

## **Sistema de control de navegación autónoma subacua que emplea redes neuronales**

*Autonomous underwater navigation control system using neural networks*

CLAUDIO CÉSAR LOPEZ

Escuela de Oficiales de la Armada, Facultad de la Armada,  
Universidad de la Defensa Nacional, Argentina  
cesarl.claudio@gmail.com

*La elaboración de este artículo tuvo como incentivo buscar una manera de mostrar una solución al modelo de autonomía de movimiento y navegación de un dispositivo a través del empleo del aprendizaje automático por medio de una red neuronal. En este caso, se trata de un dispositivo subacuático. La ventaja del empleo de esta técnica radica en el cambio de paradigma de programación, ya que se deja de pensar de manera lógica y matemática. No se hacen afirmaciones para confirmar si las propiedades del programa cumplen o no con los requisitos. En las redes neuronales, se parte de observaciones sobre un entorno incierto, se realizan experimentos y se usan estadísticas. Esto abre un abanico de posibilidades en áreas aun no exploradas. Por ejemplo, el hecho de reconocer un rostro o el habla de una persona para otro ser humano es relativamente sencillo. Sin embargo, si*

*quisiéramos escribir un código para resolver esto, sería muy complicado. El aprendizaje automático zanja este problema sin decirle a la aplicación qué hacer, solo con “mostrarle” ejemplos puede obtener resultados satisfactorios. Esto puede extender las posibilidades de crear sistemas aún más complejos. La aplicación de las redes neuronales en la navegación autónoma ha sido empleada con asiduidad por la empresa Tesla en sus modelos de automóviles autónomos que ya han salido al mercado “para aprender de su experiencia” en las rutas y caminos. El alcance de este artículo es a los efectos de mostrar el algoritmo de una red neuronal sencilla ciñéndose a un tipo de vehículo subacuático disponible. Este modelo se puede ampliar en complejidad para abarcar otros sistemas, como el del móvil de referencia.*

## **1. Introducción**

Uno de los procesos más importantes en la operatoria de cualquier vehículo autónomo subacuático es el de la navegación. Este abarca la percepción de señales de diversos dispositivos que, en base al plan establecido (derrota de navegación), transmiten las órdenes a los sistemas de gobierno y propulsión que forman en conjunto el control del vehículo. La manera en que navegue el dispositivo hará que este pueda alcanzar a cumplimentar el/los objetivo/s buscado/s. Todo esto, por supuesto, supervisado por el/los operador/es humano/s ubicado/s en una/s estación/es de trabajo que analizarán y comandarán las acciones de la embarcación. Asimismo, existirá un conjunto de submódulos relacionados que deberán manejar datos de una variedad de tipos de sensores para lograr que el movimiento del dispositivo sea el adecuado, incluso bajo condiciones hidrometeorológicas adversas.

El algoritmo que permitirá cumplimentar de manera adecuada las órdenes mencionadas podría emplear una red neuronal que obtenga la adecuada señal a cada motor

implicado en la maniobra de control de la navegación.

En este artículo, trabajaremos como modelo con un sistema de navegación basado en una plataforma de navegación inercial constituido por sensores de movimiento (acelerómetros), sensores de rotación (giróscopos) y un pequeño computador, para obtener la posición y la velocidad integrando los datos de aceleración obtenidos en cada uno de los tres ejes de movimiento x, y, z. Este sistema procesara esta información para que, luego, el sistema de control basado en controladores PID, genere las entradas a la red neuronal montada sobre una placa Arduino. También describiremos el modelo de vectores de atributos de entrada para la red neuronal que posibiliten la correcta salida de la red.

## 2. Marco teórico

### *Redes neuronales*

Estas representaciones se inspiran en las neuronas biológicas y su esquema de conexiones en el cerebro, aunque en realidad no simulan al sistema nervioso.

También se los conoce como sistemas conexionistas o sistemas de procesamiento paralelo distribuido (*parallel distributed processing*). En ellos, se desenfatisa el uso explícito de símbolos y se afirma que la inteligencia aparece en sistemas compuestos de elementos simples (neuronas artificiales) que interactúan.

El paralelismo se corporiza en el procesamiento simultáneo de todas las neuronas y está distribuido en toda la red.

Las neuronas artificiales, también llamadas unidades, tienen entradas y una salida. La salida de una neurona artificial puede conectarse a las entradas de otra y esta salida es una función no lineal de sus entradas y estas representan sus parámetros. El aprendizaje ocurre al ajustar los parámetros para hacerlos corresponder a los datos. Las redes neuronales

multicapa pueden representar una aproximación a cualquier función lógica.

El aprendizaje de una red neuronal puede ser descripto como un problema de minimización: dado un conjunto de ejemplos, encontrar un conjunto de valores de los parámetros de entrada que minimice el error cuadrático. El aprendizaje por *backpropagation* es una búsqueda siguiendo la disminución del gradiente a través del espacio de parámetros para minimizar el error cuadrático.

### *Ecuaciones de control de un ROV*

El desafío para diseñar un sistema de control para un ROV (*remote operated vehicle*) es obtener un modelo matemático apropiado. Con el creciente desarrollo de las computadoras, las aplicaciones de dinámica de fluidos computacional (CFD) en la industria marítima se han vuelto importantes. Los enfoques numéricos han evolucionado a un nivel de precisión, lo que permite que se apliquen a vehículos marinos como el AUV (*autonomous underwater vehicle*). Sin embargo, la aplicación del ROV de forma compleja aún no está disponible en la literatura de investigación.

El modelado de ecuaciones dinámicas del ROV suele ser el primer paso para desarrollar una simulación.

En general, existen básicamente dos métodos para derivar el modelo dinámico: las ecuaciones de Lagrange, que usan el método de variación, y las ecuaciones de Newton, que usan la Segunda Ley de Movimiento de Newton. La transformación de Euler (comúnmente conocida como la ecuación de la cinemática) proporciona la relación entre las dinámicas derivadas en las coordenadas fijadas en la tierra y las fijadas en el cuerpo. Existen ventajas prácticas para derivar la dinámica en la coordenada fija del cuerpo. Primero, la mayoría de los sensores y actuadores montados en el ROV miden los parámetros del cuerpo, como las velocidades del

ROV y proporcionan fuerzas propulsoras. En segundo lugar, la ecuación dinámica está parametrizada de forma inherente y, por lo tanto, es menos complicada, ya que esta forma parametrizada no depende de los ángulos de Euler.

Antes del modelado del ROV, las siguientes suposiciones pueden ser hechas al derivar la ecuación dinámica general de ROV para simplificar el esfuerzo en el modelado. Son: a) el ROV es un cuerpo rígido y está completamente sumergido una vez en el agua; b) se supone que el agua es un fluido ideal que es incompresible, no es viscosa (sin fricción) y es irracional; c) el ROV se mueve lentamente ; d) el marco de referencia fijo en tierra es inercial; e) la perturbación debida a las ondas se descuida ya que está completamente sumergida; f) no se modelan las dinámicas de enlace unidas al ROV; g) existen movimientos en la dirección x (aumento), movimientos en la dirección y (balanceo), movimientos en la dirección z (desplazamiento), rotaciones sobre el eje x (desplazamiento), rotaciones sobre el eje y (paso), rotaciones sobre el eje z (guiñada).

