



**UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA**



**DISEÑO, MODELADO Y SIMULACIÓN DE UN VEHÍCULO
SUBMARINO AUTÓNOMO NO TRIPULADO**

POR

Lamar Contreras Zambrano

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para optar al título profesional de Ingeniero(a) Civil Electrónico(a)

Profesor(es) Guía
Juan Pablo Segovia

Agosto 2023
Concepción (Chile)

©2023 Lamar Contreras Zambrano

©2023 Lamar Contreras Zambrano

Ninguna parte de esta tesis puede reproducirse o transmitirse bajo ninguna forma o por ningún medio o procedimiento, sin permiso por escrito del autor.

A mi Madre, Abuelos y Amigos.

Sumario

En este trabajo se presenta el diseño de un vehículo submarino autónomo no tripulado abreviado AUV, comprendido por sensores capaces de percibir la posición y ambiente que lo rodea.

Se describe el modelo del vehículo junto a sus variables hidrostáticas, hidrodinámicas y cinemáticas, así como también, el de los propulsores que son requeridos para la movilidad de este.

Se presenta el proceso de obtención de los parámetros hidrodinámicos, específicamente la masa agregada y las fuerzas de amortiguamiento, las cuales son requeridas para la completa definición física del móvil.

Se muestra el sistema de regulación de posición, pose y altura para el diseño propuesto, principalmente este fue diseñado para facilitar el control de navegación autónoma.

De igual forma, se describe la estrategia de control difuso para la navegación autónoma junto con la ingeniería conceptual del sistema de localización del vehículo.

Por último, se presenta un listado de instrumentos necesarios para la implementación de este proyecto.

En este trabajo se logró diseñar, modelar y simular un vehículo autónomo submarino no tripulado junto a su sistema de navegación autónoma, donde se obtuvo una buena aproximación de los parámetros hidrodinámicos y además, el sistema de control permite al móvil moverse de manera segura hacia la meta propuesta.

Summary

In the following work, the design of an autonomous unmanned underwater vehicle, abbreviated as AUV, will be presented, comprised of sensors capable of perceiving its position and surrounding environment.

The model of the vehicle will be described together with its hydrostatic, hydrodynamic and kinematic variables, as well as the thrusters that are required for its mobility.

The process of obtaining the hydrodynamic parameters will be presented, specifically the added mass and the damping forces, which are required for the complete physical definition of the robot.

The position, pose and height regulation system for the proposed design will be shown, mainly this was designed to facilitate autonomous navigation system.

Similarly, the fuzzy control strategy for autonomous navigation will be described together with the conceptual engineering of the vehicle location system.

Finally, a list of instruments necessary for the implementation of this project will be presented.

In this work, it was possible to design, model and simulate an unmanned underwater autonomous vehicle together with its autonomous navigation system, where a good approximation of the hydrodynamic parameters was obtained and, in addition, the control system allows the mobile to move safely towards the proposed goal.

Tabla de contenidos

Contenido

| | |
|--|------|
| Sumario | iv |
| Summary | v |
| Tabla de contenidos..... | vi |
| Lista de tablas..... | ix |
| Lista de figuras | x |
| Nomenclatura | xiii |
| 1. Introducción..... | 1 |
| 1.1. Introducción General. | 1 |
| 1.2. Revisión bibliográfica..... | 1 |
| 2. Modelo..... | 7 |
| 2.1. Cinemática y ejes de referencia | 7 |
| 2.2. Modelo del vehículo. | 9 |
| 2.3. Modelo de Propulsores. | 13 |
| 2.4. Obtención de Parámetros Hidrodinámicos. | 14 |
| 2.4.1. Masa agregada..... | 14 |
| 2.4.2. Fuerzas de amortiguamiento | 20 |
| 3. Diseño del vehículo. | 23 |
| 3.1. Introducción. | 23 |
| 3.2. Consideraciones con respecto al diseño..... | 23 |
| 3.2.1. Con respecto a los sensores..... | 23 |
| 3.2.2. Con respecto a la estabilidad y seguridad del vehículo..... | 23 |
| 3.3. Diseño propuesto. | 24 |
| 3.4. Modelo del Diseño..... | 27 |
| 3.4.1. Parámetros hidrodinámicos..... | 27 |

| | | |
|--------|--|----|
| 3.4.2. | Implementación del modelo en Simulink..... | 30 |
| 3.4.3. | Obtención Modelo Propulsores..... | 31 |
| 4. | Regulación | 33 |
| 4.1. | Introducción | 33 |
| 4.2. | Principio de funcionamiento..... | 33 |
| 4.2.1. | Control de ψ y Velocidad..... | 33 |
| 4.2.2. | Control de Profundidad z y θ | 34 |
| 5. | Navegación | 36 |
| 5.1. | Introducción | 36 |
| 5.2. | Sistemas de navegación. | 36 |
| 5.2.1. | Localización. | 36 |
| 5.3. | Control de navegación. | 37 |
| 5.3.1. | Lógica difusa. | 37 |
| 5.3.2. | Controlador difuso..... | 38 |
| 5.3.3. | Propuesta de Control de Navegación. | 39 |
| 5.3.4. | Fusificación..... | 41 |
| 5.3.5. | Salida e integración de los comportamientos..... | 43 |
| 5.3.6. | Motor de inferencias: | 46 |
| 5.3.7. | Defusificación. | 49 |
| 5.3.8. | Simulación..... | 49 |
| 6. | Instrumentación | 52 |
| 6.1.1. | Introducción | 52 |
| 6.1.2. | Propulsión..... | 52 |
| 6.1.3. | Percepción. | 53 |
| 6.1.4. | Controlador..... | 55 |
| 6.1.5. | Iluminación..... | 56 |

| | | |
|--------|--|----|
| 6.1.6. | Alimentación..... | 57 |
| 7. | Conclusión..... | 59 |
| 7.1. | Trabajos Futuros. | 60 |
| | Referencias | 61 |
| 8. | Anexos | 63 |
| 8.1. | Anexo A. Diagrama Simulink..... | 63 |
| 8.2. | Anexo B. Códigos para la obtención de masa agregada..... | 72 |
| 8.2.1. | Código Principal..... | 72 |
| 8.2.2. | Subdivisión de elementos de superficie. | 74 |
| 8.2.3. | Código de creación de objeto polígono..... | 78 |
| 8.2.4. | Validación de Método. | 79 |
| 8.3. | Anexo C. Procesamiento de Datos de Sensores..... | 80 |
| 8.4. | Anexo D. Implementación de Defusificación de FLC..... | 83 |
| 8.4.1. | Código empleado en la defusificación. | 83 |
| 8.5. | Anexo E. Horas hombre requeridas | 85 |

Lista de tablas

| | |
|---|----|
| Tabla 3.1. Características físicas del vehículo..... | 25 |
| Tabla 3.2. Dimensiones de esponja de flotabilidad..... | 26 |
| Tabla 5.1. Reglas para comportamiento Evasión de obstáculos..... | 47 |
| Tabla 5.2. Reglas para comportamiento de seguimiento de meta..... | 48 |
| Tabla 6.1. Características del propulsor..... | 52 |
| Tabla 6.2. Características eléctricas ESC..... | 53 |
| Tabla 6.3. Características Sonar Ping360..... | 53 |
| Tabla 6.4. Características de Sensor de Presión..... | 54 |
| Tabla 6.5. Características de los transpondedores..... | 54 |
| Tabla 6.7. Características de Controlador de navegación..... | 55 |
| Tabla 6.8. Características Luces led..... | 56 |
| Tabla 6.9. Características Batería de Ion de Litio..... | 57 |
| Tabla 6.10. Estimación de precios de los instrumentos elegidos..... | 58 |
| Tabla E.1. Horas hombre mínimas requeridas..... | 85 |

Lista de figuras

| | |
|--|----|
| Fig 2.1. Imagen Ilustrativa de Ejes de Referencia Ref_t y Ref_c :..... | 8 |
| Fig.2.2. Imagen ilustrativa del efecto de la masa agregada sobre un objeto inserto en un fluido en movimiento acelerado..... | 14 |
| Fig. 2.3. Ejemplo de malla de una esfera creada en SOLIDWORKS..... | 18 |
| Fig. 2.4. Subdivisión del elemento de superficie donde, el punto rojo representa el centro del elemento y el punto azul corresponde al centro del subelemento..... | 19 |
| Fig. 2.5. Captura de Flow Simulation actuando sobre un objeto esfera..... | 21 |
| Fig. 2.6. Función de Coeficiente $C(v)$ resultante luego del proceso descrito anteriormente..... | 22 |
| Fig. 3.1. Acción del momento ante una perturbación en la orientación del vehículo..... | 24 |
| Fig. 3.2. Vistas laterales y frontales del diseño propuesto..... | 25 |
| Fig. 3.3. Diseño propuesto en vista isométrica..... | 27 |
| Fig. 3.4. Malla del diseño simplificado para obtener la matriz de masa agregada..... | 28 |
| Fig. 3.5. Fuerzas de amortiguamiento obtenidas para velocidades longitudinales y verticales.... | 29 |
| Fig. 3.6. Fuerzas de amortiguamiento obtenidas para las velocidades angulares..... | 29 |
| Fig. 3.7. Implementación del diseño y modelo del vehículo en Simulink..... | 30 |
| Fig. 3.8. Fuerza de propulsión a Voltaje nominal..... | 31 |
| Fig. 3.9. Fuerza del Propulsor en reversa a Voltaje Nominal..... | 31 |
| Fig. 3.10. Fuerza de propulsión para distintos valores de alimentación..... | 32 |
| Fig. 3.11. Fuerza de propulsión en reversa para distintos valores de alimentación..... | 32 |
| Fig. 4.1. Situación en la cual el vehículo gira cierta cantidad ψ | 33 |
| Fig. 4.2. Respuesta ante variación senoidal de ψ y Velocidad..... | 34 |
| Fig. 4.3. Respuesta ante variaciones de profundidad y θ | 35 |
| Fig. 5.1. Diagrama representando el esquema de localización..... | 37 |
| Fig. 5.2. Estructura típica de controlador difuso..... | 38 |
| Fig. 5.3. Diagrama de bloques de Control de navegación difuso..... | 40 |
| Fig. 5.4. Diagrama de bloques de sistema de control de navegación conectado con el modelo del vehículo..... | 40 |
| Fig. 5.5. Representación de las variables lingüísticas de la orientación del obstáculo..... | 41 |
| Fig. 5.6. Funciones de pertenencia para las entradas de la evasión de obstáculos..... | 42 |
| Fig. 5.7. Funciones de pertenencia para las entradas en el seguimiento de meta..... | 43 |
| Fig. 5.8. Situación en la que el vehículo se encuentra cercano a un obstáculo..... | 44 |

| | |
|--|----|
| Fig. 5.9. Funciones de pertenencia para la salida del comportamiento de evasión de Obstáculos..... | 45 |
| Fig. 5.10. Funciones de pertenencia para la salida del comportamiento seguimiento de meta..... | 46 |
| Fig. 5.11. Vista cenital de escenario diseñado..... | 50 |
| Fig. 5.12. Preparación de la simulación de navegación | 51 |
| Fig. 5.13. Trayectoria del vehículo hacia la meta..... | 51 |
| Fig. 6.1. Propulsor T200 de BlueRobotics..... | 52 |
| Fig. 6.2. Controlador de velocidad de propulsores..... | 53 |
| Fig. 6.3. Sonar Ping360..... | 53 |
| Fig. 6.4. Sensor de Presión BAR30..... | 54 |
| Fig. 6.5. Transponedores X150 y X010..... | 54 |
| Fig. 6.6. Camara Low-Light HD..... | 55 |
| Fig. 6.7. Controlador de navegación..... | 55 |
| Fig. 6.8. Microcontrolador Raspberry Pi 4..... | 56 |
| Fig. 6.9. Luces Led del vehículo..... | 56 |
| Fig. 6.10 Bateria de Ion Litio de 14.8[V]..... | 57 |
| Fig. A.1. Bloque de mediciones del vehículo..... | 63 |
| Fig. A.2. Bloque de derivación de Ángulos de Euler..... | 64 |
| Fig. A.3. Implementación de Fuerzas Hidrodinámicas..... | 64 |
| Fig. A.3. Implementación de Fuerzas dadas por la masa agregada..... | 65 |
| Fig. A.4. Implementación de Fuerzas Coriolis-Centrípetas..... | 65 |
| Fig. A.5. Implementación de Fuerzas de Amortiguamiento..... | 66 |
| Fig. A.6. Implementación de la Fuerza de empuje al vehículo..... | 66 |
| Fig. A.7. Diagrama Simulink del modelo CAD del vehículo..... | 67 |
| Fig. A.8. Bloque de Propulsores..... | 68 |
| Fig. A.9. Diagrama de Control para ψ y v | 69 |
| Fig. A.10. Diagrama de Control para θ y profundidad..... | 69 |
| Fig. A.11. Entrada a Controlador difuso encargado del seguimiento de meta..... | 70 |
| Fig. A.12. Conexión entre Comportamiento y Bloque de defusificación..... | 70 |
| Fig. A.13. Salida del Controlador de Navegación..... | 71 |

| | |
|--|----|
| Fig. A.14. Bloques utilizados para el enlace con Unreal Engine 4..... | 70 |
| Fig. C.1. Bloque Matlab Function para el procesamiento de datos..... | 80 |
| Fig. C.2. Representación del ambiente luego de procesamiento de datos..... | 81 |

Nomenclatura

Ejes de Referencia.

Ref_t : Sistema de referencia estático.

Ref_c : Sistema de referencia del vehículo.

Escalares.

X : Fuerza que se ejerce en el vehículo dado el movimiento por el eje x .

Y : Fuerza que se ejerce en el vehículo dado el movimiento por el eje y .

Z : Fuerza que se ejerce en el vehículo dado el movimiento por el eje z .

K : Momento ejercido en el vehículo dado por rotación en eje x (roll).

M : Momento ejercido en el vehículo dado por rotación en eje y (pitch).

N : Momento ejercido en el vehículo dado por rotación en eje z (yaw).

u : Velocidad del vehículo en el eje x de Ref_c .

v : Velocidad del vehículo en el eje y de Ref_c .

w : Velocidad del vehículo en el eje z de Ref_c .

p : Velocidad angular del vehículo por rotación en eje x de Ref_c .

q : Velocidad angular del vehículo por rotación en eje y de Ref_c .

r : Velocidad angular del vehículo por rotación en eje z de Ref_c .

x : Posición del vehículo en el eje x de Ref_c .

y : Posición del vehículo en el eje y de Ref_c .

z : Posición del vehículo en el eje z de Ref_c .

ϕ : Orientación del vehículo alrededor del eje x de Ref_c .

θ : Orientación del vehículo alrededor del eje y de Ref_c .

ψ : Orientación del vehículo alrededor del eje z de Ref_c .

f_e : Fuerza de empuje hacia el vehículo actuando sobre Ref_c .

f_g : Fuerza peso del vehículo actuando sobre Ref_c .

v : Variable de salida de velocidad del control de navegación difuso.

$\Delta\psi$: Variable de salida de cambio de orientación del control de navegación difuso.

τ : Grado de aplicabilidad del comportamiento en la navegación.

Matrices.

M :Matriz de inercia del vehículo.

C :Matriz de parámetros para Fuerzas Coriolis-Centripetas.

D :Matriz de parámetros para Fuerzas de Amortiguamiento.

$g(\eta)$:Vector de Fuerzas Restorantes del vehículo.

Abreviaciones.

AUV :Autonomous Underwater Vehicle.

FRIZQ :Frontal Izquierda.

| | | |
|------|------------------|------------------|
| FIZQ | Frente Izquierda | Fuerte Izquierda |
| FDER | Frente Derecha | Fuerte Derecha |

FRDER :Frontal Derecha.

DDER :Derecha-Derecha.

IIZQ :Izquierda-Izquierda.

N_g : Negativo Grande.

N_m : Negativo Medio.

N_p : Negativo Pequeño.

Z : Cero

P_p : Positivo Pequeño.

P_m : Positivo Medio.

P_g : Positivo Grande.

C : Cerca.

RE : Regular.

M : Medio.

L : Lejos.

V_{min} : Velocidad Mínima.

V_{med} : Velocidad Media.

V_{max} : Velocidad Máxima.

MIZQ : Medio Izquierda.

PIZQ : Poco Izquierda.

V_p : Velocidad pequeña.

1. Introducción

1.1. Introducción General.

El océano cubre alrededor del setenta y uno por ciento de la superficie de la tierra, manteniéndose un gran porcentaje desconocido para la humanidad. A través de los años ha aumentado la exploración y explotación submarina, por lo que el monitoreo y la investigación del océano resulta de vital importancia. Sin embargo, es difícil y peligroso para un humano operar en este ambiente dado que es inhóspito para la vida humana (altas presiones, ambientes irrespirables, entre otros) [16].

Una opción para este tipo de actividades son los vehículos submarinos no tripulados, ya que es una alternativa segura para la observación y operación del fondo marino.

Dentro de este tipo de vehículos existen aquellos que son operados por alguien en la superficie de manera remota denominados ROV, también están aquellos que son capaces de operar de manera autónoma, a los que se les denomina AUV los cuales varían en formas y tamaños según la funcionalidad y objetivo del diseño.

En el presente trabajo se propone el diseño de un AUV, así como también el diseño de su sistema de control de navegación.

1.2. Revisión bibliográfica

En esta sección se presenta la bibliografía revisada y utilizada para realizar esta investigación. Se revisaron aquellas cuyo contenido trata acerca del modelo del vehículo, instrumentación utilizada, tipos de navegación y movimiento autónomo.

- ❖ Thor I. Fossen, *Guidance of Control of Ocean Vehicles*. John Wiley & Sons, 1994 [1].

Texto en el que se trata el modelado y diseño de vehículos marinos, detalla el modelo y sus distintas formas dependiendo del tipo de vehículo, simplificaciones, presenta distintos tipos de esquemas de control y describe esquemas de movimiento automático.

- ❖ J.N. Newman., *Marine Hydrodynamics*. Cambridge, Massachusetts. The MIT Press, 2017 [2]

Texto en donde se tratan los principios físicos de la hidrodinámica de los vehículos marinos de manera conceptual e introduce distintos métodos numéricos para obtenerlos.

- ❖ J. Refsnes, A. J. Sorensen and K. Y. Pettersen, "A 6 DOF nonlinear observer for auvs with experimental results," *2007 Mediterranean Conference on Control & Automation*, Athens, Greece, 2007, pp. 1-7.[3]

Investigación sobre cómo se desarrolla un observador Luenberg para un AUV que busca mejorar la estabilidad del sistema respecto a perturbaciones del ambiente, una medida del ruido basada en el modelo del vehículo y generar un diseño de bajo peso y costo.

El principal aporte que genera a esta investigación es la descripción de los efectos desestabilizadores que tienen las Fuerzas Coriolis en el movimiento del vehículo y que están incluidas en el modelo presentado en [1].

- ❖ Tuveri, M., Ceruti, A., & Marzocca, P. "Added masses computation for unconventional airships and aerostats through geometric shape evaluation and meshing". *International Journal of Aeronautical and Space Sciences*, vol 15, no. 3, pp. 241-257, Sep. 2014 [4].

Estudio en donde se estima la matriz de masa agregada de vehículos aéreos experimentales mediante el uso del método de elementos frontera (BEM), describe de manera detallada el fundamento teórico detrás del procedimiento y la computación de estos valores. También entrega una lista de valores para distintos diseños de vehículos aéreos.

- ❖ Giovanni Carbone, Guillaume Martinat, Dominique Farcy, Jean-Luc Harion. "Added Masses of generic shape bodies interacting with external walls". *Aerospace Science and Technology*, vol. 90, pp. 70 – 84, Jul. 2019 [5].

Trabajo en donde se utiliza el método de elementos frontera (BEM) para la obtención de los elementos de la matriz de masa agregada. A diferencia de [4], el método de computación de los parámetros es distinto y presenta una forma de obtener la matriz cuando se tienen distintos cuerpos.

- ❖ Sulin Tang, Tamaki Ura, Takeshi Nakatani, Blair Thornton, Tao Jiang, “Estimation of the hydrodynamic coefficients of the complex-shaped autonomous underwater vehicle TUNA-SAND”, *Journal of Marine Science and Technology*, vol. 14, pp 373-386, Sep. 2009, doi: <https://doi.org/10.1007/s00773-009-0055-4> [6].

Estudio en donde se realiza la estimación de los coeficientes hidrodinámicos del submarino autónomo TUNA-SAND, la aproximación de los parámetros se realiza mediante simulaciones CFD con el programa ANSYS Fluent y luego son contrastados con datos experimentales.

Este trabajo resulta de interés pues demuestra el procedimiento que se utiliza en el momento de utilizar las simulaciones CFD. Sin embargo, se utiliza un programa avanzado en donde las funciones usadas son accedidas mediante el pago del servicio.

- ❖ Jyoti Prakash Panda, Arindam Mitra and Hari V Warrior, “A review on the hydrodynamic characteristics of autonomous underwater vehicles”. *Proceedings of the institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, vol. 14, no. 1, pp 15-29, Jul. 2020 [7].

Estudio en donde se revisan los parámetros hidrodinámicos de los vehículos submarinos, los efectos que tiene la forma del móvil en su hidrodinámica, también se describen algunas técnicas numéricas utilizadas para la medición de estos.

Dentro de este estudio resultó útil la descripción detallada de la ecuación de la fuerza drag, sus parámetros y como obtenerlos.

- ❖ Robert Christ, Robert Wernli, *The ROV Manual* 1st ed. England: Butterworth-Heinemann, 2007 [9].

Texto en donde se presenta el proceso de modelado y diseño de un ROV, un vehículo submarino no tripulado similar al AUV que es manejado por un operador, no posee un enfoque muy riguroso como [1]. Sin embargo, entrega información con respecto al tipo de materiales que se utilizan en este tipo de vehículos y decisiones de diseño que se deben tomar.

- ❖ Kai Sun, Weicheng Cui, Chi Chen, “Review of Underwater Sensing Technologies and Applications”, *Sensors*, vol. 21, no. 23, p. 7849, Nov. 2021, doi: 10.3390/s21237849 [10].

En [10] se listan algunas tecnologías de sensado ocupadas en diversos escenarios o misiones realizados por vehículos submarinos, también entrega información acerca de variables que se pueden medir o encontrar en el ambiente marino.

- ❖ Ridolfi *et al.*, "FeelHippo: A low-cost autonomous underwater vehicle for subsea monitoring and inspection," *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*, Florence, Italy, 2016, pp. 1-6, doi: 10.1109/EEEIC.2016.7555607 [11].

Texto en donde se describe el diseño del AUV FeelHippo, los instrumentos utilizados y su sistema de localización. Resulta de interés conocer la instrumentación utilizada, los materiales de la capsula que los contiene y el peso del vehículo.

- ❖ Juan A. Ramirez-Macías, Persijn Brongers, Santiago Rúa, Rafael E. Vásquez, “Hydrodynamic modelling for the remotely operated vehicle Viseor3 using CFD”, *IFAC-PapersOnLine*, vol. 49, no. 23, pp 187-192, Oct. 2016 [12].

Estudio en donde se describe el proceso de obtención de los parámetros hidrodinámicos de un ROV mediante el uso de técnicas CFD, resulta de interés debido a las simplificaciones que se realizaron en el modelo del diseño al obtener los parámetros.

- ❖ M. B. Alatise and G. P. Hancke, "A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods," in *_IEEE Access_*, vol. 8, pp. 39830-39846, 2020, doi: 10.1109/ACCESS.2020.2975643 [13].

Estudio en donde se presentan distintos tipos de navegación y localización para vehículos autónomos no submarinos, así como distintos métodos de fusión de datos entregados por los sensores para estimar la pose o mejorar las mediciones realizadas.

- ❖ Bo He, Yan Liang, Xiao Feng, Rui Nian, Tianhong Yan, Minghui Li, Shujing Zhang, “AUV SLAM and Experiments Using a Mechanical Scanning Forward-Looking Sonar”, *Sensors*, vol. 12, no. 7, pp 9386-9410, Jul. 2012, doi: 10.3390/s120709386 [14].

Investigación en donde se implementa un algoritmo SLAM mediante el uso de un sonar, este es una modificación del método fastSLAM que es un método capaz de realizar esta actividad con un número mayor de puntos de referencia.

- ❖ J. Jung, Y. Lee, D. Kim, D. Lee, H. Myung and H. -T. Choi, "AUV SLAM using forward/downward looking cameras and artificial landmarks," *2017 IEEE Underwater Technology (UT)*, Busan, Korea (South), 2017, pp. 1-3, doi: 10.1109/UT.2017.7890307. [15].

Estudio en donde se utiliza un sistema de navegación SLAM en una cámara mediante el reconocimiento de distintos puntos de referencias en una piscina olímpica, proporciona una idea acerca del funcionamiento de este y la efectividad de este tipo de algoritmos en la navegación del vehículo.

- ❖ L. Paull, S. Saeedi, M. Seto and H. Li, "AUV Navigation and Localization: A Review," in *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 131-149, Jan. 2014, doi: 10.1109/JOE.2013.2278891 [16].

Investigación en donde se describen distintos tipos de sistemas de navegación utilizados en vehículos submarinos, así como también de localización, también, junto a esto se mencionan los sensores usados para lograr dichas estrategias.

- ❖ Saffiotti A., “The uses of fuzzy logic in autonomous robot navigation”, *Soft Computing* , vol 1., pp 180-197, Dec. 1997, doi:<https://doi.org/10.1007/s005000050020> [17]

Estudio en donde se describe la utilidad de la lógica difusa en los vehículos de navegación autónoma, describe diversos sistemas de navegación que la usan dividiendo las acciones del robot

en comportamientos y finalmente propone una forma de integración de estos comportamientos llamado CDB (context dependent blending).

- ❖ Edward Tunstel Jr., Tanya Lippincott, Mo Jamshidi, "Behavior Hierarchy for Autonomous Mobile Robots: Fuzzy-Behavior Modulation and Evolution", in *Intelligent Automation & Soft Computing*, vol. 3, no. 1, pp 37-49, doi: 10.1080/10798587.1997.10750691 [18].

Investigación en donde se propone un sistema de navegación difuso basado en comportamientos, propone el uso de ganancias que varían según la situación en donde se encuentre el robot para la coordinación de conductas, también propone el uso de programación genética para la evolución de las reglas del algoritmo de navegación.

- ❖ H. Seraji and A. Howard, "Behavior-based robot navigation on challenging terrain: A fuzzy logic approach," in *IEEE Transactions on Robotics and Automation*, vol. 18, no. 3, pp. 308-321, June 2002, doi: 10.1109/TRA.2002.1019461 [19].

Estudio en donde se utiliza un algoritmo de navegación difusa basado en comportamientos en terrenos difíciles para un robot terrestre, divide el actuar del robot en tres comportamientos: *Seguimiento de meta*, *Evasión de obstáculos* y *Atravesar terreno*, el sistema de control de navegación propuesto utiliza la coordinación de comportamientos propuesto en [18] y CDB propuesto por [17].

2. Modelo.

2.1. Cinemática y ejes de referencia

Los vehículos submarinos a diferencia de aquellos que trabajan en la superficie, se encuentran inmersos en un volumen con una densidad lo suficientemente grande como para que la acción del ambiente hacia el móvil se deba tomar en cuenta en todas las componentes de su movimiento, por ende, es necesario definir un modelo que represente las Fuerzas, Momentos, Velocidades Lineales, Velocidades Angulares, Posición y Orientación en sus 6 grados de libertad.

Como se sabe, el submarino es un vehículo que trabaja dentro de un volumen de agua, por lo que las variables mencionadas anteriormente se representan en literatura según la notación SNAME, sigla de la sociedad de arquitectos navales e ingenieros marinos.

Para poder describir la orientación, posición, velocidades y fuerzas en [1] se indica que es necesario definir dos referencias, una que se mueve con el vehículo que se denomina eje de referencia del cuerpo o Ref_c con origen O_c en el centro de gravedad (CG) del vehículo y otra denominada Ref_t que es estático y se ubica en algún punto de la superficie de la tierra.

Para representar el movimiento del vehículo se utilizan los vectores

$$\eta = [\eta_1^T, \eta_2^T]^T \quad \eta_1 = [x, y, z]^T \quad \eta_2 = [\phi, \theta, \psi]^T \quad (2.1)$$

$$v = [v_1^T, v_2^T]^T \quad v_1 = [u, v, w]^T \quad v_2 = [p, q, r]^T \quad (2.2)$$

$$\tau = [\tau_1^T, \tau_2^T]^T \quad \tau_1 = [X, Y, Z]^T \quad \tau_2 = [K, M, N]^T \quad (2.3)$$

En donde η es el vector que representa la posición y orientación del vehículo representadas con respecto a Ref_t donde, η_1 es la posición y η_2 la orientación, v el de velocidades lineales y angulares con respecto a Ref_c donde, v_1 es el vector de velocidad lineal y v_2 el de velocidad angular, τ representa las fuerzas y momentos actuando sobre el vehículo en Ref_c con τ_1 las fuerzas sobre los ejes x, y, z y τ_2 los momentos angulares.

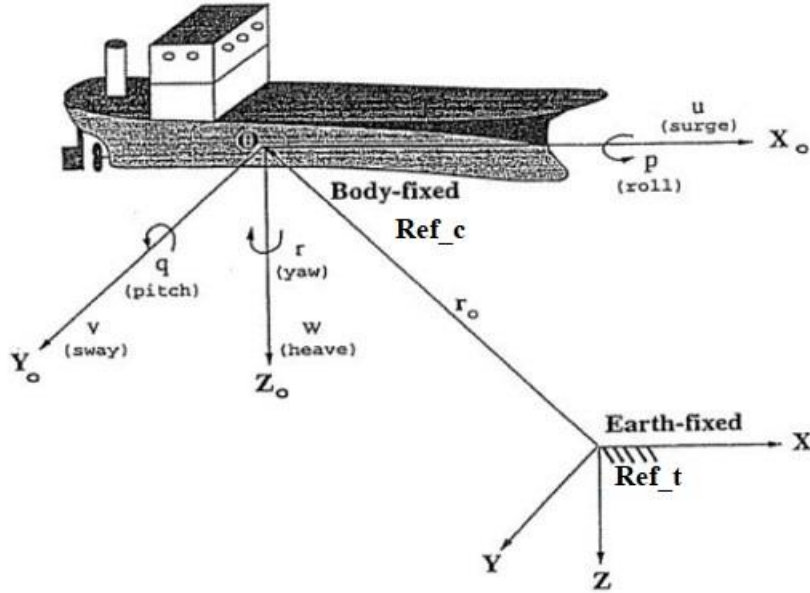


Fig. 2.1 Imagen ilustrativa de ejes de referencia Ref_c y Ref_t [1].

