



POLITÉCNICA



UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y DISEÑO INDUSTRIAL

Grado en Ingeniería Electrónica Industrial y Automática

TRABAJO FIN DE GRADO

**Planificación y seguimiento de trayectorias para formación
de robots submarinos**

Autor: Jorge Ortega Mateo

Tutor:

David Álvarez Sánchez

**Departamento de Ingeniería Eléctrica, Electrónica, Automática y
Física Aplicada**

Madrid, 02, 2023

AGRADECIMIENTOS

Quiero dar las gracias en primer lugar a mi madre, a mi padre y a mi hermana, por aguantarme y apoyarme siempre en mi vida.

Dar gracias también a mi tutor, David Álvarez, me he sentido muy arropado durante el transcurso de este trabajo fin de grado y siempre he sentido que ha buscado un hueco para mí, para poder ayudarme cuando tenía dudas.

Finalmente me gustaría dar las gracias a todos mis amigos, por apoyarme también y ayudarme a evadirme un poco de la realidad cuando las cosas no salían bien.

RESUMEN

Este trabajo se centra en el estudio de seguimiento de trayectoria y control de formaciones de un vehículo autónomo submarino. Al carecer de un prototipo y de una zona de pruebas apta, se realizarán una serie de simulaciones con el fin de obtener los objetivos del proyecto. Concretamente, estos objetivos son realizar un algoritmo capaz de seguir una trayectoria deseada y realizar que un numero de robots mantengan una formación mientras se recorre dicha trayectoria.

El trabajo se divide en cuatro bloques. El primero, corresponde a la creación de un entorno de simulación y un algoritmo de barrido. La idea es crear un espacio de tres dimensiones donde se pueda diferenciar lo que es un obstáculo y una zona libre para que el robot pueda desempeñar su movimiento. A su vez, la trayectoria debe estar formado por una serie sucesiva de puntos para que sea finita y pueda ser seguida. El segundo bloque trata sobre la cinemática de un robot submarino. Determinar la cinemática de un robot submarino permite simular los movimientos de este ajustándose lo máximo posible a la realidad. Además, se pretende crear un modelo de vehículo submarino para que la cinemática no se aplique a un solo punto del espacio. El tercer bloque corresponde con el desarrollo del algoritmo para realizar el seguimiento de trayectorias. Dicho algoritmo debe ser capaz de realizar el seguimiento en tres dimensiones y ajustarse con precisión a dicho recorrido. El cuarto bloque corresponde a la estrategia de control de formaciones que los robots van a utilizar. La formación debe mantenerse a lo largo del movimiento sin sufrir grandes deformaciones.

ABSTRACT

This work is focused on the study of path following and formation control of an autonomous underwater vehicle. Lacking a prototype and a suitable test area, a series of simulations will be performed to obtain the objectives of the project. Specifically, the objectives of the project are two, which are the following. Firstly, to create an algorithm capable of following a desired trajectory. Secondly, to make several robots maintain a formation while traveling along the trajectory.

The work is divided into four blocks. The first corresponds to the creation of a simulation environment and a scanning algorithm. The idea is to create a three-dimensional space where it is possible to differentiate between what is an obstacle and a free zone so that the robot can carry out its movement. In turn, the trajectory must be formed by a successive series of points so that it is finite and can be followed. The second block deals with the kinematics of an underwater robot. Determining the kinematics of an underwater robot allows to create its movements, adjusting it as much as possible to reality. In addition, it is intended to create a model of an underwater vehicle so that the kinematics is not applied to a singular point in space. The third block corresponds to the development of the algorithm to track trajectories. Said algorithm must be capable of tracking in three dimensions and adjust precisely to the path. The fourth block corresponds to the formation control strategy that the robots will use. The formation must be maintained throughout the movement without suffering large deformations.

ÍNDICE

AGRADECIMIENTOS.....	I
RESUMEN.....	II
ABSTRACT	III
ÍNDICE.....	I
ÍNDICE DE FIGURAS.....	III
GLOSARIO.....	VI
LISTA DE ABREVIATURAS	VI
CAPÍTULO 1. INTRODUCCIÓN.....	7
1.1. CONTEXTUALIZACIÓN	7
1.2. OBJETIVOS.....	7
1.3. ESTRUCTURA DEL DOCUMENTO	8
CAPÍTULO 2. ESTADO DEL ARTE.....	9
2.1. MODELOS DE AUVs	10
2.1.1. <i>Girona 500</i>	10
2.1.2. <i>Sparus II</i>	11
2.1.3. <i>REMUS 100</i>	12
2.2. CONTROL GEOMÉTRICO DE SEGUIMIENTO DE TRAYECTORIA	13
2.2.1. <i>Chasing the Carrot</i>	13
2.2.2. <i>Persecución pura</i>	14
2.2.3. <i>Persecución vectorial</i>	16
2.3. MÉTODOS DE BARRIDO DE EXTENSIONES	17
2.3.1. <i>Algoritmo de la cortadora de césped</i>	18
2.3.2. <i>Método clásico de descomposición celular exacta</i>	19
2.3.3. <i>Descomposición celular basada en Morse</i>	20
2.4. TIPOLOGÍA Y CONTROL DE FORMACIONES	21

2.4.1. Tipos de formaciones	21
2.4.2. Estrategias de control de formaciones.....	23
CAPÍTULO 3. SOLUCIÓN PLANTEADA.....	26
3.1. ESTRATEGIA DE BARRIDO	27
3.1.1. Entorno de simulación	27
3.1.2. Barrido de zona	28
3.2. CINEMÁTICA DEL ROBOT SUBMARINO.....	29
3.2.1. Marcos de referencia	29
3.2.2. Matriz de rotación	31
3.2.3. Transformación de velocidades lineales y angulares	32
3.3. PERSECUCIÓN PURA.....	34
3.3.1. Parametrización y aproximación de la trayectoria	34
3.3.2. Parámetro de punto de mira.....	35
3.3.3. Cálculo del punto objetivo	36
3.3.4. Error de tramo cruzado.....	40
3.4. CONTROL DE FORMACIONES	41
3.4.1. Diseño de la formación.....	41
3.4.2. Cinemática diferencial.....	42
CAPÍTULO 4. RESULTADOS.....	45
4.1. AJUSTE DE VALORES CINEMÁTICOS.....	46
4.2. PERSECUCIÓN PURA	49
4.2.1. Caso 1: Valor de distancia LA 2.	50
4.2.2. Caso 2: Valor de distancia LA 10.	55
4.2.3. Caso 3: Valor de distancia LA 21.	62
4.3. CONTROL DE FORMACIONES	66
4.3.1. Curva	66
4.3.2. Espiral.....	68
4.3.3. Barrido	70
CAPÍTULO 5. CONCLUSIONES.....	73
5.1. CONCLUSIONES.....	73
5.2. RESUMEN DE OBJETIVOS	74
5.3. PERSPECTIVAS FUTURAS	74
BIBLIOGRAFÍA.....	76
ANEXO A. REPOSITORIO DE GITHUB.....	79
A.1. CÓDIGO DEL MODELO DE SUBMARINO Y LAS SIMULACIONES	79

ÍNDICE DE FIGURAS

Figura 1: Vista Girona 500 [2]	Figura 2: Sensores Girona 500 [2]	11
Figura 3: Vista Sparus II [4]	Figura 4: Propulsores y aletas 500 [4]	12
Figura 5: Representación gráfica CC [7]		14
Figura 6: Representación gráfica PP [8]		15
Figura 7: Representación Gráfica VP [7]		16
Figura 8: Movimiento en zigzag [11]		19
Figura 9: Representación gráfica de un entorno dividido en celdas [11]		20
Figura 10: Representación gráfica de la descomposición celular basada en Morse [11]		20
Figura 11: : Formación en columna [13]		21
Figura 12: Formación en línea [13]		22
Figura 13: Formación en V [13]		22
Figura 14: Formación en diamante [13]		22
Figura 15: Estructura distancia-distancia [14]		23
Figura 16: Estructura distancia-ángulo [14]		23
Figura 17: Estructura potencial artificial [14]		24
Figura 18: Entorno submarino propuesto		27
Figura 19: Representación gráfica de la lógica empleada para obtener la altura deseada en la trayectoria		28
Figura 20: Representación de los marcos de referencia [16]		30
Figura 21: Rotación sobre los ejes [17]		32
Figura 22: Parámetro LA con un valor de distancia muy pequeña [18]		35
Figura 23: Parámetro LA con un valor de distancia excesivamente grande [18]		36

Figura 24: Representación cálculo de la altura.....	38
Figura 25: Criterio de selección de raíces	39
Figura 26: Representación gráfica EPC [8].....	40
Figura 27: Representación gráfica de la formación	41
Figura 28: Relación de la matriz Jacobiana	43
Figura 29: Simulación realiza para obtener las dimensiones de la simulación.	46
Figura 30: Simulación de giro de 90° y error de posición respectivo	47
Figura 31: Simulación de giro de 60° y error de posición respectivo	47
Figura 32: Simulación de giro de 45° y error de posición respectivo	48
Figura 33: Simulación de giro de 30° y error de posición respectivo	48
Figura 34: Simulación curva 2D con parámetro LA de 2.....	50
Figura 35: Error de posición respecto a la trayectoria en curva plana.....	50
Figura 36: Error de orientación respecto al punto objetivo.....	51
Figura 37: Simulación espiral con parámetro LA de 2.	52
Figura 38: Error de posición respecto a la trayectoria en espiral.....	52
Figura 39: Error de orientación respecto al punto objetivo.....	53
Figura 40: Simulación de barrido con parámetro LA de 2.	54
Figura 41: Simulación de curva 2D con parámetro LA de 10.	55
Figura 42: Error de posición respecto a la trayectoria en una curva plana.	55
Figura 43: Error de orientación respecto al punto objetivo.....	56
Figura 44: Simulación de espiral con parámetro LA de 10.....	57
Figura 45: Error de posición respecto a la trayectoria en espiral.....	57
Figura 46: Error de orientación respecto al punto objetivo.....	58
Figura 47: Simulación de barrido con parámetro LA de 10.	59
Figura 48: Simulación de barrido con parámetro LA de 10 aplicando un valor proporcional.....	59

Figura 49: Error de posición respecto a la trayectoria de barrido.	60
Figura 50: Error de orientación respecto al punto objetivo.....	60
Figura 51: Simulación de curva 2D con parámetro LA de 21.	62
Figura 52: Simulación de espiral con parámetro LA de 21.	62
Figura 53: Error de posición respecto a la trayectoria en espiral.	63
Figura 54: Error de orientación respecto al punto objetivo.....	63
Figura 55: Simulación de barrido con parámetro LA de 21.	64
Figura 56: Error de posición respecto a la trayectoria en espiral.	65
Figura 57: Error de orientación respecto al punto objetivo.....	65
Figura 58: Simulación de curva 2D con robots seguidores.	66
Figura 59: Posición de los robots seguidores con respecto al líder.	67
Figura 60: Distancia mantenida entre los robots seguidores en la trayectoria..	67
Figura 61: Simulación de espiral con robots seguidores.....	68
Figura 62: Posición de los robots seguidores con respecto al líder.	69
Figura 63: Distancia mantenida entre los robots seguidores en la trayectoria..	69
Figura 64: Simulación de barrido con robots seguidores.	70
Figura 65: Posición de los robots seguidores con respecto al líder.	71
Figura 66: Distancia mantenida entre los robots seguidores en la trayectoria..	71

GLOSARIO

LISTA DE ABREVIATURAS

AUV	Autonomous underwater vehicle
GDL	Grados de libertad
CST	Control de seguimiento de trayectoria
LA	Look Ahead
CC	Chasing the carrot
PP	Pure pursuit
VP	Vector Pursuit
EPC	Error de pista cruzada (Crosstrack error)

Capítulo 1. INTRODUCCIÓN

1.1. CONTEXTUALIZACIÓN

Durante las últimas décadas, la investigación del suelo marino se ha vuelto un área de interés científico creciente gracias a los robots submarinos. Son muchas las tareas que estos pueden realizar en el fondo marino como las de inspección de suelos, rescate, recogida de muestras, etc. Además, ha de considerarse que estos pueden acceder tanto a zonas como a profundidades donde el ser humano sería incapaz de acceder.

Este trabajo de fin de grado se desarrolla sobre el diseño cinemático de un robot submarino y su motivación radica en conseguir realizar un algoritmo de seguimiento de trayectoria capaz de realizar un barrido del suelo marino. A su vez, se persigue que diversos robots del mismo tipo sigan dicha trayectoria manteniendo una formación en el espacio empleando una estrategia de control.

1.2. OBJETIVOS

El objetivo de este trabajo consiste en realizar una simulación de un entorno submarino donde unos robots sean capaces de realizar el barrido de un área limitada manteniendo una formación constante.

Los objetivos del proyecto se han dividido en los siguientes puntos:

Capítulo 1

- Diseño de un entorno de simulación y de un algoritmo capaz de realizar la trayectoria de un barrido en un área delimitada.
- Diseño e implementación de un modelo cinemático de un robot submarino.
- Implementación de un algoritmo de control y seguimiento de trayectoria.
- Implementación y control de una formación de robots submarinos.

1.3. ESTRUCTURA DEL DOCUMENTO

- Capítulo 1: Breve introducción del proyecto, motivación y objetivos previstos.
- Capítulo 2: Desarrollo del estado del arte.
- Capítulo 3: Planteamientos teóricos desarrollados en el trabajo.
- Capítulo 4: Resultados.
- Capítulo 5: Conclusiones y líneas futuras.

Capítulo 2. ESTADO DEL ARTE

El afán por descubrir, de explorar nuevas zonas inexploradas, de llegar donde nunca se ha llegado, de alcanzar lo inalcanzable... ha formado siempre parte de la naturaleza y comportamiento humano. En 1492, a pesar de las creencias de haber descubierto toda la "tierra conocida", se descubrió América. En 1820 se descubrió la Antártida, siendo un semillero de exploración científica y un lugar de aventuras y riesgos. Siempre mirando a las estrellas, el hombre, en 1969 fue capaz de pisar la luna evolucionando así hasta 2012 donde el primer Rover, Curiosity, aterrizó en marte. Este afán por descubrir se ha extendido por tierra, aire y espacio, pero ¿qué hay sumergido debajo de nuestras costas?

Esta pregunta ha permanecido como una incógnita hasta los últimos 30 años donde, gracias a los avances científicos y tecnológicos, se ha podido explorar el entorno marino con mayor precisión. La robótica submarina ha permitido este avance para llegar donde el ser humano no ha podido llegar permitiendo la exploración de este cercano pero nuevo mundo.

Se guía el capítulo hasta un caso concreto: Planificación y seguimiento de trayectorias para formación de robots submarinos. Se habla de los distintos modelos de robots submarinos adaptados para misiones submarinas, apartado 2.1, se continúa con la importancia del control y del seguimiento de trayectorias que estos robot pueden seguir, apartado 2.2 y las distintas estrategias empleadas para el barrido de zonas, apartado 2.3. Se finaliza con las ventajas de la cooperación y sincronización de estos robots a la hora de llevar a cabo misiones, apartado 2.4.

