



**CENTRO DE INGENIERIA Y DESARROLLO INDUSTRIAL**

---

**MODELADO Y SIMULACIÓN DE UN ROBOT  
SUBMARINO EN UN AMBIENTE VIRTUAL**

**TESIS**

QUE PARA OBTENER EL GRADO

ACADEMICO DE

**MAESTRO EN CIENCIA Y TECNOLOGIA**

**EN LA ESPECIALIDAD DE**

**MECATRONICA**

PRESENTA

**ING. GLADYS LÓPEZ HERNÁNDEZ**

QUERETARO, QRO. DICIEMBRE 2017.





Director de Posgrado  
PICYT – CIDESI  
Querétaro

Los abajo firmantes, miembros del Comité Tutorial del alumno **Gladys López Hernández**, una vez leída y revisada la Tesis titulada “**MODELADO Y SIMULACIÓN DE UN ROBOT SUBMARINO EN UN AMBIENTE VIRTUAL**”, aceptamos que la referida tesis revisada y corregida sea presentada por el alumno para aspirar al grado de **Maestría en Ciencia y Tecnología** en la opción terminal de **Mecatrónica** durante el Examen de Grado correspondiente.

Y para que así conste firmo la presente a los 04 días del mes de Diciembre del año dos mil diecisiete.

---

Dr. Tomás Salgado Jiménez

Director de Tesis



Director de Posgrado  
PICYT – CIDESI  
Querétaro

Los abajo firmantes, miembros del Jurado del Examen de Grado de la alumna **Gladys López Hernández**, una vez leída y revisada la Tesis titulada “**MODELADO Y SIMULACIÓN DE UN ROBOT SUBMARINO EN UN AMBIENTE VIRTUAL**”, aceptamos que la referida tesis revisada y corregida sea presentada por el alumno para aspirar al grado de **Maestría en Ciencia y Tecnología** en la opción terminal de Mecatrónica durante el Examen de Grado correspondiente.

Y para que así conste firmo la presente a los 12 días del mes de diciembre del año dos mil diecisiete.

---

Dr. José Luis Sánchez Gaytán

Presidente

---

M. en C. Fernando Fonseca Navarro

Secretario

---

# Dedicatoria

---

## **A mis padres Pedro y Miriam**

Por su amor incondicional, por su apoyo en cada etapa de mi vida personal y profesional y quienes con su ejemplo me han enseñado a seguir adelante.

## **A mi esposo Gonzalo**

Con quien he compartido momentos difíciles y de alegría, por sostenerme y apoyarme en momentos especiales.

## **A mis hijos Gonzalo y Sebastián**

Por su amor incondicional y quienes son un motor importante en mi vida que me motivan a superarme además de enseñarme lo que significa el amor sincero.

## **A mi familia**

Que por su paciencia y aliento me ayudaron a concluir una etapa más en mi vida profesional.

---

# Agradecimientos

---

Al Dr. Tomás Salgado Jiménez por su apoyo, paciencia, confianza y guía a lo largo de este trabajo de tesis.

Al Centro de Ingeniería y Desarrollo Industrial (CIDESI) por su apoyo y confianza para presentar y terminar este proyecto.

A mis amigos y compañeros de trabajo que siempre me alentaron para seguir adelante. Gracias a.

---

# Resumen

---

Hoy en día la robótica submarina es de especial interés en la exploración petrolera, por su alta confiabilidad y el acceso a aguas cada vez más profundas. El desarrollo del robot Kaxan en CIDESI es estratégico debido a su contribución al impulso de la tecnología submarina en México, sin embargo y debido al alto costo que implica operarlo y el entrenamiento de sus operarios se vuelve una tarea difícil de llevar a cabo.

Con el presente trabajo de tesis, se pretende mediante el desarrollo de un sistema informático ofrecer una plataforma virtual de entrenamiento de pilotos del ROV (acrónimo del inglés Remote Operated Vehicle, Vehículo operado a distancia). En esta versión se trabajó con la implementación de los modelos cinemático e hidrodinámico del ROV Kaxan en un ambiente virtual que le proporciona al piloto del robot la sensación de operar físicamente el sistema. Adicionalmente se propuso un control deslizante de segundo orden para el control de profundidad del robot, lo que permitirá un acercamiento aproximado al comportamiento real del ROV y facilitará su operación.

Resultados en el ambiente virtual son presentados, obteniendo similitudes entre el ambiente virtual – el sistema real.

---

# Índice General

---

<b>1. Introducción .....</b>	<b>1</b>
<b>2. Definición del proyecto .....</b>	<b>6</b>
<b>3. Justificación .....</b>	<b>7</b>
<b>4. Objetivo.....</b>	<b>8</b>
<b>5. Fundamentación.....</b>	<b>9</b>
5.1 Antecedentes matemáticos .....	9
5.1.1 Localización espacial .....	9
5.1.2. Cinemática.....	16
5.2 Modelos matemáticos.....	24
5.2.1 Modelo Cinemático .....	24
5.2.2 Modelo Hidrodinámico .....	25
5.3 Leyes de control .....	29
5.3.1 Control PID (Proporcional, Integral y Derivativo) .....	30
5.3.2 Control Deslizante de segundo Orden.....	31
<b>6. Metodología .....</b>	<b>33</b>
6.1 Metodología de desarrollo .....	33
6.2 Configuraciones adicionales .....	34
6.3 Control deslizante de segundo orden .....	34
6.4 Parámetros y constantes .....	36
6.5 Consideraciones de ambiente virtual .....	36
6.6 Configuración joystick .....	38
6.7 Solución de ecuaciones. ....	41
6.8 Propulsores .....	44
6.8.1 Propulsor 520 .....	45
6.8.2 Propulsor 540 .....	45

---

6.9 Plataforma de entrenamiento .....	46
6.9.1 Interfaz Gráfica de Usuario (GUI) .....	46
6.9.2 Simulación.....	51
<b>7. Resultados.....</b>	<b>55</b>
<b>8. Conclusiones .....</b>	<b>56</b>
<b>Bibliografía.....</b>	<b>57</b>
<b>Anexo A. Hoja de datos Propulsores .....</b>	<b>58</b>
A.1 Modelo 520 .....	58
A.2 Modelo 540 .....	60



---

# Índice de figuras

---

Figura 1.1 Vehículos submarinos.....	1
Figura 1.2 Ambiente Robot Maski.....	2
Figura 1.3 Ambiente submarino y un ROV.....	3
Figura 1.4 Escenarios de trabajo en simulación de un ROV.....	4
Figura 5.1 Representación de las coordenadas Polares y Cilíndricas.....	9
Figura 5.2 Representación de un vector con coordenadas esféricas.....	10
Figura 5.3 Ángulos de Euler.....	13
Figura 5.4 Ángulos de Euler ZYZ.....	14
Figura 5.5 Ángulos de Euler: Roll, Pitch y Yaw.....	15
Figura 5.6 Variables de movimiento para vehículos marinos.....	16
Figura 6.1 Diagrama de bloques propuesto en el desarrollo.....	33
Figura 6.2 Joystick DualShock 3.....	40
Figura 6.3 Ubicación de los propulsores del ROV Kaxan.....	45
Figura 6.4 Interfaz Gráfica de Usuario (GUI).....	46
Figura 6.5 Área de Trabajo.....	47
Figura 6.6 Corriente Marina.....	48
Figura 6.7 Área de control de profundidad PID.....	48
Figura 6.8 Área de control de dirección $\psi$ PID.....	49
Figura 6.9 Área de escenarios virtuales predeterminados.....	49
Figura 6.10 Área de estados.....	50
Figura 6.11 Tipo de Controlador.....	50
Figura 6.12 Área de Start / Stop / Charts.....	51
Figura 6.13 Área de selección Mode.....	51

---

Figura 6.14 Interfaz Gráfica Modo Prueba.....	52
Figura 6.15 Parámetros de simulación modo Prueba: PD deslizante.....	53
Figura 6.16 Gráficas de resultados modo Prueba: PD deslizante.....	53
Figura 6.17 Parámetros de simulación modo Prueba: PID.....	54
Figura 6.18 Gráficas de resultados modo Prueba: PID . .....	54

---

# Índice de tablas

---

Tabla 5.1 Notación SNAME para vehículos marinos .....	17
---	----

# 1. Introducción

---

## 1. Introducción

La robótica submarina inicia con los primeros torpedos desarrollados en Austria 1864. En 1953 fue desarrollado el primer ROV (Remote Operated Vehicle) nombrado POODLE.

Los vehículos submarinos pueden clasificarse en dos grandes ramas: los Tripulados y los Vehículos no tripulados [Figura 1.1], siendo este último el Tele-Operado (ROV) el objeto de estudio durante el desarrollo de la presente Tesis.

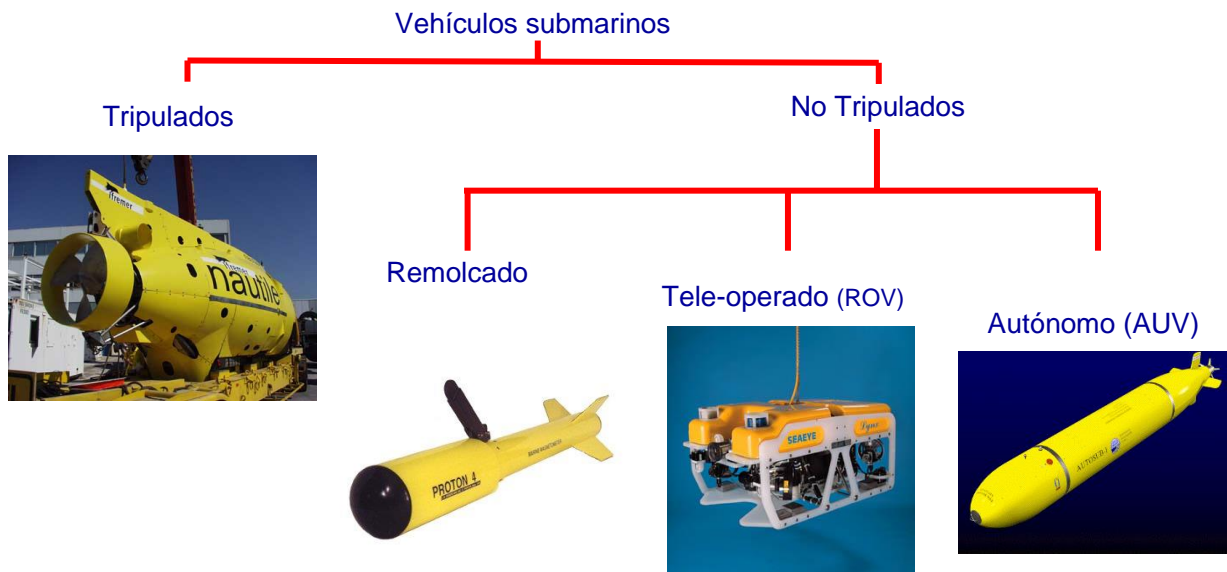


Figura 1.1 Vehículos submarinos.

### Clasificación de vehículos submarinos

Con los avances tecnológicos de hoy en día, en particular con la exploración petrolera, la robótica submarina se vuelve una parte esencial por su alta confiabilidad y el acceso a aguas cada vez más profundas, donde el acceso con vehículos tripulados se vuelve una tarea difícil de alcanzar.

# 1. Introducción

---

Dentro de los robots no tripulados se encuentra el ROV (Remote Operated Vehicle por sus siglas en inglés), que es un robot conectado a un barco en la superficie por medio de un cable. La energía y las órdenes se envían mediante un mando a distancia a través del cable al ROV.

Los ROV's pueden llevar una gran variedad de brazos manipuladores para realizar trabajos en las profundidades o simplemente una cámara fotográfica con el fin de captar las imágenes del fondo del mar. Los ROV's son utilizados en inspecciones submarinas a grandes profundidades, en la inspección de presas entre otras aplicaciones.

Una de las tendencias actuales en el estudio de la robótica submarina, es la simulación en realidad virtual con sistemas informáticos que permiten proporcionar la sensación de estar manipulando el robot en un ambiente real y predecir su comportamiento mediante la inclusión de las leyes de la ciencia que modelan su comportamiento; un buen ejemplo es el robot Maski fabricado por el instituto Hydro Québec. El robot Maski es utilizado para la inspección de presas hidroeléctricas; para su control los operarios cuentan con un tablero el cual muestra diferentes escenarios del robot, un monitor es dedicado a mostrar en realidad virtual la posición y orientación del robot el cual es de gran utilidad debido a la poca visibilidad que existe en el ambiente submarino de una cortina de presa [Figura 1.2].



a) Consola operación del Robot Maski.

b) Monitor ambiente real y virtual del robot

Figura 1.2 Ambiente Robot Maski.

# 1. Introducción

---

En la robótica submarina, los sistemas informáticos así como sus aplicaciones sigue siendo un campo de estudio de gran interés, debido a que éstas herramientas ayudan a modelar un ambiente submarino aplicando los modelos cinemático e hidrodinámico de un robot [Figura 1.3].

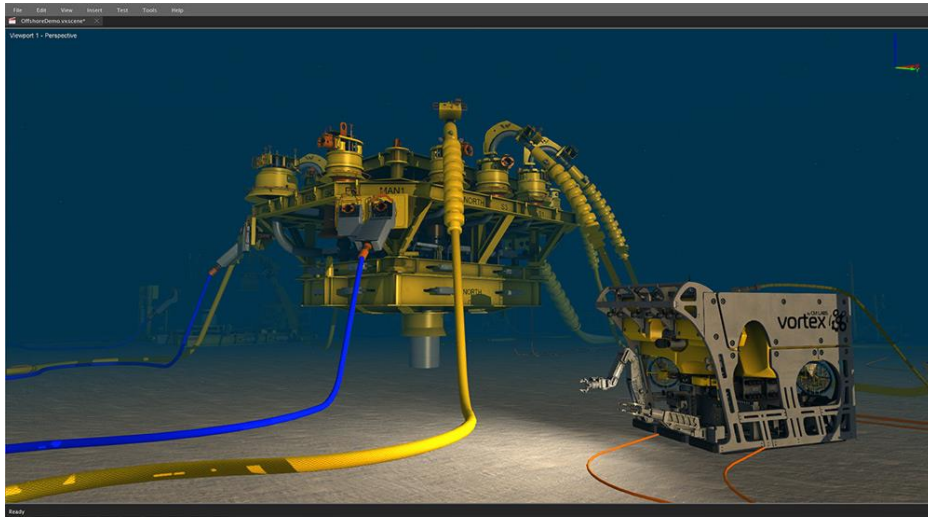


Figura 1.3 Ambiente submarino y un ROV.

Los simuladores de ROV's son una industria importante en el mercado de simuladores industriales en el entrenamiento de operarios de ROV's, tal es el caso de la empresa GRI Simulations Inc. (<http://www.grisim.com>). GRI se especializa en el desarrollo de simuladores para ROV's, los cuales son utilizados en el entrenamiento de pilotos en diferentes escenarios y situaciones de trabajo en aguas profundas. La [Figura 1.4] muestra los diferentes escenarios en los que un operario de ROV puede ser entrenado.