La velocidad del vehículo con respecto a Ref_t se puede obtener mediante la transformación:

$$\dot{\eta} = J(\eta)v \quad (2.4)$$

En donde $J(\eta)$ se describe como,

$$J(\eta) = \begin{bmatrix} J_1(\eta_2) & 0_{3 \times 3} \\ 0_{3 \times 3} & J_2(\eta_2) \end{bmatrix} \quad (2.5)$$

Como se puede ver en (2.5) la transformada se calcula con respecto a la orientación del vehículo expresada en ángulos de Euler $\eta_2 = [\phi, \theta, \psi]^T$, donde $J_1(\eta_2)$ corresponde a la transformación de las velocidades lineales v_1 y $J_2(\eta_2)$ a la transformación con respecto a las velocidades angulares v_2 .

Por ende, si se extiende (2.4) se obtiene,

$$\begin{bmatrix} \dot{\eta}_1 \\ \dot{\eta}_2 \end{bmatrix} = \begin{bmatrix} J_1(\eta_2) & 0_{3 \times 3} \\ 0_{3 \times 3} & J_2(\eta_2) \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (2.6)$$

Expandiendo en $J_1(\eta_2)$ y $J_2(\eta_2)$, la primera describe la transformación de Ref_t a Ref_c , inicialmente trasladando Ref_t hacia Ref_c y luego rotando este con respecto a sus ejes z , y , x . Por lo que la expresión $J_1(\eta_2)$ es:

$$J_1(\eta_2) = C_{z,\psi}^T C_{y,\theta}^T C_{x,\psi}^T = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\psi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & -c\psi s\phi + s\theta s\psi c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (2.7)$$

Con $C_{z,\psi}^T$, $C_{y,\theta}^T$, $C_{x,\phi}^T$ las operaciones de rotación explicadas anteriormente y $c\psi$, $s\psi$ formas abreviadas de escribir $\cos(\psi)$ y $\sin(\psi)$, respectivamente.

Para $J_2(\eta_2)$ se tiene,

$$J_2(\eta_2) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \quad (2.8)$$

2.2. Modelo del vehículo.

El modelo matemático del sistema describe la dinámica del vehículo submarino y representa las interacciones entre este y el cuerpo de agua en el que está inmerso. Es decir, dentro de este se consideran las fuerzas hidrodinámicas e hidrostáticas, que actúan sobre el móvil al desplazarse en el ambiente marino con sus 6 grados de libertad (desplazamientos lineales y rotaciones). Esto es,

$$M\dot{v} + C(v)v + D(v)v + g(\eta) = \tau \quad (2.9)$$

$$\dot{\eta} = J(\eta)v \quad (2.10)$$

Donde v representa el vector de velocidades lineales y angulares, η la posición y orientación del vehículo y τ el vector de fuerzas de propulsión que actúan en el móvil. Las matrices M , $C(v)$, $D(v)$ son constituidas por los parámetros hidrodinámicos e hidrostáticos del sistema, $g(\eta)$ es el vector de fuerzas que contiene la acción del empuje y peso del objeto, por último, $J(\eta)$ es una matriz transformada que permite la descripción de la velocidad del submarino en el eje de referencia Ref_t . Vale decir, también que las matrices M y C se pueden expresar como una suma de sus *partes hidrostáticas e hidrodinámicas* de esta forma,

$$M = M_{RB} + M_a \quad (2.11)$$

$$C(v) = C_{RB}(v) + C_a(v) \quad (2.12)$$

Cada uno de los productos de las matrices mencionadas con el vector de velocidad representados en $M\dot{v}$, $C(v)v$, $D(v)v$ así como también $g(\eta)$ representan la sumatoria de fuerzas que actúan sobre el vehículo al moverse dentro del cuerpo de agua, resulta conveniente entonces, explicar los fenómenos físicos que estos representan, así como también, los parámetros que se hallan en las matrices.

Dentro de los parámetros descritos en el modelo se pueden diferenciar dos tipos: *parámetros hidrostáticos* y *parámetros hidrodinámicos*, los primeros corresponden a parámetros del cuerpo rígido, los cuales comprenden principalmente de M_{RB} que representa a los parámetros en seco del vehículo, se define como la matriz,

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & mz_g & -my_g \\ 0 & m & 0 & -mz_g & 0 & mx_g \\ 0 & 0 & m & my_g & -mx_g & 0 \\ 0 & -mz_g & my_g & I_x & -I_{xy} & -I_{xz} \\ mz_g & 0 & -mx_g & -I_{yx} & I_y & -I_{yz} \\ -m_{yg} & m_{xg} & 0 & -I_{zx} & -I_{zy} & I_z \end{bmatrix} \quad (2.13)$$

Con m la masa en seco del vehículo, $CG = [x_g, y_g, z_g]^T$ el centro de masa del vehículo e $I_x, I_y, I_z, I_{xy}, \dots, I_{zy}$ elementos de la matriz de inercia rotacional respecto al centro del vehículo. En [1] también se cuenta con C_{RB} , pero esto debido a que $C(v)$ es una parametrización de la matriz M . Es decir, en este caso $M = M_{RB}$, sin embargo, si M cambia (como en el caso del modelo completo pues, $M = M_{RB} + M_a$), también lo hace C . Por ende, resulta conveniente definir $C(v)$ una vez se tienen los parámetros hidrodinámicos e hidrostáticos de M , donde, alguno de estos se puede obtener gracias a plataformas CAD como es el caso de SolidWorks o mediante el uso de programas especializados en el diseño de vehículos marinos.

Con respecto a los *parámetros hidrodinámicos*, estos comprenden aquellos contenidos por las matrices $M_a, D(v)$ y $C(v)$ los cuales son obtenidos mediante pruebas experimentales con un modelo físico del vehículo, lo cual requiere disponibilidad de equipamiento. Para solucionar esta problemática se utilizarán simplificaciones descritas en [1] y programas que estimen los parámetros como es el caso de CFD.

La primera simplificación que se realiza es con la masa agregada M_a , la cual se considera constante. Generalmente se define M_a como dependiente de la frecuencia de amortiguamiento ω como se menciona en [2] y [1], entonces, la matriz de masa agregada M_a queda como,

$$M_a = - \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ X_{\dot{v}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ X_{\dot{w}} & Y_{\dot{w}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ X_{\dot{p}} & Y_{\dot{p}} & Z_{\dot{p}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ X_{\dot{q}} & Y_{\dot{q}} & Z_{\dot{q}} & K_{\dot{q}} & M_{\dot{q}} & M_{\dot{r}} \\ X_{\dot{r}} & Y_{\dot{r}} & Z_{\dot{r}} & K_{\dot{r}} & M_{\dot{r}} & N_{\dot{r}} \end{bmatrix} \quad (2.14)$$

La masa agregada es un fenómeno físico que se produce al generarse un movimiento acelerado de un cuerpo en un fluido, se define como el trabajo adicional que debe realizar el fluido ante la aceleración de un móvil dentro de él, esto se traduce como una inercia adicional. Vale decir, que la masa agregada es proporcional a la densidad del fluido por lo que resulta importante mantenerlo dentro del modelo y obtener una buena aproximación de él.

Vale decir, que la matriz de inercia definida en (2.14) depende de la simetría del vehículo, pudiendo así, ser simplificada de acuerdo con los planos de simetría que tenga el submarino. Por ejemplo, si el plano de simetría del móvil es el plano xz se tiene que la matriz de inercia puede ser escrita como,

$$M = \begin{bmatrix} m_{11} & 0 & m_{13} & 0 & m_{15} & 0 \\ 0 & m_{22} & 0 & m_{24} & 0 & m_{26} \\ m_{31} & 0 & m_{33} & 0 & m_{35} & 0 \\ 0 & m_{42} & 0 & m_{44} & 0 & m_{46} \\ m_{51} & 0 & m_{53} & 0 & m_{55} & 0 \\ 0 & m_{62} & 0 & m_{64} & 0 & m_{66} \end{bmatrix} \quad (2.15)$$

Otra simplificación será con respecto a la matriz de amortiguamiento $D(v)$, se considera a $D(v)$ como una matriz diagonal cuadrada de dimensión 6, esto lo menciona [1] como una reducción válida para vehículos submarinos, esta es.

$$D(v) = \begin{bmatrix} X_u & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_v & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_w & 0 & 0 & 0 \\ 0 & 0 & 0 & K_p & 0 & 0 \\ 0 & 0 & 0 & 0 & M_q & 0 \\ 0 & 0 & 0 & 0 & 0 & N_r \end{bmatrix} \quad (2.16)$$

Como se mencionó antes la matriz Coriolis-Centripeta C es una parametrización de M y esta se define como,

$$C(v) = \begin{bmatrix} 0_{3 \times 3} & -S(M_{11}v_1 + M_{12}v_2) \\ -S(M_{11}v_1 + M_{12}v_2) & -S(M_{21}v_1 + M_{22}v_2) \end{bmatrix} \quad (2.17)$$

Con $M_{11}, M_{12}, M_{21}, M_{22}$ submatrices de 3×3 de la matriz de inercias M . Es decir,

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \quad (2.18)$$

Donde se define a S como una transformada definida como,

$$S(\lambda) = \begin{bmatrix} 0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & 0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & 0 \end{bmatrix} \quad (2.19)$$

Que genera una matriz antisimétrica del vector $\lambda = [\lambda_1, \lambda_2, \lambda_3]^T$.

Cabe destacar que en la matriz de fuerzas Coriolis-Centrípeta también se encuentra presente el momento de Munk que es una fuerza desestabilizadora que tiende a girar al vehículo en una dirección perpendicular a su movimiento [3].

Por último $g(\eta)$ comprende las fuerzas de empuje y gravitacionales que actúan sobre el centro de gravedad del sistema. Esta se define como,

$$g(\eta) = \begin{bmatrix} (W - B)s\theta \\ -(W - B)c\theta s\phi \\ -(W - B)c\theta c\phi \\ -(y_g W - y_B B)c\theta c\phi + (z_g W - z_B B)c\theta s\phi \\ (z_g W - z_B B)s\theta + (x_g W - x_B B)c\theta c\phi \\ -(x_g W - x_B B)c\theta s\phi - (y_g W - y_B B)s\theta \end{bmatrix} \quad (2.20)$$

Con W y B las magnitudes de las fuerzas, $CG = [x_g, y_g, z_g]$ y $CB = [x_B, y_B, z_B]$ centros de gravedad y empuje, respectivamente. En [1] se denomina el vector de fuerzas restaurantes esto debido a que modela el efecto estabilizador natural del vehículo dada la magnitud del brazo entre el centro de gravedad (CG) y el de empuje (CB).

2.3. Modelo de Propulsores.

Para poder desplazar el vehículo se hará uso de propulsores que constan de una hélice y un motor. La fuerza de propulsión se describe mediante la ecuación.

$$T = \rho D^4 K_T(J_0) |n|n \quad (2.21)$$

Con ρ la densidad del agua, D el diámetro de la hélice, n las revoluciones de la hélice, J_0 el coeficiente de avance y K_T el coeficiente de propulsión.

La relación de $K_T(J_0)$ es una curva no lineal de cuatro cuadrantes. Sin embargo, dado que los propulsores ocupados en este tipo de vehículos son bidireccionales, el comportamiento de K_T se puede expresar como un comportamiento lineal de la forma,

$$K_T = \alpha_1 + \alpha_2 \frac{V_a}{nD} \quad (2.22)$$

Por lo cual si reemplazamos (22) en (23), la fuerza de propulsión queda como,

$$T(n, V_a) = T_{|n|n} |n|n + T_{|n|V_a} |n|V_a \quad (2.23)$$

De aquí, la velocidad de avance V_a se puede relacionar con la velocidad del vehículo mediante el uso de,

$$V_a = (1 - \omega)V \quad (2.24)$$

Con ω el numero wake fraction, de aquí se puede expresar a la fuerza realizada por un propulsor como,

$$\tau = b_1 |n|n - b_2 |n|v \quad (2.25)$$

Con $b_1 = T_{|n|n} > 0$, $b_2 = -T_{|n|V_a}(1 - \omega)$. Según [1], el modelo de arriba se puede aproximar mediante el modelo afín

$$\tau = Bu \quad (2.26)$$

Siendo $B = B_1$, y B es una extensión multivariable del coeficiente b_1 .

2.4. Obtención de Parámetros Hidrodinámicos.

En la siguiente sección se describen los métodos utilizados para calcular el valor de los parámetros hidrodinámicos. Para la masa agregada se utilizó el método descrito en [4] y [5], la razón de esto es debido a las limitaciones de los programas utilizados, los parámetros de la matriz de amortiguamiento $D(v)$ se calcularon mediante el uso de una función del programa SOLIDWORKS.

2.4.1. Masa agregada

Como se mencionaba en la sección 2.2, la masa agregada es el trabajo adicional que realiza el fluido al generarse un movimiento acelerado dentro de él. La cual se traduce como una inercia agregada al sistema o un volumen de agua que se mueve con el vehículo modelada como una fuerza que se opone a su movimiento, así si el móvil acelera una cantidad \vec{a} el efecto de la masa agregada viene dada por,

$$f_{ma} = -M_a \vec{a} \quad (2.27)$$

Con M_a la matriz de masa agregada como la expresada en (2.14). Por lo cual, para poder estimar esta fuerza es necesario encontrar los valores de M_a correspondientes al diseño que se tiene.

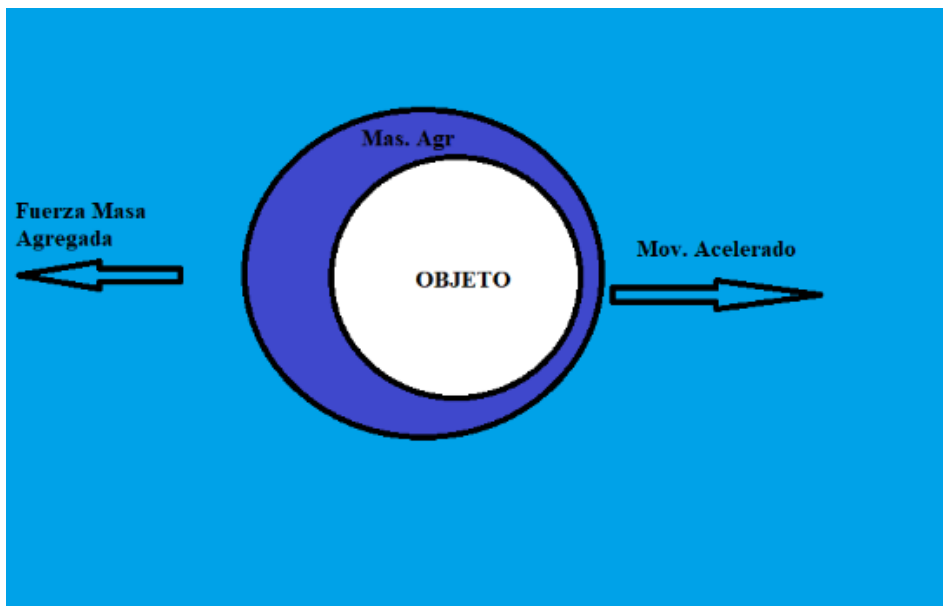


Fig. 2.2 Imagen ilustrativa del efecto de la masa agregada sobre un objeto inserto en un fluido en movimiento acelerado.

Para esto, se utiliza el método descrito en [4] y [5] en donde los parámetros de dicha matriz se

calculan mediante,

$$\lambda_{ij} = -\rho \iint_S \frac{\partial \phi_i}{\partial \hat{n}} \phi_j dS \quad (2.28)$$

Con ρ la densidad del agua o del fluido en donde esté inmerso el objeto y $\phi_{i,j}$ el potencial del fluido en distintos elementos de la superficie (i o j), el cual es obtenido mediante la solución de la ecuación de Laplace,

$$\nabla^2 \phi = 0 \quad (2.29)$$

Con condiciones frontera:

1.-

$$\left. \frac{\partial \phi}{\partial \hat{n}} \right|_S = u_n \quad (2.30)$$

Donde \hat{n} es el vector normal a la superficie S y u_n es la proyección del vector de velocidad de un punto sobre el vector \hat{n} .

2.-

$$\lim_{r \rightarrow \infty} \frac{\partial \phi}{\partial r} = 0 \quad (2.31)$$

Con r la distancia del origen a algún punto del fluido.

De aquí, hay que agregar que el uso de la ecuación (2.29) conlleva la suposición que el fluido en el cual se encuentra inmerso el objeto es incompresible e irrotacional.

Luego, para obtener una solución a (2.29) dentro del desarrollo de [4] y [5] se menciona que se puede realizar una relación entre u_n y la velocidad del centro del objeto u_0 , con esto se llega a:

$$\left. \frac{\partial \phi}{\partial \hat{n}} \right|_S = u_n = u_{0x}\alpha + u_{0y}\beta + u_{0z}\gamma + \omega_x(\gamma y - \beta z) + \omega_y(\alpha z - \gamma x) + \omega_z(\beta x - \alpha y) \quad (2.33)$$

Con $\alpha = \cos(n, x)$, $\beta = \cos(n, y)$, $\gamma = \cos(n, z)$, en donde $\cos(n, \cdot)$ representa a los cosenos direccionales entre los vectores normales a la superficie del objeto n y los ejes de referencia x, y o z también, u_0 es la velocidad del centro del objeto inmerso en el fluido con componentes

(u_{0x}, u_{0y}, u_{0z}) .

A partir de aquí para poder llegar a la solución se debe asumir la linealidad del potencial del fluido, pudiendo así definirlo como una combinación lineal de sus seis componentes,

$$\phi = u_{0x}\phi_1 + u_{0y}\phi_2 + u_{0z}\phi_3 + \omega_x\phi_4 + \omega_y\phi_5 + \omega_z\phi_6 \quad (2.34)$$

En donde u_{0x}, u_{0y}, u_{0z} corresponde a la velocidad lineal del cuerpo y $\omega_x, \omega_y, \omega_z$ las velocidades angulares del mismo.

En consecuencia, [5] argumenta que dado a que cada elemento de potencial ϕ_i es una solución a la ecuación de Laplace (2.29) es posible deducir que las condiciones fronteras para cada ϕ_i viene dado por,

$$\left. \frac{\partial \phi_i}{\partial \hat{n}} \right|_S = \begin{cases} \alpha & , i = 1 \\ \beta & , i = 2 \\ \gamma & , i = 3 \\ \gamma y - \beta z & , i = 4 \\ \alpha z - \gamma x & , i = 5 \\ \beta x - \alpha y & , i = 6 \end{cases} \quad (2.35)$$

Donde, los elementos representados en $i = 1, 2, 3, \dots, 6$ son aquellos dados en la ecuación (2.33), si se observa (2.28) se puede notar que la única variable faltante para obtener los parámetros de M_a es ϕ_j . Este se calcula mediante la solución de la ecuación de Laplace para el elemento dicho, para esto se ocupa la ecuación,

$$2\pi\phi(P) = \iint_S \phi(Q) \frac{\partial}{\partial \hat{n}} \left\{ \frac{1}{\|\vec{r}\|} \right\} dS - \iint_S \frac{\partial \phi(Q)}{\partial \hat{n}} \frac{1}{\|\vec{r}\|} dS \quad (2.36)$$

Donde $\phi(P)$ y $\phi(Q)$ corresponden al valor del potencial en dos puntos de distintos elementos de superficie del objeto y \vec{r} la distancia entre estos dos puntos. Para poder llegar a un valor numérico (2.36) se debe discretizar, lo que lleva a la expresión:

$$\phi_i = \sum_{j=1}^{N_b} \phi_j \frac{C_{ij}}{2\pi} - \sum_{j=1}^{N_b} \frac{\partial \phi_j}{\partial \vec{n}} \frac{B_{ij}}{2\pi} \quad (2.37)$$

Con,

$$C_{ij} = \frac{\partial}{\partial \vec{n}_j} \left(\frac{1}{\|\vec{r}_{ij}\|} \right) \partial S_j \quad (2.38)$$

$$B_{ij} = \frac{1}{\|\vec{r}_{ij}\|} \partial S_j \quad (2.39)$$

Donde, las variables con el subíndice i o j corresponden a aquellas en un punto de distintos elementos de superficie y N_b es la cantidad de elementos de superficie que subdividen la superficie total del objeto.

La ecuación (2.37) se puede expresar de forma matricial de la siguiente forma,

$$[2\pi I - C]\phi = -B \frac{\partial \phi}{\partial \vec{n}} \quad (2.40)$$

Siendo I la matriz identidad de $N_b \times N_b$, C la matriz de coeficientes C_{ij} dados por (2.38), B la matriz de coeficientes B_{ij} dados por (2.39) y ϕ el vector de valores del potencial.

Una vez se tiene esta expresión, el valor de ϕ se puede calcular mediante,

$$\phi = -[2\pi I - C]^{-1} B \frac{\partial \phi}{\partial \vec{n}} \quad (2.41)$$

reemplazando con $i = 1, 2, 3, \dots, 6$ se puede expresar como

$$\phi_i = -[2\pi I - C]^{-1} B \frac{\partial \phi_i}{\partial \vec{n}} \quad (2.42)$$

Finalmente, la matriz de masa agregada queda como,

$$M_a = -\rho \begin{bmatrix} \sum_{i=1}^{N_b} (\phi_1)_i \left(\frac{\partial \phi_1}{\partial \vec{n}} \right)_i dS_i & \cdots & \sum_{i=1}^{N_b} (\phi_1)_i \left(\frac{\partial \phi_6}{\partial \vec{n}} \right)_i dS_i \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{N_b} (\phi_6)_i \left(\frac{\partial \phi_1}{\partial \vec{n}} \right)_i dS_i & \cdots & \sum_{i=1}^{N_b} (\phi_6)_i \left(\frac{\partial \phi_6}{\partial \vec{n}} \right)_i dS_i \end{bmatrix} \quad (2.43)$$

A continuación, se describe el proceso de obtención de estos valores:

1. Mediante el uso de una plataforma CAD como es el caso de SOLIDWORKS se exporta la malla del objeto diseñado, esta consta de una teselación de triángulos que cubren a la superficie del objeto. Por ende, el elemento de superficie sería un triángulo cuya magnitud de superficie es indicada por el archivo.

Este tipo de archivo es un archivo de extensión .stl que consta de una lista de vértices de los triángulos y su vector normal.

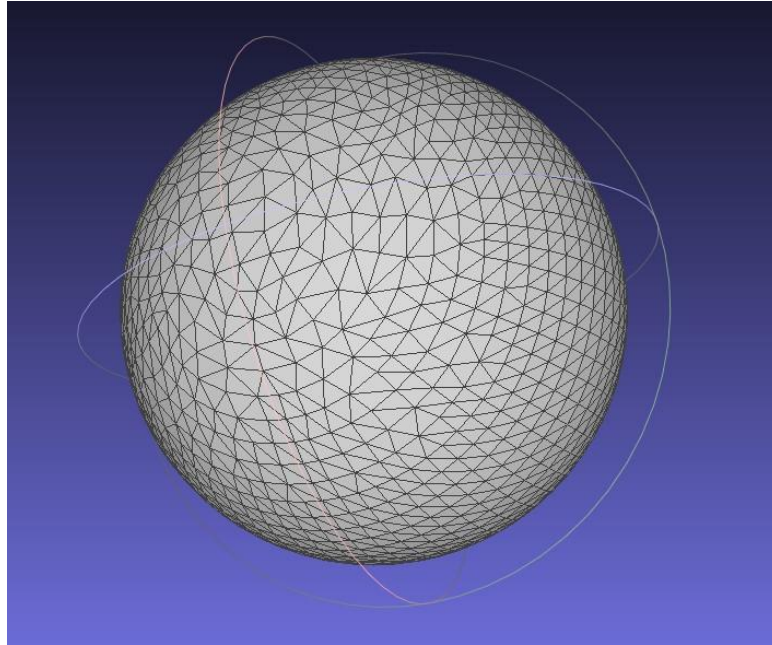


Fig. 2.3: Ejemplo de malla de una esfera creada en SOLIDWORKS.

2. Una vez se tiene el objeto, se proceden a calcular las matrices de coeficientes C y B , para esto se deben utilizar las ecuaciones (2.38) y (2.39), de aquí se observan dos métodos: uno mediante la integración numérica de cada elemento de superficie que se menciona en [4] y otra mediante la suma de subdivisiones de cada elemento de superficie, que se menciona en [5]. En este trabajo se utilizó el último método que consiste en tomar el elemento de superficie y su centro, subdividirlo en k subelementos con el centro como vértice. Con esta subdivisión la ecuación (2.37) debe quedar como,

$$2\pi\phi_i = \sum_{j=1}^{N_b} \phi_j \sum_{k=1}^{N_{sub}} C_{ijk} - \sum_{j=1}^{N_b} \frac{\partial\phi_j}{\partial\vec{n}} \sum_{k=1}^{N_{sub}} B_{ijk} \quad (2.44)$$

Con,

$$C_{ijk} = -\frac{\overrightarrow{r_{ijk}} \cdot \vec{n}_j}{\|\overrightarrow{r_{ijk}}\|^3} \delta S_{jk} \quad (2.45)$$

$$B_{ijk} = \frac{1}{\|\overrightarrow{r_{ijk}}\|} \delta S_{jk} \quad (2.46)$$

Donde, $\overrightarrow{r_{ijk}}$ es el vector distancia entre el centro del elemento y el centro del subelemento, \vec{n}_j el vector normal al elemento de superficie y δS_{jk} el área del subelemento.

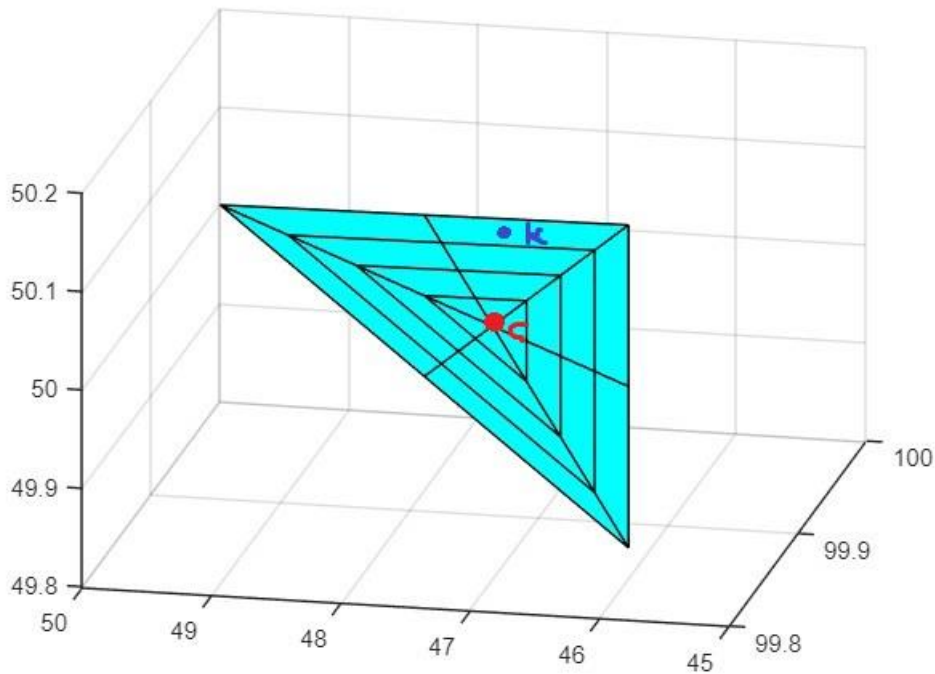


Fig. 2.4. Subdivisión del elemento de superficie donde, el punto rojo representa el centro del elemento y el punto azul corresponde al centro del subelemento.

Vale decir también, que las matrices obtenidas son de dimensión $N_b \times N_b$, con N_b el número de elementos de superficie que tiene el diseño expresado en el punto 1.

- Una vez son obtenidas las matrices de coeficientes C y B se puede resolver la ecuación (2.42) para ϕ_i con $i = 1, 2, 3, \dots, 6$ con sus correspondientes valores frontera expresados en (2.35). Por ejemplo, si se quieren obtener los valores para ϕ_1 la ecuación (2.42) queda como,

$$\phi_1 = -[2\pi I - C]^{-1} B\alpha$$

Con, $\frac{\partial \phi_i}{\partial \vec{n}} = \alpha$ para $i = 1$ según (2.35) y α corresponde al vector de cosenos direccionales respecto al eje x para cada elemento de superficie.

4. Por último, se calcula el coeficiente de la matriz M_a expresada en (2.43).

El código creado y utilizado para el método descrito anteriormente se puede ver en *Anexo B*

2.4.2. Fuerzas de amortiguamiento

En los vehículos marinos el amortiguamiento puede ser causado por radiación $D_p(v)$, fricción entre el cuerpo del vehículo y el agua para el flujo laminar o turbulento $D_s(v)$, por la interacción entre las olas y el vehículo $D_w(v)$ o por la generación de vórtices dada la interacción del móvil y el fluido $D_M(v)$ [1].

