

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**ESCUELA DE POSGRADO**



**PLANEAMIENTO DE TRAYECTORIA Y CONTROL DE UN ROBOT  
MÓVIL MARÍTIMO APLICANDO OPTIMIZACIÓN POR COLONIA DE  
HORMIGAS**

**Tesis para optar por el grado de  
Magister en Ingeniería de Control y Automatización**

**Autor: Ing. Ronald Uriol Cabrera**

**Asesor: Antonio Morán Cárdenas, Ph.D.**

**Jurado**

**Francisco Cuéllar Córdova, Mg.**

**Antonio Morán Cárdenas, Ph.D.**

**Pedro Crisóstomo Romero, Mg.**

**Lima – Perú**

**2016**

## Resumen

El presente trabajo desarrolla un algoritmo de planeamiento de trayectoria basado en Optimización por Colonia de Hormigas, en conjunto con un controlador óptimo, para el desplazamiento de un barco robot. Se presenta el modelo matemático estándar usado para vehículos marinos desarrollado por Thor Fossen. Asimismo se presentan las principales ideas detrás del marco meta-heurístico desarrollado por Marco Dorigo, llamado Optimización por Colonia de hormigas. El problema a resolver consiste en encontrar el camino más corto entre dos puntos dentro de un entorno (mapa) con obstáculos. Ambos algoritmos, tanto el de planeamiento de trayectoria como el de control óptimo, se implementaron en el entorno MatLab. Para poner a prueba el funcionamiento conjunto de estos dos algoritmos se usaron seis mapas distintos que buscan explorar el comportamiento de ambos algoritmos ante diversas variaciones de un caso base de comparación. El tiempo de convergencia de los algoritmos y los parámetros que ajustan sus desempeños fueron analizados.

A mi papá, mamá y hermanita.  
Por haberme ayudado a llegar tan lejos.

A Nadia, mi futura esposa.  
Por estar a mi lado en este largo camino.

# ÍNDICE

Resumen.....	ii
Dedicatoria.....	iii
Índice.....	iv
<b>CAPÍTULO 1. INTRODUCCIÓN.....</b>	<b>1</b>
1.1 Robots marítimos y aplicaciones.....	2
1.2 Sistemas de control para robot marítimos.....	3
1.3 Planeamiento y control de trayectoria.....	5
1.4 Estado del arte.....	5
1.4.1 Sistemas de planeamiento de trayectoria.....	5
1.4.2 Sistemas de control para ASV.....	7
1.4.3 ACO aplicado al planeamiento de trayectoria.....	7
1.5 Solución propuesta y objetivos.....	8
1.5.1 Objetivo principal.....	8
1.5.2 Objetivos específicos.....	8
<b>CAPÍTULO 2. MODELADO MATEMÁTICO DE UN BARCO.....</b>	<b>9</b>
2.1 DEFINICIÓN DE VARIABLES Y SISTEMAS DE REFERENCIA.....	9
2.1.1 Grados de libertad.....	9
2.1.2 Sistemas Coordinados.....	10
2.1.3 Definición de variables.....	12
2.2 Cinemática.....	13
2.3 Dinámica de cuerpo rígido.....	18
2.3.1 Matriz de inercia y Coriolis.....	19
2.3.2 Matriz de fuerzas restauradoras.....	24
2.3.3 Matriz de amortiguamiento.....	28
2.4 Modelo de 3 GRADOS DE LIBERTAD.....	28
2.5 Simulaciones del modelo.....	31
<b>CAPÍTULO 3: SISTEMA DE PLANEAMIENTO DE TRAYECTORIA.....</b>	<b>36</b>
3.1 Presentación del problema.....	36
3.2 Algoritmo de optimización por colonia de hormigas.....	36
3.2.1 Hormigas reales e inteligencia emergente.....	37

3.2.2	Hormigas artificiales.....	40
3.2.3	Aplicación en optimización combinatoria discreta.....	44
3.3	Diseño del algoritmo en MatLab.....	44
3.3.1	Mapa.....	44
3.3.2	Mapa de feromona.....	45
3.3.3	Estados de cada hormiga.....	46
3.3.4	Matriz de la colonia.....	48
3.3.5	Evaporación de la feromona.....	49
3.3.6	Diagramas de flujo.....	49
3.4	Pruebas realizadas.....	53
CAPÍTULO 4. SISTEMA DE CONTROL.....		62
4.1	Linealización por realimentación.....	65
4.2	Diseño del sistema de control.....	66
4.3	Simulaciones realizadas.....	67
CAPÍTULO 5. PRUEBAS DE CONTROL PARA SEGUIMIENTO DE TRAYECTORIA.....		73
CONCLUSIONES.....		87
BIBLIOGRAFÍA.....		89

# CAPÍTULO 1

## INTRODUCCIÓN

En general, se puede considerar como sistema todo dispositivo con entradas y salidas, donde las salidas son el resultado de procesar las entradas. En la investigación científica y en la solución de problemas de ingeniería, tanto científicos como ingenieros trabajan con sistemas. Dicho trabajo consiste en el modelamiento matemático y el control del comportamiento de los mismos. Diversos ejemplos de sistemas son: motor de combustión en automóviles, red de distribución eléctrica, alerones de aviones, hornos en panaderías y concentraciones de reactivos en una solución. Antiguamente el control de sistemas se llevaba a cabo enteramente mediante intervención humana. En esta forma de control rudimentario, un operario manipulaba un actuador que cambiaba el comportamiento del sistema, las acciones de control del operario se basaban en información del estado del sistema obtenida de sensores.

Con el pasar de los años se ha desarrollado un campo de la matemática aplicado al control de sistemas llamado teoría control. La teoría de control proporciona algoritmos que pueden ser implementados por sub-sistemas llamados controladores, los que se encargan de automatizar el sistema. La automatización de sistemas permite librar a científicos e ingenieros del control manual, la cual es una labor tediosa y poco productiva ya que actualmente dicho control es solo un paso intermedio dentro de un trabajo más amplio y trascendente.

Un caso de sistema automatizado es el sistema de navegación. El objetivo de estos sistemas de control es guiar un vehículo mientras se desplaza en un entorno con obstáculos y fronteras de tal forma que se cumplan los requerimientos establecidos. Estos requerimientos, que se plantean para el desplazamiento del vehículo, pueden estar relacionados con la forma y longitud del trayecto formado, el tiempo de recorrido o la cantidad de colisiones permitidas del vehículo con los obstáculos del entorno. Asimismo, los sistemas de navegación dependen de las características del vehículo que controlan; dentro de la amplia gama de vehículos que pueden ser automatizados, están los vehículos marítimos.

El presente trabajo está orientado a la automatización de un robot marítimo para que pueda navegar en una zona de mar costero. Ya que se trata de un prototipo, las pruebas del sistema de navegación se harán en una piscina, de tal forma que se evite posibles daños al robot; sin embargo, se procurará simular las perturbaciones que podría encontrar en una zona de mar costero. La Figura 1.1 muestra dicho robot.



**Figura 1.1.** Robot marítimo a ser usado para este proyecto. Fuente: propia.

El sistema de navegación que se implementará consta de dos partes: un planificador de trayectoria y un controlador óptimo para manejar el barco. El planificador de trayectoria tendrá como objetivo encontrar una ruta óptima entre dos puntos, en caso el robot se encuentre en un entorno con múltiples caminos. Una vez que se tiene una referencia de trayectoria a seguir, el controlador óptimo se encargará de controlar el desplazamiento del robot de tal forma que se siga esta referencia.

### **1.1 Robots marítimos y aplicaciones**

El uso de robots se ha ampliado en las últimas décadas. Se pueden encontrar aplicaciones robóticas en campos tan diversos como manufactura, medicina, exploración espacial, entre otros.

Las ventajas del uso de robots se pueden agrupar en cuatro categorías: seguridad, precisión, repetitividad y bajo costo. La seguridad es el factor más importante, ya que diversas tareas manuales llevadas a cabo por personas pueden resultar en lesiones. Un ejemplo del aumento en la seguridad para las personas gracias a los robots es el uso de autómatas inspectores de líneas de transmisión de alto voltaje; una tarea que antes era llevada a cabo únicamente por personas que arriesgaban su vida al acercarse a estos cables de alta tensión. El daño sufrido por un robot durante la ejecución de una tarea que antes era llevada a cabo por una persona significa que se ha evitado un daño humano, esta seguridad extra es la principal ventaja del uso de robots. La segunda ventaja que proporciona el uso de robots es la precisión, es decir, la exactitud con la que se lleva a cabo una tarea. Un robot es capaz de llevar a cabo una tarea de manera más precisa que un ser humano, debido a la naturaleza de sus sensores y actuadores, los cuales pueden ser diseñados de tal forma que sobrepasan ampliamente la capacidad humana de oír, ver, mover y sentir. Un ejemplo claro de esta ventaja se puede apreciar en las máquinas CNC, que perforan agujeros en un circuito electrónico; esta tarea podría ser llevada a cabo por una persona, pero un robot es capaz de realizar la perforación

con velocidad y exactitud superiores. La ventaja de la repetitividad se da gracias a que el desgaste mecánico que puede sufrir un robot a lo largo de su tiempo de vida es mucho más lento que el cansancio que siente una persona luego de trabajar durante un largo periodo. Un ejemplo de esta ventaja son las industrias de bienes de consumo masivo (papel, alimentos, envases, etc.). En estas industrias es posible lograr una producción masiva gracias al uso de varios robots que trabajan en paralelo de forma mucho más eficiente que una misma cantidad de personas lo harían. Finalmente, la ventaja del bajo costo se da gracias a que, una vez que se optimiza el diseño mecánico de un robot, las mejoras y el desempeño de su trabajo dependen del software que utilice. El ejemplo más resaltante en estos casos es el de los PLC (controladores lógicos programables), los cuales representan un solo gasto para la empresa, luego de lo cual su desempeño dependerá de la habilidad del programador designado para operarlo.

En el párrafo anterior, se presentaron varios ejemplos de aplicaciones de robots para señalar sus ventajas; estos ejemplos también muestran la gran variedad de aplicaciones de la robótica. Por ejemplo, uno de los tantos campos de aplicación de la robótica son los robots marítimos. Estos robots están diseñados para desplazarse en el agua, ya sea de manera superficial (botes o barcos) o de manera submarina. Asimismo, las aplicaciones de los robots marítimos son muy diversas, algunos ejemplos son: exploración submarina, búsqueda de bancos de peces, sensado de variables meteorológicas y patrullaje.

Independientemente de la tarea a la cual sea asignado el robot marítimo, este debe ser capaz de encontrar una trayectoria adecuada entre su posición inicial y un punto al cual se requiere que este se desplace. El sub-sistema dentro del robot que se encarga de encontrar dicha trayectoria se denomina sistema de planeamiento de trayectoria. Este sub-sistema se encarga de encontrar el mejor camino que puede seguir el robot para llegar a su destino dentro de un mapa que muchas veces presenta obstáculos y/o zonas restringidas. Una vez que la mejor trayectoria ha sido encontrada, otro sub-sistema se encarga de dirigir al robot de tal forma que describa dicha trayectoria, a este sub-sistema se le llama controlador. El controlador se encarga de generar las señales adecuadas que manejen a los actuadores del barco de tal forma que el robot pueda describir la trayectoria proporcionada por el planeador de trayectoria.

## **1.2 Sistemas de control para robot marítimos**

La teoría de control es un campo de la matemática aplicada que se orienta a recibir señales que representan el estado de un sistema, procesarlas matemáticamente y obtener señales que cambien el estado del sistema de tal forma que se alcance un comportamiento deseado.

El control de vehículos marítimos puede remontarse a 1908 con la invención del girocompás, un instrumento que cumplía con la función de realimentar el ángulo de rumbo (“yaw”) del barco. La Figura 1.2 muestra un girocompás antiguo. Al conocer el ángulo de rumbo del barco en cada momento, se pudo implementar sistemas de control que mantuviesen la dirección establecida. Los siguientes inventos, que facilitaron aún más el desarrollo del control de vehículos marítimos, fueron el sistema de posicionamiento global en la década de 1970, seguido por la cobertura global en 1994. Estos sistemas de posicionamiento dieron paso a sistemas de control capaces de realizar supervisión y seguimiento de trayectoria. De esta forma, el desarrollo de sistemas de posicionamiento local basados en hidroacústica, radio y ondas electromagnéticas permitieron el desarrollo de sistemas de posicionamiento dinámico para barcos. Cabe resaltar que esta tecnología es ampliamente usada en bases petroleras en alta-mar.



**Figura 1.2.** Girocompás antiguo. Fuente: Google.

El control de vehículos marítimos comprende tres sub-sistemas: planeamiento de trayectoria, navegación y control. El sistema de planeamiento de trayectoria se encarga de generar la referencia que el vehículo debe seguir. La referencia a seguir puede estar expresada en términos de posición, velocidad y/o aceleración. La trayectoria establecida se define de acuerdo con la misión del vehículo, clima, combustible disponible, etc. Asimismo, conforme el barco avanza, la referencia obtenida va actualizándose en caso de perturbaciones inesperadas en el trayecto recorrido por el barco. El sistema de navegación es el que proporciona la medición de las variables involucradas en el control del barco. Estas variables pueden ser posición, velocidad, aceleración, ángulos de giro, presión atmosférica, temperatura, etc. Este sub-sistema se encarga de realizar una verificación de la calidad de las señales adquiridas de cada sensor en el vehículo y también realiza las conversiones de unidades y marcos de referencia necesarios en cada variable. Finalmente, el sistema de control se encarga de generar las señales que van a los actuadores de tal forma que el barco siga la trayectoria establecida, por el planeamiento de trayectoria. Las señales que produce el sistema de control se generan a partir de las señales entregadas por el sistema de

navegación, las cuales brindan información acerca del estado actual del barco. Los algoritmos de control usados pueden ser lineales o no lineales.

Cuando se trabaja con sistemas de control para barcos, uno de los aspectos más importantes a tomar en cuenta es la naturaleza sub-actuada de estos sistemas. Esto significa que un vehículo marítimo posee más grados de libertad (formas en las que se puede describir su movimiento) que entradas independientes de control. Además, el control de vehículos marítimos sub-actuados ha sido un área activa de investigación en los últimos años, debido a que requiere estrategias propias de teoría control no-lineal para lograr aplicaciones altamente prácticas. Por ello, el problema principal en los vehículos marítimos sub-actuados es la ausencia de actuadores dedicados al control en el eje de deriva y arfada.

### **1.3 Planeamiento y control de trayectoria**

El planeamiento de trayectoria no es un problema exclusivo del control de vehículos marítimos, más aún, tampoco es exclusivo del ámbito de robótica. El planeamiento de trayectoria puede plantearse de manera abstracta como un problema de optimización combinatoria discreta, en el cual el camino encontrado viene a ser la solución cuya función de costo es menor. La función de costo vendría a ser la distancia recorrida al tomar dicho camino y el mapa vendría a ser el universo de posibles soluciones que pueden plantearse cuando se dan un punto de inicio y un punto final. Un caso bastante estudiado es el problema del vendedor viajero (TSP, por sus siglas en inglés). Bajo este planteamiento, los algoritmos usados para determinar el mejor camino a seguir por un barco en altamar no serían muy distintos a los algoritmos usados para determinar la mejor ruta que debe tomar un paquete de datos cuando se envía por internet desde una computadora hacia otra. Debido a su amplia aplicación, el problema de planeamiento de trayectoria ha sido ampliamente estudiado; se han planteado tanto algoritmos exactos como algoritmos aproximados. La razón de que existan algoritmos aproximados se debe al hecho que este problema pertenece a la categoría de problemas computacionales tipo NP, para los cuales solo se puede hallar una solución lo más óptima posible, pero no la más óptima con seguridad.

En el campo de la robótica se han desarrollado algoritmos de planeamiento de trayectoria para diversos tipos de robots, estos dependen principalmente del medio en el que se desplazan (medio terrestre, aéreo o marítimo).

## **1.4 Estado del arte**

### **1.4.1 Sistemas de planeamiento de trayectoria**

Debido a que el sistema de planeamiento de trayectoria depende de la forma cómo se desplaza el robot, se pueden encontrar propuestas y avances en tres áreas: robots terrestres,

robots aéreos y robots marítimos. El caso de planeamiento de trayectoria para robots terrestres es el más sencillo de todos, puesto que la mayoría de casos se encuentran en mapas bidimensionales, planos de terreno sólido en los cuales se desplaza el vehículo.

La representación del entorno del robot y la implementación del algoritmo son los dos componentes principales del planeamiento de trayectoria que están altamente influenciados en uno por el otro. La representación del entorno nos proporciona estructuras de datos que muestran las orientaciones y posiciones de objetos (y del robot) en el ambiente donde este se mueve, así como las áreas donde el movimiento está permitido (área transitable) o restringido (obstáculos). Las formas más usuales de representar el entorno del robot son:

- ✓ Diagramas de Voronoi
- ✓ Grillas regulares y grillas de ocupancia
- ✓ Conos generalizados
- ✓ Quadtree

Una vez que se cuenta con una representación del entorno del robot, se procede a elegir el algoritmo más adecuado para dicha representación. El algoritmo más adecuado es aquel que manipula de manera más eficiente la estructura de datos proporcionada por la representación del entorno elegida. Los algoritmos existentes son los siguientes:

- ✓ A\* ("A estrella")
- ✓ A\* modificado
- ✓ Campo potencial artificial
- ✓ Algoritmos genéticos

En cuanto a robots terrestres, se han llevado a cabo abundantes estudios. En [1] se diseñó un sistema de planeamiento de trayectoria basado en el algoritmo A\* para un mapa obtenido a partir de un escaneo del ambiente; en [2] se resolvió el problema usando curvas de Bézier en un mapa con restricciones de corredor; en [4] se usó el método de árbol aleatorio para encontrar la trayectoria de un robot terrestre que busca estacionarse en un área urbana; en [5] se usó el método de árbol cuaternario en un mapa desconocido en el cual el robot encuentra nuevos obstáculos conforme avanza. En [3] se evaluaron de manera comparativa los algoritmos de planeamiento de trayectoria para robots terrestres más usados en el "reto urbano" organizado por DARPA.

En robots aéreos, el planeamiento de trayectoria se vuelve un problema tridimensional. Este campo de estudio es más reciente y aún queda mucho por investigar debido al creciente uso de drones aéreos. En [53] se hace un estudio resumen que muestra 5 categorías de algoritmos para realizar planeamiento de trayectoria para robots aéreos. En [54] se muestra

un método basado en A\* mientras que en [55] se resuelve el planeamiento de trayectoria usando algoritmos genéticos.

En robots marítimos podríamos separar el planeamiento en dos categorías: bidimensional para el caso de robots marítimos superficiales (ASV por sus siglas en inglés) y tridimensionales para el caso de robots marítimos submarinos (AUV por sus siglas en inglés). En el presente trabajo se tomó en cuenta los estudios realizados para robots de superficie, más no los orientados a robots submarinos. En [56] se realiza un planeamiento de trayectoria para robots marítimos usando campos potenciales.

#### **1.4.2 Sistemas de control para ASV**

El control de robots marítimos de superficie (ASV), es un área de investigación activa en la cual numerosos estudios han sido realizados. En [6] se describe el diseño de un ASV basado en microcontroladores, con comunicación inalámbrica y de bajo costo. La mayoría de embarcaciones, tanto las de tamaño industrial como las robóticas de pequeña escala, son sistemas sub-actuados. Debido a esto, hay una gran cantidad de investigaciones de control de ASV sub-actuados. En [7] se propone un control basado en dinámica neuronal en conjunto con el algoritmo de retro-propagación. En [8] se propone un control adaptativo robusto ante un ambiente con perturbaciones, tanto constantes como variantes en el tiempo. En [9] se utilizó un controlador apoyado de un observador de perturbaciones. En [10] se diseñó un controlador de realimentación de estado. Como se puede observar en los trabajos mencionados, las formas de controlar un ASV sub-actuado son bastante diversas.

#### **1.4.3 ACO aplicado al planeamiento de trayectoria**

La optimización por colonia de hormigas (ACO por sus siglas en inglés) es una metaheurística; es decir, un conjunto de conceptos algorítmicos que guían el desarrollo de soluciones a problemas de optimización combinatoria. El ACO se inspira en el comportamiento de las colonias de hormigas cuando estas buscan alimento en su entorno. Durante esta actividad, las hormigas demuestran que un sistema conformado por numerosos individuos sencillos puede resolver una tarea compleja: determinar el camino más corto entre el nido y el alimento. Debido a esto, es natural que el ACO sea aplicado al problema de planeamiento de trayectoria. Cuando se aplica a este problema, se crea una colonia de hormigas virtuales que exploran un mapa en busca del mejor camino entre dos puntos. Asimismo en [11] se lleva a cabo un estudio del desempeño del ACO en función de sus parámetros cuando se lo aplica al problema de planeamiento de trayectoria. Mientras que en [12] se presenta una versión modificada del ACO basada en varios enfoques previos a la implementación de este algoritmo. Por último en [13] y [14] se utiliza el ACO para el planeamiento de trayectoria de un robot móvil.

## **1.5 Solución propuesta y objetivos**

En el presente trabajo se diseñará un algoritmo de planeamiento de trayectoria basado en optimización por colonia de hormigas (ACO), para un robot marítimo de superficie. El sistema de control a utilizar será un controlador óptimo; debido a que el sistema es no-lineal, se usará una linealización por realimentación. La estrategia de control para este sistema sub-actuado será expresar los estados del modelo del barco robot en función de otras variables, de tal forma que se tenga una cantidad igual de estados del nuevo modelo y entradas de control.

### **1.5.1 Objetivo principal**

Diseño de un sistema de planeamiento y control de trayectoria aplicando Optimización por Colonia de Hormigas y linealización por realimentación, de manera que el vehículo autónomo describa la trayectoria más corta para alcanzar una posición final deseada.

### **1.5.2 Objetivos específicos**

- ✓ Obtener el modelo dinámico de un robot marítimo superficial.
- ✓ Implementación de un algoritmo que determine la trayectoria óptima basado en Optimización por Colonia de Hormigas en un ambiente con obstáculos y frontera usando el entorno MatLab®
- ✓ Implementación de un algoritmo de control que lleve al robot marítimo a través de la trayectoria planificada.
- ✓ Lograr un tiempo de convergencia de ambos algoritmos (planeamiento de trayectoria y control) lo suficientemente bajo (menos de 20 segundos) para que pueda implementarse sin ningún problema en un robot marítimo de baja velocidad.

## CAPÍTULO 2

### MODELADO MATEMÁTICO DE UN BARCO

En este capítulo se demostrará que un barco puede modelarse haciendo uso las ecuaciones vectoriales mostradas en 2.1 y 2.2.

$$\dot{\eta} = J_{\theta}(\eta)v \quad (2.1)$$

$$M\dot{v} + C(v)v + D(v)v + g(\eta) + g_0 = \tau + \tau_{wind} + \tau_{wave} \quad (2.2)$$

La deducción del modelo explicado en este capítulo se obtuvo de Fossen 2011.

Además, según Fahimi, se puede modelar a un barco robot con las siguientes ecuaciones.

$$\ddot{x} = \frac{1}{m_{11}}(f_x + F \cos \psi) \quad (2.3)$$

$$\ddot{y} = \frac{1}{m_{22}}(f_y + F \sin \psi) \quad (2.4)$$

$$\ddot{\psi} = \frac{1}{I_{zz}}(f_{\psi} + T) \quad (2.5)$$

#### 2.1 DEFINICIÓN DE VARIABLES Y SISTEMAS DE REFERENCIA

##### 2.1.1 Grados de libertad

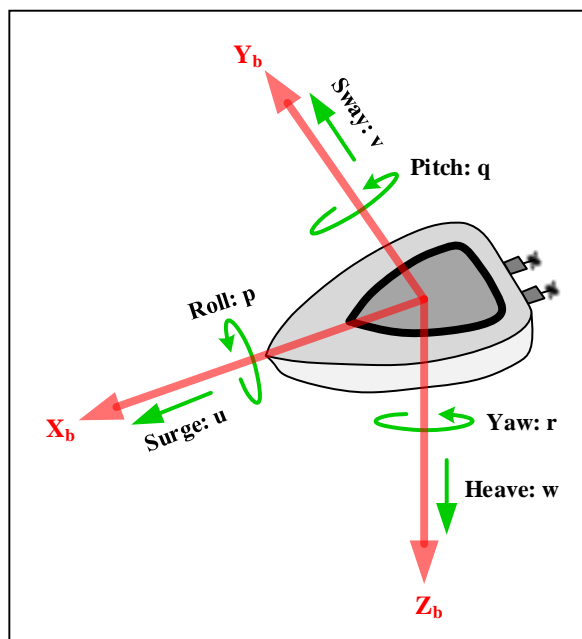
Para modelar el movimiento del barco, este será analizado como un cuerpo sujeto a seis grados de libertad. Al analizarlo como un cuerpo y no como un punto, las coordenadas del barco no solo representan su posición en un sistema coordenado, sino también su orientación respecto al mismo. Los seis grados de libertad asignados al barco son: “surge”, “sway”, “heave”, “roll”, “pitch” y “yaw”. La Figura 2.1 muestra las velocidades lineales y angulares descompuestas en los grados de libertad en un barco. El SNAME (sociedad de arquitectos navales e ingenieros marinos, por sus siglas en inglés), definió en 1950 una nomenclatura estándar para el análisis de vehículos marítimos. La Tabla 2.1 resume las principales variables definidas para cada grado de libertad.

El vector de coordenadas usadas para representar el movimiento del barco tendrá seis componentes. Las tres primeras componentes y sus derivadas en el tiempo representan la posición, velocidad y aceleración traslacional del barco; mientras que las tres coordenadas siguientes y sus derivadas en el tiempo representan la orientación, velocidad y aceleración angular del mismo. Asimismo, las fuerzas que actúan sobre el barco pueden descomponerse en tres componentes correspondientes a los grados de libertad “surge”, “sway” y “heave”;

mientras que un torque sobre el barco se puede descomponer en tres torques correspondientes a los grados de libertad “roll”, “pitch” y “yaw”.

**Tabla 2.1.** Variables definidas por el SNAME (1950).

Grado de libertad	Posiciones y ángulos de Euler	Velocidades lineales y angulares	Fuerzas y momentos
<i>Surge</i>	$x$	$u$	$X$
<i>Sway</i>	$y$	$v$	$Y$
<i>Heave</i>	$z$	$w$	$Z$
<i>Roll</i>	$\varphi$	$p$	$K$
<i>Pitch</i>	$\theta$	$q$	$M$
<i>Yaw</i>	$\psi$	$r$	$N$



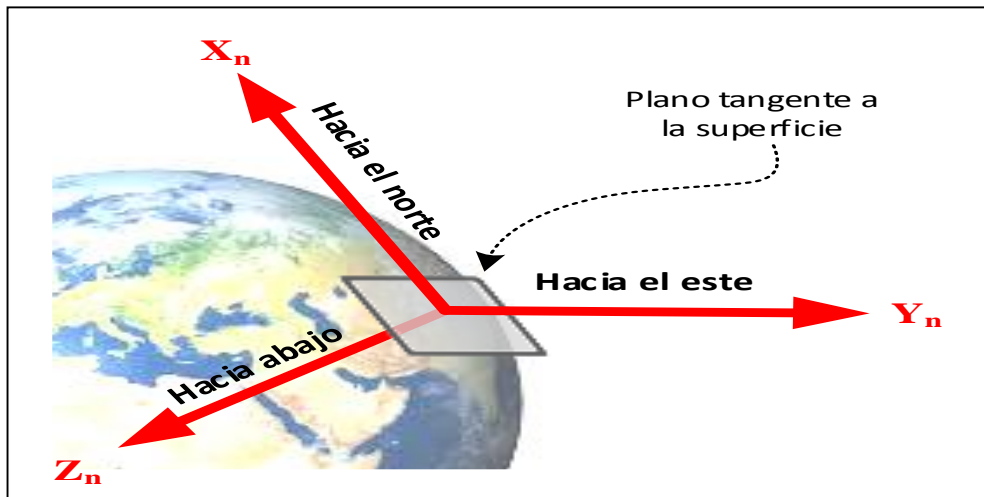
**Figura 2.1.** Velocidades descompuestas en los grados de libertad de un barco Fuente: propia.

### 2.1.2 Sistemas coordenados

Para el correcto análisis de la posición y orientación de un barco se requiere de dos sistemas coordenados: NED y BODY.

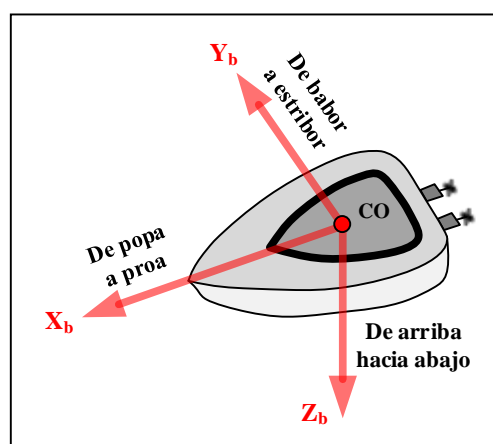
El sistema coordenado NED es un sistema que se asume inercial. Esta suposición se puede dar gracias a que el barco navegará en un área pequeña en relación con la superficie de la tierra, por lo cual se puede asumir que el barco se desplaza sobre un plano tangente al

planeta, a esto se le conoce como navegación de tierra plana. La rotación del planeta afecta tanto a este sistema coordenado como al barco. Este sistema tiene como centro un punto sobre la superficie del agua. El eje “X” apunta hacia el norte, el eje “Y” hacia el este y el eje “Z” apunta hacia debajo de forma perpendicular a la superficie. El punto de origen de este sistema coordenado se denomina  $o_n$ .



**Figura 2.2.** Sistema coordenado NED. Navegación de tierra plana. Fuente: propia.

El sistema coordenado BODY es un marco de referencia fijo al barco. Los ejes de este sistema son elegidos de tal forma que coincidan con los ejes de inercia del barco. El eje X va desde popa hacia proa; el eje Y va desde babor hacia estribor; y el eje Z va de la parte superior del barco hacia la inferior. El punto de origen de este sistema es llamado  $o_b$ , y se elige de tal forma que coincida con un punto ubicado a la mitad del barco en la línea de agua, a este punto se le denomina CO. La Figura 2.3 ilustra este sistema coordenado.



**Figura 2.3.** Sistema coordenado BODY. Fuente: propia.

Es importante notar que la orientación de los ejes en este sistema coordenado depende de la orientación del IMU (unidad de medidas inerciales) dentro del barco. Este sensor se encarga de proporcionar las aceleraciones y velocidades, tanto lineales como angulares,

descompuestas en tres componentes que corresponden a los tres ejes tomados desde el IMU. Los ejes coordenados proporcionados por el IMU serán los ejes coordenados del sistema de coordenadas BODY. Por lo tanto, las velocidades y aceleraciones se miden en el marco de referencia BODY, lo cual hace necesaria una transformación de coordenadas que involucre rotación de ejes coordenados, la cual se desarrollará más adelante en este capítulo.

### 2.1.3 Definición de variables

Debido a la existencia de dos marcos de referencia, NED y BODY, en el análisis de movimiento del barco, es necesario adoptar una notación estándar que identifique el sistema coordinado usado. La Tabla 2.2 resume los vectores (y sus componentes) a usar en adelante. Se hará uso extensivo de la notación vectorial para presentar, de manera más ordenada, las ecuaciones de los 6 grados de libertad del barco. El superíndice en la parte derecha de cada vector indica el marco de referencia en el que están expresadas sus componentes; “b” para BODY y “n” para NED.

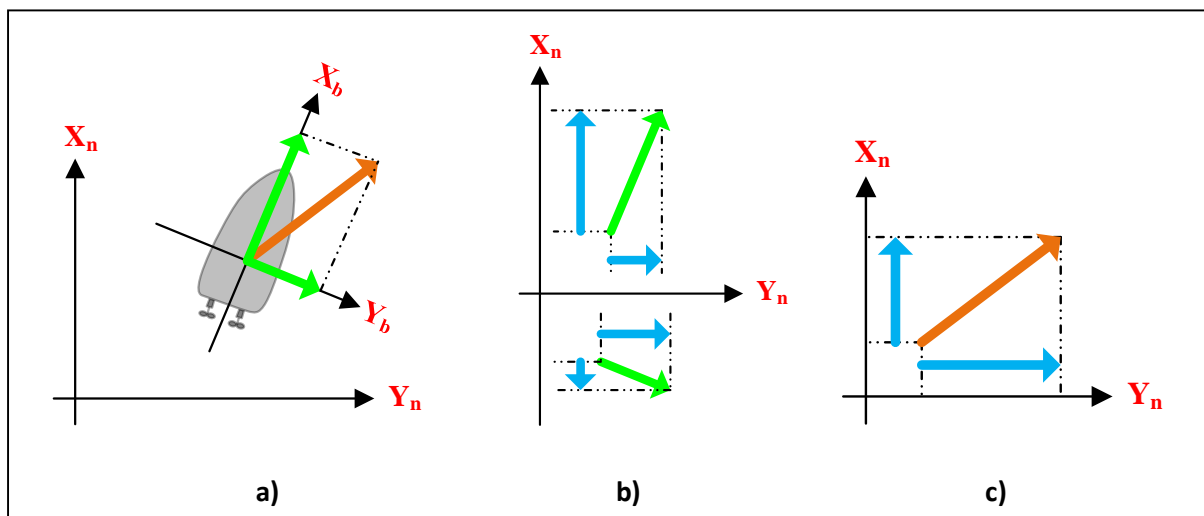
**Tabla 2.2.** Vectores usados en el análisis dinámico de un barco.

Vector y componentes	Descripción
$\mathbf{p}_{b/n}^n = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$	Distancia del punto $o_b$ respecto al punto $o_n$ expresado en el sistema NED.
$\mathbf{\Theta}_{nb} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}$	Ángulos de Euler del sistema BODY respecto al sistema NED
$\mathbf{v}_{b/n}^b = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$	Velocidad lineal del punto $o_b$ dentro del sistema coordinado NED expresado en el sistema coordinado BODY.
$\mathbf{\omega}_{b/n}^b = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$	Velocidad angular del sistema coordinado BODY respecto al sistema coordinado NED, expresado en el sistema BODY.
$\mathbf{f}_b^b = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$	Fuerza con línea de acción que pasa por el punto $o_b$ expresado en el sistema coordinado BODY.
$\mathbf{m}_b^b = \begin{bmatrix} K \\ M \\ N \end{bmatrix}$	Momento que actúan alrededor del punto $o_b$ expresado en el sistema coordinado BODY.
$\boldsymbol{\eta} = \begin{bmatrix} \mathbf{p}_{b/n}^n \\ \mathbf{\Theta}_{nb} \end{bmatrix}$	Vector de posición y ángulos.
$\mathbf{v} = \begin{bmatrix} \mathbf{v}_{b/n}^b \\ \mathbf{\omega}_{b/n}^b \end{bmatrix}$	Vector de velocidades lineales y angulares.
$\boldsymbol{\tau} = \begin{bmatrix} \mathbf{f}_b^b \\ \mathbf{m}_b^b \end{bmatrix}$	Vector de fuerzas y momentos actuando sobre el barco, descritas en el marco de referencia BODY

## 2.2 CINEMÁTICA

La Ecuación 2.1:  $\dot{\eta} = J_{\Theta}(\eta)v$ , es la primera parte del modelo dinámico de un barco propuesto por T. Fossen en 1991. Esta ecuación representa la relación geométrica entre el vector de velocidades lineales y angulares expresado en el sistema coordenado BODY y el mismo expresado en el sistema coordenado NED. Geométricamente,  $J_{\Theta}(\eta)$  es una matriz de rotación que lleva a las velocidades medidas en el marco de referencia BODY hacia el marco de referencia NED mediante una serie de rotaciones continuas determinadas por los ángulos  $\Theta$ .

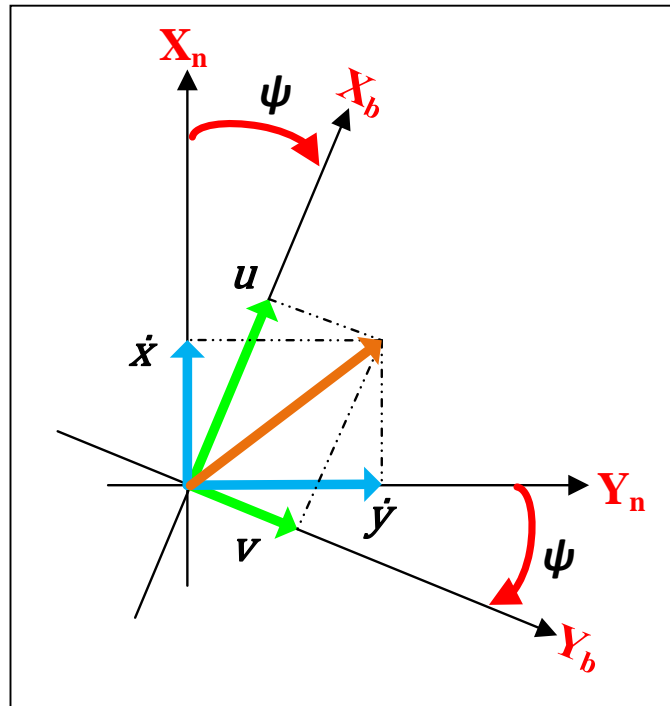
La necesidad de realizar una rotación del vector de velocidades lineales y angulares se puede apreciar mejor en el caso de dos dimensiones. La Figura 2.4 muestra un barco cuya velocidad se representa por un vector naranja. La lectura del IMU nos da las componentes de la velocidad del barco en el sistema coordenado BODY, representadas en la Figura 2.4.a con vectores verdes. Dado que la posición del barco se expresa en el sistema coordenado NED, se descomponen las componentes de velocidad medidas en el sistema coordenado BODY en el sistema coordenado NED tal que se pueda calcular las componentes resultantes de la velocidad en el sistema NED.



**Figura 2.4.** Caso bidimensional de un barco y la rotación de sus velocidades. a) componentes de la velocidad obtenidas en el sistema coordenado BODY. b) descomposición en el sistema coordenado NED de las componentes de velocidad obtenidas en el marco de referencia BODY. c) componentes resultantes de la velocidad del barco expresadas en el marco de referencia NED. Fuente: propia.

El procedimiento de descomposición vectorial nos muestra el significado físico del cambio de coordenadas en el que está expresada la velocidad del barco. Sin embargo, la forma cómo se lleva a cabo este cambio de coordenadas puede representarse mediante la rotación simple

del sistema coordenado BODY respecto al sistema coordenado NED. Esta rotación se muestra en la Figura 2.5.



