velocidad sea más alto el

valor de PWM máximo permitido también aumentará de forma acorde. Una aproximación inicial a esta expresión basándonos en las ecuaciones del motor eléctrico (Wildi 2005) podría ser la siguiente:

$$\begin{aligned} I_{max} \cdot R &\geq PWM_{max} - k_i \cdot \omega \\ PWM_{max} &\leq I_{max} \cdot R + k_i \cdot \omega \end{aligned} \quad (6.1)$$

Siendo

- I_{max} los 0.0205 A indicados antes
- R la resistencia interna del motor
- PWM_{max} el límite máximo permitido para el PWM que irá variando según varíe la velocidad de giro
- k_i una constante a calcular. Cuanto más alta esta variable más efecto tiene la velocidad en el cálculo de esta PWM
- ω la velocidad de giro del motor

Capítulo 7

Conclusiones

7 CONCLUSIONES

El objetivo de este proyecto es la validación del software SIMULANCLA. Para cumplirlo se han realizado las siguientes tareas en el transcurso del proyecto:

- Se desarrolló un modelo a escala de un buque real para poder imprimir a partir de él la maqueta real.
- Se contactó con varias empresas de fabricación mediante impresión 3D para comprobar la viabilidad de la fabricación y comparar diferentes presupuestos.
- Cuando se obtuvieron los presupuestos iniciales, se hicieron diferentes modificaciones sobre el modelo inicial a fin de abaratar la fabricación y hacerla viable.
- Una vez elegida la empresa que se encargaría de la impresión después de comparar los diferentes presupuestos, se hicieron las modificaciones pertinentes para llevar a cabo la fabricación de la maqueta a escala del buque y eso implicó rediseñar y simplificar otra vez los diseños originales.
- Se diseñó y se fabricó el sistema de instrumentación basado en una placa de desarrollo Arduino y se llevó a cabo la calibración experimental del sistema, lo que permitió la obtención de los resultados experimentales de la fuerza de tiro.
- Se realizó un programa para el control de velocidad con el Arduino, lo que permite comparar los resultados con aquellos obtenidos de la simulación.
- Se diseñó y fabricó el sistema de tiro: motor, poleas, hilo, etc, lo que permitió la ejecución de la maniobra que se quería simular.
- Se preparó y llevó a cabo la simulación de la maqueta para poder comparar estos resultados con los experimentales. No se pudieron obtener resultados de las diversas simulaciones debido a que estas no llegaban al final, teniendo salidas no controladas. Esto fue debido al gran cambio de escala, ya que este programa fue concebido para simular buques reales estos difieren mucho en masa y tamaño con respecto a nuestra maqueta. A lo largo de este proyecto se intentaron arreglar los problemas que fueron surgiendo pero no se logró una solución definitiva en el momento de presentación del trabajo.
- Se llevaron a cabo los experimentos con la maqueta recabando los datos con el sistema de adquisición fabricado.

Por lo tanto se consiguió elaborar una maqueta sobre la que obtener los resultados experimentales, diseñar y montar el sistema de control y medición, el sistema de tiro, llevar a cabo los experimentos y obtener sus resultados. Sólo se consiguió parcialmente la simulación de la maqueta ya que aunque se preparó y ejecutó la simulación, no se pudieron obtener resultados debido a que no se llegaba al final, teniendo salidas no controladas.

Se deja para el futuro la comparación sistemática de resultados experimentales con los simulados, implementación de un control de la señal de PWM máxima según la velocidad y la obtención de distintas rectas de calibración para diferentes velocidades para aumentar el rango de uso del sistema.

Bibliografía

BIBLIOGRAFÍA

Analog devices. «AD623 Datasheet.» 2017. <http://www.analog.com/media/en/technical-documentation/data-sheets/AD623.pdf>.

Arduino. «Arduino Board Leonardo Documentation.» 2017. https://www.arduino.cc/en/Main/Arduino_BoardLeonardo.

Arduino. «Arduino Software (IDE).» 2017. <https://www.arduino.cc/>.

Arduino. «PWM Documentation.» 2017. <https://www.arduino.cc/en/Tutorial/PWM>.

Blender Foundation. «Blender.» 2017. <https://www.blender.org/>.