2.1. MODELOS DE AUVs

Un vehículo autónomo submarino (AUV, en inglés, *Autonomous underwater vehicle*) es un vehículo submarino no tripulado capaz de realizar labores de manera autónoma sin intervención humana. Los AUVs nacen con la necesidad militar de detectar y desactivar minas submarinas que habían quedado abandonadas en zonas de conflicto bélico.

En la actualidad, los AUVs han evolucionado convirtiéndose en vehículos imprescindibles para obtener información marina como investigación y mapeado de relieve, determinación física, química o biológica del agua, grabación del comportamiento especies submarinas, misiones de rescate, inspección de tuberías submarinas... etc. [1]

Existen numerosos modelos de estos vehículos, cada uno con distintos diseños y adaptados a diferentes misiones. A continuación, se expondrán dos modelos creados en la universidad de Girona resumiendo las características y aplicaciones de estos AUVs al igual que del modelo REMUS.

2.1.1. GIRONA 500

Girona 500 [2] [3] es un AUV desarrollado en el laboratorio de la universidad de Girona, España, creado con un diseño flexible para poder abordar diferentes misiones tales como de grabación submarina hasta de intervención y rescate.

El vehículo está formado por tres capsulas con forma de torpedo unidos entre sí por un marco de aluminio como se puede ver en la figura 1. Este diseño es compacto y ofrece, a su vez, un buen comportamiento hidrodinámico junto a un gran espacio donde poder albergar numerosos sensores, pudiendo operar en profundidades de hasta 500 metros.

Girona 500 cuenta con un sistema de propulsión configurable, donde se puede adaptar la propulsión en función de la misión. La mínima configuración cuenta con 3 propulsores, adecuada para misiones donde la precisión del movimiento no sea muy exigente y pudiendo configurarse con hasta 7 propulsores, para misiones con requerimientos de alta precisión de movimiento.

Al igual que los propulsores, los sensores también son configurables para adaptarse al tipo de misión. Cuenta con sensor GPS, sensores de velocidad, sonares, cámaras, termómetros, giroscopio... etc. Todos estos sensores quedan representados

Capítulo 2

en la figura 2. Además de esto, se le puede acoplar un brazo robótico de 4 Grados de Libertad (GDL) para poder recoger pequeñas muestras de suelo marino u objetos.

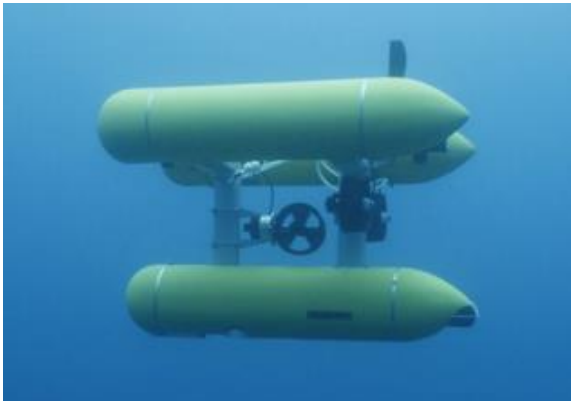


Figura 1: Vista Girona 500 [2]

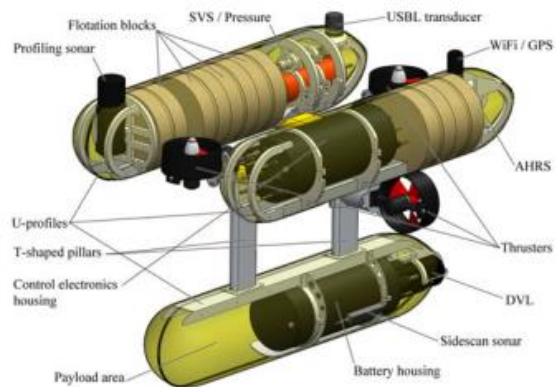


Figura 2: Sensores Girona 500 [2]

Este AUV ha completado con éxito numerosas misiones, siendo la más notoria la participación y superación del proyecto RAUVI (para más información ver [3]).

2.1.2. SPARUS II

Sparus II [4] [5] es un AUV desarrollado en el laboratorio de la universidad de Girona en 2013, España, creado para cubrir y barrer grandes extensiones de terreno submarino con fin científico. Este tipo de misiones suelen ser adecuadas para modelos de robots submarinos con forma de torpedo dado que poseen una controlabilidad, eficiencia y durabilidad mayor que otros modelos de robots submarinos.

Este vehículo en concreto tiene una forma de torpedo con una longitud de 1.6 metros, como se muestra en la figura 3, para ser eficiente cuando navega a velocidades medias y altas. También cuenta con características especiales de flotabilidad para poder mantenerse estable a una altura deseada. Sparus II puede operar en misiones de hasta 200 metros de profundidad.

El sistema de propulsión del vehículo cuenta con 3 propulsores, 2 colocados horizontalmente y 1 verticalmente teniendo hasta un total de 5 GDL y pudiendo alcanzar velocidades máximas de entre 3 y 4 nudos. Además del sistema de propulsión, cuenta con unas aletas en la parte trasera del robot que, aparte de proporcionar estabilidad en el movimiento, proporciona eficiencia a la hora de controlar la profundidad. Estas aletas pueden observarse en la figura 4.

Capítulo 2

Los sensores implementados en este tipo de AUV son más limitados dado que el espacio es mucho más reducido que en otros modelos. Cuenta con sensores de GPS, sónares para detección de obstáculos, sónares de barrido lateral para escanear el fondo marino, servomotores para controlar el movimiento de los propulsores, sensores de comunicación, giroscopio... etc. La autonomía media de este modelo es de 8 a 10 horas, aunque para velocidades más bajas, puede llegar a ser de 14 horas.

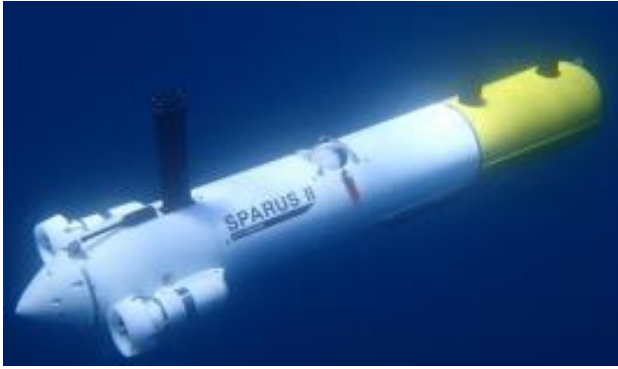


Figura 3: Vista Sparus II [4]

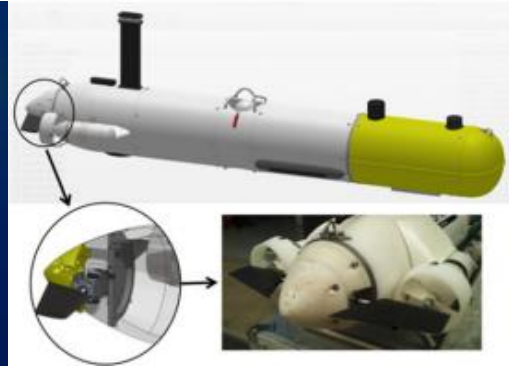


Figura 4: Propulsores y aletas 500 [4]

2.1.3. REMUS 100

REMUS 100 [20] [21] es un vehículo autónomo submarino diseñado tanto para realizar investigaciones científicas como para realizar misiones con objetivo militar. Su principal ventaja es su pequeño tamaño y su fácil manejo.

Este vehículo también tiene una forma de torpedo y cuenta con una longitud de 134 centímetros y un peso aproximado de 31 kilogramos. Su diseño es mucho más compacto en comparación con los otros robots expuestos, ganando versatilidad y controlabilidad, pero limitando su espacio para portar sensores.

Cuenta con un solo propulsor en la parte trasera de su cuerpo donde se puede variar su velocidad en función del tipo y tamaño de hélice que se emplee. Cuenta con 4 aletas situadas en la cola para poder controlar su movimiento

Este AUV está equipado con sensores de navegación acústico, además de un GPS para determinar su posición exacta. Está equipado con sensores de escaneo lateral para poder obtener información del suelo marino. Cuenta con cámaras y su batería le permite tener una autonomía de misión de hasta 120 kilómetros.

2.2. CONTROL GEOMÉTRICO DE SEGUIMIENTO DE TRAYECTORIA

El control de seguimiento de trayectoria (CST) consiste en utilizar el comportamiento cinemático/dinámico de un vehículo para realizar el seguimiento de una ruta preestablecida. Este tipo de control se puede abordar de diferentes maneras descritas en [6], algunas de estas son: i) empleando control geométrico, ii) empleando el método directo de Lyapunov, iii) empleando control robusto y iv) empleando controladores basados en la optimización entre muchos otros.

En este sub-apartado se centrará en describir los controladores más importantes entre los controladores geométricos que destacan por su sencillez y efectividad. El principio fundamental de estos modelos es establecer relaciones geométricas entre las características del vehículo y la trayectoria a seguir. Los controladores geométricos de seguimiento de trayectoria suelen depender de un punto de referencia a lo largo del recorrido conocido como el punto de mira (LA, del inglés *look ahead*).

2.2.1. CHASING THE CARROT

El algoritmo “Chasing the Carrot” (CC) fue el primer algoritmo en emplear un punto virtual para realizar seguimientos de trayectoria, siendo este a su vez el más sencillo de todos [7].

CC funciona colocando un punto virtual a una distancia denominada “look ahead distance” en la trayectoria deseada para que el robot sea capaz de seguirlo. A medida que transcurre el tiempo, el robot intenta alcanzar el punto virtual siguiendo así el camino deseado. En este enfoque, el punto virtual sería la zanahoria y el robot sería el conejo, de ahí nace el nombre del algoritmo. Estos conceptos quedan ilustrados en la figura 5.

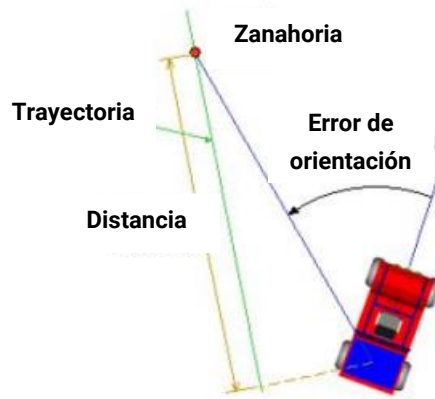


Figura 5: Representación gráfica CC [7]

En la práctica el método CC presenta algunas limitaciones. En primer lugar, el seguimiento no opera bien cuando la trayectoria contiene ángulos rectos y es estrictamente necesario su realización. Otra desventaja es que el vehículo puede oscilar si no se ajusta bien el parámetro LA. Esto se explica a mayor detalle en el apartado 4.2.

2.2.2. PERSECUCIÓN PURA

Persecución pura (PP, en inglés *pure pursuit*) es un algoritmo de seguimiento que funciona calculando una curvatura que mueve el vehículo de su posición actual hasta un punto objetivo. El algoritmo PP se implementó inicialmente para guiar misiles hacia un objetivo, evolucionando hasta la actualidad, donde se emplea principalmente en el sector de la automoción para el desarrollo de vehículos autónomos [9].

Este algoritmo funciona estableciendo un punto de referencia delante del vehículo a una distancia ajustable. Dicha distancia se llama distancia del punto de mira (en inglés, *look ahead distance*). El vehículo se encarga de seguir el punto objetivo describiendo un arco de circunferencia para alcanzar dicho punto. Cabe mencionar que, a medida que el vehículo avanza hacia el punto objetivo, este también sigue avanzando, reduciendo el ángulo de dirección y llevando suavemente el vehículo hacia la ruta. Este modelo queda ilustrado en la figura 6.

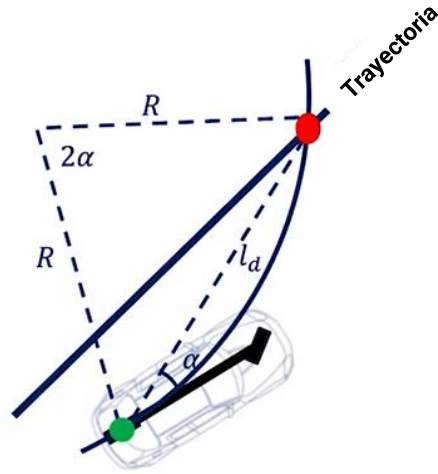


Figura 6: Representación gráfica PP [8]

donde:

- “R” es el radio de la curvatura que describe el vehículo.
- “ l_d ” es el parámetro de la distancia del punto de mira.
- “ α ” es el ángulo que forma la orientación del vehículo con el vector objetivo.

Siguiendo el planteamiento teórico en [8], para construir el controlador PP, se tiene que calcular el arco que abarca el punto de referencia de vehículo y el punto de destino de la trayectoria. Este arco se puede calcular aplicando trigonometría. Si se coge el punto objetivo de la trayectoria, el centro del cuerpo y el centro de rotación instantáneo juntos forman un triángulo con lados R a los lados y l_d de lado de longitud.

Basándose en la ley de los senos, se obtiene de la figura 6:

$$\frac{l_d}{\sin 2\alpha} = \frac{R}{\sin(\frac{\pi}{2} - \alpha)} \quad [2.1]$$

$$\frac{l_d}{2 * \sin \alpha * \cos \alpha} = \frac{R}{\cos \alpha}$$

$$\frac{l_d}{\sin \alpha} = 2R$$

$$K = \frac{1}{R} = \frac{2 * \sin \alpha}{l_d} \quad [2.2]$$

donde K es la curvatura del recorrido que tiene que realizar el vehículo para alcanzar el objetivo.

Capítulo 2

El método PP comparte aspectos comunes con el modelo CC. En ambos se define el parámetro del punto de mira y en base a éste se calcula un punto objetivo en la trayectoria a seguir. Las ventajas de este algoritmo frente a CC es que es menos oscilatorio y es mucho más preciso realizando giros.