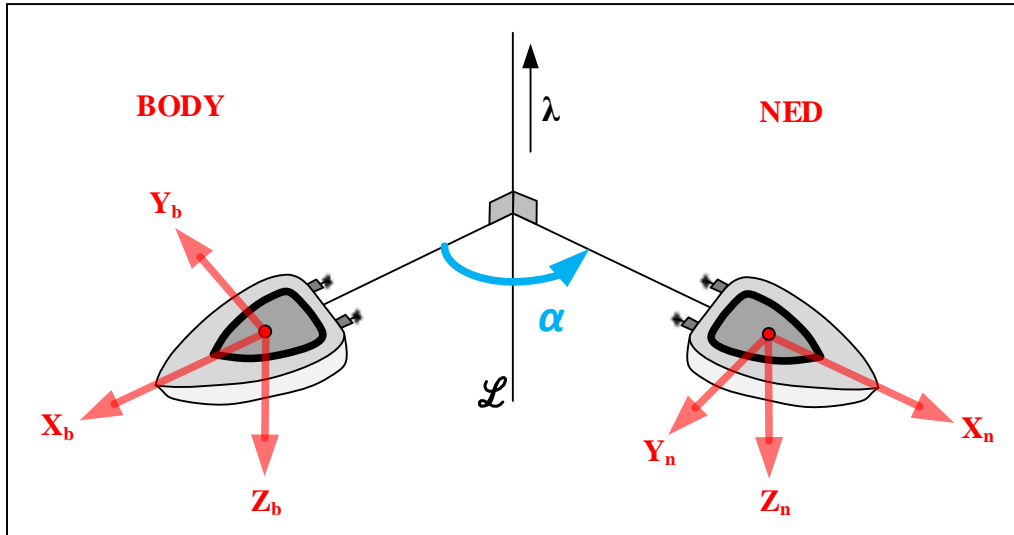
**Figura 2.5.** Interpretación geométrica del cambio de sistema coordenado. Fuente: propia.

De la Figura 2.5 podemos establecer que el vector de velocidad está dado por  $\mathbf{v}_{b/n}^b = \begin{bmatrix} u \\ v \end{bmatrix}$  en el sistema coordenado BODY. En este caso “u” y “v” vendrían a ser los vectores de color verde. Mientras que el mismo vector de velocidad expresado en coordenadas NED sería  $\mathbf{v}_{b/n}^n = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$ , donde “ $\dot{x}$ ” y “ $\dot{y}$ ” son los vectores de color azul. Para relacionar ambos pares de vectores se usa una matriz de rotación en función del ángulo de rotación entre el marco BODY y el marco NED, llamado  $\psi$ . La Ecuación 2.6 muestra esta relación geométrica.

$$\mathbf{v}_{b/n}^n = \mathbf{R}_b^n \mathbf{v}_{b/n}^b = \begin{pmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{pmatrix} \mathbf{v}_{b/n}^b \quad (2.6)$$

Donde el subíndice de la matriz  $\mathbf{R}_b^n$  determina el sistema coordenado inicial y el superíndice determina el eje coordenado al final de la rotación. Los elementos de  $\mathbf{R}_b^n$  pueden ser deducidos de las relaciones trigonométricas entre los vectores de cada sistema coordenado.

Ahora que se tiene bien definida la rotación de un sistema coordenado de dos dimensiones respecto a otro, se puede generalizar la idea de rotación para vectores en tres dimensiones. En la rotación mostrada en la Figura 2.5 se puede visualizar en tres dimensiones en la Figura 2.6.



**Figura 2.6.** Rotación del sistema BODY hasta quedar orientado con el sistema NED.

Fuente: propia.

Para generalizar la idea de rotación en tres dimensiones nos apoyaremos de las siguientes definiciones:

**Definición 2.1 (Rotación simple):** el movimiento de un cuerpo rígido en un sistema coordenado B relativo a otro cuerpo rígido o a un sistema coordenado A, se denomina rotación simple si existe una línea L, llamada eje de rotación, cuya orientación relativa a ambos sistemas coordenados permanece constante a lo largo de todo el movimiento.

**Definición 2.2 (Teorema de Euler):** cualquier cambio en la orientación relativa de dos cuerpos rígidos o sistemas coordenados A y B, puede ser producido por medio de una rotación simple de B en A.

En la Figura 2.6, se puede observar que el marco de referencia BODY gira alrededor del eje de rotación L un ángulo  $\alpha$  hasta quedar en la posición del marco de referencia NED. El vector unitario  $\lambda$  es paralelo a los ejes  $Z_b$  y  $Z_n$  en este caso. De manera similar a la Ecuación 2.6, se puede deducir una matriz  $R_b^n$  tal que relacione un vector de velocidad en el sistema NED y lo exprese en términos de un vector en el sistema BODY, como se muestra en la Ecuación 2.7.

$$v_{b/n}^n = R_b^n v_{b/n}^b \quad , \quad R_b^n := R_{(\lambda, \psi)} = I_{3 \times 3} + \sin(\alpha) S(\lambda) + [1 - \cos(\alpha)] S^2(\lambda) \quad (2.7)$$

En esta ecuación se expresa el vector de velocidades lineales  $v_{b/n}^n$  en función de las coordenadas del vector  $v_{b/n}^b$  mediante una matriz  $R_b^n$  que es función de un ángulo  $\alpha$  y un vector unitario  $\lambda$ . El ángulo  $\alpha$  es el ángulo que describe el vector desde el inicio hasta el final de la rotación. El vector unitario  $\lambda$  es un vector paralelo al eje de rotación L.

$$\mathbf{R}_b^n = \begin{bmatrix} \cos \alpha + \lambda_1^2(1 - \cos \alpha) & \lambda_1 \lambda_2(1 - \cos \alpha) - \lambda_3^2 \sin \alpha & \lambda_1 \lambda_3(1 - \cos \alpha) + \lambda_3^2 \sin \alpha \\ \lambda_2 \lambda_1(1 - \cos \alpha) + \lambda_3^2 \sin \alpha & \cos \alpha + \lambda_2^2(1 - \cos \alpha) & \lambda_2 \lambda_3(1 - \cos \alpha) - \lambda_3^2 \sin \alpha \\ \lambda_3 \lambda_1(1 - \cos \alpha) - \lambda_3^2 \sin \alpha & \lambda_3 \lambda_2(1 - \cos \alpha) + \lambda_3^2 \sin \alpha & \cos \alpha + \lambda_3^2(1 - \cos \alpha) \end{bmatrix} \quad (2.8)$$

El operador  $\mathbf{S}(\lambda)$  es denominado producto cruz mediante matriz antisimétrica. Las siguientes definiciones establecen las propiedades y el uso de esta matriz.

**Definición 2.3 (Matriz antisimétrica):** una matriz  $S$  se denomina antisimétrica si cumple  $S = -S^T$ . Esto implica que los elementos de la diagonal principal de  $S$  son iguales a cero y los elementos fuera de la diagonal satisfacen  $s_{ij} = -s_{ji} \quad \forall i \neq j$ .

**Definición 2.4 (Operador producto-cruz):** el producto cruz de dos vectores se define usando una matriz antisimétrica de la siguiente manera:

$$\lambda \times \mathbf{a} = \mathbf{S}(\lambda)\mathbf{a} \quad \mathbf{S}(\lambda) = -\mathbf{S}^T(\lambda) = \begin{bmatrix} 0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & 0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & 0 \end{bmatrix} \quad \lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \quad (2.9)$$

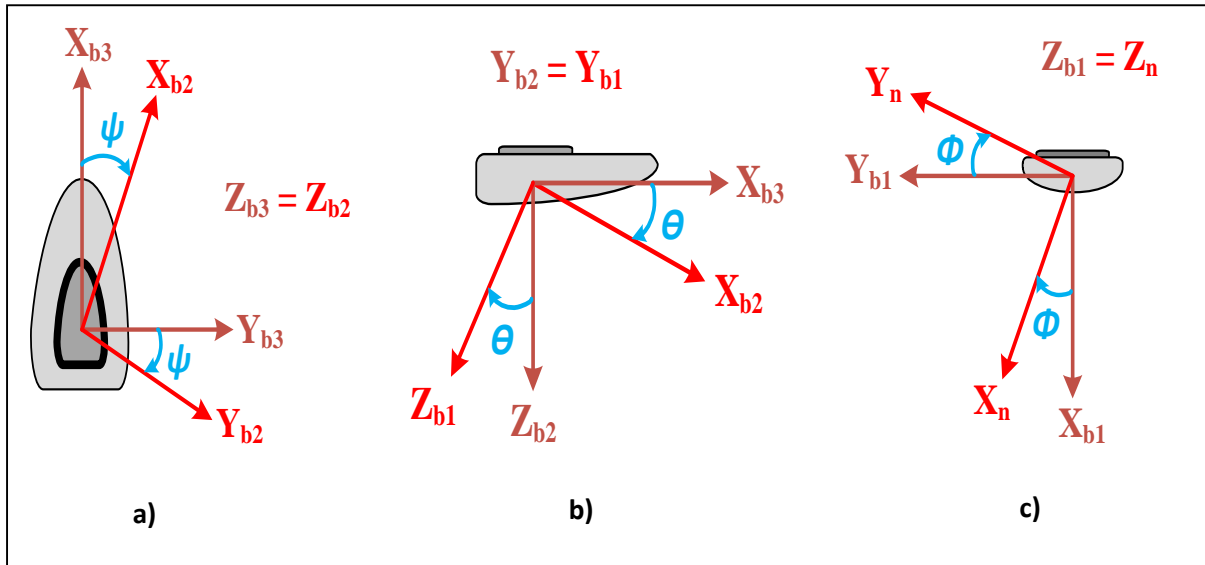
En la Figura 2.6, se mostró una rotación simple para el caso de dos sistemas coordenados, NED y BODY, cuyos ejes “Z” son paralelos y cuyos planos XY también son paralelos. En este caso particular, una sola rotación simple basta para hacer coincidir la orientación de ambos sistemas coordenados. Este alineamiento es lo que permite expresar un vector del sistema NED en función de un vector en el sistema BODY. Sin embargo, el sistema BODY se encuentra desalineado del sistema NED de tal forma que para alinearlos con el sistema NED es necesario realizar tres rotaciones simples en un orden establecido por el SNAME 1950. La primera rotación simple toma como eje  $Z_b$  y gira un ángulo  $\psi$  denominado “yaw”. La segunda rotación simple toma como eje  $Y_b$  (en su nueva orientación) y gira un ángulo  $\theta$  denominado “pitch”. Finalmente, la tercera rotación simple se realiza en torno al eje  $X_b$  (en su nueva orientación) y describe un ángulo  $\phi$  llamado “roll”. Estas tres rotaciones simples se representan algebraicamente por las ecuaciones 2.10, 2.11 y 2.12; donde los vectores unitarios  $\vec{i}$ ,  $\vec{j}$  y  $\vec{k}$  son paralelos a los ejes de cada rotación simple.

$$\mathbf{R}_{(\vec{k}, \psi)} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

$$\mathbf{R}_{(\vec{j}, \theta)} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.11)$$

$$\mathbf{R}_{(\vec{i}, \phi)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (2.12)$$

Debido a que se realizan tres rotaciones simples consecutivas, la orientación de los ejes del sistema coordinado BODY cambia tres veces hasta llegar a estar alineados con el sistema coordinado NED. Este proceso se ilustra en la Figura 2.7.



**Figura 2.7.** Rotaciones simples consecutivas. a) Rotación alrededor del eje Z. b) Rotación alrededor del eje Y. c) Rotación alrededor del eje X. Fuente: propia.

La expresión algebraica que lleva a cabo estas rotaciones sucesivas se muestra a continuación.

$$\mathbf{R}_b^n(\Theta_{nb}) = \mathbf{R}_{(\vec{k}, \psi)} \mathbf{R}_{(\vec{j}, \theta)} \mathbf{R}_{(\vec{i}, \phi)} \quad \Theta_{nb} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \quad (2.13)$$

$$\mathbf{R}_b^n(\Theta_{nb}) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (2.14)$$

$$\mathbf{R}_b^n(\Theta_{nb}) = \begin{bmatrix} \cos \psi \cos \theta & -\sin \psi \cos \phi + \cos \psi \sin \theta \sin \phi & \sin \psi \sin \phi + \cos \psi \cos \theta \sin \phi \\ \sin \psi \cos \theta & \cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi & -\cos \psi \sin \phi + \sin \theta \sin \psi \cos \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (2.15)$$

Ahora que se tiene la matriz de rotación que alinea el sistema BODY con el sistema NED, se puede plantear la Ecuación 2.16, la cual relaciona el vector de velocidad lineal expresada en el marco BODY y en el marco NED.

$$\dot{\mathbf{p}}_{b/n}^n = \mathbf{R}_b^n(\Theta_{nb}) \mathbf{v}_{b/n}^b \quad (2.16)$$

De la misma forma que la velocidad lineal en ambos marcos de referencia (BODY y NED) se relaciona mediante una matriz de rotación, la velocidad angular del barco también puede ser expresada como una rotación de sus componentes en el sistema BODY hasta llegar a

alinearse con el sistema NED. Las ecuaciones 2.17, 2.18 y 2.19 muestran esta relación análoga al caso de las velocidades lineales.

$$\boldsymbol{\omega}_{b/n}^b = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_{x,\phi}^T \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathbf{R}_{x,\phi}^T \mathbf{R}_{y,\theta}^T \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} = \mathbf{T}_{\Theta}^{-1}(\Theta_{nb}) \dot{\Theta}_{nb} \quad (2.17)$$

$$\mathbf{T}_{\Theta}(\Theta_{nb}) = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix} \quad (2.18)$$

$$\dot{\Theta}_{nb} = \mathbf{T}_{\Theta}(\Theta_{nb}) \boldsymbol{\omega}_{b/n}^b \quad (2.19)$$

Ahora que se tiene una relación entre componentes de velocidad en el sistema BODY y NED, tanto lineales como angulares, se puede establecer una ecuación que englobe las ecuaciones 2.16 y 2.19.

$$\begin{bmatrix} \dot{\mathbf{p}}_{b/n}^n \\ \dot{\Theta}_{nb} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b^n(\Theta_{nb}) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{T}_{\Theta}(\Theta_{nb}) \end{bmatrix} \begin{bmatrix} \mathbf{v}_{b/n}^b \\ \boldsymbol{\omega}_{b/n}^b \end{bmatrix} \quad \rightarrow \quad \dot{\boldsymbol{\eta}} = \mathbf{J}_{\Theta}(\boldsymbol{\eta}) \mathbf{v} \quad (2.20)$$

### 2.3 DINÁMICA DE CUERPO RÍGIDO.

La Ecuación 2.2 es la segunda parte del modelo planteado por Fossen, esta se vuelve a presentar a continuación:

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} + \mathbf{g}(\boldsymbol{\eta}) + \mathbf{g}_0 = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} \quad (2.2)$$

En esta ecuación se tienen los siguientes elementos:

- $\mathbf{M}$ : matriz de inercia.
- $\mathbf{C}(\mathbf{v})$ : matriz de Coriolis.
- $\mathbf{D}(\mathbf{v})$ : matriz de amortiguamiento.
- $\mathbf{g}(\boldsymbol{\eta})$ : vector generalizado de fuerzas gravitacionales y de empuje.
- $\mathbf{g}_0$ : fuerzas de restauración estáticas y momentos debidos a sistemas de balasto.
- $\boldsymbol{\tau}$ : fuerzas y momentos originados externamente.
- $\boldsymbol{\tau}_{wind}$ : fuerzas y momentos causados por el viento.
- $\boldsymbol{\tau}_{wave}$ : fuerzas y momentos causados por el oleaje.

El vector  $\boldsymbol{\eta}$  contiene las 6 coordenadas correspondientes a los 6 grados de libertad del barco: tres posiciones y tres ángulos de Euler. El vector  $\mathbf{v}$  contiene las velocidades lineales y angulares del barco. A continuación, se analizarán los elementos mencionados. En algunos

casos, se deducirá las expresiones algebraicas usando leyes de la física, mientras que en otros casos se verá que la obtención de las matrices es experimental.

### 2.3.1 Matriz de inercia y Coriolis

Para la obtención de las matrices  $\mathbf{M}$  y  $\mathbf{C}(\mathbf{v})$  se usará la segunda mecánica vectorial, la cual hace uso de la formulación Newton-Euler. Esta formulación es una combinación de la segunda ley de Newton y los axiomas de Euler (conservación del momento lineal y angular). Las leyes de Newton fueron publicadas en la obra *Philosophia Naturalis Principia Mathematica* en 1687. Los axiomas de Euler fueron publicados en *Novi Commentarii Academiae Scientiarum Imperialis Petropolitanae*. La segunda ley de Newton se muestra en la Ecuación 2.21, mientras que los axiomas de Euler se muestran en la Ecuación 2.22.

$$m \dot{\vec{v}}_{g/i} = \vec{f}_g \quad (2.21)$$

$$\frac{{}^i d}{dt} \vec{p}_g = \vec{f}_g \quad \vec{p}_g = m \vec{v}_{g/i} \quad \frac{{}^i d}{dt} \vec{h}_g = \vec{m}_g \quad \vec{h}_g = I_g \vec{\omega}_{b/i} \quad (2.22)$$

Donde  $\vec{f}_g$  y  $\vec{m}_g$  son las fuerzas y torques que actúan en el centro de gravedad del barco, mostrado en la Figura 2.8 como el punto CG;  $\vec{v}_{g/i}$  y  $\vec{\omega}_{b/i}$  son las velocidades lineal y angular, respectivamente, del sistema BODY respecto a un sistema coordenado inercial  $\{i\}$ ; y  $\vec{p}_g$  y  $\vec{h}_g$  son los momentos lineal y angular respectivamente.

Para deducir las expresiones de  $\mathbf{M}$  y  $\mathbf{C}(\mathbf{v})$  se tomarán en cuenta dos suposiciones:

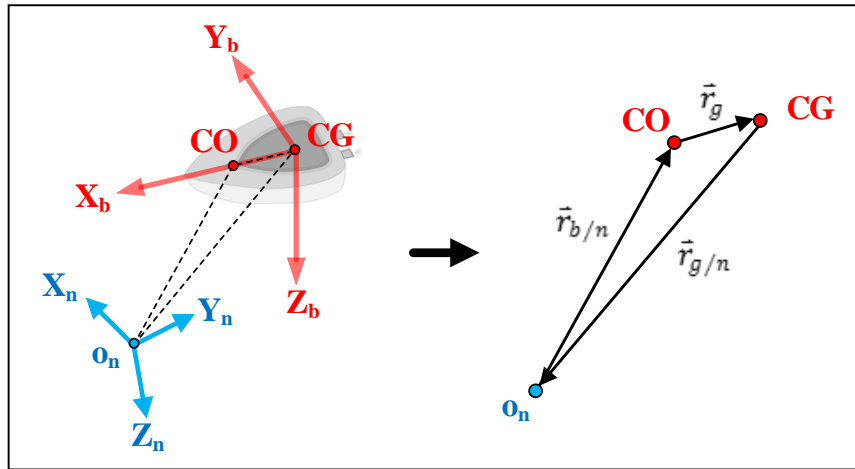
1. El barco es un cuerpo rígido, por lo tanto no se consideran las interacciones dinámicas entre elementos individuales de masa del barco:  $\frac{{}^b d}{dt} \vec{r}_g = \vec{0}$
2. El sistema coordenado NED es inercial, por lo tanto no se considera el efecto de las fuerzas debido a la rotación del planeta:  $\vec{v}_{g/i} \approx \vec{v}_{g/n}$ ,  $\vec{\omega}_{b/i} \approx \vec{\omega}$

También se hará uso de la derivada en el tiempo de un vector dentro de un sistema coordenado en movimiento, que en este caso es el sistema BODY. Esta derivada en el tiempo se muestra en la Ecuación 2.23.

$$\frac{{}^i d}{dt} \vec{a} = \frac{{}^n d}{dt} \vec{a} = \frac{{}^b d}{dt} \vec{a} + \vec{\omega}_{b/i} \times \vec{a} \quad (2.23)$$

La deducción de las matrices  $\mathbf{M}$  y  $\mathbf{C}(\mathbf{v})$  se hará en tres análisis: movimiento traslacional de CG, movimiento rotacional alrededor de CG, transformación de coordenada, movimiento traslacional del punto CO, movimiento rotacional del punto CO, y finalmente se condensará todo en las ecuaciones de movimiento de un cuerpo rígido con 6 grados de libertad. Para estos análisis se tendrá en cuenta la situación general mostrada en la Figura 2.8, en la que

se muestra en azul el marco de referencia NED y en rojo el marco de referencia BODY. Asimismo, se muestra la relación entre vectores de posición del punto CO (origen del sistema BODY) y el punto CG (centro de gravedad del barco).



**Figura 2.8.** Análisis de cuerpo rígido de un barco. Fuente: propia.

### Movimiento de traslación del centro de gravedad

En la Figura 2.8, se puede observar que el vector de posición del centro de gravedad respecto al marco de referencia NED puede ser expresado como la suma de los otros dos vectores mostrado en la misma Figura, como se muestra en la Ecuación 2.24.

$$\vec{r}_{g/n} = \vec{r}_{b/n} + \vec{r}_g \quad (2.24)$$

A dicha expresión se le toma la derivada en el tiempo y, tomando en cuenta la propiedad mostrada en la Ecuación 2.23, se obtiene la Ecuación 2.25.

$$\vec{v}_{g/n} = \vec{v}_{b/n} + \left( \frac{d\vec{r}_g}{dt} + \vec{\omega}_{b/n} \times \vec{r}_g \right) \quad (2.25)$$

Gracias a que asumimos que el barco es un cuerpo rígido, el primer término dentro del paréntesis se elimina, y nos queda la Ecuación 2.26.

$$\vec{v}_{g/n} = \vec{v}_{b/n} + \vec{\omega}_{b/n} \times \vec{r}_g \quad (2.26)$$

Se reemplaza esta expresión en el primer axioma de Euler mostrado anteriormente en la Ecuación 2.22, y se obtiene la expresión mostrada en la Ecuación 2.27.

$$\frac{d}{dt} (m \vec{v}_{g/n}) = \vec{f}_g \quad \Rightarrow \quad m (\dot{\vec{v}}_{b/n} + \vec{\omega}_{b/n} \times \vec{v}_{g/n}) = \vec{f}_g^b \quad (2.27)$$

Si se reemplaza los productos cruz por matrices antisimétricas, se obtiene la Ecuación 2.28.

$$m \dot{\vec{v}}_{b/n} + m S(\vec{\omega}_{b/n}) \vec{v}_{g/n} = \vec{f}_g^b \quad (2.28)$$

### Movimiento de rotación del centro de gravedad

De manera similar, se analizará el movimiento rotacional del barco usando la Ecuación 2.23 aplicada al vector momento angular, con lo cual se obtiene la Ecuación 2.29.

$$\frac{{}^n d}{dt} \vec{h}_g = \frac{{}^n d}{dt} (I_g \vec{\omega}_{b/n}) = \frac{{}^b d}{dt} I_g \vec{\omega}_{b/n} + \vec{\omega}_{b/n} \times (I_g \vec{\omega}_{b/n}) \quad (2.29)$$

Donde,  $I_g$  es la matriz de inercia alrededor del centro de gravedad, la cual se define como:

$$I_g = \begin{pmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{pmatrix} \quad (2.30)$$

Dicha ecuación se reemplaza en el segundo axioma de Euler,

$$\frac{{}^n d}{dt} (I_g \vec{\omega}_{b/n}) = \vec{m}_g^b \quad \Rightarrow \quad I_g \dot{\vec{\omega}}_{b/n} - (I_g \vec{\omega}_{b/n}) \times \vec{\omega}_{b/n} = \vec{m}_g^b \quad (2.31)$$

Si se usa la notación de matriz antisimétrica se tiene la Ecuación 2.32.

$$I_g \dot{\vec{\omega}}_{b/n} - S(I_g \vec{\omega}_{b/n}) \vec{\omega}_{b/n} = \vec{m}_g^b \quad (2.32)$$

### Ecuaciones de movimiento respecto al centro de gravedad

Se unen las ecuaciones 2.28 y 2.32 en una sola ecuación matricial, mostrada en la Ecuación 2.33.

$$\begin{pmatrix} m I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_g \end{pmatrix} \begin{pmatrix} \dot{\vec{v}}_{b/n} \\ \dot{\vec{\omega}}_{b/n} \end{pmatrix} + \begin{pmatrix} m S(\vec{\omega}_{b/n}) & 0_{3 \times 3} \\ 0_{3 \times 3} & -S(I_g \vec{\omega}_{b/n}) \end{pmatrix} \begin{pmatrix} \vec{v}_{g/n} \\ \vec{\omega}_{b/n} \end{pmatrix} = \begin{pmatrix} \vec{f}_g^b \\ \vec{m}_g^b \end{pmatrix} \quad (2.33)$$

Donde podemos definir la matriz de masas  $M_{RB}^{CG}$  y la matriz de Coriolis-centrípeta  $C_{RB}^{CG}$ , ambas respecto al centro de gravedad, tal que se tenga la Ecuación 2.34.

$$M_{RB}^{CG} \begin{pmatrix} \dot{\vec{v}}_{b/n} \\ \dot{\vec{\omega}}_{b/n} \end{pmatrix} + C_{RB}^{CG} \begin{pmatrix} \vec{v}_{g/n} \\ \vec{\omega}_{b/n} \end{pmatrix} = \begin{pmatrix} \vec{f}_g^b \\ \vec{m}_g^b \end{pmatrix} \quad (2.34)$$

### Transformación de coordenadas

Ahora que se tienen las ecuaciones de movimiento para el centro de gravedad, se puede deducir una transformación para que los vectores queden expresados respecto al origen del marco BODY, mostrado en la Figura 2.8 como el punto CO. Para esto volvemos a recurrir a la Ecuación 2.23, con la cual tenemos que:

$$\vec{v}_{g/n}^b = \vec{v}_{b/n}^b + \vec{\omega}_{b/n}^b \times \vec{r}_g^b \quad (2.35)$$

Esta expresión puede expresarse en términos que no contenga un producto cruz si se utiliza la notación con matriz antisimétrica, como se muestra a continuación

$$\vec{v}_{g/n}^b = \vec{v}_{b/n}^b + S^T(\vec{r}_g^b)\vec{\omega}_{b/n}^b \quad (2.36)$$

Con esta ecuación se puede establecer la siguiente matriz de transformación, en la cual la velocidad angular no se altera, ya que esta estaba definida respecto al marco de referencia BODY desde un comienzo; solo cambia la velocidad lineal.

$$\begin{pmatrix} \vec{v}_{g/n}^b \\ \vec{\omega}_{b/n}^b \end{pmatrix} = \begin{pmatrix} I_{3 \times 3} & S^T(\vec{r}_g^b) \\ 0_{3 \times 3} & I_{3 \times 3} \end{pmatrix} \begin{pmatrix} \vec{v}_{b/n}^b \\ \vec{\omega}_{b/n}^b \end{pmatrix} \rightarrow \begin{pmatrix} \vec{v}_{g/n}^b \\ \vec{\omega}_{b/n}^b \end{pmatrix} = H(\vec{r}_g^b) \begin{pmatrix} \vec{v}_{b/n}^b \\ \vec{\omega}_{b/n}^b \end{pmatrix} \quad (2.37)$$

Si reemplazamos la Ecuación 2.37 en la Ecuación matricial 2.34 obtenida anteriormente, obtenemos la siguiente expresión:

$$H^T(\vec{r}_g^b)M_{RB}^{CG}H(\vec{r}_g^b) \begin{pmatrix} \dot{\vec{v}}_{b/n}^b \\ \dot{\vec{\omega}}_{b/n}^b \end{pmatrix} + H^T(\vec{r}_g^b)C_{RB}^{CG}H(\vec{r}_g^b) \begin{pmatrix} \vec{v}_{g/n}^b \\ \vec{\omega}_{b/n}^b \end{pmatrix} = H^T(\vec{r}_g^b) \begin{pmatrix} \vec{f}_g^b \\ \vec{m}_g^b \end{pmatrix} \quad (2.38)$$

Donde podemos re-definir la matriz de masa, Coriolis-centrípeta y vector de fuerzas y momentos como:

$$M_{RB}^{CO} \begin{pmatrix} \dot{\vec{v}}_{b/n}^b \\ \dot{\vec{\omega}}_{b/n}^b \end{pmatrix} + C_{RB}^{CO} \begin{pmatrix} \vec{v}_{g/n}^b \\ \vec{\omega}_{b/n}^b \end{pmatrix} = \begin{pmatrix} \vec{f}_b^b \\ \vec{m}_b^b \end{pmatrix} \quad (2.39)$$

Si expandimos detalladamente los elementos de las nuevas matrices en la Ecuación 2.39, obtenemos:

$$\begin{pmatrix} m I_{3 \times 3} & -m S(\vec{r}_g^b) \\ m S(\vec{r}_g^b) & I_g - m S^2(\vec{r}_g^b) \end{pmatrix} \begin{pmatrix} \dot{\vec{v}}_{b/n}^b \\ \dot{\vec{\omega}}_{b/n}^b \end{pmatrix} + \begin{pmatrix} m S(\vec{\omega}_{b/n}^b) & -m S(\vec{\omega}_{b/n}^b)S(\vec{r}_g^b) \\ m S(\vec{r}_g^b)S(\vec{\omega}_{b/n}^b) & -S((I_g - m S^2(\vec{r}_g^b))\vec{\omega}_{b/n}^b) \end{pmatrix} \begin{pmatrix} \vec{v}_{g/n}^b \\ \vec{\omega}_{b/n}^b \end{pmatrix} = \begin{pmatrix} \vec{f}_b^b \\ \vec{m}_b^b \end{pmatrix} \quad (2.40)$$

### Movimiento de traslación del punto CO

Para esto, se tomará la primera fila de la Ecuación 2.40 y se le dará una forma más conveniente. El proceso se muestra en las ecuaciones 2.41, 2.42 y 2.43.

$$m(\dot{\vec{v}}_{b/n}^b - S(\vec{r}_g^b)\dot{\vec{\omega}}_{b/n}^b + S(\vec{\omega}_{b/n}^b)\vec{v}_{g/n}^b - S(\vec{\omega}_{b/n}^b)S(\vec{r}_g^b)\vec{\omega}_{b/n}^b) = \vec{f}_b^b \quad (2.41)$$

$$m(\dot{\vec{v}}_{b/n}^b + S^T(\vec{\omega}_{b/n}^b)\vec{r}_g^b + S(\vec{\omega}_{b/n}^b)\vec{v}_{g/n}^b + S^2(\vec{\omega}_{b/n}^b)\vec{r}_g^b) = \vec{f}_b^b \quad (2.42)$$

$$m(\dot{\vec{v}}_{b/n}^b + \dot{\vec{\omega}}_{b/n}^b \times \vec{r}_g^b + \vec{\omega}_{b/n}^b \times \vec{v}_{g/n}^b + \vec{\omega}_{b/n}^b \times (\vec{\omega}_{b/n}^b \times \vec{r}_g^b)) = \vec{f}_b^b \quad (2.43)$$

### Movimiento de rotación del punto CO

Para esto, se tomará la segunda fila de la Ecuación 2.40 y se le dará una forma más conveniente utilizando el teorema de ejes paralelos. Este teorema nos brinda la matriz de inercia respecto al origen del marco BODY. La Ecuación 2.44 muestra este teorema.

$$I_b = I_g - m S^2(\vec{r}_g^b) \quad (2.44)$$

El proceso se muestra en las ecuaciones 2.45, 2.46 y 2.47.

$$mS(\vec{r}_g^b)\dot{\vec{v}}_{b/n} + (I_g - mS^2(\vec{r}_g^b))\dot{\vec{\omega}}_{b/n} + mS(\vec{r}_g^b)S(\vec{\omega}_{b/n}^b)\vec{v}_{g/n} - S\left((I_g - mS^2(\vec{r}_g^b))\vec{\omega}_{b/n}^b\right)\vec{\omega}_{b/n} = \vec{m}_b^b \quad (2.45)$$

$$I_b\dot{\vec{\omega}}_{b/n} + S(\vec{\omega}_{b/n}^b)I_b\vec{\omega}_{b/n} + mS(\vec{r}_g^b)\dot{\vec{v}}_{b/n} + mS(\vec{r}_g^b)S(\vec{\omega}_{b/n}^b)\vec{v}_{g/n} = \vec{m}_b^b \quad (2.46)$$

$$I_b\dot{\vec{\omega}}_{b/n} + \vec{\omega}_{b/n}^b \times I_b\vec{\omega}_{b/n} + m\vec{r}_g^b \times (\dot{\vec{v}}_{b/n} + \vec{\omega}_{b/n}^b \times \vec{v}_{g/n}) = \vec{m}_b^b \quad (2.47)$$

### Ecuaciones de movimiento de un cuerpo rígido.

Finalmente, si se aplica la nomenclatura del SNAME (1950), mostrada en la Tabla 2.2, a las ecuaciones 2.43 y 2.47, se tendrá el modelo no-lineal del barco describiendo la evolución de sus 6 grados de libertad. Este modelo está descrito por las siguientes ecuaciones:

$$m[\dot{u} - vr + wq - x_g(q^2 + r^2) + y_g(pq - \dot{r}) + z_g(pr + \dot{q})] = X \quad (2.48)$$

$$m[\dot{v} - wp + ur - y_g(r^2 + p^2) + z_g(qr - \dot{p}) + x_g(qp + \dot{r})] = Y \quad (2.49)$$

$$m[\dot{w} - uq + vp - z_g(p^2 + q^2) + x_g(rp - \dot{p}) + y_g(rq + \dot{p})] = Z \quad (2.50)$$

$$I_x\dot{p} + (I_z - I_y)qr - (\dot{r} + pq)I_{xz} + (r^2 - q^2)I_{yz} + (pr - \dot{q})I_{xy} + m[y_g(\dot{w} - uq + vp) - z_g(\dot{v} - wp + ur)] = K \quad (2.51)$$

$$I_y\dot{q} + (I_x - I_z)rp - (\dot{p} + qr)I_{xy} + (p^2 - r^2)I_{zx} + (qp - \dot{r})I_{yz} + m[z_g(\dot{u} - vr + wq) - x_g(\dot{w} - uq + vp)] = M \quad (2.52)$$

$$I_z\dot{r} + (I_y - I_x)pq - (\dot{q} + rp)I_{yz} + (q^2 - p^2)I_{xy} + (rq - \dot{p})I_{zx} + m[x_g(\dot{v} - wp + ur) - y_g(\dot{u} - vr + wq)] = N \quad (2.53)$$

En este modelo, las primeras tres ecuaciones representan el movimiento de traslación del barco, mientras que las tres últimas representan el movimiento rotacional del mismo. La representación vectorial de este modelo se muestra en la Ecuación 2.54.

$$\mathbf{M}_{RB}\dot{\mathbf{v}} + \mathbf{C}_{RB}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau}_{RB} \quad (2.54)$$

Esto vendría a ser la segunda parte del modelo al que se deseaba llegar; es decir, la Ecuación 2.2. Este modelo proporciona toda la información necesaria para obtener información detallada de la dinámica del barco. Sin embargo, la complejidad del modelo es bastante alta. Los elementos de la matriz  $\mathbf{M}_{RB}$  son detallados en la Ecuación 2.55. Esta matriz es única y satisface las propiedades descritas en la Ecuación 2.56.

$$\mathbf{M}_{RB} = \begin{pmatrix} m \mathbf{I}_{3 \times 3} & -m \mathbf{S}(\mathbf{r}_g^b) \\ m \mathbf{S}(\mathbf{r}_g^b) & \mathbf{I}_b \end{pmatrix} = \begin{pmatrix} m & 0 & 0 & 0 & m z_g & -m y_g \\ 0 & m & 0 & -m z_g & 0 & m x_g \\ 0 & 0 & m & m y_g & -m x_g & 0 \\ 0 & -m z_g & m y_g & I_x & -I_{xy} & -I_{xz} \\ m z_g & 0 & -m x_g & -I_{yx} & I_y & -I_{yz} \\ -m y_g & m x_g & 0 & -I_{zx} & -I_{zy} & I_z \end{pmatrix} \quad (2.55)$$

$$\mathbf{M}_{RB} = \mathbf{M}_{RB}^T > \mathbf{0} \quad \dot{\mathbf{M}}_{RB} = \mathbf{0}_{6 \times 6} \quad (2.56)$$

A diferencia de la matriz  $\mathbf{M}_{RB}$ , la matriz  $\mathbf{C}_{RB}(\vec{v})$  puede tener varias representaciones. Se elegirá aquella mostrada en la Ecuación 2.57 (Fossen and Fjellstad, 1995). Los elementos de dicha representación se muestran detalladamente en la Ecuación 2.57.

$$\mathbf{C}_{RB} = \begin{pmatrix} \mathbf{0}_{3 \times 3} & -m \mathbf{S}(\mathbf{v}_1) - m \mathbf{S}(\mathbf{v}_2) \mathbf{S}(\mathbf{r}_g^b) \\ -m \mathbf{S}(\mathbf{v}_1) + m \mathbf{S}(\mathbf{r}_g^b) \mathbf{S}(\mathbf{v}_2) & -\mathbf{S}(\mathbf{I}_b \mathbf{v}_2) \end{pmatrix} \quad (2.57)$$

$$\mathbf{C}_{RB} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -m(y_g q + z_g r) & m(y_g p + w) & m(z_g p - v) \\ m(x_g q - w) & -m(z_g r + x_g p) & m(z_g q + u) \\ m(x_g r + v) & m(y_g r - u) & -m(x_g p + y_g q) \end{pmatrix}$$

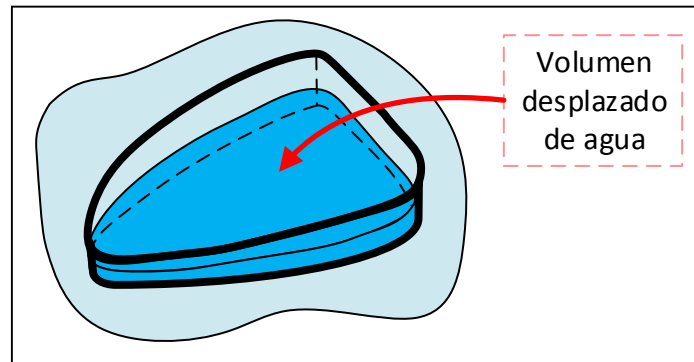
$$\left. \begin{pmatrix} m(y_g q + z_g r) & -m(x_g q - w) & -m(x_g r + v) \\ -m(y_g p + w) & m(z_g r + x_g p) & -m(y_g r - u) \\ -m(z_g p - v) & -m(z_g q + u) & m(x_g p + y_g q) \\ 0 & -I_{yz} q - I_{xz} p + I_z r & I_{yz} r + I_{xy} p - I_{yx} q \\ I_{yz} q + I_{xz} p - I_z r & 0 & -I_{xz} r - I_{xy} q + I_{yx} p \\ -I_{yz} r - I_{xy} p + I_{yx} q & I_{xz} r + I_{xy} q - I_{yx} p & 0 \end{pmatrix} \right) \quad (2.58)$$

Como se puede apreciar en las ecuaciones 2.55 y 2.58, la complejidad del modelo de 6DOF es alta, por lo cual es usado, en general, para maniobras avanzadas. Para el control de un barco robot se usará solo un modelo de 3DOF.

### 2.3.2 Matriz de fuerzas restauradoras

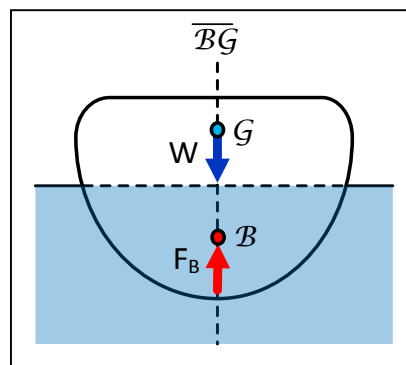
En esta parte se deducirán los elementos de las matrices  $\mathbf{g}(\boldsymbol{\eta})$  y  $\mathbf{g}_0$  de la Ecuación 2.2. Para esto, se hará uso del análisis de estabilidad de cuerpos semi-sumergidos en un fluido,

particularmente barcos. La Figura 2.9 muestra un barco en estado de equilibrio en aguas calmadas; el volumen de agua desplazado en estado de equilibrio se denomina  $\nabla$  (nabla).