# 1. Introducción

---



Figura 1.4 Escenarios de trabajo en simulación de un ROV

En el mundo se han realizado diversos estudios en el campo de la simulación y realidad virtual submarina, por ejemplo:

En [10] se propone un ambiente híbrido el cual consiste en un simulador de ambiente virtual llamado MVS (Multi Vehículo Simulator) y un par de robots submarinos, el Twin-Burger I y II. El MVS es capaz de producir información de un mundo virtual en tiempo real, el cual puede ser configurado para el área de trabajo de los robots. El mundo virtual llamado “mundo sintético” es creado usando información del mundo real y virtual. Mediante la prueba en el mundo sintético, la eficacia en el desarrollo de los sistemas submarinos está siendo mejorada debido a que el software se puede desarrollar directamente en sistemas informáticos embebidos.

La arquitectura de software de un mundo virtual submarino para un robot autónomo es descrita en [3]. Usar un mundo virtual submarino para la representación de los gráficos 3D en tiempo real es una capacidad esencial para el desarrollo eficaz de un AUV. Los requisitos

## **1. Introducción**

---

principales incluyen la flexibilidad para la visualización y la portabilidad a través de múltiples plataformas de hardware y software.

El uso de ambientes virtuales [1] (VE por sus siglas en inglés) permite la creación de ambientes de entrenamiento incluyendo la simulación en una gama de operación, configuración de vehículos, escenarios de misiones, etc.



## **2. Definición del proyecto**

---

### **2. Definición del proyecto**

Durante la fase de pruebas y/o entrenamiento de operarios de un robot submarino costoso; en tiempo y dinero. Es necesario utilizar una herramienta informática que aplique el modelado del robot para describir el funcionamiento y predecir su comportamiento de una manera aproximada.

La aplicación del modelo cinemático e hidrodinámico del robot en un ambiente submarino virtual constituye un reto importante para lograr una simulación cercana al comportamiento real del robot. Se utiliza una metodología de desarrollo de software de simulación ya probada para simular el ROV Kaxan en un ambiente virtual de trabajo.

Con el presente trabajo de tesis se pretende realizar una aportación al trabajo de simulación del ROV Kaxan, donde se implementa el control del robot, así como la visualización de diferentes Gráficas que permitirán analizar el comportamiento del ROV durante la simulación.

El simulador incluye una interfaz gráfica que permite al operador la sensación de operar al ROV mediante el uso de un dispositivo de manipulación (Joystick).

### **3. Justificación**

---

### **3. Justificación**

Con el trabajo de tesis del “Modelado y simulación de un robot submarino en un ambiente virtual” se contribuirá a obtener beneficios como:

- El desarrollo de software dedicado a la simulación aportando nueva funcionalidad que le permita al operador una mayor sensación y entendimiento del comportamiento del ROV.
- Entrenamiento de los operarios en ambientes virtuales de robots en inspección e intervención submarina.
- Implementación de una ley de control avanzado que apoye al mejor comportamiento del ROV.

## 4. Objetivo

---

### 4. Objetivo

El objetivo principal del proyecto de Tesis es el de adaptar una herramienta informática que utilizando el modelado de un ROV permita el desarrollo de ambiente submarino realista, y donde sea posible implantar técnicas de control con la posibilidad de ajustar variables en línea.

Los objetivos específicos en el desarrollo del proyecto son:

- Herramienta informática de simulación en realidad virtual:
  - De fácil uso con una interfaz gráfica y con apoyo de accesorios (joystick) que permitan una simulación realista de comportamiento del ROV en un entorno submarino.
  - Que incluya herramientas para el modelado de un robot.
  - Animación gráfica del entorno del robot.
  - Gráficas de variables que son de interés para seguir el comportamiento de trayectoria del ROV.
  
- Incrementar la robustez del ROV Kaxan mediante la implementación de un control deslizante de segundo orden.

## 5. Fundamentación

---

## 5. Fundamentación

### 5.1 Antecedentes matemáticos

#### 5.1.1 Localización espacial

##### Representación de la posición

Para localizar un cuerpo rígido es necesario contar con una herramienta que permita la localización espacial de sus puntos. En el caso de un plano tridimensional es necesario contar con tres componentes. La forma más utilizada para especificar la posición de un punto son coordenadas cartesianas.

##### Coordenadas polares y cilíndricas

Para un plano es posible caracterizar la localización de un punto o vector  $\mathbf{p}$  respecto a un sistema de ejes cartesianos de referencia OXY, utilizando las coordenadas polares  $\mathbf{p}(r, \theta)$  [Figura 5.1].

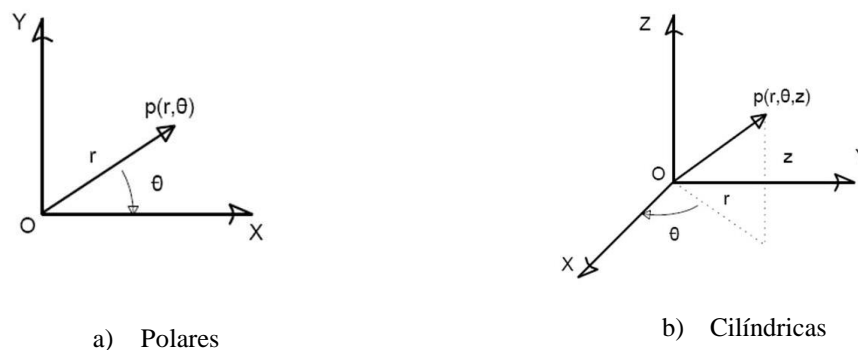


Figura 5.1 Representación de las coordenadas Polares y Cilíndricas.

En la [Figura 5.1a],  $r$  representa la distancia desde el origen hasta el extremo del vector  $\mathbf{p}$ , mientras que  $\theta$  es el ángulo que forma el vector  $\mathbf{p}$  con el eje OX.

## 5. Fundamentación

---

En la [Figura 5.1b] las coordenadas cilíndricas  $\mathbf{p}(r, \theta, z)$  las componentes  $r$  y  $\theta$  tienen el mismo significado que en el caso de las coordenadas polares, mientras que la componente  $z$  expresa la proyección sobre el eje OZ.

### Coordenadas esféricas

Es posible utilizar las coordenadas esféricas para la localización de un vector en un espacio de tres dimensiones [Figura 5.2]. Utilizando el sistema de referencia OXYZ, el vector  $\mathbf{p}$  tendrá como coordenadas esféricas  $(r, \theta, \phi)$ . Donde  $r$  es la distancia desde el origen hasta el extremo del vector  $\mathbf{p}$ ,  $\theta$  es el ángulo formado por la proyección del vector  $\mathbf{p}$  sobre el plano OXY con el eje OX y  $\phi$  es el ángulo formado por el vector  $\mathbf{p}$  con el eje OZ.

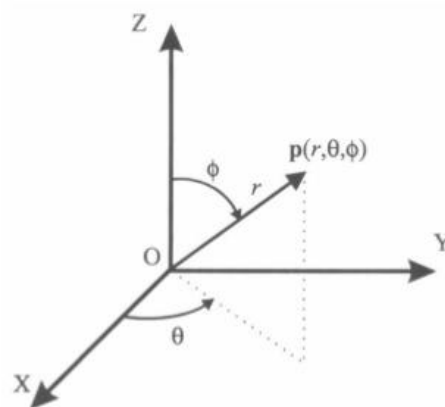


Figura 5.2 Representación de un vector con coordenadas esféricas.

### Representación de la orientación

Un punto queda totalmente definido en el espacio a través de los datos de su posición. Para el caso de un sólido es necesario definir cuál es su orientación con respecto a un sistema de referencia. La orientación en el espacio tridimensional se encuentra definida por tres grados de libertad o tres componentes linealmente independientes.

## 5. Fundamentación

---

### Matrices de rotación

Las matrices de rotación son un método extendido para la descripción de orientaciones. Una matriz de rotación define la orientación de un sistema con respecto a otro sistema, y que sirve para transformar las coordenadas de un vector en un sistema a las del otro. Recibe el nombre también de matriz de cosenos directores y se trata de una matriz orto-normal, tal que la inversa de la matriz es igual a su transpuesta.

En un espacio tridimensional, un vector  $\mathbf{p}$  del espacio podrá ser definido a cualquiera de los sistemas de la siguiente manera:

$$P_{uvw} = [P_u, P_v, P_w]^T = P_u \cdot i_u + P_v \cdot j_v + P_w \cdot k_w$$
$$P_{xyz} = [P_x, P_y, P_z]^T = P_x \cdot i_x + P_y \cdot j_y + P_z \cdot k_z$$

Se puede obtener la siguiente equivalencia:

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = R \begin{bmatrix} P_u \\ P_v \\ P_w \end{bmatrix}$$

Dónde:

$$R = \begin{bmatrix} i_x i_u & i_x j_v & i_x k_w \\ j_y i_u & j_y j_v & j_y k_w \\ k_z i_u & k_z j_v & k_z k_w \end{bmatrix}$$

La principal utilidad de esta matriz de rotación corresponde a la representación de la orientación de sistemas rotados únicamente sobre uno de los ejes principales del sistema de referencia.

Matrices básicas de rotación de un sistema espacial de tres dimensiones

## 5. Fundamentación

---

$$R(x, \alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\operatorname{sen} \alpha \\ 0 & \operatorname{sen} \alpha & \cos \alpha \end{bmatrix}$$

$$R(y, \phi) = \begin{bmatrix} \cos \phi & 0 & \operatorname{sen} \phi \\ 0 & 1 & 0 \\ -\operatorname{sen} \phi & 0 & \cos \phi \end{bmatrix}$$

$$R(z, \theta) = \begin{bmatrix} \cos \theta & -\operatorname{sen} \theta & 0 \\ \operatorname{sen} \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

### Composición de rotaciones

Las matrices de rotación pueden componerse para expresar la aplicación continua de varias rotaciones. La rotación global puede expresarse como:

$$T = R(z, \theta) R(y, \phi) R(x, \alpha) = \begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\phi & 0 & S\phi \\ 0 & 1 & 0 \\ -S\phi & 0 & C\phi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\alpha & -S\alpha \\ 0 & S\alpha & C\alpha \end{bmatrix} =$$
$$\begin{bmatrix} C\theta C\phi & -S\theta C\alpha + C\theta S\phi S\alpha & S\theta S\alpha + C\theta S\phi C\alpha \\ S\theta C\phi & C\theta C\alpha + S\theta S\phi S\alpha & -C\theta S\alpha + S\theta S\phi C\alpha \\ -S\phi & C\phi S\alpha & C\phi C\alpha \end{bmatrix}$$

Donde  $C\theta$  expresa  $\cos \theta$  y  $S\theta$  expresa  $\operatorname{sen} \theta$ .

Se considera importante el orden en que se realizan las rotaciones (el producto de matrices no es conmutativo).

## 5. Fundamentación

---

### Ángulos de Euler

Para la representación de orientación en un espacio tridimensional mediante una matriz de rotación es necesario definir tres elementos. Todo sistema fijo al cuerpo cuya orientación se requiere describir, puede definirse con respecto a un sistema de referencia mediante tres ángulos:  $\varphi$ ,  $\theta$ ,  $\psi$ ; denominados Ángulos de Euler. Es necesario definir los ejes sobre los que se realizan los giros.

### Ángulos de Euler ZXZ

Es una de las representaciones más habituales entre las que realizan los giros sobre ejes previamente girados [Figura 5.3]. Se suele asociar con los movimientos básicos de un giroscopio.

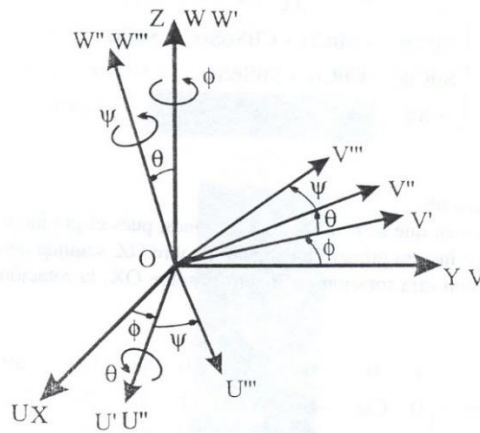


Figura 5.3 Ángulos de Euler.

Se puede colocar al sistema OUVW en cualquier orientación siguiendo los siguientes pasos:

1. Girar el sistema OUVW en un ángulo  $\varphi$  con respecto al eje OZ, convirtiéndose en  $OU'V'W'$
2. Girar el sistema  $OU'V'W'$  un ángulo  $\theta$  con respecto al eje  $OU'$ , convirtiéndose así en  $OU''V''W''$



## 5. Fundamentación

---

3. Girar el sistema  $OU''V''W''$  un ángulo  $\psi$  con respecto al eje  $OW''$ , convirtiéndose así en  $OU'''V'''W'''$

### Ángulos de Euler ZYZ

Es otra de las representaciones habituales entre las que realizan los giros sobre ejes previamente girados [Figura 5.4].

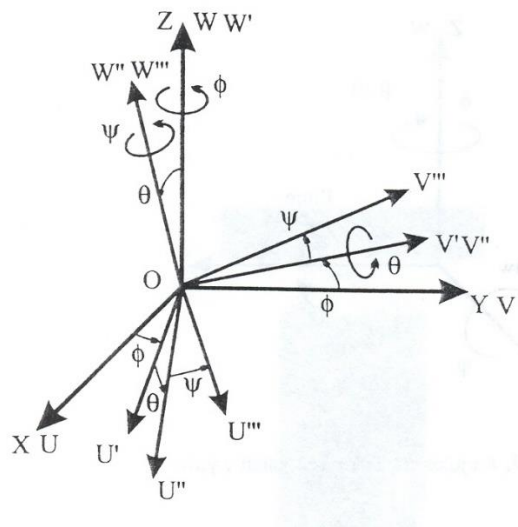


Figura 5.4 Ángulos de Euler ZYZ.

Se puede colocar al sistema OUVW en cualquier orientación siguiendo los siguientes pasos:

1. Girar el sistema OUVW en un ángulo  $\phi$  con respecto al eje OZ, convirtiéndose en  $OU'V'W'$
2. Girar el sistema  $OU'V'W'$  un ángulo  $\theta$  con respecto al eje  $OV'$ , convirtiéndose así en  $OU''V''W''$
3. Girar el sistema  $OU''V''W''$  un ángulo  $\psi$  con respecto al eje  $OW''$ , convirtiéndose así en  $OU'''V'''W'''$

## 5. Fundamentación

---

Considerar para ambos casos que el orden de los giros no es conmutativo.

### Roll, Pitch y Yaw (alabeo, cabeceo y guiñada)

Se trata de una representación utilizada generalmente en aeronáutica. Es la más habitual de entre la que se aplican a los giros sobre los ejes del sistema fijo [Figura 5.5].