Cada una de estas interacciones forman parte de la matriz de amortiguamiento $D(v)$ tal que,

$$D(v) = D_p(v) + D_s(v) + D_w(v) + D_M(v) \quad (2.47)$$

Dado a que el modelo del vehículo es un modelo de seis grados de libertad, la matriz de amortiguamiento es una matriz cuadrada de dimensión seis. Sin embargo, [1] menciona que se puede asumir una estructura diagonal para $D(v)$ como la expresada en (2.16), cada elemento de la matriz multiplicado por la velocidad V correspondiente describe la fuerza de amortiguamiento F que es descrita por.

$$F = -\frac{1}{2} \rho A_p C_i(v) V^2 \quad (2.48)$$

Con ρ la densidad del agua, A el área proyectada frontal o lateral del vehículo u objeto sumergido en el agua, C_i el coeficiente de drag y V la velocidad del vehículo. Como se puede ver en (2.48) para obtener el valor de F es necesario encontrar el $C_i(v)$ correspondiente, para esto se recurre al uso de CFD.

CFD o Computational Fluid Dynamics, corresponde al proceso en el cual se predice el flujo de un fluido de manera computacional resolviendo las ecuaciones que lo gobiernan, estas corresponden a las tres leyes de conservación: Conservación de la Masa, Conservación del Momento (ecuación de

Navier-Stokes) y Conservación de la energía (Primera ley de la termodinámica). La solución a estas ecuaciones se realiza discretizando el dominio en donde se va a trabajar en elementos de superficie los cuales pertenecen a la malla como la mostrada en la Figura 2.3.

Para la estimación de las fuerzas de amortiguamiento primero se necesita obtener una estimación del parámetro $C_i(v)$ para esto, primero se reordena (2.48) obteniendo,

$$C_i(v) = \frac{2F}{\rho A_p V^2} \quad (2.49)$$

De aquí es necesario estimar la fuerza F que se ejerce sobre el vehículo moviéndose a una velocidad V en cada eje del sistema de referencia y desplazamiento angular, para esto se utiliza la función CFD Flow Simulation de SOLIDWORKS en donde se simula un flujo de agua moviéndose en la dirección contraria a la que se mueve el vehículo con la misma velocidad V y se estima la fuerza F que este genera sobre el cuerpo, una vez se tiene F se reemplazan los valores en (2.49) y se obtiene el valor de C_i .

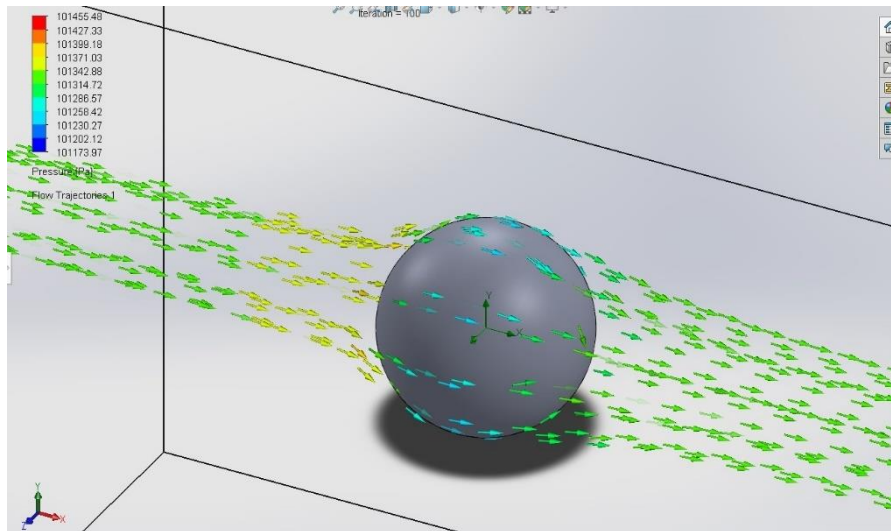


Fig. 2.5. Captura de Flow Simulation actuando sobre un objeto esfera

Como se puede ver el valor de $C_i(v)$ no es un valor constante, por ende, es necesario encontrar una función que describa el cambio de C_i con respecto a la velocidad para esto, se define un rango de velocidad R_v en donde el vehículo o el objeto opere, luego se eligen n valores dentro de R_v para después proceder a estimar F para cada uno de estos.

Por último, para estimar la curva se realiza una interpolación polinomial con los datos obtenidos.

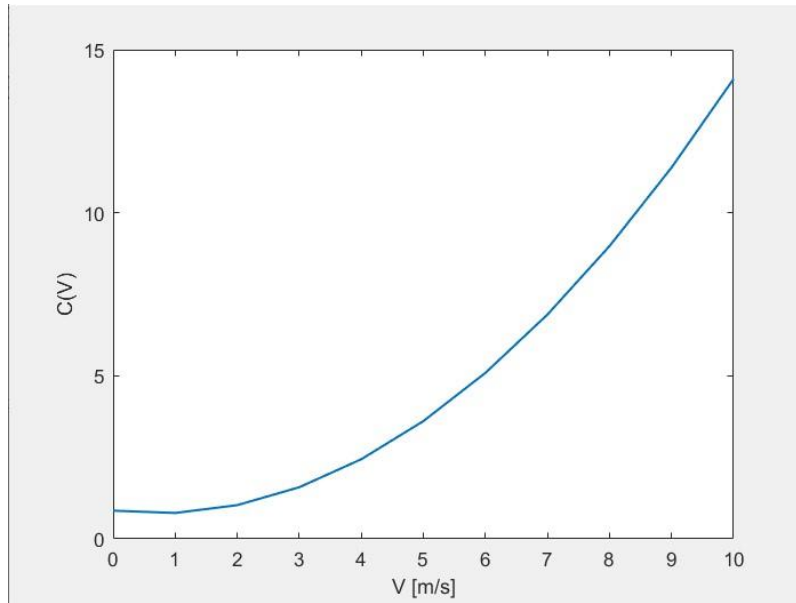


Fig. 2.6. Función de Coeficiente $C(v)$ resultante luego del proceso descrito anteriormente.

3. Diseño del vehículo.

3.1. Introducción.

En el siguiente capítulo se presenta el diseño del vehículo, las consideraciones a tomar dentro del proceso de diseño y los parámetros obtenidos para el modelo.

3.2. Consideraciones con respecto al diseño.

3.2.1. Con respecto a los sensores.

Como se había mencionado anteriormente el AUV es un vehículo submarino autónomo no tripulado, por lo cual, para que este navegue de manera segura durante el transcurso de una misión dependerá totalmente de los datos entregados por los sensores. Es decir, deberán entregar la información suficiente para que el robot pueda localizarse, percibir elementos de su exterior y regular su posición.

También, para este tipo de vehículos es necesario tomar en cuenta el grado de conocimiento que se tenga del escenario en donde actuará, pues dependiendo de esto se requerirán distintos tipos de sensores, con respecto a esto se consideraron dos alternativas:

- a) Se tiene un grado de conocimiento parcial del lugar de la misión, queriendo decir que se sabe dónde ir o la meta hacia donde se debe encaminar el vehículo, sin embargo, no se conoce en detalle el escenario.
- b) No se tiene conocimiento del lugar de la misión, es decir que el vehículo debe generar un mapa e identificar objetivos según lo sentido.

En este trabajo, el diseño del vehículo se realizó considerando la opción a) por lo que los sensores considerados fueron elegidos en base a esta opción.

3.2.2. Con respecto a la estabilidad y seguridad del vehículo.

En la sección 2.2 se menciona la presencia de fuerzas desestabilizadoras en el modelo del vehículo, para contrarrestar esta situación se hará uso de la estabilidad transversal del vehículo. La estabilidad transversal se refiere a la capacidad del vehículo de volver a la posición normal ante una perturbación en el pitch o roll, la acción de este efecto depende de la magnitud del brazo que hay entre los centros de empuje y gravedad del vehículo.

Este efecto se da debido a que al realizarse una perturbación en pitch o roll, se genera un momento que intenta contrarrestar este movimiento por la acción de las fuerzas de empuje f_e y gravedad f_g

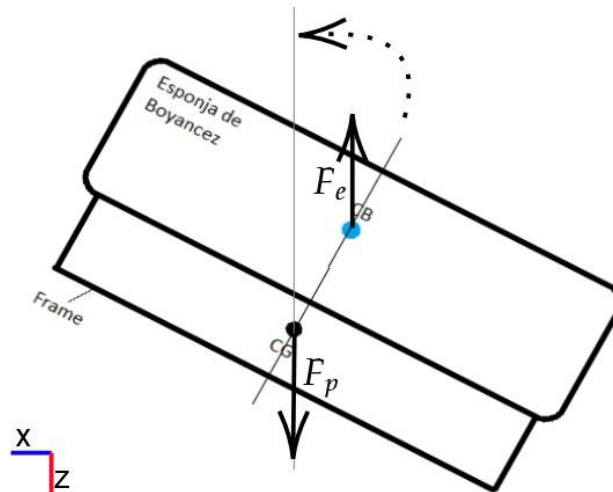


Fig. 3.1. Acción del momento ante una perturbación en la orientación del vehículo.

Por último, con respecto a la seguridad del vehículo vale la pena agregar que la magnitud de la fuerza de empuje f_e sea ligeramente mayor a la magnitud de la fuerza peso f_g de este, así en caso de falla, el vehículo pueda volver a la superficie por acción de f_e .

3.3. Diseño propuesto.

Para el diseño se tomaron en cuenta las consideraciones mencionadas anteriormente y se utilizaron como influencias los diseños descritos en [6] y [11]. Se siguió la idea de que fuera un AUV económico, con un tamaño/peso lo suficiente como para que el número de personas requeridas para trasladarlo no constituya un grupo muy grande, que opere a una profundidad máxima de 100[m], se mueva a una velocidad longitudinal entre 0-1 [nudo] y vertical entre 0-0.25 [nudos].

El diseño propuesto se puede ver en la figura 3.2.

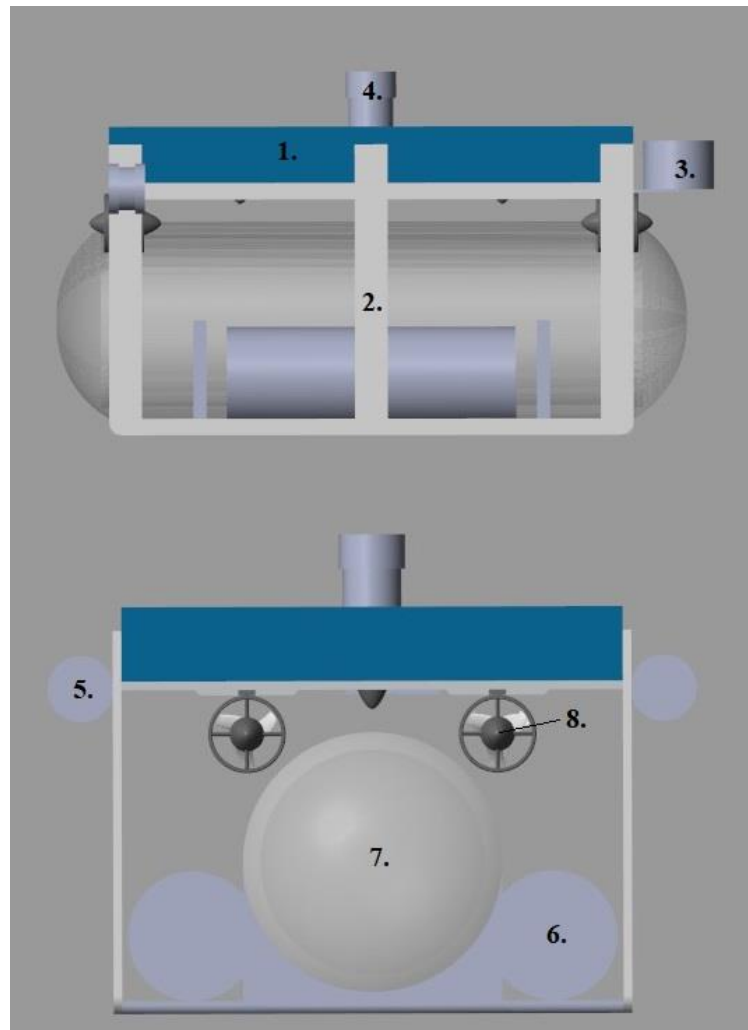


Fig. 3.2. Vistas laterales y frontales del diseño propuesto.

El Diseño Propuesto posee las siguientes características físicas:

| | |
|--------------------------|-----|
| <i>Profundidad [m]</i> | 0.8 |
| <i>Ancho [m]</i> | 0.7 |
| <i>Alto [m]</i> | 0.7 |
| <i>Peso en Seco [kg]</i> | ~40 |

Tabla 3.1. Características físicas del vehículo

Vale comentar que el Alto descrito en Tabla 3.1. incluye la altura del sensor enumerado en 4. de Fig. 3.2, el Ancho incluye al elemento 5. de la Fig. 3.2. y que Peso en Seco corresponde al peso del vehículo en el aire (no sumergido).

De la Fig. 3.2. se tiene que:

1. *Esponja de flotabilidad*: Elemento del vehículo cuya función principal es influir en la fuerza de empuje ejercida al vehículo para que haya flotabilidad cercana a la neutra. Es decir, que $f_e = f_g$. El material del elemento descrito es de poliuretano rígido y las dimensiones se eligieron para lograr el efecto de flotabilidad neutra descrito anteriormente. Estas son,

| | |
|-----------------|-------|
| Profundidad [m] | 0.8 |
| Ancho [m] | 0.58 |
| Alto [m] | 0.086 |

Tabla 3.2. Dimensiones de esponja de flotabilidad

Vale agregar que esta debe ser cubierta con algún otro material como puede ser una capa de plástico debido a que la esponja es absorbente.

2. *Frame o marco de sujeción*: Elemento del vehículo que, como lo dice su nombre, servirá para la sujeción de los componentes del móvil, el material propuesto para este elemento es el Aluminio.
3. *Transponedor USBL*: Elemento del vehículo que sirve para localizarlo en el ambiente que este inmerso.
4. *Sonar*: Sensor cuya función principal es la evasión de obstáculos, está localizado en la parte superior del vehículo debido a que el sonar elegido escanea en 360° alrededor del vehículo.
5. *Luces*: Elemento que sirve para iluminar el frente del vehículo en caso de opacidad u oscuridad en el agua.
6. *Baterías*: Componente del vehículo que sirve para la suministración de energía.
7. *Capsula contenedora*: Elemento que encapsula los componentes como el microcontrolador, cámara y otros sensores, el material de la capsula contenedora es de acrílico y se eligió con un grosor de 2 [cm] para que soporte la presión del agua a una profundidad de 100 [m], la forma de capsula es debido a su simetría y a que la forma de domo reduce la distorsión de la imagen dada por la refracción de los rayos de luz capturada por la cámara [10].
8. *Propulsores*: Componentes principales para el movimiento del vehículo, corresponden a cuatro horizontales y dos verticales.

Los sensores e instrumentos elegidos serán discutidos en la sección de instrumentación del presente trabajo.

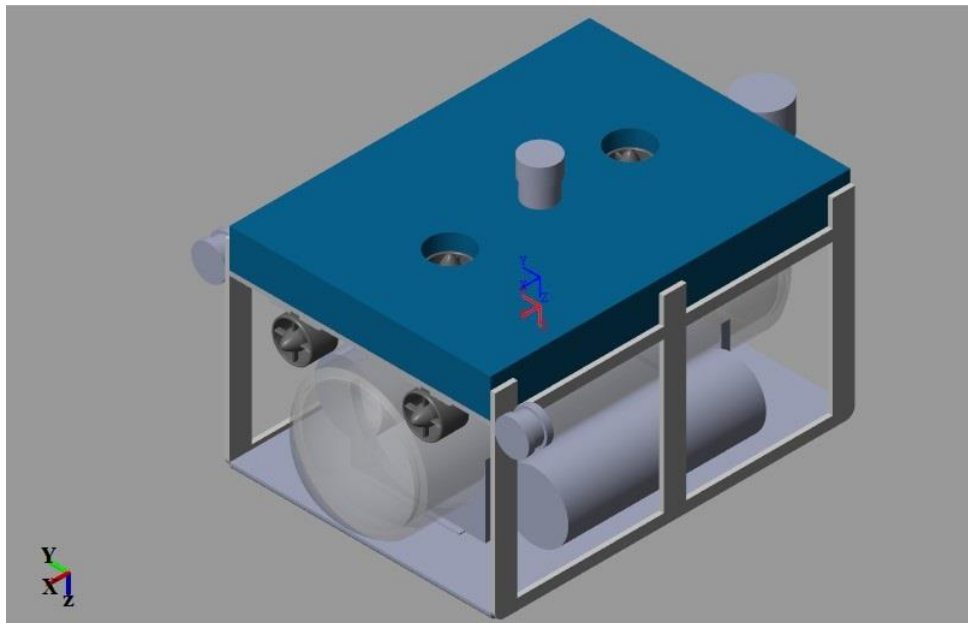


Fig. 3.3. Diseño propuesto en vista isométrica.

3.4. Modelo del Diseño

3.4.1. Parámetros hidrodinámicos.

a) Masa agregada:

Para disminuir la cantidad de elementos de la matriz a obtener, se utiliza la simplificación expresada en (2.15) presentada en [1]. Esto dado a que el diseño presenta simetría en el plano xz de Ref_c .

La obtención de los valores se realiza mediante el método descrito en la sección 2.4.1. Para esto, primero se simplifica el diseño quitándole el sonar, el transpondedor y las luces, luego se exporta la malla del modelo y por último se aplica el proceso descrito en 2.4.1.

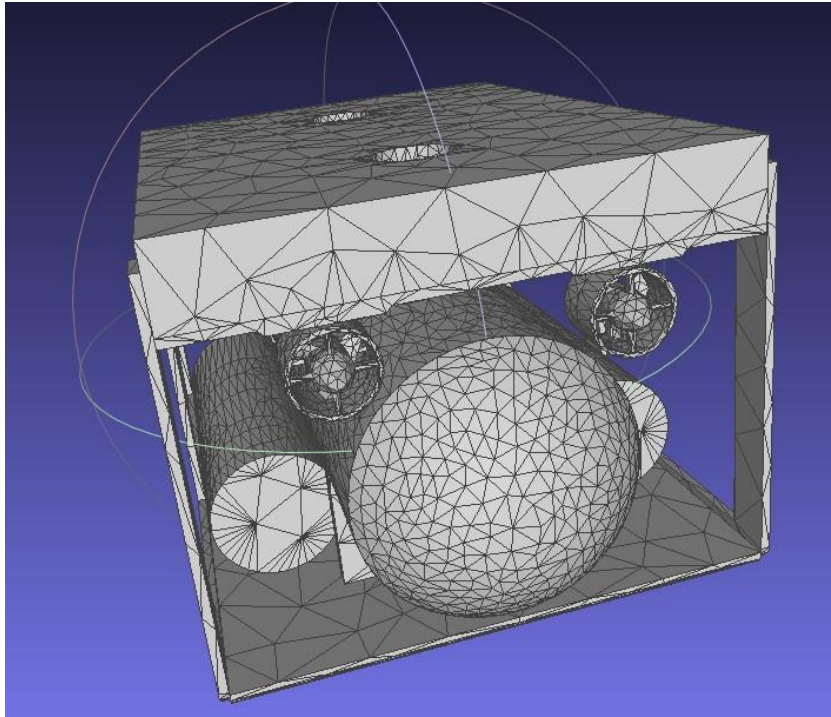


Fig. 3.4. Malla del diseño simplificado para obtener la matriz de masa agregada

La matriz obtenida es:

$$M_a = \begin{bmatrix} 57.7371 & 0 & 1.0276 & 0 & 0.8970 & 0 \\ 0 & 295.7703 & 0 & 2.9015 & 0 & 1.2861 \\ 0.2996 & 0 & 124.1854 & 0 & 0.5075 & 0 \\ 0 & 3.0601 & 0 & 5.8980 & 0 & 0.4505 \\ 1.8098 & 0 & 0.4692 & 0 & 4.7545 & 0 \\ 0 & 3.1662 & 0 & 0.3173 & 0 & 9.6350 \end{bmatrix} \quad (3.1)$$

b) Fuerzas de amortiguamiento:

Para obtener las fuerzas de amortiguamiento que actúan sobre el vehículo, se utiliza la expresión (2.16) y se aplica el proceso descrito en la sección 2.4.2.

Los valores de Fuerzas de amortiguamiento obtenidos para las velocidades lineales y angulares se pueden observar en las Fig. 3.5. y Fig.3.6

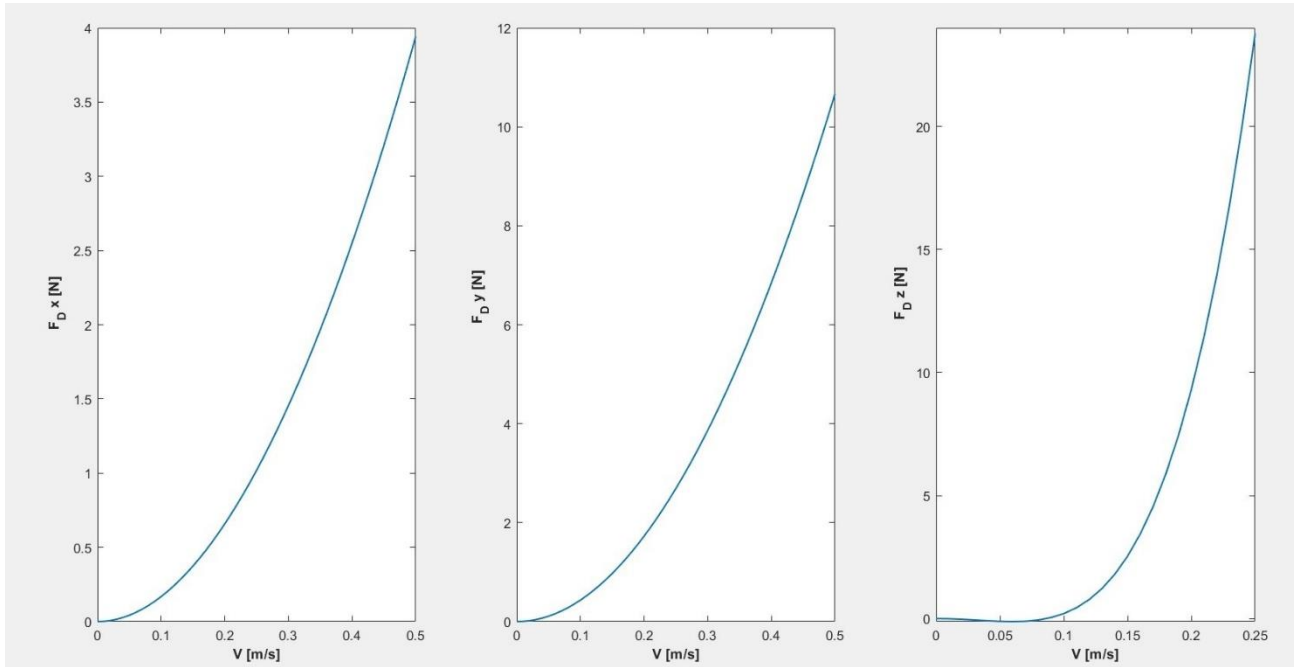


Fig. 3.5. Fuerzas de amortiguamiento obtenidas para velocidades longitudinales y verticales.

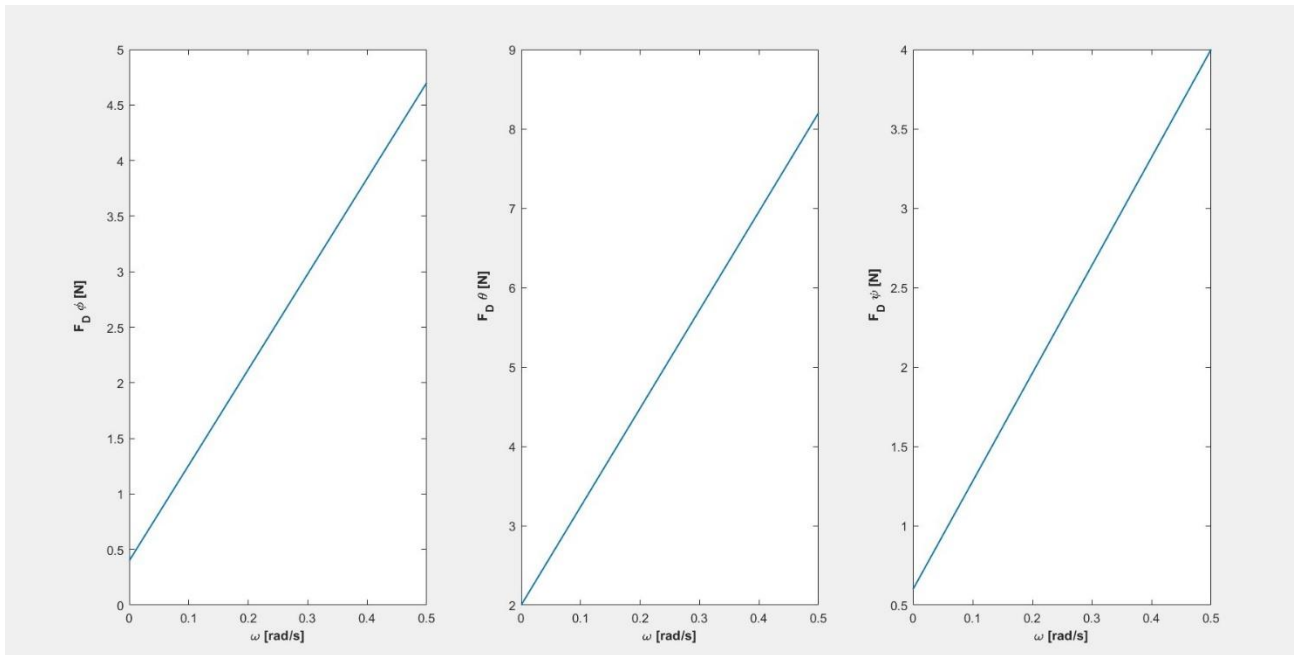


Fig. 3.6. Fuerzas de amortiguamiento obtenidas para las velocidades angulares

3.4.2. Implementación del modelo en Simulink.

Una vez se tienen los datos requeridos para poder representar el modelo, se cargan los datos obtenidos en la sección 3.4.1 y también el archivo del diseño del vehículo mediante el uso del plugin de Simscape Multibody.

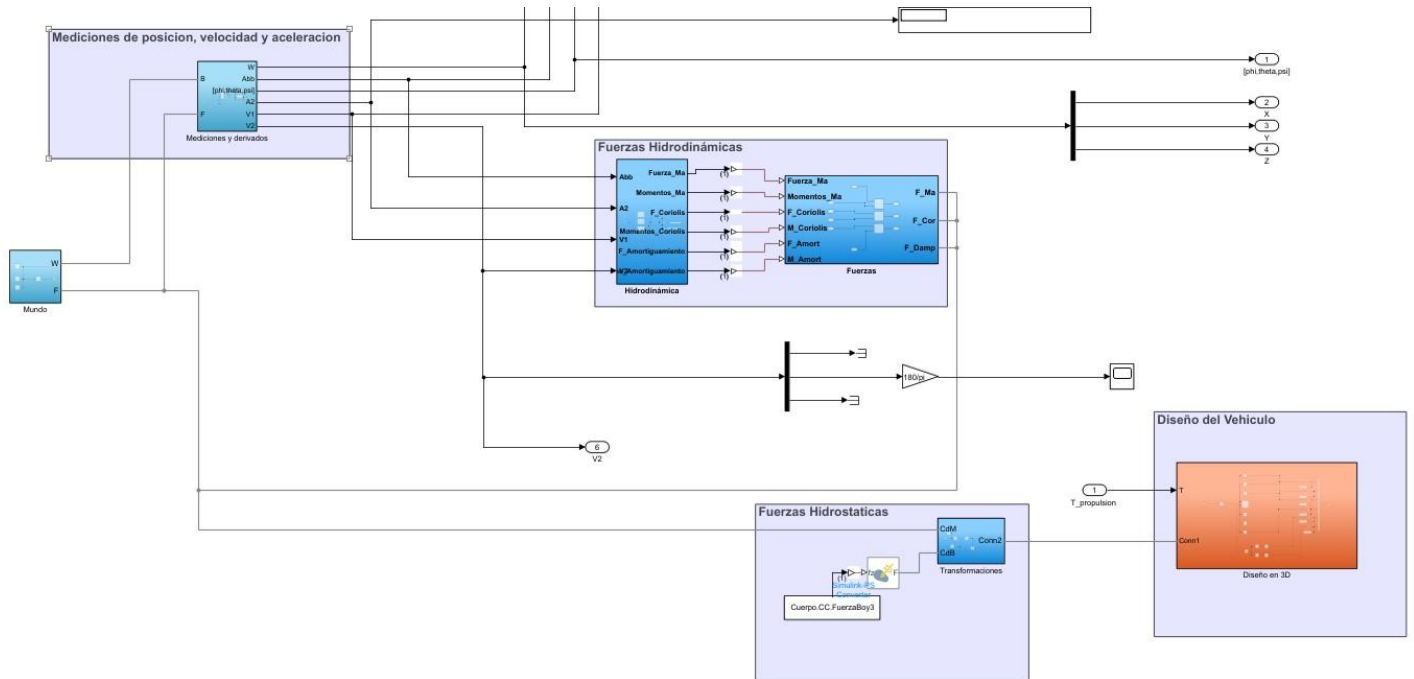


Fig. 3.7. Implementación del diseño y modelo del vehículo en Simulink.

En la Fig. 3.7. se puede ver la implementación en el programa, dentro de este se pueden diferenciar cuatro grupos principales, *Mediciones de posición, velocidad y aceleración*, *Fuerzas Hidrodinámicas*, *Fuerzas Hidrostáticas* y *Diseño del Vehículo*. En el primer grupo se realizan las mediciones de posición, orientación, velocidad y aceleración, para las dos primeras se efectúa la medición con respecto a Ref_t y las restantes se ejercen en relación con Ref_c , en el segundo grupo se implementan los parámetros obtenidos en la sección 3.4.1, para las fuerzas Coriolis se realiza la parametrización descrita por (2.17), en el tercer grupo son incluidas f_e y f_g . Por último, en el grupo de *Diseño de Vehículo* se encuentra el ensamblaje del vehículo realizado en SolidWorks con las respectivas relaciones entre sus piezas.