2.2.3. PERSECUCIÓN VECTORIAL

El modelo de persecución vectorial (VP, en inglés *Vector Pursuit*) es un método más completo que los anteriores y se basa en la "teoría de tornillos". Dicha teoría puede emplearse para describir cualquier movimiento instantáneo de un cuerpo sólido respecto a un sistema de referencia. Este control fue diseñado para que el vehículo llegase al punto destino con la orientación y curvatura adecuada. La ventaja de esto es que permite tener no solo un control sobre la posición, sino también de la orientación del vehículo. Para definir un tornillo hay que determinar un eje de giro y un paso. Este modelo queda ilustrado en la figura 7:

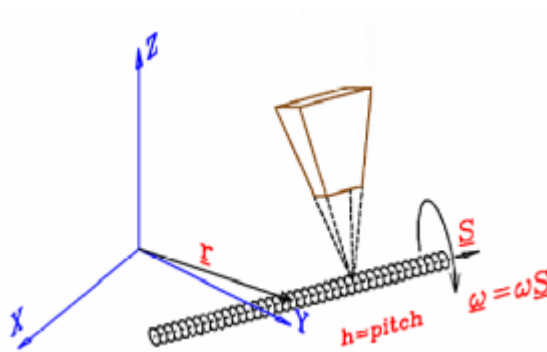


Figura 7: Representación Gráfica VP [7]

Las letras de la figura 7 representan:

- "S" es el tornillo definido por su línea central.
- " ω " es la velocidad angular.
- "r" es cualquier del origen hasta la línea central del tornillo.
- "h" es el paso del tornillo

Basándose en el desarrollo matemático expuesto en [7] [10], la manera más fácil de representar un tornillo es utilizando las coordenadas de Plücker. Siguiendo la figura 7, se tiene:

Capítulo 2

$$\bar{S} = \frac{\bar{r}_2 - \bar{r}_1}{|\bar{r}_2 - \bar{r}_1|} \quad [2.3]$$

$$\bar{S}_0 = \bar{r}_1 \times \bar{S} \quad [2.4]$$

Donde (\bar{S}, \bar{S}_0) son las coordenadas de Plücker que marcan el eje de giro del tornillo. Definiendo S como $\bar{S} = (L, N, M)$ junto ha $\bar{S}_0 = (P, Q, R)$, $\bar{r}_1 = (x_1, y_1, z_1)$ y $\bar{r}_2 = (x_2, y_2, z_2)$ siendo:

$$L = \frac{x_2 - x_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}} \quad [2.5]$$

$$M = \frac{y_2 - y_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}} \quad [2.6]$$

$$N = \frac{z_2 - z_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}} \quad [2.7]$$

$$P = y_1 N - z_1 M \quad [2.8]$$

$$Q = z_1 N - x_1 M \quad [2.9]$$

$$R = x_1 M - y_1 L \quad [2.10]$$

Empleando estas definiciones y estableciendo valores para el paso del tornillo, se puede determinar qué; la velocidad instantánea de cualquier punto de un sólido rígido, que rota con una velocidad angular ω en torno al tornillo \bar{T} siendo su eje de giro (\bar{S}, \bar{S}_0) y con paso h se llega a la ecuación:

$$\omega \bar{T} = (\omega \bar{S}; \omega \bar{S}_{0h}) \quad [2.11]$$

$$\bar{S}_{0h} = \bar{S}_0 + h \bar{S} = \bar{r} \times \bar{S} + h \bar{S} \quad [2.12]$$

2.3. MÉTODOS DE BARRIDO DE EXTENSIONES

Para realizar una buena cobertura del terreno empleando robots, es necesario tener en cuenta los siguientes requisitos:

1. Los robots se deben mover alrededor de todos los puntos del espacio para cubrir todo el terreno de interés.
2. Si los robots actúan de manera independiente, deben cubrir todo el terreno eficientemente sin que superpongan las rutas de viaje.

Capítulo 2

3. Se deben realizar operaciones secuenciales y continuas sin generar superposiciones en el barrido.
4. Si el terreno tiene obstáculos, los robots deben de ser capaces de esquivar obstáculos.
5. Se debe priorizar movimientos simples para una mayor simplicidad de control.
6. Bajo la influencia del entorno, se debe priorizar siempre la mejor ruta para todos los robots.

Como no siempre es posible cumplir todas las condiciones cuando se realiza un barrido, es necesario priorizar estas en función del tipo de terreno que se quiera explorar, según [11].

Los algoritmos de cobertura se pueden clasificar de dos formas; como algoritmos desconectados (en inglés, *off-line*), donde este tipo de algoritmos se basan principalmente en información estacionaria siendo el entorno a barrer ya conocido, y como algoritmos conectados (en inglés, *in-line*), donde el entorno a barrer es parcialmente conocido o completamente desconocido y se emplean sensores para obtener información del entorno. Dentro de estos dos tipos existen numerosos algoritmos con diferentes estrategias para cumplir los requisitos anteriormente mencionados. A continuación, se expondrán las estrategias más comunes de los algoritmos desconectados dado que, en la solución propuesta, el entorno de simulación será completamente conocido para los robots.

2.3.1. ALGORITMO DE LA CORTADORA DE CÉSPED

El algoritmo de la cortadora de césped (en inglés, *landmower algorithm*) se emplea para realizar un barrido de una zona delimitada. Su estrategia es simple, cubrir toda la zona de barrido haciendo un movimiento de *zig-zag* como si se estuviera cortando el césped de un jardín, de ahí el nombre del algoritmo. Un ejemplo queda representado en la siguiente figura:

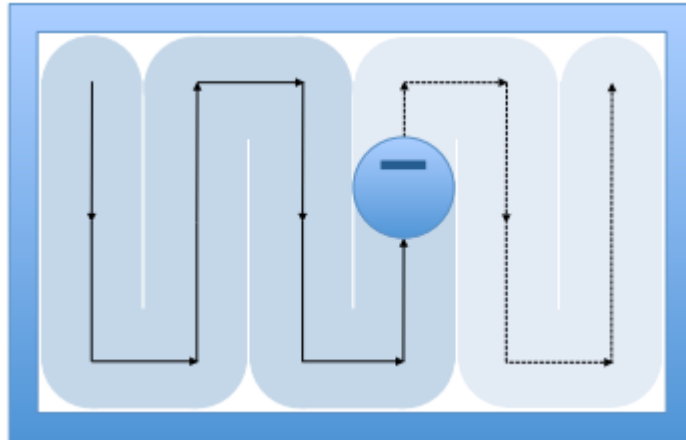
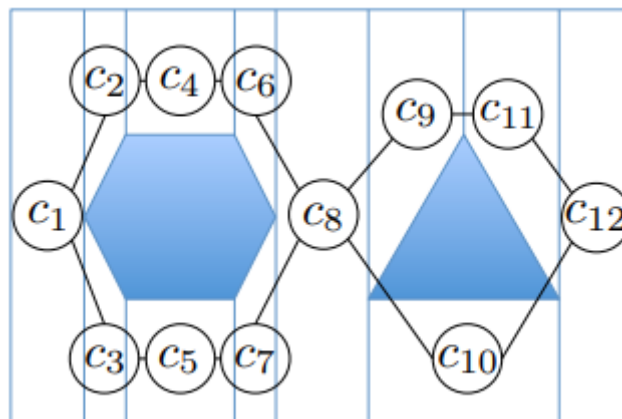


Figura 8: Movimiento en zigzag [11]

Cabe mencionar que esto solo puede realizarse si no existen obstáculos en el entorno. Existen variaciones de este algoritmo donde, empleando sensores, se pueden esquivar obstáculos, aunque el recorrido suele ser ineficiente dado que no se emplean estrategias de barrido en el entorno para poder optimizar el recorrido.

2.3.2. MÉTODO CLÁSICO DE DESCOMPOSICIÓN CELULAR EXACTA

Como viene expuesto en [11] [12], este tipo de estrategia descompone el espacio en regiones simples no superpuestas llamadas celdas. La unión de estas celdas debe ocupar todo el espacio que se quiere cubrir. Una celda se puede determinar cuándo se encuentra un punto crítico, siendo estas zonas donde se produce un cambio en el entorno. Si se parametriza el entorno en una matriz de *ceros* y *unos* siendo *cero* un lugar no ocupado y *uno* un lugar ocupado y se evaluase cada columna del mapa, se puede decir que un punto crítico del espacio es donde se produce una alteración de estos valores. Un ejemplo de una celda puede ser cuando un vector delimitador que evalúa las columnas de la matriz se encuentra con una esquina de un obstáculo, donde la secuencia sería [...0 1 0...]. En ese preciso momento, se establecería una celda quedando



Capítulo 2

Figura 9: Representación gráfica de un entorno dividido en celdas [11]

está delimitada desde el inicio de la evaluación hasta el obstáculo encontrado. Esto queda representado en la siguiente figura.

Al quedar el espacio dividido en celdas, se pueden aplicar estrategias para cubrir todas las celdas en el menor tiempo posible. El algoritmo más empleado para resolver este tipo de problemas es el "algoritmo del viajante" (en inglés, *travelling salesman problem*).

De los algoritmos que emplean métodos clásicos de descomposición celular destacan la descomposición trapezoidal, siendo esta el primer algoritmo generado siguiendo este tipo de estrategia, y la descomposición de *Boustrophedon*, donde su principal ventaja es que es capaz de optimizar las celdas que se generan en el entorno a través de uniones. Cabe mencionar que estos algoritmos están planteados para espacios de dos dimensiones.

2.3.3. DESCOMPOSICIÓN CELULAR BASADA EN MORSE

Este tipo de algoritmos se basan en los primeros mencionados con la ventaja de que son capaces de trabajar con obstáculos no poligonales. Los puntos críticos de la función de Morse son restringidos por los límites del obstáculo, donde estos se utilizan para determinar la posición de descomposición de la celda. Otra ventaja de este algoritmo es que puede utilizarse en espacios de n-dimensiones. La principal desventaja de estos tipos de algoritmos es que, dependiendo de las formas geométricas de los obstáculos, se pueden perder puntos críticos generando celdas poco aptas para el barrido o quedando zonas del mapa sin determinar. Estos tipos de algoritmos no operan bien en entornos rectilíneos [11] [12]. Un ejemplo se muestra en la figura contigua donde se puede observar una figura no poligonal en el centro de esta que representa el obstáculo y las celdas, cada una con un color distinto.

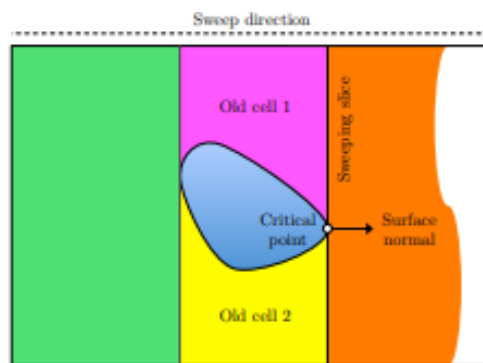


Figura 10: Representación gráfica de la descomposición celular basada en Morse [11]

Capítulo 2

Existen numerosos algoritmos basados en Morse como el algoritmo de descomposición de *Boustrophedon* basado en Morse y el algoritmo de descomposición celular basado en Morse combinado con el diagrama de *Voronoi* generalizado. Cabe mencionar que ambos algoritmos son modificaciones de los anteriores adaptados a esta nueva estrategia.

2.4. TIPOLOGÍA Y CONTROL DE FORMACIONES

Cuando se sincronizan varios robots, es importante tener determinada la formación en función de la tarea a realizar y aplicar la estrategia de control que más se adecue a está. En este sub-apartado se explican los distintos tipos de formaciones más comunes aplicados a barridos de extensiones y las respectivas estrategias de control.

2.4.1. TIPOS DE FORMACIONES

El concepto de formación viene inspirado del comportamiento de los animales tales como una banda de pájaros o un banco de peces. En ellas, un número de animales adoptan una cierta formación para aumentar la supervivencia de los individuos que lo conforman aplicando estrategias de grupo. Imitando este comportamiento de los animales puede hacer que un grupo de robots sean capaces de cumplir tareas complicadas y mejorar su autonomía [13].

Las consideraciones más importantes a la hora de adoptar una formación es que esta se adapte tanto al entorno como al tipo de misión que se le encargue, pudiendo variar en función de las circunstancias. Teniendo esto en cuenta, se puede agrupar 4 tipos de formaciones principales que se describirán a continuación:

- Formación en columna: Esta formación aporta un gran rango de barrido siendo útil para cubrir grandes extensiones sin mucho relieve. Para esta y las siguientes figuras, la flecha azul representa el sentido del movimiento de los robots. Los círculos corresponden a los robots que conforman la formación.



Figura 11: : Formación en columna [13]

Capítulo 2

- Formación en línea: Esta formación aporta un área de barrido mínimo siendo útil para entornos con una geografía más compleja, como cuevas o estrechamientos del relieve.



Figura 12: Formación en línea [13]

- Formación en V: Esta formación tiene una buena área de barrido, desenvolviéndose bien en entornos con relieves relativamente complicados. La principal ventaja de este tipo de formación es la facilidad de comunicación entre los robots que la conforman.

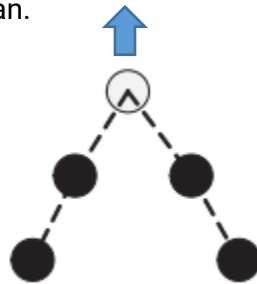


Figura 13: Formación en V [13]

- Formación en diamante: Esta formación es una variante de la formación anterior recogiendo las mismas características. La diferencia y ventaja principal es su flexibilidad.

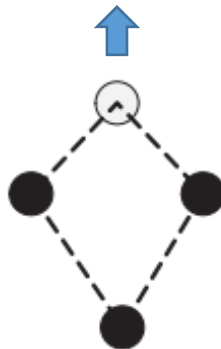


Figura 14: Formación en diamante [13]

Capítulo 2

2.4.2. ESTRATEGIAS DE CONTROL DE FORMACIONES

La estrategia de control es importante para mantener la configuración de la formación y el comportamiento del grupo. Existen numerosas estrategias de control de formaciones, las más importantes vienen expuestas a continuación basándose en [14].

Control Líder-Seguidor

Esta estrategia consiste en determinar uno o varios robots como líderes y el resto de los miembros como seguidores. La misión del líder es seguir la trayectoria deseada mientras que los seguidores deben encargarse de seguir al líder manteniendo la formación.

Los seguidores son capaces de realizar esta labor a través del control de dos posibles variables de control; estas son el método distancia-distancia o distancia-ángulo.

- Distancia-distancia: Se mantiene la configuración controlando la distancia que hay entre el líder y el resto de las distancias que existen entre los miembros que conforman la formación.

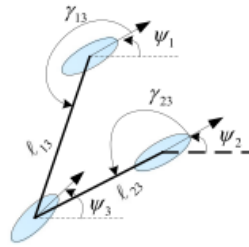


Figura 15: Estructura distancia-distancia [14]

- Distancia-ángulo: Se mantiene la configuración controlando la distancia respecto al líder y un ángulo de referencia respecto al vector que forma la posición del robot seguidor con el del líder.

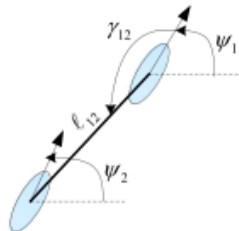


Figura 16: Estructura distancia-ángulo [14]

Capítulo 2

La principal ventaja de esta estrategia es su sencillez junto a su relativa fácil implementación. La desventaja de esta estrategia es la gran dependencia que se genera con el vehículo líder.

Estructura Virtual

Esta estrategia consiste en que los distintos robots mantengan una formación en torno a un punto virtual. Los robots formarán y se moverán como un bloque sólido. Dicha solidez es la principal ventaja de esta estrategia donde la formación tiende a deformarse menos comparado con otras estrategias. La desventaja es su falta de flexibilidad teniendo grandes limitaciones a la hora de esquivar obstáculos. Es por ello por lo que este tipo de formaciones se usa solo en el espacio con pequeñas formaciones de satélites, o en el aire, donde la presencia de obstáculos es inexistente.