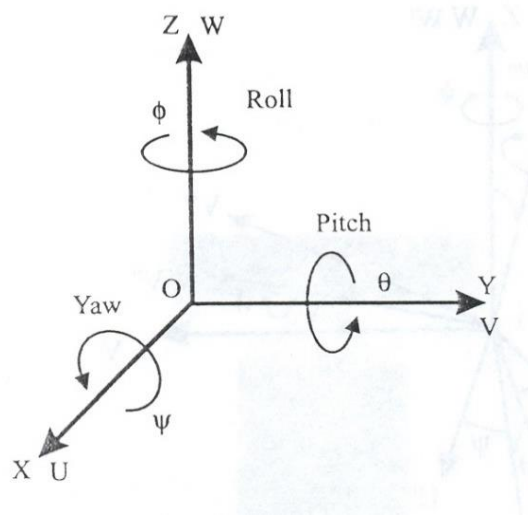


Figura 5.5 Ángulos de Euler: Roll, Pitch y Yaw.

Se puede colocar al sistema OUVW en cualquier orientación siguiendo los siguientes pasos:

1. Girar el sistema OUVW en un ángulo  $\psi$  con respecto al eje OX. Es denominado Yaw o guiñada
2. Girar el sistema OUVW un ángulo  $\theta$  con respecto al eje OY. Es denominado Pitch o cabeceo
3. Girar el sistema OUVW un ángulo  $\phi$  con respecto al eje OZ. Es denominado Roll o alabeo.

## 5. Fundamentación

---

### 5.1.2. Cinemática

El estudio de la hidrodinámica se puede dividir en dos partes: cinemática, la cual se refiere a aspectos del movimiento; y la kinetics, la cual analiza las fuerzas causadas por el movimiento [8].

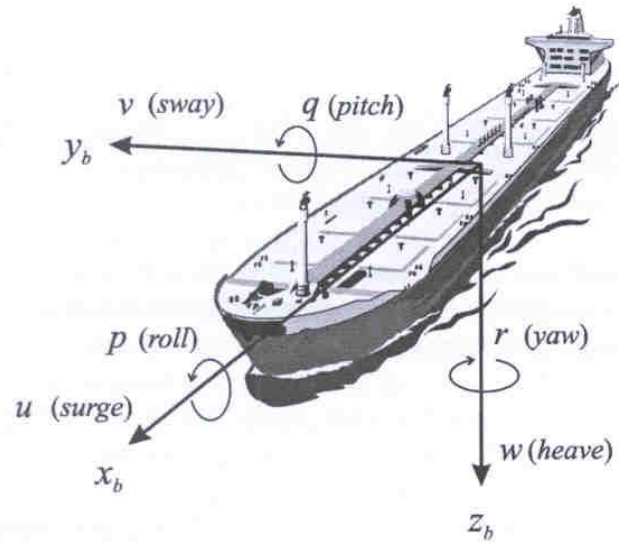


Figura 5.6 Variables de movimiento para vehículos marinos.

Notación considerando el formalismo de navegación SNAME (Society of Naval Architects and Marine Engineering 1950). Para determinar la posición y orientación de un vehículo marino [Figura 5.6] son necesarias seis coordenadas independientes (grados de libertad). Las tres primeras coordenadas y sus respectivas derivadas con respecto al tiempo, corresponden a la posición y traslación en los ejes “x”, “y” y “z”.

Las otras tres coordenadas y sus derivadas con respecto al tiempo son usadas para describir la orientación y los movimientos de rotación [Tabla 5.1].

## 5. Fundamentación

---

Grados de libertad		Fuerzas y momentos	Velocidades lineales y angulares	Posiciones y ángulos de Euler
1	Movimiento en dirección- $x$	X	$u$	X
2	Movimiento en dirección- $y$	Y	$v$	Y
3	Movimiento en dirección- $z$	Z	$w$	Z
4	Rotación sobre eje $x$	K	$p$	$\phi$
5	Rotación sobre eje $y$	M	$q$	$\theta$
6	Rotación sobre eje $z$	N	$r$	$\psi$

Tabla 5.1 Notación SNAME para vehículos marinos

### Sistemas de referencia

Al describir el movimiento de vehículos marinos con 6 grados de libertad es necesario conocer el sistema geográfico de referencia. NED ( $n$  - sistema). El sistema NED (North-East-Down por sus siglas en ingles) de coordenadas  $x_n$   $y_n$   $z_n$  es utilizado constantemente en la vida diaria. Es usualmente definido como el plano tangente en la superficie de la tierra moviéndose con el vehículo, pero con ejes apuntando a diferentes direcciones del eje del cuerpo del vehículo. Para este sistema el eje  $x$  va en dirección Norte, el eje  $y$  apunta al Este mientras que el eje  $z$  apunta al centro de la Tierra.

Para los vehículos marinos operando en un área local, la constante longitud y latitud, en el plano tangente del eje de la Tierra sobre la superficie es usado en la navegación. Este es usado como bandera de navegación y esta será por simplicidad denotada como “ $n$ -sistema”.

BODY ( $b$  – sistema). El sistema del cuerpo de referencia  $x_b$   $y_b$   $z_b$  es un sistema de coordenada de movimiento fijo al vehículo. La posición y orientación del vehículo son descritas por el sistema de referencia inercial, donde las velocidades angulares y lineales del vehículo pueden ser expresadas en este sistema de coordenadas del cuerpo. El origen de este sistema es usualmente escogido para coincidir con el centro de gravedad cuando este se

## 5. Fundamentación

---

encuentra en el plano principal de simetría o en cualquier otro punto. En vehículos marinos, los ejes del sistema  $x_b, y_b$  y  $z_b$  son elegidos para coincidir con el eje principal de inercia, y son usualmente definidos como:

- $x_b$  - eje longitudinal
- $y_b$  - eje transversal
- $z_b$  - eje normal

### Definiciones vectoriales

La siguiente notación es adoptada por los vehículos marinos para velocidades lineales y angulares cuando son descompuestas en un sistema de referencia:

$v_0^n$  = Velocidad lineal de punto 0 rotando en un sistema  $n$

$w_{eb}^n$  = Velocidad angular de sistema  $b$  con respecto al sistema  $e$  rotando en un sistema  $n$

La tabla se puede expresar en un sistema vectorial, de acuerdo a:

Posición ECF

$$p^e = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Ángulos de Euler

$$\Theta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}$$

Velocidad lineal

$$v_0^b = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

Velocidad angular

$$w_{nb}^b = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Fuerza

$$f_0^b = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \in \mathfrak{R}^3$$

Momentos

$$m_0^b = \begin{bmatrix} K \\ M \\ N \end{bmatrix} \in \mathfrak{R}^3$$

## 5. Fundamentación

---

El movimiento general de un vehículo marino con 6 grados de libertad es descrito con los siguientes vectores;

$$\eta = \begin{bmatrix} p^e \\ \Theta \end{bmatrix}, \quad v = \begin{bmatrix} v_0^b \\ w_{nb}^b \end{bmatrix}, \quad \tau = \begin{bmatrix} f_0^b \\ m_0^b \end{bmatrix}$$

Dónde:

- $\eta_{6 \times 6}$  denota el vector de posición y orientación. Está formado por el vector de posición  $p^e$  y  $\Theta$  es un vector de ángulos de Euler.
- $v_{6 \times 1}$  denota el vector de velocidad lineal y angular el cual es montado en el sistema de referencia del cuerpo.
- $\tau_{6 \times 1}$  es usado para describir las fuerzas y momentos que actúan en el vehículo

### Transformación de Ángulos de Euler

Los ángulos de Euler: roll ( $\phi$ ), pitch ( $\theta$ ), yaw ( $\psi$ ) pueden ser usados para determinar los componentes del vector de velocidad  $v_0^b$  en el sistema de referencia NED.

$R_b^n(\Theta)$  Denota la matriz de rotación de ángulos de Euler con argumento  $\Theta = [\phi, \theta, \psi]$ .

Entonces:

$$v_0^n = R_b^n(\Theta) v_0^b$$

### Rotaciones principales

---

## 5. Fundamentación

---

La rotación principal de matrices puede ser obtenida por:  $\lambda = [1, 0, 0]^T$ ,  $\lambda = [0, 1, 0]^T$  y  $\lambda = [0, 0, 1]^T$  correspondiente al eje  $x$ ,  $y$ ,  $z$ ;  $\beta = \phi$ ,  $\beta = \theta$  y  $\beta = \psi$ , en la expresión para  $R_{\lambda,\beta}$  dada por:

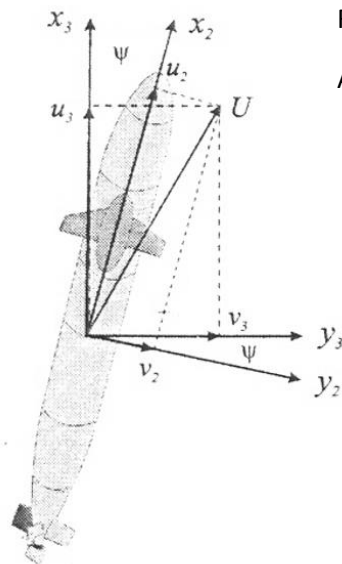
$$R_{\lambda,\beta} = I_{3 \times 3} + \text{sen}\beta S(\lambda) + (1 - \cos\beta) S^2(\lambda)$$

Dónde:

$I_{3 \times 3}$  Es la matriz identidad

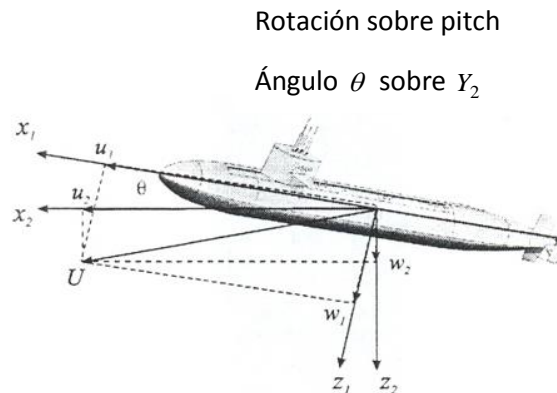
$S(\lambda)$  Es la matriz Skew-Symmetric

$S^2(\lambda) = \lambda\lambda^T - I_{3 \times 3}$  Dónde  $\lambda$  es un vector unitario



Rotación sobre yaw

Ángulo  $\psi$  sobre  $Z_3$

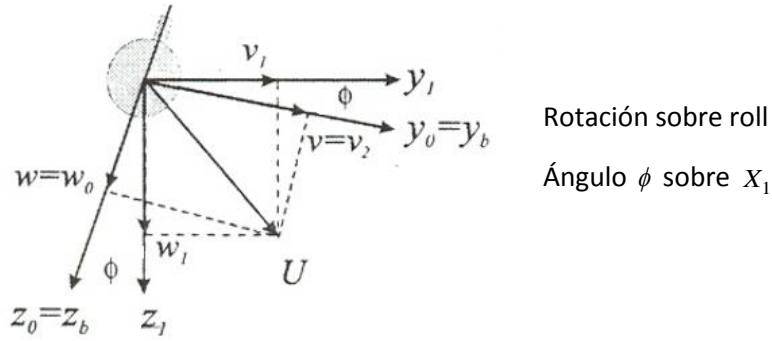


Rotación sobre pitch

Ángulo  $\theta$  sobre  $Y_2$

## 5. Fundamentación

---



Matrices de rotación

$$R(x, \phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\phi & -S\phi \\ 0 & S\phi & C\phi \end{bmatrix}, \quad R(y, \phi) = \begin{bmatrix} C\phi & 0 & S\phi \\ 0 & 1 & 0 \\ -S\phi & 0 & C\phi \end{bmatrix}, \quad R(z, \theta) = \begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Dónde:  $S \cdot = \text{sen}(\cdot)$ ,  $C \cdot = \text{cos}(\cdot)$

### Transformación de velocidades lineales

Para describir  $R_b^n(\Theta)$  por las tres principales rotaciones sobre el eje  $z$ ,  $y$  y  $x$ . En las aplicaciones de navegación y control es común usar la notación  $z$  y  $x$  desde el sistema  $n$  hasta el sistema  $b$  especificado en términos de los ángulos de Euler  $\phi$ ,  $\theta$  y  $\psi$  para las rotaciones.

La matriz es denotada  $R_b^n(\Theta) = R_b^n(\Theta)^T$ . La matriz transpuesta implica que el mismo resultado obtenido por un vector de transformación desde  $b$  hasta  $n$ . Esta secuencia de rotación es equivalente a:

$$R_b^n(\Theta) := R_{z,\psi} R_{y,\theta} R_{x,\phi}$$

Y la transformación inversa:

$$R_b^n(\Theta)^{-1} = R_b^n(\Theta) = R_{z,\psi}^T R_{y,\theta}^T R_{x,\phi}^T$$



## 5. Fundamentación

---

El sistema de coordenadas  $x_3y_3z_3$  se obtiene por la traslación del sistema de coordenadas NED  $x_ny_nz_n$  paralelo hasta que su origen coincida con el origen del sistema de coordenadas del cuerpo. El sistema de coordenadas  $x_3y_3z_3$  está rotado en un yaw ángulo ( $\psi$ ) sobre el eje  $z_3$ . Se da origen al sistema de coordenadas  $x_2y_2z_2$ . Las coordenadas del sistema  $x_2y_2z_2$  son rotadas en un pitch ángulo ( $\theta$ ) sobre el eje  $y_2$ . Se da origen al sistema de coordenadas  $x_1y_1z_1$ . Finalmente las coordenadas del sistema  $x_1y_1z_1$  son rotadas en un roll ángulo ( $\phi$ ) sobre el eje  $x_1$ .