3.4.3. Obtención Modelo Propulsores

Para los propulsores, la relación entre la fuerza de propulsión y la velocidad de rotación de la hélice se obtiene mediante la información proporcionada por el fabricante del propulsor elegido, los datos entregados varían según el voltaje al cual es alimentado el propulsor

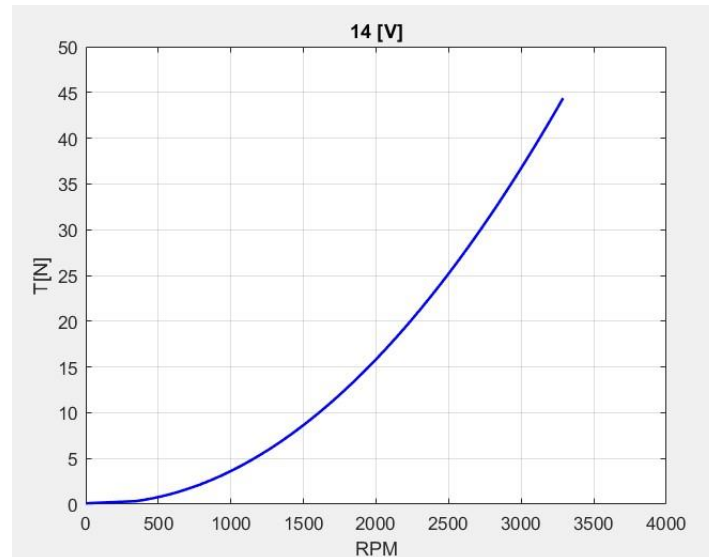


Fig. 3.8. Fuerza de propulsión a Voltaje nominal.

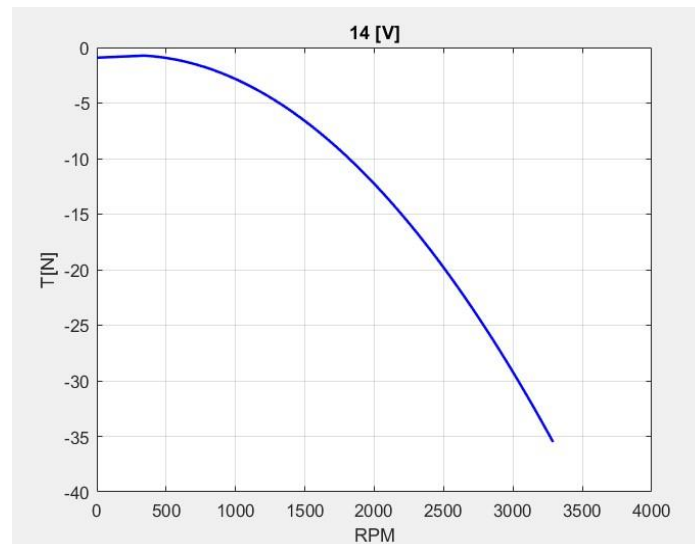


Fig. 3.9. Fuerza del Propulsor en reversa a Voltaje Nominal.

La obtención de las curvas ilustradas en Fig. 3.8. y Fig. 3.9. son obtenidas mediante una interpolación polinomial de los datos entregados, a continuación, se presentan las fuerzas de propulsión para distintos valores de alimentación.

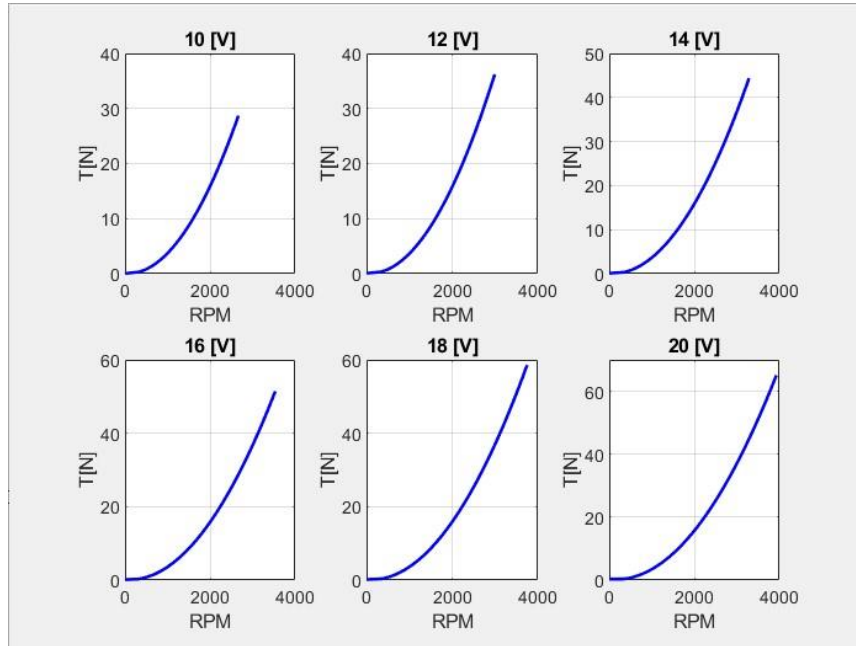


Fig. 3.10. Fuerza de propulsión para distintos valores de alimentación.

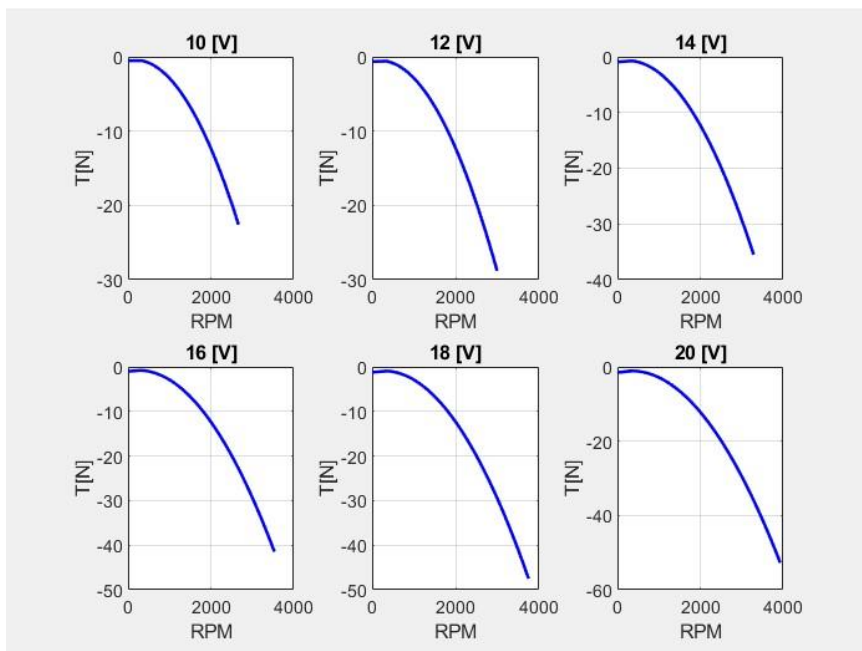


Fig. 3.11. Fuerza de propulsión en reversa para distintos valores de alimentación.

4. Regulación

4.1. Introducción

En este capítulo se presenta el sistema de control para la velocidad u , ψ (yaw), profundidad y θ (pitch) del vehículo.

4.2. Principio de funcionamiento

Para el sistema de regulación, las variables a controlar corresponden a la de profundidad del vehículo z , θ , u y ψ donde, para las primeras dos se utilizan los propulsores verticales y para las restantes los cuatro propulsores horizontales. La razón por la cual se controla la velocidad u y ψ es debido a que el control de la navegación utiliza estas variables para generar la trayectoria que debe tomar el vehículo con tal de llegar a la meta de forma segura.

El sistema de control funciona en base a la variación de la cantidad de fuerza de propulsión que entrega cada propulsor o thruster, por ejemplo, para que el vehículo gire cierta cantidad sobre su eje z (ψ), la fuerza ejercida por los propulsores horizontales internos de la rotación (encerrados por G1 en Fig. 4.1.) debe ser menor que la de los externos en la rotación (aquellos encerrados por G2).

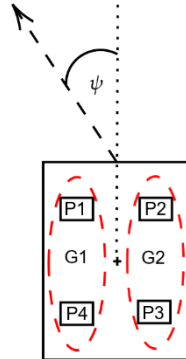


Fig. 4.1. Situación en la cual el vehículo gira cierta cantidad ψ .

Para el control de altura y θ se utiliza el mismo tipo de lógica, si se quiere el vehículo en cierto ángulo de pitch θ , el aporte de uno de los propulsores debe ser mayor que el otro para poder realizar la acción pedida.

4.2.1. Control de ψ y Velocidad.

En la presente estrategia de control se pueden distinguir dos bloques, uno que corresponde a la fuerza de propulsión F_{ψ} que necesitan ejercer los propulsores para orientar al vehículo cierto ángulo ψ_{ref} y otro que entrega la propulsión F_v necesaria para mover el robot a una velocidad u_{ref} , luego,

las fuerzas de propulsión ejercidas por cada motor horizontal del diseño se dan por las ecuaciones.

$$F_{M1} = F_v + F_\psi \quad (4.1)$$

$$F_{M2} = F_v - F_\psi \quad (4.2)$$

$$F_{M3} = F_v - F_\psi \quad (4.3)$$

$$F_{M4} = F_v + F_\psi \quad (4.4)$$

Donde, F_{Mi} corresponde a la fuerza de propulsión generada por el propulsor M_i con $i = \{1,2,3,4\}$, la numeración es correspondiente a la de Fig.4.1.

4.2.2. Control de Profundidad z y θ .

Para el control de profundidad y ángulo de pitch se utiliza la misma estrategia que la anterior, en este caso las fuerzas de propulsión aportadas por cada actuador están dados por (4.5) y (4.6),

$$F_{MV1} = F_{prof} + F_\theta \quad (4.5)$$

$$F_{MV2} = F_{prof} - F_\theta \quad (4.6)$$

Donde F_{MVi} corresponde a la fuerza de propulsión generada por el propulsor vertical MVi , con $i = \{1,2\}$ y F_{prof} , F_θ la contribución de fuerzas generadas por el sistema de regulación.

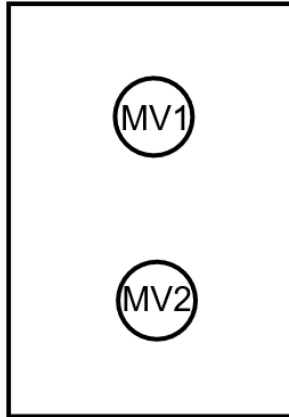


Fig. 4.2. Ilustración de Motores Verticales del Vehículo.

A continuación, se presentan respuestas de los sistemas de regulación

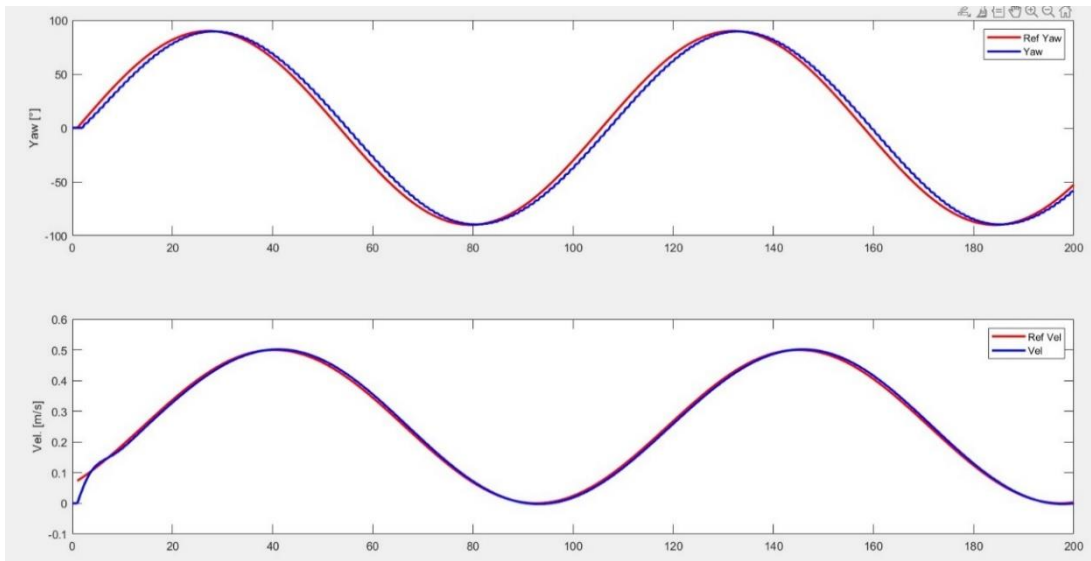


Fig.4.3. Respuesta ante variación senoidal de ψ y Velocidad.

En la Fig. 4.2. se puede ver la respuesta dinámica del sistema de regulación ante cambios senoidales de sus respectivas referencias.

Para las simulaciones, en la profundidad se utilizó la señal como la que se ve en Fig.4.4. , y para el ángulo θ se realizaron pruebas de la dinámica del sistema prueba mediante el uso de referencias senoidales.

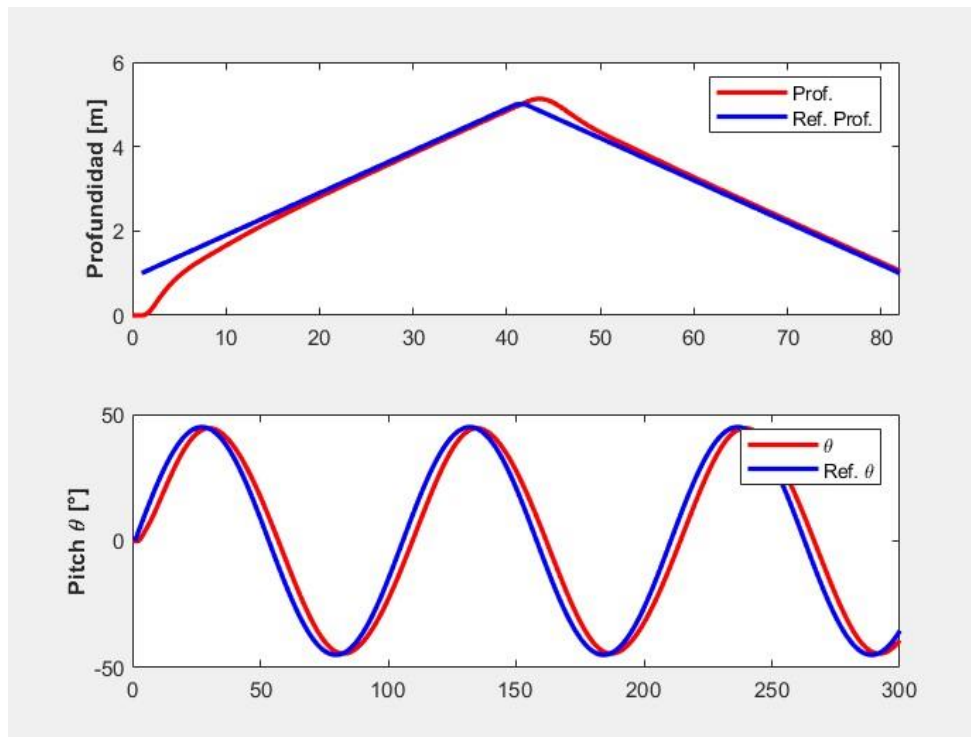


Fig.4.4. Respuesta ante variaciones de profundidad y θ .

5. Navegación

5.1. Introducción

En el siguiente capítulo se presenta el sistema de navegación y localización a proponer para el vehículo.

5.2. Sistemas de navegación.

En los vehículos de navegación autónoma el problema que se aborda depende de la cantidad de información conocida con respecto al terreno en donde se opere, de aquí [13] distingue dos grupos, de *navegación local* y de *navegación global*.

En el primero, el vehículo genera su camino dependiendo de la información instantánea que se vaya capturando del ambiente, no se tiene una meta fija, ni un mapa o una referencia previa del lugar en donde se opere. Ejemplos de este tipo de navegación en un vehículo submarino se pueden ver en [14] y [15] en donde, mediante el uso de cámaras se utilizan algoritmos SLAM (Simultaneous Localization and Mapping) en la navegación del móvil.

Por otra parte, la *navegación global* corresponde a aquella donde se tiene un conocimiento previo del ambiente de operación del vehículo y este puede seguir una ruta o llegar a una meta previamente generada.

Anteriormente, en la sección 3.2.1 se había mencionado que los sensores a utilizar serían seleccionados, considerando que se tiene un grado de conocimiento parcial del ambiente en donde el móvil está operando, por ende, el problema de navegación a abordar corresponde a un problema de *navegación global*.

Para abordar este problema es necesaria la utilización de instrumentos que permitan al vehículo localizarse y obtener información del lugar en donde esté inmerso para que este pueda moverse de forma segura en el lugar de trabajo, lo que principalmente se traduciría, llegar a la meta sin colisiones.

5.2.1. Localización.

El sistema de localización propuesto consiste en el uso de trasponedores acústicos, los cuales están dispuestos en la superficie fijado a una boya y en el vehículo. Un trasponedor ubicado en la superficie se comunicará con otro del mismo tipo sujeto en el vehículo, además, este posee la capacidad de transmitir la posición del móvil mediante coordenadas GPS, para esto es necesario conectarlo a una antena que ocupe este tipo de protocolo.

La localización del vehículo se calcula mediante el uso del tiempo de llegada (TOA) de la señal enviada y recibida por cada transpondedor.

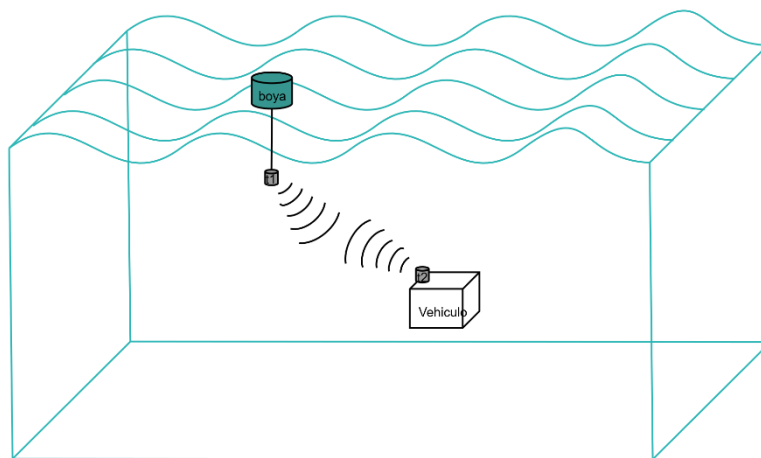


Fig. 5.1. Diagrama representando el esquema de localización

5.3. Control de navegación.

Como se había mencionado previamente el problema de navegación a abordar es del tipo de *navegación global*, donde se tiene cierto conocimiento con respecto al área de trabajo del vehículo. Si bien, se posee información del lugar en donde se esté operando, hay que considerar que esta puede ser incompleta o aproximada, también hay que agregar que este tipo de ambientes no son controlados y generalmente se presentan agentes externos como personas u otro tipo de objeto moviéndose en el ambiente. Es decir, el vehículo debe trabajar con esta incertidumbre en la información, de tal forma que se logren los objetivos propuestos de manera segura, para esto resulta conveniente el uso de la lógica difusa. A continuación, se realizará una breve descripción de la lógica difusa y la estructura de un controlador que ocupa este tipo de lógica.

5.3.1. Lógica difusa.

La lógica difusa es una forma de modelar el razonamiento lógico en donde, a diferencia de la lógica binaria el valor de verdad de una proposición no se divide en dos estados *1/0* o *Verdadero/Falso*, si no que en un valor u en el intervalo $[0,1]$.

5.3.2. Controlador difuso.

El controlador difuso es aquel que utiliza la lógica difusa para el control de un proceso, no requiere de una modelación compleja de este y actúa en base al conocimiento de un experto. Esto lo realiza mediante el uso de reglas heurísticas *IF-THEN* que son activadas según las condiciones que se presenten, la forma de estas reglas es:

*IF (son satisfechas ciertas condiciones) THEN (un conjunto de consecuencias
Puede ser inferido)*

Dentro de esta regla el antecedente es una condición en el dominio de la aplicación y el consecuente es una acción de control para el sistema.

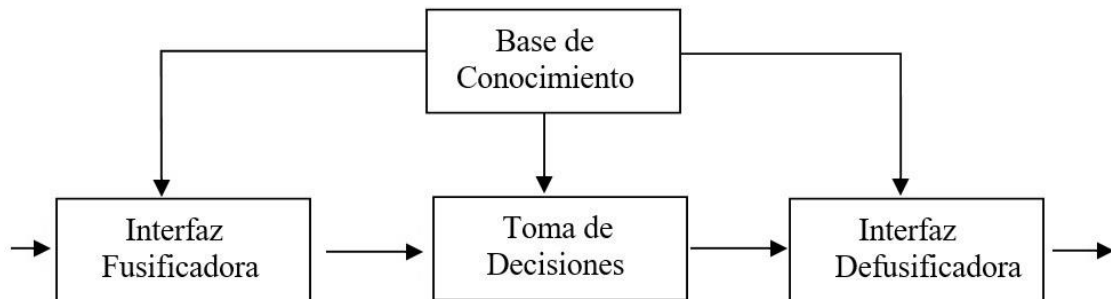


Fig. 5.2. Estructura típica de controlador difuso

En Fig. 5.1. se puede ver la estructura típica de un controlador difuso, dentro de esta se distingue la interfaz fusificadora, base de conocimiento, toma de decisiones e interfaz defusificadora.

1. *Interfaz fusificadora*: Etapa del controlador en donde se mapean los valores de entrada a un conjunto difuso, esto se hace mediante el uso de funciones de pertenencia μ_F la cual transforma el dato de entrada en una variable lingüística.
2. *Toma de decisiones*: Núcleo del controlador difuso que posee la capacidad de simular el comportamiento de toma de decisiones humana en presencia de conceptos difusos. Además, puede inferir acciones de control difuso utilizando implicaciones difusas.

3. *Base de Conocimiento*: Consiste en una base de datos y una de reglas de control lingüística. En la primera, se proveen las definiciones necesarias para definir las reglas de control lingüístico y manipulación de datos difusos, y en la segunda se caracteriza el objetivo de control y el curso del control en el dominio experto.
4. *Interfaz defusificadora*: Etapa final del controlador que se encarga de convertir las variables difusas en valores escalares. Existen diversos métodos para realizar esta operación, dentro de los cuales el más común es el método del centroide el cual ubica el centro de gravedad de la distribución de la acción de control, esto se realiza mediante la ecuación:

$$z_0 = \frac{\sum_{j=1}^n \mu_z(w_j) w_j}{\sum_{j=1}^n \mu_z(w_j)} \quad (5.1)$$

5.3.3. Propuesta de Control de Navegación.

La propuesta de control a utilizar se fundamenta en la navegación basada en comportamientos expresados en [17], [18] y [19], donde se declara que el objetivo a cumplir es logrado al subdividir la tarea completa en diferentes comportamientos, que se enfocan en la ejecución de subtarear específicas [19]. Por ejemplo, en este caso se requiere que el vehículo navegue de forma segura a una meta, para esto se divide la tarea completa en dos comportamientos: *Evasión de obstáculos* y *Seguimiento de meta*, el primero, como lo dice su nombre se enfoca en cambiar la pose del robot para mantener la seguridad de éste y el segundo se centra en direccionar el vehículo hacia la meta especificada.

Cada uno de estos comportamientos actúan de forma individual, toman los valores de entrada, los fusifican y generan una salida acorde a la conducta que representan, pueden poseer valores de entrada diferentes, pero la o las variables de salida que cada uno de estos controla son las mismas, estas variables son $\Delta\psi$ y v , donde la primera corresponde al cambio de orientación ψ y la segunda es la velocidad del vehículo sobre el eje x de Ref_c .

A continuación, se presentará un diagrama de bloques que representa el principio de funcionamiento del sistema de control difuso a proponer.

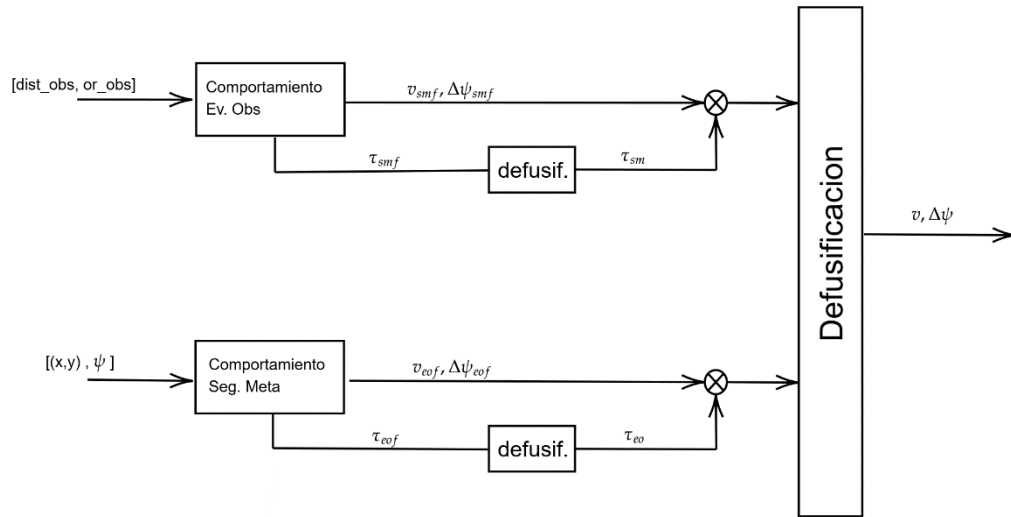


Figura 5.3. Diagrama en bloques de Control de navegación difuso

Para el comportamiento de *Evasión de obstáculos* los valores de entrada provienen de datos de sensores los cuales son procesados en un vector que entrega la distancia y orientación del obstáculo más cercano y para *Seguimiento de meta* el valor de entrada es un vector que entrega la distancia y orientación hacia la meta propuesta.

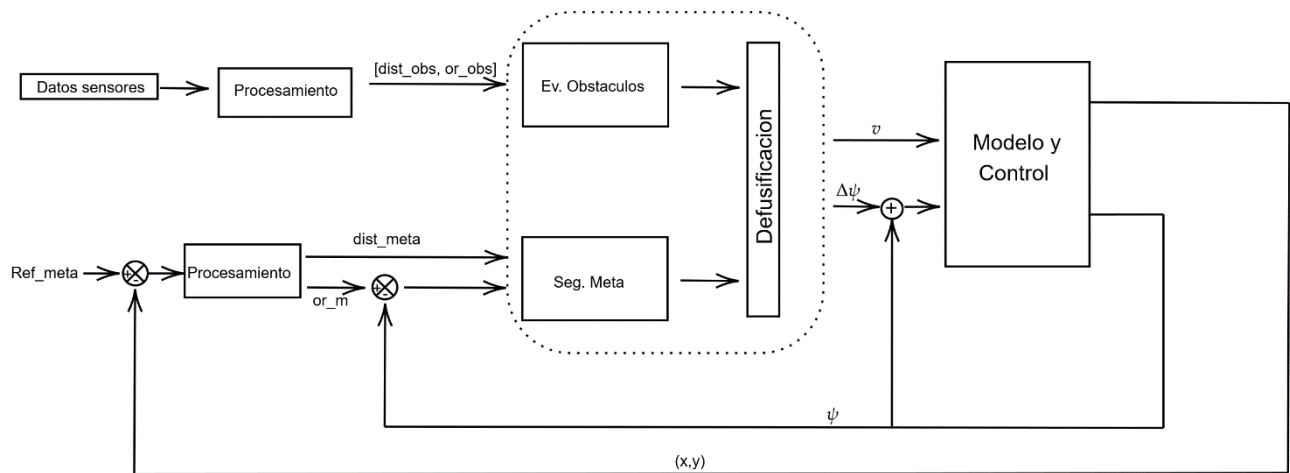


Fig. 5.4. Diagrama de bloques del sistema de control de navegación conectado con el modelo del vehículo.

5.3.4. Fusificación

Como se había mencionado anteriormente, en la etapa de fusificación a las variables de entrada se le asigna un grado de pertenencia de acuerdo con las variables lingüísticas, en este caso se deben generar funciones de pertenencia para las entradas de cada comportamiento.

1. *Evasión de obstáculos:*

En este comportamiento, se interpretan los datos entregados por los sensores, los cuales son procesados en forma de un vector de distancia/orientación. Para la distancia las funciones de pertenencia son: Cerca, Medio, Lejos y para la orientación se utilizarán: Frontal Izquierdo (FRIZQ), Frontal Derecho (FRDER), Frente Izquierdo (FIZQ), Frente Derecho (FDER), Derecha-Derecha (DDER), Izquierda-Izquierda (IIZQ), donde, los últimos dos representan valores de orientación que se encuentran al costado del vehículo.

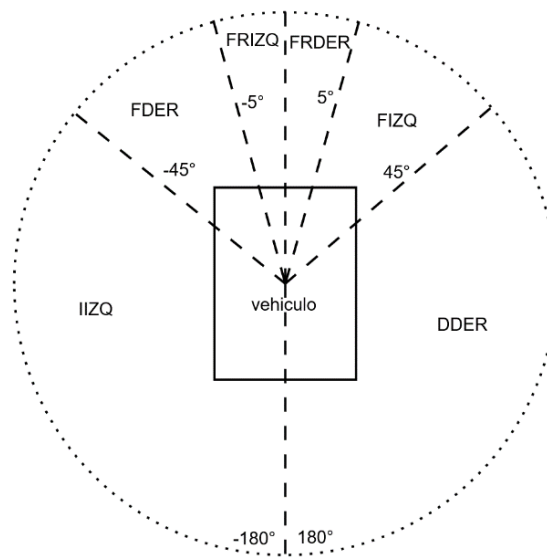


Fig. 5.5. Representación de las variables lingüísticas de la orientación del obstáculo.

Las funciones a utilizar son trapezoidales y triangulares, con solapamiento entre ellas para obtener una respuesta más fina.