**Figura 2.9.** Barco robot en estado de equilibrio dentro de aguas calmadas. Fuente: propia.

En estado de equilibrio, el barco está bajo la acción de dos fuerzas principales: la fuerza gravitacional y la fuerza de empuje. Ambas actúan en puntos distintos; la fuerza gravitacional actúa sobre el centro de gravedad, llamado  $G$ , mientras que la fuerza de empuje actúa sobre el centro de flotación. El centro de flotación se denomina  $B$  y se define como el centroide del volumen de agua desplazado por el barco. Mientras el barco esté en estado de equilibrio, tanto el punto  $G$  como el punto  $B$  estarán dentro de una línea vertical paralela al eje  $Z$  del marco  $BODY$ . La línea formada por la unión del punto  $B$  y  $G$  se denomina línea central y se denota como  $\overline{BG}$ . La Figura 2.10 ilustra el barco en equilibrio y las fuerzas que actúan sobre él.



**Figura 2.10.** Fuerzas que actúan sobre el barco en estado de equilibrio. Vista frontal.

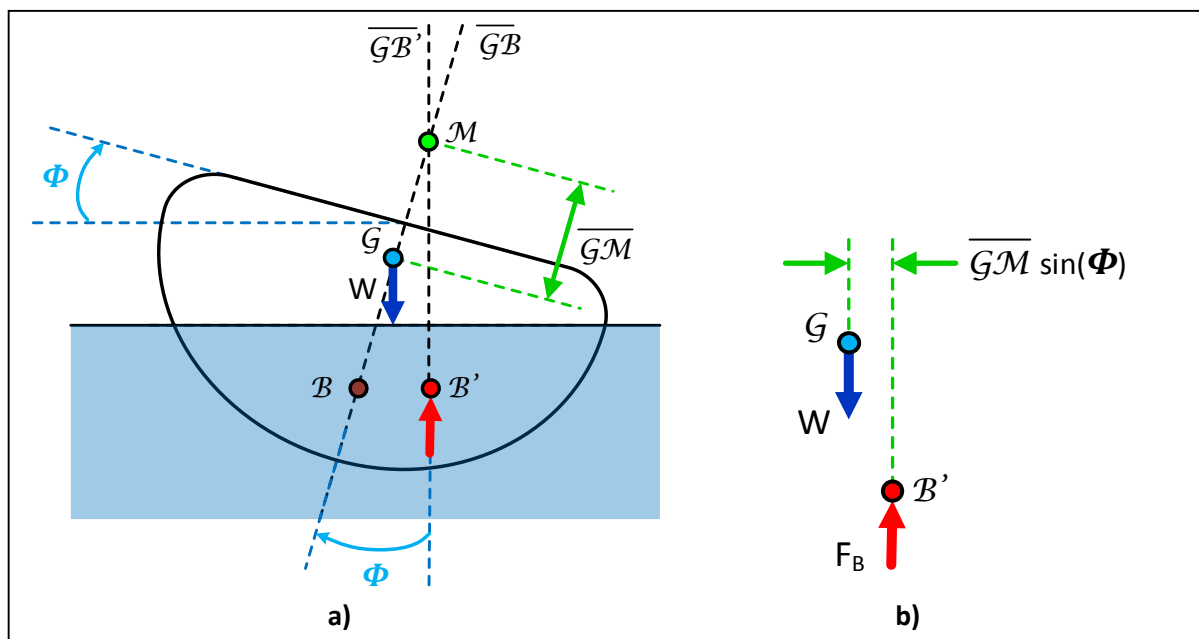
Fuente: propia.

Se cumple que:

$$W = F_B \quad \Rightarrow \quad mg = \rho g \nabla \quad (2.59)$$

Si el barco abandona su estado de equilibrio, cambia el volumen de agua desplazado y, por lo tanto, cambia la ubicación del centroide de dicho volumen; es decir, el centro de flotación. Ahora que la fuerza de empuje actúa en otro punto, se crea un nuevo punto de referencia

llamado el metacentro, y con este se define la altura metacéntrica. Gracias a este segmento se puede definir el brazo del torque generado por las fuerzas  $W$  y  $F_B$ , dado que ya no están alineadas. Al nuevo centro de flotación se le denomina  $B'$ . El metacentro se define como la intersección de las líneas  $\overline{GB}$  y  $\overline{G'B'}$ . La distancia desde el centro de gravedad  $G$  hasta el metacentro es llamada altura metacéntrica, y se representa por  $\overline{GM}$ . La Figura 2.11 muestra la situación del barco fuera de su posición de equilibrio. Todo este análisis, así como el análisis posterior, se puede aplicar tanto al plano transversal (X-Y) como al plano longitudinal (X-Z).



**Figura 2.11.** Barco fuera del estado de equilibrio, vista frontal. a) metacentro y altura metacéntrica debido a desestabilización. b) Brazo del torque causado por  $W$  y  $F_B$ . Fuente: propia.

El torque generado por  $W$  y  $F_B$  moverá el barco de tal forma que este tenderá a regresar hacia su posición de equilibrio. Por este motivo, las fuerzas analizadas en esta sección se denominan “restauradoras”, porque tienden a “restaurar” el estado de equilibrio en el que se inició el movimiento.

Dado que el barco ya no se encuentra en equilibrio, existirá una fuerza neta distinta de cero en el eje “heave”, a la que denominaremos  $Z$ . La Ecuación 2.60 muestra esta fuerza neta:

$$Z = mg - \rho g [\nabla + \delta \nabla_{(z)}] \quad (2.60)$$

El término  $\delta \nabla_{(z)}$  representa el cambio en el volumen de agua desplazada (igual a cero en caso de equilibrio) como una función del desplazamiento en  $Z$ . Este término puede calcularse de la forma mostrada en la Ecuación 2.61.

$$\delta \nabla_{(z)} = \int_0^z A_{wp}(\xi) d\xi \approx A_{wp(0)} z \quad (2.61)$$

En esta ecuación, se puede ver que el diferencial de volumen desplazado se calcula como la integral del área del plano de agua. Se asume que los cambios en Z son tan pequeños que el plano de agua permanece constante con un valor muy similar a su valor en equilibrio ( $A_{wp(0)}$ ). Reemplazando la Ecuación 2.61 en la Ecuación 2.60 se obtiene:

$$Z = mg - \rho g [\nabla + \delta \nabla_{(z)}] = mg - \rho g [\nabla + A_{wp(0)} z] = -\rho g A_{wp(0)} z \quad (2.62)$$

$$Z = -\rho g A_{wp(0)} z = Z_z z \quad (2.63)$$

Si se define una fuerza restauradora  $\delta f_r^b$  (el subíndice “r” indica “restaurador”) en el eje “heave” tal que:

$$\delta f_r^b = \begin{pmatrix} 0 \\ 0 \\ -\rho g \int_0^z A_{wp}(\xi) d\xi \end{pmatrix} \quad (2.64)$$

El término de fuerza de la Ecuación 2.64 puede ser expresado en el sistema NED de la siguiente manera:

$$\delta f_r^b = \mathbf{R}_b^n(\Theta_{nb})^{-1} \delta f_r^n \quad (2.65)$$

$$\delta f_r^b = -\rho g \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \sin(\phi) \\ \cos(\theta) \cos(\phi) \end{bmatrix} \int_0^z A_{wp}(\xi) d\xi \quad (2.66)$$

La Ecuación 2.66 representa el vector de fuerzas restauradoras del barco. Para determinar los torques restauradores se necesita tener un brazo a lo largo del cual sea ejercido el torque. Este brazo está dado en la Ecuación 2.64:

$$\mathbf{r}_r^b = \begin{pmatrix} -\overline{GM}_L \sin(\theta) \\ \overline{GM}_L \sin(\phi) \\ 0 \end{pmatrix} \quad (2.67)$$

La fuerza que actúa a lo largo del brazo de la ecuación anterior es la siguiente:

$$\mathbf{f}_r^b = \mathbf{R}_b^n(\Theta_{nb})^{-1} \begin{pmatrix} 0 \\ 0 \\ -\rho g \nabla \end{pmatrix} = \rho g \nabla \begin{pmatrix} -\sin(\theta) \\ \cos(\theta) \sin(\phi) \\ \cos(\theta) \cos(\phi) \end{pmatrix} \quad (2.68)$$

Ahora que se tiene tanto la fuerza como el brazo, se puede calcular el torque que será ejercido sobre el barco. Este torque es hallado en la Ecuación 2.69:

$$\mathbf{m}_r^b = \mathbf{r}_r^b \times \mathbf{f}_r^b = -\rho g \nabla \begin{pmatrix} GM_T \sin(\phi) \cos(\theta) \cos(\phi) \\ GM_L \sin(\theta) \cos(\theta) \cos(\phi) \\ (-GM_L \cos(\theta) + GM_T) \sin(\phi) \sin(\theta) \end{pmatrix} \quad (2.69)$$

Ahora que se cuenta con las fuerzas y torques restauradoras, se puede compactar las ecuaciones 2.65 y 2.69 en una sola ecuación matricial, mostrada a continuación:

$$\mathbf{g}(\boldsymbol{\eta}) = - \begin{bmatrix} \delta \mathbf{f}_r^b \\ \mathbf{m}_r^b \end{bmatrix} \quad (2.70)$$

El término  $\mathbf{g}_0$  está relacionado con las fuerzas y momentos generados por el sistema de balasto que normalmente se encuentra en barcos. En el presente trabajo, el barco robot a usar no posee un sistema de balasto, por lo tanto este término se anula.

### 2.3.3 Matriz de amortiguamiento

El amortiguamiento hidrodinámico se muestra en la Ecuación 2.71:

$$\mathbf{D}(\mathbf{v}_r) = \mathbf{D} + \mathbf{D}_n(\mathbf{v}_r) \quad (2.71)$$

Donde el término  $\mathbf{D}$  es la matriz de amortiguamiento lineal, la cual engloba los efectos del amortiguamiento potencial y posibles efectos de fricción laminar; mientras que la matriz  $\mathbf{D}_n(\mathbf{v}_r)$  es llamada matriz de amortiguamiento no-lineal y engloba los efectos de amortiguamiento cuadrático y términos de orden mayor. Los elementos de estas matrices se pueden determinar de manera experimental y de acuerdo a métodos ya establecidos.

## 2.4 MODELO DE 3 GRADOS DE LIBERTAD

A continuación, se muestra una simplificación del modelo de 6DOF hecha por Fahimi (2009). En dicho trabajo, el modelo de 6DOF es expresado de forma más compacta en las siguientes ecuaciones:

$$m_{11}\dot{u} - m_{22}vr + m_{33}wq + d_{11}u = W_u + F \quad (2.72)$$

$$m_{22}\dot{v} - m_{33}wp + m_{11}ur + d_{22}v = W_v \quad (2.73)$$

$$m_{33}\dot{w} - m_{11}uq + m_{22}vp + d_{33}w = W_w + mg + Z_w \quad (2.74)$$

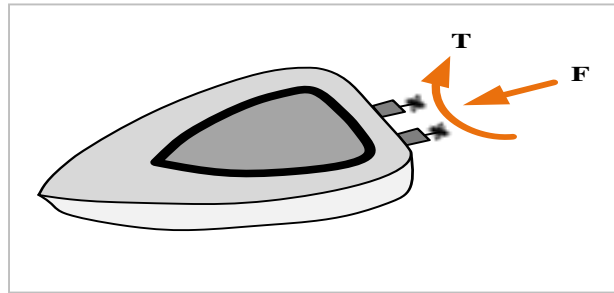
$$I_{xx}\dot{p} + (m_{33} - m_{22})wv + (I_{zz} - I_{yy})rq + d_{44}p = K_p \quad (2.75)$$

$$I_{yy}\dot{q} + (m_{11} - m_{33})uw + (I_{xx} - I_{zz})pr + d_{55}q = FFG + M_q \quad (2.76)$$

$$I_{zz}\dot{r} + (m_{22} - m_{11})vu + (I_{zz} - I_{yy})qp + d_{66}r = T \quad (2.77)$$

Donde  $m_{ii}$  denota la masa añadida del barco,  $d_{ii}$  denota los coeficientes lineales de amortiguamiento dinámico,  $I_{ii}$  representa el momento de inercia, y  $W_i$  son los componentes

de la fuerza de las olas en términos del marco de referencia BODY. El término  $FFG$  es la fuerza del torque que actúa en el eje transversal (“*pitch*”) del barco que pasa por el centro de gravedad del mismo. Las variables  $Z_w$ ,  $K_p$  y  $M_q$  son la fuerza de empuje, y los torques de restauración en “*roll*” y “*pitch*” respectivamente. Las variables  $F$  y  $T$  representan la fuerza y el torque generadas por los propulsores posteriores del barco, estos se pueden apreciar en la Figura 2.12.



**Figura 2.12.** Fuerza y torque generados por los propulsores posteriores del barco. Fuente: propia.

Para simplificar el modelo de 6DOF se tomará en cuenta las siguientes suposiciones:

- La fuerza de empuje estabiliza el movimiento en “*heave*”.
- El barco está diseñado para mantener una adecuada altura metacéntrica longitudinal y transversal. Por lo tanto, los movimientos de “*roll*” y “*pitch*” están estabilizados.

Tomando en cuenta estas dos suposiciones, se puede llegar a un modelo de 3DOF, en el cual solo se considere los movimientos en “*surge*”, “*sway*” y “*yaw*” en el marco de referencia BODY. Dicho modelo es mostrado en las siguientes ecuaciones:

$$m_{11}\dot{u} - m_{22}vr + d_{11}u = W_u + F \quad (2.78)$$

$$m_{22}\dot{v} + m_{11}ur + d_{22}v = W_v \quad (2.79)$$

$$I_{zz}\dot{r} + (m_{22} - m_{11})uv + d_{66}r = T \quad (2.80)$$

En este modelo simplificado, la Ecuación 2.79 nos muestra que el sistema es sub-actuado, ya que en esta ecuación no hay influencia de las variables de control  $F$  y  $T$ . Para adecuar el modelo de 3DOF al marco de referencia NED, se aplican las relaciones cinemáticas entre las velocidades  $\{u, v, r\}$  y  $\{\dot{x}, \dot{y}, \dot{\psi}\}$ . El resultado se muestra en las siguientes ecuaciones:

$$\ddot{x} = \frac{1}{m_{11}}(f_x + F \cos \psi) \quad (2.81)$$

$$\ddot{y} = \frac{1}{m_{22}}(f_y + F \sin \psi) \quad (2.82)$$

$$\ddot{\psi} = \frac{1}{I_{zz}}(f_{\psi} + T) \quad (2.83)$$

Donde:

$$f_x = m_r d_{22} v \sin \psi - d_{11} u \cos \psi + \dot{\psi}(v \cos \psi - m_r u \sin \psi) m_d \quad (2.84)$$

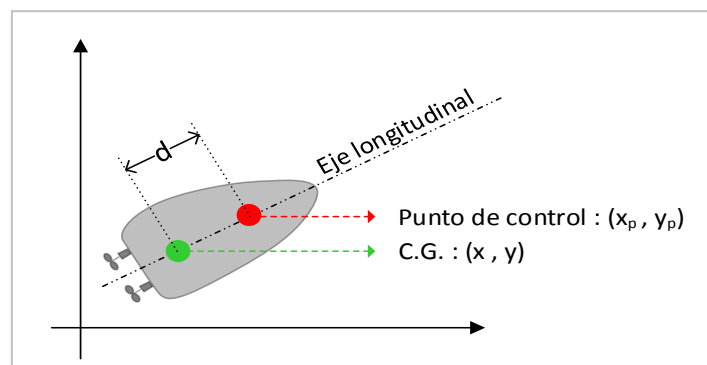
$$f_y = -m_r d_{22} v \cos \psi - d_{11} u \sin \psi + \dot{\psi}(v \sin \psi + m_r u \cos \psi) m_d \quad (2.85)$$

$$f_{\psi} = -m_d u v - d_{33} \dot{\psi} \quad (2.86)$$

$$u = \dot{x} \cos \psi + \dot{y} \sin \psi \quad v = -\dot{x} \sin \psi + \dot{y} \cos \psi \quad (2.87)$$

$$m_d = m_{22} - m_{11} \quad m_r = \frac{m_{11}}{m_{22}} \quad (2.88)$$

Dado que la Ecuación 2.79 no recibe influencia de las variables de control  $F$  y  $T$ , se tiene un sistema sub-actuado. Bajo esta observación, se podría decir que no se puede controlar un sistema de tres grados de libertad con solo dos variables de control; sin embargo, la experiencia nos muestra que a lo largo de los años así es como se ha llevado a cabo el control de barcos. Por tal motivo, deducimos que el sistema es inherentemente estable a pesar de ser sub-actuado. Gracias a esta observación, podemos diseñar un controlador cuya salida sean solo dos de las tres variables de estado del modelo propuesto. La elección de estas dos variables se realiza teniendo en cuenta que no es prudente dejar al sistema enteramente insensible ante una variable de estado. Ya que el modelo de 3DOF tiene dos variables de posición  $\{x, y\}$  y una variable de orientación  $\psi$  ("yaw"), la elección más sencilla sería elegir las coordenadas del centro de gravedad del barco. Sin embargo, esta opción nos conduciría a dejar el sistema insensible ante la variable de orientación  $\psi$ , ya que las coordenadas del centro de gravedad no nos proporcionan ninguna información respecto a la orientación del barco. Para evitar esto se elige un punto de control distinto al centro de gravedad, pero que pertenezca al eje longitudinal del barco, como se ilustra en la Figura 2.13.



**Figura 2.13.** Punto de control en el eje longitudinal del barco. Fuente: propia.

Las coordenadas de este punto están dadas por:

$$x_{p1} = x_1 + d \cos \psi \quad (2.89)$$

$$x_{p2} = x_2 + d \sin \psi \quad (2.90)$$

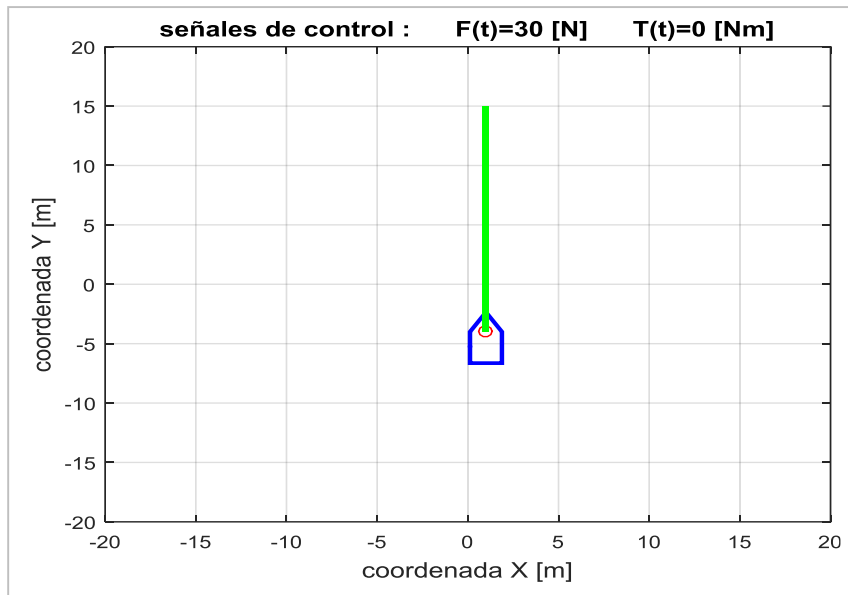
De esta forma se puede manejar el sistema con un controlador de dos salidas, pero al mismo tiempo se puede tomar en cuenta las perturbaciones en las tres variables de estado del modelo de 3DOF.

## 2.5 SIMULACIONES DEL MODELO

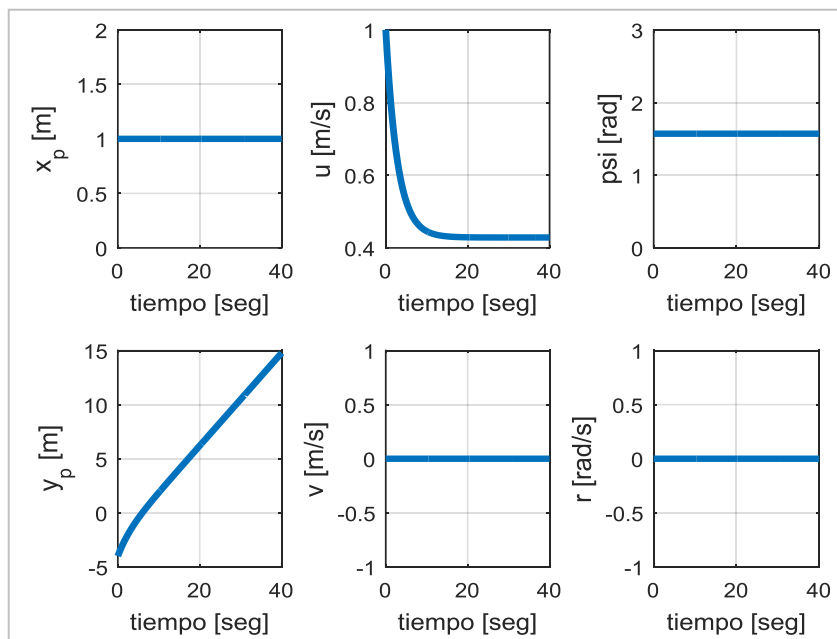
Para comprobar la validez del modelo propuesto por Fahimi (2009) se verificó si al manipular las entradas manualmente se obtienen resultados físicamente coherentes. Las entradas manuales fueron: fuerza constante y torque nulo, fuerza nula y torque constante, y fuerza constante y torque constante. Se asumió que las perturbaciones del entorno,  $W_u$  y  $W_v$ , son nulas; más adelante se considerarán los efectos que estas perturbaciones tienen sobre el sistema de control.

Si sobre el barco solo actúa una fuerza constante y el torque se mantiene igual a cero, entonces el barco debe moverse a lo largo de una línea recta; para este caso el ángulo de orientación no cambia. Ya que la matriz del modelo de 6DOF original consideraba los efectos de masa añadida de agua, la fuerza que se aplica al barco se verá anulada por la fuerza que ejerce el mar sobre este; por lo tanto, la fuerza constante aplicada se anulará rápidamente con la fuerza del mar y se tendrá una velocidad constante en la dirección del movimiento. Esta primera prueba se muestra en la Figura 2.14; allí se grafica la posición del barco robot en distintos instantes de tiempo a lo largo de la simulación. La línea verde en la Figura 2.14 representa la trayectoria descrita por el punto de control, mientras que el punto rojo representa el punto de control tomado en el barco robot. Se puede comprobar en ambas Figuras que los resultados son físicamente coherentes.

Las variables de posición del punto de control y las variables de estado se muestran en la Figura 2.15; en ella se puede observar que el ángulo de orientación no cambia, al igual que la velocidad de deriva (*sway*). Cabe resaltar que la simulación se llevó a cabo durante 40 segundos.



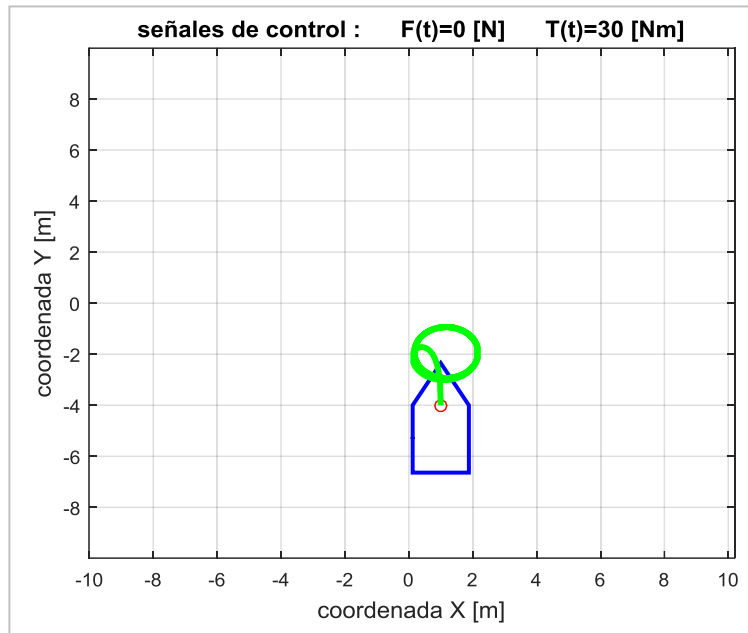
**Figura 2.14.** Pruebas del modelo propuesto por Fahimi. Caso de fuerza constante y torque nulo. Trayectoria descrita. Fuente: propia.



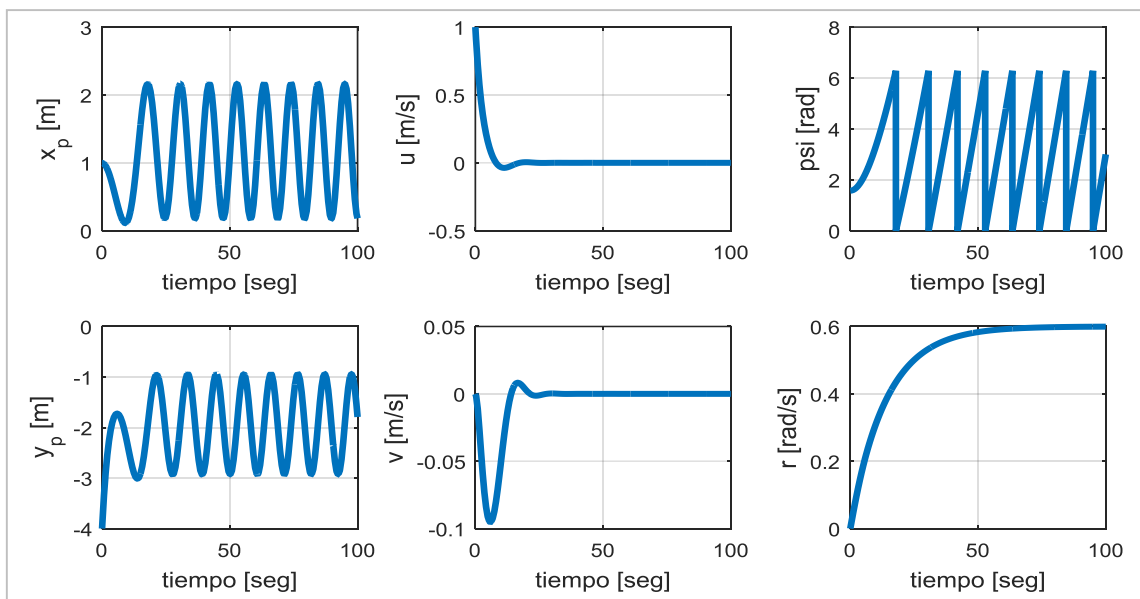
**Figura 2.15.** Pruebas del modelo propuesto por Fahimi. Caso de fuerza constante y torque nulo. Variables de estado y coordenadas del punto de control. Fuente: propia.

En la siguiente prueba realizada se estableció un torque constante y una fuerza nula, las condiciones de simulación fueron las mismas, excepto por el tiempo de simulación, el cual fue ampliado hasta 100 segundos. La trayectoria descrita se muestra en la Figura 2.16 y la evolución independiente de cada variable de estado y de las coordenadas del punto de control se muestra en la Figura 2.17. Físicamente, un torque constante y una fuerza nula corresponden al caso en que solo uno de los propulsores del barco robot está girando mientras el otro está detenido, por lo cual solo existe torque y no fuerza. En esta prueba, ya

que no existe una fuerza que empuje al barco robot hacia adelante, la trayectoria descrita debe ser una curva que converja hacia un círculo. El círculo descrito debe ser de un radio pequeño, ya que el funcionamiento de un solo motor del barco lo hará girar intempestivamente. Se puede observar en las Figuras 2.16 y 2.17 que se cumple el comportamiento esperado del sistema.



**Figura 2.16.** Pruebas del modelo propuesto por Fahimi. Caso de fuerza nula y torque constante. Trayectoria descrita. Fuente: propia.

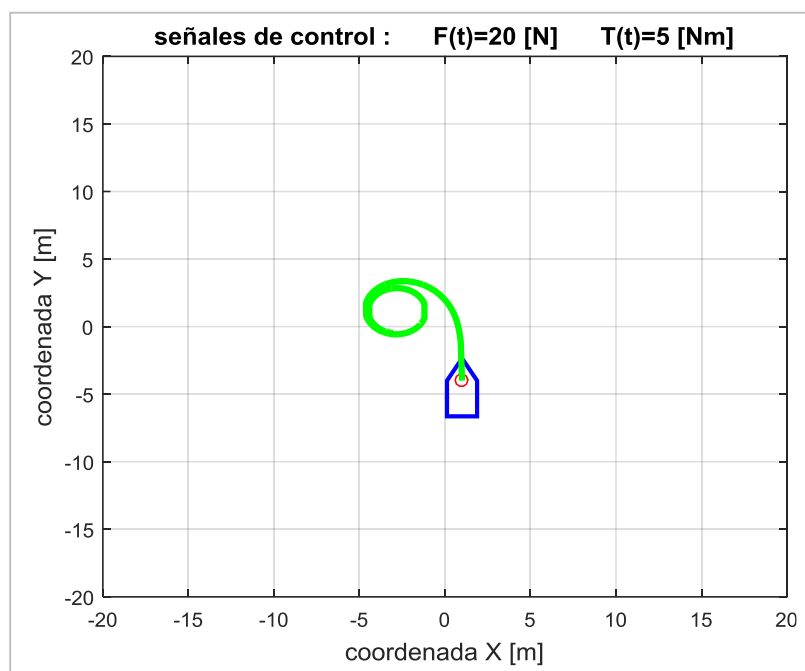


**Figura 2.17.** Pruebas del modelo propuesto por Fahimi. Caso de fuerza nula y torque constante. Variables de estado y coordenadas del punto de control. Fuente: propia.

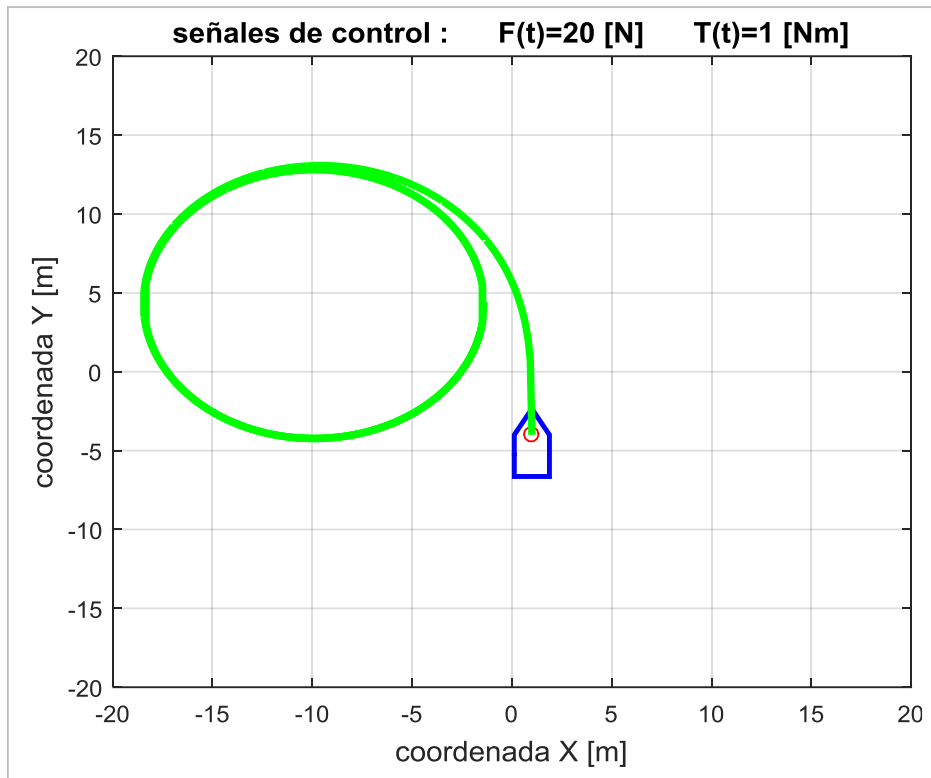
Un punto importante a notar es que bajo la acción de un torque constante, la velocidad de cambio en la orientación (“*yaw rate*”) crece exponencialmente hasta converger a un valor constante. Mientras que las velocidades lineales ( $u$  y  $v$ ) convergen a cero una vez que la trayectoria del barco robot ha convergido a un círculo. Este giro se comprobó con pruebas experimentales del barco en una piscina, el barco robot gira sobre un eje que pasa por el punto de control y, por esto, sus velocidades lineales convergen a cero.

La última prueba consiste en ingresar una fuerza constante y un torque constante al barco robot. Esta vez el tiempo de simulación será de 600 segundos. Ya que ahora se cuenta con fuerza y torque, se espera que el barco robot describa una trayectoria circular, cuyo radio y ubicación dependerán de los valores de fuerza y torque aplicados. Mientras más alto sea el valor del torque aplicado, se espera obtener un círculo de menor radio, ya que el torque obliga al barco a girar más violentamente. Mientras más alto sea el valor de la fuerza, más lejos de su posición inicial estará centrado el círculo que describa el barco. La Figura 2.18 muestra una trayectoria descrita para un primer caso de fuerza y torque constantes.

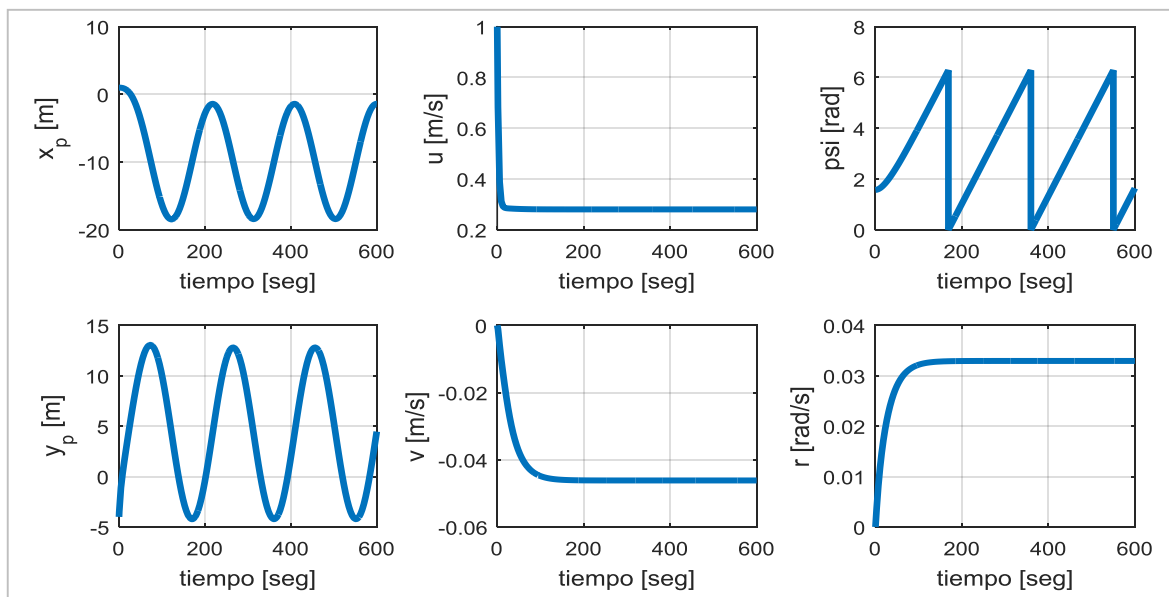
Si ahora se mantienen las condiciones de simulación y solo disminuye el valor del torque, se obtiene la trayectoria mostrada en la Figura 2.19. Para esta última trayectoria mostrada se graficaron las variables de estado y posición del punto de control a lo largo del tiempo de simulación, los resultados se muestran en la Figura 2.20. Se puede observar que ahora las velocidades lineales convergen a un valor distinto a cero.



**Figura 2.18.** Pruebas del modelo propuesto por Fahimi. Caso de fuerza constante y torque constante. Trayectoria descrita, círculo pequeño. Fuente: propia.



**Figura 2.19.** Pruebas del modelo propuesto por Fahimi. Caso de fuerza constante y torque constante. Trayectoria descrita, círculo grande debido a un torque menor. Fuente: propia.



**Figura 2.20.** Pruebas del modelo propuesto por Fahimi. Caso de fuerza constante y torque constante. Variables de estado y coordenadas del punto de control. Fuente: propia.

Con las pruebas mostradas se puede concluir que el modelo planteado por F. Fahimi, el cual se basa en el modelo de T. Fossen, genera resultados físicamente coherentes.

# SISTEMA DE PLANEAMIENTO DE TRAYECTORIA

En este capítulo, se describirá la implementación de la Optimización por Colonia de Hormigas (ACO por sus siglas en inglés) realizada en el entorno de desarrollo MatLab. Las ideas principales bajo las cuales se implementó este algoritmo están descritas en M. Dorigo (2004). También se utilizaron ideas de S. Johnson (2001) y Deborah Gordon (2013). A continuación, se realiza una explicación detallada de estas ideas antes de pasar a la descripción de la implementación.

### 3.1 PRESENTACIÓN DEL PROBLEMA

En el capítulo 1 se planteó el problema a resolver en el presente trabajo: determinar la trayectoria más corta entre dos puntos para que un robot marítimo de superficie recorra en un mapa con obstáculos y fronteras. Este problema puede plantearse de manera más abstracta de la siguiente forma: en cada instante de tiempo un individuo dentro de un grupo debe elegir un camino a seguir dentro de una serie de alternativas, la elección de este camino está influenciada por una variable que es función de la calidad de solución que se genera al tomar dicho camino. Cada individuo busca la combinación de caminos correcta, de tal forma que se logre encontrar el camino más corto entre el punto de partida y el punto destino. Este problema cae dentro de la categoría de problemas de optimización combinatoria discreta, lo cuales pueden ser resueltos por métodos heurísticos.

### 3.2 ALGORITMO DE OPTIMIZACIÓN POR COLONIA DE HORMIGAS

El algoritmo meta-heurístico ACO fue propuesto por Margo Dorigo en su tesis doctoral en 1992. Desde ese entonces, este algoritmo ha sido aplicado exitosamente a varios problemas de optimización combinatoria. La idea principal de este algoritmo es modelar el problema planteado como una búsqueda del camino de menor costo dentro de un gráfico. La particularidad de este algoritmo es que las soluciones que se van hallando provienen de agentes individuales llamados hormigas, los cuales transitan el gráfico planteado siguiendo un comportamiento bastante sencillo. Las soluciones (caminos) halladas por cada hormiga son al inicio aleatorios y, por lo general, de baja calidad (alto o mediano costo), conforme pasa el tiempo la mejor solución emerge de la interacción entre todos los individuos de la colonia y el entorno en el que se desplazan [39]. Es importante recalcar que la optimización por colonia de hormigas no es un solo algoritmo, sino un marco de conceptos algorítmicos

que puede ser usado para desarrollar algoritmos heurísticos específicos a la solución de un problema de optimización combinatoria en particular; es decir, es una meta-heurística [15].

