

Universidad de La Laguna
ESCUELA POLITÉCNICA SUPERIOR DE INGENIERÍA
Sección Náutica, Máquinas y Radioelectrónica Naval

**Trabajo presentado para
la obtención del título de**

GRADUADO EN TECNOLOGÍAS MARINAS

Presentado por

José Daniel Álamo Pérez y Gabriel Seco Herrera

Uso y adaptación de un sensor de movimiento para
la conversión de embarcaciones con motor de
combustión a embarcaciones eléctricas

Dirigido por

Carlos Efrén Mora Luis

Presentado en Septiembre 2018





INFO ABOUT RIGHTS
1 809028 251697
www.safecreative.org/work

Resumen

Uno de los problemas a la hora de convertir un sistema de propulsión convencional a otro eléctrico basado en baterías es cómo conservar el comportamiento estático y dinámico de la embarcación. El nuevo reparto de pesos debe hacerse de tal forma que pueda mantenerse la estabilidad de la embarcación (altura metacéntrica), como sus características dinámicas, especialmente en el movimiento de balance (momento de inercia transversal).

El presente trabajo describe el desarrollo de un dispositivo basado en un sensor de movimiento (unidad inercial), que permite medir los ángulos y el periodo de balance de una forma precisa. Los ángulos y los periodos de balance obtenidos permiten determinar de forma rápida la altura metacéntrica y el momento de inercia, facilitando la decisión de dónde colocar las baterías para tratar de mantener intacta la navegabilidad de la embarcación.

El dispositivo que se ha desarrollado consta de un Arduino Nano, una unidad de medición inercial MPU6050, un display LCD, y módulo para guardar los datos obtenidos en una tarjeta SD, para posteriormente poder hacer los cálculos de forma externa. Para poder testear y comprobar el correcto funcionamiento del dispositivo se ha diseñado un banco de pruebas, mediante el que se han podido validar los ángulos y los periodos medidos por el dispositivo. Los resultados obtenidos muestran la idoneidad del dispositivo para su uso en embarcaciones reales, si bien requiere de un posterior desarrollo que permita hacer los cálculos de estabilidad en campo, sin necesidad de extraer los datos de la tarjeta SD.



INFO ABOUT RIGHTS
1 809028 251697
www.safecreative.org/work

Abstract

One of the problems when converting a conventional propulsion system to an electric one based on batteries is how to preserve the static and dynamic behavior of the boat. The new distribution of weights must be done in such a way that the stability of the vessel (metacentric height) can be maintained, as well as its dynamic characteristics, especially in the balance movement (transverse moment of inertia).

The present work describes the development of a device based on a motion sensor (inertial unit), which allows to measure the angles and the balance period in an accurate way. The angles and the balance periods obtained allow to quickly determine the metacentric height and the moment of inertia, facilitating the decision of where to place the batteries to try to keep intact the navigability of the boat.

The device that has been developed consists of an Arduino Nano, an inertial measurement unit MPU6050, an LCD display, and a module for saving the data obtained on an SD card, to be able to make calculations externally. In order to test and verify the proper functioning of the device, a test bench has been designed, through which the angles and periods measured by the device have been validated. The obtained results show the suitability of the device for use in real boats, although it requires a further development that allows to perform stability calculations in the field, without the need to extract the data from the SD card.



INFO ABOUT RIGHTS
1 809028 251697
www.safecreative.org/work

Índice general

Lista de figuras	IX
Lista de tablas	XI
1. Fundamentación teórica	1
1.1. Determinación de la posición de una embarcación	1
1.2. Ángulos de balance y cabeceo	1
1.3. Ángulos de Euler y matriz de giro	2
1.4. Elementos de lectura de aceleración tangencial y de velocidad angular	4
1.4.1. Acelerómetro	4
1.4.2. Giroscopio	5
1.5. Fuerzas que actúan en la estabilidad transversal	7
1.6. Determinación del centro de masas	7
1.7. Altura metacéntrica	8
1.7.1. Condición de equilibrio y metacentro	8
1.7.2. Altura metacéntrica transversal	9
1.8. Momento adrizante y sólido de revolución	10
1.9. Momento de inercia y periodo doble de balance	11
1.10. Parámetros de estabilidad estática y dinámica	12
2. Desarrollo del dispositivo	17
2.1. Hardware	17
2.1.1. Arduino Nano	17
2.1.2. Unidad de medición inercial	17
2.1.3. Módulo de tarjeta SD	18
2.2. Desarrollo del software	18
2.2.1. Arduino IDE	19
2.3. Diagrama de flujo / Algoritmo	19
2.3.1. Calibración de la unidad inercial	20
2.3.2. Algoritmo de cálculo para la obtención de los ángulos de la IMU	21
2.3.3. Filtrado de la señal	23
2.3.4. Procedimiento de guardado de datos en la SD	24

3. Validación	25
3.1. Casco para llevar a cabo las pruebas de balance	25
3.2. Montaje de maqueta para pruebas	25
3.3. Montaje del banco de pruebas	27
3.4. Obtención del periodo de balance	27
3.5. Cálculos	28
3.6. Resultados	29
3.6.1. Determinación del periodo doble de balance	29
3.6.2. Valores de altura metacéntrica	32
3.6.3. Valores de momento de inercia	32
3.6.4. Valores de factor de corrección	33
4. Discusión de los resultados	35
5. Conclusiones	37
Bibliografía	39
Anexos	43
A. Código Fuente	43
B. Esquema de conexiones	51



Índice de figuras

1.1. Representación del sistema de coordenadas.	1
1.2. Vectores de velocidad lineal y angular de la embarcación.	2
1.3. Ángulos de Euler.	3
1.4. Esquema de acelerómetro capacitivo.	4
1.5. Acelerómetro mecánico.	5
1.6. Giroscopio MEMS.	6
1.7. Vista operativa interna de un sensor giroscópico MEMS.	6
1.8. Desplazamiento del centro de carena.	9
1.9. Brazo adrizante.	10
2.1. Unidad inercial de 3 ejes MPU6050.	18
2.2. Diagrama de flujo general.	20
2.3. Información del proceso de calibración a través del puerto serie.	21
2.4. Diagrama de flujo para la determinación del ángulo de escora.	22
3.1. Proceso de impresión 3D.	25
3.2. Perspectiva del casco diseñado.	26
3.3. Espacios entre cuadernas del casco.	26
3.4. Montaje final del banco de pruebas.	28
3.5. Montaje para cálculo de periodo con la IMU.	29
3.6. Amortiguación del balance para el casco libre de pesos (toma 3).	30
3.7. Amortiguación del balance para pesos laterales a 0,5 cm de la banda (toma 3).	31
3.8. Amortiguación del balance para pesos laterales a 2 cm de la banda (toma 2).	31
3.9. Obtención de parámetros de estabilidad estática.	32
B.1. Esquema de conexiones.	52



INFO ABOUT RIGHTS
1 809028 251697
www.safecreative.org/work

Índice de tablas

1.1. Nomenclatura utilizada para describir el movimiento del buque.	2
1.2. Periodos según tipo de barco.	12
2.1. Características de placa Arduino Nano[1].	17
2.2. Conexiones entre MPU6050 y Arduino Nano.	18
3.1. Características de elementos añadidos a la maqueta.	27
3.2. Diferencias de periodo entre escala e IMU sin peso en la maqueta.	30
3.3. Diferencias de periodo entre escala e IMU con tuercas separadas 0,5cm del casco. . .	30
3.4. Diferencias de periodo entre escala e IMU con tuercas separadas 2cm del casco. . . .	31
3.5. Resultados de altura metacéntrica.	32
3.6. Resultados de radio de giro equivalente.	33
3.7. Resultados de momento de inercia.	33
3.8. Resultados de factor de corrección f para la determinación aproximada del momento de inercia.	33
3.9. Resultados de factor de corrección en milímetros.	33



INFO ABOUT RIGHTS
1 809028 251697
www.safecreative.org/work

Simbología

E	Eslora
Epp	Eslora entre perpendiculares
Et	Eslora total
B	Manga
Mfl	Manga máxima de flotación
LF	Línea de flotación
C	Calado
Cpp	Calado de popa
Cpr	Calado de proa
A	Asiento
C	Centro de carena
GM	Altura metacéntrica
G	Centro de gravedad
M	Metacentro



INFO ABOUT RIGHTS
1 809028 251697
www.safecreative.org/work

Acrónimos

FET	Field-Effect Transistor (transistor de efecto campo)
MEMS	Micro Electro-Mechanical System (sistema micro electro-mecánico)
PWM	Pulse-Width Modulation (modulación por ancho de pulsos)
IMU	Inertial Measurement Unit (unidad de medición inercial)
MPU	Micro-Processing Unit (unidad de micro-procesamiento)
IDE	Integrated Development Environment (entorno de desarrollo integrado)



INFO ABOUT RIGHTS
1 809028 251697
www.safecreative.org/work

1 Fundamentación teórica

1.1 Determinación de la posición de una embarcación

Dinámicamente se puede considerar una embarcación como un sólido rígido con seis grados de libertad, repartidos en dos sistemas de coordenadas ortogonales [4]. Uno de los sistemas será inercial, es decir, cumple con las leyes del movimiento de Newton, por lo que la variación del momento lineal en el sistema es igual a las fuerzas reales de este. La colocación de este primer sistema se considerará solidaria a un punto en tierra (x, y, z), permitiendo conocer el movimiento de traslación. El segundo sistema de coordenadas no es inercial y se coloca su origen en la embarcación (x', y', z'), haciendo que coincidan sus ejes de forma paralela a los ejes principales de inercia del sistema (x, y, z) ubicado en tierra, tal y como se observa en la figura (1.1). Este segundo sistema de coordenadas es en el que se producen los movimientos de rotación de la embarcación [5].

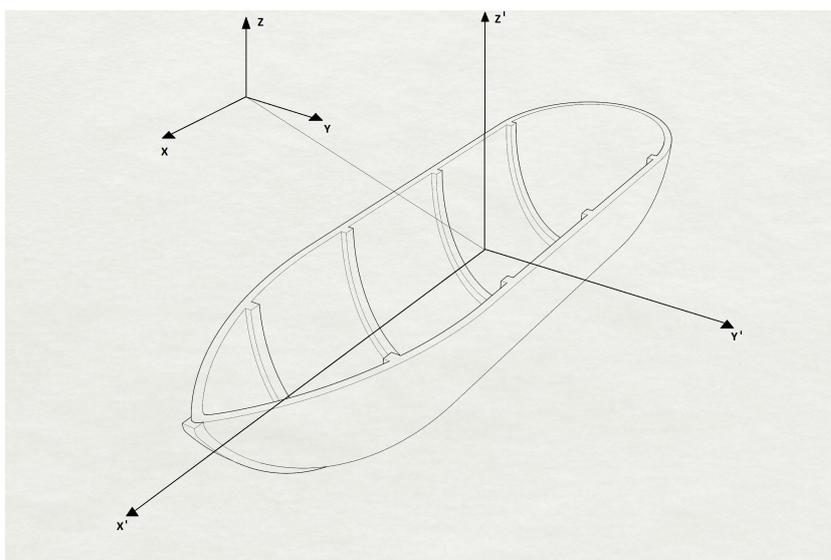


Figura 1.1: Representación del sistema de coordenadas.

Fuente: Elaboración propia

1.2 Ángulos de balance y cabeceo

Los movimientos de una embarcación se pueden dividir en rectilíneos y rotacionales. En este último grupo se puede encontrar movimientos de rotación continuos y otros que producen una oscilación. Los movimientos de rotación se ven representados mediante los ejes que se establecen en la embarcación, correspondiendo éstos con el segundo sistema de referencias de la sección 1.1. Los movimientos que se estudian en este trabajo son de carácter oscilatorio: el balance, que es el giro en el eje longitudinal del buque (x') y el cabeceo, que es el movimiento de rotación en el eje transversal (y') [5]. En la figura 1.2 se pueden apreciar ambos movimientos oscilatorios junto a sus ejes, incluyendo también el giro del eje vertical (z'), que se denomina guiñada, el cual no se trata de un movimiento oscilatorio.

La tabla 1.1 muestra la notación estándar para referenciar el movimiento de la embarcación, las fuerzas y los momentos [4].

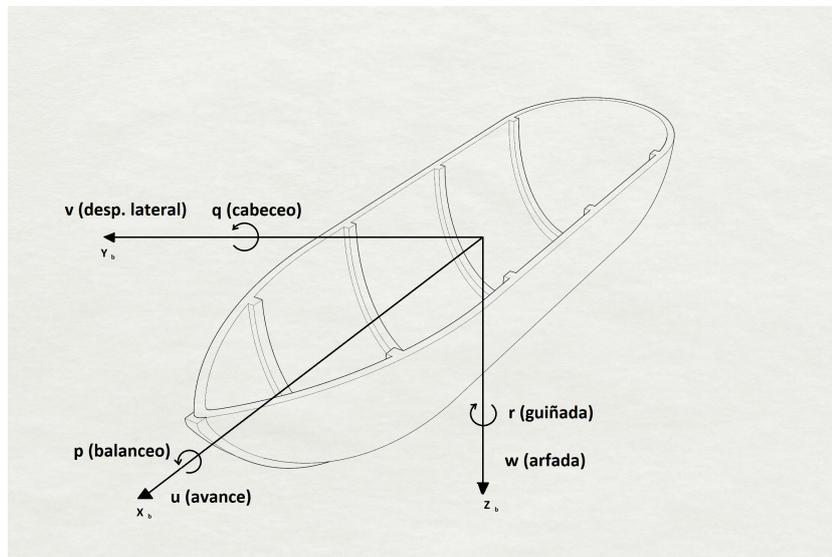


Figura 1.2: Vectores de velocidad lineal y angular de la embarcación.
Fuente: Elaboración propia

Traslación	Fuerza	Velocidad lineal	Posición
Avance	X	u	x
Desp. lateral	Y	v	y
Arfada	Z	w	z
Rotación	Momento	Velocidad angular	Ángulo
Balanceo	K	p	ϕ
Cabeceo	M	q	θ
Guiñada	N	r	ψ

Tabla 1.1: Nomenclatura utilizada para describir el movimiento del buque.
Fuente: Modelos matemáticos para el control de gobierno de buques [4].

1.3 Ángulos de Euler y matriz de giro

Los ángulos de Euler son tres ángulos que permiten describir la orientación de un cuerpo rígido con respecto a un sistema de coordenadas fijo [12]. Mediante los ángulos de Euler se especifica la orientación de un sistema de referencia de ejes ortogonales, normalmente móvil, respecto a otro sistema de referencia de ejes ortogonales normalmente fijo. Conociendo dos sistemas de coordenadas, (x, y, z) , y (X, Y, Z) , con un origen común, es posible determinar la posición de un sistema en función del otro utilizando tres ángulos, α , β , y γ .