La matriz de rotación se puede expandir a:

$$R_n^b(\Theta) = \begin{bmatrix} C\psi C\theta & -S\psi C\phi + C\psi S\theta S\phi & S\psi S\phi + C\psi S\theta C\phi \\ S\psi C\theta & C\psi C\phi + S\psi S\theta S\phi & -C\psi S\phi + S\psi S\theta C\phi \\ -S\theta & C\theta S\phi & C\theta C\phi \end{bmatrix}$$

El vector  $v_0^b$  de velocidades en el cuerpo puede ser descompuesto en el sistema NED de referencia, como:

$$\mathcal{P}^{\mathcal{B}} = R_b^n(\Theta)v_0^b$$

La transformación inversa de la velocidad es obtenida por definición como:

$$v_0^b = R_b^n(\Theta)^{-1} \mathcal{P}^{\mathcal{B}} = R_b^n(\Theta)^T \mathcal{P}^{\mathcal{B}}$$

### Transformación de velocidad angular

El vector de velocidad angular del cuerpo  $w_{nb}^b = [p, q, r]^T$  y el vector de Euler

$\Theta = [\phi, \theta, \psi]$  están relacionados a través de la matriz de transformación:

## 5. Fundamentación

---

$$\dot{\Theta} = T_{\Theta}(\Theta)w_b^n \quad (5.1)$$

El vector de velocidad angular en el cuerpo  $w_{nb}^b = [p, q, r]^T$  no puede ser integrado directamente para encontrar las coordenadas angulares en el sistema fijo. El vector  $\Theta = [\phi, \theta, \psi]^T$  representa las coordenadas generalizadas. La matriz de transformación  $T_{\Theta}(\Theta)$  puede ser derivada en diferentes maneras:

$$w_b^n = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R_{x,\phi}^T \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_{x,\phi}^T R_{y,\theta}^T \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} := T_{\Theta}^{-1}(\Theta)\dot{\Theta}$$

La relación es verificada mediante:

$$T_{\Theta}^{-1}(\Theta) = \begin{bmatrix} 1 & 0 & -S\phi \\ 0 & C\phi & C\theta S\phi \\ 0 & -S\phi & C\theta C\phi \end{bmatrix}$$

⇓

$$T_{\Theta}(\Theta) = \begin{bmatrix} 1 & S\phi T\theta & C\phi T\theta \\ 0 & C\phi & -S\phi \\ 0 & S\phi/C\theta & C\phi/C\theta \end{bmatrix}$$

Donde  $S\cdot = \text{sen}(\cdot)$ ,  $C\cdot = \text{cos}(\cdot)$ ,  $T = \text{tan}(\cdot)$

Expandiendo la Ecuación 5.1 en ecuaciones con componentes de Ángulos de Euler;

$$\dot{\phi} = p + q \text{sen} \phi \tan \theta + r \cos \phi \tan \theta$$

## 5. Fundamentación

---

$$\dot{\phi} = q \cos \phi - r \sin \phi$$

$$\dot{\psi} = q \frac{\sin \phi}{\cos \theta} + r \frac{\cos \phi}{\cos \theta} \quad \theta \neq \pm 90^\circ$$

Se hace notar que  $T_{\Theta}(\Theta)$  no está definida para ángulos  $\theta = \pm 90^\circ$ . En el caso de vehículos marinos esto no es problema debido a que un robot submarino se trabaja con ángulos  $\theta$  pequeños.

### 5.2 Modelos matemáticos

#### 5.2.1 Modelo Cinemático

En resumen el modelo cinemático de un vehículo submarino con seis grados de libertad puede ser representado de la siguiente forma, según Fossen [8]:

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

Substituyendo los valores de  $R_b^n(\Theta)$  y  $T_{\Theta}(\Theta)$  :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} R_b^n(\Theta) & 0_{3 \times 3} \\ 0_{3 \times 3} & T_{\Theta}(\Theta) \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix}$$

## 5. Fundamentación

---

Lo cual es equivalente a:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} C\psi C\theta & -S\psi C\phi + C\psi S\theta S\phi & S\psi S\phi + C\psi C\phi S\theta & 0 & 0 & 0 \\ S\psi C\theta & C\psi C\phi + S\phi S\theta S\psi & -C\psi S\phi + S\theta S\psi C\phi & 0 & 0 & 0 \\ -S\theta & C\theta S\phi & C\theta C\phi & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & S\phi T\theta & C\phi T\theta \\ 0 & 0 & 0 & 0 & C\phi & -S\phi \\ 0 & 0 & 0 & 0 & S\phi / C\theta & C\phi / C\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix}$$

### 5.2.2 Modelo Hidrodinámico

#### Mecánica Newtoniana y Lagrangiana

En esta sección se presenta la ecuación de movimiento hidrodinámico con 6 DOF de un vehículo marino, se expresa como [8]:

$$M\dot{v} + C(v)v + D(v)v + g(\eta) = \tau \quad (5.2)$$

Dónde:

- $M$  = Matriz de inercia (incluida la masa adjunta)
- $C(v)$  = Matriz de Colioris (incluyendo términos adjuntos)
- $D(v)$  = Matriz de amortiguamiento
- $g(\eta)$  = Vector de fuerzas y momentos reactivos
- $\tau$  = Vector de entradas de control

## 5. Fundamentación

---

### Ecuación de movimiento

La ecuación de movimiento de un vehículo marino puede ser representada de dos maneras:

1. Representada en el eje de referencia del vehículo
2. Representada en el eje de referencia de la tierra

La representación de la ecuación de movimiento referenciada al eje del vehículo se puede escribir como Ecuación 5.2

Dónde:

$$\begin{aligned}M &= M_{RB} + M_A \\C(v) &= C_{RB}(v) + C_A(v) \\D(v) &= D_P(v) + D_S(v) + D_W(v) + D_M(v)\end{aligned}\tag{5.3}$$

En el caso de la representación en el eje de referencia terrestre, es obtenido aplicando las siguientes transformaciones cinemáticas y asumiendo que  $J(\eta)$  es una matriz no singular

$$\begin{aligned}\dot{\eta} &= J(\eta)v \quad \Leftrightarrow \quad v = J^{-1}(\eta)\dot{\eta} \\ \dot{\eta} &= J(\eta)\dot{\eta} + \dot{J}(\eta)v \quad \Leftrightarrow \quad \dot{\eta} = J^{-1}(\eta)[\dot{\eta} - \dot{J}(\eta)J^{-1}(\eta)\dot{\eta}]\end{aligned}\tag{5.4}$$

Eliminando  $v$  y  $\dot{\eta}$  de la Ecuación 5.2

$$\begin{aligned}M_\eta(\eta) &= J^{-T}(\eta)MJ^{-1}(\eta) \\ C_\eta(v, \eta) &= J^{-T}(\eta)[C(v) - MJ^{-1}(\eta)\dot{J}(\eta)]J^{-1}(\eta)\end{aligned}$$

## 5. Fundamentación

---

$$D_{\eta}(v, \eta) = J^{-T}(\eta) D(v) J^{-1}(\eta)$$

$$g_{\eta}(\eta) = J^{-T}(\eta) g(\eta)$$

$$\tau_{\eta}(\eta) = J^{-T}(\eta) \tau \quad (5.5)$$

Finalmente la representación en el eje terrestre es:

$$M_{\eta}(\eta) \ddot{\eta} + C_{\eta}(v, \eta) \dot{\eta} + D_{\eta}(v, \eta) \eta + g_{\eta}(\eta) = \tau_{\eta} \quad (5.6)$$

### Amortiguamiento hidrodinámico

El amortiguamiento hidrodinámico para un vehículo marino es principalmente causado por:

$D_S(v)$  = fricción lineal en la superficie debida a las capas laminares del fluido y fricción cuadrática debido a las capas turbulentas del fluido. La fricción lineal debido a la teoría de las capas laminares del fluido es importante cuando consideramos el movimiento en baja frecuencia del vehículo, por lo tanto este efecto deberá ser considerado cuando se diseña el control del sistema.

$D_M(v)$  = amortiguamiento debido a las turbulencias causadas por la forma del cuerpo.

Este amortiguamiento puede ser modelado con la siguiente ecuación:

$$f(U) = -\frac{1}{2} \rho C_D(Rn) A |U| U \quad (5.7)$$

## 5. Fundamentación

---

Donde  $U$  es la velocidad del vehículo,  $A$  es el área del cuerpo,  $C_D(Rn)$  es el coeficiente de arrastre que depende del área del cuerpo y del número de Reynolds y  $\rho$  es la densidad del fluido. La ecuación anterior es reconocida como la ecuación de Morrison.

$D_P(v)$  = amortiguamiento potencial inducido causado por la fuerza de la oscilación del cuerpo.

En vehículos submarinos operando a una gran profundidad la contribución del amortiguamiento potencial es despreciable en comparación con los vehículos de superficie donde el efecto de amortiguamiento potencial es significativo.

$D_W(v)$  = amortiguamiento debido al arrastre del cuerpo sobre las olas.

Esta clase de amortiguamiento es interpretado como la resistencia de los vehículos de superficie avanzando sobre las olas. Es normal que este tipo de amortiguamiento sea muy importante cuando se hable de barcos.

En consecuencia la ecuación de amortiguamiento hidrodinámico puede ser escrita como la suma de los componentes anteriores:

$$D(v) \cong D_P(v) + D_S(v) + D_W(v) + D_M(v) \quad (5.8)$$

Dónde  $D(v)$  satisface la siguiente propiedad:

Para un cuerpo rígido moviéndose a través de un fluido ideal la ecuación de amortiguamiento hidrodinámico será real, no simétrica y estrictamente positiva.

En una implementación práctica es difícil determinar todos los elementos de la ecuación (matriz de amortiguamiento  $D$ ). Se sugiere para el presente trabajo, utilizar la siguiente aproximación [8]:

## 5. Fundamentación

---

### Vehículos de superficie (barcos)

Para barcos a baja velocidad podemos desacoplar el plano horizontal y el plano vertical, y se asume que el barco solo se mueve en plano horizontal. Las fuerzas y momentos de amortiguamiento linealizado pueden ser escritas como:

$$D(v) = \begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & N_v & N_r \end{bmatrix} \quad (5.9)$$

Notar que  $Y_r \neq N_v$

### Vehículos submarinos

En general, la ecuación de amortiguamiento de un vehículo submarino moviéndose en 6 grados de libertad a alta velocidad es altamente no lineal y acoplada. Sin embargo, una aproximación burda es asumir que el vehículo realiza un movimiento no acoplado, y los términos no lineales son despreciados. Esto sugiere una estructura diagonal de la matriz  $D$  con solo términos de amortiguamiento lineal y cuadrático en la diagonal, esto es:

$$D(v) = -diag\{X_u, Y_v, Z_w, K_p, M_q, N_r\} \\ -diag\{X_{u|u}|u|, Y_{v|v}|v|, Z_{w|w}|w|, K_{p|p}|p|, M_{q|q}|q|, N_{r|r}|r|\} \quad (5.10)$$

### 5.3 Leyes de control

El uso de una ley de control en el modelo permitirá minimizar el error y ajustar los valores de entrada al sistema de ecuaciones obteniendo un resultado de comportamiento aproximado a la realidad.



## 5. Fundamentación

---

### 5.3.1 Control PID (Proporcional, Integral y Derivativo)

El control PID (Proporcional, Integral y Derivativo) es un mecanismo de control por realimentación que es ampliamente usado en sistemas de control industrial. Este calcula la desviación o error entre un valor medido y un valor deseado.

El algoritmo del control PID requiere de tres parámetros distintos: el proporcional, el integral, y el derivativo. El valor Proporcional está relacionado con el error actual. El Integral corresponde al de los errores pasados y el Derivativo consiste en una predicción de los errores futuros. La suma de estos tres valores es usado para ajustar al proceso por medio de un elemento de control como es la posición, en el caso del ROV correspondiente al vector de propulsión en el eje y.

La ecuación que determina el control PID [5]:

$$\tau = K_p e(kh\Delta_T) + \frac{K_p \Delta_T}{T_I} \sum_{h=1}^k \frac{e(h\Delta_T) + e((h-1)\Delta_T)}{2} + K_p T_D [e(h\Delta_T) - e((h-1)\Delta_T)] \quad (5.11)$$

Dónde:

- $\Delta_T$  Representa el tiempo de muestra,
- $e(k \Delta_T)$  Es el error medido en el tiempo muestra  $k\Delta_T$
- $K_P$  Es la ganancia proporcional
- $T_I$  Es la integral del tiempo
- $T_D$  Es la derivada del tiempo

## 5. Fundamentación

---

### 5.3.2 Control Deslizante de segundo Orden

Esta ley está basada en un controlador de estructura variable de segundo orden donde se elimina el fenómeno de vibración llamado “chattering” presentado en este tipo de técnicas de control de primer orden [9].

La referencia nominal considerada:

$$\dot{\eta}_r = \dot{\eta}_d - \alpha \tilde{\eta} + S_d - K_i \int_0^t \text{sign}(S_q(\sigma)) d\sigma \quad (5.12)$$

Dónde:

$\tilde{\eta} = \eta - \eta_d$  Representa la posición de rastreo del error

$\eta_d$  Denota la trayectoria deseada

$\alpha$  y  $K_i$  Son matrices diagonales positivas de ganancias de  $n \times n$

$\text{sign}(S_q)$  Es la función signo del vector  $x$

$\tilde{\eta}$  Representa el error de seguimiento interpretada por

Además podemos definir:

$$S_q = S - S_d$$

$$S = \dot{\tilde{\eta}} + \alpha \tilde{\eta}$$

$$S_d = S(t_0) e^{-k_a t}$$

## 5. Fundamentación

---

Dónde:

$$\begin{array}{ll}
 k_\alpha & \text{Es una constante mayor a 0} \\
 S(t_0) & \text{Permanece con el valor de } S(t) \text{ en } t=0
 \end{array}$$

La variable del error extendido está dada por:

$$S_r = \dot{\eta} - \dot{\eta}_r \quad (5.13)$$

Sustituyendo (5.12) en (5.13) se tiene la siguiente ecuación, la cual será usada en la implementación del control.

$$S_r = S_q + K_i \int_0^t \text{sign}(S_q(\sigma)) d\sigma \quad (5.14)$$

Considerando la dinámica del vehículo, la ley de control estará dada por:

$$\tau_\eta = -K_d S_r \quad (5.15)$$

Dónde:

$$K_d \quad \text{Es una matriz } n \times n \text{ de ganancia positiva.}$$

Substituyendo los valores de  $S_r$  (5.14) se obtiene la siguiente ecuación:

$$\tau_\eta = -\underbrace{K_d \alpha \tilde{\eta}}_P - \underbrace{K_d \dot{\tilde{\eta}}}_D - \underbrace{K_d K_i \int_0^t \text{sign}(S_q) d\sigma}_0 \quad (5.16)$$

*Sliding*

## 6. Metodología

---

## 6. Metodología

### 6.1. Metodología de desarrollo

La metodología utilizada para la adaptación de los modelos matemáticos y leyes de control en el lenguaje de programación fue la considerada en parte del trabajo de tesis de [7], el cual nos permite implementar el modelo matemático en ambiente virtual, considerando el lenguaje de programación Visual Studio C# con librerías adicionales para obtener una solución del modelo en tiempo real visualizando su comportamiento mediante gráficos que representan una acercamiento al modelo del ROV [Figura 6.1].