Bruce, C., y R. Mancini. «Op Amps for Everyone.» En *Op Amps for Everyone*, 16-1 to 16-64. 2001.

Carro, D. «Modelado 3D del sistema de fondeo de un barco para la fabricación de maquetas y simulación dinámica con el software SIMULANCLA.» 2016.

Comas, E. «EQUIPOS Y SERVICIOS, Volumen II. Fondeo, Amarre y Remolque.» 1990.

Cuadrado, J. *Cinemática y dinámica de máquina y mecanismos por computador*. Universidad de La Coruña, 1999.

Cuadrado, J., J. Cardenal, P. Morer, y E. Bayo. «Intelligent simulation of multibody dynamics: Space-state and descriptor methods in sequential and parallel computing environments.» (*Multibody System Dynamics*) 4, nº 1 (2000): 55-73.

Dopico, D. «Simulancla.» 2018.

Dopico, D. «Software para el análisis computacional del diseño de sistemas de fondeo en buques de gran tamaño.» Ferrol, 2015.

Dopico, D., A. Luaces, M. Gonzalez, y J. Cuadrado. «Dealing with multiple contacts in a human-in-the-loop application.» (*Multibody System Dynamics*) 25, nº 2 (2011): 167-183.

- Dopico, D., F. González, J. Cuadrado, y J. J. & Kovecses. «Determination of Holonomic and Nonholonomic Constraint Reactions in an Index-3 Augmented Lagrangian Formulation With Velocity and Acceleration Projections.» (Journal of Computational and Nonlinear Dynamics, ASME) 9, nº 4 (2014): 041006-041006.
- Flores, P., M. Machado, M. Silva, y J. Martins. «On the continuous contact force models for soft materials in multibody dynamics.» (Multibody System Dynamics) 25 (2011): 357-375.
- Galeano, N., J. Quintero, y J. Lezama. «Enseñando el Funcionamiento de los Inversores Puente H: Análisis del Intercambio de Potencia entre Bobinas y Condensadores.» 2016.
- Gonthier, Y., J. McPhee, C. Lange, y J.-C. Piedboeuf. «A regularized contact model with asymmetric damping and dwell-time dependent friction.» (Multibody System Dynamics) 11, nº 3 (2004): 209–233.
- Hunt, K., y F. Crossley. «Coefficient of restitution interpreted as damping in vibroimpact.» (Journal of Applied Mechanics, Transactions ASME) 42, nº 2 (1975): 440–445.
- International Organization for Standardization. «International Standard ISO 1704.» 1973.
- Laboratorio de Ingeniería Mecánica.* s.f. lim.ii.udc.es/.
- Mathworks. «Matlab.» 2014. <https://www.mathworks.com/products/matlab.html>.
- McNeel, R. «Rhinoceros 3D.» 2013.
- Olshausen, B. «Aliasing.» *PSC 129 - Sensory Processes.* 2000.
- Pololu. «Micro Metal Gearmotor LP 6V Documentation.» 2017. <https://www.pololu.com/product/2208/>.
- Pololu. «Magnetic encoder Pair Kit for Micro Metal Gearmotors Documentation.» 2017. <https://www.pololu.com/product/3081>.
- Siemens. «Solid Edge ST8.» 2015. <http://www.solidedge.es/>.

BIBLIOGRAFÍA

MARTA FRADE MANSO

Sparkfun Electronics. «L298 Datasheet.» 2017.
https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf.