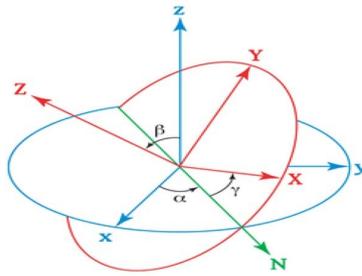


Figura 1.3: Ángulos de Euler.

Fuente: <https://commons.wikimedia.org/>

Los ángulos que se utilizarán para calcular la oscilación en una embarcación son los ángulos de Euler aplicados a la navegación: ángulo de balance (ϕ), ángulo de cabeceo (θ), y ángulo de guiñada (ψ), con los cuales se implementa la matriz de giro, que se trata de una herramienta matemática capaz de representar las rotaciones de manera concisa de un sistema de referencia móvil con relación a otro fijo[10]. Es muy útil debido al comportamiento aleatorio de la embarcación en el seno del fluido marino.

Se comienza calculando la matriz de giro de tres dimensiones, a través de la definición estándar de las rotaciones sobre los tres ejes principales [10].

Una rotación de ψ radianes alrededor del eje x se define como:

$$R_x(\psi) = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & -\sin \psi \\ 0 & \sin \psi & \cos \psi \end{vmatrix} \quad (1.1)$$

De manera similar, una rotación de θ radianes alrededor del eje y se define como:

$$R_y(\theta) = \begin{vmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{vmatrix} \quad (1.2)$$

Finalmente, una rotación de ϕ radianes alrededor del eje z se define como:

$$R_z(\phi) = \begin{vmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{vmatrix} \quad (1.3)$$

Estas tres matrices de cada eje se pueden representar en una única matriz que se establece con una secuencia de tres rotaciones, una sobre cada eje principal. Dado que la multiplicación de la matriz no se conmuta, el orden de los ejes sobre los que uno gira afectará el resultado. Por ello, se rota primero sobre el eje x, luego sobre el eje y, y finalmente sobre el eje z. Tal secuencia de rotaciones se puede representar como el producto matriz:

$$R = R_z(\phi)R_y(\theta)R_x(\psi) = \begin{vmatrix} \cos \theta \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \cos \theta \sin \phi & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi \\ -\sin \theta & \sin \psi \cos \theta & \cos \psi \cos \theta \end{vmatrix} \quad (1.4)$$

En el caso de las dos dimensiones es el único en que el grupo de matrices de rotación es conmutativo y tiene la forma siguiente:

$$R(\theta) = \begin{vmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{vmatrix} \quad (1.5)$$

Para rotar vectores columna, se opera de la forma siguiente:

$$\begin{vmatrix} x' \\ y' \end{vmatrix} = \begin{vmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix} \quad (1.6)$$

Las coordenadas (x', y') del punto (x, y) después de la rotación son:

$$x' = x \cos \theta - y \sin \theta \quad (1.7)$$

$$y' = x \sin \theta + y \cos \theta \quad (1.8)$$

La dirección del vector rotado se aprecia en el signo del ángulo proporcionado por la matriz, siendo ésta en sentido horario si el signo es negativo y antihorario si es positivo.

1.4 Elementos de lectura de aceleración tangencial y de velocidad angular

1.4.1 Acelerómetro

Un acelerómetro es un dispositivo que permite medir aceleraciones, es decir la variación de la velocidad por unidad de tiempo. Según la segunda ley de Newton ($F = m \cdot a$), teniendo en cuenta una masa constante (m), la resultante de la aceleración medida (a), será proporcional a la fuerza aplicada en el acelerómetro [3].

En los acelerómetros capacitivos (figura 1.4) entra en juego la Ley de la Elasticidad de Hooke ($F = k \cdot x$), donde (F) es la fuerza aplicada, (k) la constante de elasticidad del muelle y (x) el desplazamiento de la masa. Estos acelerómetros se diseñan de manera que una de las dos placas de los condensadores pueda moverse de forma proporcional a la fuerza externa. Mediante esto se obtienen variaciones en la señal eléctrica proporcionales a la fuerza aplicada y, por lo tanto, a la aceleración.

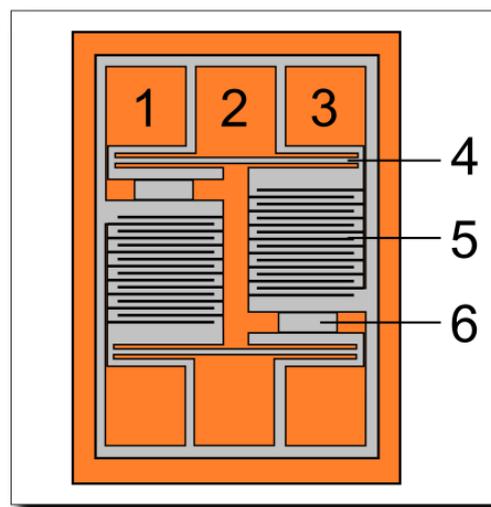


Figura 1.4: Esquema de acelerómetro capacitivo.

Fuente: <https://commons.wikimedia.org>

En un acelerómetro mecánico también entra en juego la Ley de la Elasticidad de Hooke, mencionada anteriormente. Existen dos tipos de acelerómetros mecánicos, de salida de carga de alta impedancia, y de baja impedancia.

En los acelerómetros de salida de carga de alta impedancia el cristal piezoeléctrico (figura 1.5) genera una carga eléctrica que está conectada directamente a los instrumentos de medición. El uso de este modelo de acelerómetros suele estar ligado a aplicaciones de altas temperaturas (superiores a 120 °C) en las que no se pueden utilizar modelos de baja impedancia.

Los acelerómetros de salida de baja impedancia incluyen un acelerómetro de carga en su extremo delantero, un pequeño microcircuito integrado y un transistor FET (de efecto de campo) el cual convierte la carga en una tensión de baja impedancia capaz de interactuar fácilmente con la instrumentación estándar.

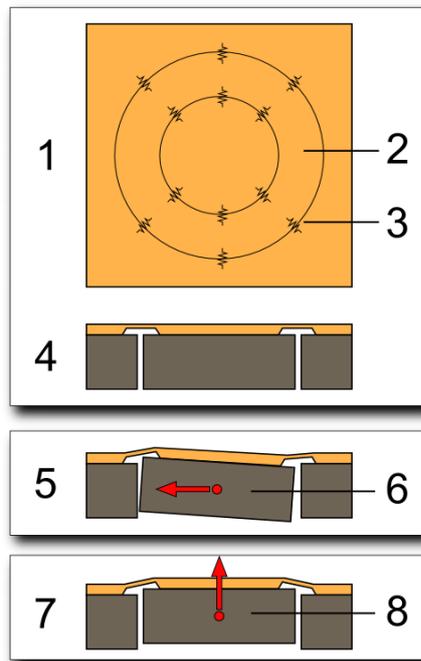


Figura 1.5: Acelerómetro mecánico.
Fuente: <https://commons.wikimedia.org>

1.4.2 Giroscopio

Un giroscopio es un dispositivo capaz de medir, mantener o cambiar la orientación en el espacio de algún aparato o vehículo. Los giroscopios MEMS (sistema micro electromecánico) (figura 1.6) son sensores capaces de medir la velocidad angular, esto es, simplemente la medición de la velocidad de rotación.

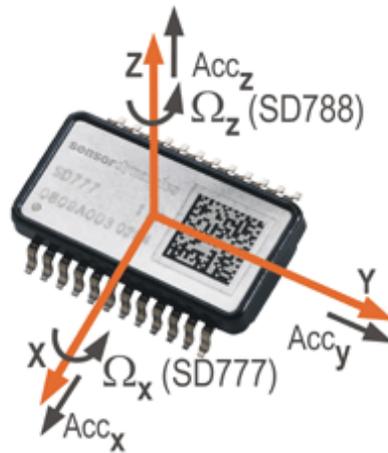


Figura 1.6: Giroscopio MEMS.
Fuente: <https://et.wikipedia.org/>

Éstos son capaces de medir la rotación alrededor de tres ejes: x, y, z. Para poder detectar la velocidad angular, cuando el giroscopio gira (figura 1.7), una pequeña masa es desplazada a medida que cambia la velocidad angular, y este movimiento pasa a convertirse en señales eléctricas de muy bajo voltaje, las cuales pueden ser amplificadas y leídas por un microcontrolador.

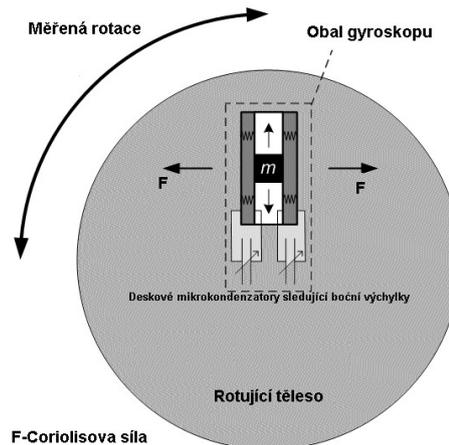


Figura 1.7: Vista operativa interna de un sensor giroscópico MEMS.
Fuente: <https://commons.wikimedia.org>

Las velocidades angulares que se obtienen con el giroscopio permiten calcular los ángulos del sistema oscilatorio del buque. A través de la integración de estas velocidades se obtienen los ángulos del sistema de referencia móvil que, siendo aplicados en las matrices de Euler, proporcionan los ángulos del sistema con respecto a otro sistema de coordenadas establecido en un punto fijo en tierra [12].

Si se conoce un registro de la velocidad angular del móvil podemos calcular su desplazamiento $q - q_0$ entre los instantes t_0 y t , mediante la integral definida.

$$\theta - \theta_0 = \int_{t_0}^t \omega dt \quad (1.9)$$

El producto ωdt representa el desplazamiento angular del móvil entre los instantes t y $t+dt$, o en el intervalo dt . El ángulo total es la suma de los infinitos desplazamientos angulares infinitesimales entre los instantes t_0 y t .

1.5 Fuerzas que actúan en la estabilidad transversal

Las fuerzas principales que actúan en un buque son el desplazamiento y el empuje. El desplazamiento (D), se determina por su peso total y es igual al volumen sumergido multiplicado por la densidad del agua, por lo que equivale al peso del agua desplazada por ese volumen. Es una fuerza vectorial que va en sentido descendente y su módulo se aplica en el centro de gravedad del buque[8].

$$D = V\mu \quad (1.10)$$

D Desplazamiento [Tm]

V Volumen sumergido [m^3]

μ Densidad del agua [Tm/m^3]

El peso total o desplazamiento de un buque pueden determinarse a través de sus marcas de calado [8].

Por otro lado, el empuje (E) es la fuerza que equilibra al desplazamiento. El empuje corresponde a la fuerza vertical ascendente equivalente al peso de la masa de agua que desaloja el buque al hundirse parcialmente. Esta fuerza se aplicada en el centro de carena¹.

$$E = P_e \cdot V = \rho_f \cdot g \cdot V \quad (1.11)$$

E Empuje [N]

P_e Peso específico del fluido [N/m^3]

ρ_f Densidad del fluido [kg/m^3]

V Volumen del fluido desplazado [m^3]

g Aceleración de la gravedad [m/s^2]

Estas fuerzas servirán de ayuda para la determinación de la altura metacéntrica y el momento adrizante, explciado en las secciones 1.7 y 1.8.

1.6 Determinación del centro de masas

El centro de masas de un sistema es el punto geométrico que dinámicamente se comporta como si en él estuviera aplicada la resultante de las fuerzas externas al sistema. Mientras que el centro de gravedad es el punto de aplicación de la resultante de todas las fuerzas de gravedad que actúan sobre las distintas porciones materiales de un cuerpo. El centro de gravedad y el centro de masas coinciden cuando el sistema se encuentra en un campo gravitatorio uniforme, por eso, se hablará siempre de centro de gravedad haciendo referencia a ambos.

El centro de gravedad (G) se corresponde con el punto de aplicación del peso del buque, el cual depende de la distribución de pesos.

La posición del centro de gravedad se calcula mediante la siguiente ecuación:

¹Centro de gravedad del volumen de agua desplazado por un cuerpo, para una condición determinada.

$$G = \frac{1}{m} \cdot \int_V r \rho(r) \cdot dV \quad (1.12)$$

m Masa del sistema [kg]

V Volumen del sistema [m^3]

r Posición de la masa en el sistema [m]

p Densidad [kg/m^3]

También la posición de G queda definida dentro del buque por sus coordenadas en los planos de referencia [7]:

- Plano horizontal: para su posición vertical (VG), la distancia desde la línea base, es decir, la quilla.
- Plano longitudinal: para su posición longitudinal (LG), la distancia desde la cuaderna maestra.
- Plan transversal: para su posición transversal (TG), la distancia desde la línea central.

La posición del centro de gravedad puede ser calculada o determinada experimentalmente. Una vez botado el barco después de su construcción el VG puede obtenerse a través de la experiencia de estabilidad² [2], el TG debe ser cero por definición debido a la simetría del buque, y el LG puede ser calculado a partir de la observación de los calados y el asiento.

1.7 Altura metacéntrica

1.7.1 Condición de equilibrio y metacentro

Para el cálculo de la altura metacéntrica transversal es necesario partir de la condición de equilibrio. Cuando una fuerza externa produce una escora, si se traza una línea correspondiente a la fuerza de empuje vertical, la cual pasa por el centro de carena inicial, y otra por el centro de carena final, el punto en el cual se cortan las dos es el punto denominado metacentro transversal, tal y como se muestra en la figura 1.8.

La condición de equilibrio corresponde a un buque en estado adrizado, es decir, su posición normal de flotabilidad. En esta condición, la línea de empuje coincide con la línea central y el metacentro recibe el nombre de metacentro transversal inicial.

²Tiene como objetivo el cálculo de la altura del centro de gravedad sobre la línea base de un buque, y a su vez, permite el cálculo del desplazamiento en rosca del buque y su periodo de balance.

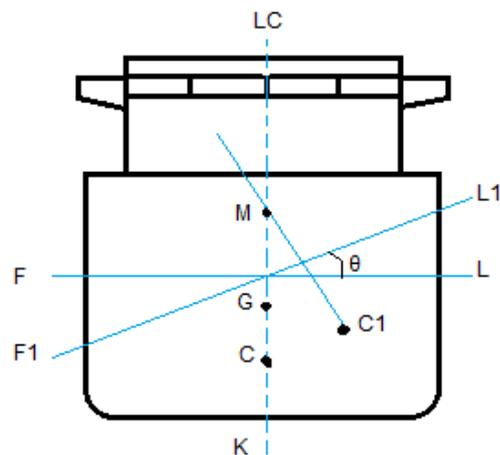


Figura 1.8: Desplazamiento del centro de carena.