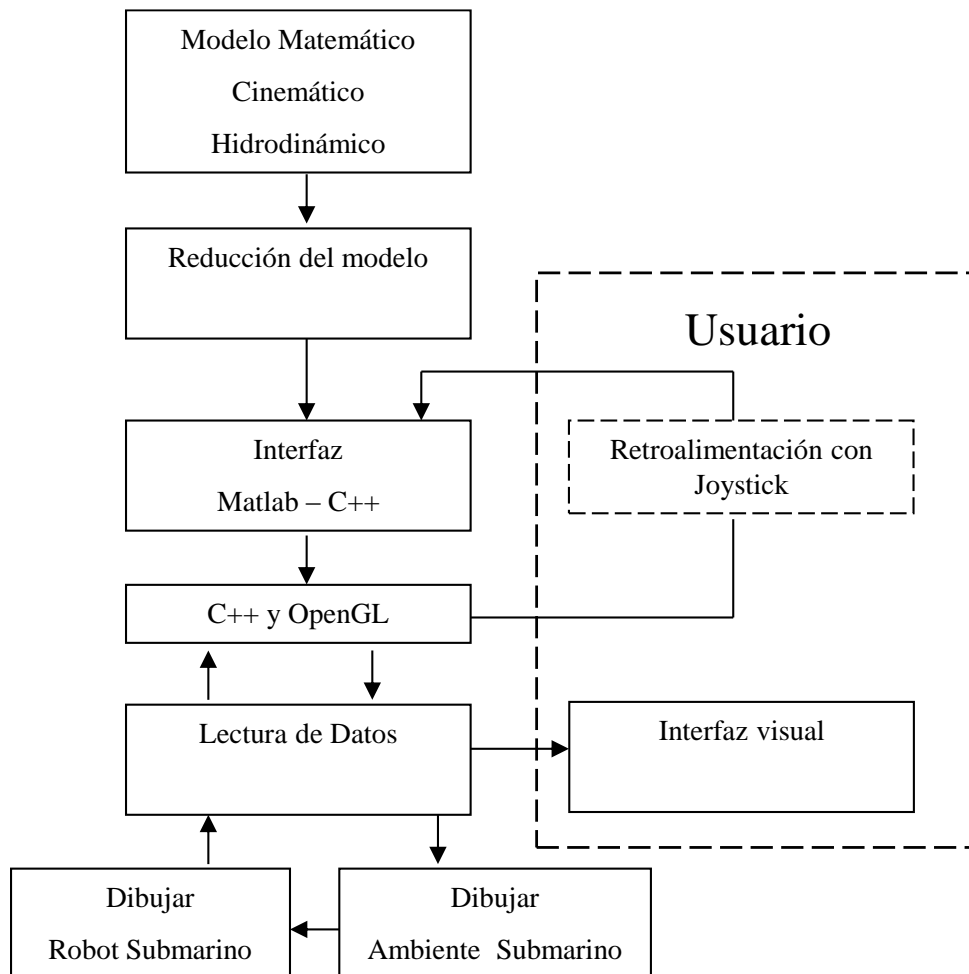


Figura 6.1 Diagrama de bloques propuesto en el desarrollo.

## 6. Metodología

---

### 6.2. Configuraciones adicionales

Se realizó una adaptación del trabajo de tesis “Towards to the development of a virtual environment for operators training for the ROV KAXAN from CIDESI” [6], el cual presenta el primer acercamiento a la modelación de un ROV en un ambiente virtual, donde se implementa el control PID adicional a la solución del modelo matemático en tiempo real y el comportamiento del ROV.

Las características principales del sistema informático adaptado son:

- Ejecución simultánea de solución de modelo matemático y visualización en pantalla de su comportamiento en ambiente virtual.
- Adaptación de un Control Deslizante de segundo orden al ROV virtual.
- Operación del sistema mediante accesorio joystick de fácil manejo.
- Lenguaje de desarrollo en Visual Studio C#.

El desarrollo en Visual Studio C#, permite la inclusión de librerías como OpenTK que apoya al uso de librerías Gráficas trabajando en conjunto con la gama de librerías OpenGL, OpenCL y OpenAL que permiten el uso de funciones adicionales utilizadas para el ambiente gráfico.

### 6.3. Control deslizante de segundo orden

La implementación del control deslizante de segundo orden, se realizó mediante la creación de la función que da solución a la ecuación (5.16), obteniendo el código:

```
// Control sliding de segundo orden
private double control_force_s(double e_val, double ed_val, double t_val, double dt_val,
    int count_val)
{
    double S, Sd, Sq, Sqi, Force = 0;
    int bandera = 0;
```

## 6. Metodología

---

```
S = ed_val + (alpha * e_val); // Se cambia le signo de la operación

if (t_val == 0.01)
{
    S_to = S;
}

Sd = S_to * (Math.Exp(-kapa * t_val));

Sq = S - Sd;

sq_array[count_val] = matrix_op.funcSign(Sq); // con función signo

Sqi = matrix_op.int_k(ref sq_array, ref sqi_array, dt_val);
sqi_array[count_val] = Sq;

Force = -Kdz * (Sq + Sq*Kiz);

if (t_val >= 7.1)
{
    bandera = 1;
}

if (Force > 0) Force += .75;
else if (Force < 0) Force -= .75;

if (Force > 5) Force = 5;
else if (Force < -5) Force = -5;

return Force;
}
```

Mediante esta función implementada en la solución informática, se da solución en tiempo real al comportamiento del ROV en la fuerza de propulsión correspondiente al eje  $z$  del ROV.

## 6. Metodología

---

### 6.4. Parámetros y constantes

En la solución del modelo se determinaron las condiciones iniciales y constantes tanto para el ROV así como para el Control Deslizante de Segundo Orden.

Obteniendo para el control deslizante segundo orden en eje  $z$ , después de realizar varias corridas hasta sintonizar las variables que nos permitieron resultados aceptables en el comportamiento del ROV, se tienen:

$$\alpha = 0.5,$$

$$K = 1,$$

$$K_{dz} = 100,$$

$$S(t_o) = \text{el valor de } S(t) \text{ en el tiempo } t=0,$$

$$K_{iz} = 0.05$$

### 6.5. Consideraciones de ambiente virtual

El objetivo de esta tesis es programar un simulador para el ROV KAXAN que sea capaz de tener un comportamiento similar al de un robot submarino real. Algunos aspectos importantes del programa serán:

- El programa debe tener un tiempo de ejecución real.
- Las entradas de las interfaces serán similares a las entradas de interfaz del ROV KAXAN.
- Un entorno visual de salida del robot será elegido por el usuario.
- El usuario tendrá la capacidad de elegir dos tipos de Control: control PID o control deslizante de segundo orden.

Para el desarrollo del simulador virtual se eligió Visual Studio C# (C Sharp). Este lenguaje fue seleccionado de entre otros debido a:

- Orientado a objetos, simple, actual y de propósitos generales.

## 6. Metodología

---

- Portabilidad del código fuente, especialmente para aquellos programadores que se encuentran familiarizados con C y C++.
- Las aplicaciones en C# pretenden ser económicas con respecto a la potencia de procesamiento y los requisitos de memoria, pero el lenguaje no pretendía competir directamente con el rendimiento y el tamaño con C o Lenguaje ensamblador.

Es fácil y rápido programar en C# mejor que en C++, el lenguaje es más fácil de usar que el tradicional C++, a pesar que un programa en C++ se ejecuta más rápido que un programa en C#, un programa ligero (en términos de tamaño) como este, no presentó ninguna velocidad durante su funcionamiento. Otro aspecto importante es que C# está orientado a objetos, por lo tanto el código y las clases creadas para este programa son adecuados para su reutilización en otros proyectos.

Para la creación del ambiente virtual, el modelo del ROV fue creado usando las librerías de OpenGL (Open Graphics Library), que es una API para renderizar gráficos, usualmente en 3D. Ya que las tarjetas gráficas suelen incluir una implementación OpenGL, y la especificación OpenGL no es una plataforma específica, gran parte del proceso gráfico se realiza directamente por la tarjeta gráfica.

Existen diferentes bibliotecas (plug-ins) para poder utilizar las funciones de OpenGL y formas creadas en C#, una de esas librerías es OpenTK. Esta librería permite a los programadores en C# utilizar OpenGL, OpenAL y OpenCL e incluye una gran cantidad de funciones útiles como: funciones matemáticas, fuentes, etc. También puede trabajar con la tarjeta gráfica para que gran parte del trabajo de cómputo sea realizado por el hardware gráfico, en lugar del procesador haciendo una aplicación más agradable y rápida.

Para trabajar con OpenTK, es necesario incluir esta biblioteca al proyecto y luego agregar un control tipo ventana al formulario. OpenTK ofrece soporte para diferentes dispositivos de entrada, como es el joystick, sin embargo la información sobre dicho soporte es escasa por



## 6. Metodología

---

lo que el TaoFrame se utilizó en su lugar. TaoFrame es una biblioteca antigua que OpenTK, pero esta ofrece soporte para diferentes dispositivos de joystick.

### 6.6. Configuración joystick

Para realizar el control del ROV en este proyecto se seleccionó un joystick genérico el cual es más portable ofreciendo opciones para conectarse como es USB o por Wireless vía Bluetooth [Figura 6.2a] y [Figura 6.2b]. Fue necesaria la actualización del driver del dispositivo, debido a que la primera programación fue realizada bajo un sistema operativo a 32bits y actualmente se está trabajando bajo un sistema operativo a 64bits. Se utilizó la herramienta de Better DS3 para realizar la configuración del dispositivo en sistema operativo Windows usando USB o Bluetooth. En esta implementación el joystick fue configurado como un control de Xbox 360, lo cual permitirá usar diversas funciones. Usando esta configuración el dispositivo tiene 16 botones, 5 ejes y un interruptor.

Como se vio en la sección 6.5, la librería TaoFramework es usada para obtener información acerca del joystick, existen dos maneras de hacerlo. La primera consiste en tener un hilo en el programa que este constantemente leyendo las banderas de joystick. Esas banderas son activadas cuando un botón es presionado o se mueve un eje. La segunda forma consiste en recuperar la información del joystick leyendo la información de entrada a petición, no de una manera continua. Esta segunda opción fue seleccionada debido a que reduce la complejidad del programa y porque no es necesario leer continuamente el estado del joystick, este puede ser leído cada vez que la escena es dibujada.

El programa inicia con las invocaciones al joystick, y busca todos los dispositivos conectados al equipo y si al menos uno está conectado el programa lo selecciona.

Búsqueda de Joystick activo:

```
//Joystick  
try
```

## 6. Metodología

---

```
{
    Sdl.SDL_Init(Sdl.SDL_INIT_JOYSTICK);
    num_joysticks = Sdl.SDL_NumJoysticks();
}
catch (Exception EX) { }
if (num_joysticks > 0)
{
    Joystick = Sdl.SDL_JoystickOpen(0);
    textBox3.Text = "Joystick Found! Number of Axes:
    "+Sdl.SDL_JoystickNumAxes(Joystick).ToString()+"\r\n";
}
```

Se creó una estructura de joystick para guardar las características del joystick, por ejemplo, el valor devuelto por el joystick en su posición máxima del eje y también en su posición mínima del eje. Para la lectura de la información actual del joystick se invoca a una actualización, entonces cada valor de interés se toma y se procesa. En la [Figura 6.2c] el eje  $x$  del joystick controla el propulsor el cual mueve al ROV hacia los lados y también es uno de los que hacen que el ROV gire a lo largo del eje Z. El eje  $y$  del joystick controla el voltaje en los propulsores que mueven al ROV hacia adelante y hacia tras, estos propulsores están localizados en el lado posterior e inferior del ROV. El eje Z cambia el valor de  $Z_d$  deseada. El control de profundidad es el encargado de suministrar el voltaje necesario para mover al ROV de posición. La  $Z_d$  deseada no puede ser menor que 0, por lo que la restricción dentro del software fue incluida.

En la [Figura 6.2d] el  $x_2$  gira la cámara a lo largo del centro del objeto, esta rotación tiene un radio de tres metros desde el centro del modelo. Finalmente el eje  $y_2$  mueve la cámara a lo largo del eje Z del ambiente. El botón de cruz del joystick es usado para restablecer la cámara a su posición inicial desde el centro del objeto. Vale la pena mencionar que la cámara en la vista Ortogonal está controlada por el eje  $x_2$  y  $y_2$ , pero en la cámara en la vista Perspectiva.

## 6. Metodología

---



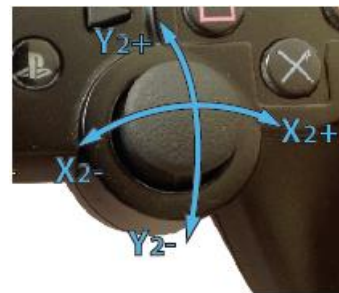
a.



b.



c.



d.



e.



f.

Figura 6.2 Joystick DualShock 3.

Al final del programa es siempre importante cerrar el uso del joystick, esto es necesario agregando la línea de código:

```
Sdl.SDL_JoystickClose(Joystick);
```

## 6. Metodología

---

### 6.7. Solución de ecuaciones.

El problema principal en los modelos matemáticos, es la solución de las ecuaciones que contienen cálculos diferenciales. Es realmente complejo realizar la programación para resolver la ecuación en tiempo real para cada cambio de posición en el ambiente del ROV, este tema fue un reto abordado en este trabajo.

La intención es crear en C# funciones para calcular los movimientos del ROV y así emular como se realiza en Matlab y Simulink.

En la solución de las ecuaciones (5.16) para integrar y diferenciar (el cálculo diferencial se aplica en la solución del control) y debido a que se está trabajando con un proceso discreto, diferentes métodos numéricos son necesarios para ambas operaciones.

Existen diferentes métodos para integrar [2], pero son dos los más conocidos. El primero es el método trapezoidal.

$$\int_a^b f(x)dx \approx (b - a) * \frac{f(a) + f(b)}{2} \quad (6.1)$$

La ventaja de este método es que solo se necesitan dos puntos para conocer el valor de la integral y si el paso del tiempo es muy corto se obtendrá una aproximación muy buena. Para grandes periodos de tiempo una propiedad de la integral puede usarse.

## 6. Metodología

---

$$\begin{aligned}\int_0^{a_2} f(x)dx &= \int_0^{a_1} f(x)dx + \int_{a_1}^{a_2} f(x)dx \therefore \\ &\text{if } A_1 = \int_0^{a_1} f(x)dx \\ A_2 &= \int_0^{a_2} f(x)dx = A_1 + \int_{a_1}^{a_2} f(x)dx \\ &\vdots \\ A_n &= \int_0^{a_n} f(x)dx = A_{n-1} + \int_{a_{n-1}}^{a_n} f(x)dx\end{aligned}$$

El segundo método es el de la regla de Simpson, donde la función  $f(x)$  se aproxima por un polígono cuadrático en el intervalo  $[a, b]$ .

$$\int_a^b f(x)dx \approx \int_a^b P(x)dx = \frac{b-a}{6} \left[ f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \quad (6.2)$$

El problema con la regla de Simpson es que el valor de la función en el punto medio  $f\left(\frac{a+b}{2}\right)$  es necesario.

Por lo anterior se elige el método trapezoidal debido a que contiene pocas operaciones y es fácil de calcular, los resultados demostraron que es suficientemente preciso para las necesidades del problema.