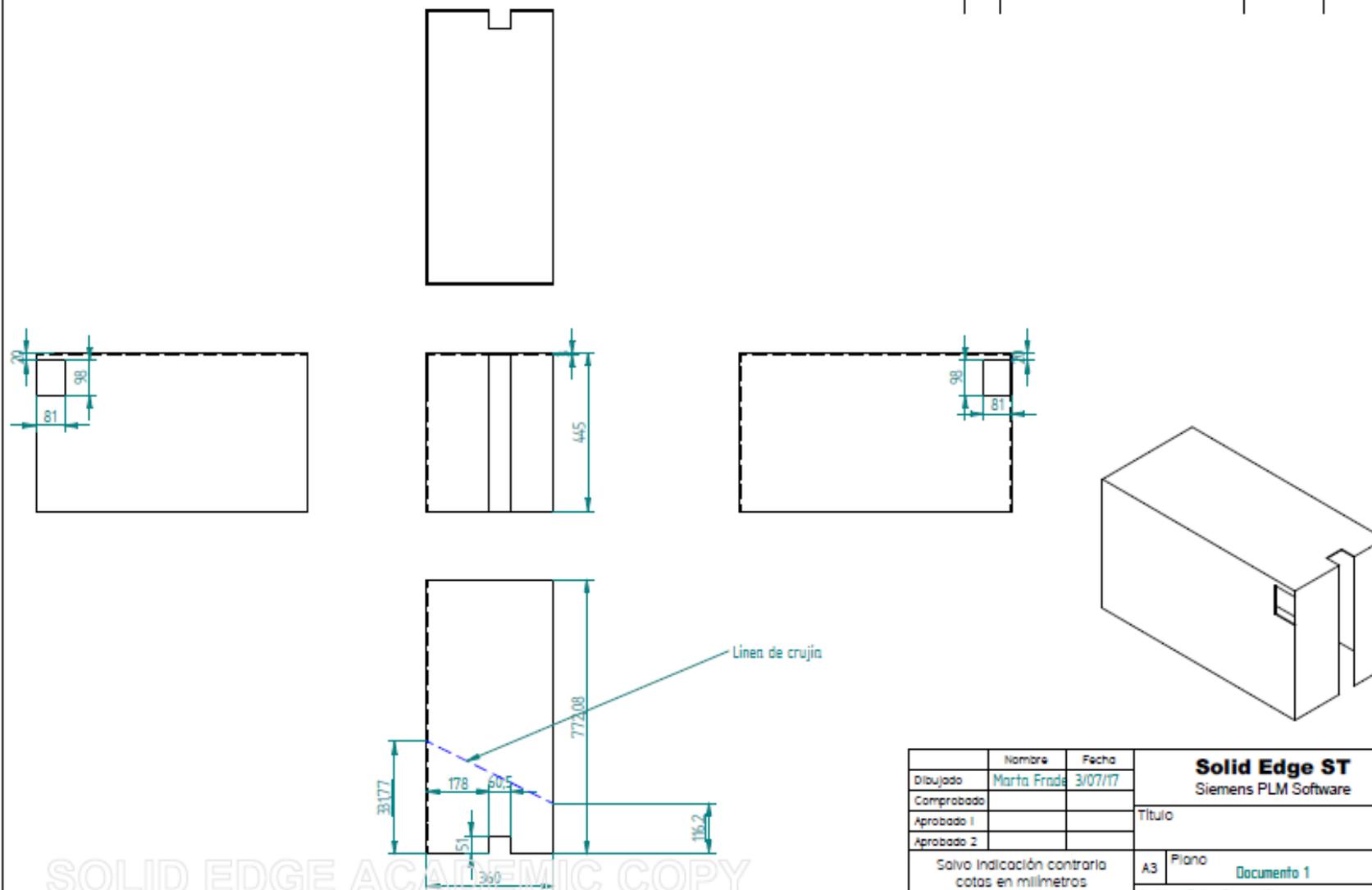
Wildi, T. «Electrical machines, Drives, and Power Systems Sixth Edition.» Pearson, 2005.

Anexos

ANEXOS

Anexo I: Plano diseño apoyo de la maqueta

Revisiones			
Rev	Descripción	Fecha	Aprobado



SOLID EDGE ACADEMIC COPY

Nombre	Fecha	Solid Edge ST	
Dibujado: Marta Frade	3/07/17	Siemens PLM Software	
Comprobado		Título	
Aprobado 1		A3 Plano Documento 1	
Aprobado 2		Rev 1	
Salvo indicación contraria cotas en milímetros ángulos en grados tolerancias ±0,5 y ±1°		Archivo: Caja_Barca.dft	
Escala	Peso	Hoja 1 de 1	