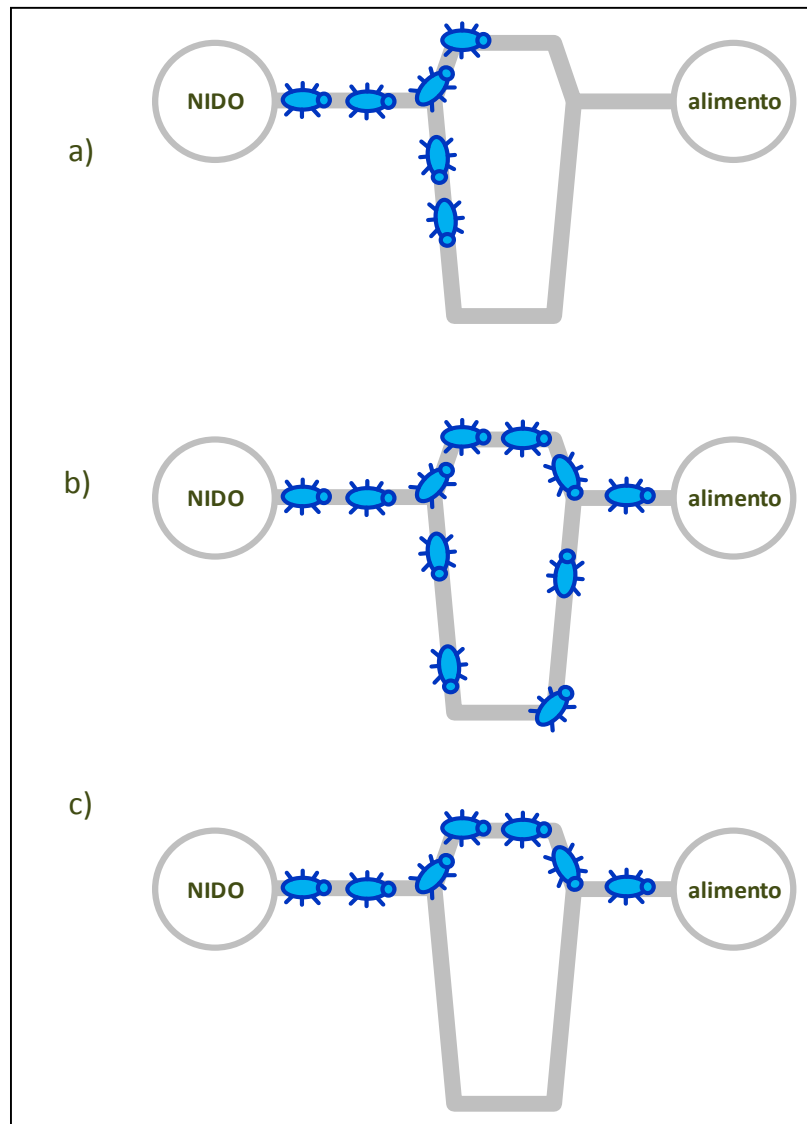
Dentro de los numerosos estudios aplicativos que se han realizado utilizando ACO se mencionan, a continuación, los más resaltantes. En [11], [12], [13] y [14] se mostró anteriormente la utilización de algoritmos de planeamiento de trayectoria para robots móviles utilizando ACO. El problema de planeamiento de trayectoria es el más obvio de resolver utilizando ACO, dado que fue este problema, resuelto por las colonias de hormigas, el que inspiró el ACO en primer lugar. Sin embargo, la aplicación del ACO en diversos problemas, aparte del planeamiento de trayectoria, ha sido abundante en las dos últimas décadas. En [57] y [61], se utilizó ACO para proponer un algoritmo de ruteo para paquetes de datos en una red de computadoras; particularmente en [57] las redes estudiadas eran redes móviles ad-hoc, y se lograron mejores resultados que con un algoritmo convencional. En [58] se utilizó ACO para proponer un algoritmo que solucione el problema de asignación de expertos; este problema nace de la cantidad escasa de expertos en China ante la cantidad inmensa de proyectos en las cuales se necesita de su participación. Dicho problema es un ejemplo de problema de optimización combinatoria, y los resultados mostrados en [58] mostraron que ACO produjo mejores resultados que un algoritmo genético estándar en cuanto al tiempo de convergencia. En [59] y [60], se utilizó ACO para la solución del problema de asignación de tareas en un taller; este problema pertenece a la categoría NP-hard de problemas de optimización combinatoria. El problema consiste en asignar una determinada cantidad de tareas a una determinada cantidad de máquinas que pueden realizarlas dentro de un proceso industrial. En [62] se presenta un algoritmo de asignación de tareas para dos robots industriales funcionalmente idénticos que buscan llevar a cabo un trabajo de pintura industrial; los resultados muestran que se logró efectuar una división del trabajo eficiente.

### **3.2.1 Hormigas reales e inteligencia emergente**

Las colonias de hormigas son un sistema biológico que puede ser descrito como un sistema emergente. La característica principal de estos sistemas es la solución de problemas complejos mediante la abundante interacción de individuos sencillos, que por sí mismos no podrían resolver el problema planteado. Existen otros sistemas naturales dentro de esta categoría, como las neuronas en un cerebro, ciudades dentro de un país, abejas buscando refugio, entre otros.

Uno de los experimentos que mejor ilustra la capacidad como sistema emergente de una colonia de hormigas es el doble-puente. En este experimento, una colonia de hormigas cuyo nido está en un punto A debe explorar el territorio circundante, encontrar alimento y traerlo de regreso al nido. El camino entre el nido y el alimento presenta una bifurcación, en la cual

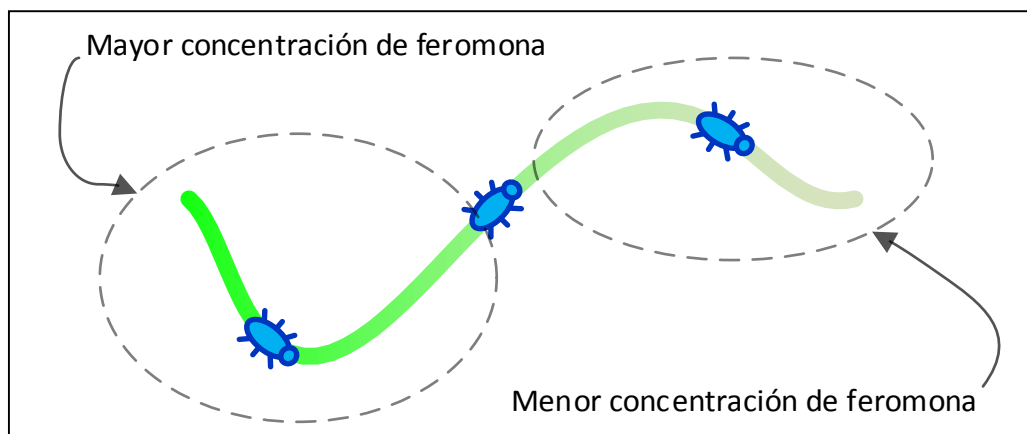
un camino es más largo que el otro. En esta situación, se puede observar que después de un tiempo la colonia de hormigas tiende a usar el camino más corto; es decir, el sistema ha alcanzado convergencia hacia la mejor solución. El desarrollo de este experimento se ilustra en la Figura 3.1.



**Figura 3.1.** Experimento doble-puente. (a) inicio del experimento. (b) fase aleatoria inicial. (c) convergencia del sistema hacia el camino de menor longitud. Fuente: propia.

Dentro de la colonia de hormigas, cada individuo (hormiga) no posee información respecto a la longitud de los posibles caminos a tomar desde el nido hacia el alimento. Asimismo, la comunicación entre hormigas es limitada, su sentido de la vista es bastante precario y no poseen la capacidad cerebral para almacenar en su memoria todos los trayectos que realizan. Por lo tanto, para lograr la convergencia de la colonia hacia el camino más corto, se hace uso de un mecanismo de comunicación denominado estigmergía. Este mecanismo consiste en la alteración del medio en el cual se desplazan los individuos de un sistema colectivo de tal

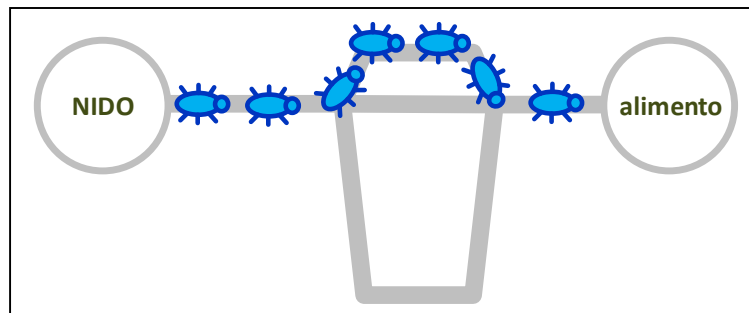
forma que dichas alteraciones puedan ser encontradas por otros individuos del sistema, y en base a esto tomar decisiones. En el caso particular de la colonia de hormigas, la estigmergía se lleva a cabo cuando estas depositan en el terreno una sustancia química llamada feromona. Cuando una hormiga encuentra el alimento, emprende el camino de retorno hacia el nido, la hormiga es capaz de regresar al nido gracias al rastro de feromona que dejó en el camino mientras buscaba la comida. La feromona tiene la propiedad de ser aditiva; es decir, conforme más hormigas depositan feromona en un mismo punto del terreno, más intensa es su influencia sobre las hormigas que encuentren este rastro; esto se ilustra en la Figura 3.2. En esta Figura se muestra tres hormigas que transitan un mismo camino depositando feromona, con cada hormiga que pasa por el camino aumenta la concentración de feromona en este (representado por el color verde de la línea).



**Figura 3.2.** Propiedad aditiva de la feromona. Fuente: propia.

Además de usar la feromona para regresar al nido, mientras la hormiga regresa al nido deposita una feromona distinta que, al ser encontrada por otras hormigas, indica la presencia de alimento si se sigue dicho camino. En el experimento del doble-puente, las hormigas que salen del nido al inicio eligen aleatoriamente entre el camino corto y el largo. Por lo tanto, se tendrá hormigas que dejan un rastro de feromona en ambos caminos indicando que ambos conducen hacia el alimento. Sin embargo, conforme pasa el tiempo, la cantidad de hormigas que transitan por el camino corto es superior a la cantidad de hormigas que van por el camino largo, por lo tanto el rastro de feromona en el camino corto se intensifica con mayor rapidez que el rastro en el camino largo. Este efecto de realimentación positiva conlleva a que, eventualmente, la gran mayoría de las hormigas en la colonia elijan el camino más corto, por el hecho de poseer una concentración de feromona muy superior a la del camino largo. Es así como el sistema resuelve el problema de hallar el camino más corto desde el nido hasta el alimento en una situación de doble-puente.

Un punto importante a tomar en cuenta en la solución encontrada por la colonia de hormigas es que el procedimiento no es adaptable ante nuevas situaciones. Esto significa que si en el experimento de doble-puente se añade un tercer camino aún más corto que el camino ya escogido por la colonia, este no será elegido; esta variante se ilustra en la Figura 3.3. Una vez que la colonia ha convergido hacia un camino, no elegirá otro aunque este sea más corto mientras exista el camino al que ya convergió.



**Figura 3.3.** Baja adaptabilidad de la colonia ante una nueva y mejor solución. Fuente: propia.

La baja adaptabilidad de la colonia de hormigas, se debe a que la feromona depositada en el terreno no se evapora con la suficiente rapidez como para descartar soluciones sub-óptimas que puedan haber sido elegidas al inicio del experimento. Este rasgo puede mejorarse cuando se utilizan hormigas virtuales dentro de un programa de computadora, de tal forma que puedan crearse algoritmos basados en este comportamiento, pero a la vez mejorando sus deficiencias. La siguiente sección aborda el tema de la creación y manejo de hormigas virtuales.

### 3.2.2 Hormigas artificiales

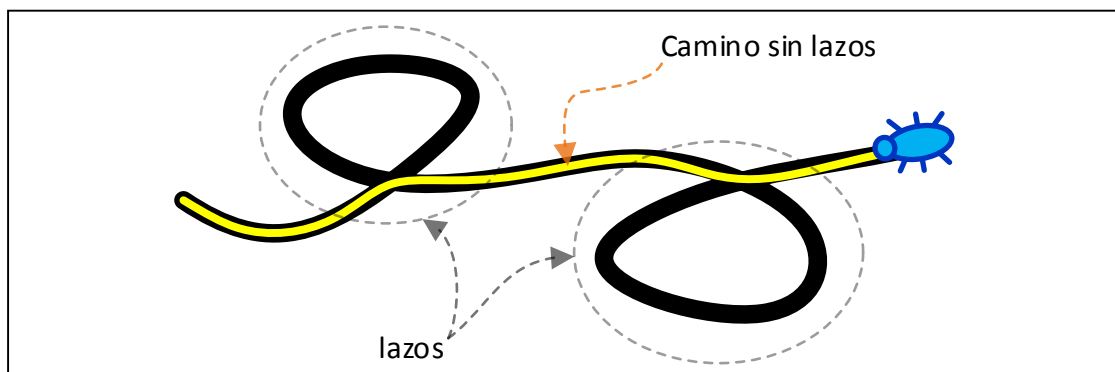
Tomando la colonia de hormigas como inspiración, se puede diseñar un programa de computadora que imite, hasta cierto punto, la solución encontrada por las hormigas para determinar el camino más corto entre dos puntos. Además de imitar al sistema biológico, se pueden introducir variantes propias al problema que se busca resolver con las hormigas virtuales. Una de las variantes principales con su contraparte biológica es la diferencia entre la marcha en busca del alimento y el retorno al nido. En el caso del sistema biológico, ambas marchas involucran el uso y depósito de feromonas distintas en cada caso. Por otro lado, las hormigas virtuales solo depositan feromona cuando marchan de regreso al nido.

Para las hormigas virtuales la marcha hacia el alimento es un proceso pseudo-aleatorio, mientras que el retorno al nido es un proceso determinístico. El proceso pseudo-aleatorio de buscar alimento consiste en explorar el territorio eligiendo el siguiente paso a realizar utilizando una probabilidad aleatoria inicial que es modificada por una serie de factores, entre

los cuales destaca la concentración de feromona presente en cada camino a elegir. Si no existe ningún rastro de feromona alrededor de la hormiga, cualquiera de los caminos a tomar podría ser elegido con la misma probabilidad. Cuando un camino ante la hormiga presenta un rastro de feromona, la probabilidad de ser elegido aumenta. De esta forma se asegura que las hormigas virtuales exploren el territorio a la vez que realizan una realimentación positiva de los caminos que ya contienen feromona. Cuando las hormigas encuentran el alimento, inician el retorno hacia el nido. Este retorno se lleva a cabo siguiendo los pasos que llevaron a la hormiga hasta el alimento, es decir, la hormiga regresa sobre sus propios pasos. Por esto se dice que el retorno es un proceso determinístico, ya que cada hormiga sabe con exactitud qué pasos dar para regresar al nido a diferencia de la búsqueda pseudo-aleatoria del alimento.

El proceso de marchar hacia el alimento y retornar al nido se lleva a cabo gracias a un conjunto de características implementadas en cada hormiga. A continuación se describen las características de las hormigas virtuales usadas para el presente trabajo:

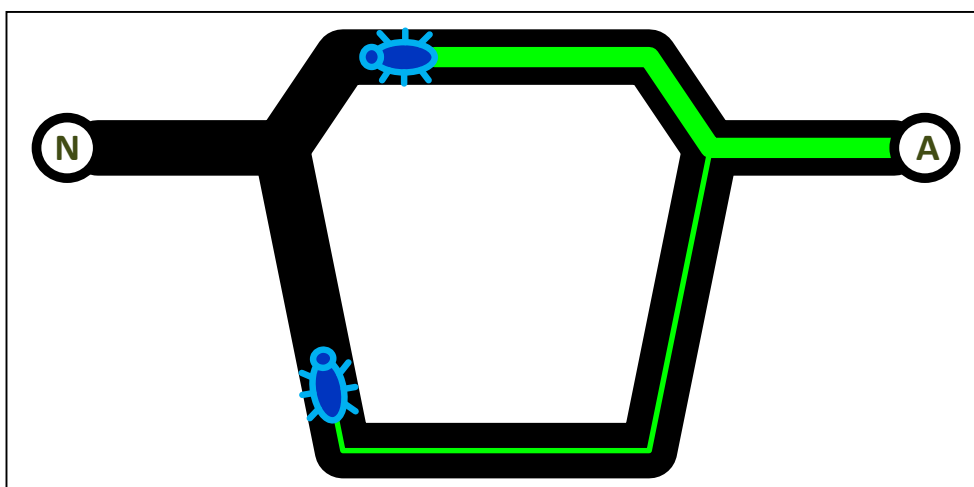
- Memoria: a diferencia de su contraparte biológica, las hormigas virtuales cuentan con una memoria donde almacenan todo el camino que han seguido desde que salen del nido hasta que encuentran el alimento. De esta forma son capaces de regresar al nido retrocediendo en sus pasos.
- Eliminación de lazos: debido a que las hormigas buscan el alimento de manera pseudo-aleatoria, durante la exploración generarán trayectorias que se cruzan a sí mismas, lo cual crea lazos. Si la hormiga siguiera sus pasos tal y como los describió desde el nido hasta el alimento, demoraría la misma cantidad de tiempo en retornar desde el alimento hasta el nido. Ya que el objetivo es solo regresar al nido, se eliminarán aquellos lazos en la trayectoria de la hormiga, de tal forma que se reduzca el tiempo que le toma a la hormiga retornar al nido. Esta característica se ilustra en la Figura 3.4, donde la línea amarilla representa el camino sin lazos, el cual es más corto.



**Figura 3.4.** Eliminación de lazos en el camino de retorno al nido. Fuente: propia.

- Brújula: debido a que el presente trabajo está orientado a un problema de planeamiento de trayectoria para un robot móvil, el punto inicial y final de la trayectoria a seguir son datos conocidos. Por lo tanto, cada hormiga de la colonia es capaz de saber la dirección que debe seguir en todo momento.
- Constancia de dirección: este factor le permite a la hormiga ajustar la importancia que le da a seguir determinado camino en base al paso anterior. Cuando el valor de este parámetro es alto, la hormiga tiende a elegir los pasos que se encuentran en la dirección del paso anterior que tomó. Por otro lado, cuando este valor es bajo, la hormiga tiende a elegir más arbitrariamente qué paso siguiente dar.

El proceso de ir desde el nido hasta el alimento para una sola hormiga representa la generación de una solución al problema de hallar el camino más corto entre nido y alimento. El proceso global de la optimización por colonia de hormigas es la generación de distintas soluciones al problema planteado. Para el caso particular de hallar el camino de menor longitud dentro de un mapa, la calidad de cada solución es una función inversamente proporcional a la cantidad de pasos que ha dado la hormiga desde que salió del nido hasta que llegó al alimento. Bajo esta lógica, mientras más largo el camino elegido, menor será la calidad de la solución. Al momento de regresar hacia el nido, las hormigas depositan feromona en una magnitud proporcional a la calidad de la solución. Esto se traduce en una concentración mayor de feromona en los caminos cortos y una concentración menor en los caminos largos. El resultado es una comunicación basada en estigmergia, ya que las demás hormigas de la colonia encontrarán rastros de feromona más intensos en los caminos cortos y, por ende, aumentarán las probabilidades de elegir dichos caminos. Esto se ilustra en la Figura 3.5.



**Figura 3.5.** Intensidad de feromona depositada proporcional a la calidad de la solución. El punto “N” denota la ubicación del nido, mientras que el punto “A” denota el alimento. Fuente: propia.

La construcción pseudo-aleatoria de una trayectoria (solución) por cada hormiga, se lleva a cabo alterando la probabilidad de elegir uno u otro paso dentro de la vecindad de la hormiga. Para esto se hace uso de un factor heurístico conocido a priori, y una lectura de la concentración de feromona en cada camino disponible en la vecindad. Para el presente trabajo se usó un factor heurístico definido por una función del valor de la brújula y un número aleatorio. Adicionalmente, cada hormiga es capaz de saber si en su vecindad se encuentran partes del borde del mapa, obstáculos, o el alimento. En caso de que la vecindad contenga obstáculos o bordes del mapa, se les asigna una probabilidad de ser elegidas igual a cero. En caso de que el alimento se encuentre en la vecindad de la hormiga, el paso que conduce hacia este será el que tome la hormiga, por lo tanto se le asignará la máxima probabilidad de ser elegido. La Ecuación 3.1 muestra la probabilidad de una hormiga  $k$  de elegir el camino  $j$  cuando su posición actual es  $i$ .

$$P_{ij}^k = \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{l \in \mathcal{N}_i^k} (\tau_{il})^\alpha (\eta_{il})^\beta} \quad \text{if } j \in \mathcal{N}_i^k \quad (3.1)$$

Donde  $(\tau_{ij})^\alpha$  es la concentración de feromona entre el punto  $i$  y el punto  $j$ , afectado por el peso  $\alpha$ . La variable  $(\eta_{ij})^\beta$  representa la información heurística entre el punto  $i$  y el punto  $j$ , la cual es una función de la brújula y una variable aleatoria, afectada por el peso  $\beta$ . La expresión  $\text{if } j \in \mathcal{N}_i^k$  refleja la habilidad de la hormiga virtual para saber si en su vecindad hay obstáculos o bordes del mapa.

Asimismo, el cambio de feromona que produce cada hormiga está dado por la Ecuación 3.2:

$$\Delta phm = \frac{phmG}{(N2F - N2Fe)^{phmE}} \quad (3.2)$$

Donde  $phmG$  y  $phmE$  son parámetros seleccionables,  $N2F$  es la cantidad de pasos dados por la hormiga desde que salió del nido hasta que encuentre el alimento, y  $N2Fe$  es la cantidad de pasos que en simulaciones previas se encontró como valor mínimo.

Adicionalmente a las características de cada hormiga virtual, se tienen sub-programas que manipulan el entorno donde se desplaza la colonia. A estos sub-programas se les llama daemons, y un daemon muy importante es el que controla la evaporación de la feromona en el terreno. A diferencia de la versión biológica, las hormigas virtuales pueden descartar soluciones sub-óptimas elegidas al inicio de la simulación. Esto se debe a que la rapidez con la que se evapora la feromona es un parámetro controlable dentro del programa. El valor asignado a este parámetro es crucial en el desempeño del algoritmo. Si la evaporación es muy rápida, no se llegará a converger hacia ninguna solución; por otro lado, si la evaporación es muy lenta, las soluciones sub-óptimas que puedan encontrarse al inicio de la simulación

podrían permanecer a lo largo del tiempo como las elegidas, a pesar de que se encuentren soluciones óptimas más adelante.

### **3.2.3 Aplicación en optimización combinatoria discreta**

El problema de encontrar el camino más corto dentro de un mapa puede interpretarse de otra manera. En cada instante de tiempo la hormiga tiene varias opciones entre las cuales elegir la siguiente posición. A cada una de estas opciones le corresponde una probabilidad de ser elegida y esta probabilidad se ve afectada por la concentración de feromona y la información heurística correspondiente. La hormiga genera una solución cuando ha recorrido un conjunto de posiciones que unen al nido con el alimento, por lo tanto se puede plantear una función de costo que sea proporcional a la cantidad de posiciones recorridas por la hormiga. Ya que cada hormiga solo usa un sub-conjunto de posiciones disponibles del total presente en el mapa, y además cada trayectoria tiene asignada una función de costo, se puede plantear el problema como una optimización combinatoria discreta.

Un problema de optimización combinatoria discreta se define formalmente como sigue: el problema  $\Pi$  es una triple  $(\mathcal{S}, f, \Omega)$ , donde  $\mathcal{S}$  es el conjunto de soluciones candidatas,  $f$  es la función de costo que asigna un valor  $f(s)$  a cada solución candidata  $s \in \mathcal{S}$ . Las soluciones que pertenecen al conjunto  $\bar{\mathcal{S}} \subseteq \mathcal{S}$  cumplen con el conjunto de restricciones  $\Omega$  y son llamadas soluciones realizables.

Al interpretar el problema planteado como un problema de optimización combinatoria discreta, las hormigas virtuales ya no recorren un espacio creado por las posiciones de un mapa geográfico, sino un espacio creado por los posibles valores que pueden tomar un conjunto de variables que son introducidas a una función de costo al final de cada recorrido. Esto permite que el ACO pueda ser aplicado a diversos problemas muy distintos al planeamiento de trayectoria para robots móviles.

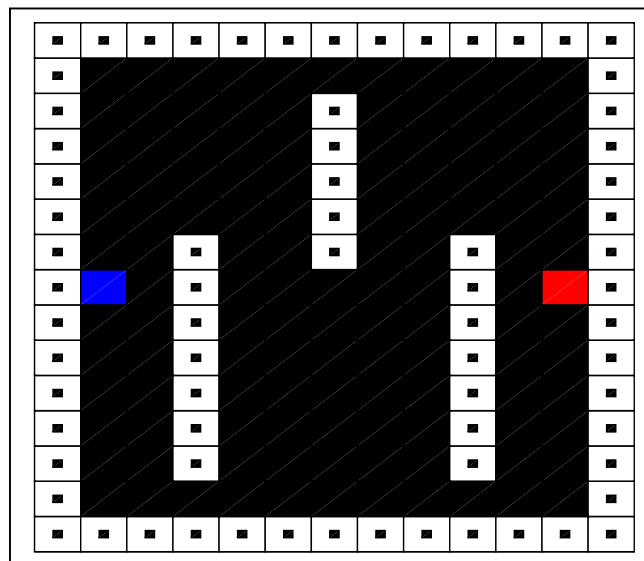
## **3.3 DISEÑO DEL ALGORITMO EN MATLAB**

Dado que el entorno de desarrollo escogido para el presente trabajo es MatLab, se utilizarán matrices para la representación de la mayoría de elementos importantes del algoritmo.

### **3.3.1 Mapa**

La representación del terreno en el cual se desplazan las hormigas está dada por una matriz cuyos elementos solo pueden tomar dos valores: uno o cero. Esta matriz de valores binarios representa los espacios transitables como unos, y los espacios no-transitables como ceros. El mapa posee un “contorno” no transitable de tal forma que la hormiga no pueda traspasar los límites del mapa. Adicionalmente se pueden encontrar obstáculos dentro del mapa, los

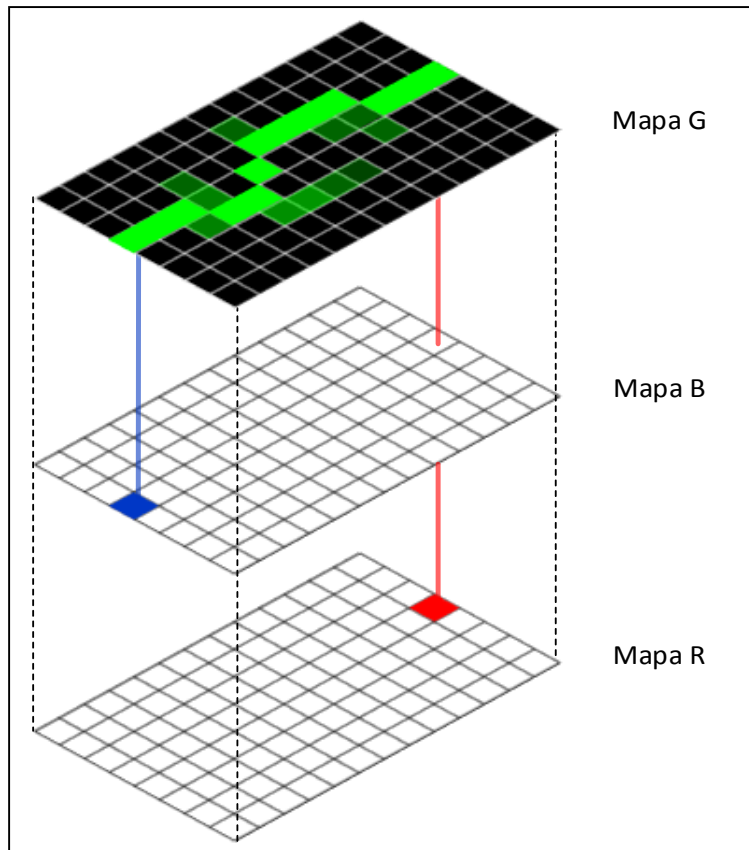
cuales se representan como aglomeraciones de ceros. Para la interfaz gráfica se usó una cuadrícula, en donde cada cuadrado representa un elemento de la matriz. Los espacios transitables se muestran en color negro y los no-transitables en color blanco con un pequeño cuadrado interior. La definición de un mapa se lleva a cabo mediante una función, la cual no solo define la matriz del mapa, sino que además define el punto donde se encuentra el nido y el alimento. En el presente trabajo se considerará un solo nido y un solo punto de alimento. La Figura 3.6 muestra un mapa típico usado para las simulaciones; el casillero azul indica el nido y el casillero rojo indica el alimento.



**Figura 3.6.** Mapa ejemplo para el problema de planeamiento de trayectoria. Fuente: propia.

### 3.3.2 Mapa de feromona

Una vez que se ha definido el mapa sobre el cual se trabajará, se definen tres matrices más cuyas dimensiones serán las mismas que la del mapa. A estas tres matrices se les llama: R, G y B. Los elementos de la matriz R y B son todos iguales a cero, excepto uno solo. En la matriz R el elemento distinto de cero marca la posición del alimento, mientras que en la matriz B dicho elemento marca la posición del nido. La matriz G posee elementos que cuyos valores son números reales mayores que uno. Los elementos de la matriz G representan la concentración de feromona en cada posición del mapa, por lo tanto la matriz G es denominada mapa de feromona. La Figura 3.7 ilustra las tres matrices descritas.

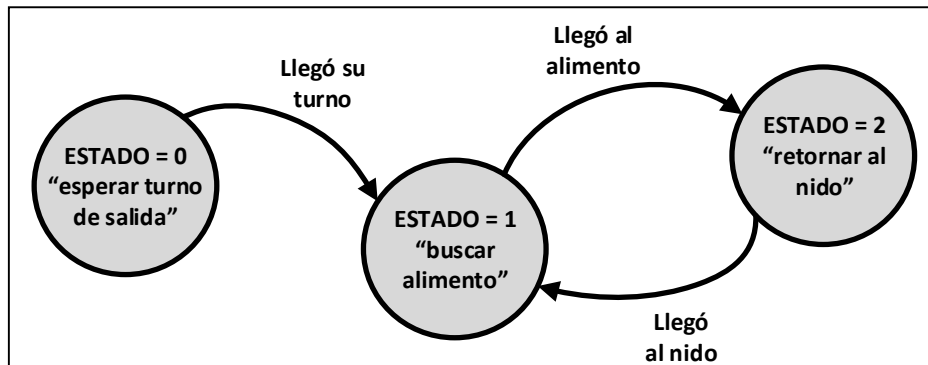


**Figura 3.7.** Mapa ejemplo para el problema de planeamiento de trayectoria. Fuente: propia.

### 3.3.3 Estados de cada hormiga

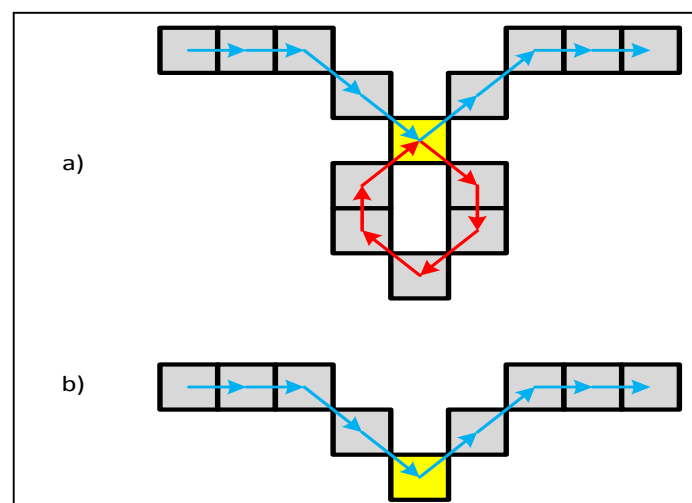
Cada hormiga posee tres estados en los que se puede encontrar: esperar turno de salida, buscar alimento o retornar al nido. Para el problema de planeamiento de trayectoria, se implementó el algoritmo de tal forma que al inicio cada hormiga abandona el nido de forma consecutiva. Para esto, cada hormiga permanece en el nido en estado de “esperar turno de salida”, y no cambiará su estado hasta que la hormiga delante de ella haya salido. La salida consecutiva de las hormigas permite que las hormigas que abandonaron primero el nido depositen feromona en su camino de regreso, al mismo tiempo que las hormigas posteriores comienzan a construir sus trayectorias. Una vez que una hormiga sale del nido, entra en el estado “buscar alimento”. En este estado la hormiga va construyendo una solución de manera pseudo-aleatoria, donde la probabilidad de elegir un camino se ve afectada por la intensidad de feromona depositada en él, así como por la información heurística correspondiente. Durante la búsqueda de alimento la hormiga no deposita feromona, solo la percibe para tomar decisiones, por lo tanto en este estado se lleva a cabo la comunicación en base a estigmergía. Durante la búsqueda de alimento, la hormiga almacena en su memoria cada paso que da. Cuando una hormiga llega hasta el alimento, pasa al estado “retornar al nido”. En este estado la hormiga retornará al nido usando el camino que siguió para llegar al alimento, y en el proceso irá depositando feromona de tal forma que la calidad de su solución quede grabada

en el terreno y pueda ser percibida por el resto de la colonia. Dado el comportamiento descrito, cada hormiga puede representarse como una máquina de estados sencilla como se ilustra en la Figura 3.8.



**Figura 3.8.** Máquina de estados que representa a una hormiga. Fuente: propia.

Antes de emprender su camino de retorno, la hormiga realiza una eliminación de lazos sobre la trayectoria que generó. La eliminación de lazos asegura un retorno más rápido de la hormiga hacia el nido. Para eliminar un lazo dentro de la trayectoria, se realiza una búsqueda de dos posiciones repetidas; si estas existen, se elimina todo el contenido en medio de estas. La Figura 3.9 muestra cómo se implementa la eliminación de lazos en trayectorias. En esta Figura se ilustra un camino con un lazo, las flechas indican la dirección del paso que va tomando la hormiga en cada instante de tiempo. Si se listan todos los pasos que ha dado la hormiga, se podrá notar que un paso se repite, en este caso vendría a ser el paso de la casilla amarilla. Esto nos indica que allí empieza y termina un lazo. Una vez identificado dicho lazo, se procede a remover todos los pasos dentro de dicho lazo. De esta forma el camino que queda ya no presenta dicho lazo y, por lo tanto, toma menos tiempo en llevar a la hormiga de un punto a otro.



**Figura 3.9.** Eliminación de lazos en una trayectoria. a) trayectoria con lazo. b) trayectoria sin lazo. Fuente: propia.

### 3.3.4 Matriz de la colonia

Para administrar de manera sistemática toda la información correspondiente a cada hormiga, se agrupó cada parámetro dentro de una sola matriz, denominada matriz de la colonia. En la matriz de la colonia cada fila representa una hormiga y cada columna representa un parámetro correspondiente a cada hormiga. La Tabla 3.1 muestra los elementos de la matriz de colonia para una hormiga como ejemplo.

**Tabla 3.1.** Ejemplo. Información de una sola hormiga dentro de la matriz de la colonia

fil	col	estado	N2F	F2N	<i>phmG</i>	<i>phmE</i>	compass EN	compass gain	Keep_dir EN	Keep_dir gain	memory
3	4	2	11	5	10	2	1	5	1	3	

A continuación se explica cada elemento de matriz de la colonia mostrada en la Tabla 3.1.

- [*fil, col*]: las dos primeras columnas de la matriz representan la posición de la hormiga dentro del mapa, la cual viene a ser una fila y una columna de esta matriz.
- *estado*: número entero que representa el estado en el que se encuentra la hormiga. Esta variable tendrá el valor de cero cuando la hormiga está en estado “esperar turno de salida”, uno si es que su estado es “buscar alimento”, o dos si su estado es “retornar al nido”.
- *N2F*: número entero positivo que representa la cantidad de pasos dados por la hormiga desde el nido hasta el alimento. Esta variable va aumentando conforme la hormiga construye una solución (estado “buscar alimento”).
- *F2N*: número entero positivo que representa la cantidad de pasos dados por la hormiga desde el alimento hasta el nido cuando esta se encuentra en estado “retornar al nido”.
- *phm gain*: es un número real mayor que 1 que determina la influencia de la feromona en la decisión de una hormiga que la percibe. Se utiliza en la Ecuación 3.2.
- *phm exp*: es el valor del exponente que relaciona la cantidad de pasos dados por la hormiga con el cambio de feromona que esta produce sobre el terreno. Se utiliza en la Ecuación 3.2.
- *compass EN*: número binario que toma el valor de cero cuando no se usa la brújula, y uno cuando si se usa. Esta variable es el habilitador de la brújula.
- *compass gain*: número entero positivo que determina la influencia de la brújula en las decisiones de la hormiga cuando debe elegir por uno u otro camino a seguir.

- *Keep dir EN*: habilitador de la constancia de dirección. Es un valor numérico binario igual a cero cuando la constancia de dirección no está habilitada e igual a uno cuando si lo está.
- *Keep dir gain*: número entero positivo que determina la influencia de la constancia de dirección en las decisiones de la hormiga.
- *memory*: representa la memoria de la hormiga. Es el conjunto de coordenadas correspondientes a los pasos que ha dado la hormiga desde el nido hasta llegar al alimento. La cantidad de pasos almacenable es un parámetro ajustable.

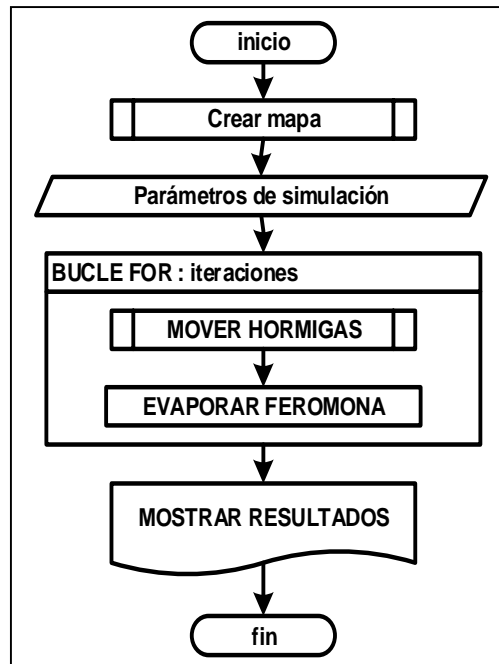
La cantidad de hormigas en la colonia es igual a la cantidad de filas en la matriz de la colonia. De esta manera se logra modificar la población de hormigas de manera versátil.

### **3.3.5 Evaporación de la feromona**

La evaporación de la feromona se lleva a cabo multiplicando al mapa de feromona (matriz G) por un número cuyo valor está entre cero y uno, a este número se le denomina factor de reducción. La evaporación toma lugar una vez que la información de cada hormiga en la matriz de la colonia ha sido actualizada. El factor de reducción de la feromona se determina al inicio del programa.

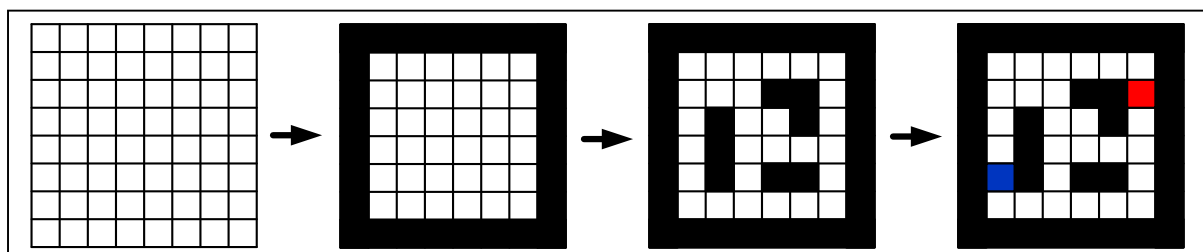
### **3.3.6 Diagramas de flujo**

A continuación se muestran los principales diagramas de flujo que representan los algoritmos implementados. El Diagrama 3.1 muestra el programa principal, el cual hace uso de dos sub-rutinas para la creación del mapa y para el movimiento de las hormigas a través de este. Los parámetros de simulación ingresados son los que determinan el desempeño del algoritmo para un mapa en particular. Uno de los parámetros de entrada es la cantidad de iteraciones que realizará el bucle FOR para mover a las hormigas una por una en una sola sub-rutina. Una vez que todas las hormigas se han movido, se procede a evaporar la feromona del mapa mediante un factor de reducción que también fue ingresado como parámetro de simulación al inicio del programa. La evaporación se lleva a cabo con una multiplicación entre la matriz de feromona y uno menos el factor de reducción de feromona. Los resultados que se muestran son el rastro de feromona dejado por la colonia y las hormigas en su última posición antes de concluir la simulación. En las pruebas de repetitividad se grafica la cantidad de pasos que tomó cada solución versus el número de prueba realizado.