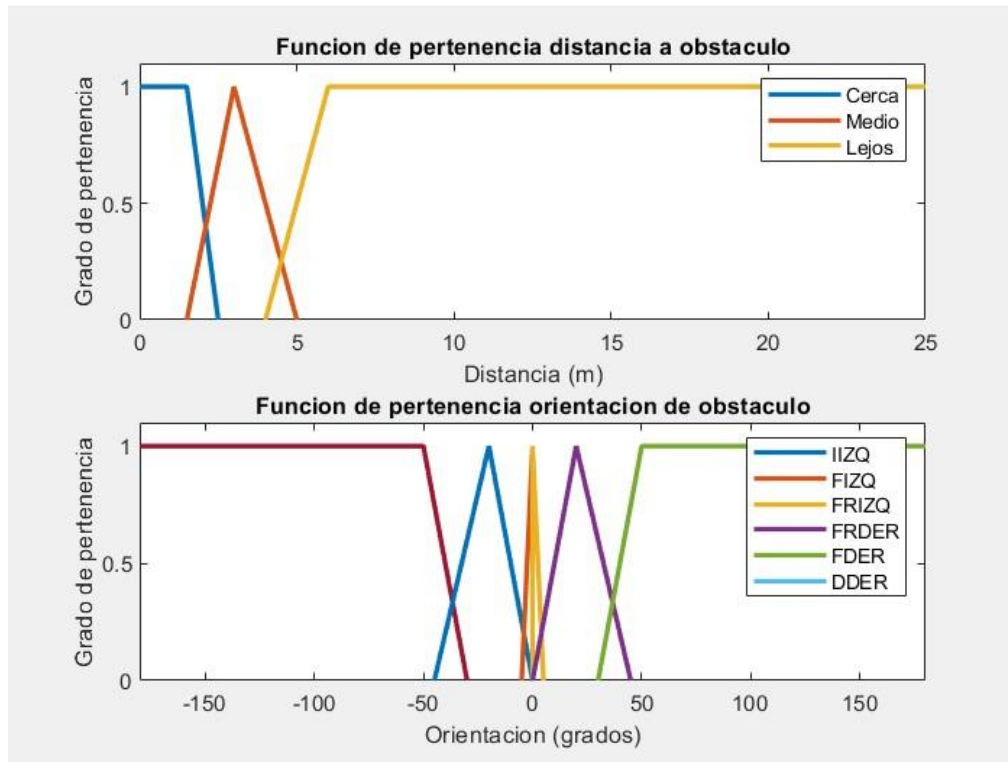


Fig. 5.6. Funciones de pertenencia para las entradas de la evasión de obstáculos.

2. *Seguimiento de meta:*

Los datos de entrada son la diferencia entre la orientación necesaria para encaminarse a la meta y la posición (ψ) del vehículo, así como también la distancia a la que se encuentra el vehículo de la meta especificada. Para la orientación necesaria las funciones de pertenencia son: Negativo Grande (Ng), Negativo Medio (Nm), Negativo Pequeño (Np), Cero (Z), Positivo Pequeño (Pp), Positivo Medio (Pm), Positivo Grande (Pg), para la distancia entre el vehículo y la meta las funciones son: Cero (Z), Cerca (C), Regular (RE), Medio (M) y Lejos (L).

Como en el comportamiento anterior, las funciones a utilizar son triangulares y trapezoidales con un solapamiento entre ellas con el fin de obtener una respuesta certera.

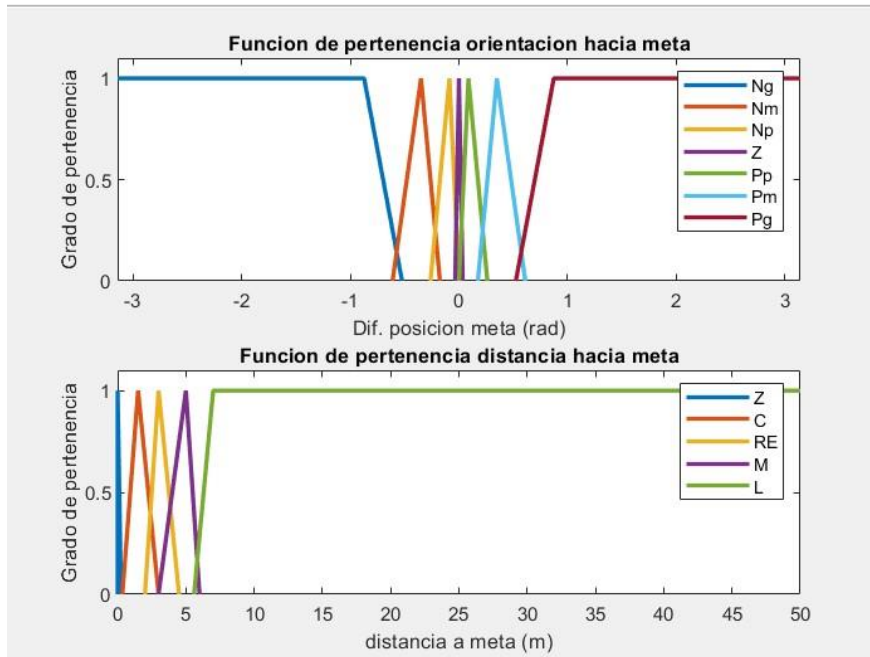


Fig. 5.7. Funciones de pertenencia para las entradas en el seguimiento de meta

5.3.5. Salida e integración de los comportamientos.

Como se puede ver en Fig. 5.2. cada comportamiento posee tres variables de salida, v la velocidad del vehículo, $\Delta\psi$ el cambio de la posición yaw (ψ) del vehículo y τ el peso de la variable a controlar según la conducta, en esta sección se explicará el proceso previo a la defusificación y la función de los pesos τ .

Las variables de salida que controla cada conducta son las mismas, sin embargo, los valores que proponen son distintos y deben ser tomados en cuenta para el valor de salida final (luego de la defusificación), en otras palabras, cada comportamiento genera una recomendación de acción en base al conocimiento que este tenga y es necesario que la decisión de acción final sea realizada en base a estas recomendaciones.

La toma de decisión de la acción se realiza dependiendo de la situación en el cual se encuentre el robot, por ejemplo, en la Fig. 5.7. se ilustra la situación en la que el vehículo se está encaminando hacia la meta propuesta, sin embargo, se encuentra en la cercanía de un obstáculo que bloquea su camino, para este caso la recomendación dada por el comportamiento encargado de la evasión de obstáculos debe tener un peso mayor en la decisión de acción final.

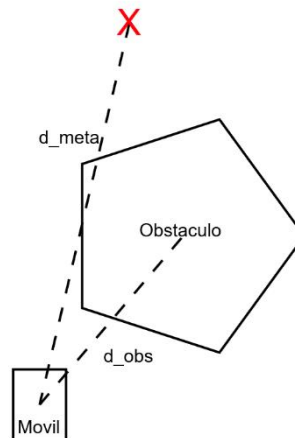


Fig. 5.8. Situación en la que el vehículo se encuentra cercano a un obstáculo.

En esta situación u otras que se presenten se hace uso del concepto de grado de aplicabilidad o DOA que es introducido en [18], el cual consiste en una medida de activación de un comportamiento, es un valor $\tau \in [0,1]$ que varía según la situación o contexto donde el vehículo se encuentre, esto lo hace mediante el uso de las reglas de aplicabilidad las que forman parte de los comportamientos, una vez obteniendo el valor, este es defusificado mediante el uso del método de la centroide, para que luego interactúe con las variables de salida o recomendaciones de cada comportamiento, esto se puede ver en el diagrama de Fig 5.2.

A continuación, se presentarán las funciones de pertenencia de salida para cada comportamiento, las cuales son v , $\Delta\psi$ y τ :

1. *Evasión de obstáculos:*

Las funciones de pertenencia para las tres variables están compuestas de conjuntos triangulares y trapezoidales. Donde, para la velocidad v las variables lingüísticas asociadas a cada conjunto son: Velocidad Mínima (Vmin), Velocidad Media (Vmed), Velocidad máxima (Vmax), aquí el rango de velocidades es $[0,0.5]$ (m/s) las cuales son acordes al rango de velocidad del diseño mencionado en la sección 3.3 del presente trabajo. Para el cambio de pose u orientación $\Delta\psi$ las variables lingüísticas asociadas son Fuerte Izquierda (FIZQ), Medio Izquierda (MIZQ), Poco Izquierda (PIZQ), Recto (REC), Poco Derecha (PDER), Medio Derecha (MDER), Fuerte Derecha (FDER), estas variables representan la fuerza del cambio de posición del vehículo la que se suma o se resta del ψ dependiendo de la dirección de giro (izquierda o derecha). Por último, para el peso o el grado de aplicabilidad τ se tienen: Bajo (B), Medio (M), Alto (A).

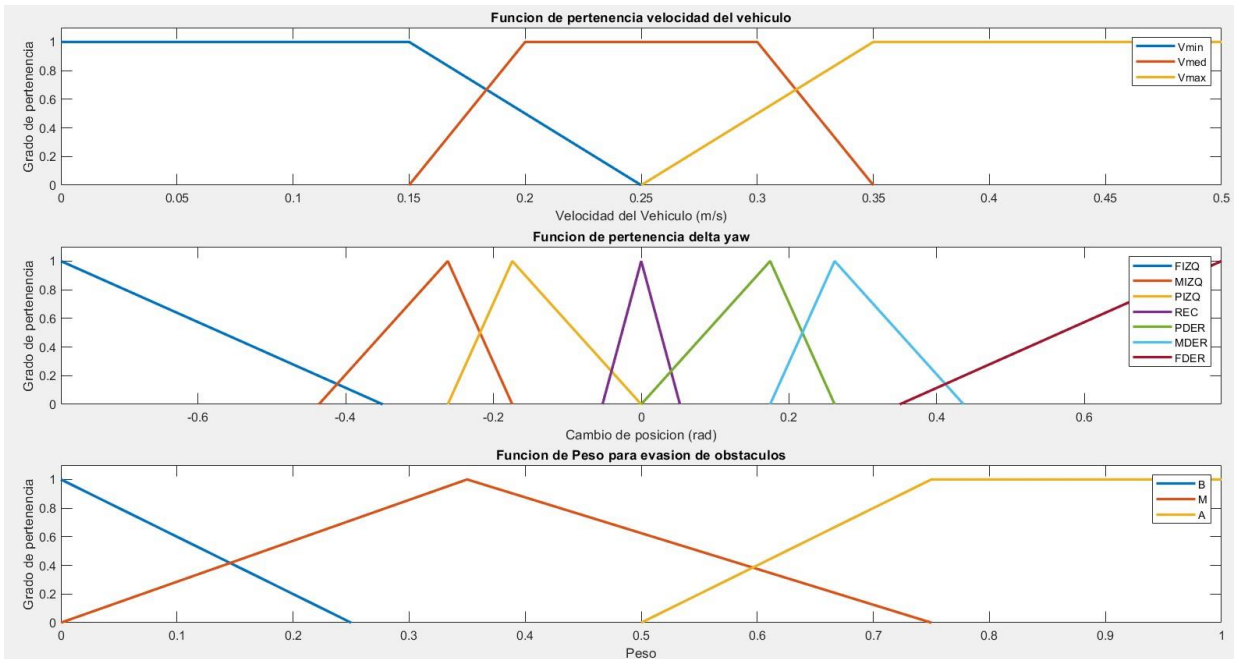


Fig. 5.9. Funciones de pertenencia para la salida del comportamiento de evasión de obstáculos.

2. Seguimiento de Meta:

Para este comportamiento las variables lingüísticas correspondientes para la velocidad v son: Velocidad Mínima (Vmin), Velocidad Pequeña (Vp), Velocidad Media (Vmed), Velocidad Máxima (Vmax), a diferencia del comportamiento anterior se considera una variable lingüística de velocidad adicional para obtener una respuesta más certera en el seguimiento de la meta, vale decir, que los rangos de velocidad son los mismos que los mencionados en la conducta anterior. Para el cambio de posición las variables lingüísticas son Negativo Grande (Ng), Negativo Medio (Nm), Negativo Pequeño (Np), Cero (Z), Positivo Pequeño (Pp), Positivo Medio (Pm), Positivo Grande (Pg), estos cumplen la misma función que la mencionada en el comportamiento anterior. Para el peso τ se tienen: Bajo (B), Medio (M), Alto (A).

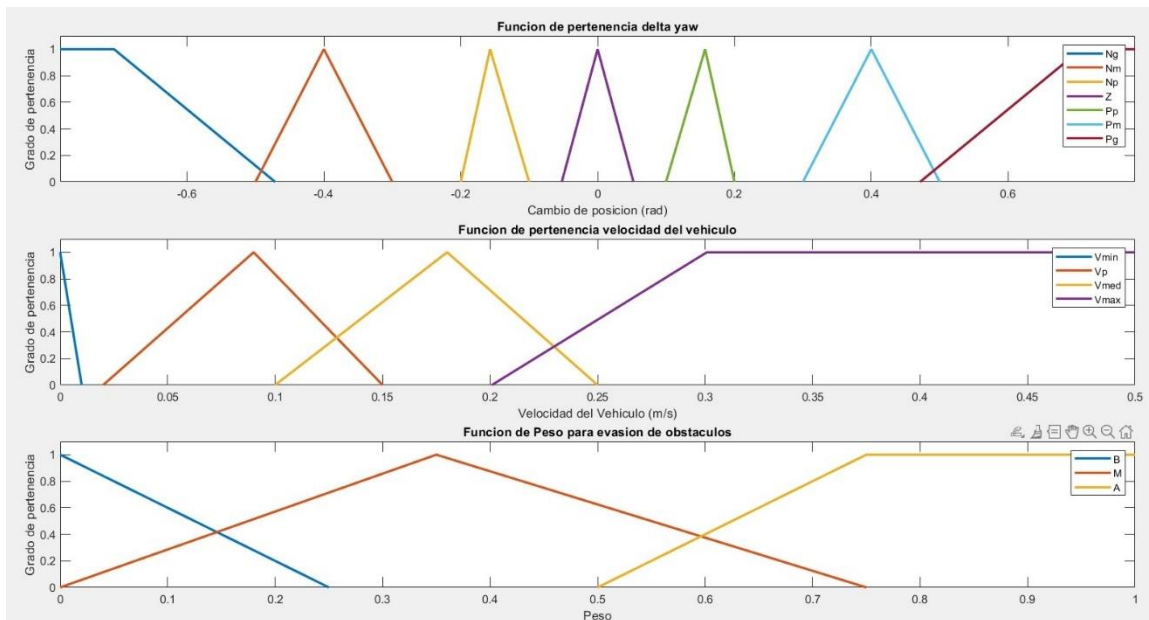


Fig. 5.10. Funciones de pertenencia para la salida del comportamiento seguimiento de meta.

5.3.6. Motor de inferencias:

A continuación, se presentarán las reglas a utilizar para la acción del control de navegación, estas son divididas según el tipo de comportamiento.

Dado el número de entradas en cada comportamiento se encuentra que son dieciocho para la *evasión de obstáculos*, treinta y cinco para el seguimiento de meta.

| N° Regla | Dist. | Or. | v | $\Delta\psi$ | τ |
|----------|-------|-------|--------|--------------|--------|
| 1 | Cerca | IIZQ | Vmedia | FDER | A |
| 2 | Cerca | FIZQ | Vmin | FDER | A |
| 3 | Cerca | FRIZQ | Vmin | FDER | A |
| 4 | Cerca | FRDER | Vmin | FIZQ | A |
| 5 | Cerca | FDER | Vmin | FIZQ | A |
| 6 | Cerca | DDER | Vmedia | FIZQ | A |
| 7 | Medio | IIZQ | Vmax | PDER | B |
| 8 | Medio | FIZQ | Vmedia | MDER | M |
| 9 | Medio | FRIZQ | Vmedia | MDER | M |
| 10 | Medio | FRDER | Vmedia | MIZQ | M |
| 11 | Medio | FDER | Vmedia | MIZQ | M |
| 12 | Medio | DDER | Vmax | PIZQ | B |
| 13 | Lejos | IIZQ | Vmax | REC | B |
| 14 | Lejos | FIZQ | Vmax | PDER | B |
| 15 | Lejos | FRIZQ | Vmax | REC | B |
| 16 | Lejos | FRDER | Vmax | REC | B |
| 17 | Lejos | FDER | Vmax | PIZQ | B |
| 18 | Lejos | DDER | Vmax | REC | B |

Tabla 5.1. Reglas para comportamiento Evasión de obstáculos.

Para la evasión de obstáculos, las reglas se centran en regular la velocidad del vehículo según la cercanía del obstáculo, el cambio de pose del vehículo es realizado en la dirección contraria a la orientación percibida y su cantidad varía dependiendo de la cercanía.

| N° Regla | Dif. Pose | Dist. | v | $\Delta\psi$ | τ |
|-----------------|------------------|--------------|-----------------------|--------------------------------|--------------------------|
| 1 | Z | Z | Vmin | Z | A |
| 2 | Nm | Z | Vmin | Z | A |
| 3 | Pm | Z | Vmin | Z | A |
| 4 | Ng | Z | Vmin | Z | A |
| 5 | Np | Z | Vmin | Z | A |
| 6 | Pp | Z | Vmin | Z | A |
| 7 | Pg | Z | Vmin | Z | A |
| 8 | Nm | C | Vmin | Nm | A |
| 9 | Z | C | Vp | Z | A |
| 10 | Pm | C | Vmin | Pm | A |
| 11 | Ng | C | Vmin | Ng | A |
| 12 | Np | C | Vmin | Np | A |
| 13 | Pp | C | Vmin | Pp | A |
| 14 | Pg | C | Vmin | Pg | A |
| 15 | Nm | RE | Vmin | Np | M |
| 16 | Z | RE | Vp | Z | M |
| 17 | Pm | RE | Vmin | Pm | M |
| 18 | Ng | RE | Vmin | Ng | M |
| 19 | Np | RE | Vmin | Np | M |
| 20 | Pp | RE | Vmin | Pp | M |
| 21 | Pg | RE | Vmin | Pg | M |
| 22 | Nm | M | Vmed | Nm | M |
| 23 | Z | M | Vmin | Z | M |
| 24 | Pm | M | Vmin | Pm | M |
| 25 | Ng | M | Vmin | Ng | M |
| 26 | Np | M | Vmin | Np | M |
| 27 | Pp | M | Vmin | Pp | M |
| 28 | Pg | M | Vmin | Pg | M |
| 29 | Nm | L | Vmin | Nm | M |
| 30 | Z | L | Vmax | Z | M |
| 31 | Pm | L | Vmin | Pm | M |
| 32 | Ng | L | Vmin | Ng | M |
| 33 | Np | L | Vmin | Np | M |
| 34 | Pp | L | Vmin | Pp | M |
| 35 | Pg | L | Vmin | Pg | M |

Tabla 5.2. Reglas para comportamiento de seguimiento de meta.

En la Tabla 5.2. se pueden ver las reglas que definirán el actuar del vehículo bajo el comportamiento del seguimiento de la meta, con respecto a las salidas v y $\Delta\psi$ hay que comentar que estas fueron elegidas de tal forma que se priorice buscar la orientación correcta sobre llegar rápido a la meta.

5.3.7. Defusificación.

Como se había mencionado antes la etapa de defusificación es la etapa final del controlador difuso, para este sistema de control se utiliza el método del centroide con una modificación como la expresada en [19], con lo cual las variables de salida quedan como.

$$v = \frac{\tau_{obs} \sum A_v^{obs} v_{obs} + \tau_{seg} \sum A_v^{seg} v_{seg}}{\tau_{obs} \sum A_v^{obs} + \tau_{seg} \sum A_v^{seg}} \quad (5.2)$$

$$\Delta\psi = \frac{\tau_{obs} \sum A_\psi^{obs} \Delta\psi_{obs} + \tau_{seg} \sum A_\psi^{seg} \Delta\psi_{seg}}{\tau_{obs} \sum A_\psi^{obs} + \tau_{seg} \sum A_\psi^{seg}} \quad (5.3)$$

Donde, τ_{obs} o τ_{seg} corresponde al valor del peso según el comportamiento (*obs*: evasión de obstáculos y *seg*: seguimiento de meta), A_v^{obs} , A_v^{seg} , A_ψ^{obs} , A_ψ^{seg} son el área de las funciones de pertenencia de salida correspondiente y v_{obs} , v_{seg} , $\Delta\psi_{obs}$, $\Delta\psi_{seg}$ los valores de cada comportamiento.

5.3.8. Simulación.

Para la simulación del controlador se diseña un escenario en el cual el vehículo pueda interactuar, esto se realiza con el programa *Unreal Engine 4*, la principal razón es debido a que este permite simular sensores que interactúen con el ambiente generado mediante la conexión entre el programa y Simulink.

El diseño del escenario se realizó gracias al uso de objetos ya provistos por el programa, en Fig. 5.10. se puede ver una vista cenital del escenario, así como también el punto de inicio del vehículo en la simulación marcado por un asterisco.

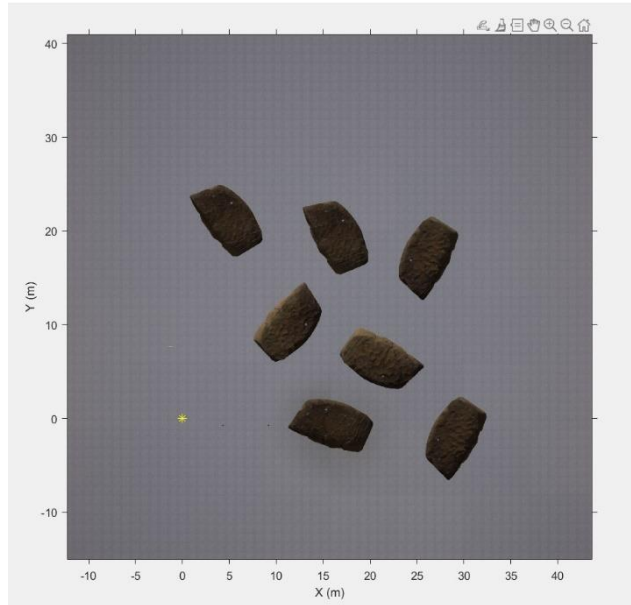


Fig. 5.11. Vista cenital de escenario diseñado.

Para la percepción de los obstáculos en la simulación se utiliza el bloque de sensor LiDAR provisto por Simulink, el cual posee un funcionamiento análogo al sonar, provee un conjunto de puntos los cuales son procesados para que la visión sea similar a este.

En la simulación se genera una meta la cual se puede ver en Fig. 5.12 y se busca que el vehículo llegue a esta.

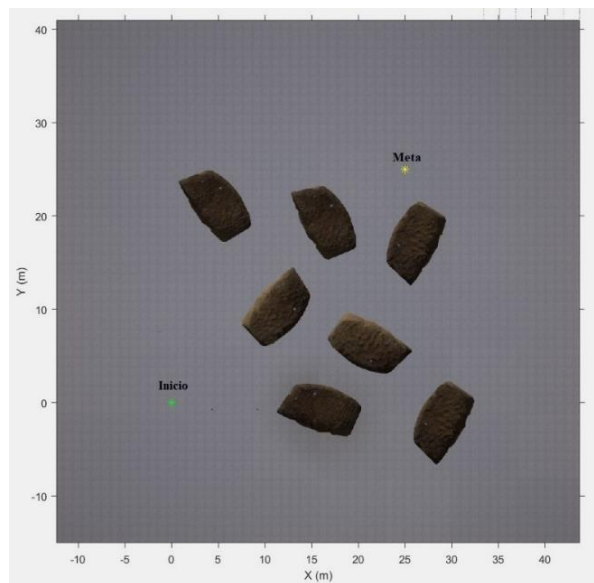


Fig. 5.12. Preparación de la simulación de navegación

En la Fig.5.13. se puede ver la trayectoria tomada por el vehículo, se observa que inicialmente se dirige a la meta propuesta en la preparación y que se desvía en función de la cercanía de los obstáculos que encuentre en su viaje.

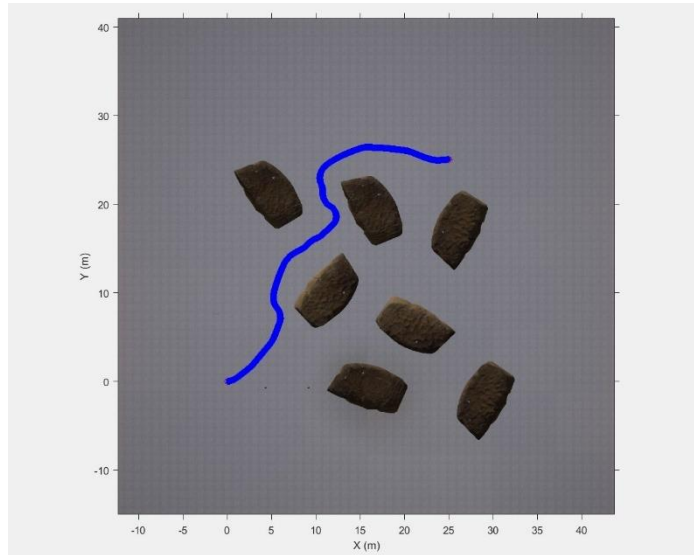


Fig. 5.13. Trayectoria del vehículo hacia la meta.

Donde, las variaciones de velocidad y orientación se muestran en Fig. 5.14 y Fig.5.15

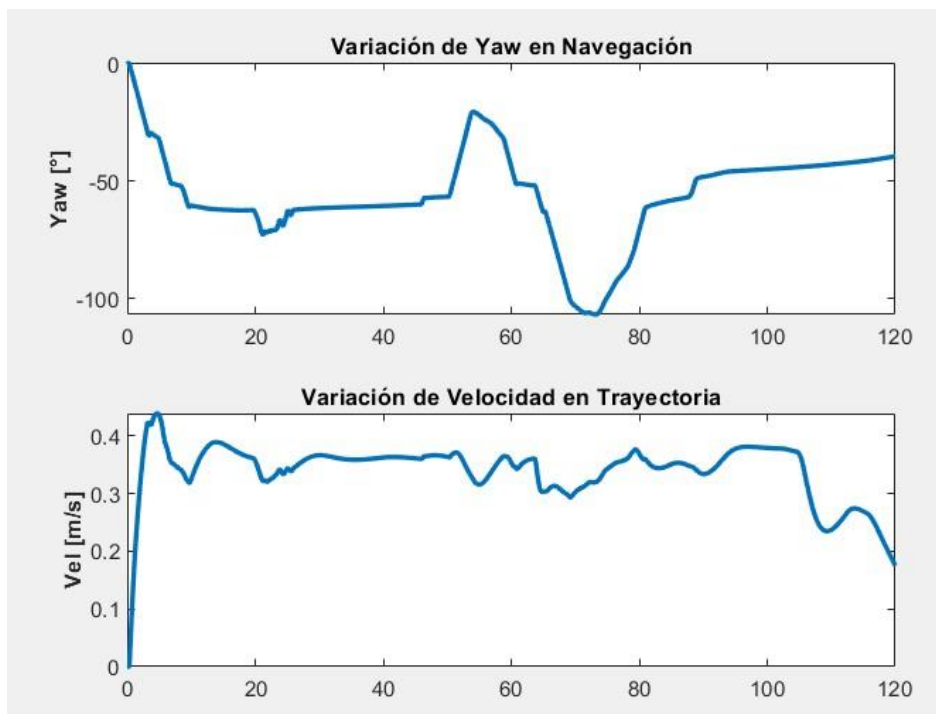


Fig.5.14 Variaciones de Yaw y Velocidad en trayectoria a meta.

6. Instrumentación

6.1.1. Introducción

En el siguiente capítulo, se presentarán los instrumentos considerados para el diseño del vehículo propuesto.

6.1.2. Propulsión

1. Propulsor:

El propulsor elegido corresponde a “*T200 Thruster*” de BlueRobotics, el motor utilizado por este es un motor sin escobillas trifásico de imán permanente, posee un diámetro de 100 [mm] y la operación nominal de este es en el rango 12-16 [V].



Fig. 6.1. Propulsor T200 de BlueRobotics.

| Alimentación | 12 [V] | 16[V] |
|--------------------------|--------------------|--------------------|
| Máxima Propulsión | 34.8 [N] | 50.0[N] |
| Max Propulsión (inv.) | 29.4 [N] | 40.2 [N] |
| Propulsión Mínima | 0.098 [N] | 0.098 [N] |
| Velocidad revolución (n) | 300-3800 [rev/min] | 300-3800 [rev/min] |
| Potencia Máxima | 200 [W] | 350 [W] |
| Corriente Máxima | 17 [A] | 25[A] |

Tabla 6.1. Características del propulsor.

1. Controlador de velocidad (ESC):

Adicional a los propulsores es necesario un controlador de velocidad que actúa como intermediario entre el microcontrolador y el propulsor, se encarga de controlar la velocidad de rotación del motor tomando de entrada un tren de pulsos PWM de 3-5 [V] generadas por el microcontrolador, para luego generar la señal requerida para hacer girar el motor a la revolución que se quiere (la cual depende del periodo de la señal PWM).



Fig. 6.2. Controlador de velocidad de propulsores.

| | |
|--------------------------------|-------------------|
| Voltaje de alimentación | 7-26 [V] (2 – 6S) |
| Corriente Máxima | 30 [A] |

Tabla 6.2. Características eléctricas ESC.

6.1.3. Percepción.

1. Ping 360 Scanning Imaging Sonar:

Sonar mecánico capaz de obtener datos alrededor de él en 360°, posee un rango máximo de 50[m] y un rango mínimo de 0.75[m], la resolución de este a 50[m] es de 4.1 [cm].



Fig. 6.3. Sonar Ping360.

| | |
|-----------------------------------|----------------------------|
| Voltaje | 11-25 [V] |
| Consumo máximo de potencia | 5 [W] |
| Protocolo de comunicación | USB, Ethernet (UDP), RS485 |

Tabla 6.3. Características Sonar Ping360.

2. Bar30, Sensor de Presión/Profundidad:

Sensor de presión ocupado para la medición de la profundidad, posee una resolución de 2 [mm].