Fuente: Elaboración propia

LC	Línea central
FL	Flotación
F1L1	Plano de inclinación transversal
G	Posición transversal del centro de gravedad
C	Posición transversal inicial del centro de carena
C1	Posición transversal final del centro de carena
M	Posición transversal del metacentro
θ	Ángulo de escora

1.7.2 Altura metacéntrica transversal

Se entiende como altura metacéntrica (valor GM), refiriéndose a (G) como el centro de gravedad del buque y (M) a su metacentro, siendo la altura metacéntrica la distancia desde el centro de gravedad al metacentro. La altura metacéntrica será positiva si el metacentro se encuentra por encima del centro de gravedad, y negativa en el caso contrario. La altura metacéntrica es un valor representativo de la estabilidad estática transversal [8].

Existen dos componentes que determinan la altura metacéntrica, una está determinada por el propio diseño de las formas del buque, ya que los ingenieros que lo diseñan intentan establecer el metacentro en un punto positivo para que la embarcación sea estable, obteniendo este dato a través de las curvas hidrostáticas³. La otra componente (G) depende de la distribución de los pesos a bordo, su carga y su descarga.

La carga de un peso afecta al centro de gravedad del buque desplazándolo en el plano vertical y en el horizontal en la dirección que se haya colocado la carga.

³Las curvas hidrostáticas son gráficas cuyos datos han sido previamente calculados a partir de los planos de formas del buque, por lo que se corresponden con la geometría concreta de cada barco. Entrando con el calado medio del buque como dato inicial, se obtienen a través de estas gráficas otros datos como el desplazamiento, posición vertical y longitudinal del centro de carena, posición longitudinal del centro de flotación, etc.

El equilibrio de un buque está determinado por el centro de carena (C), el metacentro (M), y el centro de gravedad (G).

Para que el buque tenga una buena estabilidad, el metacentro (M), siempre debe de encontrarse por encima del centro de gravedad (G). Esto significa que el barco tiene una altura metacéntrica inicial positiva, y este tenderá siempre a volver a su posición inicial.

La altura metacéntrica también afecta a la estabilidad del buque, siendo mayor cuanto mayor sea ésta, es decir, la distancia entre el centro de gravedad (G), y el metacentro (M).

1.8 Momento adrizante y sólido de revolución

Como se comentó anteriormente en la sección 1.5, las fuerzas que actúan en la estabilidad transversal, empuje y desplazamiento serán útiles para entender el concepto de momento adrizante. Si mediante una fuerza externa, la embarcación es sacada de la condición inicial de equilibrio y se establece con un ángulo (θ) como se muestra en la figura 1.9, el desplazamiento se queda invariable aplicado su peso en el centro de gravedad (G). El empuje también se mantiene constante, pero no así su punto de aplicación que cambia al nuevo centro de carena (C1). De esta manera se crea una recta de acción (GZ), que se designa como brazo adrizante. Esto establece una nueva distribución de las fuerzas en la embarcación, produciendo un par de fuerzas denominado momento adrizante (ecuación 1.13), que produce que el buque vuelva a su posición de inicio [11].

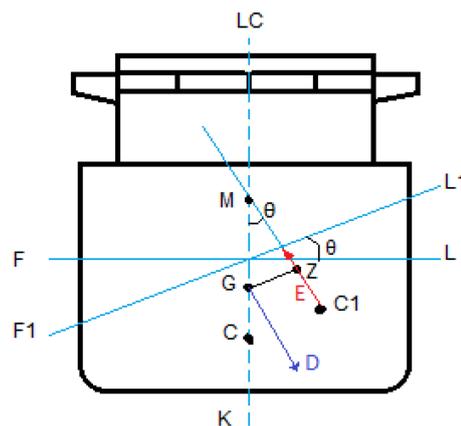


Figura 1.9: Brazo adrizante.
Fuente: Elaboración propia.

$$M_{\alpha} = D \cdot GZ = D \cdot GM \cdot \sin \theta \approx D \cdot GM \cdot \theta \quad (1.13)$$

La ecuación 1.13 junto con la de un sólido que gira alrededor de un eje (ecuación 1.14), permite ligar el periodo doble de balance con la altura metacéntrica.

$$\Sigma M = I_{\alpha} = I_g \frac{d^2 \theta}{dt^2} \quad (1.14)$$

I_g Momento de inercia de la masa del buque alrededor de un eje longitudinal que pasa por su centro de gravedad [$kg \cdot m^2$]

θ Ángulo de escora [$^{\circ}$]

$\frac{d^2 \theta}{dt^2}$ Aceleración angular [s^{-2}]

La ecuación 1.14 será igual a la ecuación 1.13, según el planteamiento expuesto, donde el periodo doble de balance se calcula en aguas tranquilas y sin considerar el amortiguamiento por lo que la suma de ambas será igual a cero, proporcionando la ecuación 1.15 con la cual es posible calcular el periodo doble de balance [9] en el siguiente apartado.

$$I_g \frac{d^2\theta}{dt^2} + M_a = 0 \quad (1.15)$$

1.9 Momento de inercia y periodo doble de balance

El momento de inercia depende de la distribución de masa de un cuerpo o sistema de partículas en rotación respecto a un eje de giro. Por lo tanto, este momento depende de la geometría de dicho cuerpo, de la posición del eje de giro y de la distribución de su masa sin que influyan las fuerzas que intervienen en el movimiento [9].

Se entiende como periodo doble de balance al tiempo necesario para que el sistema, en este caso el barco, se encuentre en su mismo estado al realizar un movimiento oscilatorio, es decir, cuando alcanza la misma posición, velocidad y amplitud [7].

Para calcular el momento de inercia de un determinado sistema de partículas y un eje arbitrario, se considera la suma de los productos de las masas de las partículas por el cuadrado de la distancia mínima de cada partícula a dicho eje. Se expresa matemáticamente como [9]:

$$I_g = \sum m_i \cdot r_i^2 = \frac{D}{g} \cdot k^2 \quad (1.16)$$

m_i Masas parciales [kg]

r_i Radios de las masas parciales [m]

D Desplazamiento del buque [Tm]

g Aceleración de la gravedad [m/s^2]

k Radio de giro equivalente de la masa del buque con respecto al eje longitudinal que pasa por su centro de gravedad [m]

Si se incorpora a la ecuación 1.16 anterior la ecuación del momento adrizante (ecuación 1.13), se podrá obtener el periodo doble de balance a través de la siguiente deducción:

$$\frac{D}{g} \cdot k^2 \cdot \frac{d^2\theta}{dt^2} + D \cdot GM \cdot \theta = 0 \quad \frac{d^2\theta}{dt^2} + \frac{g \cdot GM}{k^2} \cdot \theta = 0 \quad (1.17)$$

Esta ecuación es similar a la del movimiento vibratorio armónico simple, de la cual se puede deducir la frecuencia angular del movimiento oscilatorio del buque en función de su radio de giro equivalente.

$$\omega = \sqrt{\frac{g \cdot GM}{k^2}} \quad (1.18)$$

Teniendo en cuenta la fórmula estándar del periodo obtenida a través de la frecuencia (ω), la cual sería:

$$T_d = \frac{2\pi}{\omega} \quad (1.19)$$

Si sustituimos la ecuación 1.18 en la ecuación 1.19, se obtendrá la fórmula del periodo doble de balance en función de la altura metacéntrica y del radio de giro equivalente.

$$T = \frac{2\pi k}{\sqrt{gGM}} \quad (1.20)$$

De la ecuación anterior se puede deducir que el periodo natural de balance del buque no depende del valor de la escora que tome durante la oscilación. Dicho de otra forma: el movimiento de balance es isócrono ⁴. Cabe mencionar que el periodo doble de balance será algo inferior a su valor real, debido, a que en esta ecuación no se considera la masa de agua arrastrada por el buque en su movimiento [9].

Para obtener una buena medición del periodo es recomendable llevar a cabo varias pruebas de comprobación del periodo de balance. Si se obtiene un periodo muy elevado, es un indicativo de que el buque tiene poca estabilidad. Un periodo muy corto, indica una gran estabilidad, pero a costa de hacer la vida a bordo muy incómoda. Un periodo ideal, es uno intermedio, en torno a 15 ó 20 segundos [2]. En la siguiente tabla se puede apreciar el periodo de balance recomendable para determinados tipos de buque [6]:

Buque grande para pasajeros	20 a 25 segundos
Buque mediano para pasajeros	18 a 20 segundos
Portacontenedores	16 a 19 segundos
Carguero grande	12 a 14 segundos
Carguero mediano o pequeño	10 a 12 segundos
Cruceros de guerra	12 a 15 segundos
Portaaviones	18 a 20 segundos

Tabla 1.2: Periodos según tipo de barco.

1.10 Parámetros de estabilidad estática y dinámica

Para calcular los parámetros de la estabilidad estática transversal. Se traslada un peso transversalmente en el buque, lo que producirá un ladeo en el sentido del movimiento de dicha carga. El ángulo de escora producido de esta forma permite determinar la altura metacéntrica de forma sencilla [2](ver ecuación 1.21).

$$\tan \theta = \frac{GG_2}{GM} \quad GG_2 = \frac{w \cdot d}{D} \quad GM = \frac{w \cdot d}{D \cdot \tan \theta} \quad (1.21)$$

w Peso movido [kg]

D Desplazamiento [Tm]

d Distancia transversal a la que se ha movido el peso [m]

GM Altura metacéntrica del buque [m]

$\tan \theta$ Tangente del ángulo de escora [rad]

Todas las variables de la ecuación 1.21 son conocidas excepto el ángulo de escora (θ) y el desplazamiento del buque. Para obtener el ángulo de escora el buque tiene que estar estable, adrizado y sin

⁴Dos o más movimientos que tienen una misma duración de tiempo.

asiento. Para medir dicho ángulo de forma manual se emplean péndulos de longitud aproximadamente igual a la distancia entre dos cubiertas colocados en la línea de crujía⁵[2]. En este trabajo, sin embargo, se emplea un acelerómetro de 3 ejes para medir la escora de forma más sencilla. Por otro lado, para calcular el desplazamiento del buque se precisa de la densidad del agua en el momento que se realiza la experiencia de estabilidad y del volumen sumergido en metros cúbicos, el cual se calcula de la siguiente manera:

$$V_s = E \cdot B \cdot C_m \cdot C_a \quad (1.22)$$

V_s Volumen sumergido [m^3]

E Eslora [m]

B Manga [m]

C_m Calado medio [m]

C_a Coeficiente de afinamiento

Pudiendo hallar con esto el desplazamiento del buque:

$$D = V_s \cdot p \quad (1.23)$$

p Densidad del agua [kg/m^3]

Una vez se ha obtenido la altura metacéntrica es posible determinar la posición del centro de gravedad del buque [2] en la condición de carga que presenta en el momento de realizar la experiencia.

$$KG = KM - GM \quad (1.24)$$

KG Distancia desde la quilla al centro de gravedad [m]

KM Distancia desde la quilla al metacentro [m]

GM Distancia desde el centro de gravedad al metacentro [m]

KM lo proporcionan las curvas hidrostáticas, entrando con el calado medio del buque en el momento de la prueba de estabilidad.

La altura metacéntrica también puede calcularse de forma aproximada cuando el buque está bajo un movimiento de balance [2], como se muestra en la ecuación 1.25.

$$GM = \left(f \cdot \frac{B}{T} \right)^2 \quad (1.25)$$

GM Altura metacéntrica [m]

B Manga máxima del buque [m]

T Duración de un balance completo (Br a Er y Er a Br) [s]

⁵Se define como la línea imaginaria que, pasando de proa a popa de la embarcación, divide al buque en dos mitades iguales.

f Factor de corrección (20 por 100 de su carga = 0,78; 10 por 100 de su carga = 0,75; 5 por 100 de su carga = 0,73)

El sensor cuyo diseño es objeto de este trabajo permite realizar los cálculos de estabilidad anteriores a través de la obtención de periodos de balance y cabeceo, además de sus ángulos de manera automática, como se demuestra en los capítulos siguientes.



INFO ABOUT RIGHTS
1 809028 251697
www.safecreative.org/work



INFO ABOUT RIGHTS
1 809028 251697
www.safecreative.org/work

2 Desarrollo del dispositivo

El dispositivo desarrollado en este proyecto permite hacer un análisis de la estabilidad de forma rápida tomando datos de manera automatizada. Estos datos tomados se corresponden con los movimientos oscilatorios de balance y cabeceo del buque. Se obtienen a través del cálculo de su posición angular a lo largo del tiempo para que, una vez extraídos estos datos, obtener el periodo de la oscilación que presenta el buque con distintos tipos de pesos y diferentes distribuciones de carga.

Para llevar a cabo este proyecto se han usado varios dispositivos para obtener los datos necesarios mencionados anteriormente. En primer lugar, se ha utilizado una unidad compacta IMU MPU6050 que dispone de un acelerómetro y un giroscopio, ambos capaces de tomar medidas en tres ejes. El acelerómetro toma medidas de aceleraciones tangenciales y el giroscopio a su vez mide la velocidad angular. A través de los cálculos explicados en la sección 2.3.2, se obtienen los ángulos del sistema a lo largo del tiempo.

Todos los cálculos necesarios para la obtención de los ángulos son procesados en una placa microcontroladora Arduino Nano.

2.1 Hardware

2.1.1 Arduino Nano

Arduino Nano es la placa microcontroladora que mejor se adapta a las necesidades de procesamiento requeridas, con un bajo coste y tamaño reducido, contando en este caso con un procesador ATMEGA328 de 8 bit, con una velocidad de reloj de 16 MHz y con una memoria SRAM de 2 KB, lo cual es suficiente para realizar las operaciones requeridas por el dispositivo. En la tabla 2.1 se muestran sus características principales.

Microprocesador	ATMEGA328
Voltaje de funcionamiento	5 V
Tensión de entrada (recomendada)	7 - 12 V
Tensión de entrada (límite)	6 - 20 V
E/S digitales	22, de las cuales 6 son PWM
Entradas analógicas	8
Corriente DC por E/S	40 mA
Longitud	45 mm
Anchura	18 mm

Tabla 2.1: Características de placa Arduino Nano[1].

2.1.2 Unidad de medición inercial

Se ha decidido utilizar una unidad de medición inercial (IMU MPU6050) como dispositivo para la obtención de los datos necesarios.

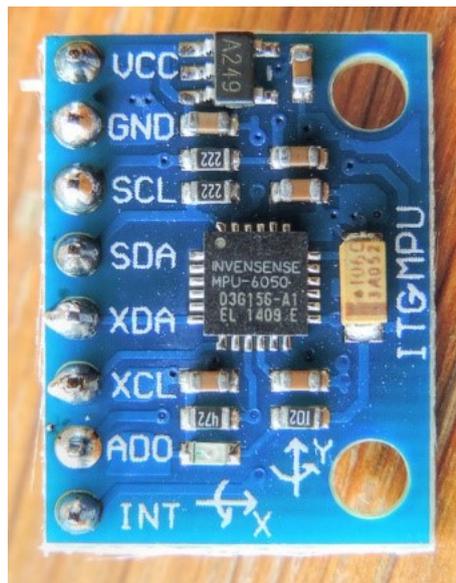


Figura 2.1: Unidad inercial de 3 ejes MPU6050.