Estructura potencial artificial

El control de esta estrategia consiste en generar fuerzas potenciales para poder controlar la formación, la evasión de obstáculos y la llegada al objetivo. En la figura 17, se puede ver como una formación de robots se ve afectada por estas fuerzas potenciales. El objetivo es generar “una fuerza de atracción” entre los distintos robots cuando, por ejemplo, se distancian demasiado entre sí o se quiere llegar a cierto punto de interés. Se genera “una fuerza de repulsión” cuando estos se acercan demasiado entre sí o hay un obstáculo por el camino.

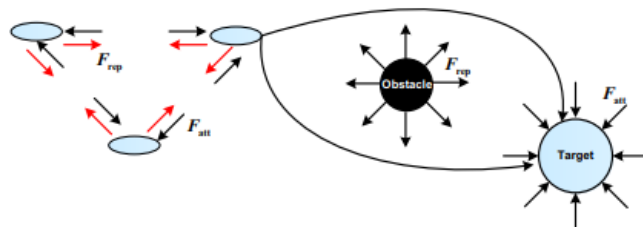


Figura 17: Estructura potencial artificial [14]

Las fuerzas potenciales atractivas son funciones cuadráticas basadas en las distancias, es decir, estas aumentan cuadráticamente a medida que la distancia entre los robots aumenta. Las funciones de fuerzas potenciales repulsivas son inversamente proporcionales a la distancia de los obstáculos, es decir, estas aumentan cuando la distancia disminuye. Finalmente se hace un balance de la fuerza potencial haciendo el gradiente de las distintas fuerzas que actúan.

Las principales ventajas de esta estrategia es que la formación no depende de un robot líder y que tiene baja complejidad computacional. La principal debilidad de este

Capítulo 2

algoritmo es que se pueden generar problemas a la hora de calcular mínimos locales a la hora de calcular las fuerzas.

Capítulo 3. SOLUCIÓN PLANTEADA

En el siguiente capítulo se presentan los desarrollos teóricos aplicados a la práctica de este trabajo fin de grado. El capítulo se divide en los siguientes subapartados: En primer lugar, se comienza el capítulo desarrollando el entorno de simulación y la estrategia de barrido, apartado 3.1. En segundo lugar, se establece el modelo y la cinemática empleada para el AUV, apartado 3.2. En tercer lugar, se explica el procedimiento empleado para el seguimiento de trayectoria, apartado 3.3. En cuarto lugar, se justifica la estrategia de control empleada para el control de formaciones, apartado 3.4.

3.1. ESTRATEGIA DE BARRIDO

En este subcapítulo se muestra la solución planteada a la hora de desarrollar el entorno de simulación, apartado 3.1.1 y el planteamiento utilizado para establecer el barrido de la zona, apartado 3.1.2.

3.1.1. ENTORNO DE SIMULACIÓN

Para construir el mapa de donde se va a realizar la simulación, se ha decidido dividir este en “cubos” (en inglés, *voxels*), donde cada cubo está representado por coordenadas en el espacio. Dichos cubos pueden contener dos valores de, 0 o 1. Los cubos que contengan un 0 representaran un espacio vacío mientras que las cubos que contengan un 1 representaran un espacio ocupado. Siguiendo esta lógica, se ha creado una matriz que representa un entorno submarino como se muestra en la siguiente figura.

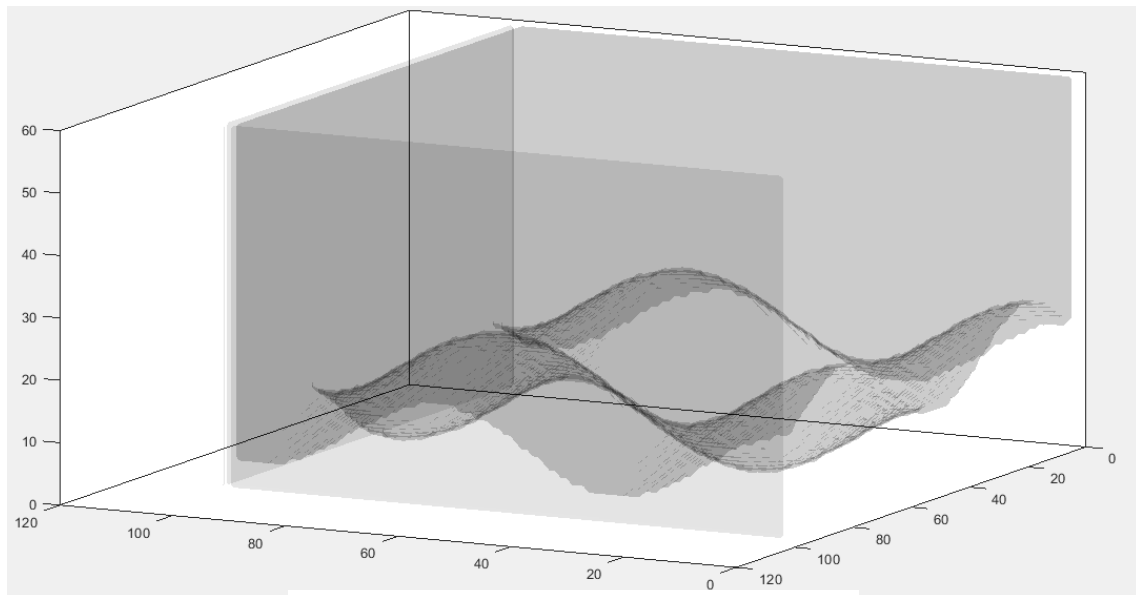


Figura 18: Entorno submarino propuesto

Cabe mencionar que cada cubo posee una unidad de distancia, es decir, ocupa un espacio físico real. Las unidades de distancia determinadas para cada cubo vienen explicadas en el desarrollo experimental, apartado 4.1.

Se ha decidido modelar el suelo submarino empleando la siguiente función:

$$z = \text{sen}(x) + \cos(y)$$

A pesar de que el suelo submarino en profundidades bajas suele ser llano completamente, se ha decidido generar estas formas dunosas para generar mayor complejidad en la trayectoria y afrontar un barrido en tres dimensiones en vez de en un

Capítulo 3

plano. El mapa queda delimitado por 5 planos colocados en los límites frontal, trasero, laterales y superior para limitar el entorno.

3.1.2. BARRIDO DE ZONA

Para realizar el barrido del entorno se ha usado el algoritmo de la cortadora de césped dado que los AUV van a barrer la zona a una altura determinada sin presencia de obstáculos. Para generar el recorrido, se ha creado un algoritmo que, con una sucesión de bucles y condiciones, cubre el mapa realizando un recorrido de *zigzag*. Las coordenadas de los puntos generados quedan guardadas sucesivamente en una variable que acaba conformando la ruta. Para que el recorrido se adapta al relieve del entorno, se ha creado una función adicional que evalúa la altura en la que se debe posicionar el punto de la trayectoria correspondiente. Esto es posible analizando la columna de ceros y unos que conforman la altura de la posición X e Y que se está evaluando. Dicha función recorre la columna desde su valor final, que correspondería a la parte más alta del entorno, hacia abajo hasta encontrarse con un obstáculo. Cuando se encuentra con un obstáculo, guarda dicha posición de altura y le suma un valor de 20 unidades para colocar el punto de la trayectoria en la altura deseada. Se ha seleccionado el valor de 20 unidades inspirado en la altura en la que se suelen barrer zonas submarinas usando sonares de escaneo lateral (en inglés, *side scan sonars*). 20 metros (ver apartado 4.1 para ver cómo se han determinado las unidades) es la altura adecuada para obtener buenos resultados del sensor dentro del umbral. En la siguiente figura se representa lo explicado anteriormente.

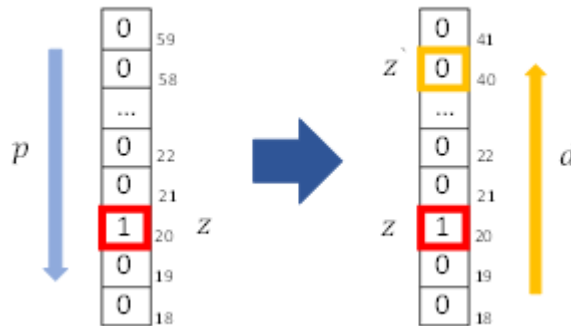


Figura 19: Representación gráfica de la lógica empleada para obtener la altura deseada en la trayectoria

donde:

- p : sentido en el que se recorre el vector que representa la altura.
- z : posición en la que se encuentra el obstáculo.

Capítulo 3

- p : suma del valor correspondiente para determinar la altura deseada.
- z^* : posición calculada para determinar la altura del punto de la trayectoria.

3.2. CINEMÁTICA DEL ROBOT SUBMARINO

En este subcapítulo se presenta el modelado cinemático de un robot submarino. El modelado cinemático consiste en definir las ecuaciones que relacionan los diferentes tipos de coordenadas utilizadas para expresar la velocidad del robot. Se ha seleccionado un modelo de robot submarino con forma de torpedo dado que es el modelo más común y adecuado para realizar misiones de barrido y rastreo submarino. Se ha inspirado el modelado escogiendo el AUV REMUS (ver [15] para más información) contando con 6 GDL pudiendo realizar movimientos en todas las direcciones del espacio.

Para entender la cinemática, se comienza el capítulo explicando los marcos de referencia y los movimientos posibles en el espacio, apartado 3.2.1. Se continúa describiendo las rotaciones a través de matrices de rotación, apartado 3.2.2. Se desarrolla relacionando estas rotaciones con los ejes coordenados generando las transformaciones de velocidades necesarias para definir el movimiento, apartado 3.2.3.

3.2.1. MARCOS DE REFERENCIA

Como viene representado en la figura 20, a la hora de analizar el movimiento de un AUV, es necesario definir dos marcos de referencia para poder describir su movimiento.

- Marco de referencia inercial: Este sistema de referencia también se conoce como global. Dicho sistema se emplea para expresar la ubicación lineal y angular de todos los elementos que conforman el sistema a controlar. El origen se suele colocar en puntos de interés para delimitar el espacio.
- Marco de referencia del cuerpo: Este sistema de referencia está fijo al cuerpo y se suele situar en el centro de gravedad del cuerpo o en algún punto de interés del cuerpo. Dicho sistema se emplea para expresar las velocidades lineales y angulares del UAV. Bajo regla general, el marco de referencia x_a es el eje longitudinal, el y_a es el eje transversal y el z_a va dirigido hacia arriba.

Capítulo 3

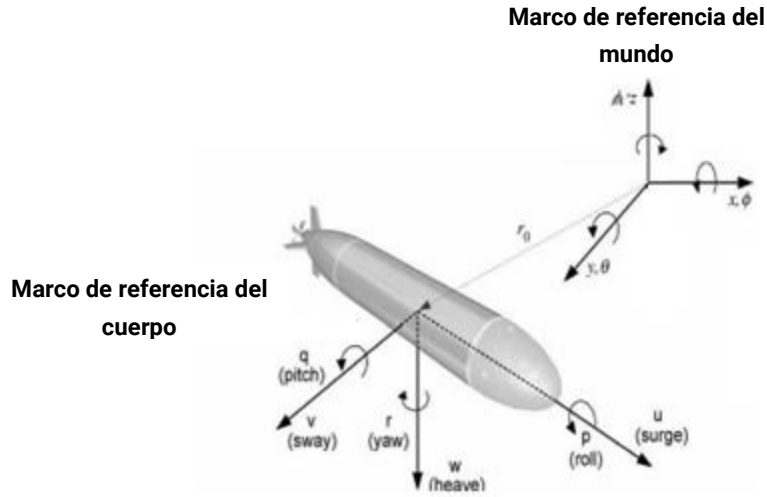


Figura 20: Representación de los marcos de referencia [16]

Teniendo en cuenta una vez ya las referencias empleadas para poder describir los movimientos y orientaciones, pasaremos a describir estos en la siguiente tabla:

Grado de Libertad	Movimiento	Nombre	Posición	Velocidad
1	Traslación en x	Avance	x	u
2	Traslación en y	Desvío	y	v
3	Traslación en z	Viraje	z	w
4	Rotación en x	Alabeo	Φ	p
5	Rotación en y	Cabeceo	θ	q
6	Rotación en z	Guiñado	ψ	r

Tabla 1: Movimientos en el espacio. [16]

A continuación, se expondrán las siguientes formulas extraídas de [15] [17]. Para representar la posición del vehículo con respecto al marco inercial absoluto se puede expresar el siguiente vector:

$$\eta = \begin{bmatrix} \xi_o^a \\ \theta \end{bmatrix} \quad [3.2]$$

donde:

- ξ_o^a : vector que representa la posición del marco de referencia del AUV con respecto a marco de referencia global.
- θ : vector que representa la orientación relativa entre los dos sistemas de referencias.

Capítulo 3

Siguiendo el mismo procedimiento, la velocidad del AUV se puede expresar en el siguiente vector como:

$$V = \begin{bmatrix} v_o^a \\ \omega_a^a \end{bmatrix} \quad [3.3]$$

donde:

- $v_o^a = [u \ v \ w]^T$: es la matriz que expresa la velocidad lineal del vehículo medido en coordenadas del marco de referencia del cuerpo del AUV.
- $\omega_a^a = [p \ q \ r]^T$: es la matriz que representa la velocidad angular del vehículo medido en coordenadas del marco de referencia del cuerpo del AUV.

3.2.2. MATRIZ DE ROTACIÓN

Las matrices de rotación se emplean para describir la orientación entre dos sistemas coordenados y son útiles para realizar transformaciones lineales entre espacios vectoriales. Dichas matrices se pueden definir como matriz elemental que representa una rotación alrededor de uno de los ejes del marco de referencia.

Teniendo un vector \bar{r}_1 expresado en coordenadas del sistema de referencia del cuerpo, dicho vector puede expresarse en coordenadas globales realizando la siguiente operación:

$$\bar{r}_1 = R_2^1 \bar{r}_b \quad [3.4]$$

Siguiendo este procedimiento, si se tienen tres sistemas coordenados 1, 2 y 3 y siendo:

- Si R_2^1 es la matriz de rotación de 2 a 1 y,
- Si R_3^2 es la matriz de rotación de 3 a 2.

Se puede decir que la matriz de rotación de 3 a 1 es:

$$R_3^1 = R_2^1 R_3^2 \quad [3.5]$$

Definiendo así la regla de composición para rotaciones que se empleará en el siguiente apartado.

3.2.3. TRANSFORMACIÓN DE VELOCIDADES LINEALES Y ANGULARES

De acuerdo con L. Euler, cualesquiera dos marcos coordenados ortonormales independientes, con un origen común, pueden ser relacionados por una sucesión de no más de tres rotaciones alrededor de los ejes coordenados [17], mostrado en la figura 21.