**Diagrama 3.1.** Programa principal. Fuente: propia.

El Diagrama 3.2 muestra la sub-rutina “Crear mapa” detalladamente. Esta sub-rutina fue creada de tal forma que se agilizará el proceso de creación de mapas. La implementación de “Crear mapa” está basada en una interfaz gráfica que permite al usuario definir de manera más eficiente los parámetros del mapa que se usará para probar el algoritmo ACO. Lo primero que hace el usuario es determinar las dimensiones del mapa (filas por columnas) así como el nombre del mismo. A continuación, se define el contorno del mapa como fronteras, es decir, la hormiga no puede traspasar los bordes del mapa. Luego el usuario elige dos puntos dentro del mapa, las cuales son los extremos de una línea recta de obstáculos. El usuario puede ingresar cuantas líneas de obstáculos crea conveniente. Para crear obstáculos con diversas formas geométricas, se usarán varios obstáculos en forma de línea juntos. Una vez que el usuario elige ya no ingresar más obstáculos, este pasa a elegir el punto de inicio y el punto final del mapa, los cuales son marcados con color azul y rojo respectivamente. La Figura 3.10 ilustra el proceso de creación de un mapa paso a paso.

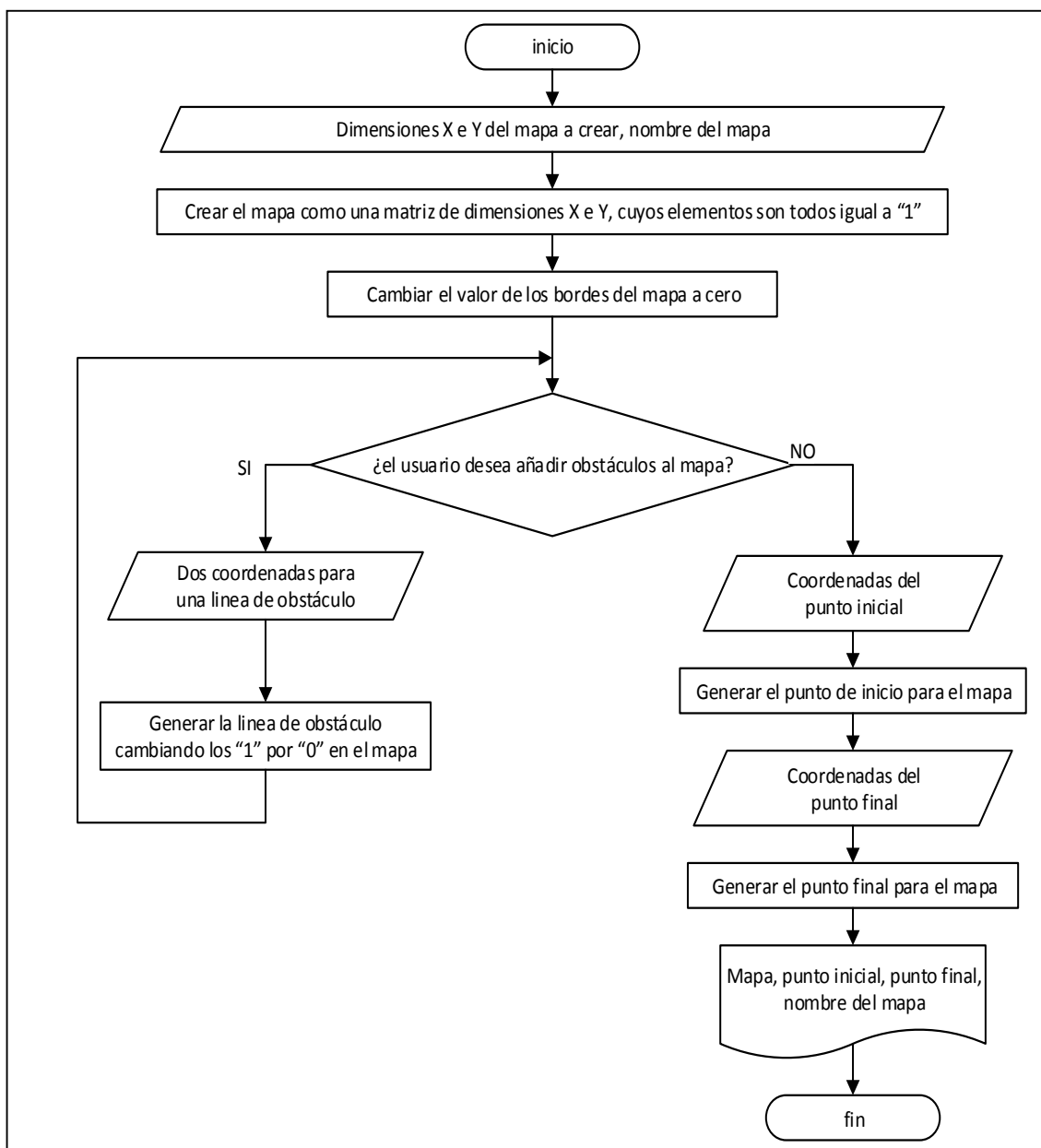


**Figura 3.10.** Proceso de creación de un mapa. Fuente: propia.

El resultado obtenido de esta sub-rutina es una matriz cuyos elementos toman valores iguales a “1” para los espacios transitables y “0” para los espacios no transitables (obstáculos y

fronteras). Además, la sub-rutina entrega dos pares de números, que son las coordenadas del punto inicial y final del mapa, y una cadena de caracteres que viene a ser el nombre del mapa.

El Diagrama 3.3 muestra con mayor detalle la sub-rutina “MOVER HORMIGAS”. La máquina de estados mostrada anteriormente en la Figura 3.8 se implementó usando una estructura del tipo “case”. Esta estructura se representa en el diagrama de flujo como múltiples “IF” consecutivos. La variable que determina qué acciones se llevarán a cabo durante esta sub-rutina es la variable de “estado” de la hormiga. En particular, en los estados 1 y 2 hay bloques condicionales que hacen que una hormiga solo oscile entre estos dos estados. Este ir y venir entre el estado 1 y 2 representa el ir y venir de la hormiga entre el punto inicial y el punto final.



**Diagrama 3.2.** Sub-rutina “Crear mapa”. Fuente: propia.

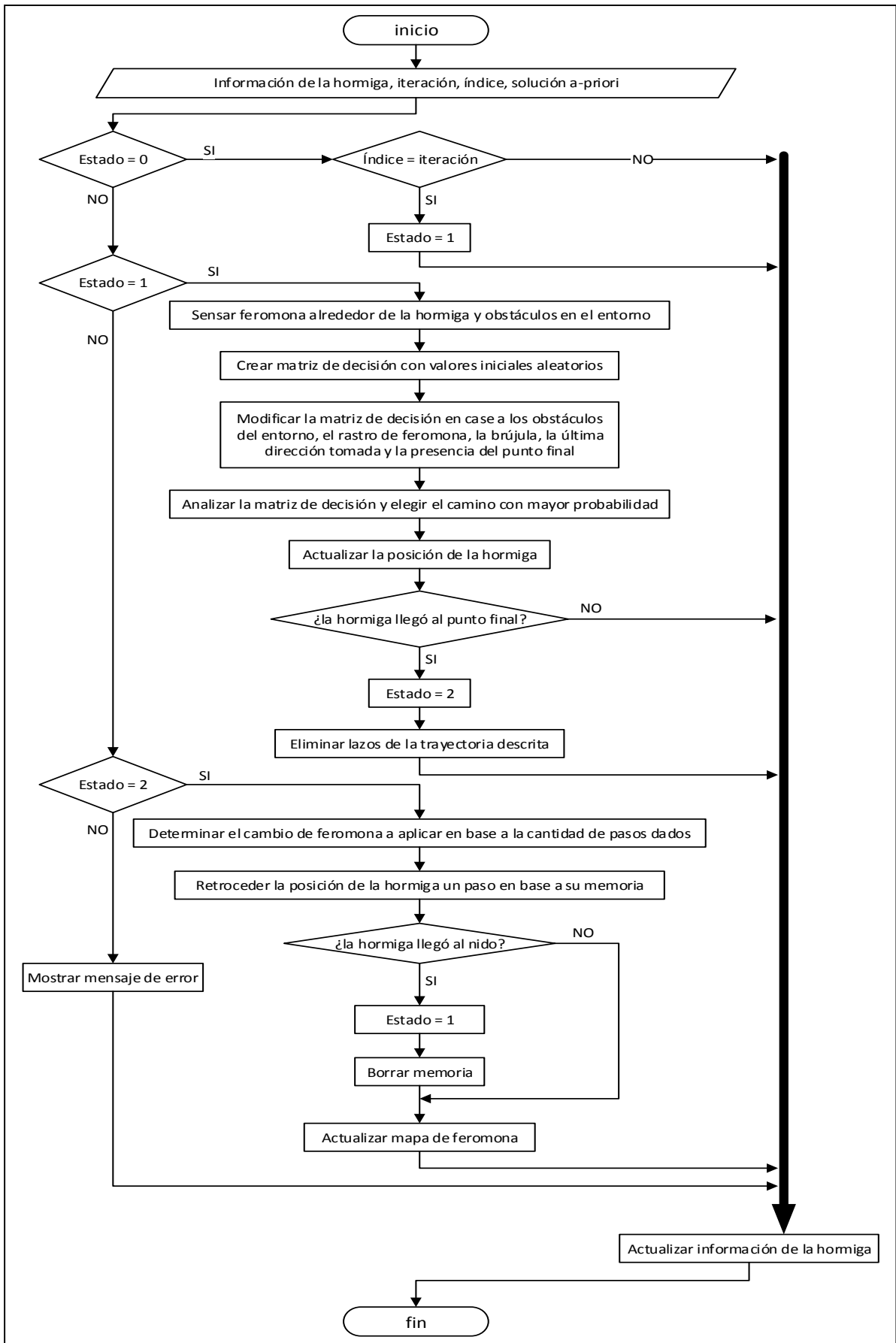


Diagrama 3.3. Sub-rutina "Mover Hormigas". Fuente: propia.

### 3.4 PRUEBAS REALIZADAS

Se realizaron una serie de pruebas para demostrar la eficacia del ACO en la solución a problemas de planeamiento de trayectoria. Las pruebas realizadas van incrementando su complejidad conforme son presentadas. El gráfico de trayectoria hallada muestra los parámetros del algoritmo bajo los cuales se logró alcanzar la respuesta mostrada. Se mostrarán 6 pruebas en total, de las cuales las 3 primeras son par mapas del tipo *camino* y las 3 siguientes son mapas tipo *campo*. Los mapas tipo camino son aquellos en establecen rutas bien definidas que la hormiga puede seguir; son análogos a la situación en las que una hormiga está dentro de su nido y tiene varios túneles por los cuales puede transitar. Los mapas tipo campo son aquellos en los que la hormiga tiene varias áreas disponibles para moverse; estos mapas son análogos a la situación en que la hormiga está en un campo despejado y puede moverse hacia donde desee. Los mapas tipo campo presentan un problema más complejo para el algoritmo ACO, ya que la cantidad de posibles caminos a tomar aumenta considerablemente. La trayectoria hallada se representa por una línea roja que une los puntos hacia los cuales se debe desplazar la colonia para alcanzar su objetivo.

La primera prueba muestra una bifurcación con dos caminos iguales en longitud, por lo cual la convergencia es hacia cualquiera de ellos. Este primer mapa se buscó resolver el problema usando los parámetros que generen la respuesta más rápida (alrededor de 1 segundo); el resto de mapas por probar se resuelven con parámetros que resultan de variar esta primera combinación de parámetros. La Figura 3.11 muestra la solución hallada para este mapa, se puede apreciar el rastro de feromona representado por los casilleros verdes en distintas tonalidades.

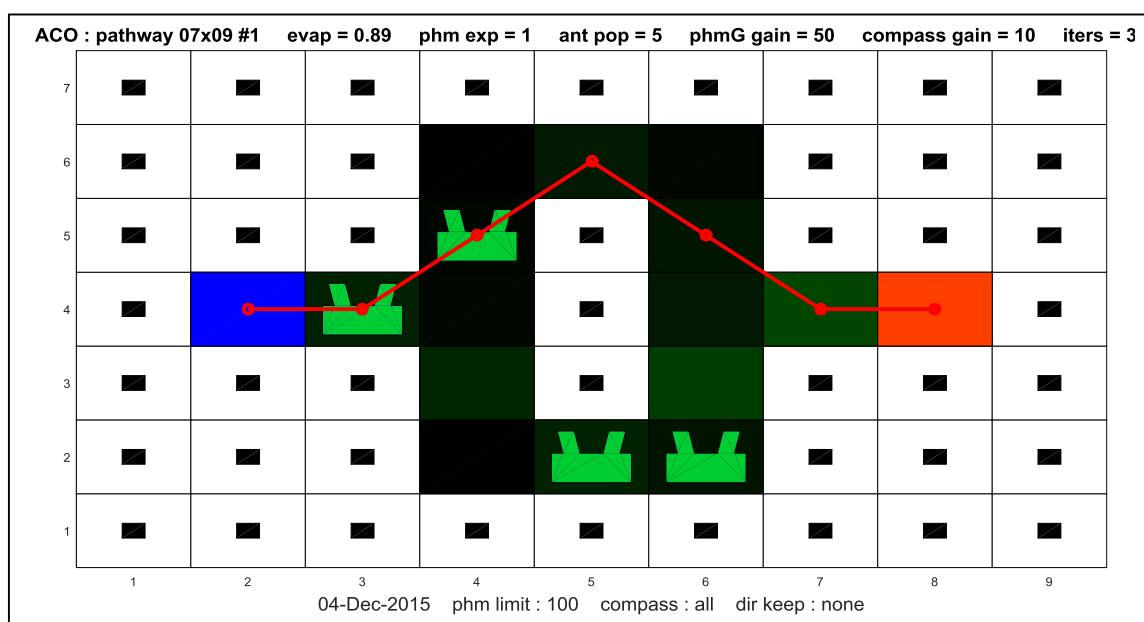
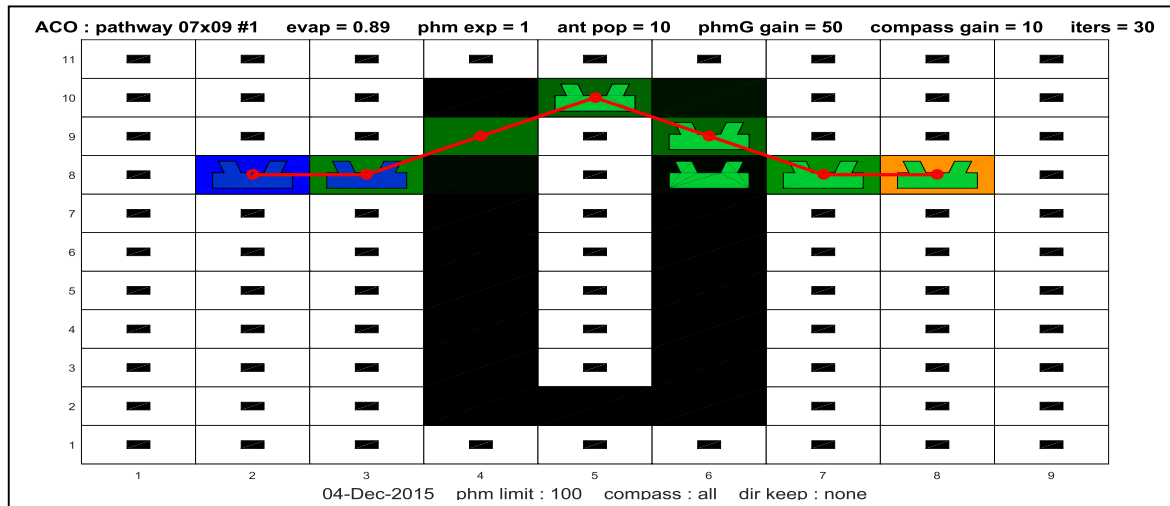


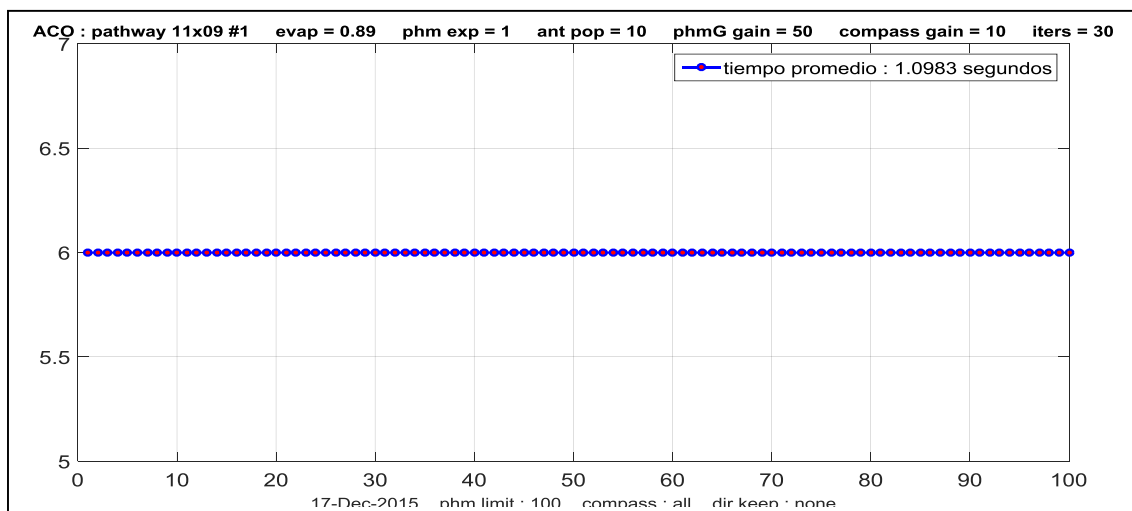
Figura 3.11. Primera prueba del ACO. Trayectoria hallada. Fuente: propia.

Para la siguiente prueba se usó un mapa muy similar al primer, pero con un camino más largo que el otro. Esta prueba y la anterior vendrían a ser las pruebas de doble-puente que se presentaron anteriormente. Se observa que para resolver este problema basta con aumentar la cantidad de hormigas usadas en el problema anterior; se aumentaron de 5 hormigas a 10 hormigas. La Figura 3.12 muestra los resultados para este nuevo mapa.



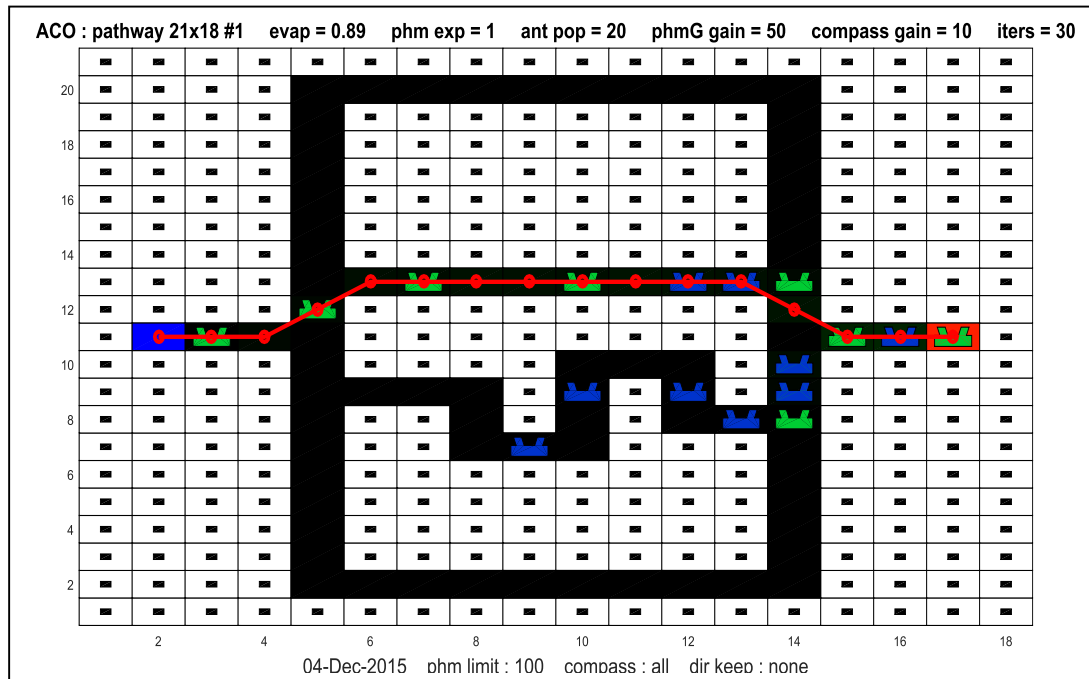
**Figura 3.12.** Segunda prueba del ACO. Trayectoria hallada. Fuente: propia.

Para validar que el algoritmo es uniforme en cuanto a su desempeño, se realizaron 100 pruebas consecutivas con este mapa. De esta forma se demuestra que se obtendrá el mismo resultado para las mismas condiciones de prueba. La Figura 3.13 muestra un gráfico que resume los cien resultados obtenidos; se puede apreciar que el número de pasos del camino obtenido es 6 para las cien pruebas realizadas. Además se muestra que el tiempo que le toma al algoritmo converger a este resultado es prácticamente 1 segundo. Los parámetros de simulación se muestran en la parte superior de la Figura 3.13, estos fueron los mismos usados en la prueba de la Figura 3.12.



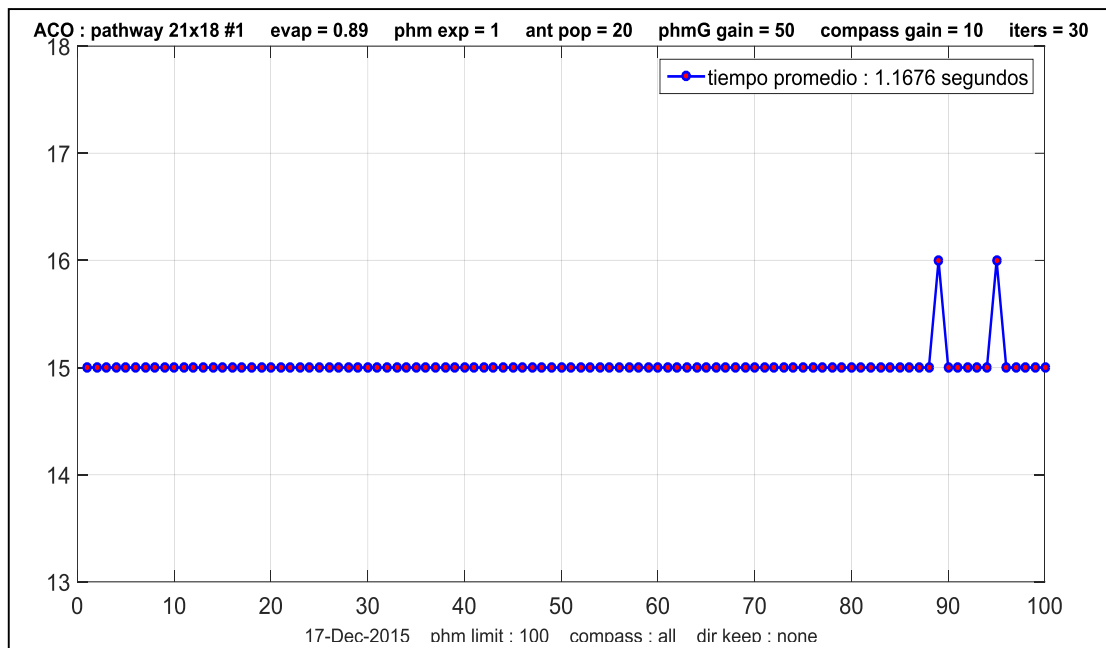
**Figura 3.13.** Segunda prueba del ACO. Uniformidad del resultado. Fuente: propia.

Para el siguiente mapa se plantearon 4 caminos distintos, aquí vemos la ventaja de implementar una “brújula” dentro de las características de las hormigas. Gracias a esto, la colonia descarta los caminos más externos (superior e inferior) y solo se concentra en probar los caminos más próximos a la línea recta que une nido y alimento. Nuevamente, se aumenta la cantidad de hormigas en la colonia para resolver el problema; la colonia ahora cuenta con 20 individuos. La solución esta prueba se muestra en la Figura 3.14 junto con su solución y los parámetros elegidos.



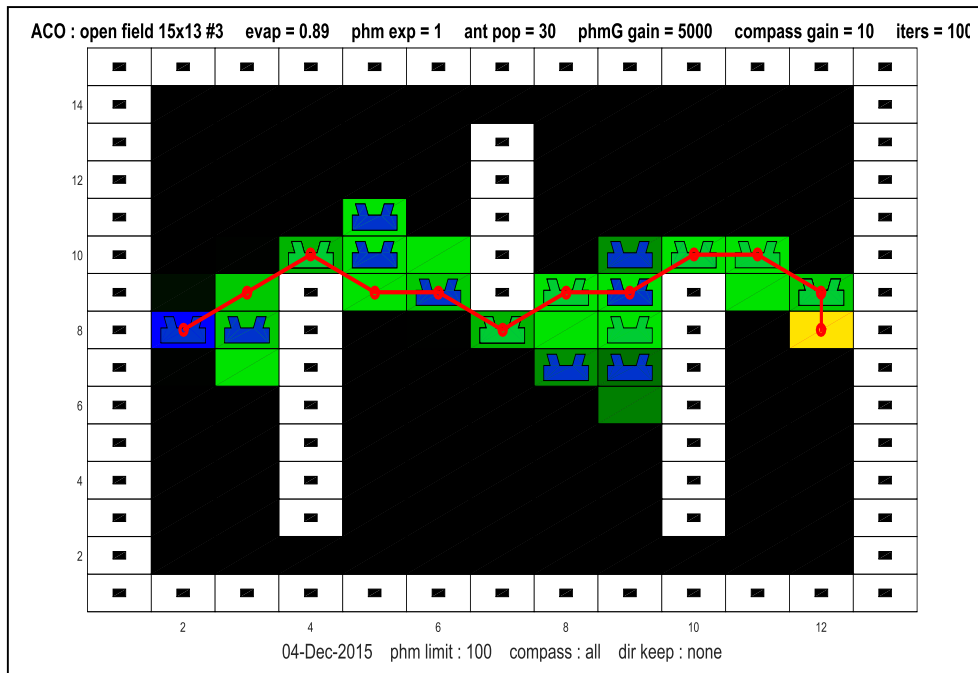
**Figura 3.14.** Tercera prueba del ACO. Trayectoria hallada. Fuente: propia.

De la misma forma que en la prueba anterior, se comprobó la uniformidad del resultado para cien pruebas. La Figura 3.15 muestra los cien resultados obtenidos. Para este caso se puede observar que el tiempo promedio que le toma al algoritmo encontrar el camino más corto es aproximadamente 1.17 segundos. A diferencia del caso anterior, dentro de las cien pruebas se obtuvieron dos que no son iguales al resto. Sin embargo, las dos pruebas que sobresalen en la Figura 3.16 corresponden al mismo camino al que convergieron las hormigas en las 98 pruebas restantes; esto se puede comprobar manualmente observando la longitud de los caminos propuestos. Por lo tanto, los resultados mostrados en la Figura 3.15 nos permiten apreciar la naturaleza heurística del algoritmo, el cual, al no poder llegar al resultado óptimo, nos proporciona un resultado que se aproxima a este. Los parámetros de simulación en las cien pruebas fueron los mismos usados en la prueba individual mostrada en la Figura 3.14 anteriormente.

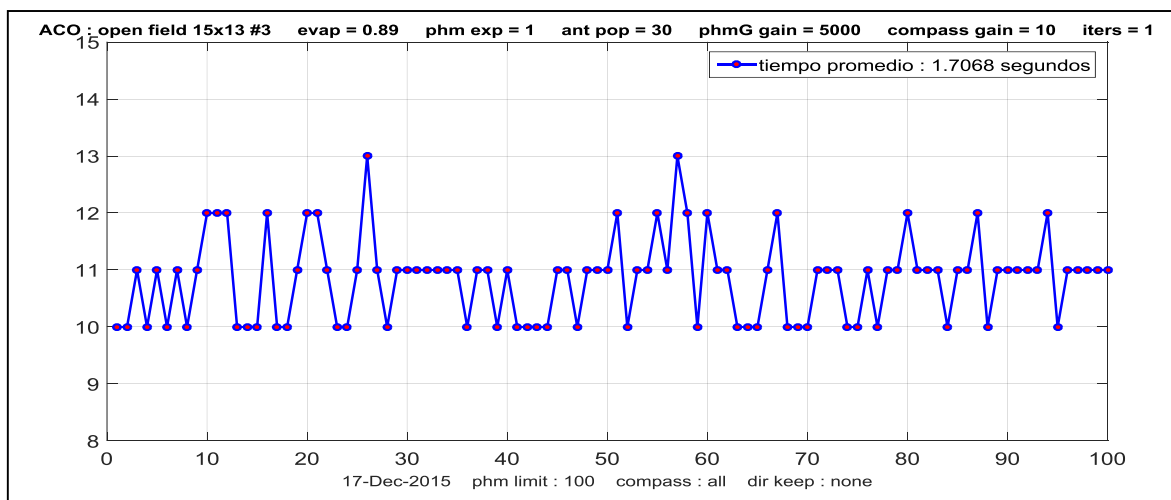


**Figura 3.15.** Tercera prueba del ACO. Uniformidad del resultado. Fuente: propia.

Luego de poner a prueba el algoritmo ACO con mapas tipo camino, se procede a plantear problemas más complejos con el uso de mapas tipo campo. El primer mapa tipo campo es pequeño en dimensiones (largo por ancho) y posee tres obstáculos que proporcionan a la hormiga dos opciones a seguir en cada uno. Ya que ahora la hormiga debe explorar más para encontrar el camino, se deben hacer dos ajustes importantes para solucionar este tipo de problemas. El primer ajuste es aumentar significativamente la ganancia de la feromona; esto corresponde a una mayor concentración de feromona depositada por cada hormiga. Este aumento en la ganancia de la feromona se debe a que ahora el algoritmo tardará más en encontrar el camino más corto y, por lo tanto, la evaporación de la feromona acabaría con la feromona si es que esta se deposita en la misma cantidad que cuando se resolvieron los mapas tipo camino. Debido a que el tiempo de convergencia hacia el camino más corto ha aumentado, también debe aumentar la concentración de feromona depositada para que esta no se evapore por completo antes de que se llegue a una solución aceptable. El segundo ajuste sería la cantidad de iteraciones que se realizarán para resolver el problema. Como se acaba de explicar, el tiempo de convergencia aumenta, por lo tanto el algoritmo debe darle más tiempo a la colonia para encontrar la solución. Estos ajustes se traducen en el cambio de dos parámetros en la ejecución: la ganancia de la feromona ahora es 5000 y las iteraciones realizadas son ahora 100. Esta nueva prueba, sus parámetros y su solución se muestran en la Figura 3.16. Asimismo, en la Figura 3.17 se muestran cien pruebas realizadas para evaluar la uniformidad del resultado.

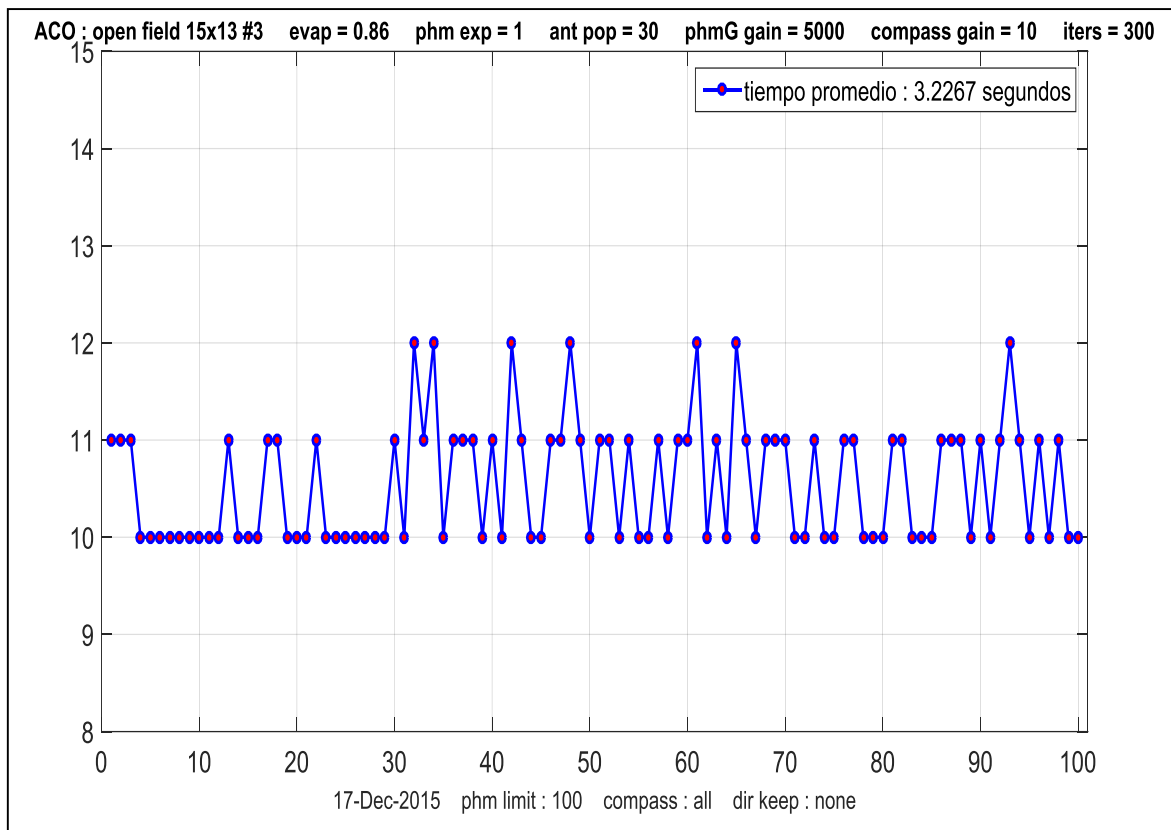


**Figura 3.16.** Cuarta prueba del ACO. Trayectoria hallada. Fuente: propia.



**Figura 3.17.** Cuarta prueba del ACO. Uniformidad del resultado. Fuente: propia.

En la Figura 3.17 se observó que cuando se realiza la simulación mostrada en la Figura 3.16 cien veces, se obtienen resultados de trayectorias con longitudes que oscilan entre 10, 11 y 12 pasos de longitud, y en dos casos se llega hasta 13 pasos de longitud. Se puede apreciar claramente que la mayoría de resultados obtenidos fueron de longitud 11. Al observar esto, se buscó mejorar el desempeño del algoritmo con la variación del factor de evaporación de feromona y la cantidad de iteraciones del algoritmo. El factor de evaporación de feromona se disminuyó a 0.86, de tal forma que se logre olvidar más rápidamente las soluciones menos óptimas (trayectorias más largas). El aumento en el número de iteraciones a 300 nos permite darle más tiempo al algoritmo para que converja hacia el mejor resultado. La Figura 3.18 muestra cien pruebas más con los nuevos parámetros de simulación.

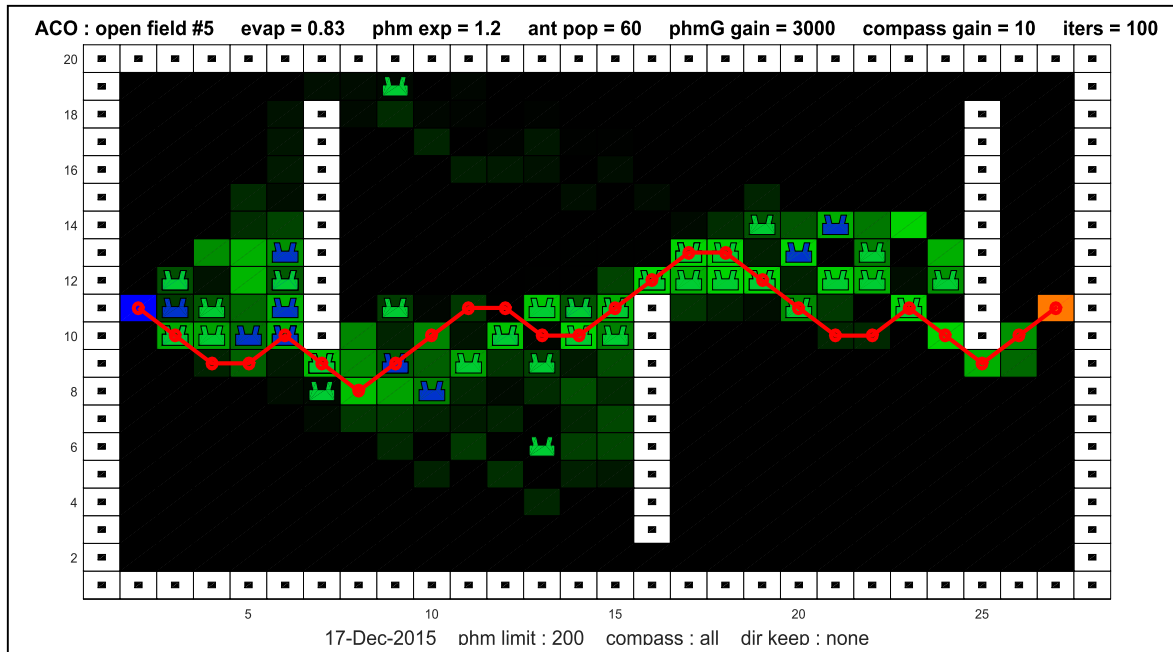


**Figura 3.18.** Cuarta prueba del ACO. Uniformidad del resultado. Mejoras. Fuente: propia.

Ahora se puede observar que la gran mayoría de trayectorias obtenidas contienen 10 o 11 pasos, y solo algunas contienen 12 pasos. Este tipo de mejoras son posibles gracias a las pruebas de uniformidad.