Fuente: <https://commons.wikimedia.org>

El MPU6050 es una unidad de medición inercial (IMU), la cual combina un sensor giroscópico MEMS de 3 ejes y un acelerómetro MEMS de 3 ejes en el mismo chip, junto con una tarjeta de procesador de movimiento digital. Esta IMU contiene un convertor analógico/digital de 16 bit.

Las conexiones entre Arduino Nano y la MPU6050 usan el protocolo I2C¹ estándar:

MPU6050	Arduino Nano
VCC	5 V
GND	GND
SCL	A5
SDA	A4

Tabla 2.2: Conexiones entre MPU6050 y Arduino Nano.

2.1.3 Módulo de tarjeta SD

Como dispositivo para el almacenamiento de los datos obtenidos por la IMU se ha optado por una tarjeta micro SD con 2 GB de capacidad de almacenamiento, lo cual es más que suficiente para almacenar dichos datos. La escritura de datos en la tarjeta micro SD se ha hecho mediante un módulo de tarjetas SD conectado a la microcontroladora Arduino Nano.

2.2 Desarrollo del software

El software y la gestión de la comunicación entre los periféricos se ha desarrollado en Arduino. Se optó por el uso de este entorno de programación debido a que es una herramienta para la elaboración de aplicaciones electrónicas de código abierto, con un entorno de programación sencillo y extensible mediante librerías en C++.

En el entorno de programación no existe un simulador con el que comprobar el funcionamiento del código, por lo que se empleó directamente el hardware para llevar a cabo las pruebas de funcionamiento.

¹Es un estándar que facilita la comunicación entre microcontroladores, memorias y otros dispositivos.

2.2.1 Arduino IDE

Arduino dispone de un IDE, es decir, un entorno de desarrollo integrado (Integrated Development Environment). Es un programa informático formado por un conjunto de herramientas de programación. Consta de un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). También contamos con herramientas que nos posibilitan cargar el programa una vez compilado en la memoria flash de la placa Arduino[1].

2.3 Diagrama de flujo / Algoritmo

El diagrama de flujo es una simbolización del algoritmo del dispositivo (figura 2.2), representando los pasos que sigue el código fuente para calcular los ángulos de balance y cabeceo. El código fuente correspondiente a este algoritmo se puede consultar en el anexo A. A continuación se explica brevemente el funcionamiento de cada uno de los bloques del algoritmo representado en la figura 2.2:

- Calibración: mediante la calibración el sistema calcula la constante de integración necesaria para determinar el ángulo en cada instante a partir de la velocidad angular medida por el giroscopio (ver líneas de código 208 hasta la 450 en el anexo A).
- Lectura de los datos de la IMU: se procede al almacenamiento de los datos proporcionados por la unidad de medición inercial en las variables que utilizaremos para calcular los ángulos de balance y cabeceo (ver líneas de código 166 y 167 en el anexo A).
- Transformación de lecturas a grados: se procesan las medidas guardadas en las variables anteriores para poder obtener de ellas los datos necesarios para la realización de los cálculos (ver líneas del código 169 hasta la 175 en el anexo A).
- Cálculo de ángulos de inclinación con el giroscopio: se obtienen los ángulos de balance y cabeceo a través de la realización de los cálculos del arco tangente del eje x o eje y partido por el valor de la aceleración en el eje vertical z, aplicando las medidas procesadas obtenidas previamente por el acelerómetro (ver líneas de código 170 y 171 en el anexo A).
- Cálculo de los ángulos de inclinación con el giroscopio: se calculan a partir de las medidas procesadas obtenidas por la IMU correspondientes con el giroscopio, en este caso se obtienen velocidades angulares. Integrando estas velocidades angulares se hallan los ángulos de balance y cabeceo obtenidos por el giroscopio (ver líneas de código 181 y 182 en el anexo A).
- Obtención de ángulos finales aplicando el filtro complementario: se calculan los ángulos finales proporcionados por el sistema a través de la ponderación ejercida por el filtro complementario a un mismo ángulo tomado de dos formas distintas, con el acelerómetro y el giroscopio, proporcionando un único ángulo resultante (ver líneas de código 181 y 182 en el anexo A).
- Almacenamiento de datos en la SD: Se procede al guardado de los ángulos finales de balance y cabeceo, junto con una variable de tiempo en la tarjeta de memoria para su posterior estudio (ver líneas de código 104 hasta la 147 en el anexo A).

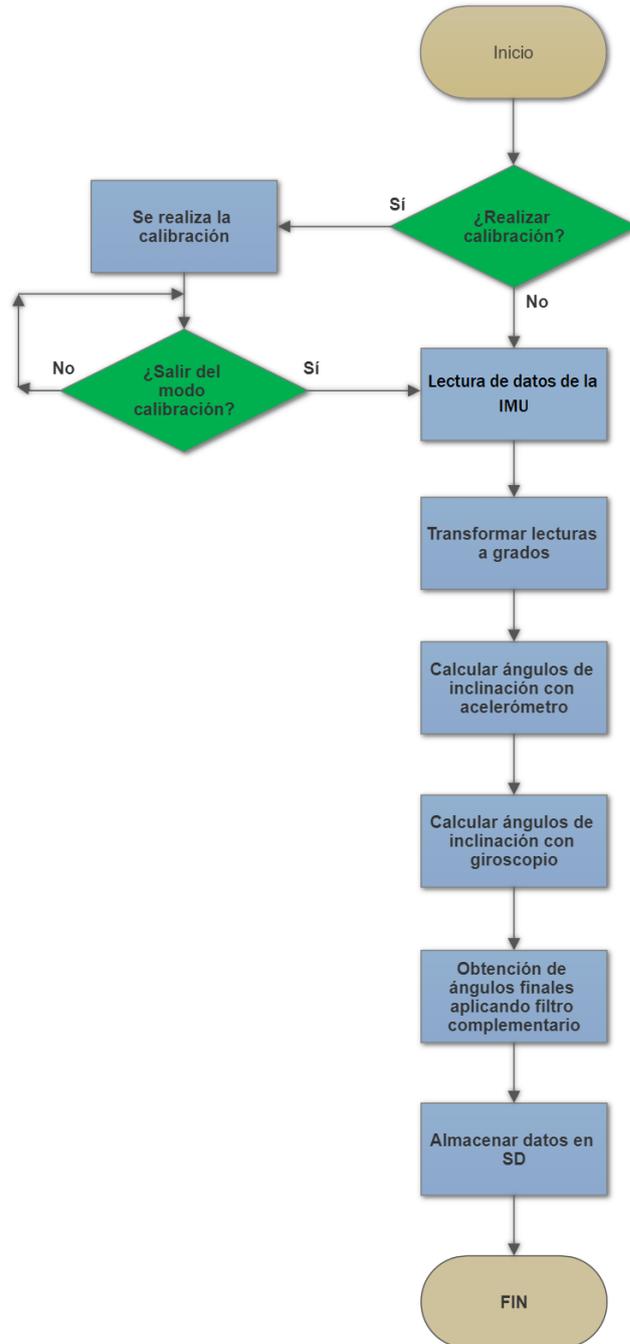


Figura 2.2: Diagrama de flujo general.
Fuente: Elaboración propia.

2.3.1 Calibración de la unidad inercial

La calibración de la IMU se lleva a cabo mediante un código obtenido de Jeff Rowberg (citado en anexo A), adaptándolo al software de este proyecto para calcular las compensaciones que necesitan ser aplicadas en la lectura de datos que realizan tanto el acelerómetro como el giroscopio, para lograr así obtener una medida correcta del ángulo de balance.

La calibración es posible ejecutarla al comienzo de la ejecución del software mediante la pulsación de un botón (líneas 150 a 155). Al pulsarlo se ejecutan las funciones que realizan la calibración (líneas 208 a 450). En caso de que se haya pulsado el botón por error se dispone de 3 segundos para volver a pulsarlo y que no realice la calibración, regresando al modo de cálculo de ángulos.

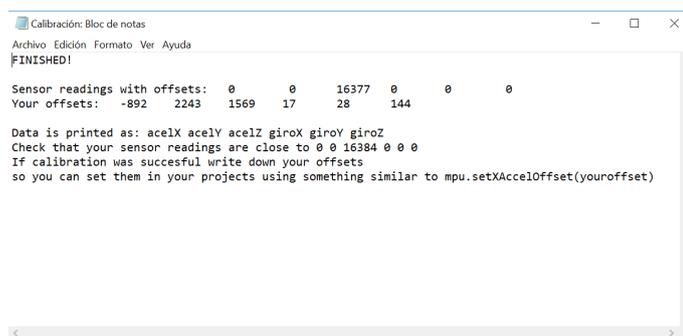
El proceso de calibración se inicia en la función de control denominada *start_call()*. Comienza imprimiendo por la pantalla del display las instrucciones para realizar la calibración. Se especifica que se ha de colocar la IMU en posición horizontal en una superficie lo más recta posible (se calcula la inclinación de la superficie donde se realiza la calibración con un nivel). Seguidamente en la función de control se pueden observar tres estados, siendo estos 0, 1 y 2, y estando almacenados en una variable. El programa está diseñado para que acceda a cada estado en orden ascendente. Estos estados se describen a continuación:

- **Estado 0:** se corresponde con la función *meansensors()*: Realiza la toma de las lecturas sin procesar del giroscopio y del acelerómetro en cada uno de sus ejes.

- **Estado 1:** se corresponde con la función *calibration()*: Calcula la media de las medidas obtenidas en la función anterior y se guardan en las variables offset de cada uno de los ejes del acelerómetro y del giroscopio, cargándose posteriormente estas medidas en las variables internas de compensación de la IMU.

- **Estado 2:** se corresponde con la función *meansensors()*: Se toman de nuevo las lecturas de la IMU, pero esta vez, con las compensaciones calculadas, presentando ya valores útiles.

Una vez se ha terminado el proceso del paso a través de todos los estados de la función *start_call()*, se procede a imprimir por puerto serie todos los parámetros que se han calculado en las funciones anteriores, como se ve en la figura 2.3. Cuando se ha finalizado la calibración el programa permite salir de este modo accionando el pulsador, haciendo que se retorne de nuevo a la función principal *loop()*, donde se calcularán los ángulos aplicando los offset calculados durante el proceso de calibrado.



```

Calibración: Bloc de notas
Archivo Edición Formato Ver Ayuda
FINISHED!

Sensor readings with offsets:  0      0    16377  0      0      0
Your offsets:    -892   2243   1569   17     28    144

Data is printed as: accelX accelY accelZ gyroX gyroY gyroZ
Check that your sensor readings are close to 0 0 16384 0 0 0
If calibration was succesful write down your offsets
so you can set them in your projects using something similar to mpu.setXAccelOffset(youroffset)

```

Figura 2.3: Información del proceso de calibración a través del puerto serie.

Fuente: Elaboración propia

2.3.2 Algoritmo de cálculo para la obtención de los ángulos de la IMU

El código del algoritmo de cálculo de los ángulos de la IMU (figura 2.4) se define entre las líneas 165 y 186. En esta parte del código se comienza leyendo los valores del acelerómetro y el giroscopio sin procesar. Los ángulos se obtienen a partir de las aceleraciones y de las velocidades angulares según se explica en las subsecciones siguientes:

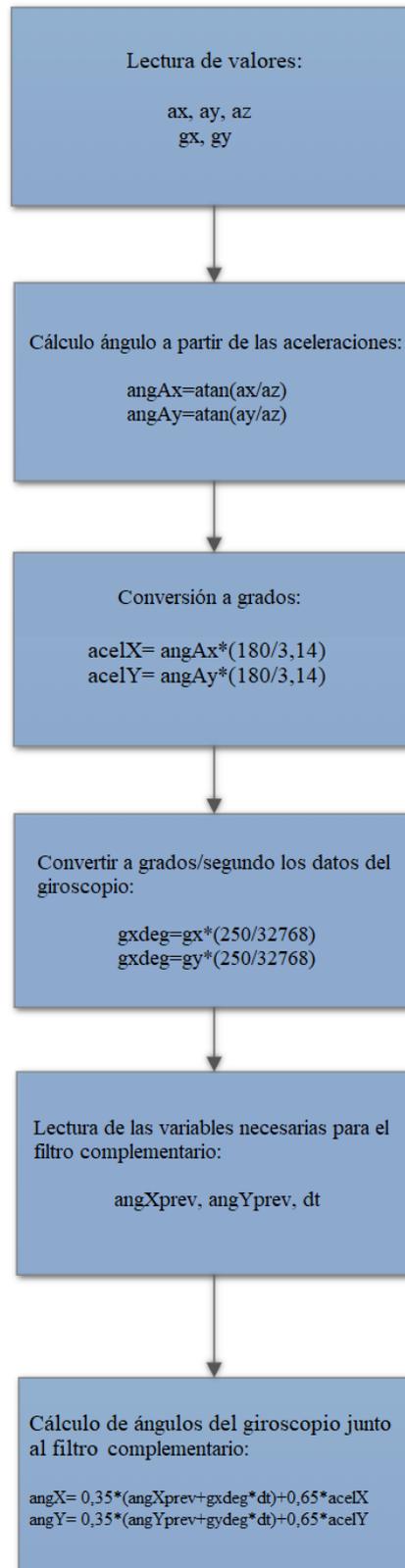


Figura 2.4: Diagrama de flujo para la determinación del ángulo de escora.

Fuente: Elaboración propia.

2.3.2.1 Determinación de los ángulos estáticos de balance y cabeceo mediante el acelerómetro

El acelerómetro de la IMU permite la obtención de los ángulos estáticos de balance ϕ y cabeceo θ mediante transformaciones trigonométricas de las aceleraciones. Teniendo en cuenta la fuerza de la gravedad, conociendo las aceleraciones en los 3 ejes del acelerómetro, es posible determinar los ángulos de balance y cabeceo:

$$\phi = \text{atan} \left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}} \right) \quad (2.1)$$

$$\theta = \text{atan} \left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}} \right) \quad (2.2)$$

2.3.2.2 Determinación de los ángulos dinámicos de balance y cabeceo mediante el giroscopio

El giroscopio procesa las velocidades angulares, de las cuales se puede obtener el ángulo recorrido, haciendo la integral definida de la velocidad angular en un rango de tiempo.