El código para la implementación del método seleccionado, es:

```
//Integral
public static double[,] int_m(ref ArrayList func, ref ArrayList func1, double time)
{
    double[,] integrate;
    double[,] f0, f1;
    int count = func.Count;
```

## 6. Metodología

---

```
if (count != 1)
{
    f0 = (double[,])func[count - 2];
    f1 = (double[,])func[count - 1];
    //Método Trapezoidal (f1+f0)*dt/2
    integrate = mul_k(sum_m(f1, f0,true), .5*time);
    //Agregar el valor de la integral a los valores anteriores,
    // Esto es igual a la integración de 0 al valor actual
    integrate = sum_m(integrate, (double[,])func1[count - 2],true);
}
else
    //Si existe un solo valor en el arreglo,
    //El resultado de la integración es f(0)*dt/2
    return mul_k((double[,])func[0],time / 2);
return integrate;
}

//Derivada
public static double der_k(ref ArrayList func, ref ArrayList time, double d_time)
{
    double derivate = 0;
    int count = func.Count;
    double t0, t1, t2, f0, f1, f2;

    if (count == 1)
    {
        //Esta función es similar a la función en Matlab
        /* f0 = (double)func[0];
        derivate = f0 / d_time;
        return derivate;
        */
        //Cuando es un solo elemento el valor es:
        return 0;
    }
    else if (count <= 2)
    {
        f0 = (double)func[0];
        f1 = (double)func[1];
        derivate = (f1 - f0) / (d_time);
    }
}
```

---

## 6. Metodología

---

```
    }  
    else  
    {  
        t0 = (double)time[count - 3];  
        t1 = (double)time[count - 2];  
        t2 = (double)time[count - 1];  
        f0 = (double)func[count - 3];  
        f1 = (double)func[count - 2];  
        f2 = (double)func[count - 1];  
  
        double p1 = ((t2 - t1) * (f0) / ((t0 - t1) * (t0 - t2)));  
        double p2 = ((t2 - t0) * (f1) / ((t1 - t0) * (t1 - t2)));  
        double p3 = (((2 * t2) - t1 - t0) * (f2) / ((t2 - t1) * (t2 - t0)));  
  
        derivate = p1 + p2 + p3;  
    }  
    return derivate;  
}
```

### 6.8. Propulsores

El ROV KAXAN tiene 4 propulsores, dos colocados en la parte posterior para proporcionar movimientos: hacia adelante, hacia atrás y giros por diferencias de velocidades en los mismos, además cuenta con un propulsor lateral para proporcionar movimiento laterales, y el último en la posición vertical para controlar la profundidad del ROV [4]. La ubicación de estos propulsores se puede ver en la [Figura 6.3].

## 6. Metodología

---

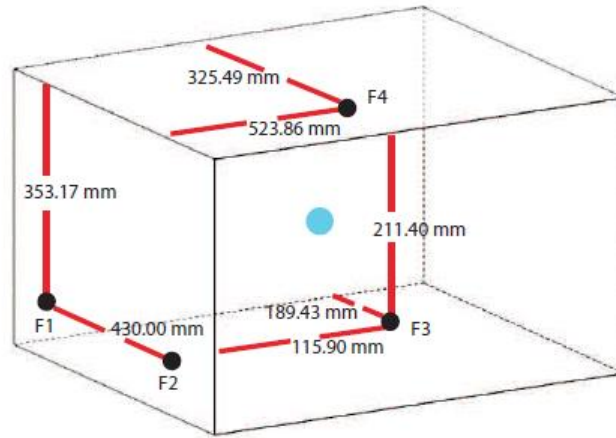


Figura 6.3 Ubicación de los propulsores del ROV Kaxan.

Los propulsores seleccionados para mover el ROV son propulsores brushless de CD, fabricados por Tecnam. Los dos propulsores en la parte trasera son del modelo 520 y los otros dos son propulsores del modelo 540. Fue usado un método gráfico para obtener un modelo matemático de los dos modelos de los propulsores y de esta manera incorporar su comportamiento al simulador virtual.

### 6.8.1. Propulsor 520

Revisando la hoja de datos del modelo del propulsor 520 [Anexo A.1], primero se observa que el voltaje de operación de control es  $\pm 5v$ , ahora el gráfico empuje, libras contra voltaje de control muestra el Comportamiento del propulsor hacia adelante y hacia atrás. El modo hacia adelante (avance) presenta una tendencia bastante incremental durante el rango de 0.75v a 4.75v.

### 6.8.2. Propulsor 540

Analizando la hoja de datos [Anexo A.2] se puede observar que ambos modos, hacia adelante y hacia atrás, del modelo 540 tienen la misma inconsistencia de crecimiento en su



## 6. Metodología

---

comportamiento tal como el modo hacia atrás del modelo 520. El mismo enfoque fue usado para ambos modos.

### 6.9. Plataforma de entrenamiento

El producto final de este proyecto es el simulador o la plataforma de entrenamiento. En esta sección se presentarán las especificaciones sobre el programa, por ejemplo; cómo se visualiza el programa, qué parámetros son necesarios para su ejecución así como los resultados obtenidos durante un proceso de ejecución.

#### 6.9.1. Interfaz Gráfica de Usuario (GUI)

La plataforma de entrenamiento GUI (Graphical User Interface por sus siglas en inglés) [Figura 6.4] fue diseñada para proporcionar al operador toda la información para su entrenamiento. En esta se puede leer la posición del ROV en el simulador, se pueden establecer los parámetros para una corriente oceánica, las ganancias en los controladores, la posición de la cámara, el control de voltaje de los propulsores, seleccionar el tipo de Control, observar las gráficas de resultados entre otros.

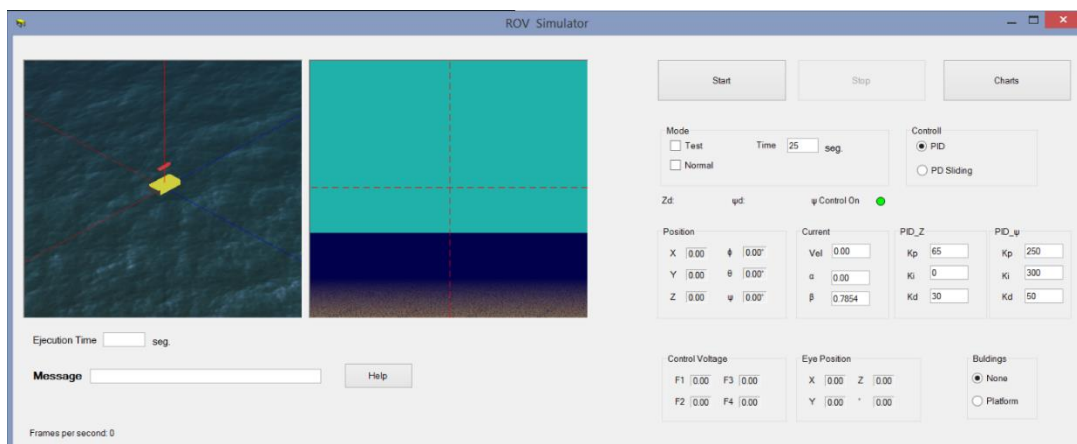


Figura 6.4 Interfaz Gráfica de Usuario (GUI).

## 6. Metodología

---

En la [Figura 6.5] se pueden ver ambas visualizaciones gráficas, la vista ortogonal [Figura 6.5a] nos da la posibilidad de mover la cámara alrededor del objeto para seleccionar la vista deseada, en esta opción la perspectiva no es influenciada en la visualización, todo objeto tiene el mismo tamaño, este tipo de vista es buena para trabajos técnicos, se utiliza principalmente en software de CAD. La segunda pantalla [Figura 6.5b] es la vista en perspectiva, aquí cambia el tamaño de los objetos de acuerdo con la distancia entre el objeto y la cámara. Esta es una perspectiva más realista y de hecho representa cómo el operador vería el océano desde el punto de vista de la cámara del ROV. En la vista en perspectiva, el movimiento de la cámara no está configurado cuando el ROV se está moviendo.

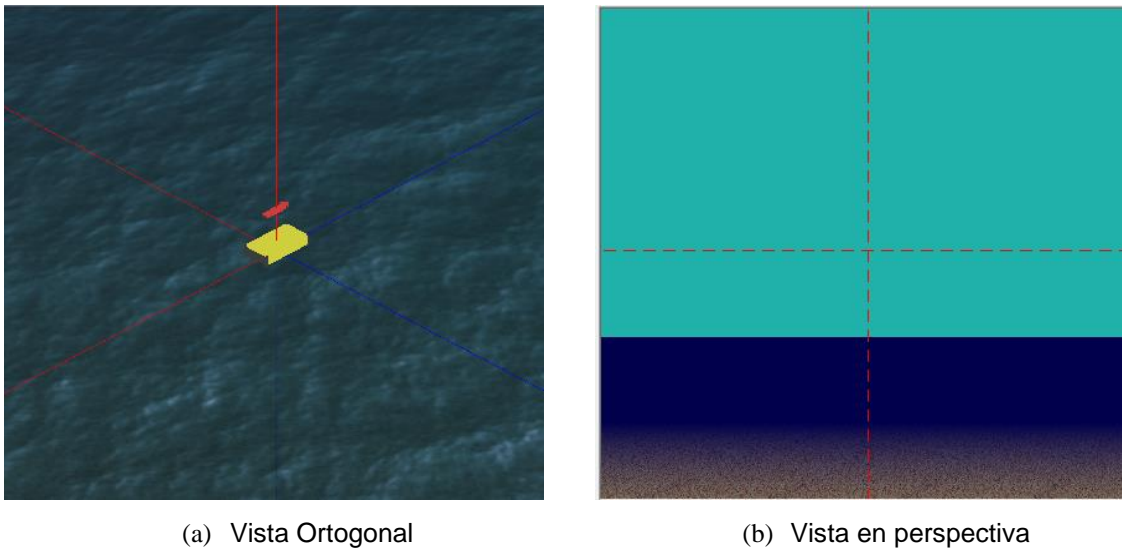


Figura 6.5 Área de Trabajo.

En esta plataforma de entrenamiento es posible configurar el valor de la corriente submarina.

La [Figura 6.6] se muestra la interface para configurar los ángulos  $\alpha$  y  $\beta$  de la corriente y su velocidad. La velocidad está dada en  $\frac{m}{s}$  y los ángulos en radianes.

## 6. Metodología

---

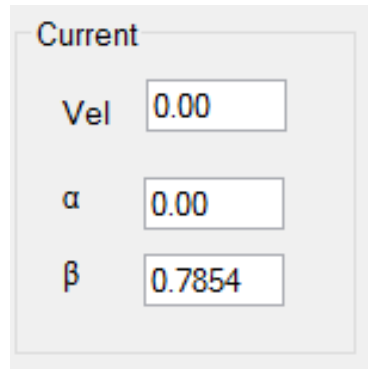


Figura 6.6 Corriente Marina.

En este panel [Figura 6.7] la profundidad deseada  $Z_d$  para el control PID y las ganancias del control pueden ser configuradas. Los valores predeterminados son:  $K_p = 65$ ,  $K_i = 0$  y  $K_d = 30$ .

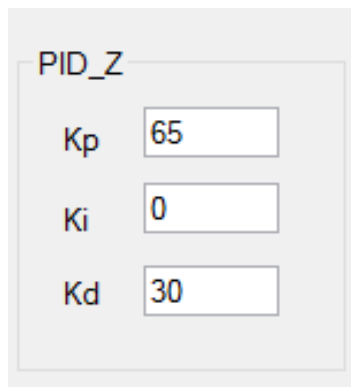
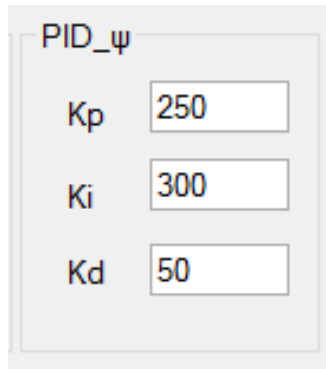


Figura 6.7 Área de control de profundidad PID.

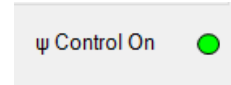
El controlador de dirección  $\psi$  con PID incluye el área para las ganancias del controlador [Figura 6.8a] y un LED [Figura 6.8b] que muestra si se encuentra activado el control PID con el color verde, mientras que el color rojo indica si se encuentra inactivo. Los valores de ganancia de default son:  $K_p = 250$ ,  $K_i = 300$  and  $K_d = 50$ .

## 6. Metodología

---



(a) Ganancias controlador PID



(b) LED Encendido / Apagado

Figura 6.8 Área de control de dirección  $\psi$  PID.

En el área de escenarios virtuales predeterminados es posible seleccionar el ambiente en el cual el ROV se estará desplazando [Figura 6.9], la opción de default es Ninguno y el segundo valor representa una plataforma petrolera.

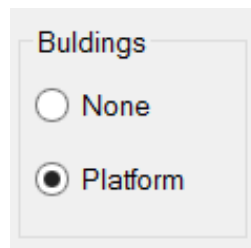


Figura 6.9 Área de escenarios virtuales predeterminados.

En el área de estatus de GUI, como se muestra en la [Figura 6.10], el usuario puede observar la posición actual y la rotación del ROV, así como la  $Z$  deseado ( $Z_d$ ) y  $\psi$  deseado ( $\psi_d$ ). La rotación está representada en grados y la posición en metros.

El voltaje de control de los propulsores es actualizado cada 16 milisegundos, y las unidades que se muestran son volts (v).

La sección de los ejes de posición muestran las coordenadas en el cual la cámara está posicionada para la vista ortogonal. La cámara está siempre  $\sqrt{18} \approx 4.2426$  unidades

## 6. Metodología

---

separadas en el plano de coordenadas XY del centro de masa del ROV, y el ángulo de rotación es usado para calcular las coordenadas “x” y “y”.

The screenshot shows a GUI interface with the following elements:

- Labels: Zd: and  $\psi$ d:
- Section: Position
- Inputs: X (0.00),  $\phi$  (0.00°), Y (0.00),  $\theta$  (0.00°), Z (0.00),  $\psi$  (0.00°)
- Section: Control Voltage
- Inputs: F1 (0.00), F3 (0.00), F2 (0.00), F4 (0.00)
- Section: Eye Position
- Inputs: X (0.00), Z (0.00), Y (0.00), \* (0.00)

Figura 6.10 Área de estados.

Es posible determinar el tipo de Controlador a usar durante la prueba, mediante la selección de un controlador PID o un control deslizante de segundo Orden [Figura 6.11].

The screenshot shows a GUI interface with the following elements:

- Section: Controll
- Radio buttons:  PID,  PD Sliding

Figura 6.11 Tipo de Controlador.

La última sección en la interfaz GUI [Figura 6.12], es el botón de Inicio (Start), Parar (Stop) y Graficas (Charts). En esta sección el entrenamiento puede ser iniciado, detenido o visualizar en gráficas los resultados del comportamiento del ROV durante la simulación.

## 6. Metodología

---

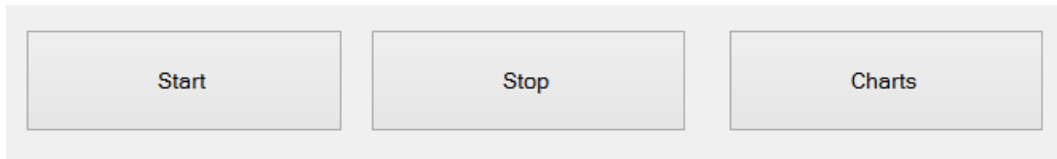


Figura 6.12 Área de Start / Stop / Charts.