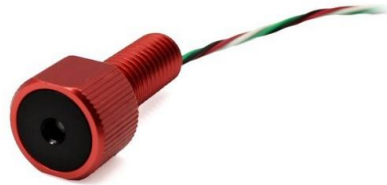


Fig. 6.4. Sensor de Presión BAR30.

| | |
|----------------------------------|---------------|
| Alimentación | 2.5 – 5.5 [V] |
| Voltaje lógico I2C | 2.5 – 3.6 [V] |
| Presión operacional | 0 – 30 [bar] |
| Profundidad de Operación | 0 – 300 [m] |
| Protocolo de Comunicación | I2C |

Tabla 6.4. Características de Sensor de Presión.

3. Seatrac Lightweight X150 y X010:

Transpondedores utilizados para la localización del vehículo, consta de dos hidrófonos, los cuales, como se había dicho en la sección 5.2.1. están ubicados en una boya y en el vehículo, para este caso el dispositivo conectado a el vehículo corresponde al hidrófono X010.



Fig. 6.5. Transpondedores X150 y X010.

| Transponedor | X150 | X010 |
|----------------------------------|--------------|-------------|
| Alimentación | 9-28 [V] | 9-28[V] |
| Consumo de Potencia | 0.6 – 6 [W] | 0.6 – 6[W] |
| Protocolo de Comunicación | RS232 | RS232 |
| Profundidad máxima | 100-2000 [m] | 300 [m] |

Tabla 6.5. Características de los transpondedores.

Vale agregar, que la alimentación necesaria para ambos transpondedores varía dependiendo de la profundidad a la que se esté midiendo, es decir, a la profundidad máxima que el sistema puede medir (en este caso 300[m]) la alimentación necesaria será de 28[V].

4. Unidad de medida inercial (IMU):

Sensor utilizado para la medida de velocidades lineales, angulares, posición y orientación. La unidad de medida inercia a utilizar será aquella que poseen el controlador de navegación y el transpondedor fijado junto al vehículo (X010).

5. Low-Light HD USB Camera:

Cámara de alta sensibilidad a la luz, de 2 MP, 1080p con la capacidad de realizar compresión de video en el momento.

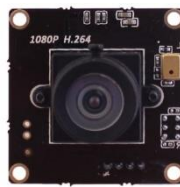


Fig. 6.6. Cámara Low-Light HD

| | |
|-----------------------------------|----------|
| Voltaje de alimentación | 5 [V] |
| Consumo máximo de potencia | 220 [mA] |

Tabla 6.6. Características eléctricas de cámara.

6.1.4. *Controlador.*

1. Controlador de Navegación

Controlador de navegación para Raspberry Pi 4, posee una unidad de medida inercial (IMU), barómetro y expansiones de puertos PWM, junto con otros para comunicación serial e I2C.



Fig. 6.7. Controlador de navegación.

| | |
|-----------------------------|--------------|
| Alimentación | 5 – 5.35 [V] |
| Consumo de corriente | 5-200 [mA] |

Tabla 6.7. Características de Controlador de navegación.

2. Microcontrolador.

Dado los instrumentos elegidos para el vehículo, el microcontrolador u ordenador a bordo a proponer para este diseño es Raspberry Pi 4.

El Raspberry Pi es una computadora de bajo costo y tamaño compacto que es capaz de correr un sistema operativo Linux, posee un conjunto de pines GPIO que le permiten controlar distintos dispositivos electrónicos.



Fig. 6.8. Microcontrolador Raspberry Pi 4.

Este microcontrolador, junto con el control de navegación, puede conectarse a los sensores propuestos en este estudio.

6.1.5. Iluminación.

Es necesaria la iluminación en este tipo de vehículos debido a la posible turbiedad u oscuridad del fondo marino que impidan la visibilidad.

1. Luces Led Lumen:



Fig. 6.9. Luces Led del vehículo.

| | |
|--|-------------------|
| Voltaje de alimentación (V_{in}) | 7 – 48 [V] |
| Corriente pico | $(15/V_{in})$ [A] |
| Brillosidad máxima | 1500 lumens |

Tabla 6.8. Características Luces led.

6.1.6. Alimentación.

1. Baterías de Ion de Litio:

Para la alimentación del vehículo se consideraron las baterías de Ion de Litio debido a su alta densidad de energía y bajo peso, el tamaño de éstas fue elegido en base a la alimentación requerida para los controladores de velocidad de cada propulsor y el voltaje de operación estos.



Fig. 6.10. Batería de Ion Litio de 14.8 [V].

Por ende, para la alimentación se consideraron baterías de Ion de Litio de 14.8 [V] o 4S y 15.6 [Ah].

| | |
|--------------------------|-----------|
| Voltaje nominal | 14.8 [V] |
| Capacidad Nominal | 15.6 [Ah] |

Tabla 6.9. Características Batería de Ion de Litio.

La potencia máxima de cada propulsor en este rango de alimentación es alrededor de 300 [W], considerando que se poseen seis de estos la cantidad aumenta a 1800 [W]. Las baterías poseen 14.8 [V] y 15.6 [Ah], lo que significa que entregará la energía total de 230.88 [Wh], considerando siete de estas se obtiene 1616.16 [Wh].

En base a lo anterior, se puede calcular el tiempo de descarga mediante,

$$T = \frac{1616.16 [Wh]}{1800 [W]} = 0.898 [h] \approx 54 [min]$$

Por lo tanto, se espera que la batería se descargue en 0.898 [h] o 54 [min], vale agregar que este cálculo se realizó considerando que los propulsores están operando a la potencia máxima necesaria en este rango de operación, además hay que considerar que este tipo de baterías se dañan al descargarse completamente, teniendo en cuenta esto y que el porcentaje de descarga para mantener la seguridad de la batería es ~40% se tiene que.

$$T_{40\%} \approx 21 [min]$$

También hay que agregar que esto se calculó considerando a los propulsores trabajando a la potencia máxima necesaria en el rango de operación, por lo que el tiempo estimado puede variar.

A continuación, se presenta una estimación del costo de los instrumentos requeridos para la construcción de la propuesta del vehículo.

| Dispositivo | Cantidad | Precio (USD) | Precio Total (USD) |
|------------------------|-----------------|-----------------------------|---------------------------|
| Propulsores | 6 | 236 | 1416 |
| ESC | 6 | 36 | 216 |
| Sonar | 1 | 2450 | 2450 |
| Cámara | 1 | 99 | 99 |
| Presión | 1 | 85 | 85 |
| Transponedores | 1 | 4200 | 4200 |
| Control de Nav. | 1 | 320 | 320 |
| RPI4 | 1 | 155 | 155 |
| Luces | 1 | 325 | 325 |
| Baterías | 7 | 330 | 2310 |
| | | Precio Total (USD) | 11576 |
| | | Precio Total + Envío | 15628 |

Tabla 6.10. Estimación de precios de los instrumentos elegidos.

Para la cantidad de horas necesarias para la implementación del proyecto referirse a Anexo E.

7. Conclusión

En este trabajo se logró diseñar, modelar y simular un vehículo autónomo submarino no tripulado junto a su sistema de navegación autónoma, donde se obtuvo una buena aproximación de los parámetros hidrodinámicos y además, el sistema de control permite al móvil moverse de manera segura hacia la meta propuesta, a continuación se presentan conclusiones con respecto al trabajo realizado.

De la obtención de parámetros físicos y modelo en Simulink:

- Resulta conveniente el uso de las plataformas CAD para el diseño de vehículos submarinos pues, si bien la simulación de flujos mediante CFD no es tan dedicada o precisa como sería el de otros programas utilizados en distintos estudios, facilita el prototipado pudiendo realizarlo de manera fácil y eficiente manteniendo una buena perspectiva de sus variables físicas.
- Si bien la computación de la matriz de masa agregada provee de una buena aproximación (ver Anexo B), la exactitud de esta depende del número de elementos de superficie que posea el modelo del diseño, lo que requiere una mayor capacidad de procesamiento para el ordenador realizando esta aproximación.

Del control de navegación e integración de programas:

- El uso de FLC designer de Matlab facilita en gran manera el proceso de diseño y prueba de un sistema de control difuso, resulta sencillo el diseño o cambio de funciones de pertenencia, reglas de control y método de defusificación.
- Resulta conveniente el uso de Unreal Engine para la simulación de vehículos o robots autónomos pues, permite generar algoritmos de procesamiento de datos en base a los sensores permitidos para una simulación con Matlab.
- Si bien, el sistema de control de navegación logra generar una trayectoria que permita al vehículo llegar de forma segura a la meta, esta no se preocupa de obtener una trayectoria óptima.
- Una de las ventajas del control de navegación es el hecho que permite complejizar el actuar del vehículo aumentando la cantidad de comportamientos de este siguiendo la misma estructura.

7.1. Trabajos Futuros.

Este trabajo presenta el diseño, modelación y simulación de un prototipo de un vehículo submarino autónomo, el paso siguiente correspondería a la implementación de este, para esto presentan algunas recomendaciones para el diseño e implementación.

- Si bien la disposición de los propulsores para el diseño propuesto genera una buena maniobrabilidad para el vehículo, esta puede ser mejorada mediante un cambio en la orientación de los propulsores, un ejemplo de esta es la configuración vectorial como la que se puede ver en el AUV de [6].
- El Controlador de Navegación propuesto en este diseño está creado para el funcionamiento con el microcontrolador Raspberry Pi 4, para la implementación completa se puede instalar el sistema operativo diseñado para facilitar la comunicación y operación entre el microcontrolador y los dispositivos que se tengan.

Referencias

- [1] Thor I. Fossen, *Guidance of Control of Ocean Vehicles*. John Wiley & Sons, 1994.
- [2] J.N. Newman., *Marine Hydrodynamics*. Cambridge, Massachusetts. The MIT Press, 2017
- [3] J. Refsnes, A. J. Sorensen and K. Y. Pettersen, "A 6 DOF nonlinear observer for auvs with experimental results," *2007 Mediterranean Conference on Control & Automation*, Athens, Greece, 2007, pp. 1-7.
- [4] Tuveri, M., Ceruti, A., & Marzocca, P. "Added masses computation for unconventional airships and aerostats through geometric shape evaluation and meshing". *International Journal of Aeronautical and Space Sciences*, vol 15, no. 3, pp. 241-257, Sep. 2014.
- [5] Giovanni Carbone, Guillaume Martinat, Dominique Farcy, Jean-Luc Harion. "Added Masses of generic shape bodies interacting with external walls". *Aerospace Science and Technology*, vol. 90, pp. 70 – 84, Jul. 2019.
- [6] Sulin Tang, Tamaki Ura, Takeshi Nakatani, Blair Thornton, Tao Jiang, "Estimation of the hydrodynamic coefficients of the complex-shaped autonomous underwater vehicle TUNA-SAND", *Journal of Marine Science and Technology*, vol. 14, pp 373-386, Sep. 2009, doi: <https://doi.org/10.1007/s00773-009-0055-4>.
- [7] Jyoti Prakash Panda, Arindam Mitra and Hari V Warrior, "A review on the hydrodynamic characteristics of autonomous underwater vehicles". *Proceedings of the institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, vol. 14, no. 1, pp 15-29, Jul. 2020.
- [8] What is CFD, <https://www.simscale.com/docs/simwiki/cfd-computational-fluid-dynamics/what-is-cfd-computational-fluid-dynamics/>.
- [9] Robert Christ, Robert Wernli, *The ROV Manual* 1st ed. England: Butterworth-Heinemann, 2007.
- [10] Kai Sun, Weicheng Cui, Chi Chen, "Review of Underwater Sensing Technologies and Applications", *Sensors*, vol. 21, no. 23, p. 7849, Nov. 2021, doi: 10.3390/s21237849.
- [11] A. Ridolfi *et al.*, "FeelHippo: A low-cost autonomous underwater vehicle for subsea monitoring and inspection," *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*, Florence, Italy, 2016, pp. 1-6, doi: 10.1109/EEEIC.2016.7555607.

- [12] Juan A. Ramirez-Macías, Persijn Brongers, Santiago Rúa, Rafael E. Vásquez, “Hydrodynamic modelling for the remotely operated vehicle Viseor3 using CFD”, *IFAC-PapersOnLine*, vol. 49, no. 23, pp 187-192, Oct. 2016.
- [13] M. B. Alatise and G. P. Hancke, "A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods," in *_IEEE Access_*, vol. 8, pp. 39830-39846, 2020, doi: 10.1109/ACCESS.2020.2975643.
- [14] Bo He, Yan Liang, Xiao Feng, Rui Nian, Tianhong Yan, Minghui Li, Shujing Zhang, “AUV SLAM and Experiments Using a Mechanical Scanning Forward-Looking Sonar”, *Sensors*, vol. 12, no. 7, pp 9386-9410, Jul. 2012, doi: 10.3390/s120709386.
- [15] J. Jung, Y. Lee, D. Kim, D. Lee, H. Myung and H. -T. Choi, "AUV SLAM using forward/downward looking cameras and artificial landmarks," *2017 IEEE Underwater Technology (UT)*, Busan, Korea (South), 2017, pp. 1-3, doi: 10.1109/UT.2017.7890307.
- [16] L. Paull, S. Saeedi, M. Seto and H. Li, "AUV Navigation and Localization: A Review," in *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 131-149, Jan. 2014, doi: 10.1109/JOE.2013.2278891.
- [17] Saffiotti A., “The uses of fuzzy logic in autonomous robot navigation”, *Soft Computing* , vol 1., pp 180-197, Dec. 1997, doi: <https://doi.org/10.1007/s005000050020>.
- [18] Edward Tunstel Jr., Tanya Lippincott, Mo Jamshidi, “Behavior Hierarchy for Autonomous Mobile Robots: Fuzzy-Behavior Modulation and Evolution”, in *Intelligent Automation & Soft Computing*, vol. 3, no. 1, pp 37-49, doi: 10.1080/10798587.1997.10750691.
- [19] H. Seraji and A. Howard, "Behavior-based robot navigation on challenging terrain: A fuzzy logic approach," in *IEEE Transactions on Robotics and Automation*, vol. 18, no. 3, pp. 308-321, June 2002, doi: 10.1109/TRA.2002.1019461.

8. Anexos

8.1. Anexo A. Diagrama Simulink

El modelo Simulink consta de tres partes principales: Control de Navegación, Modelo y Control y Visualización. El primero consiste en el sistema de control propuesto en el capítulo 5, en el segundo se encuentra el modelo del vehículo junto al sistema de regulación de posición, altura y velocidad descrito en el capítulo 4, por último, en la visualización se encuentra el enlace entre Simulink y Unreal Engine 4, Junto con la simulación de los sensores y obtención de datos.

1. Modelo y Control:

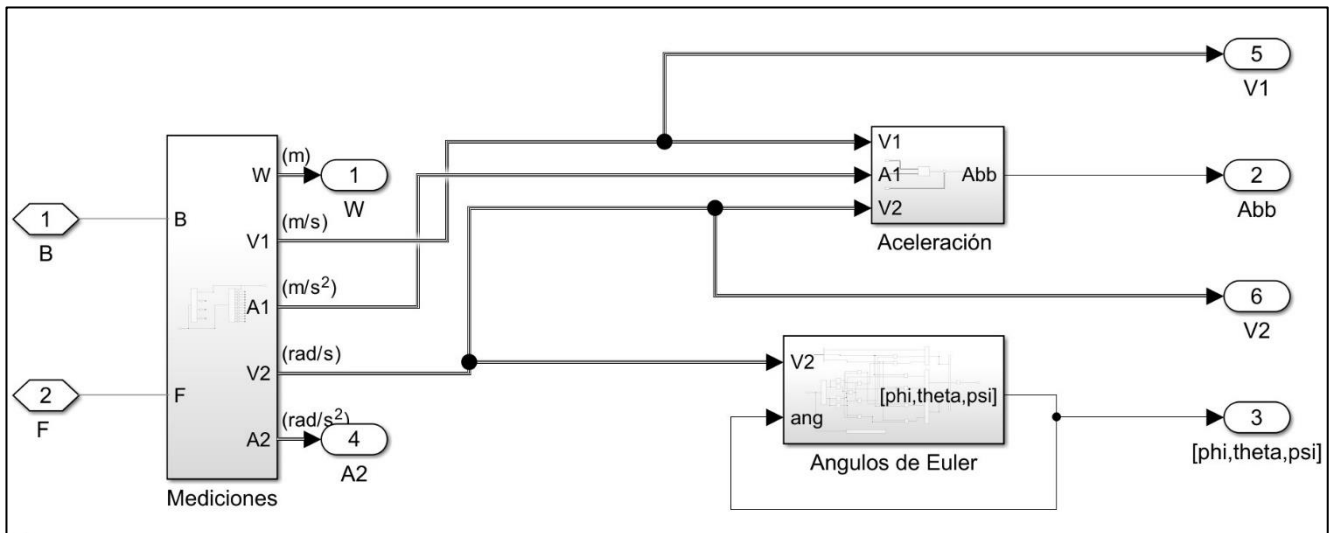


Fig. A.1. Bloque de Mediciones del vehículo

En Fig. A.1. Se pueden ver las mediciones de Velocidad (lineal y angular), Aceleración (lineal y angular), Posición en el eje cartesiano (x, y, z) y su orientación (ϕ, θ, ψ), estas están hechas de acuerdo con el modelo descrito en el Capítulo 2 en base a los dos sistemas de referencia Ref_t y Ref_c .

Para los ángulos de Euler y la Aceleración se requieren otras derivaciones las cuales se pueden ver en las siguientes figuras.

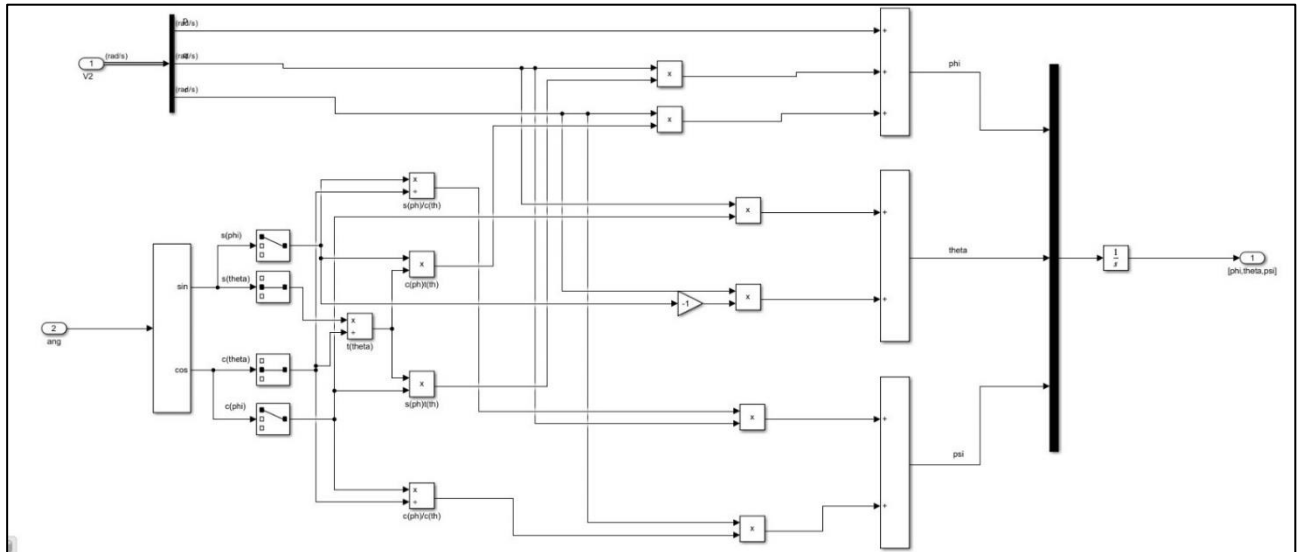


Fig. A.2. Bloque de derivación de Ángulos de Euler.

En la derivación de ángulos de Euler se hizo uso de la ecuación (2.8) y (2.6) pues si se reemplaza (2.6) en (2.8) se puede obtener,

$$\eta_2 = J_2(\eta_2)v_2 \quad (A.1)$$

A continuación, se presenta la implementación de las fuerzas hidrodinámicas en Simulink,

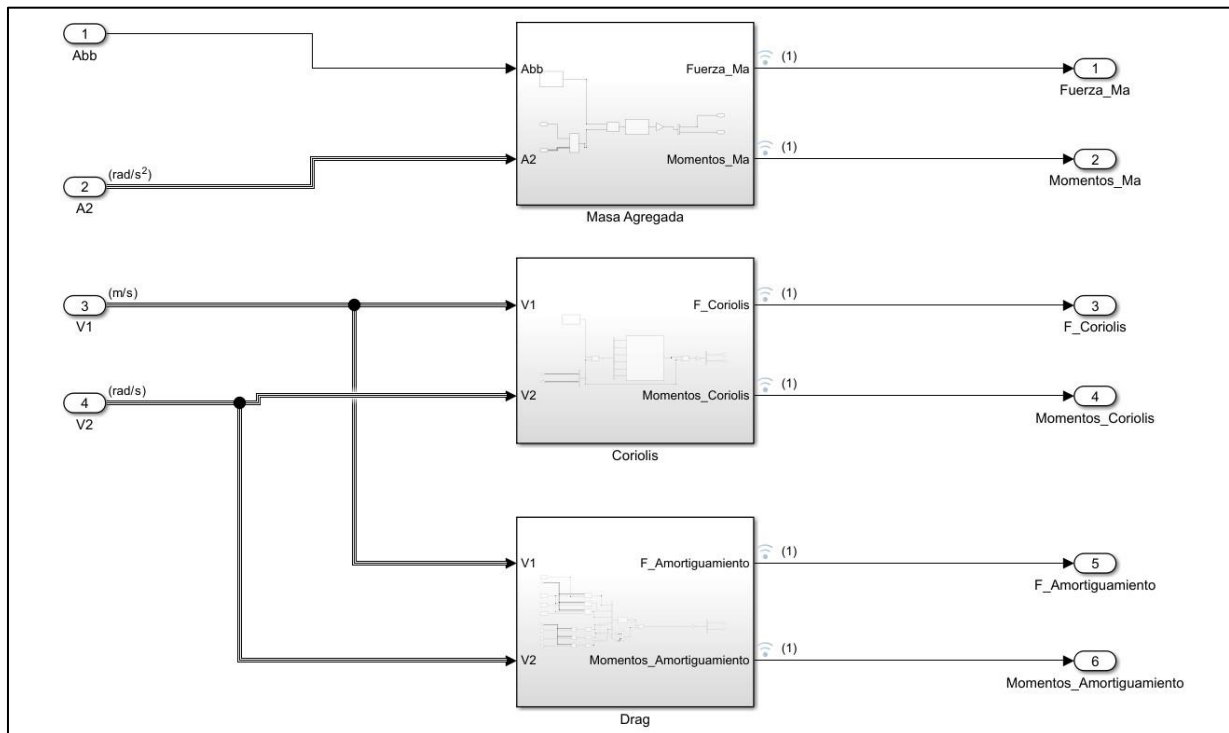


Fig. A.3. Implementación de Fuerzas hidrodinámicas.

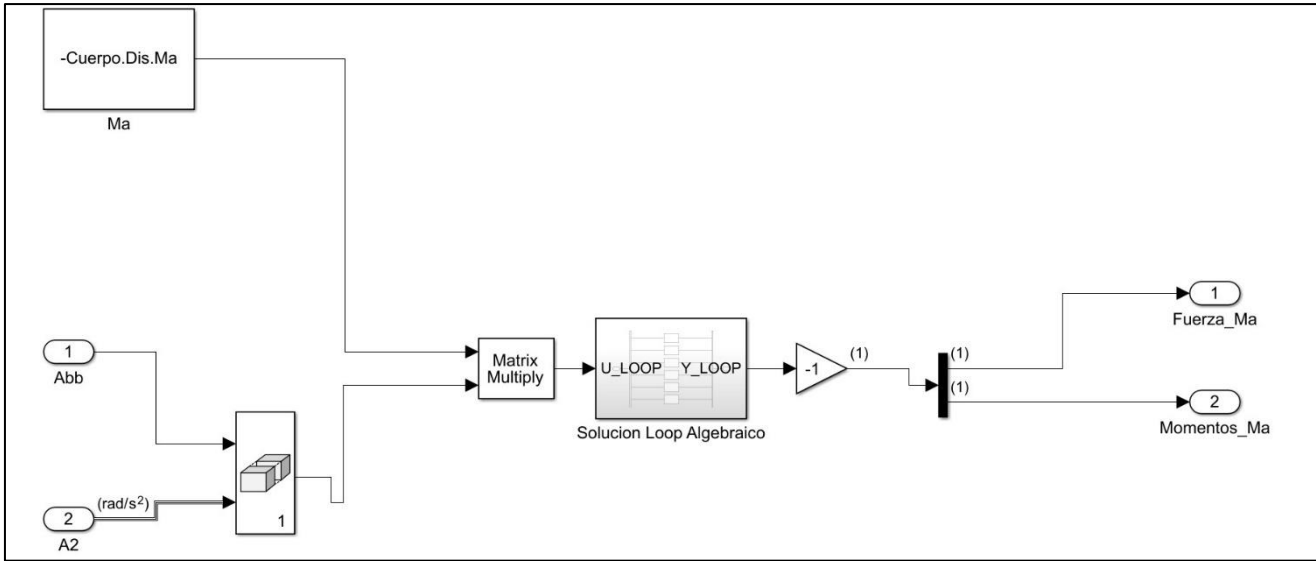


Fig. A.3. Implementación de Fuerzas dadas por la masa agregada.

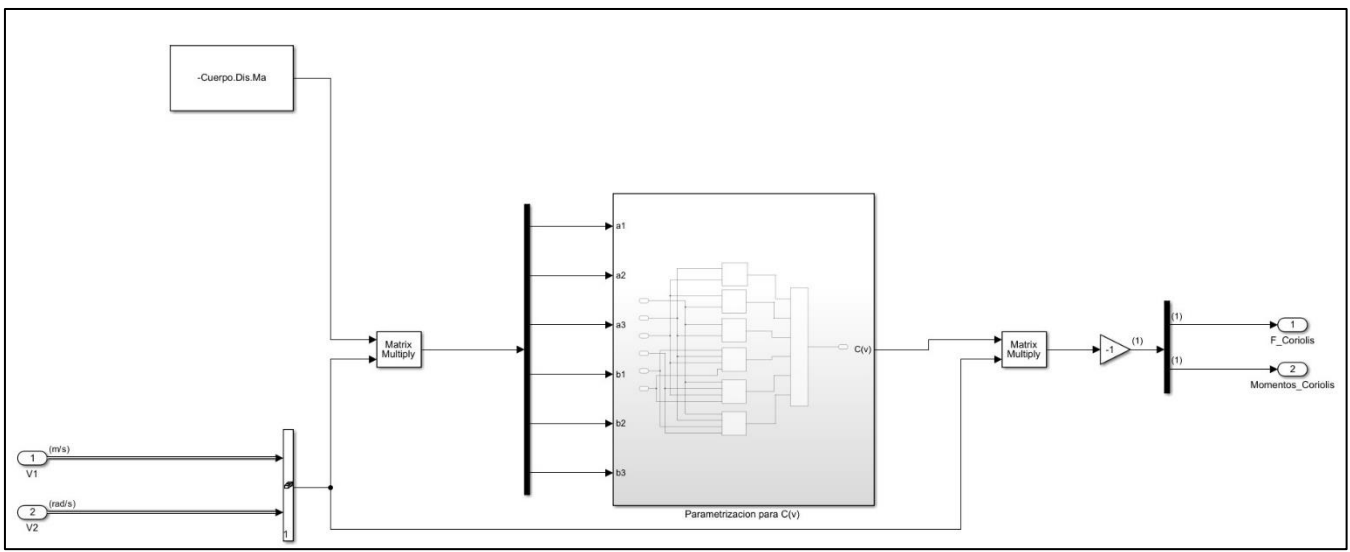


Fig. A.4. Implementación de Fuerzas Coriolis-Centrípetas.

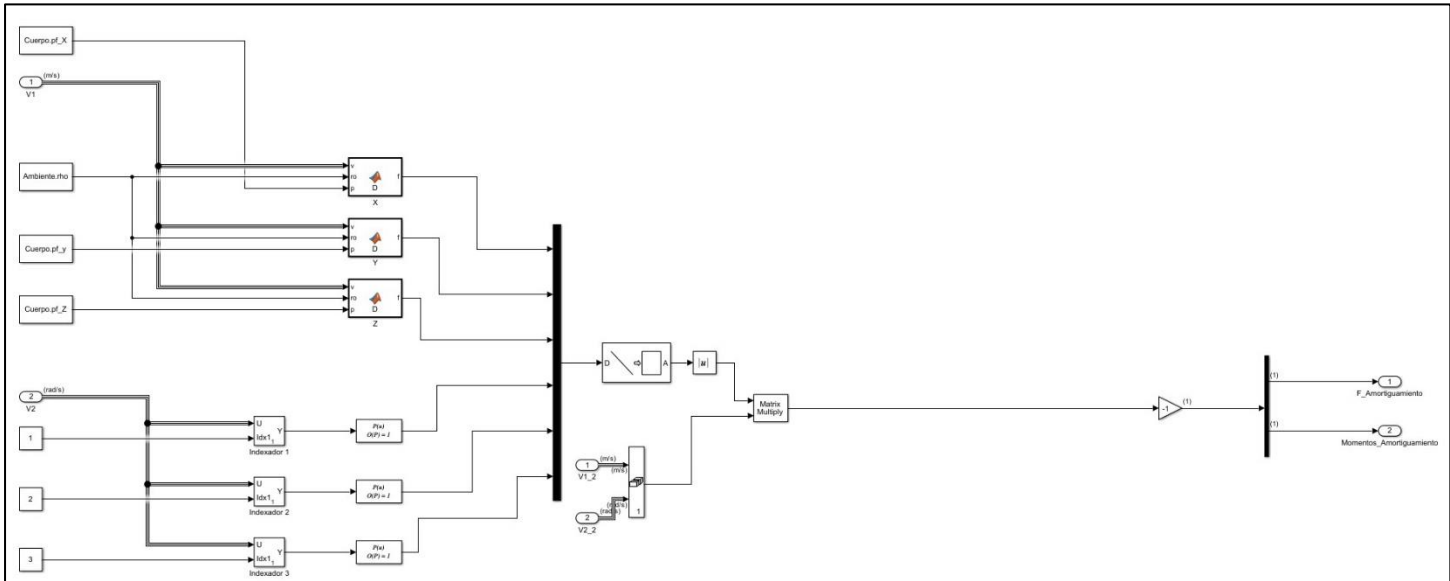


Fig. A.5. Implementación de Fuerzas de Amortiguamiento.