Para calcular el ángulo del sistema con el giroscopio es necesario aportar un ángulo inicial, ya que el giroscopio no detecta la posición en que se encuentra el sistema, sino su velocidad angular. El ángulo inicial que necesita el giroscopio se le aporta con las medidas tomadas por el acelerómetro en reposo sobre una superficie horizontal, que si es capaz de medir los ángulos en un estado estacionario. La integración de la señal del giroscopio se realiza en el algoritmo según las siguientes ecuaciones:

$$\phi = \phi \text{ anterior} + \Omega_{\phi} \text{ Giroscopio} \cdot \Delta_t \quad (2.3)$$

$$\theta = \theta \text{ anterior} + \Omega_{\theta} \text{ Giroscopio} \cdot \Delta_t \quad (2.4)$$

2.3.3 Filtrado de la señal

Mediante el uso de un algoritmo de filtrado es posible suprimir la deriva y el ruido, y se logra que el acelerómetro no cambie de ángulo cuando otra fuerza que no sea la de la gravedad incide sobre él [3]. Existen distintos tipos de filtros, siendo uno de los mejores el filtro Kalman, el cual calcula el error de cada medida a partir de medidas anteriores, eliminándolo y proporcionando el valor real de la inclinación.

El filtro Kalman, a pesar de ser uno de los mejores, se ha descartado en este proyecto debido al alto consumo de los recursos del microprocesador de la placa utilizada (Arduino Nano) no cuenta con una gran capacidad de procesamiento. Por este motivo se ha optado por el uso de un filtro complementario, el cual tiene una precisión aceptable y requiere un menor trabajo del microprocesador.

Para el funcionamiento de este filtro complementario se combinan dos filtros, un filtro de paso bajo (FPB) para el acelerómetro y un filtro de paso alto (FPA) para el giroscopio. El filtro de paso bajo corresponde a un filtro que permite el paso de las frecuencias más bajas y atenúa las más altas, en cambio, el filtro de paso alto se caracteriza por atenuar la señal a valores de frecuencia bajos.

La fórmula resultante de la combinación de los dos filtros será la siguiente:

$$\theta = a_1 \cdot (\theta + \theta \text{ Giro} \cdot \delta_t) + a_2 \cdot \theta \text{ acelerometro} \quad (2.5)$$

Para obtener una mejor calibración del filtro complementario, se puede cambiar mediante prueba y error los valores de a_1 y a_2 , teniendo en cuenta que éstos siempre han de sumar uno [3].

2.3.4 Procedimiento de guardado de datos en la SD

El grueso del código que se encarga del almacenamiento de datos se encuentra ubicado en el programa entre las líneas 104 y la 147. Este fragmento de programa, en primer lugar, se encarga de comprobar si existe algún archivo ya guardado en la SD. Si el caso es negativo, procede a crearlo y escribe su encabezado. En segundo lugar, se abre el archivo dentro del bucle loop y se procede a guardar en la tarjeta SD las variables de tiempo, ángulo en el eje X y ángulo en el eje Y. Por último se procede al cierre del archivo, para evitar la corrupción de los datos. Este cierre se realiza al final del loop en la línea de código 203.

3 Validación

3.1 Casco para llevar a cabo las pruebas de balance

Para la realización de las pruebas de estabilidad se ha diseñado la maqueta de un barco de unos 200 milímetros de eslora, 50 milímetros de manga, 40 milímetros de puntal, y un espesor de los costados de 3 milímetros. La eslora de la maqueta estaba limitada por la cama de impresión de la impresora 3D. El peso final del casco es de 56,6 gramos.

Para diseñar la maqueta se usó Rhinoceros, al estar ya familiarizados con esta herramienta.

3.2 Montaje de maqueta para pruebas

Al diseño final de la maqueta (figura 3.2) se le han añadido una serie de pesos repartidos por el interior de ésta, ya que el plástico con el que se ha impreso el casco (ver figura 3.1) es un material muy ligero. También se instaló transversalmente un espárrago roscado para poder ir añadiendo tuercas a diferentes distancias del eje longitudinal, y así aumentar o disminuir el momento de inercia para calcular distintos índices de estabilidad.

Se ha seleccionado un espárrago de 4 milímetros de diámetro. Este espárrago tiene una longitud de 150 milímetros, sobresaliendo 50 milímetros por cada banda de la maqueta para la colocación de los pesos. Este espárrago, se encuentra sujeto por dos pletinas de 50 milímetros de longitud y 15 milímetros de diámetro.

Para añadir lastre a la maqueta se han utilizado dos espárragos de 37 mm de longitud, y 9 mm de diámetro, con un peso cada uno de 17 gramos. Estos espárragos se han colocado entre las cuadernas, en los espacios 2 y 4 representados en la figura 3.3.

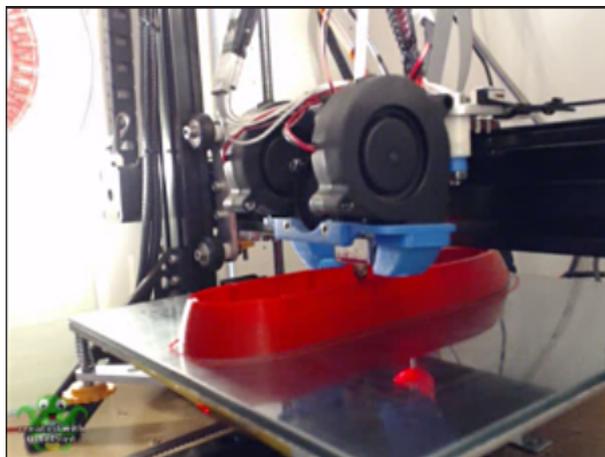


Figura 3.1: Proceso de impresión 3D.
Fuente: Elaboración propia.

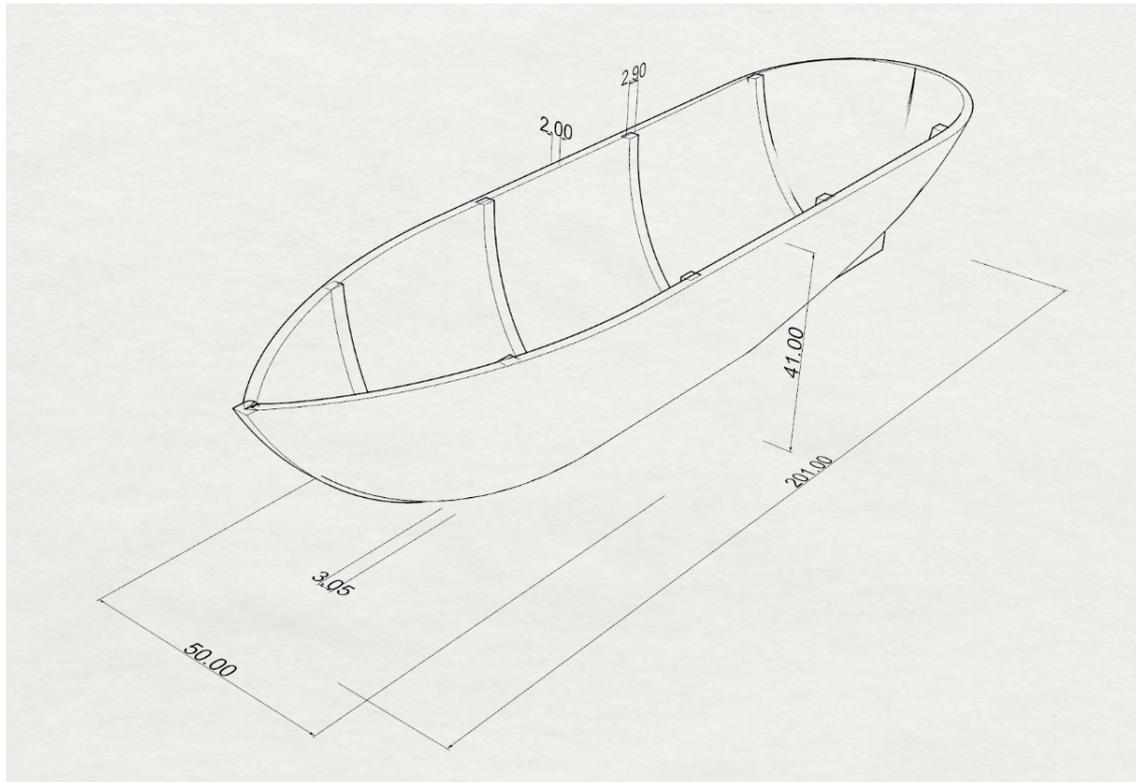


Figura 3.2: Perspectiva del casco diseñado.
Fuente: Elaboración propia.

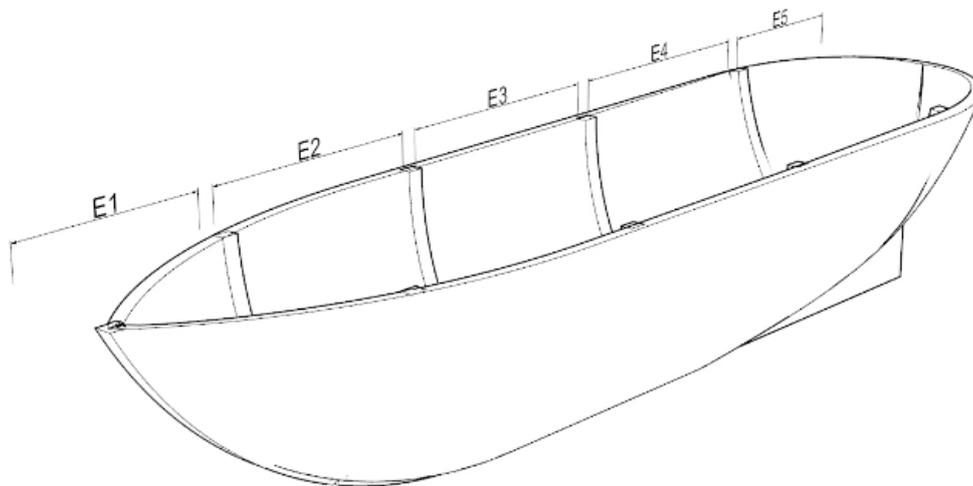


Figura 3.3: Espacios entre cuadernas del casco.
Fuente: Elaboración propia.

También se ha añadido una tuerca en el espacio 5, a la popa de la maqueta, ya que, al ser la proa más fina, ésta desplaza una menor cantidad de agua, lo que produce un calado ligeramente mayor en esta zona. Esta tuerca que se ha añadido tiene un diámetro interior de 6 mm, un espesor de 7 mm, y un peso de 5 gramos.

En la tabla 3.1 se muestran los pesos de todos los elementos que posee la maqueta, tanto los añadidos para posibilitar la realización de las pruebas, como los mencionados anteriormente para darle una mayor estabilidad.

Elemento	Peso unitario (g)	Cantidad	Longitud (mm)	Diámetro (mm)
Pletina	4,7	2	50	15
Espárrago de carga de pesos	18	1	150	4
Espárrago de lastre	17	2	37	9
Tuerca de lastre	5	1	7	6
Tuerca para pruebas grande	12	2	7	9
Tuerca para pruebas pequeña	1	2	4	4

Tabla 3.1: Características de elementos añadidos a la maqueta.

3.3 Montaje del banco de pruebas

Para llevar a cabo las pruebas de estabilidad ha sido necesario disponer de un recipiente rectangular, de unos 11 litros de capacidad, al cual se le colocó una escala graduada para poder comprobar la inclinación real del casco y contrastarla con la obtenida por la unidad de medición inercial. Para poder comparar ambas medidas, también se instaló en la esquina superior izquierda de la escala graduada un display LCD conectado a la placa Arduino Nano.

Tras tener montado el banco de pruebas, el siguiente paso fue colocar la maqueta. Para esto, primero se llenó de agua el recipiente hasta que el casco coincidiera con la escala graduada, de forma que las medidas tomadas sean lo más exactas posibles.

Para que la maqueta no cambiase de posición durante las pruebas de balance, ésta se ha sujetó al recipiente mediante un hilo de nailon en la parte superior de la proa y de la popa. Ambas sujeciones se colocaron de manera que no estuvieran demasiado tensas, para no afectar al movimiento de balance, evitando así resultados falsos en las pruebas.

Finalmente, se preparó un soporte en la maqueta para colocar la IMU en una superficie plana, para realizar la calibración antes de situarla en el casco. Este ha sido un punto muy delicado del proceso de montaje, ya que la IMU en ningún caso puede entrar en contacto con el agua del depósito. Como soporte se optó por acoplar un alfiler a cada banda del casco. Se tomó esta decisión ya que, por una parte, son piezas muy ligeras que prácticamente no afectan a la estabilidad del casco, y por otra parte, porque estos alfileres encajan perfectamente con los agujeros de la placa a la que se encuentra soldada la IMU, por lo que esta queda bien sujeta y sin riesgo de caerse al agua del depósito.

3.4 Obtención del periodo de balance

Para testear el funcionamiento del dispositivo se optó por dos métodos distintos para la obtención de datos en el banco de pruebas:

a) El primer método ha sido la obtención del periodo a través de la escala graduada, situada detrás del casco diseñado, que permite ver los ángulos de inclinación, al mismo tiempo que se calcula el tiempo de dichas oscilaciones a través de una serie de grabaciones en vídeo, mediante a partir de la velocidad en fps (fotogramas por segundo). Para poder obtener la mayor precisión posible, los vídeos se grabaron a 120 fps. Posteriormente estas grabaciones se han reproducido a cámara lenta, para poder calcular el periodo de una manera fiable y cómoda.

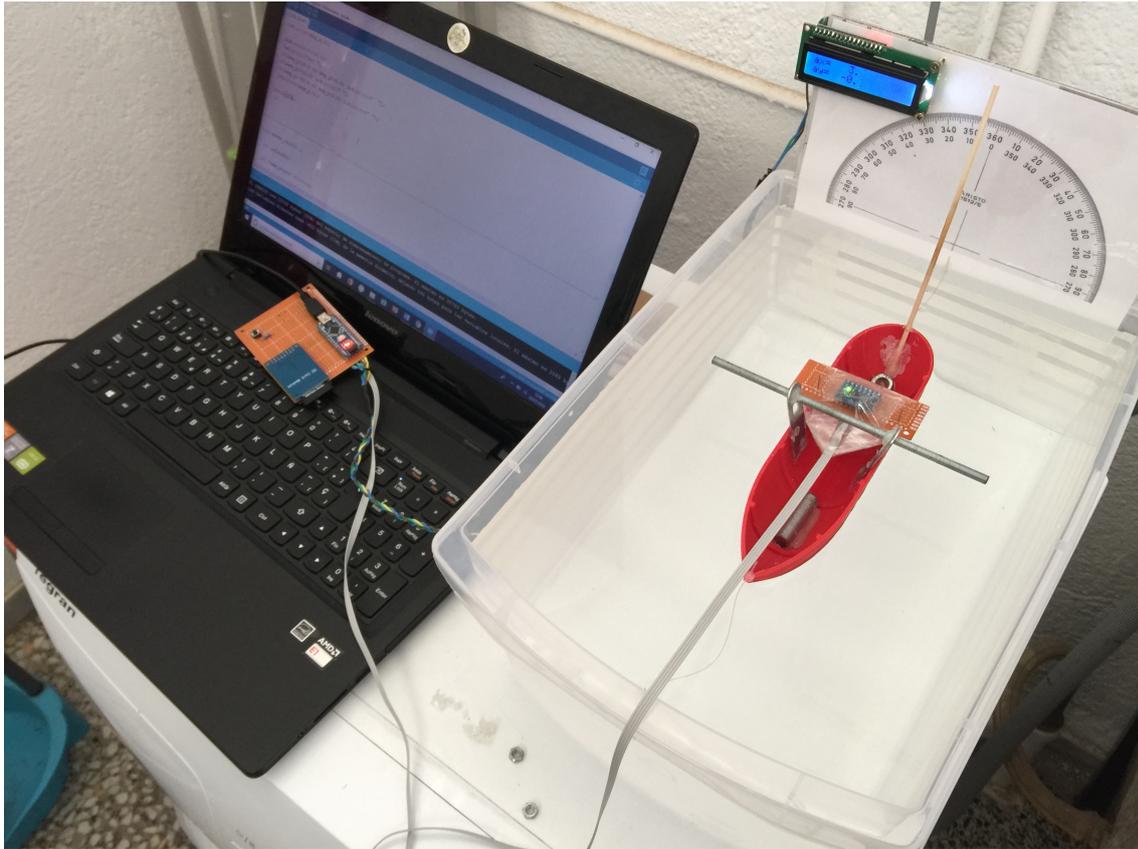


Figura 3.4: Montaje final del banco de pruebas.