Existen dos modos de entrenamiento para el simulador [Figura 6.13]. El modo Normal (Normal) y el modo de Prueba (Test). El modo Normal, mediante la selección de opción de la caja de texto, después de iniciada la simulación esta se detendrá hasta que el usuario presione el botón de detener (Stop). Si el joystick no se encuentra conectado a la computadora durante la simulación en el modo Normal, entonces el valor del voltaje para el primer, segundo y tercer propulsor es de 2.5v. Con la selección del modo de Prueba es necesario seleccionar el tiempo (en segundos) deseado; la profundidad deseada siempre será de 5m, la orientación deseada es  $0^\circ$  y el voltaje en el propulsor uno y dos es de 2.5v.

Al final de cada ejecución los vectores de estados  $(\eta, v, w)$  y el vector de la salida  $\tau$  (F4) [Figura 6.3] son almacenados en archivos de texto así como el tiempo y los valores deseados de profundidad y orientación  $(Z_d$  y  $\psi_d$ ).

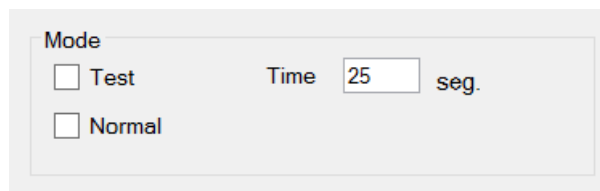


Figura 6.13 Área de selección Mode.

### 6.9.2. Simulación

Durante la ejecución del simulador del ROV, el usuario experimentará la sensación de estar manipulando al ROV KAXAN con la opción de elegir uno de los dos controles de profundidad automática implementados en el modelo informático:

## 6. Metodología

---

- Ejecución en modo Prueba (Test): control deslizante de segundo orden
- Ejecución en modo Prueba (Test): PID

Al terminar la simulación el resultado de la ejecución puede analizarse mediante gráficas de variables de interés.

Los parámetros establecidos durante la simulación son mostrados en la [Figura 6.14].

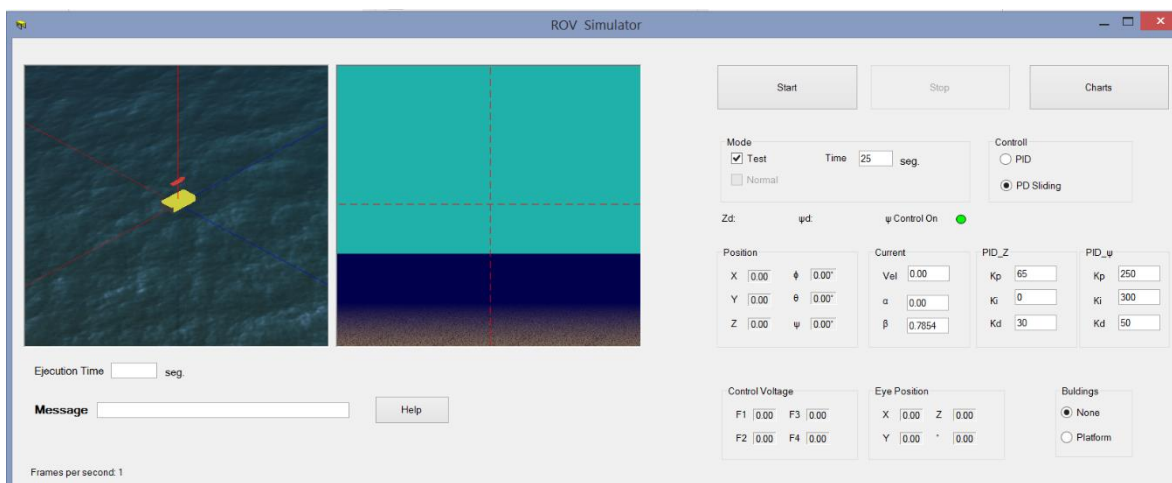


Figura 6.14 Interfaz Gráfica Modo Prueba.

### Ejecución en modo Prueba (Test): control deslizante de segundo orden

Con la elección del control deslizante de segundo orden para el control de la profundidad [Figura 6.15], el comportamiento del ROV se puede interpretar con las gráficas de las variables de interés.

## 6. Metodología

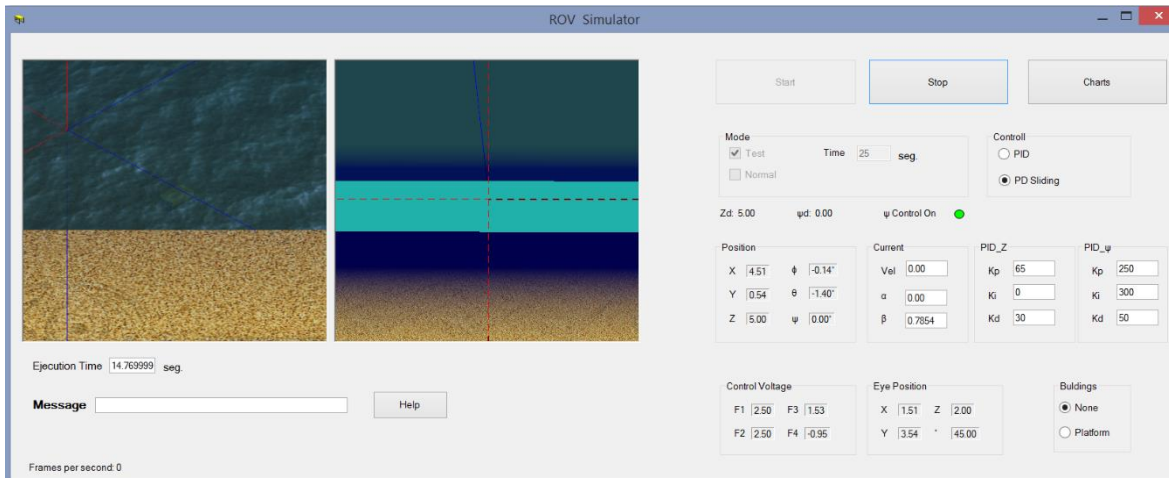
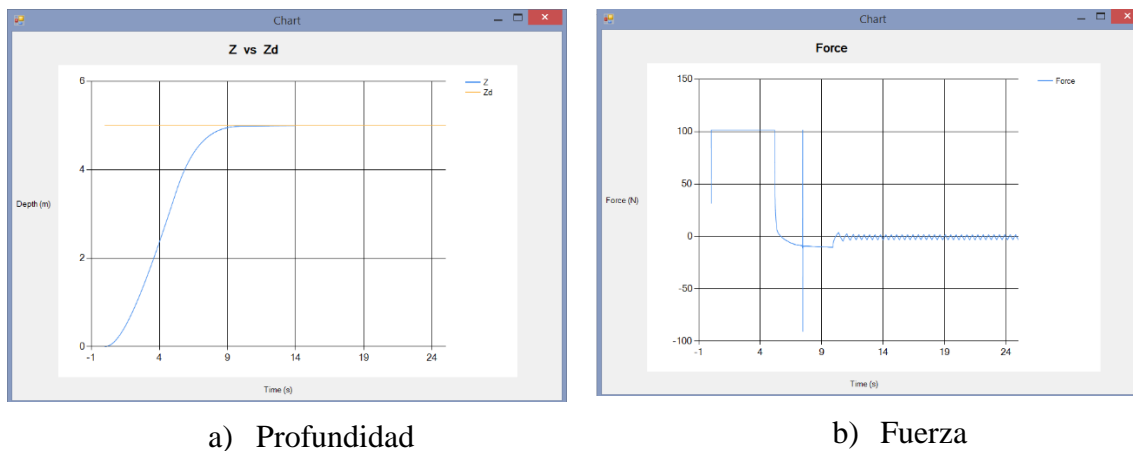


Figura 6.15 Parámetros de simulación modo Prueba: PD deslizante.

Al finalizar el tiempo establecido en la casilla ‘Time’ de los parámetros, se ejecutan las gráficas de resultados, donde en la [Figura 6.16a] se observa que el comportamiento del ROV fue de al menos en 9seg en llegar a la profundidad deseada 5m. En la [Figura 6.16b] el propulsor F4 toma alrededor de 10 segundos en alcanzar una estabilidad.



a) Profundidad

b) Fuerza

Figura 6.16 Gráficas de resultados modo Prueba: PD deslizante.

### Ejecución en modo Prueba (Test): PID

Durante la ejecución en este modo y seleccionando el modo Control PID [Figura 6.17] con los mismos parámetros de ejecución, el comportamiento del ROV es casi semejante.



## 6. Metodología

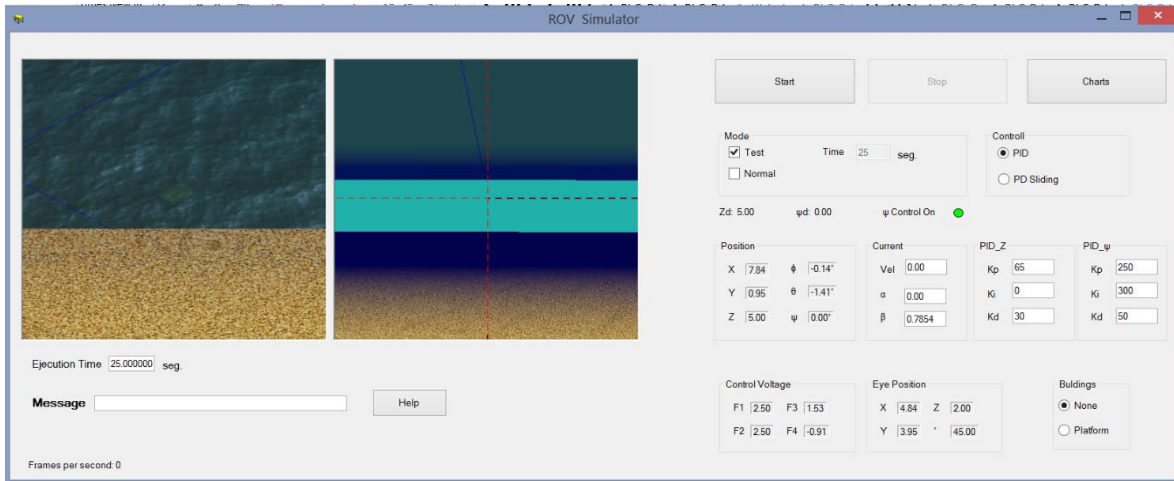
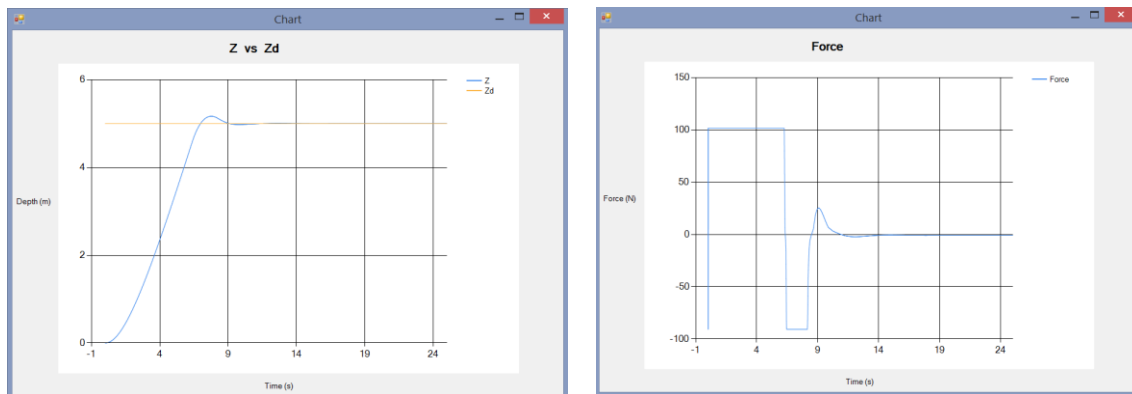


Figura 6.17 Parámetros de simulación modo Prueba: PID.

En las gráficas de comportamiento se puede observar que para la profundidad [Figura 6.18a] el ROV está alcanzando la profundidad deseada en 9seg, a diferencia del PD deslizante, el PID presenta una llegada no tan suave. Para el caso del propulsor F4 [Figura 6.18b] la gráfica muestra que a partir del segundo 11 se obtiene la estabilidad del mismo.



a) Profundidad

b) Fuerza

Figura 6.18 Gráficas de resultados modo Prueba: PID.

## 7. Resultados

---

### 7. Resultados

El objetivo del proyecto de tesis fue cumplido, se logró complementar al desarrollo de la solución informática; la opción de seleccionar entre dos controles para el ROV: PID y Sliding segundo orden, así como la visualización de gráficas con variables que son de interés en el análisis del comportamiento de los controladores del ROV.

Con el análisis de gráficas de comportamiento ( $\tau$  y  $Z$  vs  $Z_d$ ), los resultados de las simulaciones, ya sea en un modo Normal o bien en el modo Prueba, permite determinar si las variables de entorno fueron las ideales.

Durante la ejecución de la simulación en modo Prueba y comparando los dos controles, se puede determinar que en ambos el comportamiento es similar, logrando llegar casi en un mismo tiempo a la profundidad deseada (5m) y la estabilidad del propulsor F4 también es en un tiempo similar.

La implementación de un controlador Deslizante de segundo orden para el control de profundidad en un ambiente virtual es la principal contribución de este trabajo de tesis.

## 8. Conclusiones

---

### 8. Conclusiones

El desarrollo de aplicaciones informáticas para ambientes virtuales se puede realizar en diferentes lenguajes de programación, cada uno con sus limitantes. El uso del lenguaje de programación orientado a objetos, C# de VisualStudio, permite la facilidad de implementar diferentes librerías que aportan sensación visual, así como la solución en línea de las ecuaciones de los modelos matemáticos dando la impresión que se trabaja en lenguajes de programación especializados para realidad virtual. La incursión de librerías como OpenGL para ambientes virtuales, además del uso de dispositivos adicionales (Joystick) crea un ambiente de simulación donde se permite al operario la sensación “realista” de pilotear un ROV.

El reto existirá siempre con el avance de las tecnologías de información, debido a que cuando se concluye con el desarrollo de una aplicación informática con condiciones técnicas idóneas; su uso al paso del tiempo puede implicar la actualización del software, el hardware o inclusive en casos críticos descontinuar la solución tecnológica. Sin embargo estas nuevas tecnologías permitirán una aproximación cada vez más exacta hacia la aplicación real en el manejo, en este caso, del ROV.