Una ecuación dinámica de ROV de cuerpo rígido se expresa en el marco fijo del cuerpo, ya que las fuerzas de control y los dispositivos de medición están relacionados intuitivamente con este marco de referencia del cuerpo. Usando el enfoque newtoniano, el movimiento de un cuerpo rígido con respecto al marco de referencia fijo al cuerpo en el origen viene dado por la ecuación:

$$M_{RB} \dot{v} + C_{RB}(v) = \tau$$

Donde:

$M_{RB}$ : es la matriz de inercia

$C_{RB}$ : es la matriz de Coriolis y centrípeta

$\tau$  = es el vector de fuerzas externas y movimientos

Para controlar el ROV, se pueden emplear distintas técnicas de control que se han propuesto para otros sistemas. Este es el caso de los controladores en lazo cerrado que realimentan las señales medidas por los sensores de un dispositivo. Es el caso de un controlador PID. El algoritmo del control PID consta de tres parámetros distintos: el proporcional, el integral, y el derivativo. El valor proporcional depende del error actual, el integral depende de los errores pasados y el derivativo es una predicción de los errores futuros. La suma de estas tres acciones es usada para ajustar el proceso por medio de un elemento de control, como la posición de una válvula de control o la potencia suministrada a un calentador.

Históricamente, se ha considerado que, cuando no se tiene conocimiento del proceso, el controlador PID es el controlador más adecuado. Ajustando estas tres variables en el algoritmo de control del PID, el controlador puede proveer una acción de control adaptada a los requerimientos del proceso en específico. La respuesta del controlador puede describirse en términos de respuesta del control ante un error, el grado al cual el controlador sobrepasa el punto de ajuste, y el grado de oscilación del sistema. Nótese que el uso del PID para control no garantiza un control óptimo del sistema o su estabilidad.

Un controlador PID puede ser llamado también PI, PD, P o I en la ausencia de las acciones de control respectivas.

La ley de control se expresa de la siguiente manera:

$$\tau_{PID} = J(v)^T \left[ K_p \tilde{\eta} + K_d \dot{\tilde{\eta}} + K_i \int \tilde{\eta}(o) do \right]$$

donde  $\tilde{\eta} = \eta_d - \eta$  es el error entre la posición real y la deseada,

$K_p$  es la matriz de ganancias proporcionales,

$K_d$  ganancias derivativas, y

$K_i$  es la matriz de ganancias integrales.

Por otro lado, se ha propuesto también el controlador PID con realimentación de aceleración.

$$\tau_{PIDA} = J(v)^T [K_p \tilde{\eta} + K_d \dot{\eta} + K_i \int \tilde{\eta}(o) do] - H \dot{v}$$

Donde la matriz  $H$  puede ser constante o dependiente de la frecuencia. El propósito de la realimentación de aceleración es que el sistema sea menos sensible a perturbaciones externas como lo pueden ser las corrientes marinas.

### *Detección de obstáculos*

Los robots submarinos están equipados con sistemas de sensores dedicados a determinar la localización del vehículo, su velocidad y aceleración, así como su estado de funcionamiento. Los sistemas de visión entran en esta categoría de dispositivos cuando son utilizados para determinar la posición y orientación de objetos con respecto al vehículo. Los sensores permiten controlar los movimientos del robot, lo cual permite a su vez que este pueda realizar la misión que le fue encomendada.

Una clase de dispositivo muy importante, empleado para caracterizar el fondo marino, es el sonar. El sonar está basado en varios proyectores y arreglos de hidrófonos. En función de la diferencia de tiempo entre la emisión de la señal acústica y el eco recibido, se determina la distancia a la que se encuentra un objeto al frente o en ciertos ángulos. Para la detección de obstáculos también es posible utilizar emisores laser y sus receptores, pero estos pueden presentar problemas en aguas

muy turbias.

### 3. El problema

**Niveles de autonomía de un ROV.** El operador de un ROV con funciones automatizadas puede utilizar diferentes niveles de control del vehículo. En general, el nivel de autonomía se establece por la complejidad de la misión, la dificultad ambiental y la independencia del factor humano.

En el modo manual más básico, el operador tiene control directo de cada propulsor. Es un desafío simplemente mantener el ROV estacionario en presencia de corrientes oceánicas en este modo. Controlar cuatro o más propulsores simultáneamente mientras se lucha contra fuerzas externas no es una tarea fácil para un operador. El uso de un joystick con la asignación de propulsores hace que el piloto controle cada grado de libertad de movimiento en lugar de los propulsores, lo que mejorará considerablemente el rendimiento. Sin embargo, el operador aún debe compensar las fuerzas externas y la dinámica del vehículo, y el mantenimiento de la estación requiere un enfoque constante y un operador experto.

El seguimiento de puntos de paso o “piernas” con seguimiento del fondo es un ejemplo de un modo que se incluye en el sistema de control de movimiento. El operador especifica una lista de puntos de paso, velocidad de cruce y distancia sobre el fondo del mar. Luego, el ROV se mueve automáticamente de un punto a otro mientras mantiene la altitud deseada a pesar de las corrientes oceánicas y la geometría desconocida del terreno. El operador no interactúa en el control del vehículo, pero monitorea los datos de las cámaras y los sensores. Si el operador ve algo interesante o si el ROV se mete en problemas, debe interactuar y dar nuevas instrucciones al vehículo. Sin embargo, en este modo el operador es más un supervisor que un piloto. En este nivel, las habilidades del operador no se evalúan en función de su capacidad para maniobrar el vehículo, sino de qué datos e

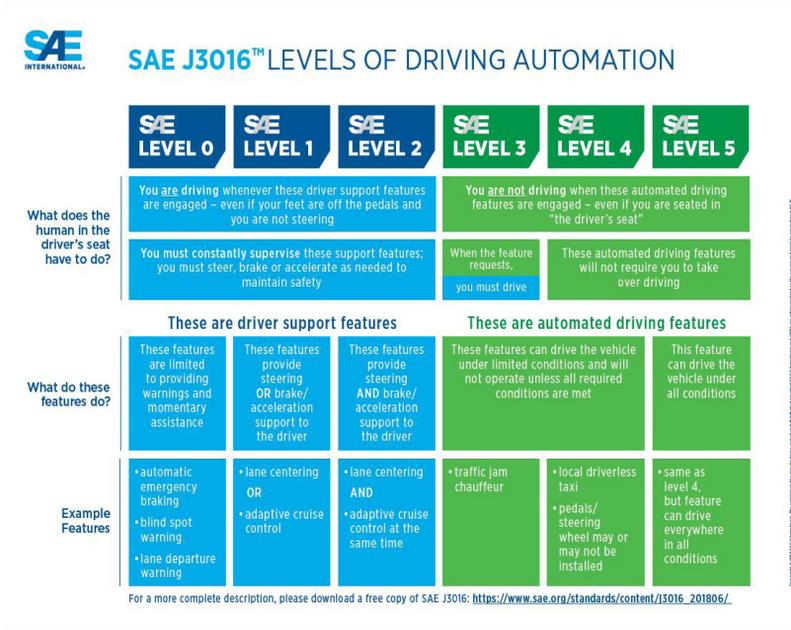
información obtiene del uso del ROV. Así, este trabajo puede considerarse como un paso hacia la autonomía.