Fuente: Elaboración propia.

b) El segundo método se basa en la obtención del periodo de balance a través del dispositivo diseñado. Para esto se toman los ángulos y el tiempo de las oscilaciones a través de la IMU y el software diseñado. Tras esto se procede al guardado de los datos en una tarjeta de memoria externa SD. Los datos de esta tarjeta SD se introducen en un documento de Excel donde se representan gráficamente para poder apreciar el movimiento oscilatorio, de donde se pueden medir los intervalos necesarios para tomar el periodo.

Los datos de las oscilaciones se han tomado en tres montajes distintos, para comprobar la diferencia del periodo cuando se varía el momento de inercia. Esta variación del momento se produce al añadir carga o modificar los pesos internos de la maqueta, en este caso se optó por aplicar pesos en el espárrago de carga a distintas distancias del eje longitudinal del casco. Estos montajes son los siguientes:

- Montaje 1: sin pesos en el espárrago de carga.
- Montaje 2: colocar en el espárrago de carga del casco una tuerca de 1 gramo de masa por cada costado del casco, separado del mismo cada una 0,5 cm.
- Montaje 3: colocar en el espárrago de carga del casco una tuerca de 1 gramo de masa por cada costado del casco, separado del mismo cada una 2 cm.

3.5 Cálculos

Con los datos obtenidos a través del dispositivo (ángulos y periodo doble de balance), se intentó calcular la altura metacéntrica y el momento de inercia a través la ecuación 1.25.

Sin embargo, el dato del factor de corrección (f) de la ecuación 1.25 es un supuesto, debido al desconocimiento de este por las características de la maqueta, ya que esta se ha diseñado desde cero

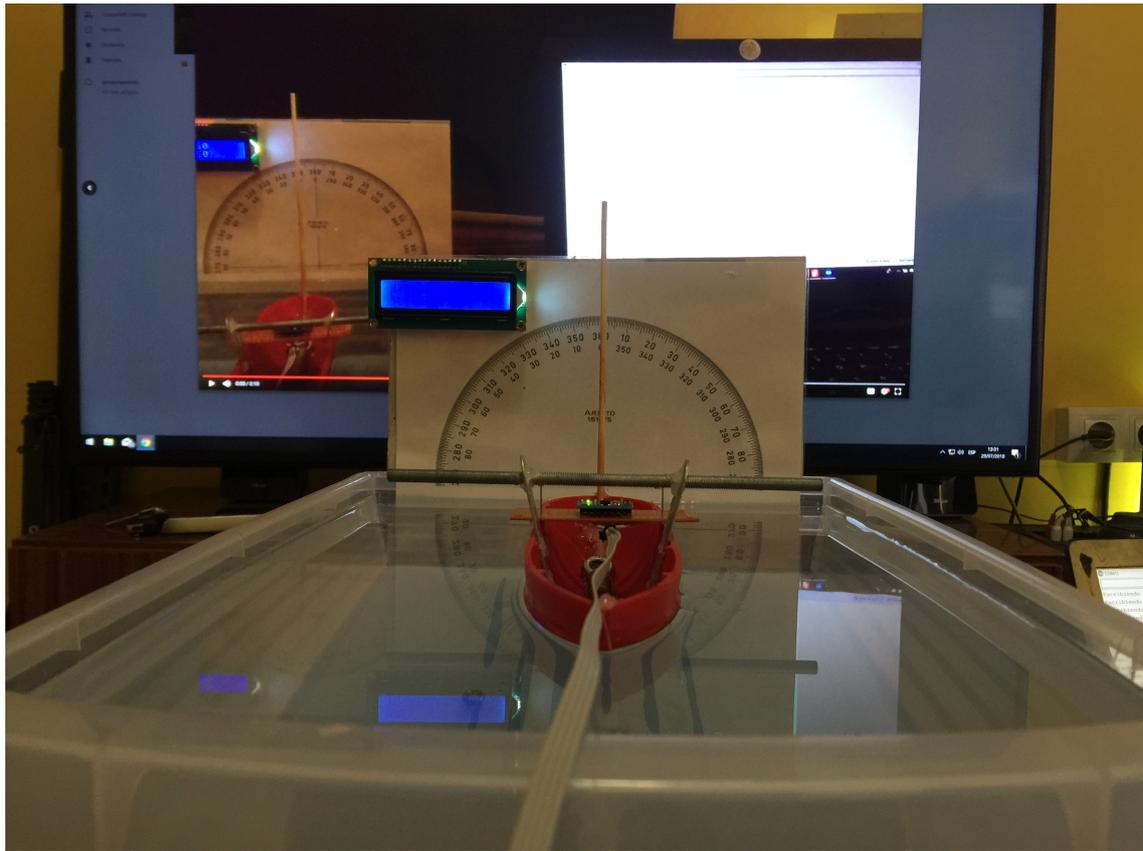


Figura 3.5: Montaje para cálculo de periodo con la IMU.

Fuente: Elaboración propia

sin basarse en ningún tipo de buque real. Al desconocer este factor, se ha intentado obtener mediante un cálculo iterativo¹. No obstante, el resultado que se ha obtenido no es coherente, obteniendo una altura metacéntrica demasiado baja, y en algunos casos negativa, lo que no se correspondía con el comportamiento de la maqueta.

Por este motivo, se optó por hallarla a través de la fórmula de la estabilidad estática (1.21). Calculada la altura metacéntrica de esta forma, es posible sustituir GM en la ecuación 1.20, y obtener el radio de giro equivalente (k), para calcular posteriormente, el momento de inercia total (I_t).

Teniendo el dato de la altura metacéntrica, obtenido de una manera fiable, se puede obtener un factor de corrección para el modelo diseñado, que puede servir para el cálculo aproximado del momento de inercia de embarcaciones de características similares.

3.6 Resultados

3.6.1 Determinación del periodo doble de balance

Para calcular los periodos de los montajes planteados en el apartado 3.4 es necesario fijarse en los gráficos obtenidos de la amortiguación del balance del casco, tomando una inclinación determinada como referencia, y midiendo el tiempo transcurrido hasta que la maqueta se encuentre exactamente en la misma posición. Estas tomas de periodos se muestran en las tablas 3.5, 3.6 y 3.7.

A modo de ejemplo, bajo cada tabla se muestra uno de los gráficos obtenidos por la IMU representando la amortiguación del balance.

¹Proceso mediante el cual se aplica una función repetidamente, usando la salida de una iteración como la entrada a la siguiente.

Montaje 1: se han realizado 5 tomas del periodo utilizando simultáneamente la escala graduada y la IMU.

Toma de periodo	Escala (T)	IMU (T)	Diferencia
1	1,63s	1,50s	0,13s
2	1,70s	1,67s	0,03s
3	1,51s	1,50s	0,01s
4	1,50s	1,75s	0,25s
5	1,73s	1,69s	0,04s
Periodo medio	1,61 s	1,62 s	0,01 s

Tabla 3.2: Diferencias de periodo entre escala e IMU sin peso en la maqueta.

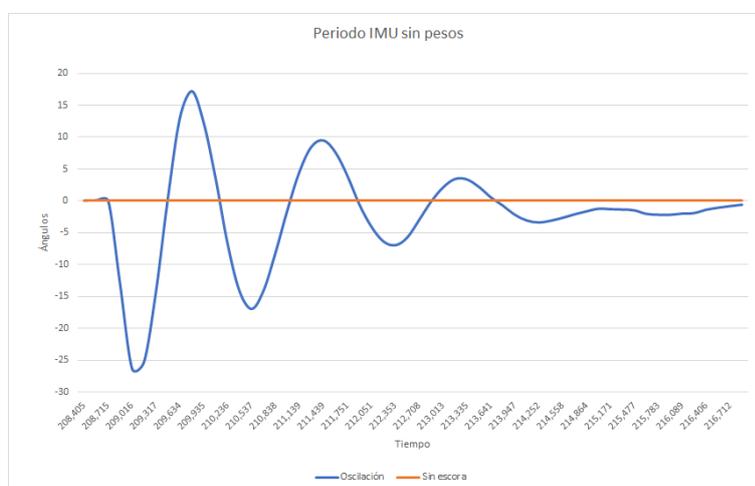


Figura 3.6: Amortiguación del balance para el casco libre de pesos (toma 3).

Fuente: Elaboración propia.

Montaje 2: se han realizado 5 tomas del periodo utilizando simultáneamente la escala graduada y la IMU.

Toma de periodo	Escala (T)	IMU (T)	Diferencia
1	1,87s	1,98s	0,11s
2	1,90s	2,00s	0,10s
3	1,95s	2,10s	0,15s
4	2,19s	1,96s	0,23s
5	2,20s	1,90s	0,30s
Periodo medio	2,02s	1,98 s	0,04 s

Tabla 3.3: Diferencias de periodo entre escala e IMU con tuercas separadas 0,5cm del casco.



Figura 3.7: Amortiguación del balance para pesos laterales a 0,5 cm de la banda (toma 3).

Fuente: Elaboración propia

Montaje 3: se han realizado 5 tomas del periodo utilizando simultáneamente la escala graduada y la IMU.

Toma de periodo	Escala (T)	IMU (T)	Diferencia
1	1,63s	1,58s	0,05s
2	1,75s	1,70s	0,05s
3	1,55s	1,65s	0,10s
4	1,16s	1,10s	0,06s
5	1,80s	1,87s	0,07s
Periodo medio	1,57s	1,58s	0,01 s

Tabla 3.4: Diferencias de periodo entre escala e IMU con tuercas separadas 2cm del casco.



Figura 3.8: Amortiguación del balance para pesos laterales a 2 cm de la banda (toma 2).

Fuente: Elaboración propia

Como se ha observado en las gráficas anteriores, las figuras 3.6 y 3.7, presentan una oscilación amortiguada óptima para su estudio. Estas difieren levemente entre ellas debido a la variación del momento de inercia y de la altura metacéntrica, causada por la adición de peso y separación de éste del eje longitudinal del casco.

Observando la representación gráfica de la amortiguación del balance utilizando pesos laterales a 2 cm de la banda, se estima que los datos obtenidos no son adecuados para su estudio, ya que no proporcionan periodos coherentes. Esto se debe a que, durante la realización de las pruebas, cuando se intentó determinar la oscilación con este supuesto, al aumentar el momento de inercia con una separación mayor de los pesos, el casco no es capaz de continuar con un periodo estable. Cuando se hace girar hacia una banda para dar la fuerza que produce la oscilación, al recuperar e ir hacia la otra, se produce una zozobra que arrastra los cables de nailon y el cableado de la IMU, produciendo una tensión que reduce drásticamente el balance, proporcionando los datos que se representan en la gráfica (figura 3.8).

3.6.2 Valores de altura metacéntrica

Mediante el uso de las ecuaciones correspondientes para el cálculo de la altura metacéntrica (GM) mencionadas en el apartado 3.5, se han obtenido los siguientes datos:

Montaje	$w(kg)$	$d(m)$	$D(kg)$	$\theta(^{\circ})$	GM (m)
1	0,001	0,030	0,127	8,7	1,544E-3
2	0,001	0,035	0,129	15	1,012E-3

Tabla 3.5: Resultados de altura metacéntrica.

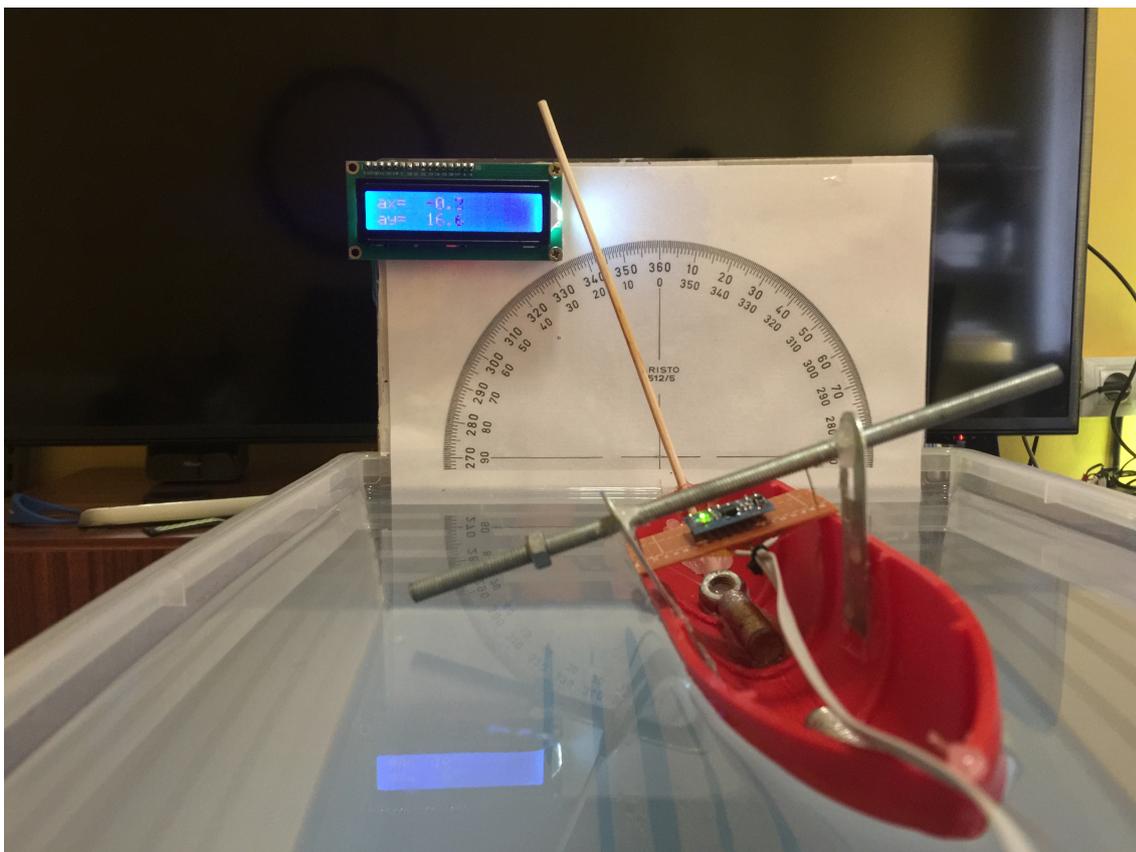


Figura 3.9: Obtención de parámetros de estabilidad estática.