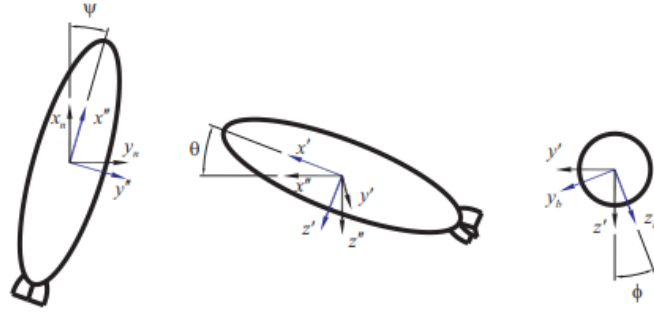


Figura 21: Rotación sobre los ejes [17]

Aplicando esta definición y dada una sucesión de ángulos de Euler, también conocidos en geometría como los ángulos de Tait-Bryan, es posible encontrar las matrices de rotación correspondientes para cada eje coordenado. Las matrices de rotación del AUV son:

- Matriz de rotación que representa la rotación respecto al eje x

$$R_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad [3.6]$$

- Matriz de rotación que representa la rotación respecto al eje y

$$R_{y,\theta} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad [3.7]$$

- Matriz de rotación que representa la rotación respecto al eje z

$$R_{z,\psi} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad [3.8]$$

Cabe mencionar que todas las rotaciones expuestas se llevan a cabo desde el marco de referencia inercial al del cuerpo. Aplicando la regla de composición de rotaciones, se puede obtener la siguiente matriz de rotación en los tres ejes coordenados:

Capítulo 3

$$R_a^n = R_{z,\psi} * R_{y,\theta} * R_{x,\phi} \quad [3.9]$$

$$= \begin{bmatrix} \cos \psi \cos \theta & -\sin \psi \cos \theta + \cos \psi \sin \theta \sin \phi & \sin \psi \sin \theta + \cos \psi \cos \theta \sin \phi \\ \sin \psi \cos \theta & \cos \psi \cos \theta + \sin \psi \sin \theta \sin \phi & -\sin \psi \sin \theta + \cos \psi \cos \theta \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix}$$

Con la matriz de rotación definida, se puede establecer la relación cinemática del vehículo con el marco de inercia del mundo siendo esta:

$$\dot{\xi}_o^a = R_a^n v_o^a \quad [3.10]$$

donde:

- $\dot{\xi}_o^a$: vector de velocidades lineales resultantes.
- R_a^n : matriz de transformación que relaciona las velocidades lineales del cuerpo con el vector de velocidades de los ángulos de Euler.
- v_o^a : matriz de velocidades lineales del vehículo en los tres ejes coordenados.

De la misma manera que se ha hecho con las velocidades lineales, se puede hacer con las velocidades angulares. La transformación de velocidades angulares del cuerpo y el vector de velocidad de los ángulos de Euler viene dada por:

$$\dot{\theta} = T_\theta \omega_a^a \quad [3.11]$$

donde:

- $\dot{\theta}$: vector de velocidades angulares resultantes.
- T_θ : matriz de transformación que relaciona las velocidades angulares del cuerpo con el vector de velocidades de los ángulos de Euler.

$$T_\theta = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix} \quad [3.12]$$

- ω_a^a : matriz de velocidades angulares del vehículo en los tres ejes coordenados.

Si se agrupan las expresiones 3.8 y 3.9 se puede obtener:

$$\dot{\eta} = JV \quad [3.13]$$

donde:

$$[3.14]$$

$$J(V) = \begin{bmatrix} R_a^n & 0 \\ 0 & T_\theta \end{bmatrix}$$

3.3. PERSECUCIÓN PURA

Para realizar el seguimiento de trayectoria, se ha decidido emplear el algoritmo PP. Se ha seleccionado dicho algoritmo dado que es el controlador más eficaz y empleado dentro de los controladores geométricos. Este algoritmo ha sido ligeramente modificado para que pueda funcionar en las tres dimensiones del espacio.

Este subcapítulo relata el procedimiento seguido para poder emplear dicho algoritmo en un AUV. El capítulo comienza explicando cómo parametrizar la trayectoria y aproximar esta para que los robots sean capaces de seguirla, apartado 3.3.1. Se continúa desarrollando el concepto de la distancia LA y como obtener el valor óptimo, apartado 3.3.2. Se plantea el problema y la solución de calcular el punto LA, apartado 3.3.3. Se finaliza el capítulo explicando el error de tramo cruzada (en inglés, *cross-track error*), apartado 3.3.4.

3.3.1. PARAMETRIZACIÓN Y APROXIMACIÓN DE LA TRAYECTORIA

Para hacer que el punto objetivo sea capaz de avanzar a lo largo de la trayectoria es necesario aplicar una serie de conceptos matemáticos que se describirán a continuación.

En primer lugar, para poder aplicar el algoritmo de control PP es necesario tener una trayectoria definida a la que aproximarse. Dicha trayectoria tiene que estar formada por puntos consecutivos, en coordenadas cartesianas, denominados puntos de referencia. Para ello, se emplean diversas ecuaciones paramétricas para poder marcar los distintos recorridos a estudiar. Las ecuaciones cartesianas empleadas en el desarrollo experimental son las siguientes:

- Ecuación paramétrica de la recta:

$$x = x_0 + a * t; y = y_0 + b * t; \quad [4.1]$$

- Ecuación paramétrica de la circunferencia:

$$(x - C_x)^2 + (y - C_y)^2 = r^2 \quad [4.2]$$

- Ecuación paramétrica de la espiral:

$$[4.3]$$

Capítulo 3

$$x(t) = r\cos(t); y(t) = r\sin(t); z(t) = a * t$$

Una vez parametrizada la trayectoria y obtenidos los puntos de referencia, para geometrías más complejas como puede ser el del barrido de una superficie, es necesario encontrar funciones que sean capaces de ajustarse a la trayectoria. Este problema se puede abordar de distintas maneras. La solución seleccionada consiste en aproximar el recorrido mediante rectas. Cuantas más rectas se use para aproximar la trayectoria; más fiel será la aproximación de esta misma, aunque esto puede provocar la pérdida del punto LA si la distancia de este es muy grande. Este concepto se explica más detalladamente en el apartado 3.3.3.

3.3.2. PARÁMETRO DE PUNTO DE MIRA

Como se ha mencionado en el apartado 2.2, el parámetro LA es la distancia de referencia en la que el robot debe mirar desde su posición para poder calcular así su trayectoria. La elección de este parámetro es importante para el resultado final del algoritmo. Elegir tanto una distancia muy pequeña como muy grande puede hacer que el robot no siga con precisión el recorrido deseado. Siguiendo la explicación en [18]:

Si se elige una distancia muy pequeña, el robot “se estaría mirando sus propios pies”, haciendo que reaccionase con brusquedad ante cambios de dirección provocando oscilaciones en el recorrido, figura 22.

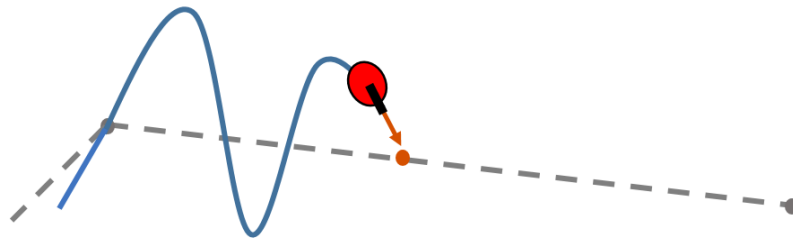


Figura 22: Parámetro LA con un valor de distancia muy pequeña [18]

Por el contrario, si escoge una distancia demasiado grande, el robot tendera a eludir giros bruscos provocando curvaturas mayores aumentando el error de posición, figura 23.

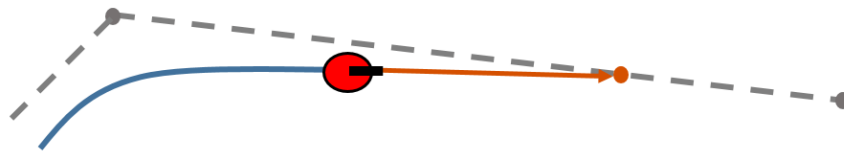


Figura 23: Parámetro LA con un valor de distancia excesivamente grande [18]

Como se ha demostrado, hay que seleccionar un parámetro LA para suavizar giros o cambios bruscos en el recorrido sin aumentar en exceso el error de posición. Este valor se calcula de manera experimental como viene desarrollado en la sección 4.2, no obstante, hay ciertas consideraciones que se pueden tener en cuenta para poder ajustar el valor adecuado. Estas son:

- La velocidad a la que se moverá el vehículo.
- El ángulo de giro del vehículo.

La velocidad es una variable influyente a considerar para determinar la distancia LA. La velocidad es directamente proporcional al parámetro LA; a mayor velocidad, la distancia LA aumentará y a menor velocidad, la distancia LA se verá reducida.

El ángulo de giro del vehículo es una restricción importante a tener en cuenta. Es decir, si la curva necesaria para realizar el giro es mayor que la capacidad que tiene el propio vehículo de girar, éste no seguirá la trayectoria fielmente y el resultado de la curva calculada sería erróneo. Al no haber encontrado un trabajo en el que aparezca descrito dicho ángulo, este se calcula de manera experimental en el apartado 4.1 aplicando la cinemática del AUV propuesta.

3.3.3. CÁLCULO DEL PUNTO OBJETIVO

Con la trayectoria y la distancia LA definidas, se puede calcular el punto objetivo que seguirá el robot a lo largo del recorrido. Como se ha explicado anteriormente, la trayectoria que debe realizar el vehículo tiene que ser parametrizada y aproximada. La idea de la aproximación es emplear rectas a través de puntos contenidos en la trayectoria. Teniendo esto en cuenta, se puede entonces establecer la distancia LA como el radio de una circunferencia y obtener el punto objetivo a través de la intersección de la recta y la circunferencia. El desarrollo empleado en la práctica es la siguiente.

Capítulo 3

Solución planteada

Para calcular la intersección recta-circunferencia se ha empleado el siguiente desarrollo matemático que calcula la intersección de una circunferencia con una recta.

Partiendo de las ecuaciones [4.1] y [4.2], expandiendo [4.2]:

$$x^2 - 2C_x x + C_x^2 + y^2 - 2C_y y + C_y^2 - r^2 = 0 \quad [4.4]$$

Sustituyendo [4.1] en [4.4]:

$$(x_0 + at)^2 - 2 * C_x(x_0 + at) + C_x^2 + (y_0 + bt)^2 - 2 * C_y(y_0 + bt) + C_y^2 - r^2 = 0 \quad [4.5]$$

Simplificando y agrupando [4.5], queda:

$$t^2(a^2 + b^2) + 2 * t(x_0 a - C_x a + y_0 b - C_y b) + x_0^2 + y_0^2 + C_x^2 + C_y^2 - r^2 - 2 * C_x x_0 - 2 * C_y y_0 = 0 \quad [4.6]$$

Por tanto:

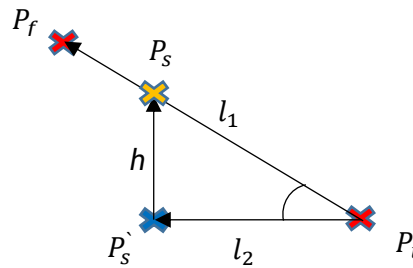
$$t^2(\bar{d} \cdot \bar{d}) + 2 * t(\bar{e} \cdot \bar{d} - \bar{d} \cdot \bar{c}) + \bar{e} \cdot \bar{e} - 2 * (\bar{e} \cdot \bar{c}) + \bar{c} \cdot \bar{c} - r^2 = 0 \quad [4.7]$$

Finalmente, simplificando se obtiene:

$$t^2(\bar{d} \cdot \bar{d}) + 2 * t(\bar{d}(\bar{e} - \bar{c})) + (\bar{e} \cdot \bar{c})(\bar{e} \cdot \bar{c}) - r^2 = 0 \quad [4.8]$$

Siendo esta la ecuación que se emplea en el código para resolver dicha intersección. Como se puede observar, se trata de una ecuación polinómica de orden dos. El criterio de selección de raíces y los posibles problemas que esto pueda acarrear quedan expuestos más abajo.

Antes de finalizar este apartado, cabe mencionar que el PP es un algoritmo que funciona en 2 dimensiones. Para resolver el problema en 3 dimensiones se ha decidido separar la variable z y calcularla aparte con una función. Dicha función recibe la intersección de la recta-circunferencia (que son las dimensiones x e y) y genera dos vectores en 3 dimensiones. El primer vector l_1 formado por los dos puntos que conforman la recta que aproxima la trayectoria. El segundo vector l_2 formado por el



Capítulo 3

Figura 24: Representación cálculo de la altura

punto inicial P_i de la recta que aproxima la trayectoria y el punto que forma la intersección recta-circunferencia en 2 dimensiones situado a la altura del punto anterior P_s^{\wedge} (figura 24).

Las variables de la figura se definen de la siguiente manera:

- P_i : punto inicial proveniente de la trayectoria que conforma el vector que aproxima el recorrido.
- P_f : punto final proveniente de la trayectoria que conforma el vector que aproxima el recorrido.
- P_s : punto con la altura deseada.
- P_s^{\wedge} : punto intersección entre la recta y la circunferencia situado a la misma altura que P_i .
- l_1 : vector que aproxima el recorrido.
- l_2 : vector formado desde el punto inicial del vector recorrido hasta la intersección recta circunferencia.
- h : altura deseada.

Con ambos vectores del espacio se puede calcular la altura del punto fácilmente aplicando reglas trigonométricas. El valor de salida de la función es el valor z de la intersección.

Consideraciones

La intersección entre una recta y una circunferencia puede acarrear dos inconveniencias que hay que tener en cuenta:

1. Que las rectas sean demasiado pequeñas.
2. Al resolver el problema con la intersección recta-circunferencia, existen 3 soluciones posibles.

Desarrollando el primer concepto, es importante que las rectas no tengan menor longitud que la distancia LA dado que la recta puede quedar contenida dentro de la circunferencia. En su contra, también interesa que estas tengan menor longitud posible para obtener una mayor precisión en la aproximación de la trayectoria. Se ha desarrollado una solución para que las rectas tengan la menor longitud posible a pesar de que puedan quedar contenidas dentro de la distancia LA sin perder el punto objetivo.

Capítulo 3

Esta solución consiste en generar rectas hasta que una interseca con la circunferencia, generando rectas acorde con la intersección a medida que se avanza a lo largo de la trayectoria.

Desarrollando el segundo concepto, cuando se realiza la intersección entre un punto y una recta pueden ocurrir tres cosas:

1. Que la intersección recta-circunferencia de como solución dos intersecciones posibles.
2. Que la intersección recta-circunferencia de como solución una sola intersección.
3. Que la intersección recta-circunferencia no interseca.

En el caso de 1) y 2), se ha aplicado en el algoritmo un sistema de evaluación para determinar el punto que va a favor del movimiento o que interseca con la recta como se muestra en la siguiente figura.

```
Si [primera raíz >= 0 y primera raíz <= 1]
    Selección del punto objetivo
Fin Si
Si [segunda raíz >= 0 y segunda raíz 2 <= 1]
    Selección del punto objetivo
Fin Si
```

Figura 25: Criterio de selección de raíces

La primera raíz es la raíz más pequeña de todas, por tanto, se evalúa primero. Si está dentro de los márgenes de la solución, se actualiza el punto objetivo, si no, se realiza el mismo procedimiento con la segunda raíz.

En el caso 3), Matlab da un mensaje de error dado que no se evalúa una respuesta ante este caso en el código. Este caso se evita en el planteamiento presentado anteriormente en este apartado.