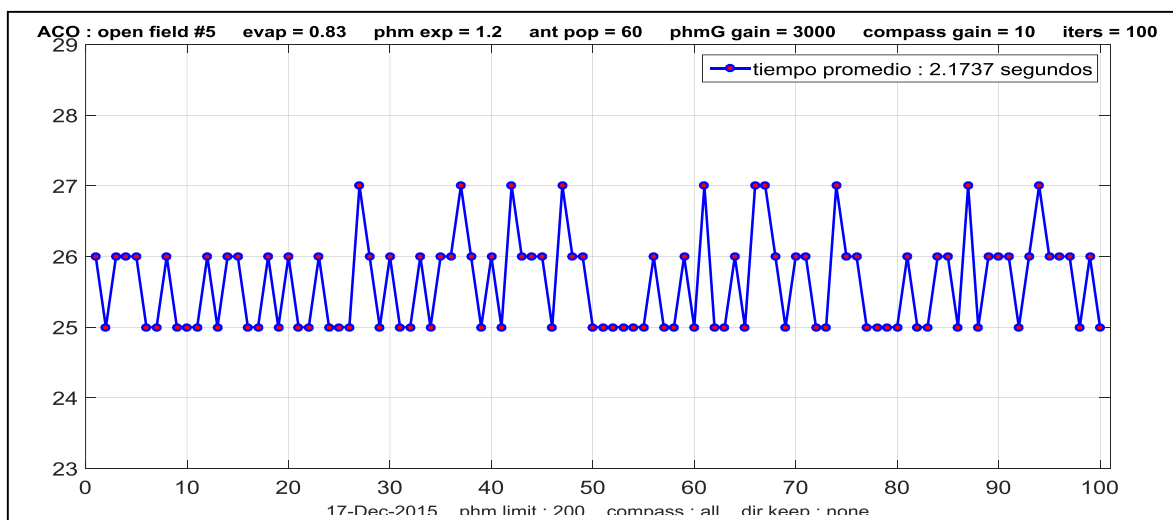
Para la siguiente prueba se mantiene el mismo número de obstáculos, pero complica el problema al aumentar las dimensiones del mapa. Ahora para resolver el problema se hacen dos ajustes más en los parámetros de ejecución manteniendo los cambios realizados en el problema anterior. Ya que el mapa ha aumentado de dimensiones, se generarán muchas más soluciones sub-óptimas, muchas de las cuales tienen calidades similares y pueden llevar a una solución final sub-óptima. Para evitar esto se necesita que la solución que tenga menos pasos destaque significativamente entre las demás; es decir, que su calidad aumente mucho más rápido conforme los pasos dados por la hormiga se reducen. Para lograr este comportamiento se aumenta el exponente de la feromona a un valor de 1.2, de esta forma aquella hormiga que logre encontrar el camino más corto dejará un rastro de feromona mucho más fuerte que las demás hormigas, lo cual llevará a las siguientes hormigas a, probabilísticamente, tomar ese camino con mayor frecuencia. Asimismo, uno de los cambios adicionales que se realizó fue el aumento de la cantidad de hormigas en la colonia. Este cambio se debe a que las pruebas realizadas hasta el momento sugieren que cuanto más grande sea el mapa, mayor debe ser la cantidad de hormigas usadas para la solución. La

colonia ahora cuenta con 60 hormigas. Finalmente, el factor de evaporación disminuyó a 0.83 y la ganancia de la feromona disminuyó a 3000 para descartar más rápidamente las soluciones más largas. La Figura 3.19 muestra el problema planteado, los parámetros de la ejecución del programa y la solución hallada.



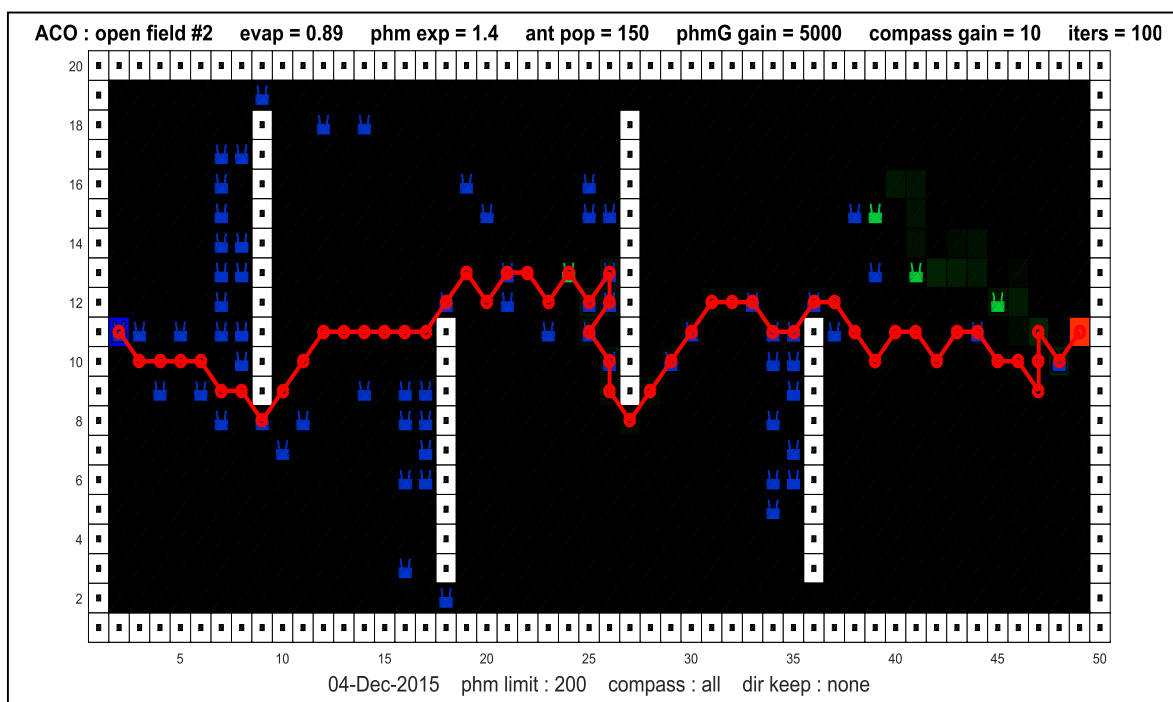
**Figura 3.19.** Quinta prueba del ACO. Trayectoria hallada. Fuente: propia.

Al igual que en las pruebas anteriores, se analizó la uniformidad del desempeño del algoritmo utilizando cien pruebas consecutivas para el mapa de la Figura 3.19. Los resultados se muestran en la Figura 3.20. En dicha Figura se puede observar que la mayoría de caminos hallados tienen longitudes de 25 o 26 pasos, mientras que algunos fueron de 27 pasos, con lo cual se demuestra que el algoritmo logra encontrar soluciones óptimas y aproximadamente óptimas.



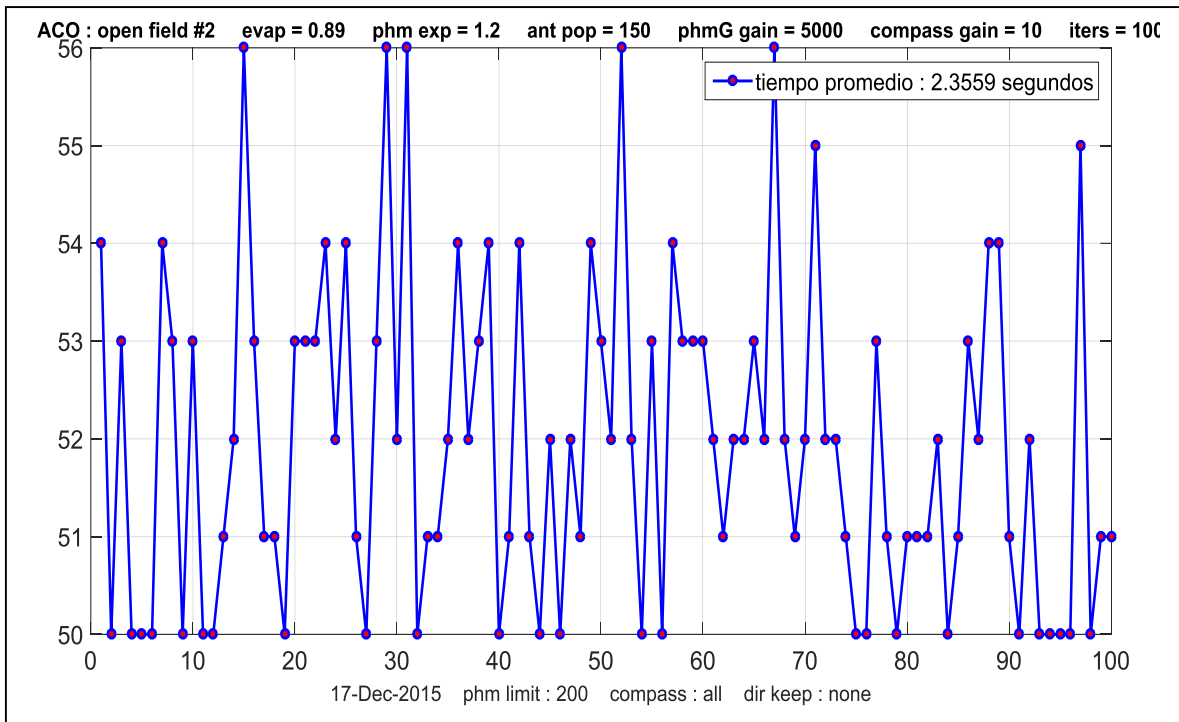
**Figura 3.20.** Quinta prueba del ACO. Uniformidad de la solución. Fuente: propia.

Finalmente, se plantea un último mapa tipo campo con 4 obstáculos y con dimensiones aún mayores. Las dimensiones de este mapa son equivalentes a las dimensiones de la piscina en la cual se llevaron a cabo las pruebas con el barco robot, 50x20 metros cuadrados. Al ser un mapa más grande, la cantidad de hormigas en la colonia vuelve a aumentar, esta vez hasta 150 individuos. Asimismo, el aumento en el tamaño del mapa también genera muchas más soluciones sub-óptimas, por lo cual el exponente de la feromona vuelve a aumentar; el valor usado ahora es 1.4. La solución de este problema es mostrada en la Figura 3.21 junto con los parámetros usados para la ejecución del algoritmo ACO y la trayectoria encontrada (solución). La prueba de uniformidad del resultado se muestra en la Figura 3.22, en esta se puede apreciar que la longitud de los caminos hallados oscila entre 50 y 56. El amplio rango de oscilación de las soluciones se debe a la mayor complejidad del mapa. Ya que este mapa es más grande, presenta mayores opciones a elegir para la colonia, por lo tanto las soluciones encontradas son más variadas. Sin embargo, debe notarse que las soluciones halladas siempre tienden al valor óptimo, lo cual prueba una vez más la eficacia del ACO como método heurístico.



**Figura 3.21.** Sexta prueba del ACO. Trayectoria hallada Fuente: propia.

Se puede observar en las pruebas realizadas que las trayectorias encontradas muchas veces avanzan en “zig-zag”, este es un problema que nace de la resolución del mapa planteado. En la Figura 3.16 cada división del mapa representa 1 metro cuadrado de espacio, si se reduce el tamaño de la división del mapa se puede lograr mejores trayectorias pero el tiempo de convergencia del algoritmo también aumentaría.



**Figura 3.22.** Sexta prueba del ACO. Uniformidad de la solución. Fuente: propia.

## CAPÍTULO 4

### SISTEMA DE CONTROL

En este capítulo se mostrará el diseño de un controlador óptimo utilizando la estrategia propuesta por Fahimi (2009). En esta estrategia se usa un punto de control de tal forma que se logre reducir el número de variables de estado a 2 (partiendo del modelo de 3DOF de Fossen), tal que se tenga el mismo número de variables de estado, que de variables de control. Esta estrategia es usada para lograr controlar el barco robot debido a que es un sistema sub-actuado.

Como se mencionó en el capítulo 2, el modelo a usar, expresado en el sistema BODY, es el siguiente:

$$m_{11}\dot{u} - m_{22}vr + d_{11}u = W_u + F \quad (4.1)$$

$$m_{22}\dot{v} + m_{11}ur + d_{22}v = W_v \quad (4.2)$$

$$I_{zz}\dot{r} + (m_{22} - m_{11})uv + d_{66}r = T \quad (4.3)$$

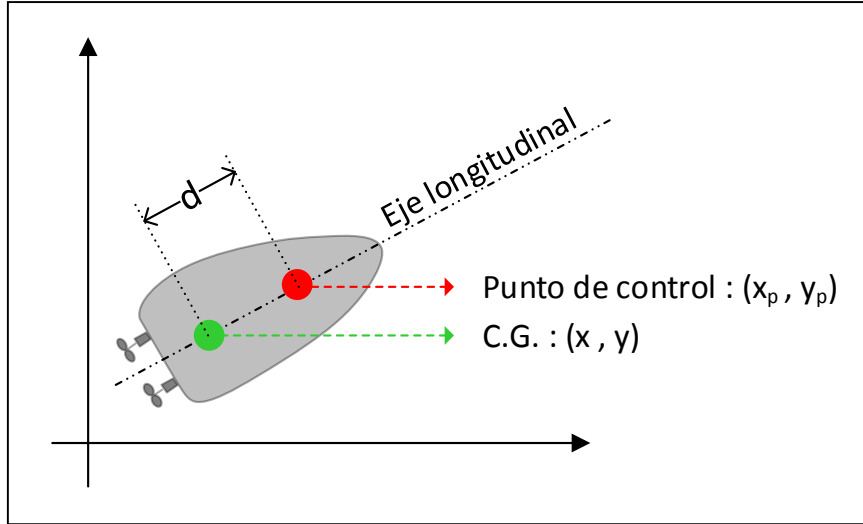
El cual se puede expresar en el sistema NED como se muestra a continuación:

$$\ddot{x} = \frac{1}{m_{11}}(f_x + F \cos \psi) \quad (4.4)$$

$$\ddot{y} = \frac{1}{m_{22}}(f_y + F \sin \psi) \quad (4.5)$$

$$\ddot{\psi} = \frac{1}{I_{zz}}(f_\psi + T) \quad (4.6)$$

Este modelo es sub-actuado, ya que posee dos entradas de control (F y T) y tres variables de estado (BODY:  $[u, v, r]$  o NED:  $[x, y, \psi]$ ). Ante esta situación, Fahimi (2009) propone elegir dos variables para controlar directamente; se eligen dos debido a que hay dos entradas de control. Por lo tanto, se eligen las dos variables de posición. Sin embargo, si se eligen las coordenadas del centro de gravedad, la tercera variable ( $\psi$ ) quedaría indetectable para el sistema, ya que al hacer un seguimiento de las coordenadas del centro de gravedad del barco no se puede saber cuál es su orientación. Debido a esto, las variables de posición que se eligen como salida del sistema (que luego serán retroalimentadas para el control), son las coordenadas de un punto que yace en la línea longitudinal del barco a una distancia “d” del centro de gravedad. Este punto se denomina “punto de control” y se ilustra en la Figura 4.1.



**Figura 4.1.** Punto de control en el barco robot. Fuente: propia.

Ahora que se usan las variables de posición del punto de control en vez del centro de gravedad, se puede realimentar las perturbaciones en la variable  $\psi$  de manera indirecta al realimentar las coordenadas de posición. Este sensado indirecto se aprecia mejor en la Ecuación 4.7.

$$x_p = x + d \cos \psi \quad y_p = y + d \sin \psi \quad (4.7)$$

A partir de las expresiones de la Ecuación 4.7, y sus derivadas en el tiempo, se puede obtener un modelo de segundo orden del sistema. La Ecuación 4.8 muestra la primera derivada de dichas expresiones, y las ecuaciones 4.9 y 4.10 muestran la segunda derivada de las mismas.

$$\dot{x}_p = u \cos(\psi) - (v + rd) \sin(\psi) \quad \dot{y}_p = u \sin(\psi) + (v + rd) \cos(\psi) \quad (4.8)$$

$$\ddot{x}_p = \dot{u} \cos(\psi) - u\dot{\psi} \sin(\psi) - (\dot{v} + \dot{r}d) \sin \psi - (v + rd)\dot{\psi} \cos(\psi) \quad (4.9)$$

$$\ddot{y}_p = \dot{u} \sin(\psi) + u\dot{\psi} \cos(\psi) + (\dot{v} + \dot{r}d) \cos \psi - (v + rd)\dot{\psi} \sin(\psi) \quad (4.10)$$

Se despeja  $\dot{u}$ ,  $\dot{v}$  y  $\dot{r}$  de las Ecuaciones 4.1, 4.2 y 4.3 respectivamente; luego se reemplazan en las Ecuaciones 4.9 y 4.10. Lo que queda, es el modelo de segundo orden del barco, de tal forma que la acción de la variable  $\psi$  es alimentada de manera indirecta a través de la salida, como se muestra en las Ecuaciones 4.11 y 4.12. Gracias a esta estrategia, las perturbaciones en la variable  $\psi$  serán percibidas por el sistema de control.

$$\ddot{x}_p = \left( \frac{\cos \psi}{m_{11}} \right) F - \left( \frac{d \sin \psi}{I_{zz}} \right) T + f_{x1} \quad (4.11)$$

$$\ddot{y}_p = \left( \frac{\sin \psi}{m_{11}} \right) F + \left( \frac{d \cos \psi}{I_{zz}} \right) T + f_{x2} \quad (4.12)$$

Donde:

$$f_x = (m_{22}vr - d_{11}u)\frac{\cos\psi}{m_{11}} - ur \sin\psi + (m_{11}ur + d_{22}v)\frac{\sin\psi}{m_{22}} - \frac{d \sin\psi}{I_{zz}}((m_{11} - m_{22})uv - d_{66}r) - (v + rd)r \cos\psi \quad (4.13)$$

$$f_y = (m_{22}vr - d_{11}u)\frac{\sin\psi}{m_{11}} + ur \sin\psi - (m_{11}ur + d_{22}v)\frac{\cos\psi}{m_{22}} + \frac{d \cos\psi}{I_{zz}}((m_{11} - m_{22})uv - d_{66}r) - (v + rd)r \sin\psi \quad (4.14)$$

Las ecuaciones 4.11 y 4.12 pueden ser expresadas en una sola ecuación matricial de la siguiente forma:

$$\begin{bmatrix} \ddot{x}_p \\ \ddot{y}_p \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix} + \begin{bmatrix} \frac{\cos\psi}{m_{11}} & -\frac{d \sin\psi}{I_{zz}} \\ \frac{\sin\psi}{m_{11}} & \frac{d \cos\psi}{I_{zz}} \end{bmatrix} \begin{bmatrix} F \\ T \end{bmatrix} \quad (4.15)$$

$$\ddot{z} = f + b u \quad (4.16)$$

Una vez que el sistema tiene esta forma, lo que se busca es una ley de control que lleve a la variable de salida  $z$  hacia un valor deseado  $z^d$ .

Gracias al punto de control, se ha logrado obtener dos señales de salida y dos señales de control en un sistema que originalmente se encontraba sub-actuado (tres señales de salida y dos señales de entrada). La Figura 4.2 ilustra la relación entre el punto de control y el sistema original. En vez de realimentar las señales  $u$ ,  $v$  y  $r$ , se realimentará la ubicación del punto de control. Como se observó en la Ecuación 4.7, las dos variables que representan la ubicación del punto de control poseen información del ángulo de orientación del barco robot.

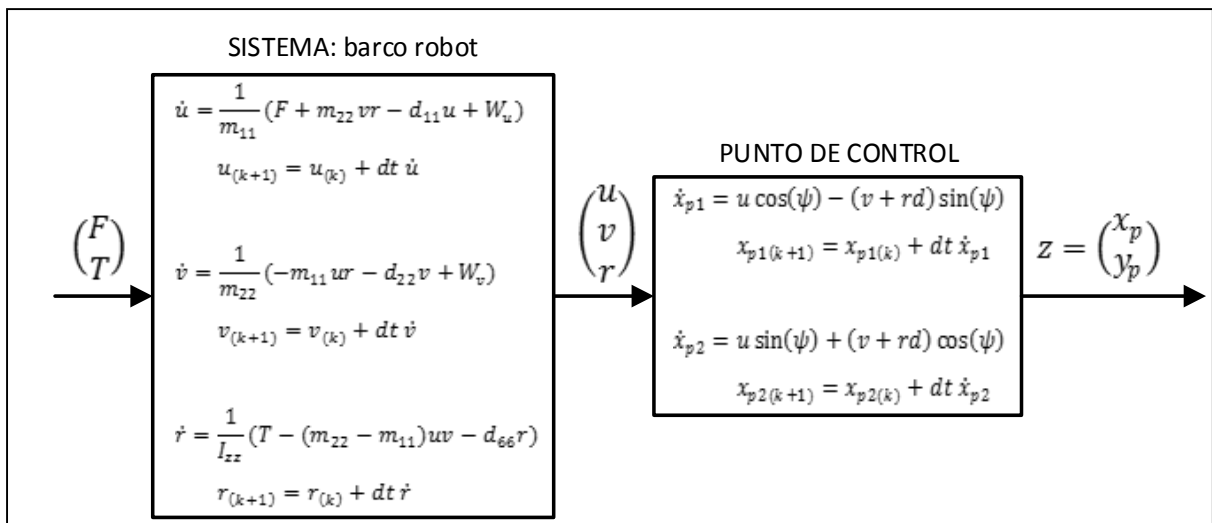


Figura 4.2. Estrategia para trabajar con un sistema sub-actuado. Fuente: propia.

#### 4.1 LINEALIZACIÓN POR REALIMENTACIÓN

La estrategia de linealización por realimentación es tomada del trabajo de Fahimi (2009). Se define una nueva variable de control tal que la Ecuación 4.16 sea lineal, de la siguiente forma:

$$\mathbf{v} = \mathbf{f} + \mathbf{b}\mathbf{u} \quad (4.17)$$

Tal que ahora se tiene la ecuación lineal:

$$\ddot{\mathbf{z}} = \mathbf{v} \quad (4.18)$$

Donde el error respecto a la referencia es:

$$\tilde{\mathbf{z}} = \mathbf{z} - \mathbf{z}^d \quad (4.19)$$

Gracias a la nueva variable “ $\mathbf{v}$ ”, se puede buscar una ley de control despejando “ $\mathbf{u}$ ”:

$$\mathbf{u} = \mathbf{b}^{-1}(\mathbf{v} - \mathbf{f}) \quad (4.20)$$

Donde:

$$\mathbf{v} = -(\mathbf{K}_z \quad \mathbf{K}_{\dot{z}}) \begin{pmatrix} \mathbf{z} - \mathbf{z}^d \\ \dot{\mathbf{z}} \end{pmatrix} \quad (4.21)$$

Cabe resaltar que la salida del bloque “Punto de Control” es tanto la ubicación del punto de control como su derivada en el tiempo. Sin embargo, solo la ubicación del punto de control es utilizada para compararse con una referencia y producir un error que ingresará al bloque del controlador. La velocidad del punto de control no se compara con ninguna referencia ingresada al sistema, e ingresa directamente al bloque del controlador. La Figura 4.3 ilustra el sistema de control en lazo cerrado. En esta figura, se puede apreciar que solo las coordenadas del punto de control son comparadas con la referencia proporcionada al sistema. Esto significa que, el usuario (o sistema superior) que vea este sistema de control como un bloque, la variable de salida son las coordenadas del punto de control.

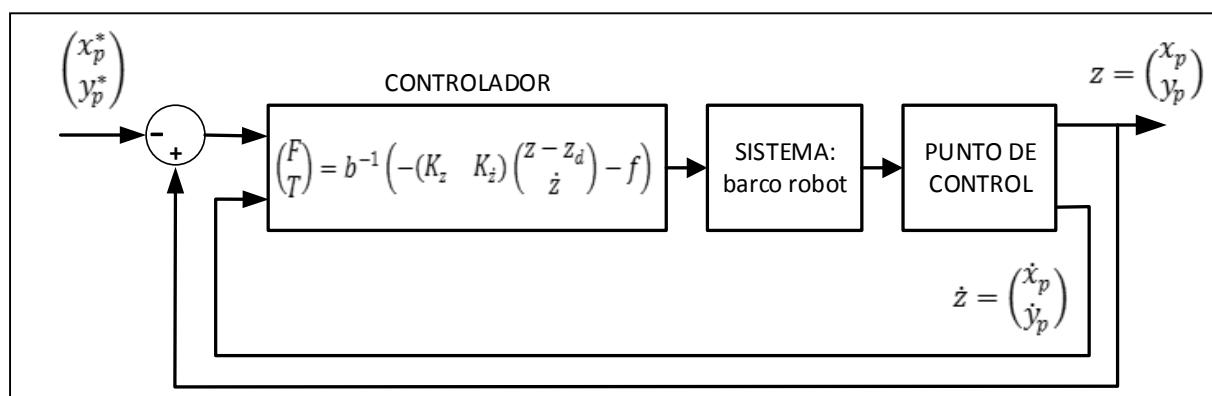


Figura 4.3. Bloque de control y sistema completo en lazo cerrado. Fuente: propia.

Por tratarse de una linealización por realimentación, el sistema quedará expresado como se muestra en las Ecuaciones 4.22, 4.23 y 4.24

$$\dot{z} = Az + Bv \quad (4.22)$$

$$\begin{pmatrix} \dot{z} \\ \ddot{z} \end{pmatrix} = \begin{pmatrix} 0 & \mathbb{I} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} z \\ \dot{z} \end{pmatrix} + \begin{pmatrix} 0 \\ I \end{pmatrix} v \quad (4.23)$$

$$\begin{pmatrix} \dot{x}_p \\ \dot{y}_p \\ \dot{x}_p \\ \dot{y}_p \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_p \\ y_p \\ \dot{x}_p \\ \dot{y}_p \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} v \quad (4.24)$$

## 4.2 DISEÑO DEL SISTEMA DE CONTROL

Las ganancias del controlador, mostradas en la Ecuación 4.21, se obtienen mediante la solución de la ecuación de Riccati. La ecuación de Riccati, resuelta en el entorno MatLab con el comando “are”, es la que se muestra en la Ecuación 4.25.

$$A^T P + P A - P B B^T P + Q = 0 \quad (4.25)$$

En esta ecuación se utilizan las matrices A y B de la Ecuación 4.22. Se utiliza el término  $BB^T$  de tal forma que se obtenga una matriz simétrica y no-negativa definido. Además, se toma una matriz simétrica de pesos mostrada en la Ecuación 4.26. Los elementos de la diagonal de esta matriz de pesos se ajustan hasta obtener una respuesta satisfactoria del sistema de control.

$$Q = \begin{pmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & q_4 \end{pmatrix} \quad (4.26)$$

Luego de resolver la ecuación de Riccati, los elementos de la matriz P se utilizan para obtener las ganancias del controlador óptimo. Esto se muestra en la Ecuación 4.27.

$$K = B^T P = \begin{pmatrix} K_z \\ K_{\dot{z}} \end{pmatrix} \quad (4.27)$$

Ya que la dimensión de la matriz B es de 4x2, y la de la matriz P es de 4x4, la dimensión de la matriz K de ganancias será de 2x4. En esta matriz de ganancias, el primer bloque de 2x2 corresponde a la ubicación del punto de control, mientras que el segundo bloque de 2x2 corresponde a la velocidad del mismo. Esto se ilustra en la Ecuación 4.28, en la cual se puede apreciar que el resultado de todo el cálculo son los valores del vector  $v$ , mostrado anteriormente en la Ecuación 4.21, el cual se utilizará más adelante para determinar los valores de la señal de control como lo muestra la Ecuación 4.20.

$$\begin{pmatrix} K_{z 11} & K_{z 21} & K_{\dot{z} 11} & K_{\dot{z} 21} \\ K_{z 12} & K_{z 22} & K_{\dot{z} 12} & K_{\dot{z} 22} \end{pmatrix} \begin{pmatrix} x_p \\ y_p \\ \dot{x}_p \\ \dot{y}_p \end{pmatrix} = \mathbf{v} \quad (4.28)$$

Es importante notar que todos los resultados hasta ahora mostrados se basan en que la matriz  $b$  de la Ecuación 4.20 es invertible. A continuación se demuestra que el determinante de esta matriz nunca será igual a cero para el barco robot. La Ecuación 4.29 muestra la matriz  $b$  y su determinante se muestra en la Ecuación 4.30.

$$b = \begin{pmatrix} \frac{\cos \psi}{m_{11}} & -\frac{d \sin \psi}{I_{zz}} \\ \frac{\sin \psi}{m_{11}} & \frac{d \cos \psi}{I_{zz}} \end{pmatrix} \quad (4.29)$$

$$\det(b) = \frac{d \cos^2(\psi)}{m_{11} I_{zz}} + \frac{d \sin^2(\psi)}{m_{11} I_{zz}} \quad (4.30)$$

Para que la matriz  $b$  sea siempre invertible, su determinante nunca debe ser igual a cero. Si se reagrupan los términos de la Ecuación 4.30, se tiene la ecuación 4.31:

$$\det(b) = \frac{d}{m_{11} I_{zz}} (\sin^2(\psi) + \cos^2(\psi)) = \frac{d}{m_{11} I_{zz}} \quad (4.31)$$

En la Ecuación 4.31 se puede comprobar que el determinante de la matriz  $b$  nunca será igual a cero para el caso del barco robot ya que  $m_{11}$  y  $I_{zz}$  son parámetros físicos del barco, y  $d$  es la distancia que separa al punto de control del centro de gravedad.

### 4.3 SIMULACIONES REALIZADAS

Para probar la efectividad del controlador diseñado, se utilizó el ejemplo proporcionado por Fahimi (2009). En este ejemplo, se proporcionan todos los coeficientes necesarios para realizar una simulación del barco robot de acuerdo al modelo planteado en las Ecuaciones 4.1, 4.2 y 4.3. Estos parámetros de la simulación se muestran en la Tabla 4.1.

**Tabla 4.1.** Parámetros dinámicos del barco utilizados para la prueba del controlador.

$m_{11}$	$m_{22}$	$d_{11}$	$d_{22}$	$d_{66}$	$I_{zz}$
200 Kg	250 Kg	70 Kg/s	100 Kg/s	50 Kg m <sup>2</sup> /s	700 Kg m <sup>2</sup>

El objetivo de esta prueba es que el barco siga una trayectoria circular de radio igual a 10 metros centrado en el punto (0,0). Se asume que el barco se mueve a una velocidad lineal constante de 1 m/s y al inicio de la simulación el barco está ubicado en el punto (5;-5). La trayectoria a seguir y su derivada en el tiempo, se puede describir por las siguientes ecuaciones:

$$x_p^d(t) = x_c + R \cos\left(V \frac{t}{R}\right) \quad y_p^d(t) = y_c + R \sin\left(V \frac{t}{R}\right) \quad (4.24)$$

$$\dot{x}_p^d(t) = -V \sin\left(V \frac{t}{R}\right) \quad \dot{y}_p^d(t) = V \cos\left(V \frac{t}{R}\right) \quad (4.25)$$

Las condiciones iniciales bajo las cuales empieza la simulación de esta prueba se listan en la Tabla 4.2.

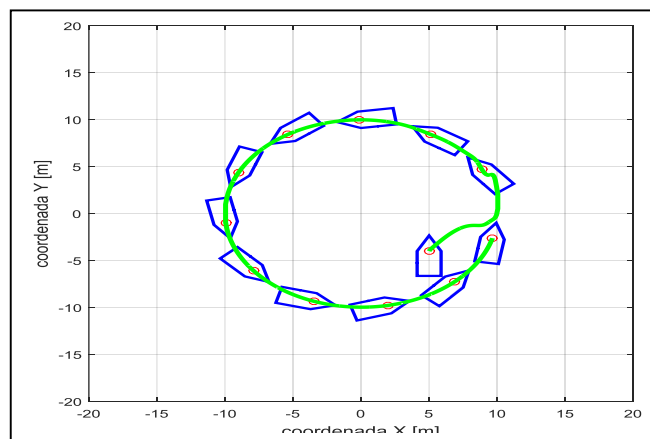
**Tabla 4.2.** Condiciones iniciales para la prueba del controlador.

$x(0)$	$y(0)$	$\psi(0)$	$u(0)$	$v(0)$	$r(0)$
5 m	-5 m	$\pi/2$	1 m/s	0 m/s	0 rad/s

Las primeras pruebas a realizar buscan determinar los valores correctos de la matriz Q en el controlador óptimo tal que se obtenga el comportamiento deseado en el barco robot. Cada elemento en la diagonal de la matriz Q está relacionada a una variable de salida dentro del modelo de Fahimi (2009), estas relaciones se muestra en la Ecuación 4.26.

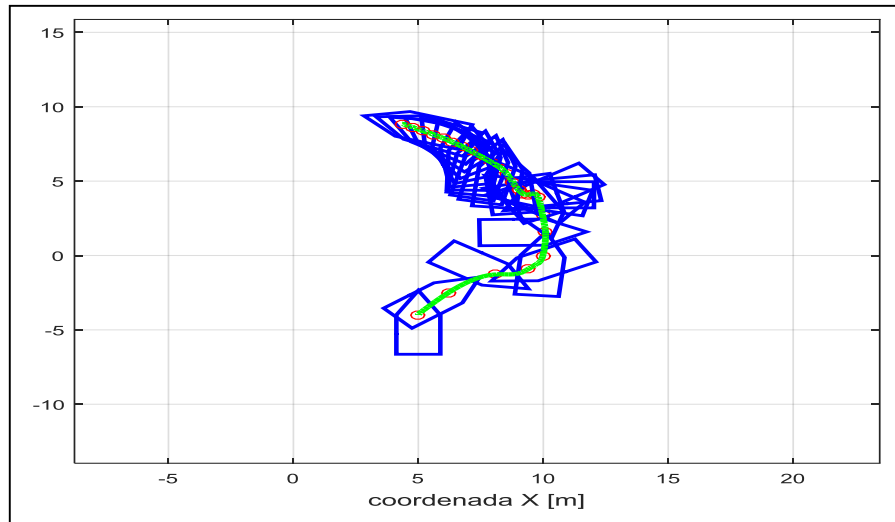
$$Q = \begin{pmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & q_4 \end{pmatrix} \begin{matrix} \rightarrow x_p \\ \rightarrow y_p \\ \rightarrow \dot{x}_p \\ \rightarrow \dot{y}_p \end{matrix} \quad (4.26)$$

En la primera prueba realizada se usaron los siguientes valores en la matriz de pesos Q :  $q_1 = 10$   $q_2 = 10$   $q_3 = 0$   $q_4 = 0$ , como se puede observar, no se usaron pesos correspondientes a la velocidad del punto de control, es decir, la velocidad del punto de control no está siendo controlada. Se eligió empezar por estos valores para ver el efecto que tiene el control de posición y velocidad del punto de control. Los resultados de esta prueba se muestran en la Figura 4.4. En esta figura se muestra la posición del barco robot en distintos instantes de tiempo, se puede comprobar que el controlador cumple con el seguimiento de la trayectoria circular.



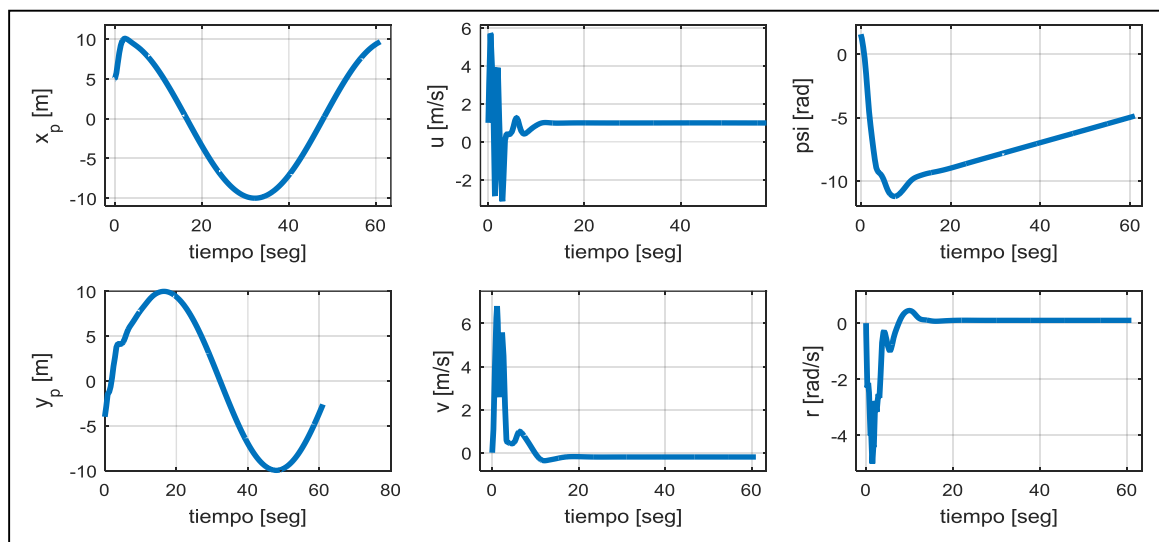
**Figura 4.4.** Seguimiento de trayectoria circular. Primera prueba. Fuente: propia.

Sin embargo, es importante notar que al inicio de la simulación la trayectoria del barco es irregular. Este comportamiento transitorio merece un análisis más detallado, por lo cual se vuelve a ejecutar la simulación solo para este tramo del barco. En esta primera prueba no se consideran perturbaciones. La simulación se muestra en Figura 4.5.

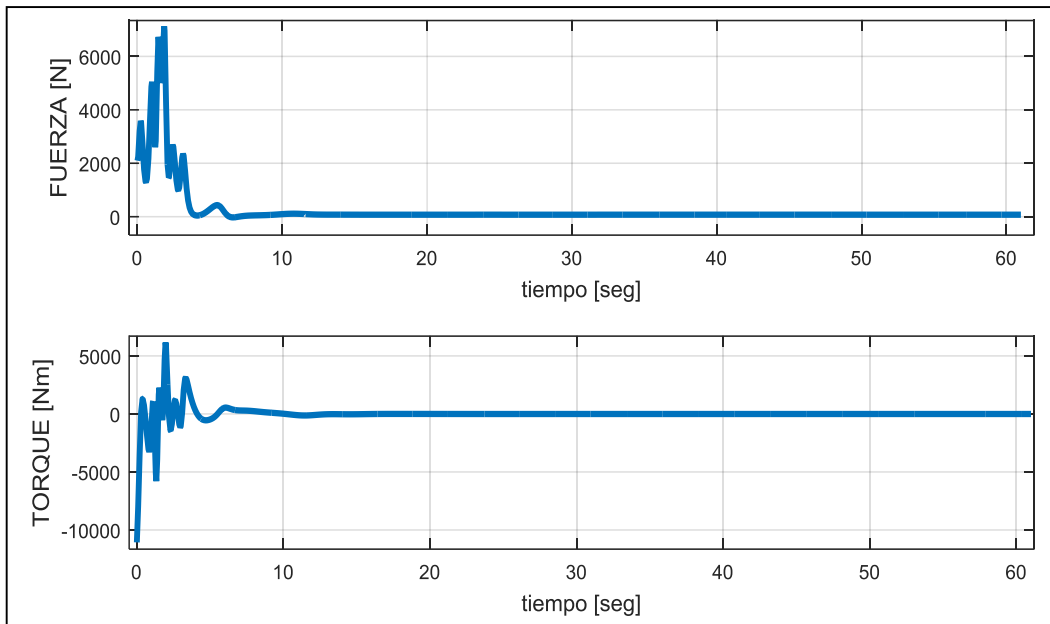


**Figura 4.5.** Primeros instantes de la simulación anterior. Comportamiento agresivo del controlador. Fuente: propia.

La evolución en el tiempo de las coordenadas del punto de control y las variables de estado se grafican en la Figura 4.6. La variable “ $r$ ” representa la velocidad de cambio del ángulo de orientación (“ $yaw$ ”); se puede observar que sufre cambios transitorios muy rápidos al inicio (picos), lo cual conlleva al comportamiento errático del barco al inicio de la simulación mostrado en la Figura 4.5. Las variables de control generadas se muestran en la Figura 4.7, se observan picos muy altos, lo cual es otro aspecto desfavorable con la matriz  $Q$  elegida.