El otro extremo, en cuanto al nivel de interacción humana, es la completa autonomía. En este modo de control, el operador asigna una tarea y el vehículo la realiza de forma autónoma sin más instrucciones del operador. Por ejemplo, el operador muestra una imagen de un elemento sumergido cualquiera y le pide al ROV que lo encuentre y lo traiga. El ROV comienza la misión y realiza la búsqueda y planificación de rutas por sí solo, encuentra el objeto, lo recoge y lo entrega sin ninguna interacción del operador.

La bibliografía respecto de los niveles de autonomía para estos dispositivos muestra muchas clasificaciones. Nosotros nos hemos basado en aquella elaborada por SAE International (Society of Automotive Engineers - <http://sae.org>) que se muestra como Figura 1.

**FIGURA 1**

*Niveles de conducción autónoma de SAE internacional*



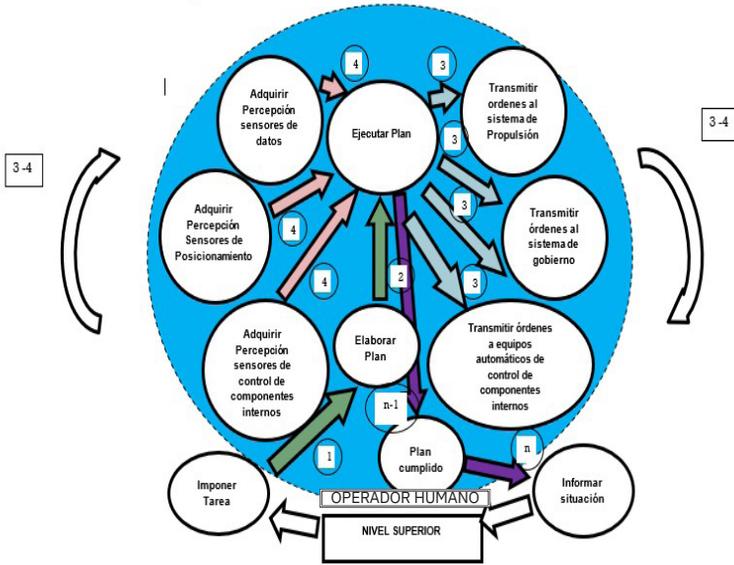
## **Software del sistema de navegación autónoma.**

**Características.** La estructura del software de un sistema de navegación marítima autónoma se compone, desde nuestro punto de vista, de los siguientes procesos:

- Percepción de los sensores de posicionamiento.
- Percepción de los sensores de reconocimiento e identificación electrónica.
- Percepción de los sensores de control interno.
- Interpretación de la tarea impuesta y elaboración del Plan de Acción.
- Situación del sistema respecto a la derrota planeada.
- Situación del sistema respecto al correcto funcionamiento de sus componentes internos.
- Situación del sistema respecto a la misión.
- Ejecución del Plan de Acción:
  - o Trasmisión de las órdenes correspondientes al sistema de gobierno.
  - o Trasmisión de las órdenes correspondientes al sistema de propulsión.
  - o Trasmisión de las órdenes correspondientes a los equipos automáticos de control interno de componentes.
  - o Trasmisión de información al operador humano respecto de la situación del sistema

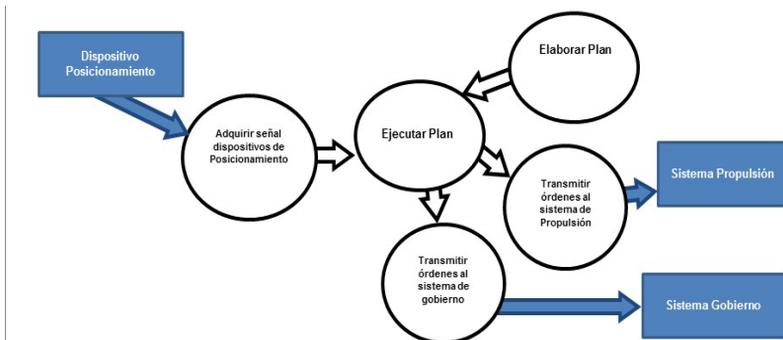
Gráficamente, lo expresado sería:

**FIGURA 2**  
*Estructura del software de un sistema de navegación marítima autónoma.*



Puntualizando aquellos procesos que tienen que ver específicamente con el desplazamiento del vehículo, podremos extraer de este esquema otro como se muestra en la Figura 3.

**FIGURA 3**  
*Esquema de procesos para la navegación.*

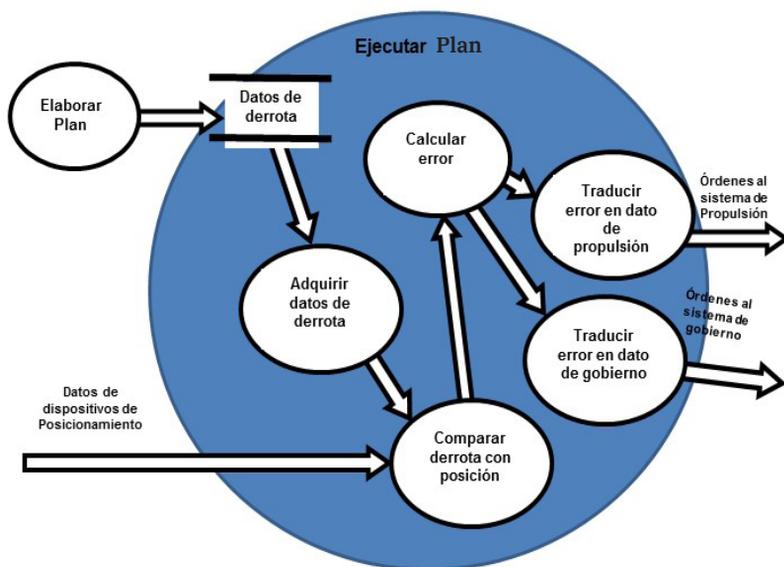


En la Figura 3, las figuras en de forma ovoide representan los procesos del vehículo. El proceso **Ejecutar Plan** contiene el algoritmo de decisión que comparara la actual posición del vehículo con la prevista en el plan de navegación, elaborado por el proceso **Elaborar Plan**, que envía las señales que harán accionar el hardware correspondiente a los procesos relacionados con los sistemas de gobierno y propulsión.