Capítulo 3

3.3.4. ERROR DE TRAMO CRUZADO

Como procede en [8], el error de tramo cruzado (EPC, en inglés, *Crosstrack error*), se puede definir como la distancia entre el vector de dirección y el punto objetivo. Dicho error viene representado en la siguiente figura.

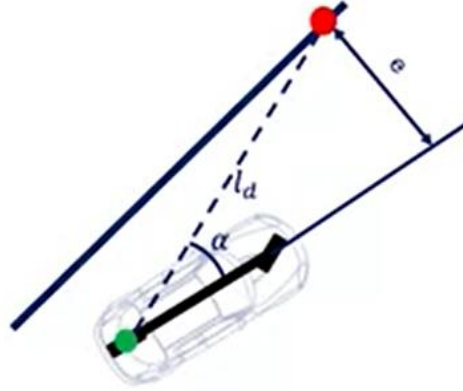


Figura 26: Representación gráfica EPC [8]

donde:

- e : error de pista cruzada.
- l_d : distancia LA.
- α : es el ángulo que forma la dirección del coche con el vector de distancia l_d .

De aquí, se pueden sacar las siguientes expresiones:

$$\left. \begin{aligned} \sin \alpha &= \frac{e}{l_d} \quad [4.9] \\ K &= \frac{2 \sin \alpha}{l_d} \quad [4.10] \end{aligned} \right\} K = \frac{2}{l_d^2} e \quad [4.11]$$

Combinando esto con la expresión para la curvatura 4.11, se puede llegar a la conclusión que la curvatura de la ruta creada por el controlador PP es proporcional al EPC en el punto LA. Es decir, a medida que aumenta el error, también lo hace la curvatura, trayendo al vehículo de vuelta a la ruta de manera más agresiva.

3.4. CONTROL DE FORMACIONES

Para controlar el comportamiento del grupo de AUVs, se ha decidido seguir la estrategia del líder-seguidor. Se ha seleccionado esta estrategia de control dado que solo es necesario que un robot siga la trayectoria y por la ausencia de obstáculos.

Este subcapítulo detalla el planteamiento y el diseño empleado para crear y mantener la formación, apartado 3.4.1. Continúa con el desarrollo del controlador empleado para los robots seguidores, apartado 3.4.2.

3.4.1. DISEÑO DE LA FORMACIÓN

Como se ha visto en el estado del arte, el tipo de formación es importante y esta depende de la misión y del tipo de terreno en el que se opere. Como el terreno modelado del suelo marino carece de obstáculos y los AUVs van a estar a una altura segura, la formación puede mantenerse fija evitando cambiar a otro tipo de formación por circunstancias del terreno. Como la misión es el barrido de una extensión, se ha optado por adoptar una formación triangular de tal forma que se pueda abarcar una considerable extensión sin dificultar el seguimiento de los AUVs seguidores. En dicha formación, el líder queda en la posición más avanzada para poder seguir la trayectoria y los seguidores toman las posiciones posteriores para poder seguir al líder. Esto queda reflejado en la siguiente figura.

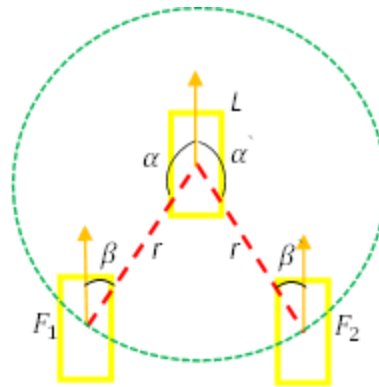


Figura 27: Representación gráfica de la formación

donde:

- α : ángulo formado entre la dirección del AUV líder con el vector que une al robot líder con el seguidor.

Capítulo 3

- β : ángulo formado entre la dirección del AUV seguidor con el vector que une al robot líder con el seguidor.
- r : distancia que los robots seguidores deben mantener.

La idea de esta formación es mantener a los robots seguidores a una distancia del líder, por ende, esta distancia se mantendrá como el radio de una circunferencia donde los seguidores están en una posición fija de esta misma. Para conseguir esto, se emplea la ecuación paramétrica de la esfera dado que el movimiento se ejerce en tres direcciones del espacio.

$$\begin{aligned}x &= r \cos(u) \sin(v), \\y &= r \sin(u) \sin(v), \\z &= r \cos(v)\end{aligned}\tag{5.1}$$

donde:

- r : es la distancia deseada de separación entre el líder y los seguidores.
- u : ángulo orientación líder respecto al eje z
- v : ángulo orientación líder respecto al eje y .

Situando el centro de la esfera en la posición del líder (C_x, C_y, C_z) y añadiendo un ángulo μ se puede determinar la posición de los AUVs seguidores, modificando la fórmula de la siguiente manera:

$$\begin{aligned}x &= C_x + r \cos(u + \mu) \sin(v), \\y &= C_y + r \sin(u + \mu) \sin(v), \\z &= C_z + r \cos(v)\end{aligned}\tag{5.2}$$

De esta manera, si el líder cambia de posición o de orientación en cualquier dirección del espacio, esto se verá afectado y actualizará las nuevas posiciones de los seguidores. Para mantener esta formación, es necesario establecer un control para poder ajustar el cambio de posición y orientación de los robots seguidores a medida que el líder sigue la trayectoria. Dicho control viene desarrollado en el siguiente apartado.

3.4.2. CINEMÁTICA DIFERENCIAL

Aprovechando la cinemática del robot desarrollada en el capítulo 3.2, se ha decidido aplicar un control diferencial basado en la cinemática para poder controlar la

Capítulo 3

posición y orientación de los robots seguidores. La cinemática diferencial describe la relación de velocidades con la postura del robot. Con ello, se puede establecer un sistema de control que puede establecer qué velocidades son necesarias para conseguir que el AUV tenga una posición y orientación concreta, Como queda reflejado en la figura 28.

Para realizar el movimiento del robot, se ha empleado anteriormente la matriz Jacobiana; donde dicha matriz relaciona dos vectores de velocidades expresados en un espacio distinto.

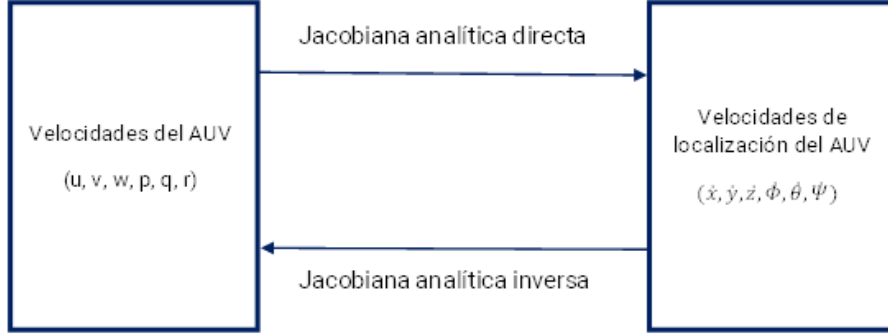


Figura 28: Relación de la matriz Jacobiana

Las matrices respectivas del AUV son las siguientes (obtenidas en el capítulo 3):

$$R_a^n = R_{z,\psi} * R_{y,\theta} * R_{x,\phi} = \begin{bmatrix} \cos \psi \cos \theta & -\sin \psi \cos \theta + \cos \psi \sin \theta \sin \phi & \sin \psi \sin \phi + \cos \psi \cos \phi \sin \theta \\ \sin \psi \cos \theta & \cos \psi \cos \theta + \sin \phi \sin \theta \sin \psi & -\cos \psi \sin \phi + \sin \theta \sin \psi \cos \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad [5.3]$$

$$T_\theta = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix} \quad [5.4]$$

Si se invierten estas matrices y multiplicando por el error de posición y orientación respectivo en cada eje coordenado, como se hace en [19], se obtiene:

$$v_o^a = (R_a^n)^{-1} e_p \quad [5.5]$$

$$\omega_a^a = (T_\theta)^{-1} e_o \quad [5.6]$$

donde:

- v_o^a : vector de velocidades lineales resultante.
- ω_a^a : vector de velocidades angulares resultante.

Capítulo 3

- $(R_a^n)^{-1}$: la inversa de la matriz de transformación que relaciona las velocidades lineales del cuerpo con el vector de velocidades de los ángulos de Euler.
- $(T_\theta)^{-1}$: la inversa de la matriz de transformación que relaciona las velocidades angulares del cuerpo con el vector de velocidades de los ángulos de Euler.
- e_p : error de posición.
- e_o : error de orientación.

El resultado de esta operación son los vectores de velocidades lineales y angulares para poder alcanzar un estado dado. Si se aplica la cinemática directa a dichas velocidades, se puede establecer las posiciones respectivas.

Capítulo 4. RESULTADOS

En esta sección se muestra los resultados obtenidos de las distintas áreas del proyecto. El capítulo comienza enseñando los resultados obtenidos para el ajuste cinemático, apartado 4.1. Se desarrolla el trabajo experimental para ajustar la distancia LA, apartado 4.2. Se finaliza el trabajo experimental mostrando los resultados obtenidos con el control de formación, apartado 4.3.

4.1. AJUSTE DE VALORES CINEMÁTICOS

A fin de establecer un criterio para determinar un ángulo de giro para el AUV, se han establecido unos valores fijos para poder realizar este desarrollo experimental. Uno de ellos es la velocidad, que se fija en 25 km/h, basándose en la velocidad media que pueden alcanzar ciertos vehículos submarinos. Para ajustar los valores de simulación, se han realizado las siguientes consideraciones para obtener dimensiones.

El planteamiento ha sido el siguiente, para una velocidad de 25 Km/h se tardaría un tiempo de 11.4s aproximadamente para recorrer una distancia de 80 metros de longitud. Estableciendo en el código un valor de velocidad de $v=7$ m/s (que es aproximadamente 25 Km/h), con un tiempo de simulación de $t=0,1$ s y una trayectoria de 80 voxels de longitud en línea recta. Se obtiene que el tiempo total de simulación es de $t_t = 11.2$ s. Por tanto, se puede establecer que un voxel es equivalente a 1 metro en la simulación. La simulación queda mostrada en la siguiente figura.

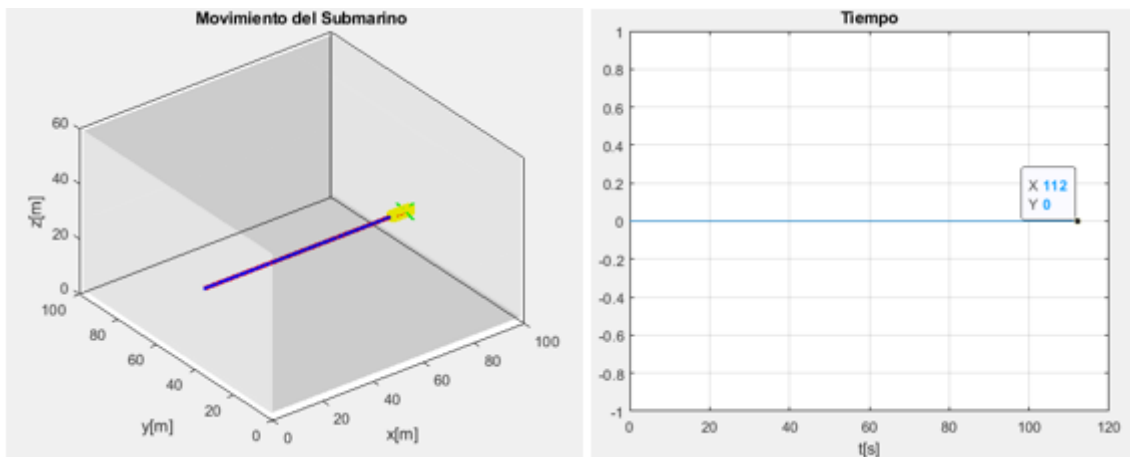


Figura 29: Simulación realiza para obtener las dimensiones de la simulación.

Para establecer un criterio de ángulo de giro, se ha realizado un recorrido con curvas de 90, 60, 45 y 30 grados para ver cómo responde el AUV ante los distintas inclinaciones. Estas simulaciones se han realizado con una distancia de punto de mira de 8 metros.