Anexo II: Código Arduino

```
// con estos comandos incluimos estas bibliotecas (Encoder y studio) para
// poder usar las funciones que se ven en // este programa
#include <Encoder.h>
#include <stdio.h>

// Modos de frecuencia definidos para TIMER1
#define PWM62k 1 //62500 Hz
#define PWM8k 2 // 7812 Hz
#define PWM1k 3 // 976 Hz
#define PWM244 4 // 244 Hz
#define PWM61 5 // 61 Hz

// Variables para el cambio directo del timer de los pines 9,10 y 11
#define PWM9 OCR1A
#define PWM10 OCR1B
#define PWM11 OCR1C

// Configuración del reloj del PWM
// El argumento es uno de los cinco modos definidos antes
void pwm91011configure(int mode)
{
    // Configuración TCCR1A
    TCCR1A = 1;

    // Configuración TCCR1B
    //
    TCCR1B = mode | 0x08;

    // Configuración TCCR1C
    TCCR1C = 0;
}

// Establecer PWM a D10
// Argumento es PWM entre 0 y 255
void pwmSet10(int value)
{
    OCR1B = value; // Establece el valor de PWM
    DDRB |= 1 << 6; // Establece Output Mode B6
    TCCR1A |= 0x20; // Activa canal
}

// Establecer PWM a D11
// Argumento es PWM entre 0 y 255
void pwmSet11(int value)
{
    OCR1C = value; // Establece el valor de PWM
    DDRB |= 1 << 7; // Establece Output Mode B7
    TCCR1A |= 0x08; // Activa canal
}

// Aquí se definen los pines donde llega la información de la posición del
// encoder del motor.
// Se eligen estos pines porque en el arduino leonardo son los que tienen
// la función interrupt y la biblioteca
// encoder.h funciona de forma óptima cuando ambos pines tienen función
// interrupt.
Encoder enc_motor(2, 3);
```

```
// inicialización de la variable que representa velocidad real del motor
que la obtenemos con el encoder del motor
float velocidad_real = 0;
// inicialización de la variable que representa velocidad al que se quiere
// que vaya el motor,
float velocidad_real_filtrada = 0;
float velocidad_deseada = 0;
// inicialización de la variable que representa ángulo del motor que
obtenemos con el encoder y que se usa para calcular la velocidad
float angulo_real = 0;
// inicialización de la variable donde guardamos el ángulo del motor del
// instante anterior, para así calcular la diferencia con el actual, y
// dividiéndolo entre la diferencia de tiempos obtenemos la velocidad.
double angulo_real_original = 0;
// inicialización de la variable donde guardamos el error, que se expresa
como la diferencia entre la velocidad deseada y la real
float error = 0;
// inicialización de la variable donde guardamos la suma de error para la
// parte integral del PID
float error_suma = 0;
// variable que representa la constante del proporcional del controlador
// PID.
int kp = 2;
// inicialización de la variable PWM
long pwm = 0;
// valor del pin donde conectamos M1
int in1 = 11;
// valor del pin donde conectamos M2
int in2 = 10;
// variable donde guardamos el valor del potenciómetro.
int Pot = 0;
int Pot_old = 0;
int flag = 0;
// variable donde guardaremos la tensión de salida filtrada
int v_salida = 0;
int v1 = 0;
int v2 = 0;
// variable donde guardamos el valor antiguo del error para calcular el
// error de la nueva iteración
float error_old = 0;
// valor de la constante pi
const float pi = 3.14159;

float t_salida = 0;
// aquí se configuran los pins y la tasa de transmisión. Solo se realiza
// una vez al principio, y cada vez que se resetea el arduino.
void setup()
{
  // Se configura el Timer 1 (Pins 10 y 11)
  // A 62kHz
  pwm91011configure(PWM62k);
  // Se preparan los pin 10 & 11 para usar PWM
  pwmSet10(0);
  pwmSet11(0);
  // se establece la tasa de transmisión
  Serial.begin(115200);
  // En esta línea se configuran los pins de salida. Se eligen estos porque
  // son PWN y por lo tanto permiten emular una salida analógica.
  // Esto es necesario ya que queremos que la salida de estos pins
  // comunique distintos pulsos, no solo un valor (0 o 1)
  pinMode(in1, OUTPUT);
```

```

pinMode(in2, OUTPUT);
// En esta línea se configura el pin A0 y A5 como de entrada. Este pin no
// se tiene que declarar como los anteriores ya que el arduino ya
// reconoce este nombre como nombre del pin.
pinMode(A0, INPUT);
pinMode(A5, INPUT);
Pot=analogRead(A0);
Pot_old= Pot;
}
// se inicializa la variable oldposition que se va a usar para saber la
// posición del encoder
long oldPosition1 = -999;
// inicialización de variables para guardar el instante de tiempo actual y
// el anterior.
unsigned long t = 0;
unsigned long t_old = 0;
unsigned long dt = 0;
unsigned long t_ms = 0;
// A partir de aquí es donde empieza el programa. Está en continuo
// funcionamiento hasta que se quite la corriente.
void loop()
{
// Aquí guardamos en la variable t el numero de microsegundos que ha
// pasado desde que el arduino empezó a ejecutar este programa.
t = micros();
Pot=analogRead(A0);
if (Pot - Pot_old >0)
{flag=1;}
Pot_old= Pot;

// En esta línea se comprueba si han pasado 714 microsegundos entre una
// iteración del programa y otra y se indica que se quiere iniciar
// el programa con el potenciómetro (la flag se pondría a 1). Sino no
// ejecuta. Así obtenemos un cálculo de velocidad constante cada 714
// microseg y controlamos cuando queremos que empiece.
if ((t - t_old >= 714) && (flag ==1))
// Aquí empieza lo que hemos llamado una iteración de programa.
// Empieza leyendo el valor del encoder del motor.
{
// En esta línea se lee el valor actual del encoder
long newPosition1 = enc_motor.read();
// En esta línea se guarda el valor antiguo del ángulo, para después
//poder calcular la diferencia.
angulo_real_original = angulo_real;
// En esta línea se convierte el valor de posición en pulsos a
// radianes.
angulo_real = (newPosition1 * 2 * pi) / (12 * 298);
// En esta línea se obtiene el valor de la velocidad del motor
// calculando la diferencia entre el valor del ángulo
// actual y el del instante anterior y dividiéndolo por los 714
// microsegundos. Se multiplica por 1000000 para
// convertirlo en segundos
velocidad_real = (angulo_real - angulo_real_original) * 1000000.0 / (t
- t_old);
// En esta línea se obtiene un valor promediado de la velocidad para
// intentar paliar los efectos de la baja resolución del encoder
velocidad_real_filtrada = velocidad_real_filtrada * 0.3 +
velocidad_real * 0.7;
// En esta línea se guarda el valor de t antiguo para volver a contar
// 714 microsegundos desde el inicio de esta iteración
dt = t - t_old;