Viendo desde adentro el proceso mencionado, contendría los siguientes subprocesos:

### FIGURA 4

*Esquema de subprocesos del proceso EJECUTAR*



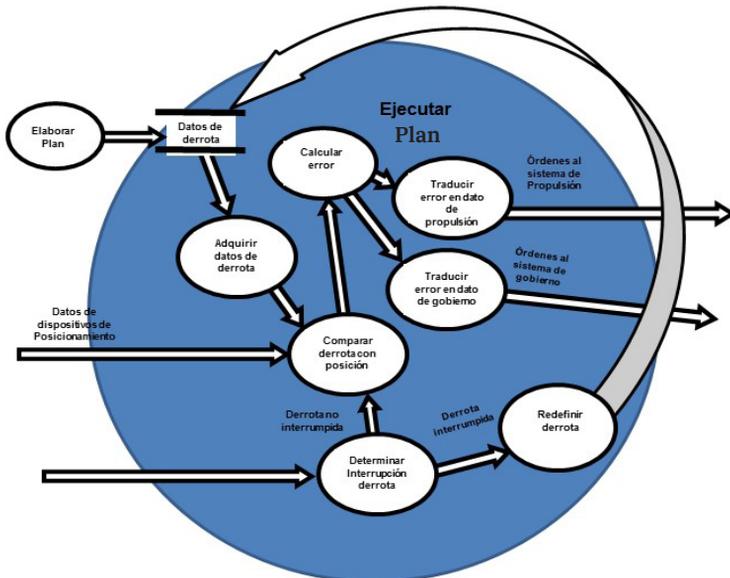
En la Figura 4, se aprecia la cantidad de tareas que se desarrollan en el proceso Ejecutar Plan y que principalmente están orientadas a la consecución del plan pre almacenado que elaboró el mismo sistema. Esto se realiza de manera permanente con una frecuencia configurada mientras el vehículo está en el agua.

La navegación se complica cuando, además de los datos de posicionamiento que envían los dispositivos, se reciben datos de sensores que detectan obstáculos o amenazas no previstos en el plan. Para ello, el proceso tiene previsto un algoritmo de decisión que posibilita continuar cumpliendo con la misión. Este algoritmo está materializado en un subproceso denominado **Redefinir derrota** y se aplica al detectarse ese impedimento en la prosecución de lo planeado. En el gráfico de la Figura 5, se puede observar lo que mencionamos.

La realización de todas estas tareas implica configurar una secuencia de trabajo para que las estas no se interfieran. Una secuencia posible sería la siguiente:

1. Adquirir datos de posicionamiento.
2. Adquirir datos de derrota.

**FIGURA 5**  
*Esquema de subprocesos del proceso EJECUTAR PLAN con la inclusión del subproceso Redefinir derrota.*



3. Adquirir datos de sensores.
4. Determinar si hay interrupción de derrota.
5. Si no hay interrupción de derrota, entonces continuar con:
  - 5.1. Comparar los elementos de 1. y 2. y determinar el error de navegación.
  - 5.2. Calcular el error de navegación.
  - 5.3. Convertir esos datos en señales para los sistemas de propulsión y gobierno.
6. Si hay interrupción:
  - 1.1. Ejecutar el procedimiento de evitación/ evasión del obstáculo/amenaza.
  - 1.2. Redefinir derrota.
  - 1.3. Elaborar y almacenar nuevos datos de derrota.
  - 1.4. Continuar con los puntos 5.2. en adelante.

Esta secuencia constituiría un ciclo que terminaría al alcanzar el/los objetivos.

El tipo de navegación a ejecutarse también influirá en la manera de desplazamiento del vehículo. Si bien, en general, este se moverá siguiendo un plan, es probable que deba, por determinadas circunstancias, navegar por sus propios medios sin una derrota que le indique el camino a seguir. Estas circunstancias podrán ser casos de emergencia como mal funcionamiento, evasión, procedimientos ante incidencias desconocidas o es un entorno totalmente desconocido y no se cuenta con cartografía del área.

*INS.* No importa cuánto retrocedamos en el tiempo, siempre ha sido fundamental saber dónde se está posicionado. Cuando era un Neanderthal que iba a cazar, el hombre debía poder encontrar el camino de vuelta al hogar o al campamento; en la

actualidad, un misil de crucero necesita ubicar su localización para alcanzar su objetivo con alta precisión.

A través del tiempo, se han utilizado varios enfoques diferentes para localizar y navegar a un destino. Usando las estrellas, el sol o marcas especiales en el paisaje. Hoy en día, el GPS es el sistema de navegación más extendido, pero en ciertas aplicaciones el GPS no es una solución viable. En ciertas áreas, como por ejemplo en entornos montañosos, la señal GPS es muy imprecisa o no está presente debido a la interferencia multitrayecto. Asimismo, si el vehículo está ubicado bajo el agua o bajo tierra, la señal GPS no está presente.

Un INS es un sistema de navegación que depende completamente de mediciones inerciales para la navegación. Un INS consiste en acelerómetros que miden la aceleración y giroscopios que miden la rotación angular del sistema. Este conjunto de sensores es llamado Unidad de Medición Inercial (IMU, por sus siglas en inglés). Usando las mediciones de la IMU, el INS puede calcular la actitud actual, la velocidad y la posición del sistema a partir de algún punto inicial conocido. Esto significa que el INS no depende de ninguna tercera aplicación, como GPS, y por lo tanto siempre funcionarán independientemente de las influencias externas. Esto lo hace adecuado para la navegación donde la señal GPS no está presente o si las señales son bloqueadas. Sin embargo, cuando se basa solo en mediciones inerciales, la precisión del INS se degrada con el tiempo debido a imprecisiones de la medición. Usado en un corto intervalo de tiempo, un INS ofrece una precisión mucho mayor que el GPS. Los INS de alta precisión también son adecuados para submarinos donde se necesita exactitud para los desplazamientos y las ayudas externas de posicionamiento no están disponibles fácilmente.

Se emplean tres giróscopos que permiten hacer las mediciones de los ángulos de: cabeceo, alabeo y guiñada, para luego transmitirlos a otro equipo por medio de un canal de comunicación. La unidad de navegación inercial es un equipo

pequeño que integra los giróscopos y acelerómetros en la misma electrónica, incluso hay algunas que tienen una brújula electrónica, permitiendo obtener la posición del dispositivo.

Las entradas del sistema de navegación serán entonces datos de sensores que nos indicaran la presencia de obstáculos, la información del INS que permitirá el mantenimiento del rumbo de navegación y profundidad y datos del sistema de posicionamiento

#### 4. Modelo de solución

El sistema de control es el encargado de calcular las fuerzas necesarias que debe generar el sistema de propulsión para que el ROV alcance las referencias deseadas. La disponibilidad de un sistema de control involucra la sintonización de las Leyes de Control que regularan sus movimientos.

Por otro lado, los datos del sonar de detección de obstáculos nos permitirán, además, maniobrar para que el dispositivo los evite y retome a su rumbo original

Integrar estos datos como entrada a una red neuronal nos posibilitará controlar los sistemas de gobierno y propulsión.

El modelo de aprendizaje estará constituido por una red de tres capas con 4 neuronas de entrada, 6 ocultas y 4 salidas una para cada motor con una función de activación tangente hiperbólica:

$$f(x) = \frac{2}{1 + e^{-2x}} - 1$$

La función tangente hiperbólica transforma los valores introducidos a una escala (-1,1), donde los valores altos tienen de manera asintótica a 1 y los valores muy bajos tienden de manera asintótica a -1. Además, las salidas serán entre 0 y 1 (apagar o encender motor). También cambiaremos las entradas para que todas comprendan valores entre -1 y 1. En

este caso, las entradas comprenderán las diferencias obtenidas por el algoritmo de control representado por las salidas de un controlador PID en los ejes  $y$  (desvío) y  $z$  (Movimiento vertical). El fin de esta red será mantener la derrota y los movimientos del ROV de acuerdo al Plan.