---

## Bibliografía

- [1] B. Fletcher, S. Harris (1996). Development of a virtual environment based training system for ROV pilots. In OCEANS '96. MTS/IEEE. Prospects for the 21st Century. OCEANS.1996.
- [2] Chapra, Steven C. y Canale, Raymond P. (2007). Métodos numéricos para ingenieros. McGraw-Hill Interamericana. México, D.F.
- [3] D. Brutzman (1995). Virtual world visualization for an autonomous underwater vehicle. In OCEANS '95. MTS/IEEE.
- [4] Hernández Alvarado, Rodrigo (2016). Control auto-sintonizado utilizando redes neuronales. Tesis Doctorado del PICYT, CIDESI.
- [5] Ogata, K. (1995). Discrete Time Control Systems. 2a. Edition. Prentice Hall.
- [6] Rivera Resendiz, Adrian. (2013). Towards to the development of a virtual environment for operators training for the ROV KAXAN from CIDESI. Master's thesis, Maestría Germano – Mexicana, CIDESI.
- [7] Soto Cajiga, Jorge Alberto (2006). Generación automática de trayectorias para un robot manipulador utilizando procesamiento de imágenes. Tesis Maestría del PICYT, CIDESI.
- [8] Thor I., Fossen (1994). Guidance and control of ocean vehicles. John Wiley & Sons, Ltd.
- [9] Tomás. Salgado-Jiménez, Luis G. García-Valdovinos and Guillermo Delgado-Ramírez (2011). Control of ROVs using a Model-free 2nd-Order Sliding Mode Approach, Sliding Mode Control. Prof. Andrzej Bartoszewicz (Ed.), InTech, DOI: 10.5772/15951. Book chapter available from:  
<https://www.intechopen.com/books/sliding-mode-control/control-of-rovs-using-a-model-free-2nd-order-sliding-mode-approach>

---

## Anexo A. Hoja de datos Propulsores

### A.1 Modelo 520

#### MODEL 520 DC BRUSHLESS THRUSTERS



*Proven in the world's harshest subsea environments, Tecnadyne's thrusters have been at the forefront of propulsion technology for over 20 years. With over 1,400 units delivered to customers worldwide, the Model 520 is Tecnadyne's most successful thruster. Recently superceded by the Model 560, which is 30% more efficient and produces 70% more thrust with no increase in size and weight, the legacy Model 520 will be available for use in existing designs for the foreseeable future.*

*The propeller of the Model 520 is magnetically coupled using a design perfected by Tecnadyne. With this design, a magnet array in the hub of the propeller is driven by a matching magnet array attached to the drive motor. By eliminating the rotating drive shaft and shaft seals that always seem to leak over time, the Model 520 acheives extremely high reliability. Additionally, the magnetic coupling will ratchet if overloaded, preventing damage caused by objects jammed in the propeller. And since the water lubricated propeller bearings are external to the pressure housing, they can be easily replaced in several minutes.*

*Employing a high RPM, low inertia DC brushless motor, coupled to a 6/1 ratio planetary gearset, the Model 520 delivers maximum reliability, high efficiency and high power in an extremely compact , lightweight and easy to maintain package. A Nylon propeller and Kort nozzle combine to give the Model 520 extremely high Bollard thrust and open water efficiency.*

*For depths to 750 meters, the power and control electronics are housed within the hard anodized aluminum motor casing, greatly simplifying the installation and electrical interface. For 1,500 meters, stainless steel portions of the pressure housing are replaced with titanium and for full ocean depth rating, the electronics are installed in a remote, one atmosphere housing (either the customers housing or one supplied by Tecnadyne) and the thruster is oil filled for pressure tolerance.*

*The Model 520 is available for operation at voltages from 24vdc to 330vdc (150vdc standard) supplied by a well filtered battery bank, rectified and filtered AC or a DC power supply. In addition to the main power, the thruster requires isolated 12vdc instrumentation power and a +/-5v analog speed and direction control signal. Alternately, a full servo RS232 or RS485 input controller is available but this must be installed in a remote, one atmosphere housing. Please refer to the Tecnadyne website for detailed installation and interface instructions.*

*The standard depth rating of the Model 520 is 750 meters –1,500 meters and full ocean depth are available options. Customer specified subsea connectors and cables, stainless steel or titanium housings and custom mountings are also available.*

#### MODEL 520 SPECIFICATIONS

**Bollard Output**  
23lbf (10.4kg) forward  
13lbf (5.9kg) reverse  
w/ Nylon propeller

**Input**  
150vdc, 3.1A power  
(525 watts at alternate  
voltages)  
+12v, 200mA isolated  
instrumentation power  
+/-5v analog speed  
command

**Weight**  
4.0lb (1.8kg) in air  
3.0lb (1.4kg) in water

**Depth Rating**  
2,500ft (750m) standard  
5,000ft (1,500m) and  
full ocean depth (oil  
filled) optional

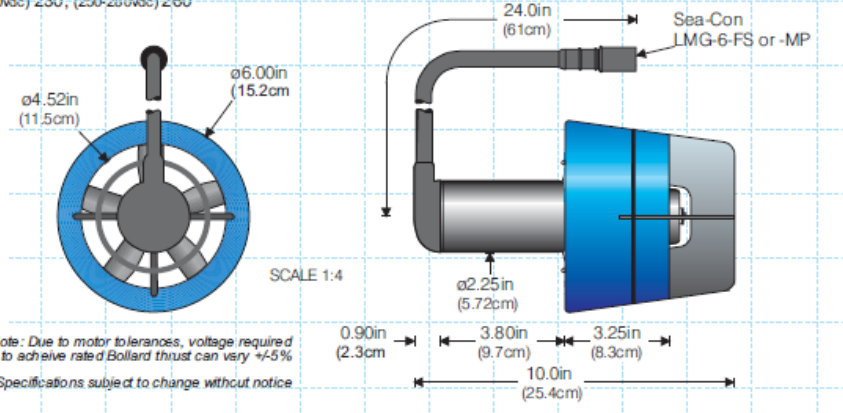
(1,000m & greater depth subject to  
US Govt. export approval)

**TECNADYNE**  
A TECNNOVA INC. COMPANY

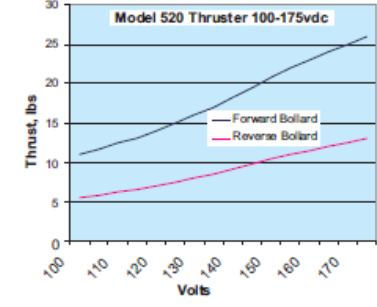
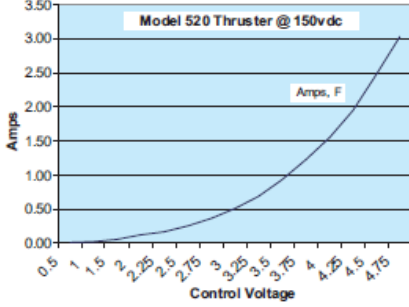
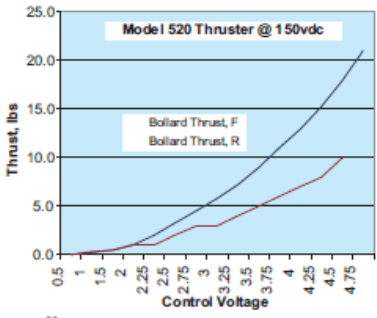
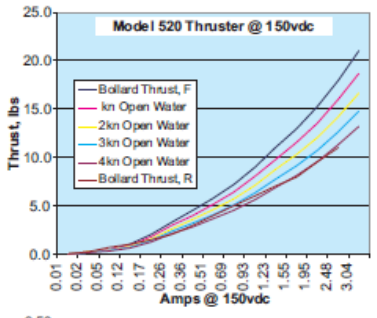
# MODEL 520 DC BRUSHLESS THRUSTERS

Part Number: 520 -

(22-26vdc) 024,	(28-32vdc) 030	Ln (SeaCon LMG-6FS w/ cable length of n meters) Mn (SeaCon LMG-6MP w/ cable length of n meters) X (Customer specified connector)	0750 (750m depth)
(40-44vdc) 044,	(46-50vdc) 048		1500 (1500m depth)
(56-60vdc) 060,	(70-75vdc) 075		OFRE (oil filled remote electronics)
(95-105vdc) 100,	(125-135vdc) 130		
(145-155vdc) 150,	(190-210vdc) 200		
(220-240vdc) 230,	(250-280vdc) 260		



Note: Due to motor tolerances, voltage required to achieve rated Bollard thrust can vary +/-5%  
 Specifications subject to change without notice



**TECNADYNE** A TECNOVA INC. COMPANY  
 14627 Calle Diegueno, PO Box 676086, Rancho Santa Fe, CA 92067, USA  
 Voice: 1.858.756-9660x101 Fax: 1.858.756-9880  
 E-mail: tecsales@tecnadyne.com URL: http://www.tecnadyne.com

## A.2 Modelo 540

### MODEL 540 DC BRUSHLESS THRUSTERS

The Model 540 is the tunnel thruster version of the Model 520 & 560 ducted thrusters, of which Tecnadyne has delivered over 1,400 units to customers worldwide. The Model 540 fits within a tunnel duct of 15.2cm inside diameter. Since it produces the same Bollard thrust in both forward and reverse directions, the Model 540 is ideally suited as a lateral or vertical thruster on mid-size AUV's and on hydrodynamic ROV's.



The precision stainless steel propeller of the Model 540 is magnetically coupled using a design perfected by Tecnadyne. With this design, a magnet array in the hub of the propeller is driven by a matching magnet array attached to the drive motor. By eliminating the rotating drive shaft and shaft seals that always seem to leak over time, the Model 540 achieves extremely high reliability. Additionally, the magnetic coupling will ratchet if overloaded, preventing damage caused by objects jammed in the propeller. And since the water lubricated propeller bearings are external to the pressure housing, they can be easily replaced in several minutes.

Employing a high RPM, low inertia DC brushless motor, coupled to a 6/1 ratio planetary gearset, the Model 540 delivers maximum reliability, high efficiency and high power in an extremely compact, lightweight and easy to maintain package.

For depths to 1,000 meters and optionally to 2,500 meters, the power and control electronics are housed within the hard anodized aluminum motor casing, greatly simplifying the installation and electrical interface. For full ocean depth rating, the electronics are installed in a remote, one atmosphere housing (either the customers housing or one supplied by Tecnadyne) and the thruster is oil filled for pressure tolerance.

The Model 540 is available for operation at voltages from 48vdc to 330vdc (150vdc standard) supplied by a well filtered battery bank, rectified and filtered AC or a DC power supply. In addition to the main power, the thruster requires an isolated +/-5v analog speed and direction control signal and 12vdc instrumentation power. Alternately, a full servo RS232 or RS485 input controller is available which, due to its size, must be installed in a remote, one atmosphere housing. Please refer to the Tecnadyne website for detailed installation and interface instructions.

The standard depth rating of the Model 540 is 850 meters. 2,500 meters and full ocean depth with remote one atmosphere electronics is an available option. Remote electronics options include the extremely compact Tecnadyne controller module or larger, full servo brushless or sensorless units. For applications requiring extremely low noise, Tecnadyne offers an optional remote linear drive. Customer specified subsea connectors and cables, stainless steel or titanium housings and custom mountings are also available.

### MODEL 540 SPECIFICATIONS

**Bollard Output**

21lbf (9.5kg) forward  
21lbf (9.5kg) reverse

**Input**

150vdc, 3A power  
(450 watts at alternate voltages)  
12vdc, 200mA isolated instrumentation power  
+/-5v analog speed command

**Weight**

4.2lb (1.9kg) in air  
3.0lb (1.4kg) in water

**Depth Rating**

2,800ft (850m) standard,  
8,000ft (2,500m) and  
full ocean depth (oil filled) optional

(1,000m & greater depth subject to US Govt. export approval)

 **TECNADYNE**  
A TECNNOVA INC. COMPANY

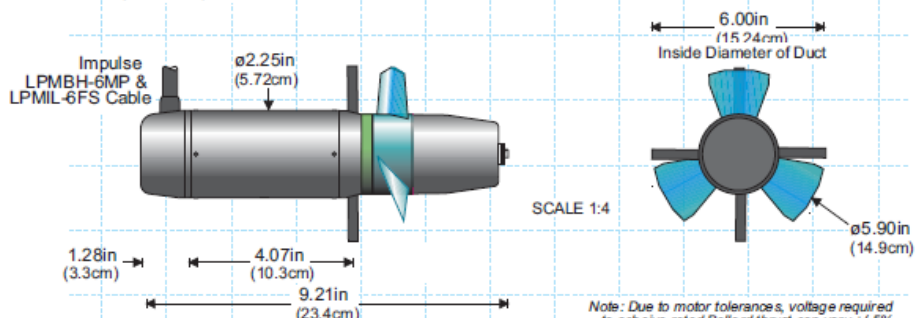
# MODEL 540 DC BRUSHLESS THRUSTERS

Part Number: 540 -

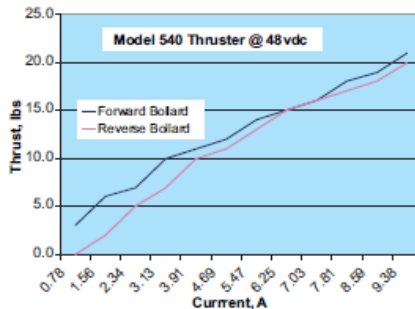
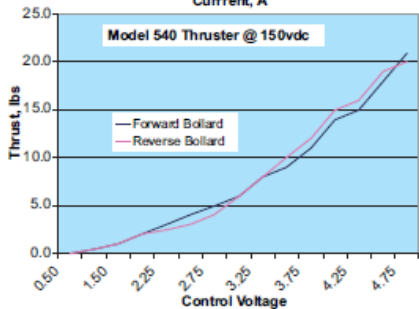
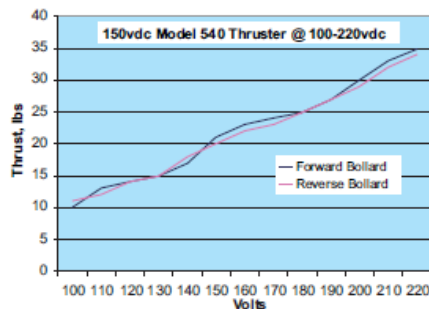
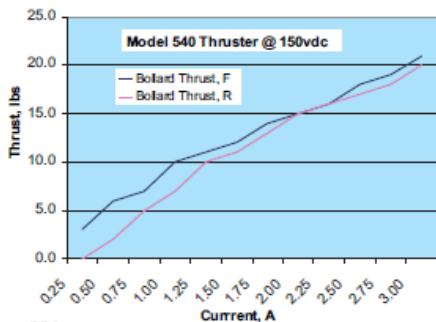
- (44-52vdc) 048
- (65-75vdc) 070
- (110-130vdc) 120
- (140-160vdc) 150
- (190-210vdc) 200
- (250-280vdc) 260
- (300-330vdc) 300

- Bn (LPMBH-6MP / LPMIL-6FS w/ cable of n meters)
- Ln (SeaCon LMG-6FS w/ cable length of n meters)
- Mn (SeaCon LMG-6MP w/ cable length of n meters)
- X (Customer specified connector)

- 0850 (850m depth)
- 2500 (2500m depth)
- OFRE (oil filled remote electronics)



Note: Due to motor tolerances, voltage required to achieve rated Bollard thrust can vary +/-5%  
Specifications subject to change without notice



0306