Capítulo 4

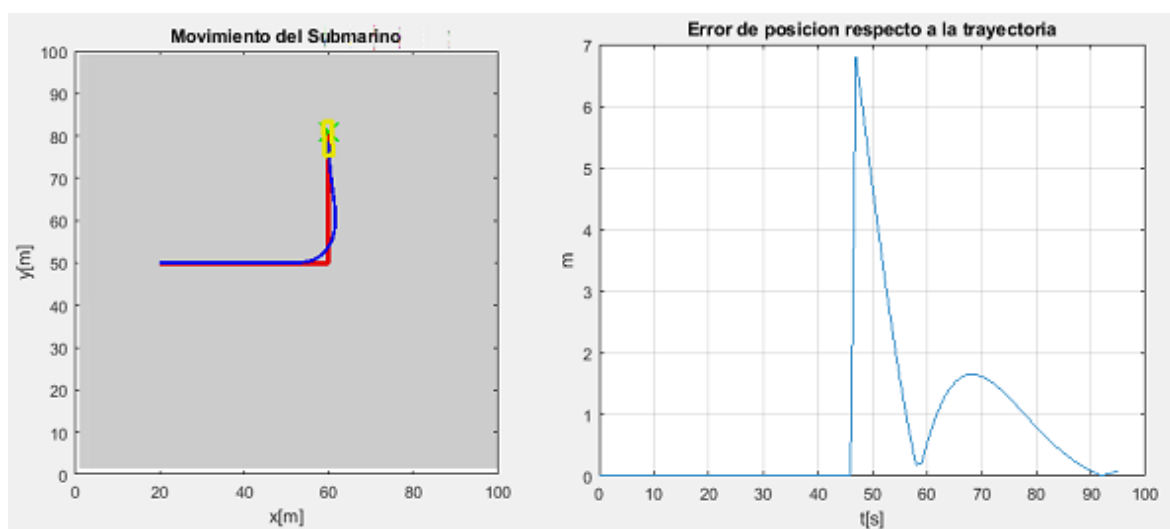


Figura 30: Simulación de giro de 90° y error de posición respectivo

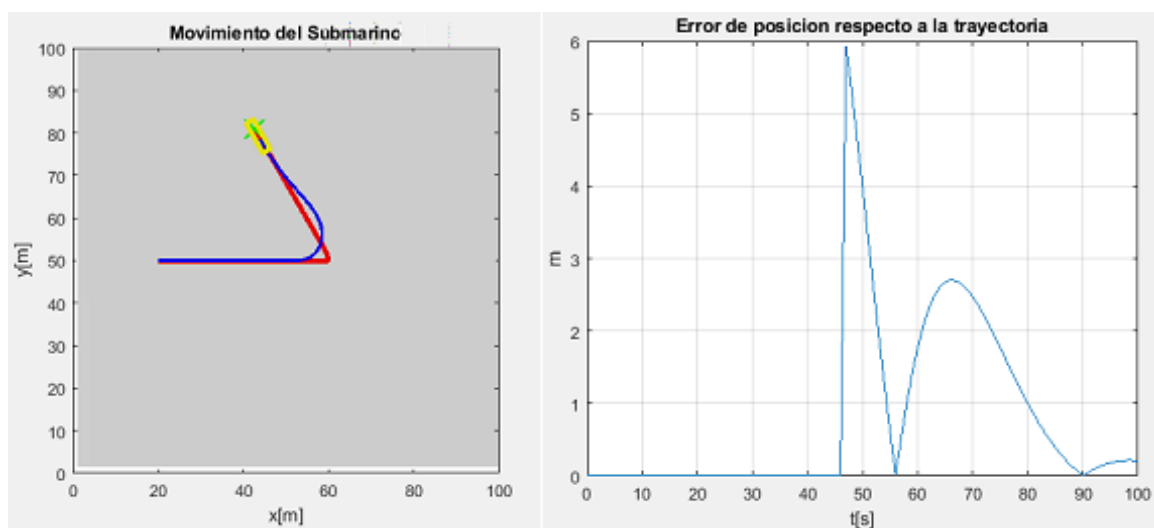


Figura 31: Simulación de giro de 60° y error de posición respectivo

Capítulo 4

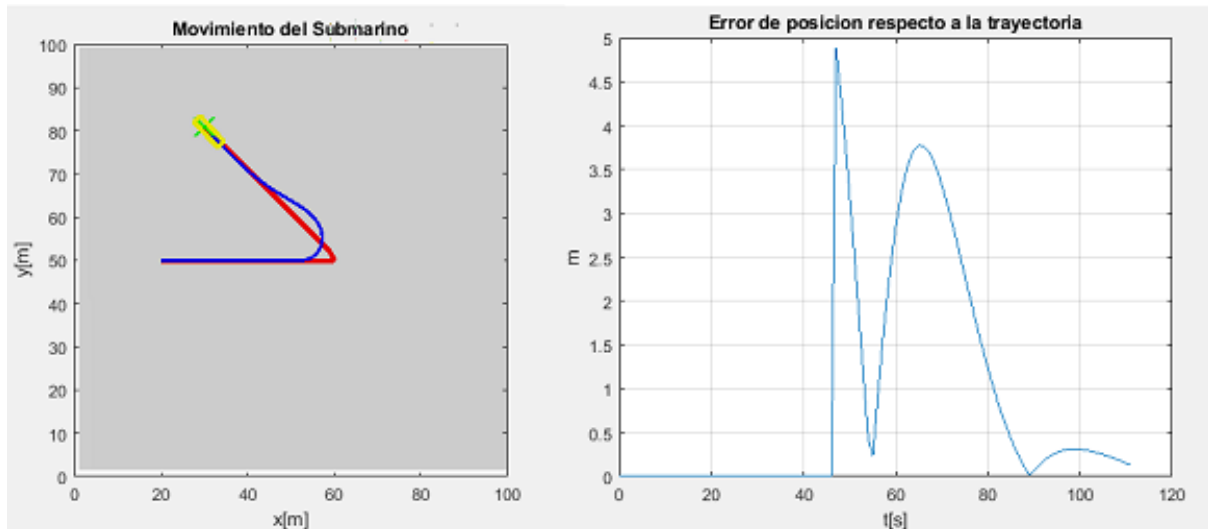


Figura 32: Simulación de giro de 45° y error de posición respectivo

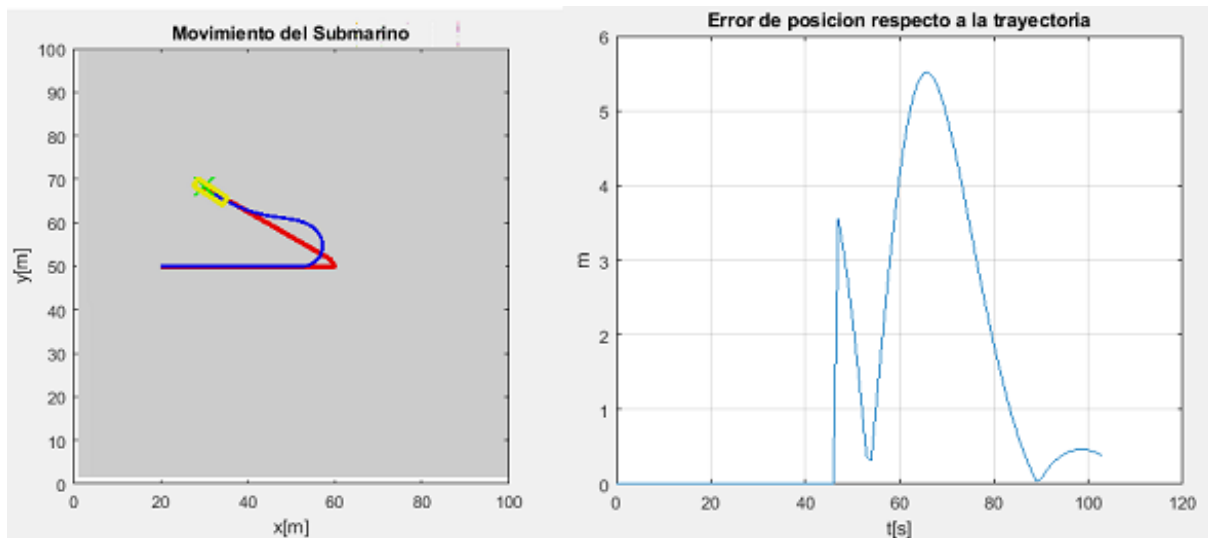


Figura 33: Simulación de giro de 30° y error de posición respectivo

Como se pueden observar en las figuras 29-32, se ha representado la simulaciones respectivas junto a una gráfica del error de posición. Para obtener el error de posición, se ha escogido el punto inicial y final de la primera recta que conforma la inclinación y se ha generado un vector. Se ha generado un segundo vector desde el punto inicial de la recta hasta la posición del robot. Se ha proyectado este último vector en el primer vector y se le ha sumado la posición del punto inicial de la recta. El resultado de todas estas operaciones es la proyección de la posición del AUV en la recta. Se ha realizado el mismo procedimiento para la otra recta que conforma la inclinación. En todas las gráficas se puede observar un pico de error, esto se debe a que se produce un cambio en la proyección de las rectas cuando el punto de mira cambia de recta. El verdadero error se mide en la curva posterior. En las figuras de las simulaciones se

Capítulo 4

puede observar que siempre existe una pequeña curva que sobresale de la inclinación cuando efectúa el giro, esto se debe al comportamiento natural del algoritmo PP que calcula el seguimiento de trayectoria generando trayectorias curvilíneas.

Partiendo de estas aclaraciones y de los resultados, se puede concluir que el robot efectúa giros óptimos sin mucho desvío de error de posición hasta giros de 60° de inclinación, donde la desviación llega hasta un tope de 2,5 metros. A partir de dicha inclinación, los valores obtenidos se consideran demasiado grandes como para determinar que el giro ha sido óptimo.

4.2. PERSECUCIÓN PURA

Como se ha explicado en el apartado 3.3.2, el valor de la distancia LA es un parámetro importante para que el algoritmo funcione correctamente y el robot siga la trayectoria deseada fielmente. Se han realizado las siguientes simulaciones para poder determinar un valor óptimo. Las distancias LA empleadas han sido 2, 10 y 21 metros respecto al centro del AUV. Se han simulado los distintos valores en tres trayectorias distintas: una curva en 2D, una espiral y finalmente un barrido de zona con relieve. Cabe mencionar que se han realizado más simulaciones de las mostradas en este apartado con el fin de ajustar dicho valor.

Antes de comenzar el análisis de las gráficas, hay que mencionar que el error de posición mostrado corresponde a la distancia entre cada punto de la trayectoria ejecutada por el AUV respecto al punto más cercano en la trayectoria calculada. Como la trayectoria está discretizada en coordenadas enteras y la posición de AUV están calculadas en coordenadas reales, el resultado produce picos que indican los saltos finitos de los puntos discretizados. Cabe mencionar también que la diferencia de orientación respecto al punto objetivo es la diferencia angular que existe entre la tangente formada desde el centro del submarino hasta el punto objetivo con la orientación del robot. En el análisis, se denominará como “error de orientación”. Partiendo de esta aclaración, se procede a realizar el desarrollo experimental.

Capítulo 4

4.2.1. CASO 1: VALOR DE DISTANCIA LA 2.

Curva

El primer análisis consistirá en una trayectoria sencilla formada por una curva en un plano para ver el comportamiento del robot en solo dos dimensiones.

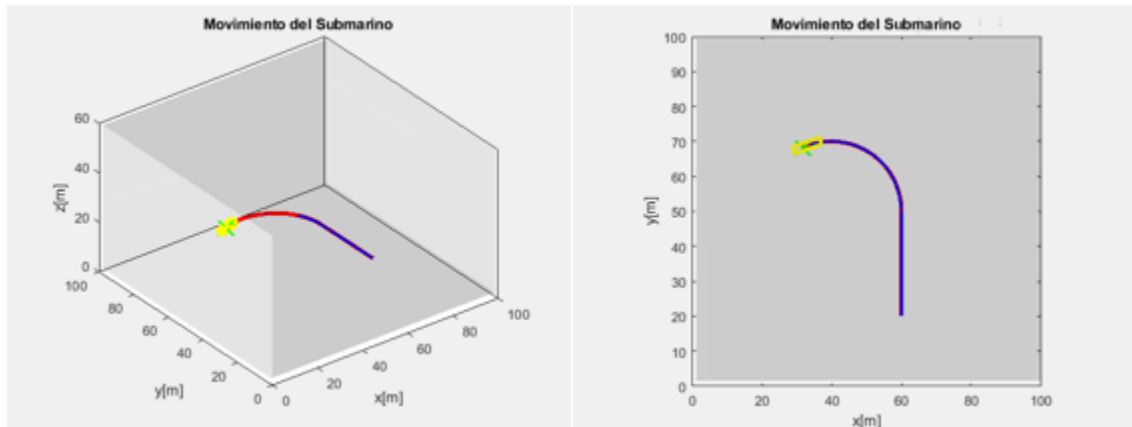


Figura 34: Simulación curva 2D con parámetro LA de 2.

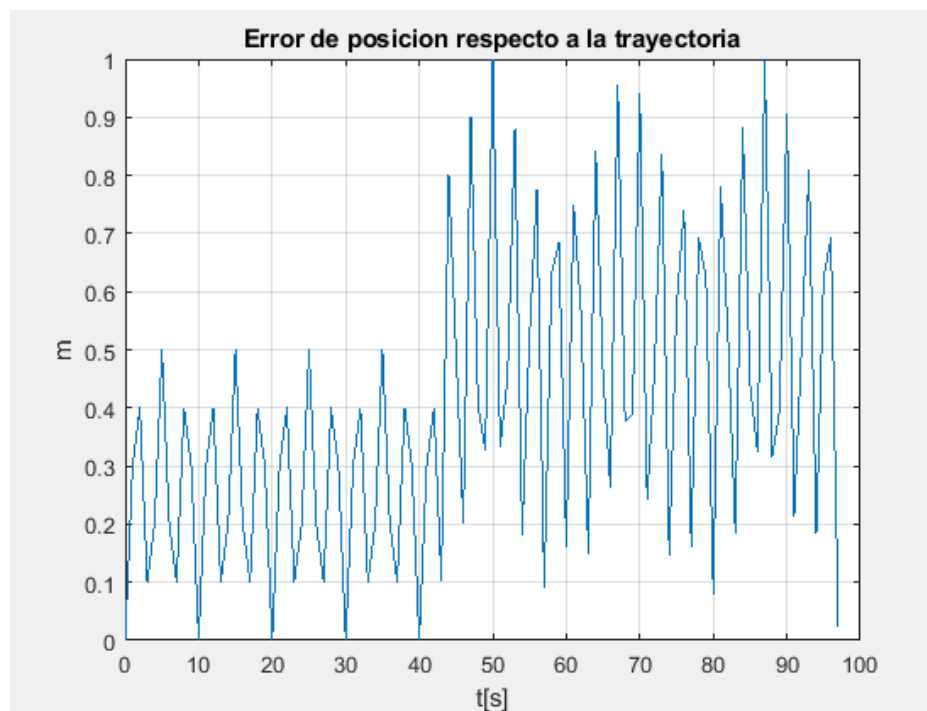


Figura 35: Error de posición respecto a la trayectoria en curva plana.



Figura 36: Error de orientación respecto al punto objetivo.

Como se puede ver en los resultados obtenidos, el error de posición es muy pequeño dado que el seguimiento de la trayectoria es casi exacto estando el punto tan cerca. En contra partida, se puede observar como el robot oscila cuando la trayectoria comienza a girar, realizando giros y correcciones de orientación bruscas y poco estables. Cabe mencionar que la polaridad del ángulo depende de la dirección de giro que ha tomado el AUV. Es decir, los giros realizados en sentido antihorario son los valores positivos en la gráfica y los giros realizados en sentido horario son los ángulos negativos.

Capítulo 4

Espiral

En segundo lugar, se complicará un poco la trayectoria anterior añadiendo el elemento de la altura para así poder comprobar el comportamiento del robot en una trayectoria de tres dimensiones.

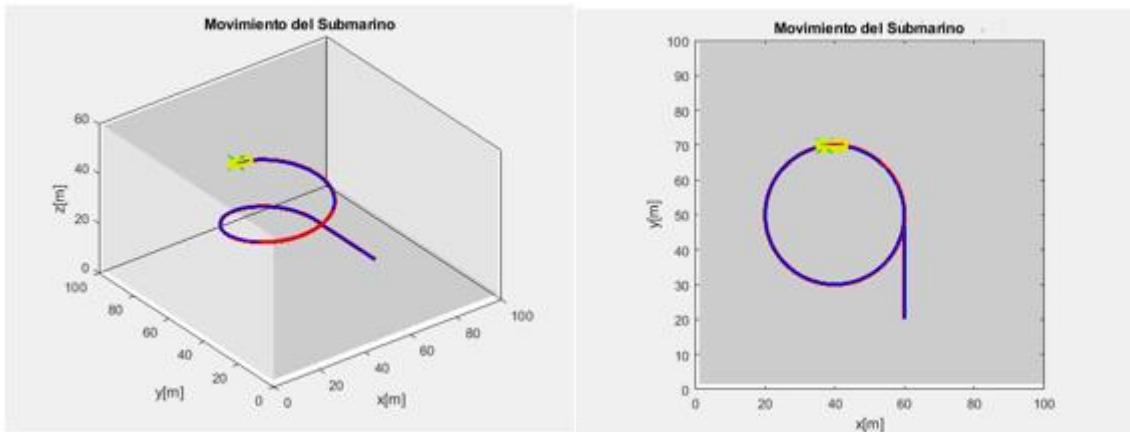


Figura 37: Simulación espiral con parámetro LA de 2.