**TABLA 1**

*Instancias de vectores de atributos de entrada de enteramiento de la red neuronal*

Error PID y	Error PID z	Posición obstáculo	Distancia obstáculo	Motor 1	Motor 2	Motor 3	Motor 4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	-1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	1	0	0
0	0	1	-1	-1	-1	0	0
0	0	-1	0	0	0	0	0
0	0	-1	1	1	0	0	0
0	0	-1	-1	-1	-1	0	0
0	1	0	0	0	0	1	1
1	0	0	-1	-1	-1	0	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	0	0
1	0	1	-1	-1	-1	0	0
1	0	-1	0	0	1	0	0
1	0	-1	1	1	0	0	0
1	0	-1	-1	-1	-1	0	0
1	1	0	0	0	0	1	1
1	1	0	1	0	0	1	1
1	1	0	-1	-1	-1	0	0
-1	0	0	1	1	0	0	0
-1	0	0	-1	-1	-1	0	0
-1	0	1	0	1	0	0	0
-1	0	1	1	1	0	0	0
-1	0	1	-1	-1	-1	0	0
-1	0	-1	0	1	0	0	0
-1	0	-1	1	1	0	0	0
-1	0	-1	-1	-1	-1	0	0
-1	1	0	0	0	0	1	1
-1	1	0	1	0	0	1	1

También, estableceremos los siguientes parámetros continuos para contribuir a la simplicidad del modelo:

- La velocidad de los motores será constante
- Los valores de referencia de los motores para el giro de sus hélices serán. 1: giro en sentido horario (avance), 0: sin propulsión, -1: giro en sentido antihorario (reversa)
- El tiempo de operación de los mismos también será constante
- Los intervalos de realimentación de la red serán de al menos 20 segundos

-

Los valores de cada entrada representan:

### Error PID y

Valor	Representación
0	El error de salida del PID es $> -3$ y $< 3$ en eje y
1	El error de salida del PID es $< 3$ en eje y
-1	El error de salida del PID es $< -3$ en eje y

### Error PID z

Valor	Representación
0	El error de salida del PID es $> -3$ y $< 3$ en eje z
1	El error de salida del PID es $< 3$ en eje z
-1	El error de salida del PID es $< -3$ en eje z

## Posición obstáculo

Valor	Representación	Observaciones
0	No hay obstáculo	
1	Obstáculo a la derecha	Si Distancia Obstáculo es 0 no hay efecto de este valor
-1	Obstáculo a la izquierda	Si Distancia Obstáculo es 0 no hay efecto de este valor

## Distancia Obstáculo

Valor	Representación	Observaciones
0	No hay obstáculo	
1	Obstáculo a 50 m o +	Si Posición Obstáculo es 0 no hay efecto de este valor
-1	Obstáculo a 50 m o -	Si Posición Obstáculo es 0 no hay efecto de este valor

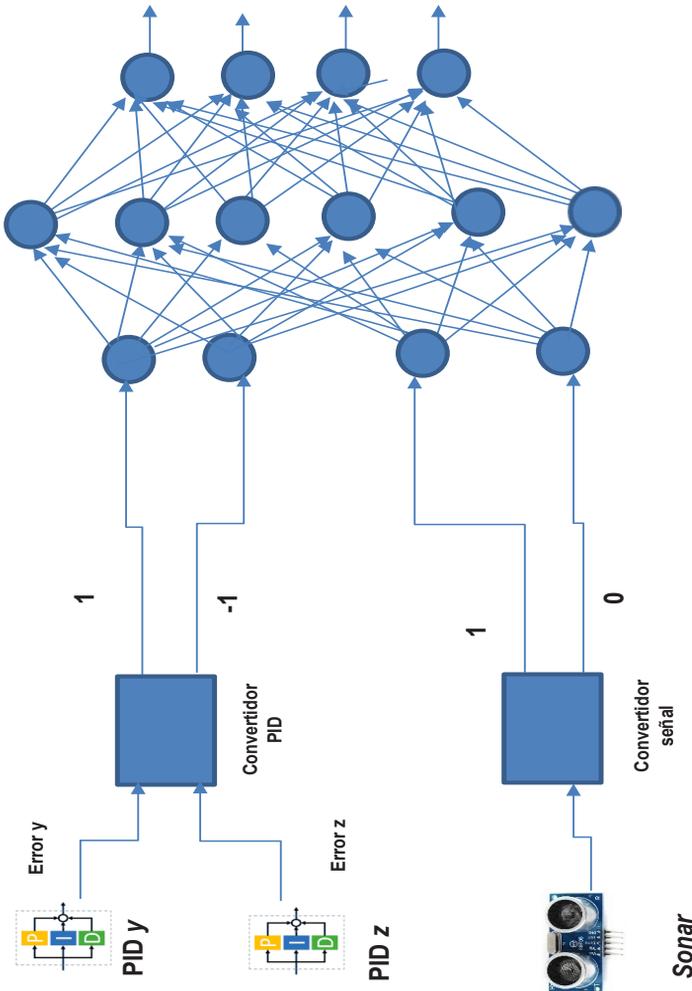
Las acciones de los motores del ROV serán:

**TABLA 2**  
*Acciones de los motores*

Acción	Motor 1	Motor 2	Motor 3	Motor 4
Avanzar	1	1	0	0
Retroceder	-1	-1	0	0
Giro derecha	1	0	0	0
Giro izquierda	0	1	0	0
Inmersión	0	0	1	1
Emersión	0	0	-1	-1

Las señales que representan la salida del controlador PID y del sonar para la detección de obstáculos deberán pasar antes de ingresar la red neuronal por sendos convertidores que le asignen el valor correspondiente de acuerdo a la Tabla 1. Gráficamente, lo podemos ver de esta manera:

**FIGURA 6**  
*Esquema de la red neuronal*

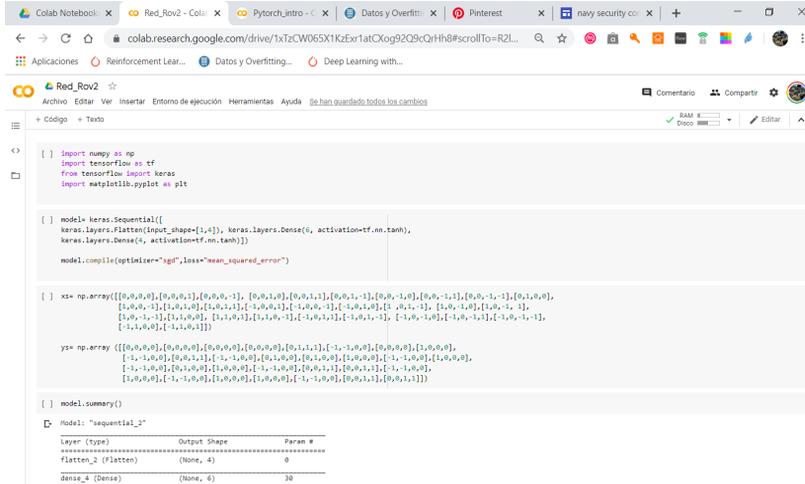


El código de la red en *tensorflow* será el siguiente:

## FIGURAS 7 A 10

*Código de la red neuronal en Google Colab*

### FIGURA 7



```
[ ] import numpy as np
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt

[ ] model=keras.Sequential([
keras.layers.Flatten(input_shape=[1,4]), keras.layers.Dense(6, activation=tf.nn.tanh),
keras.layers.Dense(4, activation=tf.nn.tanh)])

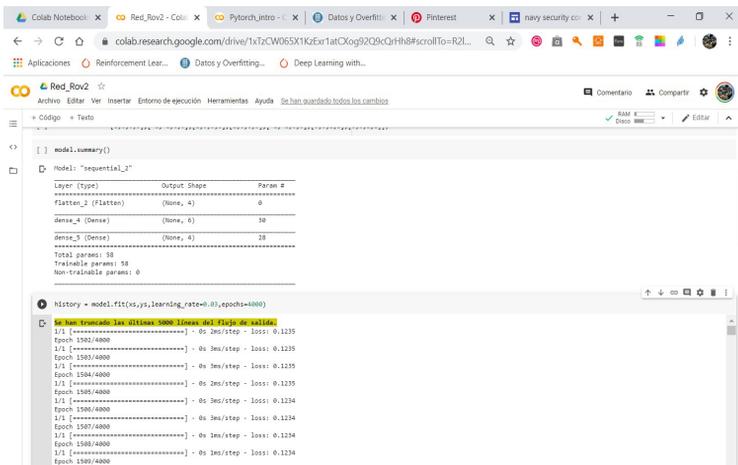
model.compile(optimizer="sgd",loss="mean_squared_error")

[ ] xs= np.array([[0,0,0,0],[0,0,0,1],[0,0,0,-1], [0,0,1,0],[0,0,1,1],[0,0,1,-1],[0,0,-1,0],[0,0,-1,-1],[0,1,0,0],
[1,0,0,-1],[1,0,0,0],[1,0,0,1],[1,0,0,-1],[1,0,0,1],[1,0,0,-1],[1,0,-1,0],[1,0,-1,1],
[1,0,-1,-1],[1,1,0,0],[1,1,0,1],[1,1,0,-1],[1,1,0,1],[1,1,0,-1],[1,-0,-1,0],[1,-0,-1,1],[1,-0,-1,-1],
[1,-1,0,0],[1,-1,0,1]])

ys= np.array ([[0,0,0,0],[0,0,0,0],[0,0,0,0],[0,1,1,1],[1,-1,0,0],[0,0,0,0],[1,0,0,0],
[1,-1,0,0],[0,0,1,1],[1,-1,0,0],[0,1,0,0],[0,1,0,0],[1,0,0,0],[1,-1,0,0],[1,0,0,0],
[1,-1,0,0],[0,1,0,0],[1,0,0,0],[1,-1,-1,0,0],[0,0,1,1],[1,-1,-1,0,0],
[1,0,0,0],[1,-1,-1,0,0],[1,0,0,0],[1,0,0,0],[1,-1,0,0],[0,0,1,1],[0,0,1,1]])

[ ] model.summary()
Model: "sequential_2"
-----
Layer (type)                Output Shape              Param #
-----
flatten_2 (Flatten)         (None, 4)                  0
dense_4 (Dense)             (None, 6)                  30
```

### FIGURA 8

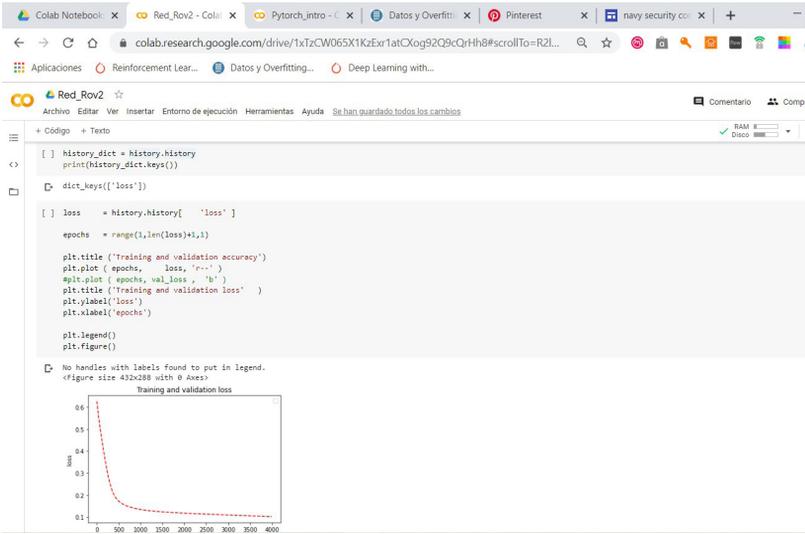


```
[ ] model.summary()
Model: "sequential_2"
-----
Layer (type)                Output Shape              Param #
-----
flatten_2 (Flatten)         (None, 4)                  0
dense_4 (Dense)             (None, 6)                  30
dense_5 (Dense)             (None, 4)                  28
-----
Total params: 58
Trainable params: 58
Non-trainable params: 0

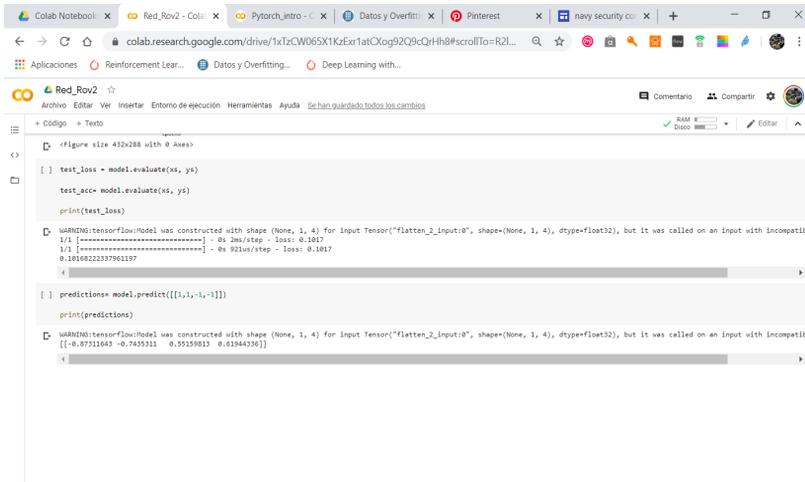
History = model.fit(xs,ys,learning_rate=0.03,epochs=4000)
[ ] %matplotlib inline
plt.plot(history.history['loss'])
plt.show()

1/1 [.....] - 0s 2ms/step - loss: 0.1235
Epoch: 1500/4000
1/1 [.....] - 0s 2ms/step - loss: 0.1235
Epoch: 1500/4000
1/1 [.....] - 0s 2ms/step - loss: 0.1235
Epoch: 1504/4000
1/1 [.....] - 0s 2ms/step - loss: 0.1235
Epoch: 1505/4000
1/1 [.....] - 0s 2ms/step - loss: 0.1234
Epoch: 1506/4000
1/1 [.....] - 0s 2ms/step - loss: 0.1234
Epoch: 1507/4000
1/1 [.....] - 0s 2ms/step - loss: 0.1234
Epoch: 1508/4000
1/1 [.....] - 0s 2ms/step - loss: 0.1234
Epoch: 1509/4000
```