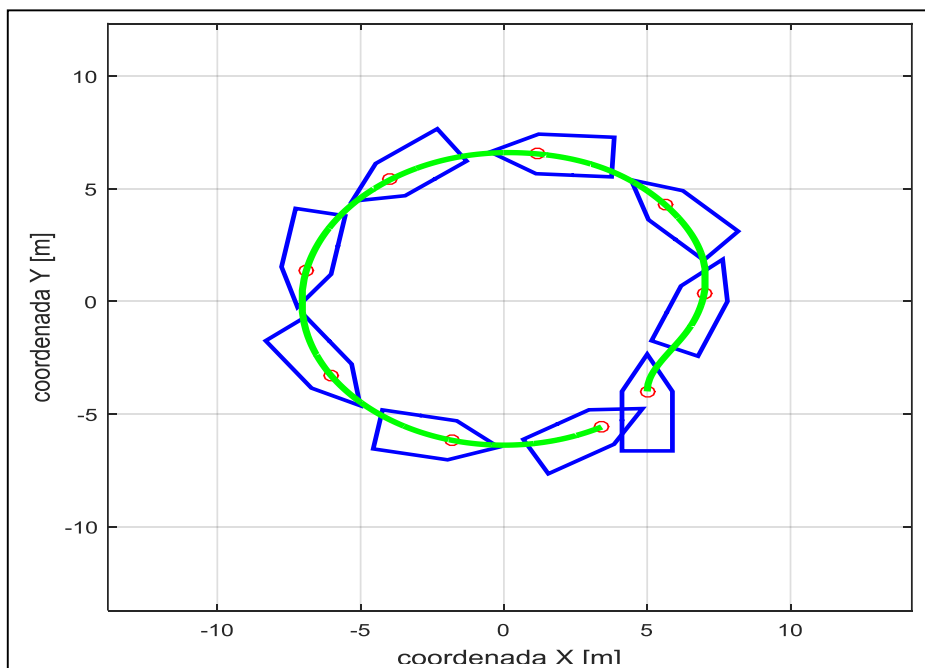


**Figura 4.6.** Prueba del controlador para seguimiento de trayectoria circular. Coordenadas del punto de control y variables de estado. Fuente: propia.

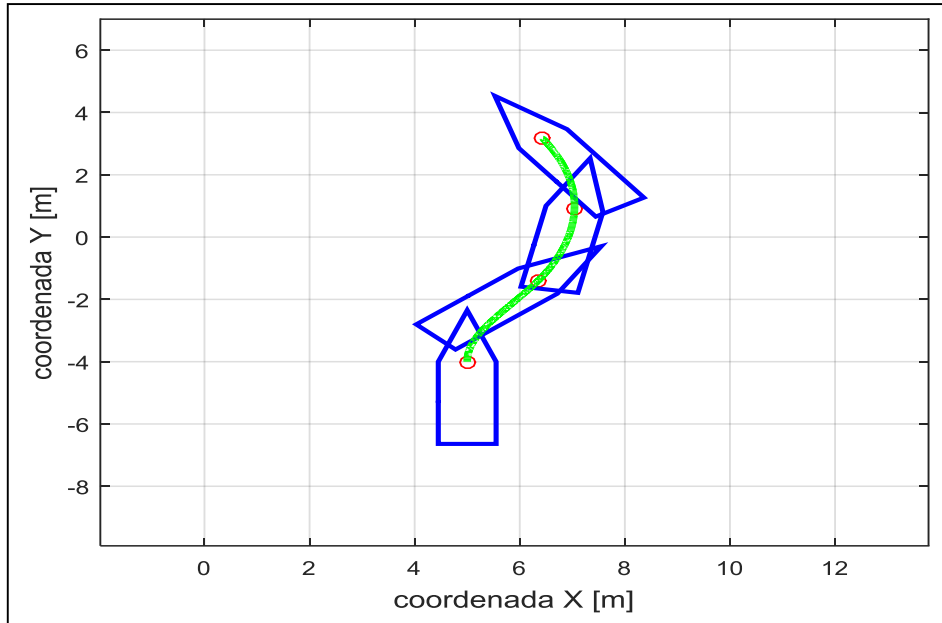


**Figura 4.7.** Prueba del controlador para seguimiento de trayectoria circular. Señales de control. Fuente: propia.

Ahora se buscará mejorar el comportamiento transitorio inicial del barco robot utilizando nuevos valores para la matriz  $Q$  :  $q_1 = 0.01$   $q_2 = 0.01$   $q_3 = 1$   $q_4 = 1$ . Estos valores fueron obtenidos luego de varias pruebas previas. A continuación se muestran los resultados obtenidos con estos nuevos valores: la Figura 4.8 muestra la trayectoria descrita, en donde se observa que la trayectoria es alcanzada satisfactoriamente y, además, el transitorio inicial ha sido eliminado. Para asegurarnos de esto, se muestra la simulación solo para el tramo inicial en la Figura 4.9.

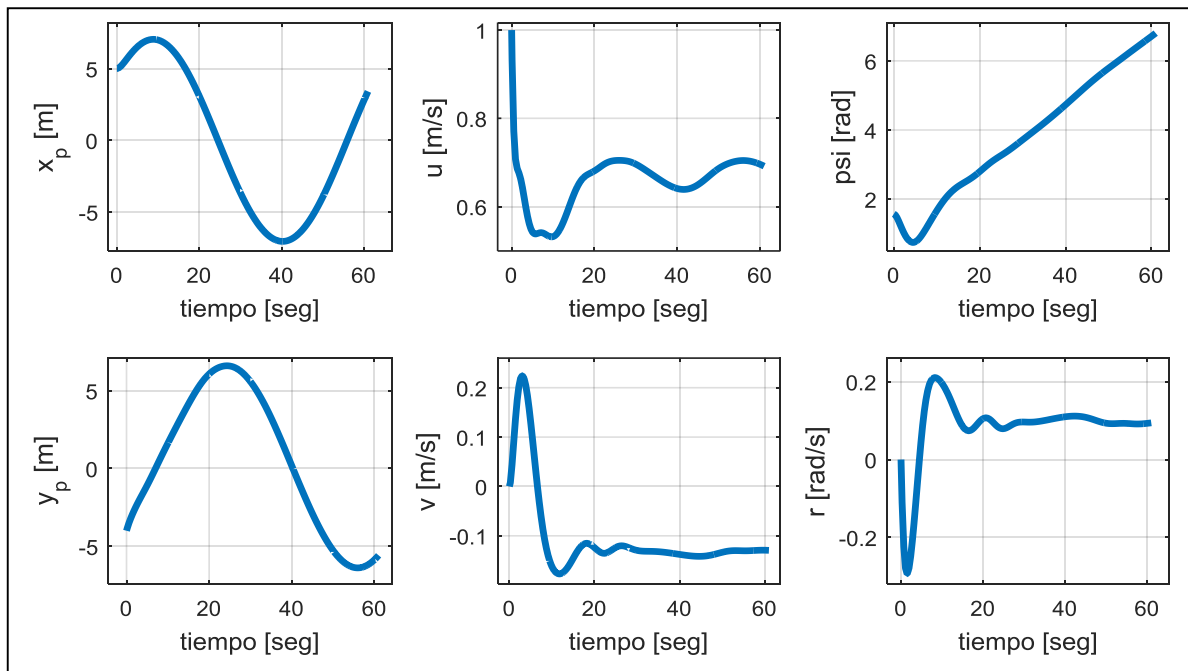


**Figura 4.8.** Seguimiento de trayectoria circular. Segunda prueba. Fuente: propia.

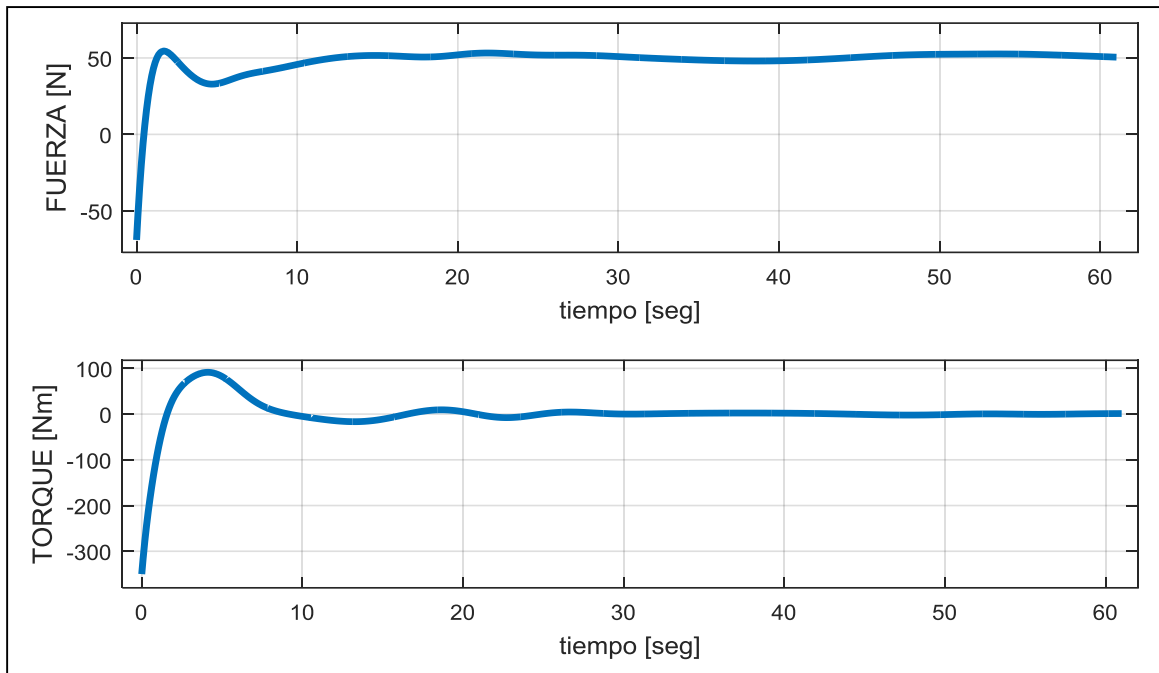


**Figura 4.9.** Primeros instantes de la simulación anterior. Transitorio eliminado. Fuente: propia.

Las variables de posición del punto de control y variables de estado en el tiempo se muestran en la Figura 4.6; las señales de control se muestran en la Figura 4.10



**Figura 4.10.** Prueba del controlador para seguimiento de trayectoria circular. Coordenadas del punto de control y variables de estado. Transitorio eliminado. Fuente: propia.



**Figura 4.11.** Prueba del controlador para seguimiento de trayectoria circular. Señales de control. Fuente: propia.

Con estas pruebas se ha comprobado que el controlador usado genera la respuesta deseada por el barco robot, asimismo se halló los valores necesarios de la matriz  $Q$  del controlador. El tiempo de convergencia del algoritmo es adecuado para la aplicación propuesta.

### PRUEBAS DE CONTROL PARA SEGUIMIENTO DE TRAYECTORIA

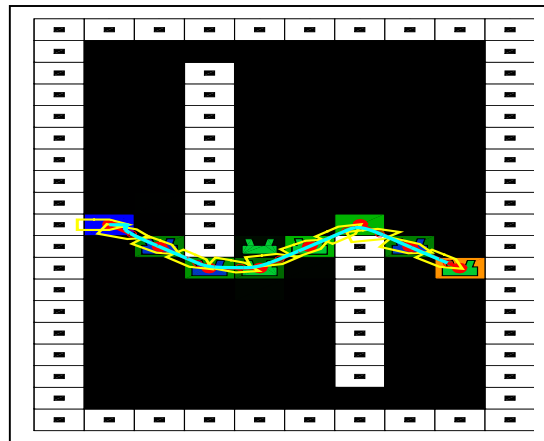
En este capítulo, se pondrá a prueba el funcionamiento del algoritmo de optimización por colonia de hormigas para el planeamiento de trayectoria, visto en el capítulo 3, en conjunto con el controlador óptimo, visto en el capítulo 4.

Las pruebas a realizar se dividen en 3 categorías: aumento de obstáculos de manera uniforme, aumento de resolución del mapa, y aumento de obstáculos de manera no-uniforme. Se empezará por establecer un mapa base, el cual es el mapa más sencillo de todos. Los parámetros que solucionan este mapa base servirán para comparar los parámetros que solucionan los demás mapas en las diversas pruebas realizadas. De esta manera, se podrá apreciar cómo varían los parámetros del algoritmo ACO conforme aumenta la complejidad de los mapas, y cómo responde el sistema de control del barco para recorrer las distintas trayectorias halladas. Los parámetros de simulación propuestos como solución de cada prueba han sido sometidos a una prueba de repetitividad; es decir, se simuló los mapas de cada prueba cien veces con los parámetros propuestos y se obtuvo solo soluciones óptimas o aproximadamente óptimas. Una solución aproximadamente óptima se entiende como una solución que no difiere demasiado de una solución óptima; por ejemplo, si la solución óptima a un mapa es un trayecto de 20 pasos, entonces una solución aproximadamente óptima sería un trayecto de 18 pasos. Las soluciones aproximadamente óptimas son comunes al usar algoritmos heurísticos.

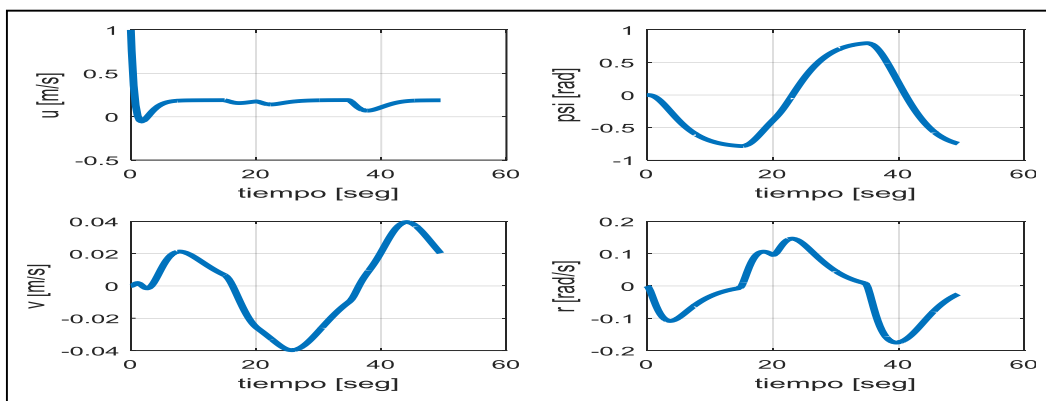
Las gráficas que muestran la solución encontrada por el sistema son similares a las gráficas del capítulo 3, con la diferencia de que en estas ahora se muestra la trayectoria que describiría el barco robot. La trayectoria descrita por el barco robot se muestra como una línea celeste que tiende a tomar la forma de la trayectoria establecida por las hormigas. Por encima de esta línea celeste se ha graficado algunos instantes de la posición y orientación del barco robot para comprobar que efectivamente está siguiendo dicha trayectoria. El barco robot se ilustra como un dibujo sencillo de color amarillo. Además del mapa y la trayectoria del barco robot, se ha incluido gráficas de velocidades lineales y angulares, ángulo de orientación, fuerza y torque. A cada prueba le corresponde una tabla que resume los principales parámetros que generan una solución satisfactoria.

El mapa base usado como punto de comparación para las pruebas se muestra en la Figura 5.1. Este mapa posee solo dos obstáculos, lo cual hace sencillo detectar por simple

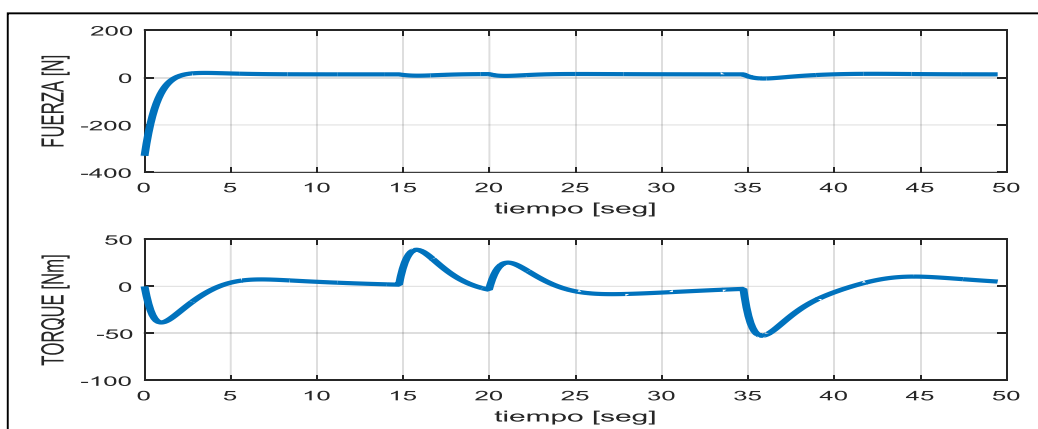
inspección cuál es el camino óptimo a seguir. Las Figuras 5.2 y 5.3 muestran velocidades angulares, ángulo de orientación, fuerzas y torques. Se puede observar en la Figura 5.3 que la fuerza desarrollada por el barco robot solo tiene un valor alto al inicio de la simulación, luego de ello mantiene un valor casi constante. Por otro lado, el torque varía debido a los cambios de dirección que debe hacer el barco robot para esquivar los obstáculos en su camino.



**Figura 5.1.** Mapa base para comparar con las pruebas a realizar. Fuente: propia.



**Figura 5.2.** Velocidades y posición angular para el mapa base. Fuente: propia.



**Figura 5.3.** Fuerza y torque para el mapa base. Fuente: propia.

Los parámetros que llevan al algoritmo ACO a producir una solución óptima o aproximadamente óptima, para este mapa base, se obtuvieron probando distintas

combinaciones hasta obtener los valores adecuados. Sin embargo, la búsqueda de estos valores no fue aleatoria. Tomando en cuenta experimentos pasados, el número de hormigas se fijó a ser el doble que el número de divisiones verticales del mapa; es decir, se inició con 20 hormigas. Luego de esto se estableció un número de iteraciones igual a 4 veces la población de hormigas; es decir, 80 iteraciones. El resto de parámetros tuvo valores iniciales que se muestran en la Tabla 5.1; en otras palabras, no cambiaron. A continuación se disminuyó, de forma alternada, la población de hormigas y la cantidad de iteraciones hasta obtener 100 pruebas consecutivas con resultados óptimos o aproximadamente óptimos. Los parámetros finales a lo que se llegó se muestran en la Tabla 5.1. Se optó por disminuir estos dos parámetros debido a que estos están directamente relacionados al costo computacional de la ejecución del algoritmo. Al disminuir estos dos parámetros se reduce el uso de recursos (como RAM y tiempo de CPU) y el tiempo de ejecución. El algoritmo ACO encontró el camino más corto en 1.71 segundos.

**Tabla 5.1**

Población de hormigas	15
Factor de reducción de feromona	0.3
Ganancia de feromona	50
Exponente de feromona	1
Iteraciones realizadas	65
Ganancia de la brújula	10

A partir del mapa base se estudiarán distintas variaciones. La primera variación será un aumento uniforme de obstáculos. Estas pruebas consisten en aumentar la cantidad de obstáculos sin variar su tamaño, forma u orientación. Se realizaron dos de estas pruebas, una con dos obstáculos adicionales y la otra con cuatro obstáculos adicionales. La solución a la prueba que involucra dos obstáculos adicionales se muestra en la Figura 5.4. Las gráficas de velocidades lineales y angulares, ángulo de orientación, fuerza y torque, se muestran en las Figuras 5.5 y 5.6. Las velocidades, fuerzas y torques son bastante similares a las del mapa base; esto se debe a que las maniobras que debe hacer el barco son las mismas ya que los obstáculos adicionales son iguales a los obstáculos anteriores. La Tabla 5.2 resume los parámetros que solucionan este nuevo mapa.

**Tabla 5.2**

Población de hormigas	40
Factor de reducción de feromona	0.15
Ganancia de feromona	50
Exponente de feromona	1
Iteraciones realizadas	65
Ganancia de la brújula	10

Se observa que al aumentar las dimensiones del mapa, se deben usar más hormigas para la simulación. Sin embargo, al ser más grande el mapa, el tiempo que demoran las hormigas en retornar al nido también se incrementa. Este aumento en el tiempo de retorno significa que la feromona se evaporará más rápido si se utiliza el mismo factor de reducción que en el mapa base. La pronta evaporación de los rastros de feromona eliminan las posibles soluciones encontradas al inicio de la simulación. Este efecto puede llevar a que el algoritmo no determine ningún resultado; es decir, se perdería la convergencia hacia una solución. Para evitar la evaporación de los rastros de feromona cuando estos se están formando, se disminuye el factor de reducción de feromona; esto produce rastros de feromona más duraderos de tal forma que se permita converger a una solución. El resto de parámetros se mantiene constante. El algoritmo ACO se ejecutó en 2.38 segundos.

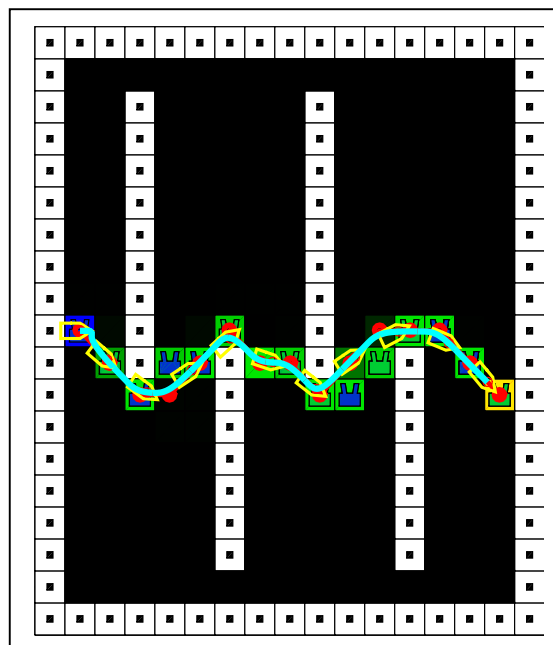


Figura 5.4. Mapa 2 Fuente: propia.

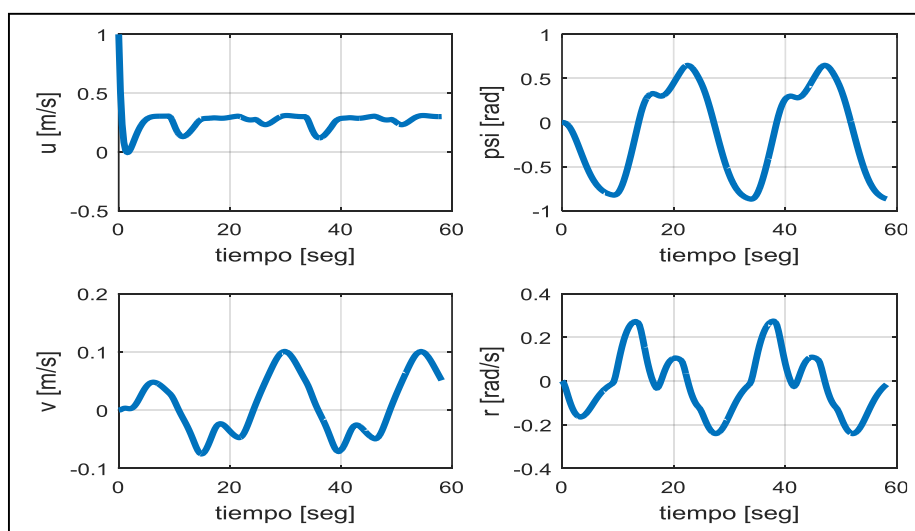
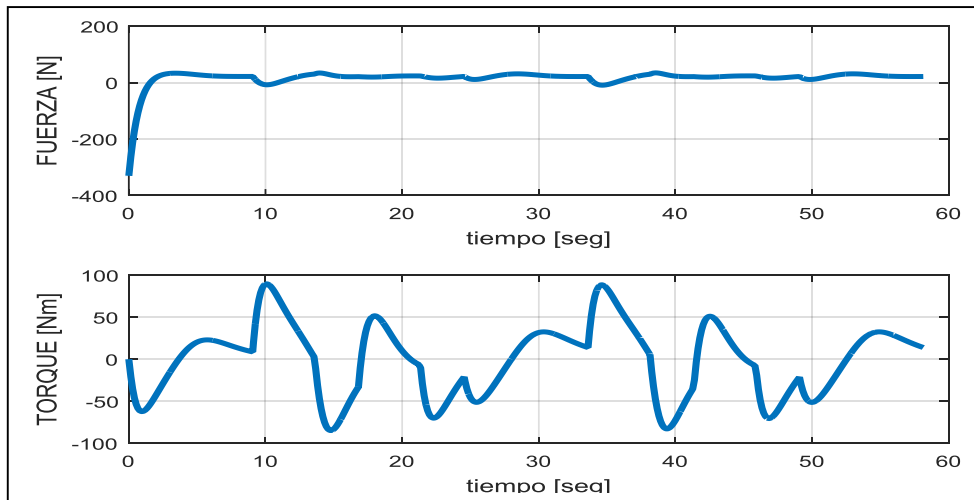
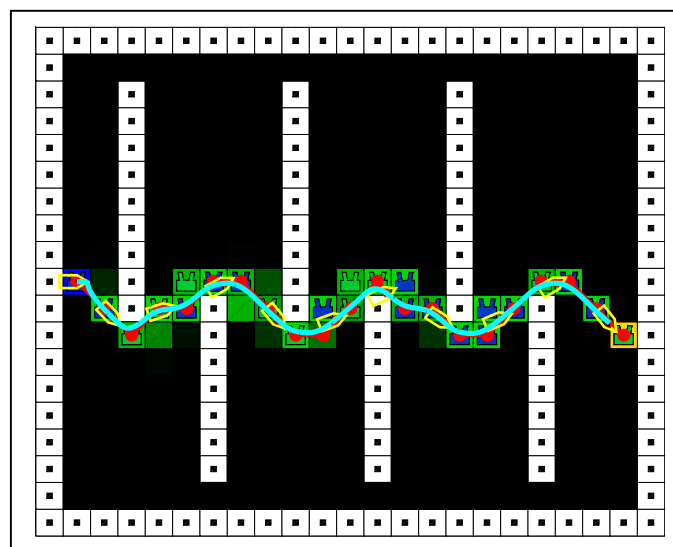


Figura 5.5. Velocidades y posición angular para el mapa 2. Fuente: propia.

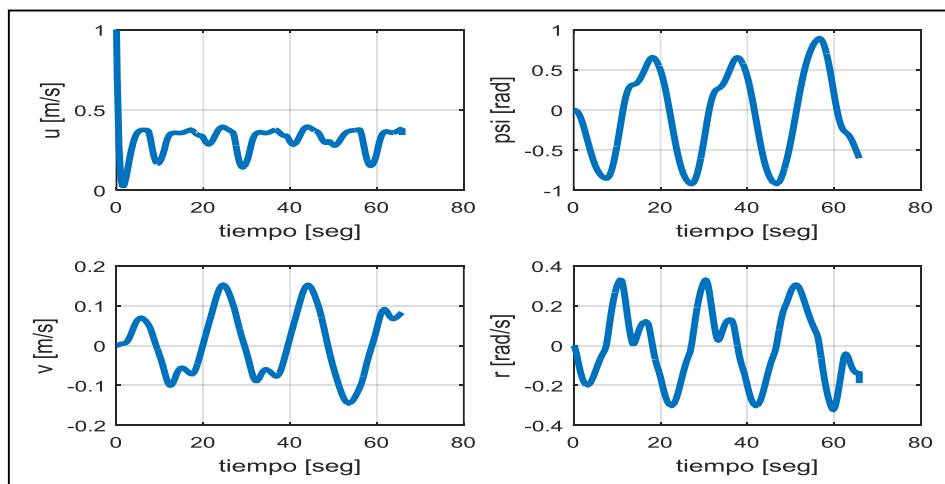


**Figura 5.6.** Fuerza y torque para el mapa 2. Fuente: propia.

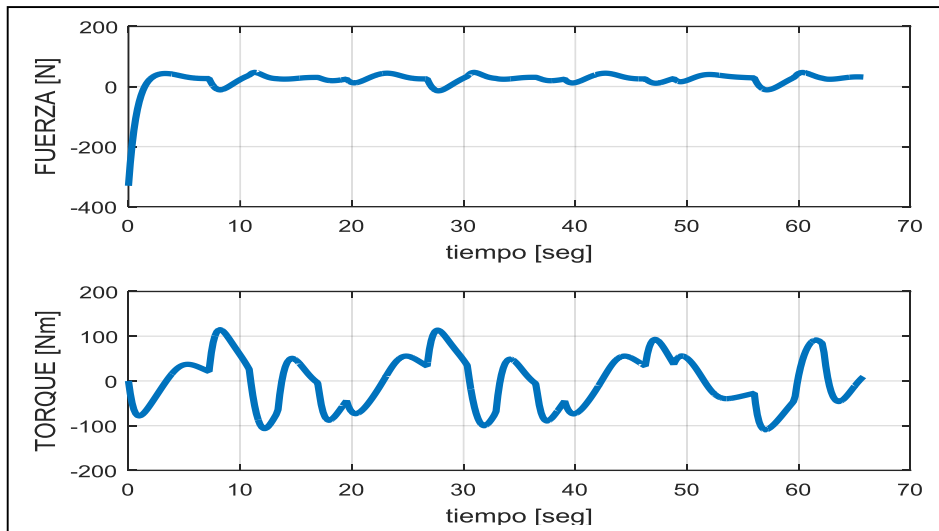
La siguiente prueba contó con 4 obstáculos adicionales. La solución de este mapa se muestra en la Figura 5.7. Las velocidades, posición angular, fuerza y torque están graficadas en las Figuras 5.8 y 5.9.



**Figura 5.7.** Mapa 3. Fuente: propia.



**Figura 5.8.** Velocidades y posición angular para el mapa 3. Fuente: propia.



**Figura 5.9.** Fuerza y torque para el mapa 3. Fuente: propia.

Se observa que las velocidades, posición angular, fuerza y torque mantienen el mismo comportamiento que en la prueba anterior y en el mapa base. Este resultado es conforme con la uniformidad de los obstáculos añadidos. Los parámetros que generan una solución satisfactoria a este mapa se resumen en la Tabla 5.3. Se observa que la población de hormigas ha aumentado nuevamente ya que el mapa ha aumentado de tamaño. Este aumento proporcional de la población de hormigas respecto a las dimensiones del mapa se debe a que cuanto más grande el mapa, mayor cantidad de soluciones (rutas posibles) se necesitan entre el punto inicial y el punto final. Si no se generan suficientes soluciones candidatas al comienzo del algoritmo, se puede converger hacia una solución sub-óptima. Para este nuevo mapa se eligió mantener el factor de reducción de feromona constante y, para evitar la evaporación temprana de las soluciones propuestas, se aumentó la ganancia de la feromona. Con este cambio, a pesar que la feromona está expuesta a la evaporación un mayor tiempo, la cantidad de feromona depositada es mayor, por lo cual su evaporación tomará más tiempo. Sin embargo, el tamaño del mapa ha aumentado de tal manera que es necesario incrementar la cantidad de iteraciones que realizará el algoritmo ACO a 120. Este aumento de iteraciones permite la evaporación de las soluciones sub-óptimas y la convergencia hacia una solución óptima o aproximadamente óptima. Se eligió duplicar la cantidad de iteraciones para mantener una proporción constante con la población de hormigas. El algoritmo ACO se ejecutó en 4.51 segundos.

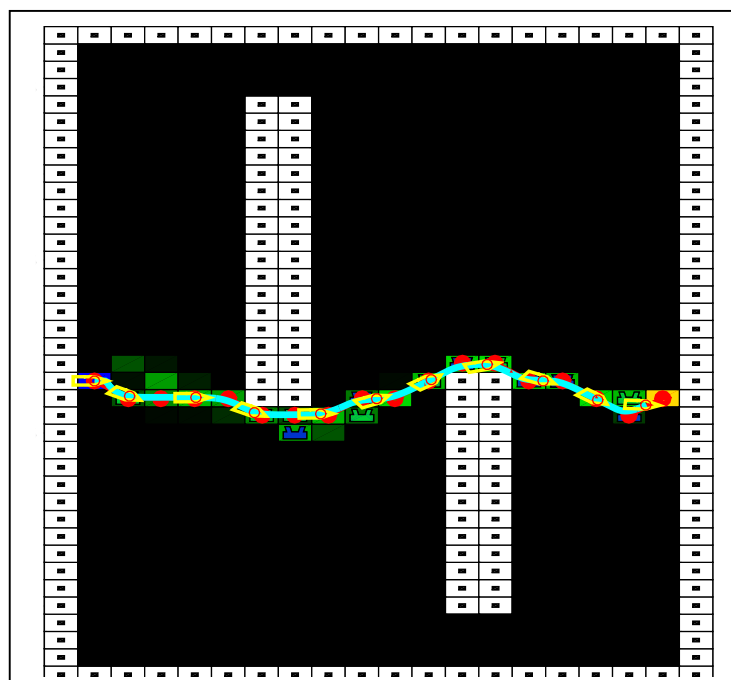
**Tabla 5.3**

Población de hormigas	80
Factor de reducción de feromona	0.15
Ganancia de feromona	100
Exponente de feromona	1.5
Iteraciones realizadas	120
Ganancia de la brújula	10

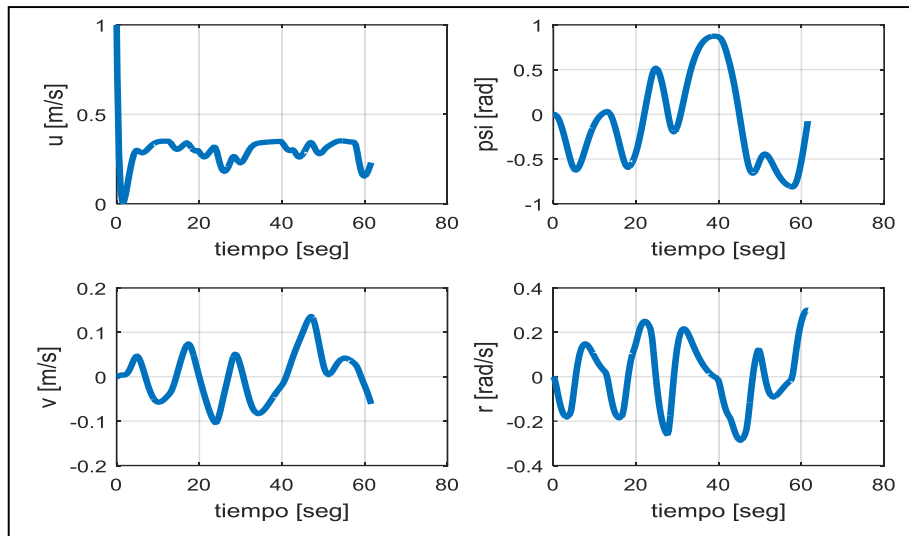
El exponente de la feromona ha aumentado. Este cambio tiene repercusiones sobre la cantidad depositada de feromona en función de la cantidad de pasos que posee una solución. En la ecuación 3.2 se puede observar el efecto de este incremento será una diferencia más marcada entre la feromona depositada cuando la cantidad de pasos tiende al caso óptimo que cuando se encuentra lejos de este valor. Esta mayor diferencia es una forma de destacar las soluciones que tienden a ser óptimas de las que se encuentran lejos del valor óptimo. Incluso, se puede variar este parámetro de tal forma que se tenga un incremento súbitamente alto cuando se esté cerca del valor óptimo.

La siguiente categoría de pruebas está relacionada con la resolución del mapa. Esto se refiere a qué tan particionado está la representación digital del espacio en el cual se desplaza el barco robot. Para esto se llevaron a cabo dos pruebas, con factores de aumento de resolución de 4 y de 16. El aumento de resolución se lleva a cabo tomando una unidad de área del mapa y partiéndola en un número igual de partes. Para la primera prueba de este tipo cada unidad de área pasó a ser 4 unidades de área, mientras que para la segunda prueba se obtuvieron 16 nuevas unidades de área por cada unidad de área del mapa base. Se mantuvo constante la forma y proporción de los obstáculos del mapa base.

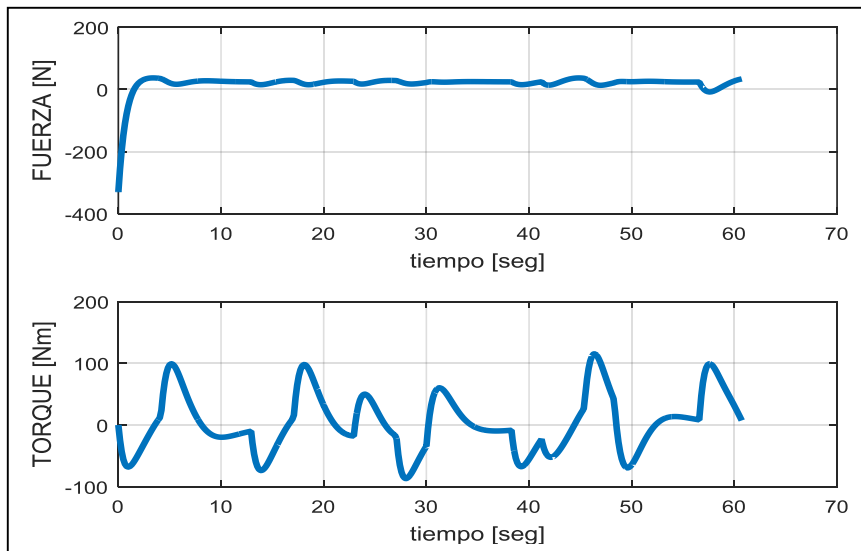
El mapa resuelto para el aumento de resolución con un factor de 4 se muestra en la Figura 5.10. Las velocidades, posición angular, fuerza y torque correspondientes al movimiento del barco robot en este mapa se muestran en las Figuras 5.11 y 5.12. Se observa que la fuerza solo muestra un valor elevado al inicio de la simulación, y se mantiene casi constante después.



**Figura 5.10.** Mapa 4. Fuente: propia.



**Figura 5.11.** Velocidades y posición angular para el mapa 4. Fuente: propia.



**Figura 5.12.** Fuerza y torque para el mapa 4. Fuente: propia.

Los parámetros que solucionan este mapa de resolución aumentada se muestran en la tabla 5.4. Teniendo en cuenta los resultados de las pruebas anteriores, es de esperarse que la población de hormigas aumente. En este caso, además de disminuir el factor de reducción de feromona, se aumentó la ganancia de la feromona. Ambos aumentos se deben a que ahora las dimensiones del mapa han aumentado significativamente, a diferencia del aumento gradual en las anteriores pruebas. El algoritmo ACO demoró 3.63 segundos en ejecutarse.

**Tabla 5.4**

Población de hormigas	30
Factor de reducción de feromona	0.15
Ganancia de feromona	60
Exponente de feromona	2
Iteraciones realizadas	130
Ganancia de la brújula	10

El exponente de la feromona y el número de iteraciones también ha aumentado. En resumen, el programa se ha vuelto mucho más demandante en cuanto a recursos y tiempo. El controlador del barco es capaz de maniobrar por entre los obstáculos al igual que en las anteriores pruebas. Cabe resaltar que los obstáculos no son necesariamente representaciones fieles al obstáculo real con el que podría encontrarse el barco robot. Los obstáculos mostrados en las pruebas realizadas son contornos de seguridad que se dibujan alrededor de un obstáculo real, de tal forma que se tiene un margen de seguridad para evitar posibles colisiones en aplicaciones reales.