Fuente: Elaboración propia.

3.6.3 Valores de momento de inercia

En la siguiente tabla 3.6 se pueden observar los resultados obtenidos utilizando las ecuaciones correspondientes de la sección 3.5. Se comienza calculando el periodo de balance para cada montaje.

Posteriormente se desplaza un peso para obtener una escora (ver figura 3.9 en el apartado anterior) y calcular la altura metacéntrica de forma estática. Con estos dos datos se puede obtener el radio de giro equivalente a través de la ecuación del periodo doble de balance.

Montaje	$T(s)$	$GM(m)$	$g(m/s^2)$	$k(m)$
1	1,62	$1,544E - 3$	9,81	0,0317
2	1,98	$1,012E - 3$	9,81	0,0314

Tabla 3.6: Resultados de radio de giro equivalente.

Los datos anteriores de radio de giro equivalente permiten calcular el momento de inercia total de cada montaje (ver tabla 3.7) a través de la ecuación 1.16.

Montaje	$m(kg)$	$r(m)$	$m_t(kg)$	$k(m)$	$I(kg \cdot m^2)$
1	0	0	0,127	0,0317	$1,277E - 4$
2	0,002	0,030	0,129	0,0314	$1,271E - 4$

Tabla 3.7: Resultados de momento de inercia.

3.6.4 Valores de factor de corrección

Los resultados obtenidos del cálculo del factor de corrección utilizando las ecuaciones mencionadas en el apartado 3.5 son los que se muestran en la siguiente tabla:

Montaje	Manga (m)	$T(s)$	$GM(m)$	f
1	0,05	1,62	$1,544E - 3$	$1,273E - 4$
2	0,05	1,98	$1,012E - 3$	$1,279E - 4$
Media	0,05	1,80	$1,278E - 3$	$1,276E - 4$

Tabla 3.8: Resultados de factor de corrección f para la determinación aproximada del momento de inercia.

Dadas las dificultades presentadas para la obtención del factor de corrección, y a la inexistencia de referencias externas, se ha pensado en establecer una correlación para modelos de cascos a escala. Se ha llevado a cabo el cálculo del factor de corrección, pero en este caso, con las medidas trasladadas a milímetros.

Montaje	Manga (mm)	$T(s)$	$GM(mm)$	f
1	50	1,62	1,544	0,0403
2	50	1,98	1,012	0,0398
Media	50	1,62	1,278	0,0400

Tabla 3.9: Resultados de factor de corrección en milímetros.



INFO ABOUT RIGHTS
1 809028 251697
www.safecreative.org/work

4 Discusión de los resultados

El banco de pruebas utilizado ha servido para la obtención de ángulos y frecuencias a través de la observación y detección de los movimientos realizados por la maqueta. La toma de datos se ha dispuesto de manera que se ha realizado una doble comprobación de los ángulos y las frecuencias tal y como se ve en las secciones 3.4 y 3.6.1. Se puede garantizar que es un modo fiable de toma de datos ya que los resultados obtenidos son similares en ambos métodos, con una diferencia menor a un 2 %. Se pueden añadir mejoras en base a la experiencia obtenida en el montaje 3, como ya se explicó en la sección 3.6.1. Estos errores se dan debido al diseño inadecuado de la maqueta, puesto que está no se ha basado en ninguna embarcación real. Además, al poseer un tamaño reducido, por los motivos que se exponen en el apartado 3.1, no es posible agregarle un peso elevado en el esparrago de carga (solo se han podido añadir 2 tuercas de 1 gramo cada una) para poder realizar un mayor abanico de mediciones.

Basándose en los datos obtenidos con las pruebas y los cálculos realizados (sección 3.6), se ha podido observar en los resultados de altura metacéntrica, periodo de balance y momento de inercia, que es posible llevar a cabo estos mismos cálculos en una embarcación de recreo. Se ha comprobado que cuando colocamos pesos en el esparrago de carga a una distancia considerable al punto de crujía de la embarcación se modifica el periodo de balance y el momento de inercia, de manera creciente cuanto mas peso se añada y más sea la separación de estos pesos del centro longitudinal, tal y como es posible apreciar en la justificación teórica (apartado 1.9). De igual manera, se puede ver que en el apartado 3.6.2 se observa que la altura metacéntrica disminuye cuando se añaden las tuercas al esparrago de carga. Esto produce que el centro de gravedad del buque se eleve, ya que el esparrago se encuentra situado por encima de la cubierta a 20 mm. Como consecuencia se reduce la altura entre el metacentro y el centro de gravedad, produciendo un resultado lógico, tal y como se ha visto en el punto 1.7.2. En cuanto al momento de inercia, éste disminuye ligeramente en vez de aumentar. Esto puede deberse a los efectos de fricción y a el agua arrastrada durante el movimiento de la maqueta, frente al corto periodo de balance medido.

En el caso de que las mediciones se llevasen a cabo en embarcaciones de recreo, como se mencionó en el párrafo anterior, si se quisieran llevar a cabo comprobaciones para verificar el correcto funcionamiento del dispositivo, se podría hacer de las siguientes maneras:

- a) Utilizando un péndulo unido a una escala graduada como la utilizada en este proyecto. De este modo se puede verificar la exactitud de la medición de la inclinación de la IMU.
- b) Instalando en el propio dispositivo un nivel para comprobar que la calibración se haya llevado a cabo correctamente y marque una inclinación de 0° cuando deba.

En las embarcaciones de recreo se podría llevar a cabo la determinación de GM y K de una manera sencilla, realizando una modificación en el código de programación, de manera que introduciendo los datos dimensionales requeridos de la embarcación, se realicen las operaciones necesarias de manera automática para determinar estos parámetros.

Con la realización de los cálculos descritos en este proyecto, si se trasladasen a una embarcación real, en la cual se quiera cambiar su propulsión a eléctrica, se podría saber con exactitud donde colocar las baterías sin que el peso de estas varíen la estabilidad ni el comportamiento dinámico de la embarcación. El dispositivo permite calcular el radio de giro equivalente y la altura metacéntrica de la antigua distribución de pesos de la embarcación. De esta manera, al realizar el cambio a propulsión eléctrica, es posible mantener los parámetros mencionados anteriormente.



INFO ABOUT RIGHTS
1 809028 251697
www.safecreative.org/work

5 Conclusiones

El objetivo del proyecto en un principio era el desarrollo e implementación de un dispositivo capaz de calcular la altura metacéntrica y el momento de inercia de manera automática para establecer la estabilidad de una embarcación.

El fin de este proyecto se ha visto ligeramente alterado, debido a la complejidad del diseño del software, y agravado por la falta de tiempo necesario para un proyecto de estas características. Por esto se ha optado por un diseño más sencillo, el cual nos proporciona los datos de tiempo y ángulos de oscilación, los cuales son procesados externamente para llegar al objetivo principal del proyecto.

Este tipo de dispositivo permite su implementación en cualquier tipo de embarcación, conociendo un índice de su estabilidad, permitiendo el traslado de peso en su interior y comprobar si la estabilidad sigue siendo la óptima. El dispositivo diseñado es de utilidad para el objetivo que ha dado pie a este proyecto: la conversión de una embarcación de recreo propulsada por un motor de combustión interna, a una propulsión 100 % eléctrica, ya que una de las mayores complicaciones que tiene este tipo de conversión es la colocación de las baterías, debido a su elevado peso.

Finalmente, durante el desarrollo del proyecto, han ido surgiendo ideas que se pueden aplicar como posibles mejoras, las cuales por motivos de plazos no se pudieron llevar a cabo. Estas mejoras son las siguientes:

- Desarrollo de una maqueta para pruebas de mayor tamaño, como se mencionó en el capítulo 4, para que sea posible realizar estudios de amortiguación del balance evitando la zozobra del casco y la atenuación del balance con los elementos de sujeción.
- Instalar un dispositivo bluetooth para obtener los datos de la IMU directamente en la pantalla de un smartphone mediante el diseño de una aplicación e interactuar con el hardware.
- Diseñar el software de manera que el dispositivo calcule automáticamente los parámetros requeridos.
- Realizar un diseño de hardware más compacto, el cual cuente con un nivel en la parte superior para realizar una calibración exacta, y un imán en su parte posterior para su fácil colocación en la embarcación.



INFO ABOUT RIGHTS
1 809028 251697
www.safecreative.org/work

Bibliografía

- [1] ARDUINO, “Arduino — Home”. 2018. URL <https://www.arduino.cc/>.
- [2] CARO, R. V., “La importancia de la experiencia de estabilidad en los buques”. *Revista general de marina*, 269(4), 2015: 721–732.
- [3] GALÁN, F.; LUIS, C. E. M., *Diseño de un sensor para una boya oceanográfica de bajo coste*. Univ. de La Laguna, 2015.
- [4] GONZÁLEZ, F. J. V.; RUEDA RODRÍGUEZ, T. M.; MOYANO PÉREZ, E., “Modelos Matemáticos Para El Control De Gobierno De Buques”. 2004.
- [5] GRAELLS, J. A.; LÓPEZ, J.; CASIN, J. Q., “Modelización de la dinámica de un buque”. 4(3), 1980: 137–146.
- [6] GUERRERO B., B. L., *Periodo de balance*. Humboldt Marine Training, 2011.
- [7] MOLLAND, A. F., *The maritime engineering reference book: a guide to ship design, construction and operation*. Elsevier, 2011.
- [8] PALOS FORCANO, C., *Análisis de la estabilidad de un buque durante la operación de descarga de una carga pesada*. Univ. Politèc. de Catalunya, 2014.
- [9] PUIG, J. O., *Teoría del buque: ola trocoidal, movimientos y esfuerzos*. Univ. Politèc. de Catalunya, 1998.
- [10] SLABAUGH, G. G., “Computing Euler angles from a rotation matrix”. 6(2000), 1999: 39–63.
- [11] WIKIPEDIA, “Estabilidad transversal — Wikipedia, La enciclopedia libre”. 2016. URL <https://es.wikipedia.org/>.
- [12] WIKIPEDIA, “Ángulos de Euler — Wikipedia, La enciclopedia libre”. 2017. URL <https://es.wikipedia.org/>.



INFO ABOUT RIGHTS
1 809028 251697
www.safecreative.org/work

Anexos



INFO ABOUT RIGHTS
1 809028 251697
www.safecreative.org/work

A Código Fuente

```
1  /*****
2  * Inclination Calculation
3  * @version 1.0
4  * @author Gabriel Seco Herrera <gabri.s.6.22@gmail.com> <alu0100829594@ull.edu.es>
5  * @author Jose Daniel Alamo Perez <jdanielap93@gmail.com> <alu0100811391@ull.edu.es>
6  * @author Carlos Efren Mora <cemora@gmail.com> <carmora@ull.edu.es>
7  *
8  * This code is licensed under GPLv3 GNU General Public License
9  *****/
10
11 /* =====
12 I2Cdev device library code is placed under the MIT license
13 Copyright (c) 2011 Jeff Rowberg
14
15 Permission is hereby granted, free of charge, to any person obtaining a copy
16 of this software and associated documentation files (the "Software"), to deal
17 in the Software without restriction, including without limitation the rights
18 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
19 copies of the Software, and to permit persons to whom the Software is
20 furnished to do so, subject to the following conditions:
21
22 The above copyright notice and this permission notice shall be included in
23 all copies or substantial portions of the Software.
24
25 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
26 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
27 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
28 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
29 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
30 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
31 THE SOFTWARE.
32 =====*/
33
34 //I2C Libraries to control the mpu6050
35 //The library MPU6050.h needs I2Cdev.h, I2Cdev.h needs Wire.h
36 #include <I2Cdev.h>
37 #include <MPU6050.h>
38 #include <Wire.h>
39 #include <math.h>
40 #include <SD.h>
41 #include <LiquidCrystal_I2C.h>
42
43 LiquidCrystal_I2C lcd(0x3F,16,2);
44
45 int ready;
46
47 //Change this 3 variables if you want to fine tune the skecth to your needs
48 int buffersize=1000; //Amount of readings used to average, make it higher to
49 //get more precision but sketch will be slower (default:1000)
50 int acel_deadzone=8; //Acelerometer error allowed, make it lower to get more
51 //precision, but sketch may not converge (default:8)
52 int giro_deadzone=1; //Giro error allowed, make it lower to get more precision,
53 //but sketch may not converge (default:1)
54
55 //The address of the MPU6050 can be 0x68 or 0x69, depending on the state of ADO If
56 //not specified, 0x68 will be implicit
57 MPU6050 sensor(0x68);
58
59
```

```

55 //Variables obtained by accelerometer and the gyroscope of the x, y, z axes without
    processing
56 int16_t ax,ay,az;
57 int16_t gx,gy,gz;
58
59 int mean_ax, mean_ay, mean_az, mean_gx, mean_gy, mean_gz, state=0;
60 int ax_offset, ay_offset, az_offset, gx_offset, gy_offset, gz_offset;
61
62 //Initial angles of inclination calculated with the accelerometer
63 float angAccel_x, angAccel_y;
64
65 //Variables of allocation of time: previous time and variation of time. To
    calculate the angle with the gyroscope
66 long time_prev, dt;
67
68 //Gyroscope variables scaled to degrees
69 float gx_deg;
70 float gy_deg;
71
72 //Angles calculated after the filter
73 float ang_x;
74 float ang_y;
75
76 //The previous angles to take them into account in the following loop
77 float ang_x_prev;
78 float ang_y_prev;
79
80 //Constant to assign pin 7 to the control button
81 const int button_call=7;
82
83 File myFile;
84
85 void setup() {
86
87     //Pin 7 is assigned as an input, to be used by the button
88     pinMode(7,INPUT);
89
90     //join I2C bus (I2Cdev library doesn't do this automatically)
91     Wire.begin();
92
93     //400kHz I2C clock (200kHz if CPU is 8MHz
94     TWBR = 24;
95
96     //Initialize device
97     sensor.initialize();
98     lcd.init();
99     lcd.backlight();
100
101     //Initialize serial communication
102     Serial.begin(9600);
103
104     SD.begin(9);
105
106     if(!SD.exists("datalog.csv"))
107     {
108         myFile = SD.open("datalog.csv", FILE_WRITE);
109         if(myFile) {
110             Serial.println("New file, typing header");
111             myFile.println("Time(ms),Angle_X,Angle_Y");
112             myFile.close();
113         } else{
114
115             Serial.println("Error creating the file datalog.csv");
116         }
117     }
118 }
119
120
121 void loop() {
122
123     //the file opens
124     myFile = SD.open("datalog.csv", FILE_WRITE);
125