## FIGURA 9

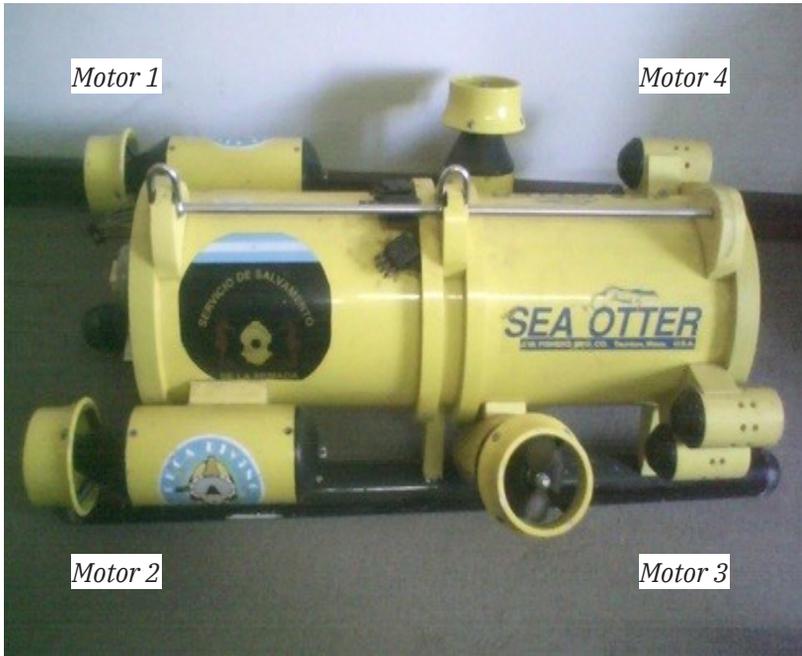


## FIGURA 10



Se muestra una imagen del dispositivo de montaje del sistema con la descripción de aquellas partes de interés.

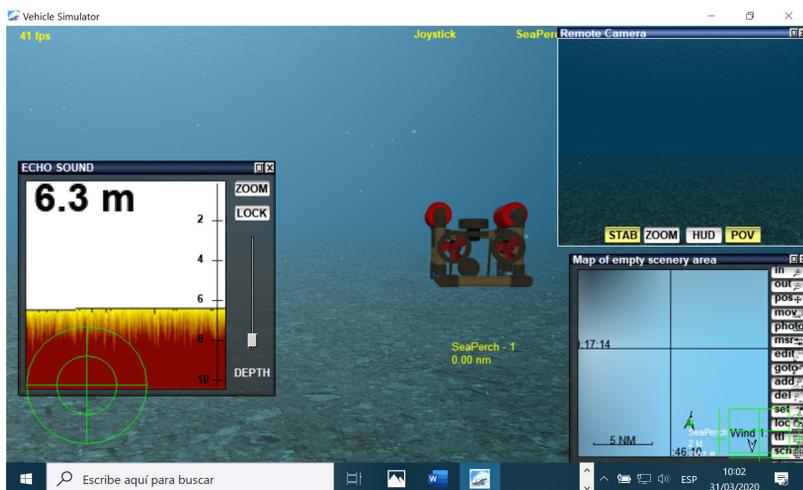
**FIGURA 11**  
*Dispositivo que se empleó de modelo.*



## 5. Próximos pasos

Antes de realizar un ensayo en el dispositivo en sí, se empleará una simulación. Esta se ejecutará empleando el sistema de simulación *vehicle simulator* sobre una estación de trabajo standard (PC con sistema operativo Windows 10). Este sistema tiene un nivel de realidad virtual alto y herramientas avanzadas, lo que nos permitirá verificar el resultado de la aplicación del algoritmo de aprendizaje.

**FIGURA 12**  
*Imagen de programa simulador vehicule simulator:*



## 6. Conclusiones

El sistema de navegación de una plataforma autónoma subacua como un ROV necesita un adecuado control para su traslado de una posición a otra, teniendo en cuenta el medio en que se desempeña. Las corrientes marinas o los obstáculos no previstos (cardúmenes, cables o redes de pesca, entre otros) ocasionan inconvenientes de operación que se hacen incontrolables a medida que el dispositivo se sumerge haciendo difícil llevar a cabo la misión, entre otras cosas por el arrastre que sufre el cable umbilical más allá de cierta profundidad. Las soluciones a este problema han sido asignarme más autonomía al ROV, manteniendo una unión física con este de otras maneras.

Esta autonomía se podrá dar aplicando a su sistema de control un paradigma de aprendizaje automático. En nuestro caso, formalizado a través de una red neuronal entrenada

montada sobre una placa Arduino, cuya salida será una señal para cada uno de los motores del ROV. Esta salida se podrá modificar en función del tipo de propulsión que tenga el ROV. Como entradas, hemos pensado en dos dispositivos controladores PID, también montados sobre placas Arduino, que permitan la precepción de las diferencias entre el curso dado al ROV y el verdadero que este está siguiendo. Estos controladores obrarán sobre los ejes de movimiento vertical y horizontal, accionando sobre las hélices del ROV para estabilizarlo, una vez obtenido el resultado de la red neuronal.

Asimismo, hemos incorporado un sonar que posibilite la detección de obstáculos y reúna datos que ayuden a tener un panorama exterior del desplazamiento del ROV. La red neuronal puede, entonces, ser más certera en cuanto al mantenimiento efectivo del control autónomo.

Las pruebas al principio de este algoritmo se realizarán sobre un programa de simulación al cual se le modificarán algunos parámetros sobre el vehículo representado en pantalla para que pueda moverse de acuerdo a lo que indique la red neuronal. Con esto, se lo podrá aplicar después a una prueba real.

## Referencias bibliográficas

- Angeles, J. (2007). *Fundamentals of robotic mechanical systems: theory, methods, and algorithms.*, New York, Berlin: Springer.
- Beeson, B.; Hillenbrand, C. (2006). “Development of the Anti- Submarine Warfare Unmanned Surface Vehicle (ASW USV) Engineering Developmental Model to Meet US Navy Needs”, Disponible en: [http://www.gdrs.com/about/profile/pdfs/UDTPacific2006\\_4A3.pdf](http://www.gdrs.com/about/profile/pdfs/UDTPacific2006_4A3.pdf). Fecha de consulta: 11/03/2011
- Boland, R. (2009). “Vehicles Flies Rapidly into the Next Generation”, En: Revista SIGNAL del mes de septiembre de 2009, pages 27-30.
- Brandt, D. (2006). Comparison of A\* and RRT-Connect Motion Planning Techniques for Self-Reconfiguration Planning., *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, Vol., Oct. 2006, pp.892–897.
- Bresina, J.L., Jónsson, A.K., Morris, P.H., Rajan, K. (2005). Activity planning for the mars exploration rovers., *15th International Conference on Automated Planning and Scheduling, 2005*, pp. 40–49.
- Carsten, J., Rankin A., Ferguson D., Stentz A. (2009). Global planning on the Mars Exploration Rovers: Software integration and surface testing., *J. Field Robot.*, Vol. 26, No. 4, 2009, pp. 337–357.
- Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., Thrun, S. (2005). *Principles of robot motion: theory, algorithms, and implementation.*, Cambridge, Mass.: MIT Press.
- Christensen, R.; Fogh, N. (2008). *Inertial Navigation System. Master project Section of Automation and Control Department of Electronic Systems Faculty of Engineering,*

Science and Medicine Fredrik Bajers Vej 7 C3 DK-9220  
Aalborg Øst Dinamarca.