Para las fuerzas hidrostáticas se debe agregar la fuerza de empuje del vehículo que actúa sobre el centro de flotabilidad de este.

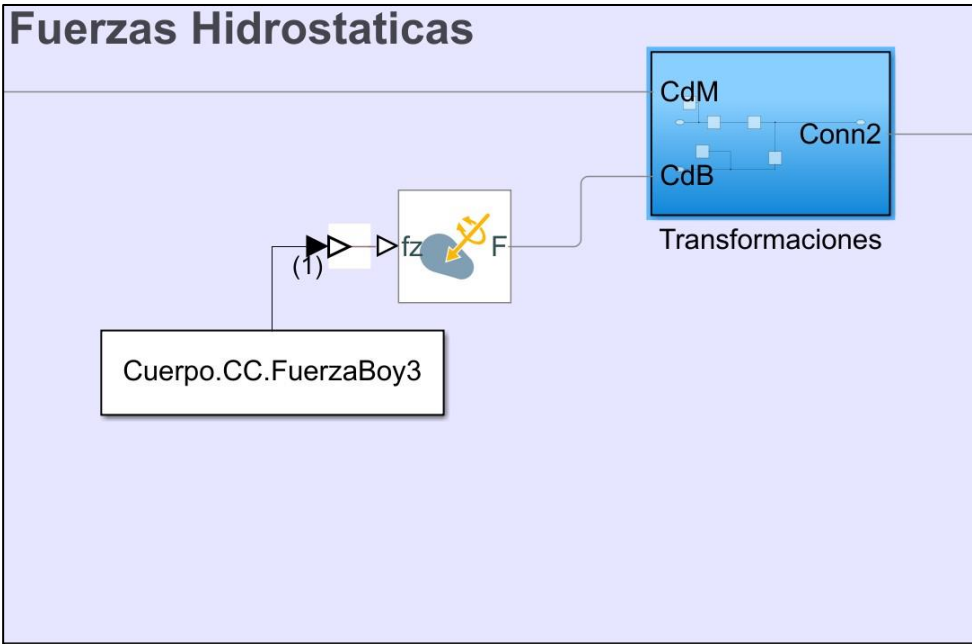


Fig. A.6. Implementación de la fuerza de empuje al vehículo.

El efecto de las Fuerzas mostradas anteriormente y las mediciones necesarias deben actuar sobre el diseño CAD del vehículo. A continuación, se muestra la implementación en Simulink del modelo CAD realizado en SolidWorks.

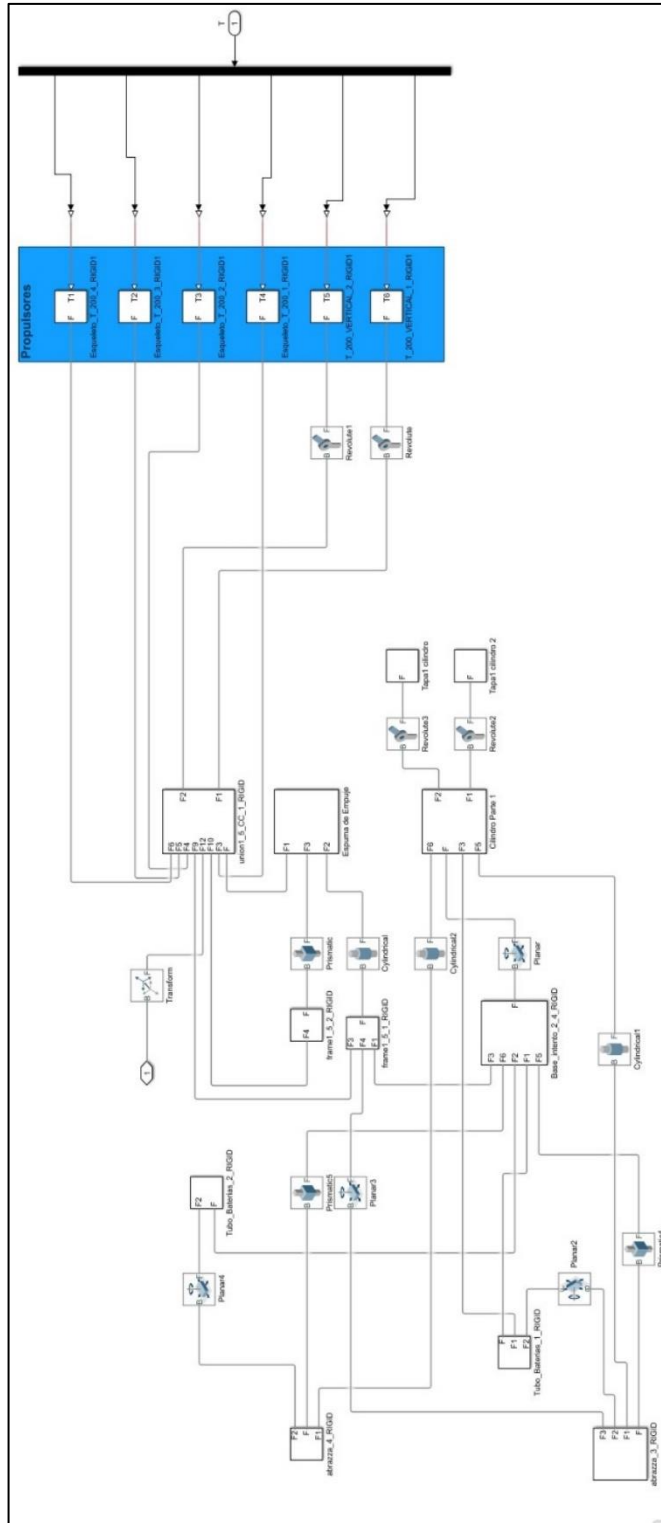


Fig. A.7. Diagrama Simulink del modelo CAD del vehículo

Dentro de esta implementación, se encuentra la integración de las piezas modeladas en Simulink con sus respectivas interacciones, las cuales se ven representadas por bloques como “*Revolute*”, “*Prismatic*”, “*Cylindrical*” y “*Planar*”.

En Fig. A.7. se pueden ver los propulsores del vehículo encerrados en un recuadro color celeste, para la implementación de la fuerza que genera cada uno de estos se utiliza el bloque “*External Force and Torque*” que actúa sobre el bloque “*Revolute Joint*”.

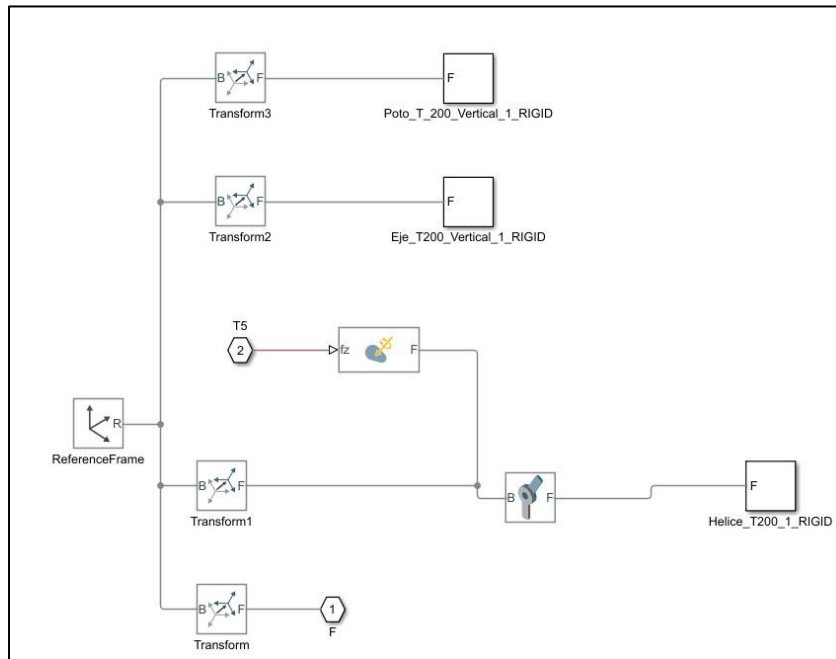


Fig. A.8. Bloque de Propulsores.

Para el sistema de regulación de velocidad v , ψ , θ y profundidad, se utilizaron los diagramas ilustrados en Fig. A.9 y Fig. A.10.

En Fig.A.9. se puede ver el diagrama de control de v y ψ , para la regulación de ψ se consta de una estrategia de control cascada donde, el controlador externo ajusta el valor de la velocidad de rotación r sobre el eje z . Se hace uso de limitadores para restringir valores excesivos de fuerza de propulsión.

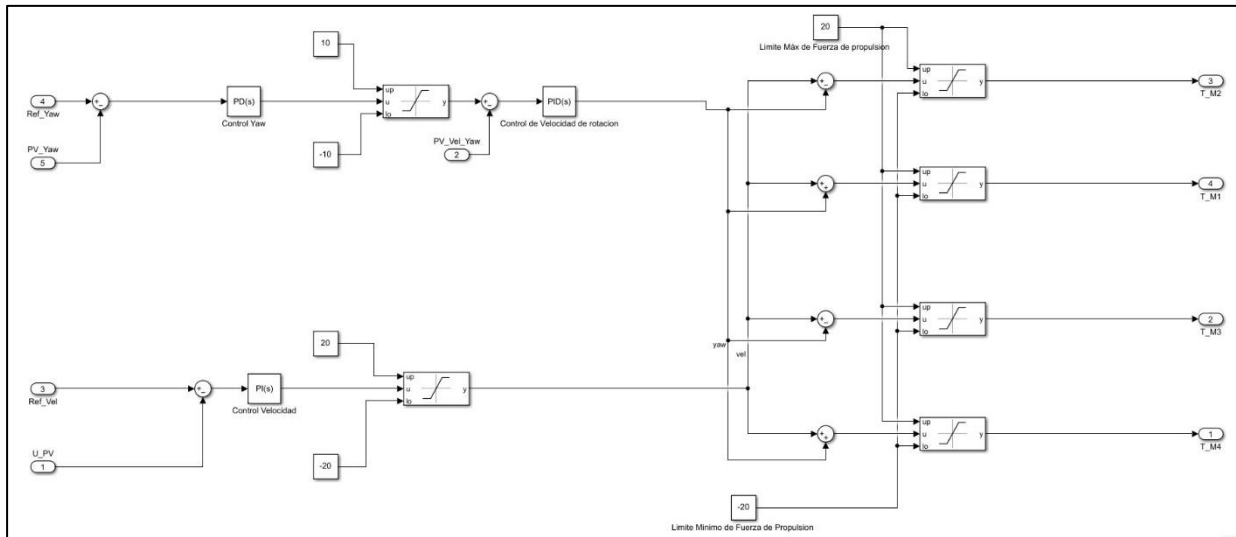


Fig. A.9. Diagrama de Control para ψ y v .

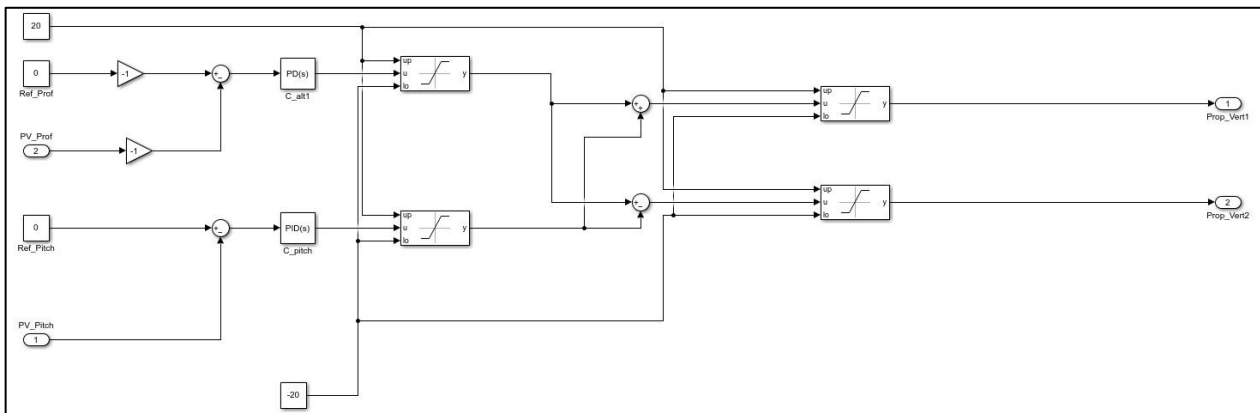


Fig. A.10. Diagrama de Control para θ y Profundidad

2. Control de Navegación.

Para la implementación en Simulink del controlador difuso, se hará uso del bloque “*Fuzzy Logic Controller*”, en donde se requiere acceder a las variables de entrada fusificadas y agregadas, esto es luego de la etapa de composición de las entradas, para esto se activa la opción “*Aggregated output (ao)*” del bloque del controlador difuso. Luego, se defusifican mediante el uso del bloque función encargado de realizar esta tarea.

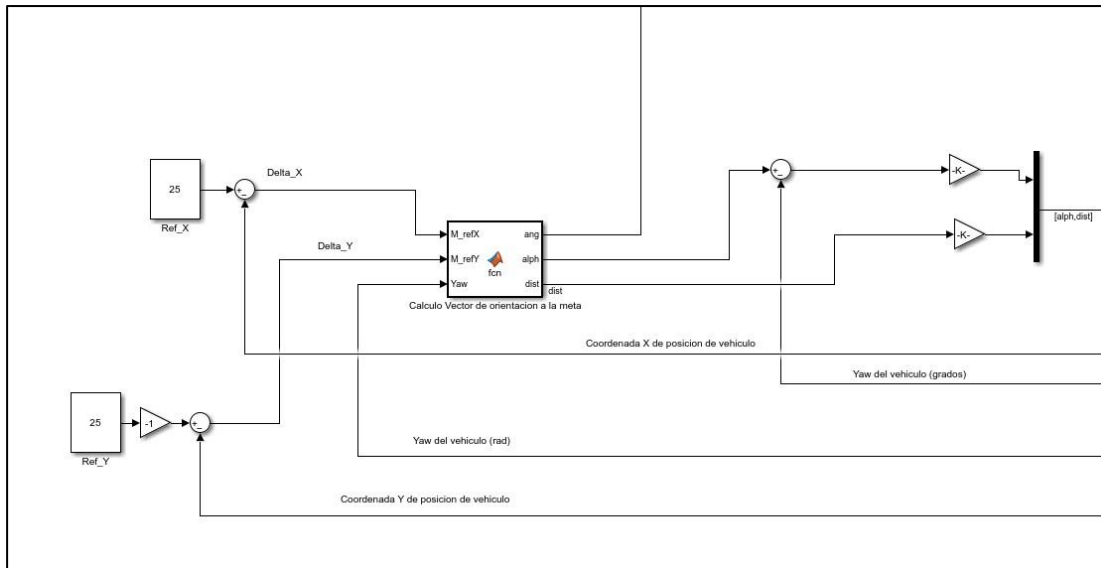


Fig. A.11. Entrada a Controlador difuso encargado del seguimiento de meta.

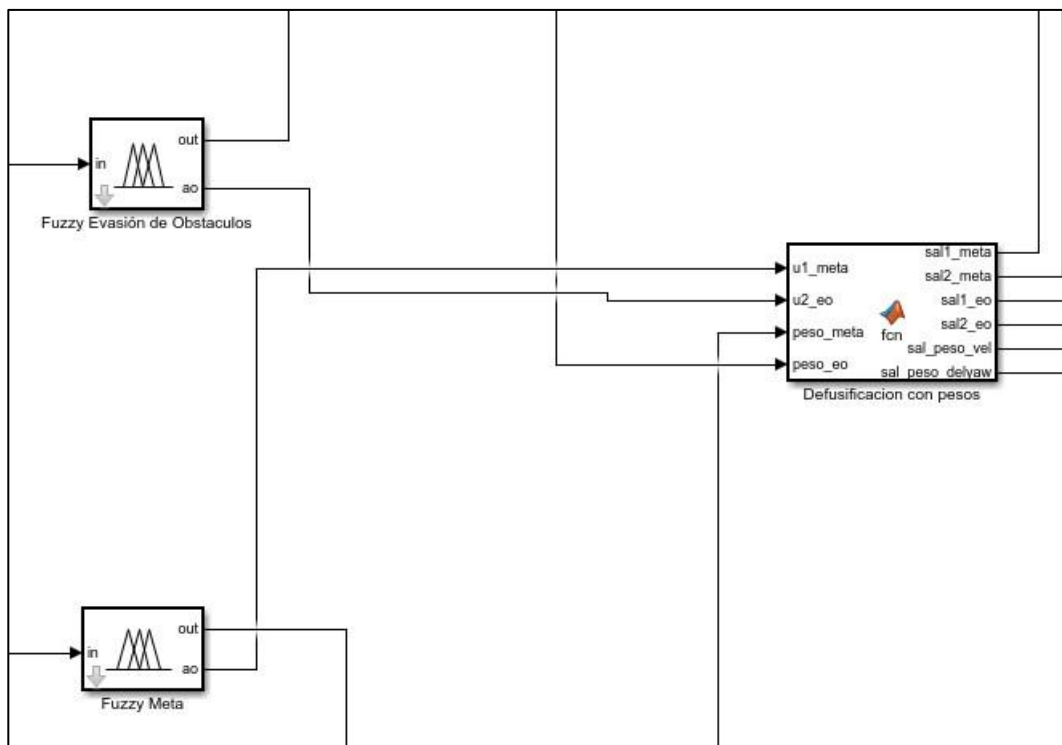


Fig. A.12. Conexión entre Comportamientos y Bloque de defusificación.

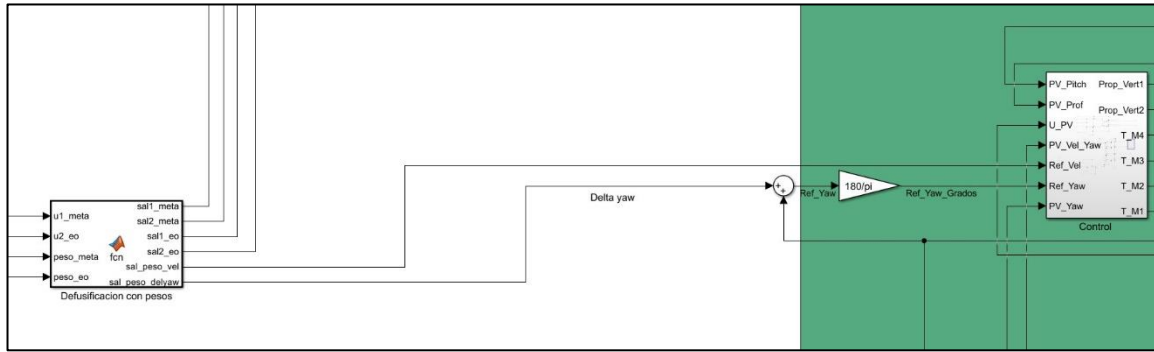


Fig. A.13. Salida del Controlador de Navegación.

3. Visualización.

En esta sección del diagrama completo se implementa el enlace entre el ambiente creado en *Unreal Engine 4* y Simulink, para esto, se utilizan los bloques: “*Simulation 3D Scene Configuration*”, “*Simulation 3D Actor Transform Set*” y “*Simulation 3D Lidar*”. El primero se encarga del enlace entre el editor de *Unreal Engine* y Simulink, el segundo, cumple la función de trasladar los ejes de referencia y sus interacciones a los del editor de *Unreal Engine*, por último, “*Simulation 3D Lidar*” genera la simulación del sensor a ocupar.

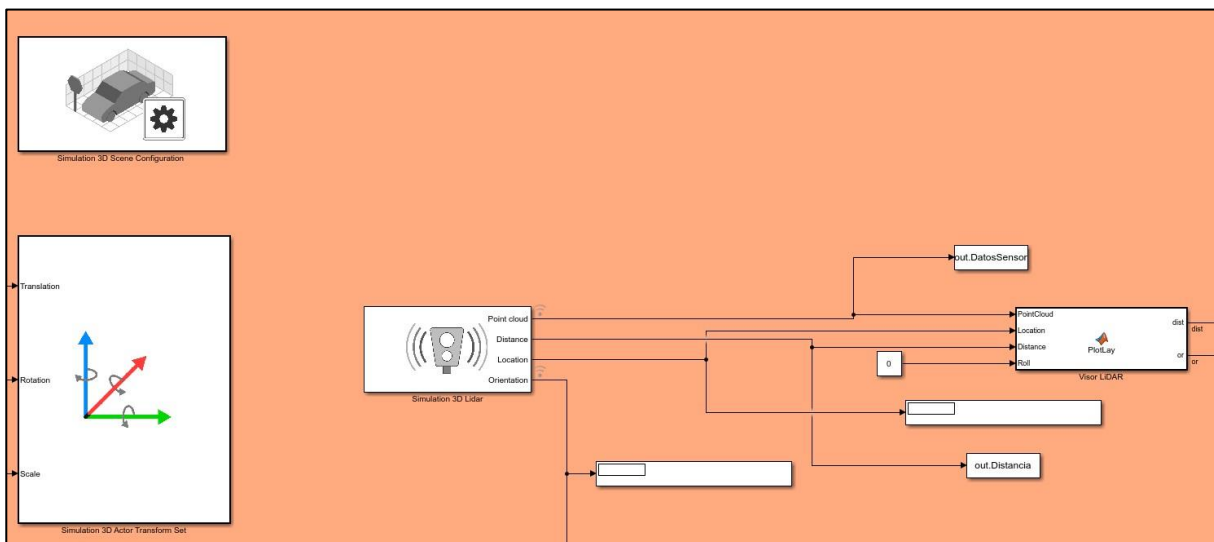


Fig. A.14. Bloques utilizados para el enlace con Unreal Engine 4.

8.2. Anexo B. Códigos para la obtención de masa agregada.

En el siguiente anexo se presentarán los códigos creados para la obtención del valor de la matriz de masa agregada M_a , los cuales son: “Código principal”, “Subdivisión de elementos de superficie” y “Código de creación de objeto polígono”. En el primero se cargan los datos del archivo de extensión .stl y se realizan los cálculos necesarios para la obtención de la matriz, dentro de este, es necesaria la subdivisión de los elementos de la superficie, para esto se hace uso del segundo código, el cual, es una función que entrega el área, vectores normales e incentro de las subdivisiones, para facilitar este proceso, se crea una clase polígono que se encarga de realizar los cálculos necesarios dependiendo de la cantidad de lados que posea. Finalmente, se realiza la comparación de los resultados con el valor teórico de una esfera de radio $r = 0.25$ [m].

8.2.1. Código Principal.

```
%Carga de archivo stl y obtencion de datos
Nombre = "ejemplos solidworks/Diseño1_copia_parte_menos_tr_stl.stl";
TR = stlread(Nombre);
pt = TR.Points; ct = TR.ConnectivityList;
ctt = ct.';

pt2 = pt/1000;
TR2 = triangulation(ct,pt2);

%.....

centr = sum(pt)/length(pt);

%.....

IIN = incenter(TR2); VecNor = faceNormal(TR2);
C = zeros(length(ct),length(ct));
B = zeros(length(ct),length(ct));
n_subdiv = 42;n_uf = length(ct);
M =12; prp = cell(n_uf,3); ptt1 = pt(ctt(1:3*n_uf),:);
cont = 3*ones(1,n_uf); ptt = mat2cell(ptt1,cont); dS = zeros(length(ct),1);
VN = VecNor + IIN;
%.....Centrado a el cg.....

ptt2 = ptt1 - centr; %origen en el centro geometrico
ptt2 = ptt2/1000; %Cambio de escala a [m]
ptcell = mat2cell(ptt2,cont);

%cada 3 elementos se suma.....
[n,col] = size(ptt2);
index = 1:n;
elem = [repmat(3,1,floor(n/3))];
endv = n-sum(elem);
```

```

if(~endv)
    endv = [];
end
index = mat2cell(index,1,[elem,endv])';
inc = cell2mat(cellfun(@(x) sum(ptt2(x,:),1),index,'un',0));
incentr = inc*(1/3);

%.....Generación de matriz d(dx,dy,dz).....

d = incentr; % - centr ---> depende de donde está el origen.

%subdivisión para cada elemento de
superficie.....
tic
for i=1:n_uf
    ppt = ptcell{i};
    [NN,Arr,INN] = optsubdivmas(ppt);
    f = poligono(ppt(1,:),ppt(2,:),ppt(3,:)); %objeto poligono
    dS(i) = f.area;
    prp{i,1}=NN;prp{i,2}=Arr;prp{i,3}=INN;
end
toc

%.....
%Calculo de C y B.....

tic
parfor i=1:n_uf
    p1 = ptcell{i};
    t_1 =poligono(p1(1,:),p1(2,:),p1(3,:));
    in1 =t_1.incentro;
    for j=1:n_uf
        p2 = ptcell{j}; %#ok<PFBNS>
        Nj=prp{j,1}; ARj=prp{j,2}; INj=prp{j,3};%#ok<PFBNS>
        N_og = VecNor(j,:); %#ok<PFBNS>
        Csum = 0;
        Bsum = 0;
        for k=1:n_subdiv
            rijk = INj(k,:) - in1; rn = norm(rijk);
            Sjk = ARj(k);
            Cijk = -(dot(rijk,N_og)/rn^3)*Sjk;
            Bijk = (1/rn)*Sjk;
            Csum = Csum + Cijk;
            Bsum = Bsum + Bijk;
        end
        C(i,j) = Csum;
        B(i,j) = Bsum;
    end
display((i/n_uf)*100);
end

%.....
iden = eye(n_uf);
temp = (2*pi*iden) - C;
Vx = VecNor(:,1); Vy = VecNor(:,2) ; Vz = VecNor(:,3);
dx = d(:,1); dy = d(:,2); dz = d(:,3);

```

```

Mag = sqrt(sum(VecNor.^2,2));
%.....Valores d(phi)/dn.....
alpha = Vx./Mag ; beta = Vy./Mag ; gamma = Vz./Mag;
uno = (dy.*gamma) - (dz.*beta); duo =(dz.*alpha) - (dx.*gamma); tres=(dx.*beta)-
(dy.*alpha);
%.....
K1 = B*alpha; K3 = B*gamma; K5 = B*duo;
K2 = B*beta; K4 = B*uno; K6 = B*tres;
%.....Valores de Phi.....
phi1 = temp\K1; phi4 = temp\K4;
phi2 = temp\K2; phi5 = temp\K5;
phi3 = temp\K3; phi6 = temp\K6;
%.....Obtención de Ma.....
Ma_temp = zeros(6,6);
Ma_temp(1,1:6) = [sum(phi1.*alpha.*dS), sum(phi1.*beta.*dS), sum(phi1.*gamma.*dS),
sum(phi1.*uno.*dS), sum(phi1.*duo.*dS), sum(phi1.*tres.*dS)];
Ma_temp(2,1:6) = [sum(phi2.*alpha.*dS), sum(phi2.*beta.*dS), sum(phi2.*gamma.*dS),
sum(phi2.*uno.*dS), sum(phi2.*duo.*dS), sum(phi2.*tres.*dS)];
Ma_temp(3,1:6) = [sum(phi3.*alpha.*dS), sum(phi3.*beta.*dS), sum(phi3.*gamma.*dS),
sum(phi3.*uno.*dS), sum(phi3.*duo.*dS), sum(phi3.*tres.*dS)];
Ma_temp(4,1:6) = [sum(phi4.*alpha.*dS), sum(phi4.*beta.*dS), sum(phi4.*gamma.*dS),
sum(phi4.*uno.*dS), sum(phi4.*duo.*dS), sum(phi4.*tres.*dS)];
Ma_temp(5,1:6) = [sum(phi5.*alpha.*dS), sum(phi5.*beta.*dS), sum(phi5.*gamma.*dS),
sum(phi5.*uno.*dS), sum(phi5.*duo.*dS), sum(phi5.*tres.*dS)];
Ma_temp(6,1:6) = [sum(phi6.*alpha.*dS), sum(phi6.*beta.*dS), sum(phi6.*gamma.*dS),
sum(phi6.*uno.*dS), sum(phi6.*duo.*dS), sum(phi6.*tres.*dS)];
%.....
%Matriz
Ma = -rho_m*Ma_temp;
toc

```

8.2.2. Subdivisión de elementos de superficie.

```

function [normal,area,incentro] = optsubdivmas(tr)
%Vectores normales a elementos de superficies se dan de la variable F
%Los valores x,y,z son las coordenadas del vector r de distancia entre dos superficies

punts = tr;
in = 1/3*(sum(punts,1));
set = [punts;in];

%Inicio subdivisión
vert = [punts(1,:);punts(2,:);punts(3,:)];
ult = [punts(3,:);punts(3,:);punts(1,:)];
lad = [punts(2,:) - punts(3,:);punts(1,:) - punts(3,:);punts(2,:) - punts(1,:)];
M_og = [punts(1,:) - in;punts(2,:) - in;punts(3,:) - in];
n1 = 7;
lad = lad*(1/2); % subdivisiones lados del triangulo (AB,BC,CA)

k = 1:n1;
k = repmat(k,3,1);
k = reshape(k,[],1);
k = 1/n1*k;