```

```

126   if (myFile) {
127       Serial.print("writing in the SD: ");
128
129       myFile.print(millis());
130       myFile.print(",");
131       myFile.print(ang_x);
132       myFile.print(",");
133       myFile.println(ang_y);
134
135       Serial.print("Time(ms)=");
136       Serial.print(millis());
137       Serial.print(",Angle_X=");
138       Serial.print(ang_x);
139       Serial.print(",Angle_Y=");
140       Serial.println(ang_y);
141
142   } else {
143       //If the file didn't open, print an error:
144       Serial.println("Error opening the file");
145   }
146   }
147   delay(100);
148
149
150   int value = digitalRead(button_call);
151
152   if(value){
153       delay(1000);
154
155       start_call();
156       sensor.setXAccelOffset(ax_offset);
157       sensor.setYAccelOffset(ay_offset);
158       sensor.setZAccelOffset(az_offset);
159
160       sensor.setXGyroOffset(gx_offset);
161       sensor.setYGyroOffset(gy_offset);
162       sensor.setZGyroOffset(gz_offset);
163   }
164
165   //Read accelerations and angular velocities
166   sensor.getAcceleration(&ax,&ay,&az);
167   sensor.getRotation(&gx,&gy,&gz);
168
169   //Calculate the inclination angles on the X and Y axis with the accelerometer
170   angAccel_x = atan(float(ax)/float(az))*180.0/3.14;
171   angAccel_y = atan(float(ay)/float(az))*180.0/3.14;
172
173   //Scale the readings of the gyroscope to degrees, the maximum it measures are 250
   degrees
174   gx_deg = gx * (250.0/32768.0);
175   gy_deg = gy * (250.0/32768.0);
176
177   dt = (millis() - time_prev) / 1000,0;
178   time_prev = millis();
179
180   //Calculate rotation angles with complementary filter (gyroscope calculation is
   done here)
181   ang_x = 0.35 * (ang_x_prev + gx_deg * dt) + 0.65 * angAccel_x;
182   ang_y = 0.35 * (ang_y_prev + gy_deg * dt) + 0.65 * angAccel_y;
183
184   //Assignment of the angles calculated to the variables of the previous angles to
   take them into account in the next loop
185   ang_x_prev = ang_x;
186   ang_y_prev = ang_y;
187
188
189   lcd.setCursor(0,0);
190   lcd.print("ax= ");
191   if(ang_x<10.0 && ang_x>=0.0) lcd.print(" ");
192   if(ang_x>=10.0) lcd.print(" ");
193   if(ang_x>-10.0 && ang_x<0.0) lcd.print(" ");
194   lcd.print(ang_x,1);
195

```

```

196   lcd.setCursor(0,1);
197   lcd.print("ay= ");
198   if(ang_y<10.0 && ang_y>=0.0) lcd.print(" ");
199   if(ang_y>=10.0) lcd.print(" ");
200   if(ang_y>-10.0 && ang_y<0.0) lcd.print(" ");
201   lcd.print(ang_y,1);
202   delay(50);
203   myFile.close(); //cerramos el archivo
204
205 }
206
207
208 ////////////////////////////////////////  FUNCTIONS  ////////////////////////////////////////
209
210 void start_call() {
211
212   int button;
213
214   int time_start= millis();
215
216   //wait for ready
217   do{
218     lcd.setCursor(0,0);
219     lcd.print("Press the button");
220     lcd.setCursor(0,1);
221     lcd.print("to cancel");
222     delay (3000);
223     lcd.clear();
224
225     button = digitalRead(button_call);
226
227     if(button) {
228       lcd.setCursor(0,0);
229       lcd.print("Returning to");
230       lcd.setCursor(0,1);
231       lcd.print("normal mode");
232       delay(2000);
233       lcd.clear();
234       return;
235     }
236   }
237
238   //Empty buffer again
239   while ((millis()-time_start)<3000);
240
241   //Start message
242   lcd.setCursor(0,0);
243   lcd.print("MPU6050");
244   lcd.setCursor(0,1);
245   lcd.print("Calibration");
246   delay(2000);
247   lcd.clear();
248
249   lcd.setCursor(0,0);
250   lcd.print("Place the mpu");
251   lcd.setCursor(0,1);
252   lcd.print("in horizontal");
253   delay(2000);
254   lcd.clear();
255
256
257   //Reset offsets
258   sensor.setXAccelOffset(0);
259   sensor.setYAccelOffset(0);
260   sensor.setZAccelOffset(0);
261   sensor.setXGyroOffset(0);
262   sensor.setYGyroOffset(0);
263   sensor.setZGyroOffset(0);
264
265   if (state==0){
266     lcd.setCursor(0,0);
267     lcd.print("Reading MPU");
268     lcd.setCursor(0,1);

```

```

269     lcd.print("for first time");
270     meansensors();
271     state++;
272     delay(1500);
273     lcd.clear();
274 }
275
276 if (state==1) {
277     lcd.setCursor(0,0);
278     lcd.print("Calculating");
279     lcd.setCursor(0,1);
280     lcd.print("offsets");
281
282     calibration();
283     state++;
284     delay(1500);
285     lcd.clear();
286 }
287
288 if (state==2) {
289     meansensors();
290
291     /*uncomment to see the offset measurements through the serial port
292
293     Serial.println("FINISHED!");
294     Serial.print("\nSensor readings with offsets:\t");
295     Serial.print(mean_ax);
296     Serial.print("\t");
297     Serial.print(mean_ay);
298     Serial.print("\t");
299     Serial.print(mean_az);
300     Serial.print("\t");
301     Serial.print(mean_gx);
302     Serial.print("\t");
303     Serial.print(mean_gy);
304     Serial.print("\t");
305     Serial.println(mean_gz);
306     Serial.print("Your offsets:\t");
307     Serial.print(ax_offset);
308     Serial.print("\t");
309     Serial.print(ay_offset);
310     Serial.print("\t");
311     Serial.print(az_offset);
312     Serial.print("\t");
313     Serial.print(gx_offset);
314     Serial.print("\t");
315     Serial.print(gy_offset);
316     Serial.print("\t");
317     Serial.println(gz_offset);
318     Serial.println("\nData is printed as: acelX acelyZ giroX giroY giroZ");
319     Serial.println("Check that your sensor readings are close to 0 0 16384 0 0 0");
320     Serial.println("If calibration was succesful write down your offsets so you can
321         set them in your projects using something similar to mpu.setXAccelOffset(
322         youroffset)");
323
324     lcd.setCursor(0,0);
325     lcd.print("gx=");
326     lcd.print(mean_gx);
327
328     lcd.print(" gy=");
329     lcd.print(mean_gy);
330
331     lcd.print(" gz=");
332     lcd.print(mean_gz);
333
334     lcd.setCursor(0,1);
335     lcd.print("Press to exit");
336
337     do{
338         button = !digitalRead(button_call);
339     }

```

```

340     while (button);
341         delay(1500);
342         lcd.clear();
343     }
344 }
345 }
346 }
347 }
348 }
349 void meansensors(){
350
351     long i=0, buff_ax=0, buff_ay=0, buff_az=0, buff_gx=0, buff_gy=0, buff_gz=0;
352
353     while (i<(buffersize+101)){
354
355         //Read raw accel/gyro measurements from device
356         sensor.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
357
358         //First 100 measures are discarded
359         if (i>100 && i<=(buffersize+100)){
360             buff_ax=buff_ax+ax;
361             buff_ay=buff_ay+ay;
362             buff_az=buff_az+az;
363             buff_gx=buff_gx+gx;
364             buff_gy=buff_gy+gy;
365             buff_gz=buff_gz+gz;
366         }
367
368         if (i==(buffersize+100)){
369             mean_ax=buff_ax/buffersize;
370             mean_ay=buff_ay/buffersize;
371             mean_az=buff_az/buffersize;
372             mean_gx=buff_gx/buffersize;
373             mean_gy=buff_gy/buffersize;
374             mean_gz=buff_gz/buffersize;
375         }
376         i++;
377
378         //Needed so we don't get repeated measures
379         delay(2);
380     }
381 }
382
383
384 void calibration(){
385
386     ax_offset=-mean_ax/8;
387     ay_offset=-mean_ay/8;
388     az_offset=(16384-mean_az)/8;
389
390     gx_offset=-mean_gx/4;
391     gy_offset=-mean_gy/4;
392     gz_offset=-mean_gz/4;
393
394     while (1){
395         ready=0;
396         sensor.setXAccelOffset(ax_offset);
397         sensor.setYAccelOffset(ay_offset);
398         sensor.setZAccelOffset(az_offset);
399
400         sensor.setXGyroOffset(gx_offset);
401         sensor.setYGyroOffset(gy_offset);
402         sensor.setZGyroOffset(gz_offset);
403
404         meansensors();
405
406         for(int i=0;i<=16;i++){
407             {
408
409                 //The screen is cleaned
410                 lcd.clear();
411
412                 //The cursor is placed

```

```
413     lcd.setCursor(i, 0);
414
415     //The text is written
416     lcd.print("...");
417
418     //Wait command
419     delay(100);
420 }
421
422 for(int i=16;i>=1;i--)
423 {
424     lcd.clear();
425     lcd.setCursor(i, 1);
426     lcd.print("...");
427     delay(100);
428     lcd.clear();
429 }
430
431 if (abs(mean_ax)<=acel_deadzone) ready++;
432 else ax_offset=ax_offset-mean_ax/acel_deadzone;
433
434 if (abs(mean_ay)<=acel_deadzone) ready++;
435 else ay_offset=ay_offset-mean_ay/acel_deadzone;
436
437 if (abs(16384-mean_az)<=acel_deadzone) ready++;
438 else az_offset=az_offset+(16384-mean_az)/acel_deadzone;
439
440 if (abs(mean_gx)<=giro_deadzone) ready++;
441 else gx_offset=gx_offset-mean_gx/(giro_deadzone+1);
442
443 if (abs(mean_gy)<=giro_deadzone) ready++;
444 else gy_offset=gy_offset-mean_gy/(giro_deadzone+1);
445
446 if (abs(mean_gz)<=giro_deadzone) ready++;
447 else gz_offset=gz_offset-mean_gz/(giro_deadzone+1);
448 if (ready==6)break;
449 }
450 }
```



INFO ABOUT RIGHTS
1 809028 251697
www.safecreative.org/work

B Esquema de conexiones

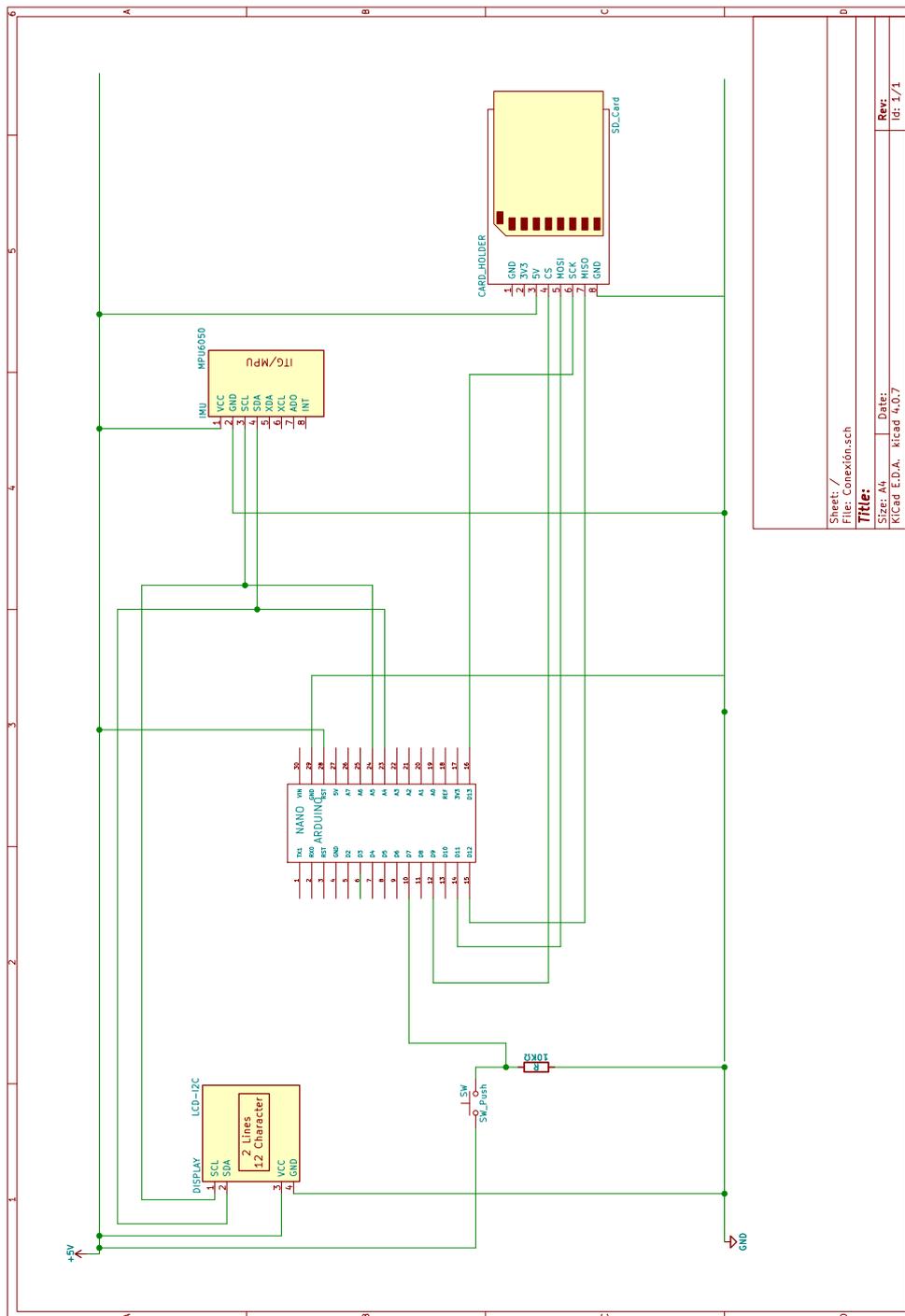


Figura B.1: Esquema de conexiones.

Fuente: Elaboración propia