- Cortes, J., Jaillet, L., Simeon, T. (2008). Disassembly Path Planning for Complex Articulated Objects., Robotics, IEEE Transactions on, Vol.24, No.2, April 2008a, pp.475– 481.
- Cortes, J., Jaillet, L., Simeon, T. (2008). Transition-based RRT for path planning in continuous cost spaces., Intelligent Robots and Systems, 2008, IROS 2008. IEEE/RSJ International Conference on, 22–26 Sept. 2008b, pp.2145– 2150.
- Craig, J. (2004). Introduction to Robotics: Mechanics and Control., Addison–Wesley, 1989. Funge, J.D., Artificial intelligence for computer games: an introduction., Wellesley, Mass.: Peters.
- Dunkan, F. (2014). “ROV Motion Control System”. Oslo: tesis doctoral.
- Goldberg, K., Planar Robot Simulator with Obstacle Avoidance (Configuration Space). Diponible en: <http://ford.ieor.berkeley.edu/cspace/>.
- Goldman, A. (1994). Path planning problems and solutions., in Book Path planning problems and solutions, Series Path planning problems and solutions, Vol.1, 1994, pp. 105–108.
- Goo Bong, C., Soo Gang, L., Sungmin, K., Byung-Ju, Y., Whee-Kuk, K., Se Min, O., Young Soo, K., Jong, I.I.P., Seong Hoon, O. (2005). A robot-assisted surgery system for spinal fusion., in Book A robot-assisted surgery system for spinal fusion, Series A robot-assisted surgery system for spinal fusion, 2005, pp. 3015-3021.
- Hsu, D. (2000). Randomized single-query motion planning in expansive spaces, Tesis doctoral, Department of Computer Science, Stanford University.
- Hsu, D. (2008). Automation Science and Engineering, 2008. CASE 2008, IEEE International Conference on Aug. 2008,

pp.41–46, 23–26.

Jijie, X., Duindam, V., Alterovitz, R., Goldberg, K. (1981). Motion planning Lozano–Pérez, T., Auto-matic planning of manipulator transfer movements., IEEE Trans. Systems, Man, and Cybernetics SMC–11, No. 110, 1981, pp. 681–689– 681–689.

Keller, J. “Unmanned underwater vehicles take advantage of advanced sensors and processors for navigation and artificial intelligence” Disponible en: <http://www.militaryaerospace.com/index/display/article-display/336352/articles/military-aerospace-electronics/exclusive-content/video-unmanned-under>. Fecha de consulta: 11/03/2011.

Kenyon, H. S. (2009). “Robo boats on Horizon”, en: Revista SIGNAL de septiembre de 2009, pages 23-26.

Maio, D., Rizzi, S. (1995). CICERO: An Assistant for Planning Visits to a Museum., Lecture notes in computer science, No. 978, 1995, pp. 564.

Manh Hung V. (2009). Indoor Navigation System for Handheld Devices. A Major Qualifying Project Report submitted to the faculty of the Worcester Polytechnic Institute Worcester, Massachusetts, EE. UU. in partial fulfillment of the requirements of the Degree of Bachelor of Science on this day of October 22, 2009.

Mohd Rizal Arshad y Mohamed Yusof Radzak (2004). Design and Development of an Autonomous Underwater Vehicle Test-Bed (USM-AUV I). Underwater Robotics Research Group (URRG), School of Electrical and Electronic Engineering, University Science of Malaysia (USM), Engineering Campus, 14300 Seberang Prai Selatan, Pulau Pinang Malasia.

Moreno, H. A.; Saltaren, R.; Puglisi, L.; Carrera, I.; Cárdenas, P., Álvarez C. (2014). “Robótica Submarina: Conceptos, Elementos, Modelado y Control. Science Direct, Revista

Iberoamericana de Automática e Informática industrial.

- Nasri, N. (1) ; Andrieux, L. (2) ; Kachouri, A (1) y Samet, M. (1) (2009). Behavioral Modeling and Simulation of Underwater Channel LETI-ENIS, B.P.868-3018- (1) SFAX-Túnez ;(2) LATTIS-IUT Blagnac Toulouse, Francia.
- Prasanth, K. P. (2004). Modelling and Simulation of an Underwater Acoustic Communication Channel. Tesis Hochschule Bremen University of applied sciences Bremen, Alemania.
- Polle, D.; Mackworth, A.; Goebel, R. (2003). “Computation Intelligence: A logical Approach”, 2nd edition, Oxford University Press.
- Russel, S.; Norvig, P.; Hall, P. (2002) “Artificial Intelligence: A modern Approach” 2<sup>nd</sup> edition.
- Sim, C., Wan Sing, N., Ming Yeong, T., Yong-Chong, L., Tseng Tsai, Y. (2001). Image-guided manipulator compliant surgical planning methodology for robotic skull-base surgery, in Book Image-guided manipulator compliant surgical planning methodology for robotic skull-base surgery, Series Image guided manipulator compliant surgical planning methodology for robotic-skull-base surgery, pp. 26-29.
- Volker, B. (2008). “Unmanned Surface Vehicles – A survey”. Disponible en [http://www.skibstekniskskab.dk/public/dokumenter/Skibsteknisk/Download%20materiale/2008/10%20marts%2008/USVsurvey\\_DTU.pdf](http://www.skibstekniskskab.dk/public/dokumenter/Skibsteknisk/Download%20materiale/2008/10%20marts%2008/USVsurvey_DTU.pdf)  
Fecha de consulta: 04/03/2011

**Palabras clave:** *Autonomía – navegación – red neuronal – vehículo subacuático.*

**Keywords:** *Autonomy – navigation – neural network – underwater vehicle.*

## **Abstract**

The development of this article aimed at finding a way to show a solution to the autonomy model of movement and navigation of a device through the use of machine learning through a neural network. In this case it is an underwater device. The advantage of using this technique lies in the change of programming paradigm since it stops thinking logically and mathematically. Claims are no longer made to confirm whether or not the properties of the program meet the requirements. In neural networks, observations are made in an uncertain environment, experiments are carried out and statistics are used. This opens up a range of possibilities in areas not yet explored. For example, recognizing a person's face or speech for another human being is relatively straightforward. However, if we wanted to write a code to solve this, it would be very complicated. Machine learning solves this problem without telling the application what to do, only by "showing" examples you can get satisfactory results. This can extend the possibilities of creating even more complex systems. The application of neural networks in autonomous navigation has been used frequently by the Tesla Company in its autonomous car models that have already entered the market "to learn from their experience" on routes and roads. The scope of this article is for the purpose of showing the algorithm of a simple neural network sticking to an available underwater vehicle type. This model can be expanded in complexity by encompassing other systems such as the reference mobile.