```



```

M_og = repmat(M_og,n1,1);
M_old = M_og.*k;
M_subdiv = in + M_old; %subdivisiones triangulos interiores del elemento
lad_r = lad + ult; %subdivisiones lados del triangulo (AB,BC,CA)
M_off = [set;M_subdiv;lad_r];

% obtención de puntos faltantes
vec_in = lad_r - in; l_vecin = length(vec_in);
k = [1:n1]*(1/n1); k = k';
vec_in = repmat(vec_in,length(k),1);
k = kron(k,ones(l_vecin,1));
vec_r = vec_in.*k;
vec_r = vec_r + in;
%fin de obtención de puntos faltantes

%Todos los puntos de la subdivisión
M_ultra_off = [M_off;vec_r];
dist_vec = M_ultra_off - in; dist_og = sqrt(sum((dist_vec).^2,2));
dist = sort(sqrt(sum((dist_vec).^2,2)));
dist(1,:) = [];
dist1 = dist(1:6,:); dist2 = dist(7:12,:);dist3 = dist(13:18,:);
dist4 = dist(19:21,:); dist5 = dist(22:24,:); dist6 = dist(25:30,:);
tol = 1e-6;

%Dividirlo en grupos o tandas
grupo1 = [M_subdiv(1:3,:);vec_r(1:3,:)] ;
grupo2 = [M_subdiv(4:6,:);vec_r(4:6,:)] ;
grupo3 = [M_subdiv(7:9,:);vec_r(7:9,:)] ;
grupo4 = [M_subdiv(10:12,:);vec_r(10:12,:)] ;
grupo5 = [M_subdiv(13:15,:);vec_r(13:15,:)] ;
grupo6 = [M_subdiv(16:18,:);vec_r(16:18,:)] ;
grupo7 = [set(1:3,:);lad_r] ;

%Dividirlo en formas ,(calcular ortocentro y area)
%Definición Formas
%Triangulos
tr1_f = poligono(in,grupo1(1,:),grupo1(6,:));
tr2_f = poligono(in,grupo1(6,:),grupo1(2,:));
tr3_f = poligono(in,grupo1(2,:),grupo1(4,:));
tr4_f = poligono(in,grupo1(4,:),grupo1(3,:));
tr5_f = poligono(in,grupo1(3,:),grupo1(5,:));
tr6_f = poligono(in,grupo1(5,:),grupo1(1,:));
%Trapezoides
trap1_f =poligono(grupo1(1,:),grupo1(5,:),grupo2(5,:),grupo2(1,:));
trap2_f = poligono(grupo1(1,:),grupo2(1,:),grupo2(6,:),grupo1(6,:));
trap3_f = poligono(grupo1(6,:),grupo2(6,:),grupo2(2,:),grupo1(2,:));
trap4_f = poligono(grupo1(2,:),grupo2(2,:),grupo2(4,:),grupo1(4,:));
trap5_f = poligono(grupo1(4,:),grupo2(4,:),grupo2(3,:),grupo1(3,:));
trap6_f = poligono(grupo1(3,:),grupo2(3,:),grupo2(5,:),grupo1(5,:));
trap7_f = poligono(grupo2(1,:),grupo2(5,:),grupo3(5,:),grupo3(1,:));
trap8_f = poligono(grupo2(1,:),grupo3(1,:),grupo3(6,:),grupo2(6,:));
trap9_f = poligono(grupo2(6,:),grupo3(6,:),grupo3(2,:),grupo2(2,:));
trap10_f= poligono(grupo2(2,:),grupo3(2,:),grupo3(4,:),grupo2(4,:));
trap11_f = poligono(grupo2(4,:),grupo3(4,:),grupo3(3,:),grupo2(3,:));
trap12_f= poligono(grupo2(3,:),grupo3(3,:),grupo3(5,:),grupo2(5,:));
trap13_f = poligono(grupo3(1,:),grupo3(5,:),grupo4(5,:),grupo4(1,:));

```

```

trap14_f= poligono(grupo3(1,:),grupo4(1,:),grupo4(6,:),grupo3(6,:));
trap15_f = poligono(grupo3(6,:),grupo4(6,:),grupo4(2,:),grupo3(2,:));
trap16_f= poligono(grupo3(2,:),grupo4(2,:),grupo4(4,:),grupo3(4,:));
trap17_f = poligono(grupo3(4,:),grupo4(4,:),grupo4(3,:),grupo3(3,:));
trap18_f= poligono(grupo3(3,:),grupo4(3,:),grupo4(5,:),grupo3(5,:));
trap19_f = poligono(grupo4(1,:),grupo4(5,:),grupo5(5,:),grupo5(1,:));
trap20_f= poligono(grupo4(1,:),grupo5(1,:),grupo5(6,:),grupo4(6,:));
trap21_f = poligono(grupo4(6,:),grupo5(6,:),grupo5(2,:),grupo4(2,:));
trap22_f= poligono(grupo4(2,:),grupo5(2,:),grupo5(4,:),grupo4(4,:));
trap23_f = poligono(grupo4(4,:),grupo5(4,:),grupo5(3,:),grupo4(3,:));
trap24_f= poligono(grupo4(3,:),grupo5(3,:),grupo5(5,:),grupo4(5,:));
trap25_f = poligono(grupo5(1,:),grupo5(5,:),grupo6(5,:),grupo6(1,:));
trap26_f= poligono(grupo5(1,:),grupo6(1,:),grupo6(6,:),grupo5(6,:));
trap27_f = poligono(grupo5(6,:),grupo6(6,:),grupo6(2,:),grupo5(2,:));
trap28_f= poligono(grupo5(2,:),grupo6(2,:),grupo6(4,:),grupo5(4,:));
trap29_f = poligono(grupo5(4,:),grupo6(4,:),grupo6(3,:),grupo5(3,:));
trap30_f= poligono(grupo5(3,:),grupo6(3,:),grupo6(5,:),grupo5(5,:));
trap31_f = poligono(grupo6(1,:),grupo6(5,:),grupo7(5,:),grupo7(1,:));
trap32_f= poligono(grupo6(1,:),grupo7(1,:),grupo7(6,:),grupo6(6,:));
trap33_f = poligono(grupo6(6,:),grupo7(6,:),grupo7(2,:),grupo6(2,:));
trap34_f= poligono(grupo6(2,:),grupo7(2,:),grupo7(4,:),grupo6(4,:));
trap35_f = poligono(grupo6(4,:),grupo7(4,:),grupo7(3,:),grupo6(3,:));
trap36_f= poligono(grupo6(3,:),grupo7(3,:),grupo7(5,:),grupo6(5,:));

```

%Incentros

%Triángulos

```

in1=tr1_f.incentro; in2=tr2_f.incentro; in3=tr3_f.incentro; in4=tr4_f.incentro;
in5=tr5_f.incentro; in6=tr6_f.incentro;

```

%Trapezoides

```

in7 = trap1_f.incentro ; in8 = trap2_f.incentro ; in9 = trap3_f.incentro ;
in10 = trap4_f.incentro ; in11 = trap5_f.incentro ;
in12= trap6_f.incentro ; in13 = trap7_f.incentro ; in14 = trap8_f.incentro ;
in15 = trap9_f.incentro ; in16 = trap10_f.incentro;
in17= trap11_f.incentro ; in18 = trap12_f.incentro; in19 = trap13_f.incentro ;
in20 = trap14_f.incentro ; in21 = trap15_f.incentro;
in22= trap16_f.incentro ; in23 = trap17_f.incentro; in24 = trap18_f.incentro ;
in25 = trap19_f.incentro ; in26 = trap20_f.incentro;
in27= trap21_f.incentro ; in28 = trap22_f.incentro; in29 = trap23_f.incentro ;
in30 = trap24_f.incentro ; in31 = trap25_f.incentro;
in32= trap26_f.incentro ; in33 = trap27_f.incentro; in34 = trap28_f.incentro ;
in35 = trap29_f.incentro ; in36 = trap30_f.incentro;
in37= trap31_f.incentro ; in38 = trap32_f.incentro; in39 = trap33_f.incentro ;
in40 = trap34_f.incentro ; in41 = trap35_f.incentro;
in42= trap36_f.incentro ;

```

IN =

```

[in1;in2;in3;in4;in5;in6;in7;in8;in9;in10;in11;in12;in13;in14;in15;in16;in17;in18;in19;
in20;in21;in22;in23;in24;in25;in26;in27;in28;in29;in30;in31;in32;in33;in34;...
in35;in36;in37;in38;in39;in40;in41;in42];

```

%Areas

%Triángulos

```

ar1 = tr1_f.area; ar2 = tr2_f.area; ar3 = tr3_f.area; ar4 = tr4_f.area;
ar5 = tr5_f.area; ar6 = tr6_f.area;

```

%Trapezoides

```

ar7 = trap1_f.area ; ar8 = trap2_f.area ; ar9 = trap3_f.area;
ar10 = trap4_f.area ; ar11= trap5_f.area ; ar12 = trap6_f.area;
ar13 = trap7_f.area ; ar14 = trap8_f.area ; ar15 = trap9_f.area;
ar16 = trap10_f.area ; ar17 = trap11_f.area ; ar18 = trap12_f.area;
ar19 = trap13_f.area ; ar20 = trap14_f.area ; ar21 = trap15_f.area;
ar22 = trap16_f.area ; ar23 = trap17_f.area ; ar24 = trap18_f.area;
ar25 = trap19_f.area ; ar26 = trap20_f.area ; ar27 = trap21_f.area;
ar28 = trap22_f.area ; ar29 = trap23_f.area ; ar30 = trap24_f.area;
ar31 = trap25_f.area ; ar32 = trap26_f.area ; ar33 = trap27_f.area;
ar34 = trap28_f.area ; ar35 = trap29_f.area ; ar36 = trap30_f.area;
ar37 = trap31_f.area ; ar38 = trap32_f.area ; ar39 = trap33_f.area;
ar40 = trap34_f.area ; ar41 = trap35_f.area ; ar42 = trap36_f.area;
AR =
[ar1;ar2;ar3;ar4;ar5;ar6;ar7;ar8;ar9;ar10;ar11;ar12;ar13;ar14;ar15;ar16;ar17;ar18;ar19;
ar20;ar21;ar22;ar23;ar24;ar25;ar26;ar27;ar28;ar29;ar30;ar31;ar32;ar33;...
ar34;ar35;ar36;ar37;ar38;ar39;ar40;ar41;ar42];

```

%Vectores normales:

%Triangulos

```

N1 = cross(tr1_f.vertices(1,:) - in1, tr1_f.vertices(2,:) - in1);
N2 = cross(tr2_f.vertices(1,:) - in2, tr2_f.vertices(2,:) - in2);
N3 = cross(tr3_f.vertices(1,:) - in3, tr3_f.vertices(2,:) - in3);
N4 = cross(tr4_f.vertices(1,:) - in4, tr4_f.vertices(2,:) - in4);
N5 = cross(tr5_f.vertices(1,:) - in5, tr5_f.vertices(2,:) - in5);
N6 = cross(tr6_f.vertices(1,:) - in6, tr6_f.vertices(2,:) - in6);

```

%Trapeacios

```

N7 = cross(trap1_f.vertices(1,:) - in7, trap1_f.vertices(2,:) - in7);
N8 = cross(trap2_f.vertices(1,:) - in8, trap2_f.vertices(2,:) - in8);
N9 = cross(trap3_f.vertices(1,:) - in9, trap3_f.vertices(2,:) - in9);
N10 = cross(trap4_f.vertices(1,:) - in10, trap4_f.vertices(2,:) - in10);
N11 = cross(trap5_f.vertices(1,:) - in11, trap5_f.vertices(2,:) - in11);
N12 = cross(trap6_f.vertices(1,:) - in12, trap6_f.vertices(2,:) - in12);
N13 = cross(trap7_f.vertices(1,:) - in13, trap7_f.vertices(2,:) - in13);
N14 = cross(trap8_f.vertices(1,:) - in14, trap8_f.vertices(2,:) - in14);
N15 = cross(trap9_f.vertices(1,:) - in15, trap9_f.vertices(2,:) - in15);
N16 = cross(trap10_f.vertices(1,:) - in16, trap10_f.vertices(2,:) - in16);
N17 = cross(trap11_f.vertices(1,:) - in17, trap11_f.vertices(2,:) - in17);
N18 = cross(trap12_f.vertices(1,:) - in18, trap12_f.vertices(2,:) - in18);
N19 = cross(trap13_f.vertices(1,:) - in19, trap13_f.vertices(2,:) - in19);
N20 = cross(trap14_f.vertices(1,:) - in20, trap14_f.vertices(2,:) - in20);
N21 = cross(trap15_f.vertices(1,:) - in21, trap15_f.vertices(2,:) - in21);
N22 = cross(trap16_f.vertices(1,:) - in22, trap16_f.vertices(2,:) - in22);
N23 = cross(trap17_f.vertices(1,:) - in23, trap17_f.vertices(2,:) - in23);
N24 = cross(trap18_f.vertices(1,:) - in24, trap18_f.vertices(2,:) - in24);
N25 = cross(trap19_f.vertices(1,:) - in25, trap19_f.vertices(2,:) - in25);
N26 = cross(trap20_f.vertices(1,:) - in26, trap20_f.vertices(2,:) - in26);
N27 = cross(trap21_f.vertices(1,:) - in27, trap21_f.vertices(2,:) - in27);
N28 = cross(trap22_f.vertices(1,:) - in28, trap22_f.vertices(2,:) - in28);
N29 = cross(trap23_f.vertices(1,:) - in29, trap23_f.vertices(2,:) - in29);
N30 = cross(trap24_f.vertices(1,:) - in30, trap24_f.vertices(2,:) - in30);
N31 = cross(trap25_f.vertices(1,:) - in31, trap25_f.vertices(2,:) - in31);
N32 = cross(trap26_f.vertices(1,:) - in32, trap26_f.vertices(2,:) - in32);
N33 = cross(trap27_f.vertices(1,:) - in34, trap27_f.vertices(2,:) - in34);
N34 = cross(trap28_f.vertices(1,:) - in34, trap28_f.vertices(2,:) - in34);
N35 = cross(trap29_f.vertices(1,:) - in35, trap29_f.vertices(2,:) - in35);
N36 = cross(trap30_f.vertices(1,:) - in36, trap30_f.vertices(2,:) - in36);

```

```

N37 = cross(trap31_f.vertices(1,:) - in37,trap31_f.vertices(2,:) - in37);
N38 = cross(trap32_f.vertices(1,:) - in38,trap32_f.vertices(2,:) - in38);
N39 = cross(trap33_f.vertices(1,:) - in39,trap33_f.vertices(2,:) - in39);
N40 = cross(trap34_f.vertices(1,:) - in40,trap34_f.vertices(2,:) - in40);
N41 = cross(trap35_f.vertices(1,:) - in41,trap35_f.vertices(2,:) - in41);
N42 = cross(trap36_f.vertices(1,:) - in42,trap36_f.vertices(2,:) - in42);

N =
[N1;N2;N3;N4;N5;N6;N7;N8;N9;N10;N11;N12;N13;N14;N15;N16;N17;N18;N19;N20;N21;N22;N23;N24
;N25;N26;N27;N28;N29;N30;N31;N32;N33;N34;N35;N36;...
N37;N38;N39;N40;N41;N42];
N_norm = N./(sqrt(sum(N.^2,2)));

```

%Conjunto de vectores normales a las subdivisiones del triangulo

```

normal = N_norm;
area = AR;
incentro = IN;
end

```

8.2.3. Código de creación de objeto polígono.

```

%genera una clase triangulo o cuadrilatero (más de 4 lados da error)
classdef poligono
    properties
        vertices
        incentro
        area
    end
    methods
        function v = poligono(a,b,c,varargin) % a = vertices(1,:), b = vertices(2,:),
c = vertices(3,:), d = vertices(4,;)

            v.vertices = [a;b;c;cell2mat(varargin)];
            v.incentro = v.incentrop();
            v.area = v.aro();
        end
        function i = incentrop(obj)
            x = obj.vertices;
            i = sum(x)/length(x);
        end
        function a = aro(obj)
            y = obj.vertices;
            n = length(y);
            cg = 1/n*(sum(y));
            vb = y-cg;
            t_sum = 0;
            for i=1:n
                if i == n
                    cr = cross(vb(i,:),vb(1,:));
                else
                    cr = cross(vb(i,:),vb(i+1,:));
                end
                t_sum = t_sum + cr;
            end
        end
    end
end

```

```

end
a = 1/2*norm(t_sum);
end
end
end

```

8.2.4. Validación de Método.

Para validar los valores entregados por el método de obtención de la matriz de masa agregada propuesto, se calcula la matriz M_a para una esfera de radio $r = 0.25$ [m].

La razón por la cual se calcula el valor de la matriz para una esfera es, debido a que se sabe que los valores de los elementos vienen dados por la ecuación,

$$m_{11} = m_{22} = m_{33} = \frac{2}{3}\pi\rho r^3 \quad (B.1)$$

Con ρ la densidad del agua y r el radio de la esfera, por lo tanto, la matriz M_a queda como,

$$M_a = \begin{bmatrix} m_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & m_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & m_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (B.2)$$

Donde, reemplazando los valores de la esfera en (B.1) se obtendría,

$$m_{11} = m_{22} = m_{33} = \frac{2}{3}\pi\rho 0.25^3 = 32.7249$$

Utilizando el método propuesto, la matriz M_a computada es,

$$M_a = \begin{bmatrix} 32.6892 & 0.0008 & 0.0001 & -0.0012 & -0.0107 & 0.0273 \\ 0.0008 & 32.6807 & 0.0004 & 0.0106 & 0.0025 & 0.0043 \\ 0.0006 & 0.008 & 32.6881 & 0.0273 & 0.0043 & 0.0013 \\ 0.0012 & 0.0106 & 0.0273 & 0.000 & 0.000 & 0.000 \\ 0.0107 & 0.0025 & 0.0043 & 0.000 & 0.000 & 0.000 \\ 0.0273 & 0.0043 & 0.0013 & 0.000 & 0.000 & 0.000 \end{bmatrix} \quad (B.3)$$

8.3. Anexo C. Procesamiento de Datos de Sensores.

Para el procesamiento de los datos del sensor, se hará uso del bloque “*Matlab function*” de Simulink.



Fig. C.1. Bloque Matlab Function para el procesamiento de datos.

Dentro de este se realiza la ejecución de la función encargada de procesar los datos, de tal forma que se pueda obtener el vector *dist/or* para el comportamiento de *Evasión de obstáculos*, para realizar esta tarea es necesario ocupar el motor de Matlab.

1. Código de procesamiento en bloque Simulink .

```
function [dist , or] = PlotLay(PointCloud,Location,Distance,Roll)
    coder.extrinsic("Seg_Sen2")
    vec = zeros(1,2);
    [vec(1),vec(2)] = Seg_Sen2(PointCloud,Location,Distance,Roll);
    dist = vec(1);
    or = vec(2);
end
```

2. Código función “Seg_Sen2”.

```
function [distmin,or] = Seg_Sen2(u,l,d,r)
persistent player
Loc = l; %Localización del Sensor en coordenadas de la simulación
roll = r;
pc = pointCloud(u); %Generación del PointCloud
assignin('base','pc_bruto',pc);
pc2 = processPtCld(pc);
assignin('base','pc2',pc2);

dist = d; %Matriz de distancia
PointData = pc2.Location;
if isempty(player)
    xlims = [-10,10];
    ylims = [-15,15];
    zlims = [-5,5];
    player = pcplayer(xlims,ylims,zlims); %Iniciar visor de LiDAR
end
if ~isempty(player) && isOpen(player)
    xlabel(player.Axes,'X(m)');
    ylabel(player.Axes,'Y(m)');
    zlabel(player.Axes,'Z(m)');
    view(player,pc2);
```

```

end

%Definicion de colores para segmentacion
colorlabels = [...
    0      0.4470  0.7410;...%Sin Etiqueta
    0.9290 0.6940  0.1250;...%Ego
    0.6350 0.0780  0.1840;...%Obstaculo
];
colors.Unlabeled = 1;
colors.Ego = 2;
colors.Obstacle = 3;
colormap(player.Axes,colorlabels);

%Segmentar Vehiculo

Xlim = [-0.5,0.5];
Ylim = [-1,1];
Zlim = [-1,1];
%
Puntos = struct();
Puntos.EgoPoints = pc2.Location(:,:,1) > Xlim(1) ...
    & pc2.Location(:,:,1) < Xlim(2) ...
    & pc2.Location(:,:,2) > Ylim(1) ...
    & pc2.Location(:,:,2) < Ylim(2) ...
    & pc2.Location(:,:,3) > Zlim(1) ...
    & pc2.Location(:,:,3) < Zlim(2);

closePlayer = false;

%Obstaculos
SensorLocation = Loc;
radio = 15;

%Tratado PointCloud
pc3 = removeInvalidPoints(pc2);
pc2D = [pc3.Location(:,1),pc3.Location(:,2)];%Transladar a un vision del plano xy
T = dbscan(pc2D,1,5); %Clusterizacion por densidad
idk = find(T ~= -1); %Identificacion de indices para obstaculos
if ~isempty(idk)
    pc_obs = pc2D(idk,:);
    mag_pc2D = sqrt(sum(pc_obs.^2,2));
    distmin = min(mag_pc2D);
    idmin = find(mag_pc2D == distmin);
if length(idmin) > 1
    pc_temp = pc_obs(idmin,:);
    pc_min = mean(pc_temp);
else
    pc_min = pc_obs(idmin,:);
end
or = -atan2(pc_min(1),-pc_min(2))*(180/pi);
assignin('base','pc_obs',pc_obs);
assignin('base','pc_min',pc_min);
figure(2)
scatter(pc_obs(:,1),pc_obs(:,2),5,'filled','red')

```

```

hold on
plot(pc_min(1,1),pc_min(1,2), '*')
hold off

else
    distmin = 20;
    or = 0;
end

O_puntos = ~Puntos.EgoPoints;
ptCloudSegmented = select(pc2,O_puntos,'OutputSize','full');
Puntos.ObstaclePoints = findNeighborsInRadius(ptCloudSegmented,SensorLocation,radio);
ptObs = select(pc2,Puntos.ObstaclePoints);

Actualizar(player,pc2,Puntos,colors,closePlayer);%Funcion actualizacion de visor
end

```

Luego del procesamiento de los datos entregados por la simulación de Lidar en Unreal Engine, se obtiene una representación del ambiente que rodea al vehículo como la que se ve en Fig.C.2.

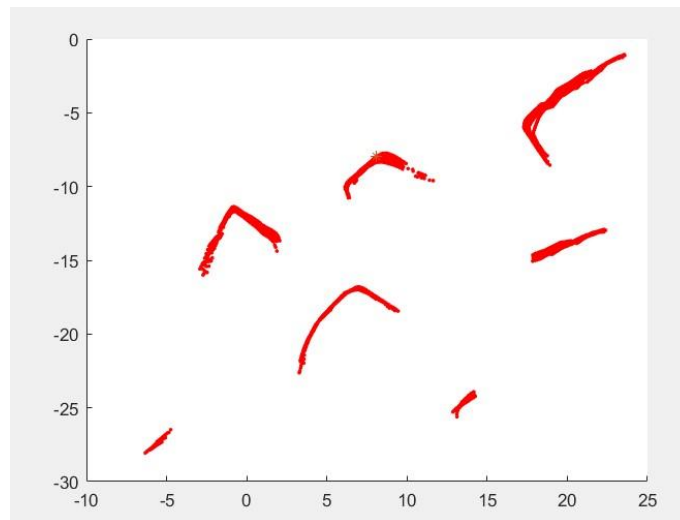


Fig. C.2. Representación del ambiente luego de procesamiento de datos

8.4. Anexo D. Implementación de Defusificación de FLC

Como se menciona en la sección 5.3.7, el método de defusificación a utilizar es una modificación al método del centroide dada por las ecuaciones 5.2 y 5.3, para esto se utiliza el bloque “*Matlab Function*” de Simulink, en el cual se implementa la función encargada del proceso de defusificación.



Fig. D.1. Función de decodificación empleado en Simulink.

En Fig. D.1. se puede ver que el bloque posee de entradas los pesos correspondientes a cada comportamiento y las variables de entrada fusificadas y agregadas mencionadas como: “*u1_meta*”, “*u2_eo*”. Para las salidas se tienen: “*sal1_meta*”, “*sal2_meta*”, “*sal1_eo*”, “*sal2_eo*”, “*sal_peso_vel*” y “*sal_peso_delyaw*”, donde, las primeras cuatro corresponden a las variables de salida sin el actuar de los pesos y las últimas dos son aquellas en donde se incluyen los grados de aplicabilidad (DOA).

8.4.1. Código empleado en la defusificación.

```
Function [sal1_meta,sal2_meta,sal1_eo,sal2_eo,sal_peso_vel,sal_peso_delyaw] =  
fcn(u1_meta,u2_eo,peso_meta,peso_eo)  
  
coder.extrinsic("assignin")  
coder.extrinsic("find")  
idx = 0;  
armax = 0;  
% tamaño = size(u);  
% Rangos para la meta  
rang1_meta = [-0.7854 0.7854]; %del_yaw_meta  
rang2_meta = [0 0.5]; %vel_meta  
% Rangos para evasión de obstáculos  
rang1_eo = [0 0.5]; %vel_eo  
rang2_eo = [-0.7854 0.7854]; %del_yaw_eo  
% Valores fuzzificados y agregados  
del_yaw_eo = u2_eo(:,2);  
assignin('base','del_yaw_eo',del_yaw_eo);  
vel_eo = u2_eo(:,1);  
del_yaw_meta = u1_meta(:,1);
```

```

assignin('base','del_yaw_meta',del_yaw_meta);
vel_meta = u1_meta(:,2);
%Rangos
[n1_meta,temp] = size(del_yaw_meta);
assignin('base','n1_meta',n1_meta);
[n2_meta,temp] = size(vel_meta);
[n1_eo,temp] = size(del_yaw_eo);
assignin('base','n1_eo',n1_eo);
[n2_eo,temp] = size(vel_eo);
%Universos de discurso
x1_meta = linspace(rang1_meta(1),rang1_meta(2),n1_meta)'; %del_yaw_meta
nx1 = length(x1_meta); nx13 = nx1/3;
assignin('base','x1_meta',x1_meta);
x2_meta = linspace(rang2_meta(1),rang2_meta(2),n2_meta)'; %vel_meta
x1_eo = linspace(rang2_eo(1),rang2_eo(2),n1_eo)'; %del_yaw_eo
assignin('base','x2_meta',x2_meta);
x2_eo = linspace(rang1_eo(1),rang1_eo(2),n2_eo)'; %vel_eo
%Areas.
area1_meta = sum(del_yaw_meta); %del_yaw_meta
assignin('base','area1_meta',area1_meta);
area2_meta = sum(vel_meta); %vel_meta
assignin('base','area2_meta',area2_meta);
area1_eo = sum(del_yaw_eo); %del_yaw_eo
assignin('base','area1_eo',area1_eo);
area2_eo = sum(vel_eo); %vel_eo
assignin('base','area2_eo',area2_eo);
%Comprobación valores de salida
sal1_meta = sum(x1_meta.*del_yaw_meta)/sum(del_yaw_meta);
sal2_meta = sum(x2_meta.*vel_meta)/sum(vel_meta);

%Salida Final
sal_peso_vel=(peso_meta*(sum(x2_meta.*vel_meta))+peso_eo*(sum(x2_eo.*vel_eo)))/(peso_me
ta*sum(vel_meta) + peso_eo*sum(vel_eo));

sal_peso_delyaw =
(peso_meta*sum(x1_meta.*del_yaw_meta)+peso_eo*sum(x1_eo.*del_yaw_eo))/(peso_meta*sum(de
l_yaw_meta)+peso_eo*sum(del_yaw_eo));

end

```

8.5. Anexo E. Horas hombre requeridas

Para el tiempo mínimo que se requeriría en llevar a cabo la implementación de este proyecto, se listan tareas junto al tiempo requerido por un ingeniero para ejecutarlas.

| Actividad | HH |
|--|-------------|
| Optimizaciones de diseño propuesto. | 40 |
| Adquisición de Instrumentación y materiales de construcción. | 160 |
| Ensamblaje de Diseño. | 160 |
| Pruebas de equipo y comunicación con sensores. | 160 |
| Pruebas de Localización con hidrófono | 120 |
| Adquisición experimental de parámetros hidrodinámicos. | 80 |
| Programación y sintonización de sistemas de regulación. | 80 |
| Pruebas de funcionamiento y corrección de sistema de regulación en piscina. | 120 |
| Programación de control de navegación. | 160 |
| Pruebas de funcionamiento y corrección del sistema de navegación en piscina. | 120 |
| Total HH | 1200 |

Tabla E.1. Horas hombre mínimas requeridas.

**UNIVERSIDAD DE CONCEPCION – FACULTAD DE INGENIERIA
RESUMEN DE MEMORIA DE TITULO**

Departamento : Departamento de Ingeniería Eléctrica
Carrera : Ingeniería civil electrónica
Nombre del memorista : Lamar Contreras Zambrano
Título de la memoria : Diseño, Modelado y Simulación de un Vehículo Submarino Autónomo No Tripulado.
Fecha de la presentación oral : 01/09/2023
Profesor(es) guía : Juan Pablo Segovia Vera.
Profesor(es) revisor(es) : Lautaro David Salazar Silva.
Leonardo Manuel Palma Fanjul.
Concepto :
Calificación :

Resumen (máximo 200 palabras)

En el siguiente trabajo se presenta el diseño de un vehículo submarino autónomo no tripulado abreviado AUV, donde se describe el modelo del vehículo junto a sus variables hidrostáticas, hidrodinámicas y cinemáticas, así como también, el de los propulsores que son requeridos para la movilidad de este.

Se presenta el proceso de obtención de los parámetros hidrodinámicos, específicamente la masa agregada y las fuerzas de amortiguamiento, las cuales son requeridas para la completa definición física del móvil.

Se muestra el sistema de regulación de posición, pose y altura para el diseño propuesto, principalmente este fue diseñado para facilitar el control de navegación autónoma.

De igual forma, se describe la estrategia de control difuso para la navegación autónoma junto con la ingeniería conceptual del sistema de localización del vehículo.

Por último, se presenta un listado de instrumentos necesarios para la implementación de este proyecto.

En este trabajo se diseñó, modeló y simuló un vehículo autónomo submarino no tripulado junto a su sistema de navegación autónoma, donde se obtuvo una buena aproximación de los parámetros hidrodinámicos y, además, el sistema de control permite al móvil moverse de manera segura hacia la meta propuesta.