La siguiente prueba de aumento de resolución fue por un factor de 16, la solución a esta prueba se muestra en la Figura 5.13. Sus gráficas correspondientes de velocidades, posición angular, fuerza y torque se muestran en las Figuras 5.14 y 5.15. A diferencia que en las demás pruebas, la gráfica de torque en la Figura 5.15 muestra una oscilación mayor de esta señal, esto se debe a que ahora el barco robot debe maniobrar más debido a que hay más espacios para moverse. Esto también tiene repercusión en el gráfico de fuerza en la Figura 5.14, el cual muestra el mismo patrón que en las anteriores pruebas, pero ahora con mayores oscilaciones.

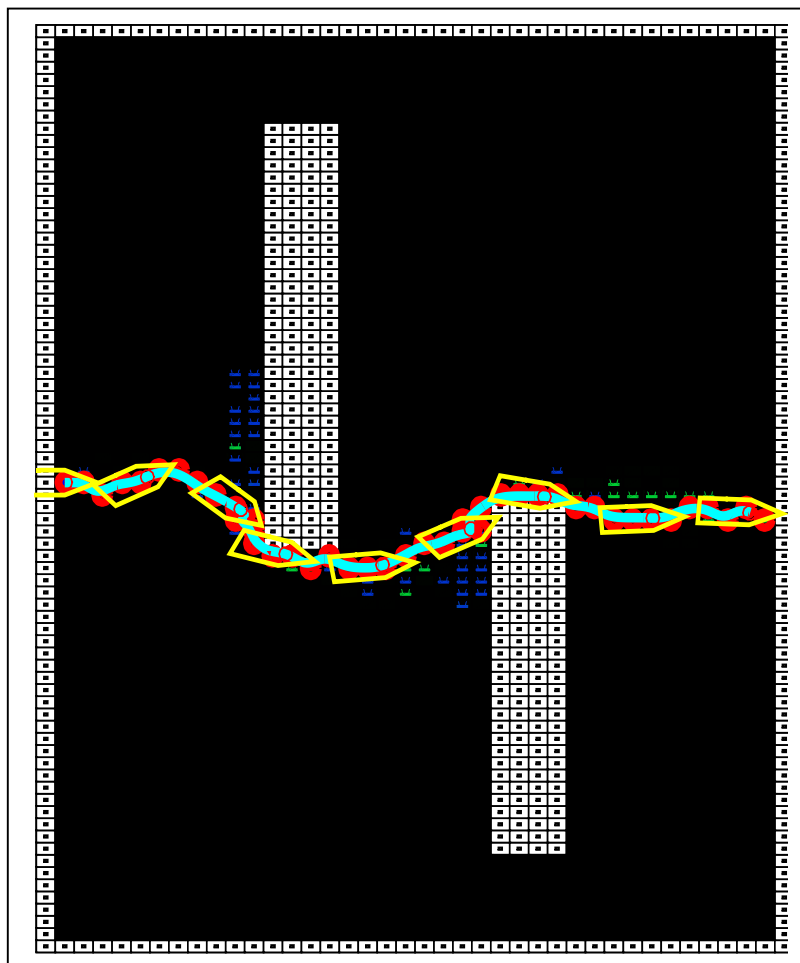
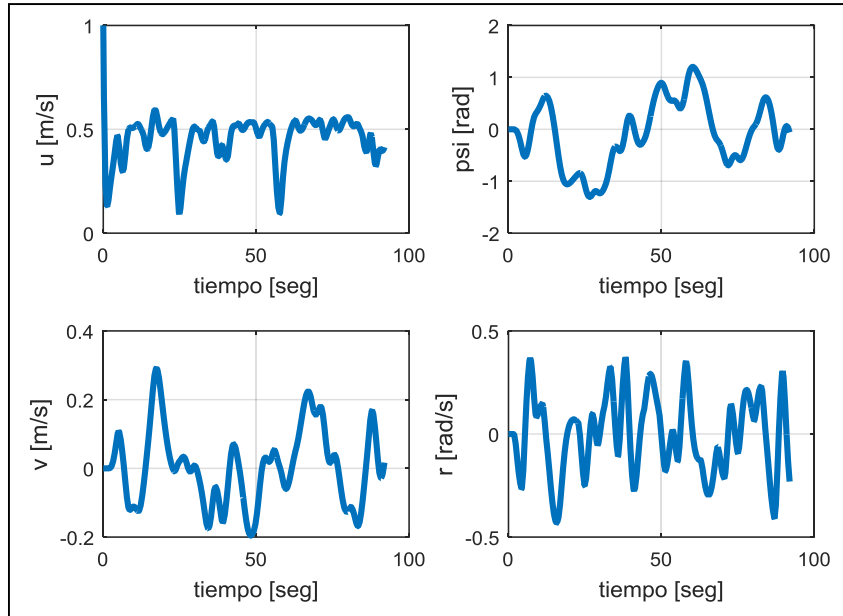
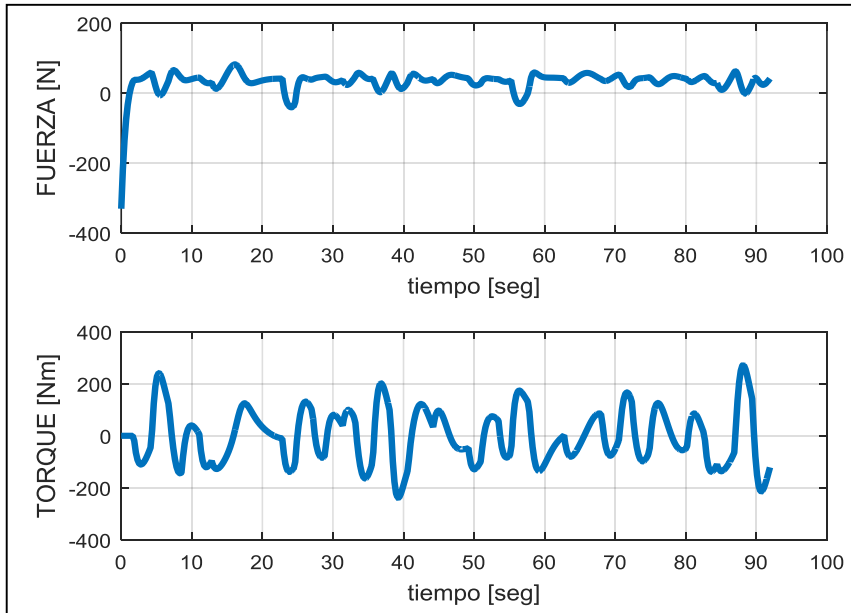


Figura 5.13. Mapa 5. Fuente: propia.



**Figura 5.14.** Velocidades y posición angular para el mapa 5. Fuente: propia.



**Figura 5.15.** Fuerza y torque para el mapa 5. Fuente: propia.

Los parámetros que resolvieron este mapa se muestran en la Tabla 5.5. Se observa que la población de hormigas y la cantidad de iteraciones ha aumentado considerablemente. De todas las pruebas realizadas, esta es la que más recursos y tiempo consume. El algoritmo ACO demoró 12.98 segundos.

**Tabla 5.5**

Población de hormigas	120
Factor de reducción de feromona	0.12
Ganancia de feromona	500
Exponente de feromona	2
Iteraciones realizadas	200
Ganancia de la brújula	10

La última categoría de pruebas se realizó de tal forma que se aumentó la cantidad de obstáculos de manera no-uniforme. Esto significa que se añadieron obstáculos de distinta forma, tamaño y orientación. Este tipo de pruebas se asemeja más a una aplicación real, en la cual los obstáculos encontrados son aleatorios en cuanto a posición y forma. Asimismo, se cambió la ubicación del punto final al cual debe llegar el barco. Se realizaron dos pruebas de este tipo; sin embargo, ambas pruebas son resueltas por el mismo conjunto de parámetros. La Figura 5.16 muestra la primera prueba de este tipo. Las Figuras 5.17 y 5.18 muestran las gráficas correspondientes a las velocidades, posición angular, fuerza y torque, en función del tiempo, para esta prueba.

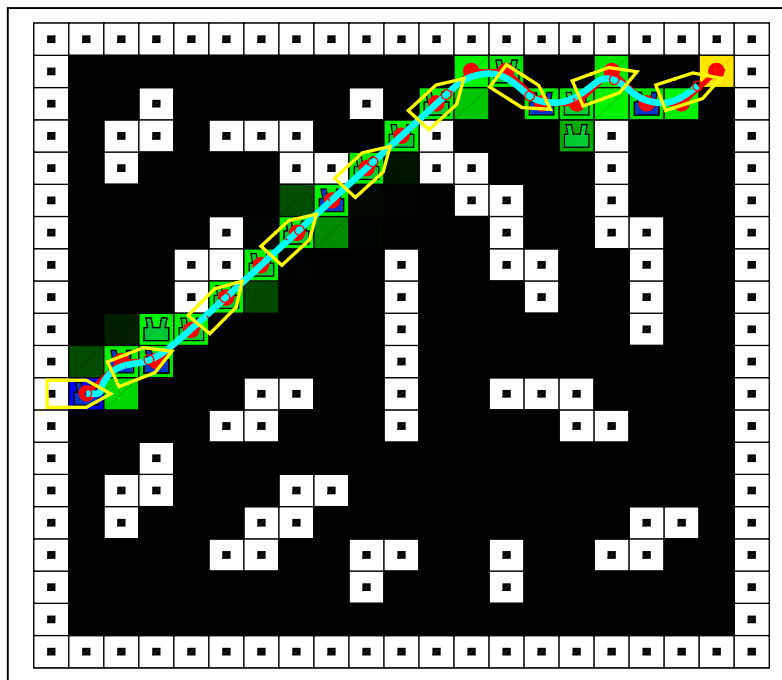


Figura 5.16. Mapa 6. Fuente: propia.

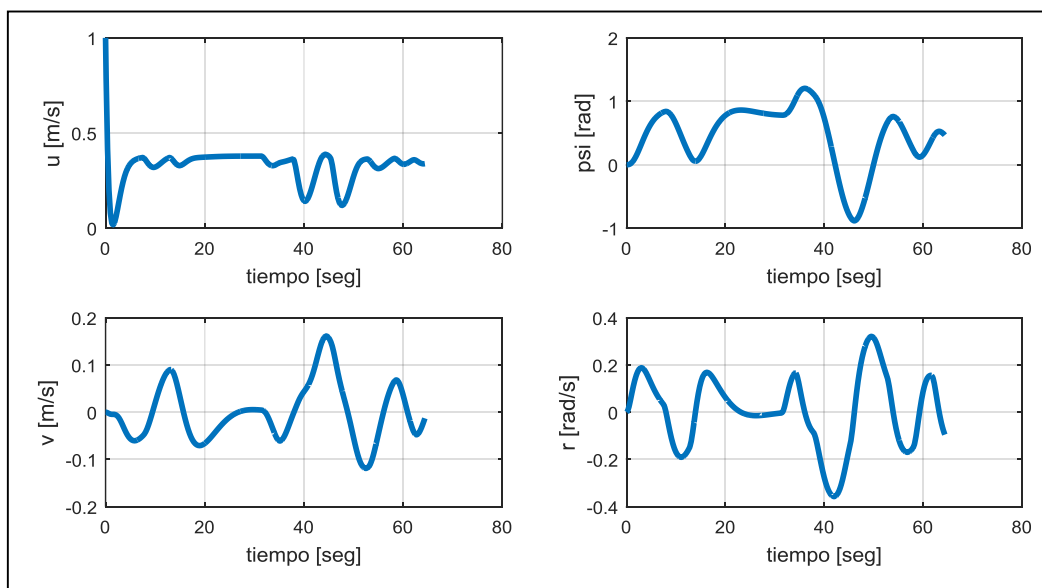
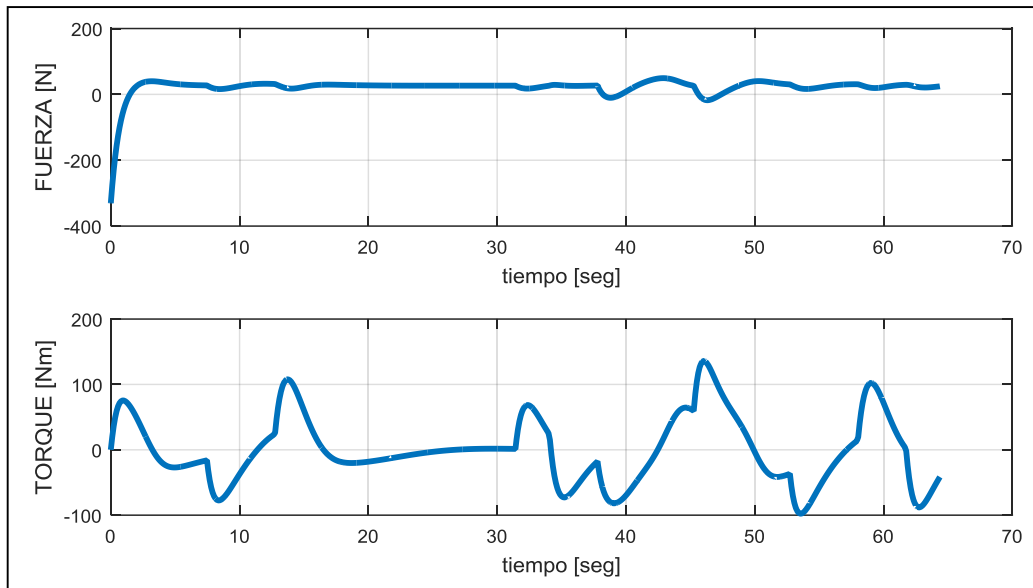
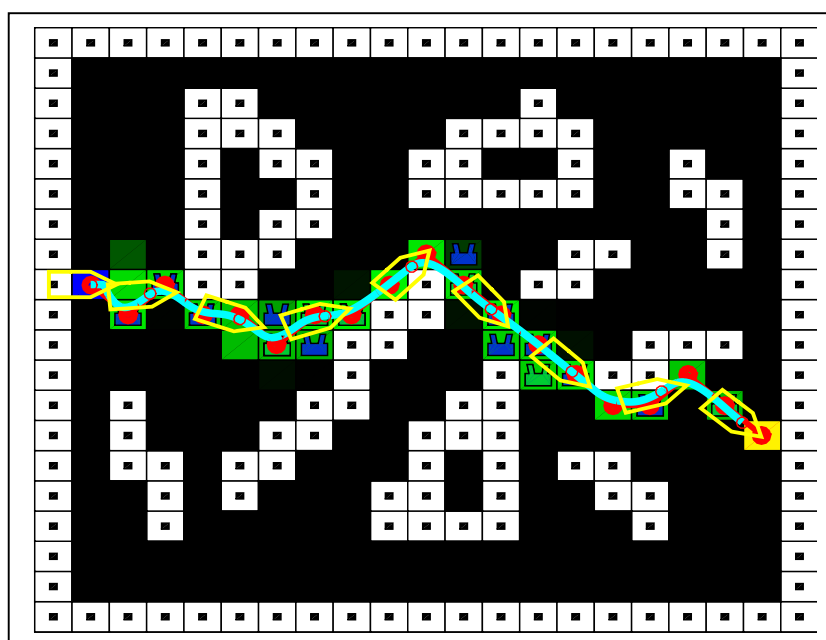


Figura 5.17. Velocidades y posición angular para el mapa 6. Fuente: propia.

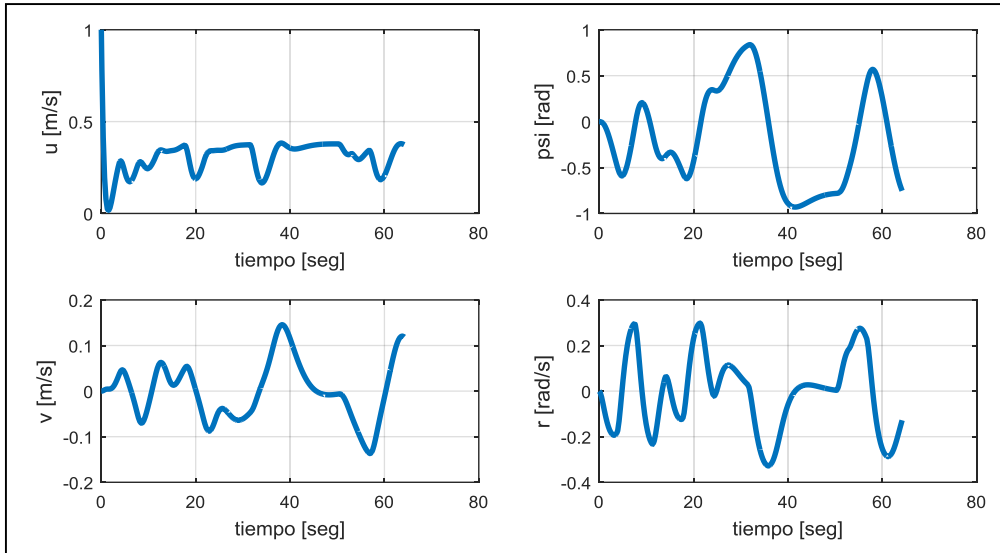


**Figura 5.18.** Fuerza y torque para el mapa 6. Fuente: propia.

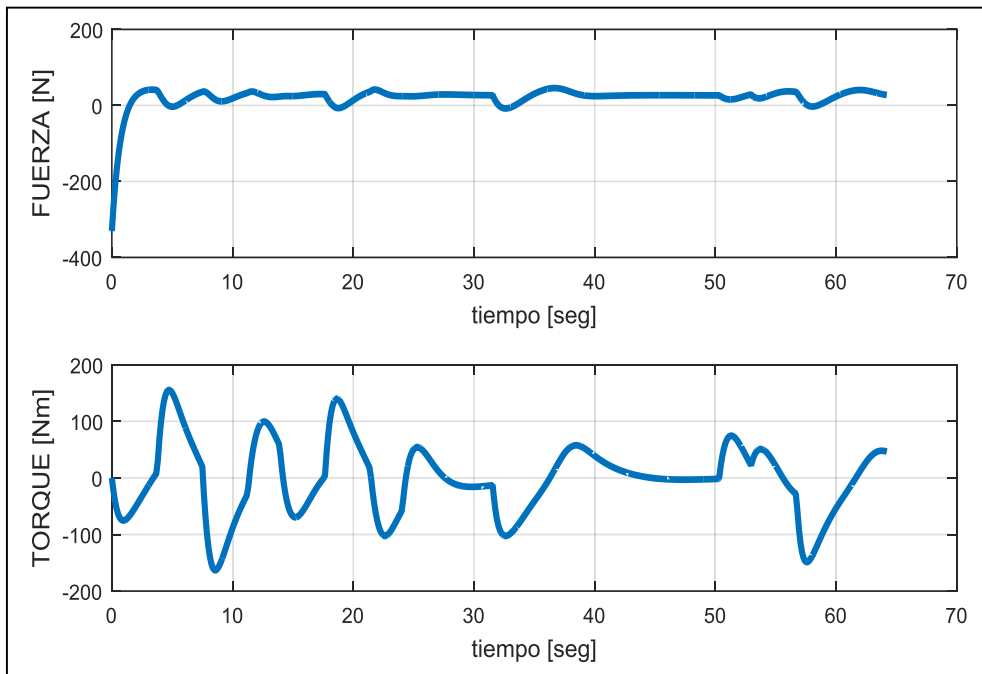
La segunda prueba, a diferencia de la primera prueba de este tipo, presenta obstáculos más grandes. Asimismo, la ubicación del punto destino es cambiada nuevamente. La solución para este mapa se muestra en la Figura 5.19. En ambos mapas de este tipo se puede observar que el controlador del barco robot logra maniobrar entre los obstáculos. Las gráficas de velocidades, posición angular, fuerza y torque se muestran en las Figuras 5.20 y 5.21. La Figura 5.20 muestra que la señal de fuerza sigue el patrón mostrado hasta ahora en todas las pruebas. Con esto podemos confirmar que el barco robot solo necesita una fuerza alta al inicio de la simulación; el resto de la trayectoria la fuerza se mantiene casi constante o con pequeñas oscilaciones. Los obstáculos de este mapa podrían interpretarse más fácilmente como contornos de seguridad alrededor de obstáculos reales.



**Figura 5.19.** Mapa 7. Fuente: propia.



**Figura 5.20.** Velocidades y posición angular para el mapa 7. Fuente: propia.



**Figura 5.21.** Fuerza y torque para el mapa 7. Fuente: propia.

Los parámetros que logran resolver ambos mapas (Figura 5.16 y 5.19) se resumen en la tabla 5.6. Se pueden observar cambios esperables respecto al mapa base. Sin embargo, hay una variación en los parámetros que no se había dado en las anteriores pruebas: la ganancia de la brújula ha disminuido.

**Tabla 5.6**

Población de hormigas	30
Factor de reducción de feromona	0.05
Ganancia de feromona	100
Exponente de feromona	1.5
Iteraciones realizadas	200
Ganancia de la brújula	2

Esta variación se puede explicar debido a que la presencia de múltiples obstáculos ofrece una gran cantidad de posibles caminos candidatos a ser el camino más corto. Ante tantas posibles combinaciones de caminos, es necesario que el algoritmo favorezca más la exploración de las hormigas en vez de guiarlas directamente hacia donde se encuentra el punto destino. Al disminuir la ganancia de la brújula las hormigas tienen más probabilidades de elegir caminos que no necesariamente apuntan en dirección hacia el punto destino. Este comportamiento es clave para resolver mapas de este tipo. El algoritmo ACO demoró 3.49 segundos en ejecutarse para el mapa de la Figura 5.16 y 3.51 segundos para el mapa de la Figura 5.19.

La Tabla 5.7 resume los tiempos (en segundos) que se tomó el algoritmo ACO y el algoritmo de control en ejecutarse. Se debe tener en cuenta que estos tiempos corresponden a la ejecución de estos algoritmos en el entorno MatLab; además, se utilizó una laptop con procesador tipo core i5 y con 4GB de memoria RAM para la ejecución de los mismos.

**Tabla 5.7.** Tiempos usados por el algoritmo ACO y el algoritmo de control. Todos los tiempos mostrados en esta tabla son medidos en segundos.

Tipo de prueba	(Mapa base)	Incremento uniforme de obstáculos		Incremento de resolución del mapa		Incremento no-uniforme de obstáculos	
		2	3	4	5	6	7
<b>Mapa</b>	1	2	3	4	5	6	7
<b>Tiempo usado para el algoritmo ACO</b>	1.71	2.38	4.51	3.63	12.98	3.49	3.51
<b>Tiempo usado para el algoritmo del controlador</b>	5.92	6.99	7.86	7.51	10.86	7.73	7.62

## CONCLUSIONES

Se utilizó un modelo dinámico para un robot marítimo basado en investigaciones existentes. Este modelo muestra que el barco robot es un sistema sub-actuado. El modelo usado se obtuvo en base a análisis cinemático y dinámico del barco.

Se diseñó un sistema de planeamiento de trayectoria basado en optimización por colonia de hormigas para determinar el camino más corto a seguir por un robot marítimo de superficie. El entorno donde se desplaza el robot puede ser representado por un espacio de áreas transitables y no transitables en el cual se presentan obstáculos estáticos. El modelo usado para controlar el barco fue obtenido de Fossen (2011). La estrategia para linealizar el modelo de este barco sub-actuado se obtuvo de Fahimi (2009). El control del robot se realizó mediante un controlador óptimo.

Se puso a prueba el funcionamiento de ambos sistemas, planeamiento de trayectoria y control de barco, utilizando 6 mapas con características distintas y un mapa base para las comparaciones. Se realizaron tres tipos distintos de pruebas de tal forma que se observe la variación de los parámetros de simulación en relación con las distintas características de los mapas propuestos.

Se logró encontrar el camino más corto para todos mapas propuestos. En todos los casos el ACO fue capaz de determinar el camino más corto cuando se le asignó los parámetros adecuados para su funcionamiento. La trayectoria hallada podría mejorar si se aplica un algoritmo de suavización de curva o si se aumenta la resolución del mapa; es decir, disminuir el tamaño de las divisiones del mapa. Sin embargo dicho aumento de resolución traería consigo un mayor tiempo de ejecución del algoritmo y una mayor demanda de recursos (por ejemplo: RAM y tiempo de CPU). En las pruebas de aumento de resolución del mapa se pudo observar, en la Tabla 5.7, que un aumento por un factor de 16 en la resolución produce el tiempo de convergencia más largo de todas las pruebas realizadas (más de 7 veces el tiempo del mapa base). Por este motivo, se recomienda usar un algoritmo de suavización de curva en vez de aumentar la resolución del mapa.

Una vez determinada la trayectoria a seguir, el sistema de control fue capaz de mover al barco robot a lo largo de la trayectoria hallada. Para esto se realizaron pruebas con el controlador para una trayectoria circular de tal forma que se puedan hallar los valores adecuados de la matriz de pesos  $Q$  (propia del control óptimo) que permitió obtener el comportamiento deseado del barco robot. Con los valores de la matriz  $Q$  adecuados, se pudo seguir las trayectorias halladas por el ACO cuando se ejecutaron ambos algoritmos en conjunto para

guiar al barco robot dentro de un entorno con obstáculos. Se tomó en cuenta que los obstáculos pueden ser representaciones de un contorno de seguridad dibujado alrededor de un obstáculo real; de esta forma, se tiene un margen de error que el barco puede utilizar para evitar colisiones.

El tiempo utilizado por el algoritmo de control es el mismo para todas las pruebas, excepto en el caso de aumento de resolución por un factor de 16. Se puede observar en la Tabla 5.7 que el tiempo de ejecución del controlador óptimo es 1.5 segundos más en los mapas 2, 3, 4, 6 y 7 que en el mapa base. Para el mapa 5, el tiempo de ejecución aumenta en más de 3 segundos, por lo cual, nuevamente, se recomienda no aumentar la resolución del mapa.

En los gráficos de fuerza y torque (Figuras 5.3, 5.6, 5.9, 5.12, 5.15, 5.18 y 5.21) se puede observar un pico al inicio de la simulación en la señal de fuerza. Este pico es un factor de diseño al momento de implementar este sistema en un barco robot real. Para este trabajo, se tomó el ejemplo proporcionado por Fahimi (2009); sin embargo, para futuros trabajos que incluyan implementación en un barco real, se debe tomar en cuenta este valor de fuerza al momento de seleccionar los motores del barco robot.

## BIBLIOGRAFÍA

- [1] L. Weng y D.Y. Song, "Path planning and path tracking control of unmanned ground vehicles (UGVs)" en *Proceedings of the Thirty-Seventh Southeastern Symposium on System Theory* , EEUU, 2005.
- [2] Ji-wung, R. Curry, G. Elkaim, "Path Planning Based on Bézier Curve for Autonomous Ground Vehicles", en *Special Edition of the World Congress on Engineering and Computer Science 2008, WCECS '08. Advances in Electrical and Electronics Engineering*, San Francisco, 2008, pp 158-166.
- [3] H. J. Woo, S. B. Park y J. H. Kim, "Research of the optimal path planning methods for unmanned ground vehicle in DARPA Urban Challenge", en *International Conference on Control, Automation and Systems*, Seoul, 2008, pp 586-589.
- [4] J. Yoon, C.D. Crane, "Path planning for Unmanned Ground Vehicle in urban parking area", en *11th International Conference on Control, Automation and Systems*, Gyeonggi-do, 2011, pp 887-892.
- [5] Y.M. Han, J.B. Jeong y J.H. Kim, "Quadtree based path planning for Unmanned Ground Vehicle in unknown environments", en *12th International Conference on Control, Automation and Systems*, JeJu Island, 2012, pp 992-997.
- [6] N. Ahmad , S.I.S. Bajwa, S. Ahmad y S.L. Khan, "Autonomous Unmanned Surface Vehicle a microcontroller based approach", en *International Conference on Electric Information and Control Engineering*, Wuhan, 2011, pp 1193-1196.
- [7] C.Z. Pan, X.Z. Lai, S.X. Yang y M. Wu, "Backstepping neurodynamics based position-tracking control of underactuated autonomous surface vehicles", en *25th Chinese Control and Decision Conference*, Guiyang, 2013, pp 2845-2850 .
- [8] K.D. Do, J. Pan, "Global robust adaptive path following of underactuated ships", en *Automatica* vol. 42, no. 10, pp 1713–1722, oct. 2006.
- [9] K.D. Do, "Practical control of underactuated ships", en *Ocean Engineering*, vol 37, no. 13, Sep. 2010, pp 1111–1119.
- [10] Z. Guoqing, Y. Lijun, M. Xiangzhen y Z. Lijun, "Tracking control of underactuated ship based on partial state feedback scheme", en *International Conference on Complex Medical Engineering*, Harbin Heilongjiang, 2011, pp 678-683.

- [11] F. Gholami, M.J. Mahjoob, “An Investigation of Parameters in Ant Colony Optimization for a Path Optimization Algorithm”, en *International Conference on Mechatronics and Automation*, Harbin, 2007, pp 463-468.
- [12] D.H. Nguyen, J. C. Juang, “A refined ant colony algorithm for optimal path planning”, en *International Conference on System Science and Engineering*, Macau, 2011, pp 125–130.
- [13] S.H. Chia, K.L. Su, Jr.H. Guo y C.Y. Chung, “Ant Colony System Based Mobile Robot Path Planning”, en *Fourth International Conference on Genetic and Evolutionary Computing*, Shenzhen, 2010, pp 210–213.
- [14] Y.Z. Cong, S.G. Ponnambalam, “Mobile robot path planning using ant colony optimization”, en *International Conference on Advanced Intelligent Mechatronics*, Singapore, 2009, pp 851–856.
- [15] M. Dorigo y T. Stützle, *Ant Colony Optimization*, The MIT press, Cambridge, Massachusetts, 2004.
- [16] N. Pinter-Wollmana, A. Balaa, A. Merrella, J. Queiroloa, M. C. Stumpec, S. Holmesb, D. M. Gordon, “Harvester ants use interactions to regulate forager activation and availability”, en *Animal Behaviour*, Vol. 86, no. 1, Jul. 2013, pp 197–207.
- [17] S. Johnson, *Emergence*, SCRIBNER, New York, EEUU, 2001.
- [18] SNAME Hydrodynamics Subcommittee, “Nomenclature for Treating the Motion of a Submerged Body through a Fluid”, en *American Towing Tank Conference*, New York, 1950.
- [19] T. Fossen, *Marine Craft Hydrodynamics and Motion Control*, John Wiley & Sons Ltd, West Sussex, United Kingdom, 2011.
- [20] T. Pérez, *Ship Motion Control*, Springer-Verlag, Londres, UK, 2005.
- [21] F. Fahimi, *Autonomous Robots*, Springer Science+Business Media, New York, EEUU, 2009.
- [22] C. Petres, M.A. Romero-Ramirez, F. Plumet y B. Alessandrini, “Modeling and reactive navigation of an autonomous sailboat”, en *International Conference on Intelligent Robots and Systems*, San Francisco, CA, 2011, pp 3571-3576.
- [23] X. Xianbo y X. Guohua, “Suvery on control of underactuated ships: Point stabilization and path tracking”, en *31st Chinese Control Conference*, Hefei, 2012, pp 4312-4317.

- [24] I. Boussaïda, J. Lepagnotb y P. Siarry, “”, en *Information Sciences*, Vol. 237, Jul. 2013, pp 82–117.
- [25] L. Di Gaspero, A. Schaerf y T. Stützle, *Advances in Metaheuristics*, Springer Science+Business Media, New York, EEUU, 2013.
- [26] E. Alba, A. Nakib y P. Siarry, *Metaheuristics for Dynamic Optimization*, Springer-Verlag Berlin Heidelberg, 2013.
- [27] M. Birattari, *Tuning Metaheuristics*, Springer Berlin Heidelberg, 2009.
- [28] A. Lazarowska, “Ant Colony Optimization based Navigational Decision Support System”, en *Knowledge-Based and Intelligent Information & Engineering Systems 18th Annual Conference*, Poland, 2014, pp 1013–1022.
- [29] A. Lazarowska, “Parameters influence on the performance of an Ant algorithm for safe ship trajectory planning”, en *2nd International Conference on Cybernetics*, Gdynia, 2015, pp 140–145.
- [30] J. van Ast, R. Babuska y B. De Schutter, “Ant Colony Optimization for optimal control”, en *IEEE Congress on Evolutionary Computation*, Hong Kong, 2008, pp 2040–2046.
- [31] S. Xiaolu, Y. Yang y G. Bin, “Application of ant colony algorithm in optimal control for the under-actuated system pendubot”, en *27th Chinese Control Conference*, Kunming, 2008, pp 526-529.
- [32] N. Ganganath y C.T. Cheng, “A 2-Dimensional ACO-Based Path Planner for Off-Line Robot Path Planning”, en *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, Beijing, 2013, pp 302–307.
- [33] S. Cui, H. Wang y J. Li, “Mobile Robot Path Planning Using an Improved Ant Colony Algorithm in Uncertain Environments”, en *Third International Conference on Instrumentation, Measurement, Computer, Communication and Control*, Shenyang, 2013, pp 200–203.
- [34] B. Nagaraj y N. Muruganath, “A comparative study of PID controller tuning using GA, EP, PSO and ACO”, en *IEEE International Conference on Communication Control and Computing Technologies*, Ramanathapuram, 2010, pp 305–313.
- [35] M.Tomera, “Ant Colony Optimization Algorithm Applied to Ship Steering Control”, en *Knowledge-Based and Intelligent Information & Engineering Systems 18th Annual Conference*, Poland, 2014, pp 83–92.

- [36] M. Omar, M. Soliman, A.M. Abdel Ghany y F. Bendary, "Ant Colony Optimization based PID for single area load frequency control", en *Proceedings of International Conference on Modelling, Identification & Control*, Cairo, 2013, pp 119-123.
- [37] H.C. Lu y H.K Liu, "Ant colony fuzzy neural network controller for cruising vessel on river", en *Applied Ocean Research*, vol 42, Ago. 2013, pp 43–54.
- [38] S. Kumar y R. Bhatt, "Multi objective ant algorithm to parameter optimized of PID controller and compare to genetic algorithm", en *IEEE International Conference on Computational Intelligence and Computing Research*, Enathi, 2013, pp 1-4.
- [39] M. Ünal, A. Ak, V. Topuz y H. Erdal, *Optimization of PID Controllers Using Ant Colony and Genetic Algorithms*, Springer-Verlag Berlin Heidelberg, 2013.
- [40] R.A. Hanifah, S.F. Toha y S. Ahmad, "PID-Ant Colony Optimization (ACO) control for Electric Power Assist Steering system for electric vehicle", en *IEEE International Conference on Smart Instrumentation, Measurement and Applications*, Kuala Lumpur, 2013, pp 1-5.
- [41] C. Chengzhi, G. Xiaofeng y L. Yang, "Research on Ant Colony Neural Network PID Controller and Application", en *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, Qingdao, 2007, pp 253-258.
- [42] S. Ghosal, R. Darbar, B. Neogi, A. Das y D. N. Tibarewala, "Application of Swarm Intelligence Computation Techniques in PID Controller Tuning: A Review", en *Proceedings of the International Conference on Information Systems Design and Intelligent Applications*, India, 2012, pp 195-208.
- [43] W. Zhao, X. Cai y Y. Lan , "A New Ant Colony Algorithm for Solving Traveling Salesman Problem", en *International Conference on Computer Science and Electronics Engineering*, Hangzhou, 2012, pp 530-533.
- [44] J.B. Escario, J.F. Jimenez y J. M. Giron-Sierra, "Ant Colony Extended: Experiments on the Travelling Salesman Problem", en *Expert Systems with Applications*, Vol. 42, no. 1, ene. 2015, pp 390–410.
- [45] Pragya, M. Dutta y Pratyush, "TSP Solution Using Dimensional Ant Colony Optimization", en *Fifth International Conference on Advanced Computing & Communication Technologies*, Haryana, 2015, pp 506-512.

- [46] M. Dorigo, M. Birattari, S. Garnier, H. Hamann, M. Montes de Oca, C. Solnon y T. Stützle, Swarm Intelligence, en 9th International Conference, Brussels, Belgium, Sept. 2014.
- [47] M. Dorigo y C. Blum, “Ant colony optimization theory: A survey”, en *Theoretical Computer Science*, Vol. 344, no. 2–3, Nov. 2005, pp 243–278.
- [48] M. Dorigo y K. Socha, *An Introduction to ant colony optimization*, IRIDIA Technical Report Series, 2006.
- [49] Z. Juan-ping, L. Jin-gang ; X.W. Gao y C. Ying-qiao, “Research of path planning for mobile robot based on improved ant colony optimization algorithm”, en 2nd International Conference on Advanced Computer Control (vol 2), Shenyang, 2010, pp 241-245.
- [50] S.N. Sivanandam y S.N. Deepa, “Introduction to Particle Swarm Optimization and Ant Colony Optimization”, en *Introduction to Genetic Algorithms*, Springer-Verlag Berlin Heidelberg, 2008, pp 403-424.
- [51] M.P. Song y G.C. Gu, “Research on particle swarm optimization: a review”, en *International Conference on Machine Learning and Cybernetics (Vol 4)*, Shanghai, 2004, pp 2236-2241.
- [52] K.D. Do y J. Pan, *Control of Ships and Underwater Vehicles, Design for Underactuated and Nonlinear Marine Systems*, Springer London, 2009.
- [53] L. Yang, J. Qi, J. Xiao y X. Yong, “A literature review of UAV 3D path planning”, en *11th World Congress on Intelligent Control and Automation*, Shenyang, 2014, pp 2376–2381.
- [54] Z. Wang, L. Liu, T. Long y C. Yu, “Enhanced sparse A\* search for UAV path planning using dubins path estimation”, en *33rd Chinese Control Conference*, Nanjing, 2014, pp 738-742.
- [55] Y.H. Qu, Q. Pan, y J.G. Yan, “Flight path planning of UAV based on heuristically search and genetic algorithms”, en *31st Annual Conference of IEEE Industrial Electronics Society*, 2005.
- [56] C. Shi, M. Zhang y J. Peng, “Harmonic Potential Field Method for Autonomous Ship Navigation”, en *7th International Conference on ITS Telecommunications*, Sophia Antipolis, 2007, pp 1-6.
- [57] S. Gai-ping, G. Hai-wen, W. Dezhi y W. Jiang-hua, “A Dynamic Ant Colony Optimization Algorithm for the Ad Hoc Network Routing”, en *Fourth International Conference on Genetic and Evolutionary Computing*, Shenzhen, 2010, pp 358 – 361.

- [58] N.N. Li , Z. Zhao, J.H. Gu y B.Y. Liu, “Ant colony optimization algorithm for expert assignment problem”, en *International Conference on Machine Learning and Cybernetics (Vol 2)*, Kunming, 2008, pp 660 – 664.
- [59] R.M. Chen, S.T. Lo, C.L. Wu, T.H. Lin, “An effective ant colony optimization-based algorithm for flow shop scheduling”, en *IEEE Conference on Soft Computing in Industrial Applications*, Muroran, 2008, pp 101-106.
- [60] T. Nishi, Y. Isoya y M. Inuiguchi, “Column generation for sequence dependent flowshop scheduling to minimize the total weighted tardiness”, en *IEEE International Conference on Systems, Man, and Cybernetics*, Anchorage, AK, 2011, pp 1187–1192.
- [61] K.M. Sim y W.H. Sun, “Multiple ant-colony optimization for network routing”, en *First International Symposium on Cyber Worlds*, 2002, pp 277-281.
- [62] Y. Fu y W. Luo, “Optimal task assignment for serial-parallel hybrid robots cooperation via ant colony optimization”, en *9th International Conference on Electronic Measurement & Instruments*, Beijing, pp 3-795 - 3-800.