```

```
t_old = t;
// En esta línea se indica la velocidad que quiere que siga el motor
//(en rad/s)
velocidad_deseada = 1;
// En esta línea calculamos el parámetro para la parte proporcional del
// PID que multiplica a la constante k_p
error = velocidad_deseada + velocidad_real_filtrada;
// En esta línea calculamos el parámetro para la parte integral del PID
// que multiplica a la constante k_i
error_suma = error_suma + error;
// En esta línea se calcula el valor del PWM
pwm = (error_suma * 0.5) + (kp * error);
// con esta función restringimos los valores de pwm para que vayan de 0
// a 255
pwm = constrain(pwm, 0, 255);
// En esta línea se pasa el valor de microsegundos a milisegundos para
// imprimir un valor más pequeño por pantalla
t_ms = t / (1000);
// Aquí se lee el valor de la tensión en bornes de la resistencia
v_salida = analogRead(A5);
//esta línea limita la fuerza de tiro que puede sufrir la cadena
if (v_salida > 85)
{
    pwm = 0;
}
// Se asigna el valor de pwm a los pines. Se pueden usar las funciones
// pwmSet11 y pwmSet10 pero las macros PWM11 y PWM10 son más rápidas de
// ejecutar
PWM11 = pwm;
PWM10 = 0;
// Se imprime por pantalla el valor de la tensión, el tiempo y la
// velocidad
String stringOne = String(String(v_salida) + "/" + t_ms + "/" +
velocidad_real_filtrada + "/");
// función que imprime por pantalla la variable stringOne. Esto nos
// servirá para que el script de matlab lo pueda leer
Serial.println(stringOne);
}
}
// Aquí finaliza el programa
```