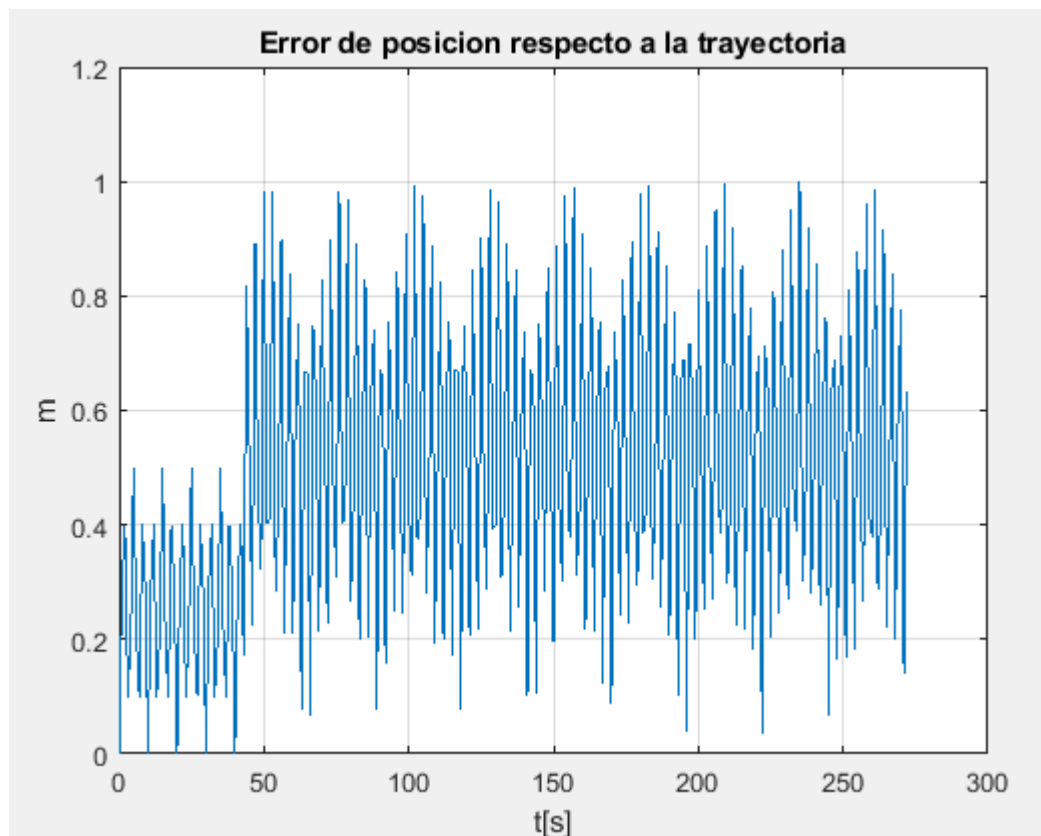


Figura 38: Error de posición respecto a la trayectoria en espiral.

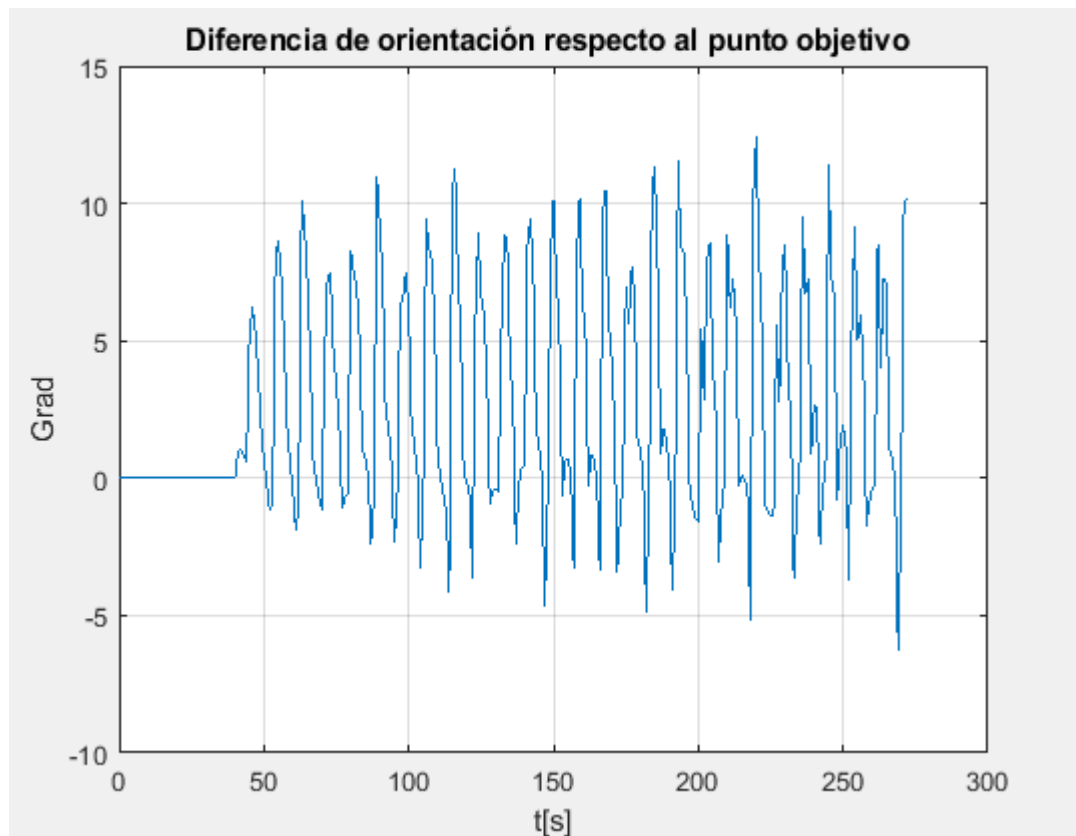


Figura 39: Error de orientación respecto al punto objetivo.

Los resultados obtenidos son muy parecidos a los de la simulación anterior y refuerzan la teoría expuesta en el apartado 3.3.2. El error de trayectoria sigue siendo muy pequeño dado que el seguimiento de la trayectoria es casi exacto, pero comprobando la gráfica de error de orientación, se puede apreciar mejor como el sistema se vuelve oscilatorio e inestable.

Barrido

Como se ha mencionado a lo largo del trabajo fin de grado, se ha seleccionado el barrido como trayectoria final inspirado en el algoritmo de la cortadora de césped añadiendo también como dificultad diferencias de altura adaptadas al relieve. Esta trayectoria es mucho más exigente que las otras expuestas, por ello, los resultados serán peores.

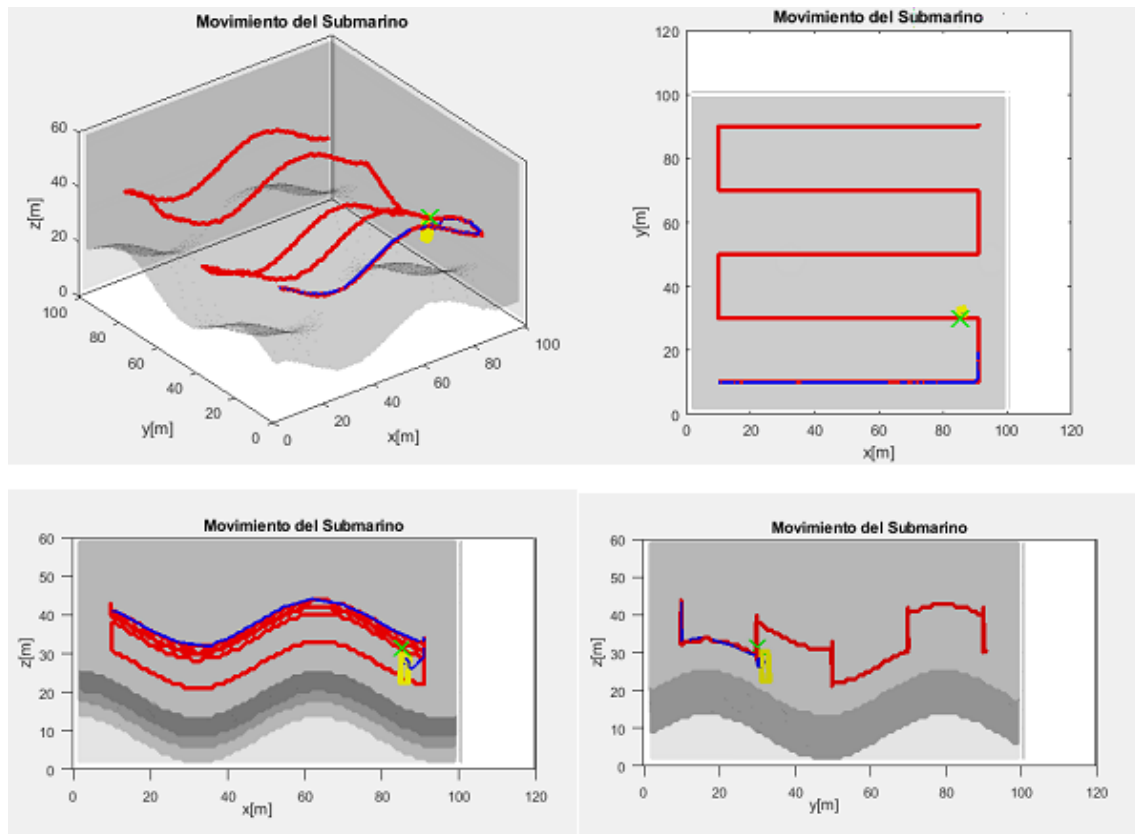


Figura 40: Simulación de barrido con parámetro LA de 2.

En esta última simulación se puede ver como el error de posición sigue siendo mínimo, pero el robot está todo el rato oscilando. Como se puede observar, el robot no consiguió cumplir el objetivo dado que el sistema se volvió inestable generando un giro muy brusco haciendo que el robot perdiese el punto objetivo, generando así un error de simulación.

Como no se completó la simulación, no se pudo proporcionar las gráficas de los resultados, ya que estas se generan una vez finaliza el recorrido satisfactoriamente.

Capítulo 4

4.2.2. Caso 2: Valor de distancia LA 10.

En esta simulación y las posteriores de este apartado, se ha ajustado el valor LA a un valor donde se han obtenido los mejores resultados de simulación.

Curva

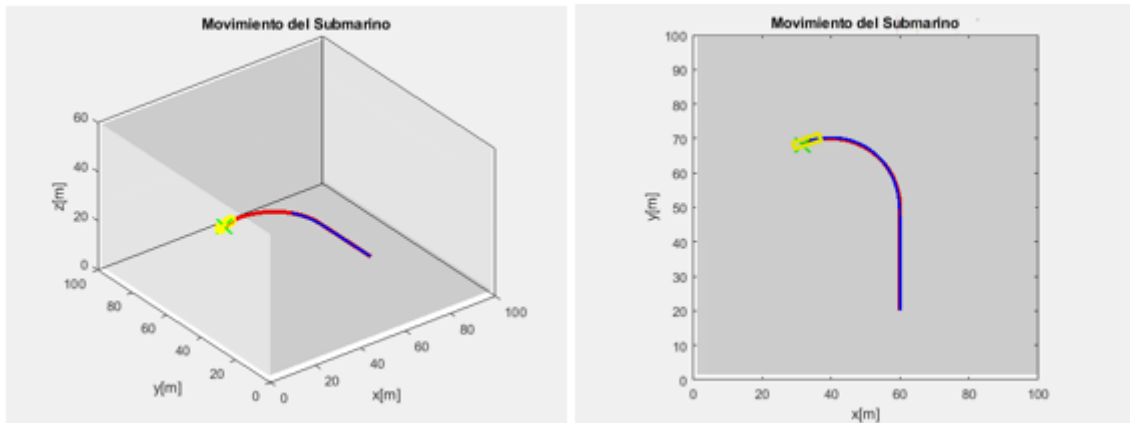


Figura 41: Simulación de curva 2D con parámetro LA de 10.

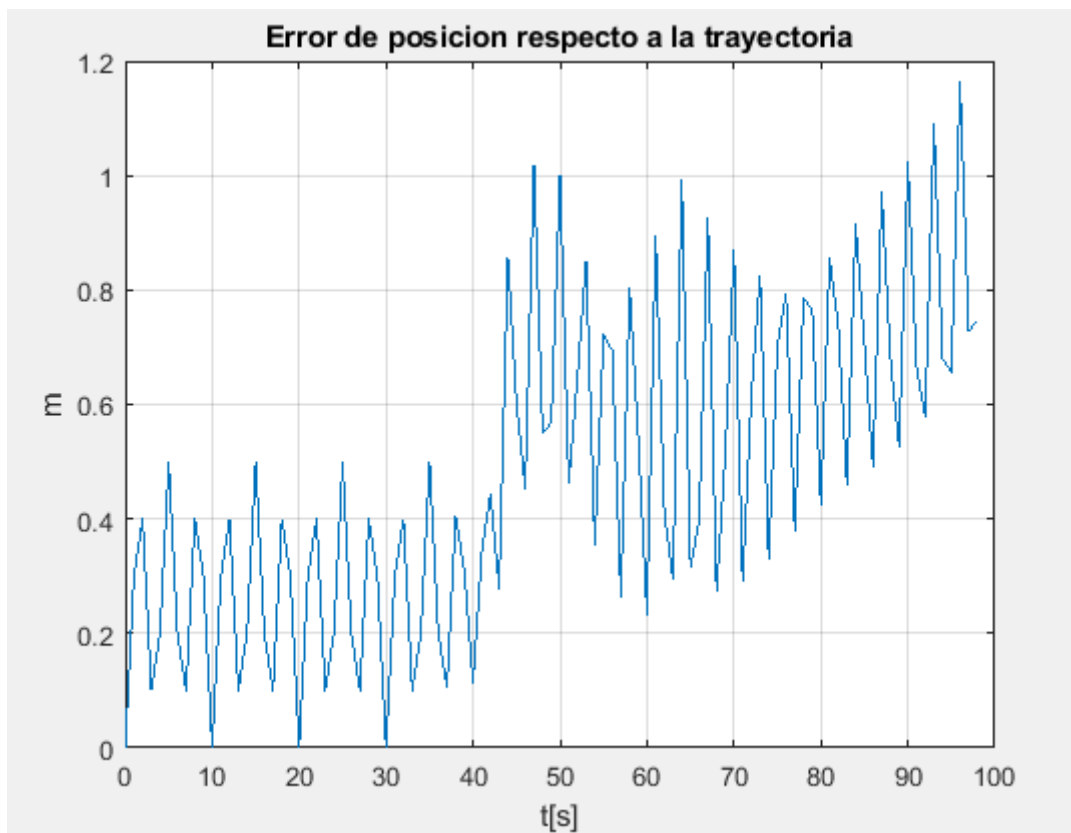


Figura 42: Error de posición respecto a la trayectoria en una curva plana.



Figura 43: Error de orientación respecto al punto objetivo.

Como se puede ver en los resultados de la simulación de esta trayectoria, el error de posición es un poco mayor que en el apartado anterior, pero sigue siendo muy bajo, realizando una buena aproximación a la trayectoria. A diferencia de la primera, el robot no oscila debido a que el punto objetivo se encuentra a una distancia suficientemente lejana. Esto queda reflejado en la gráfica porque no se alcanzan ángulos negativos. Las curvaturas presentes representan al robot intentando ajustar su orientación a 0 con respecto al punto objetivo. Se puede ver cómo el robot intenta corregir un error de aproximadamente 14° de error a lo largo del recorrido.

Capítulo 4

Espiral

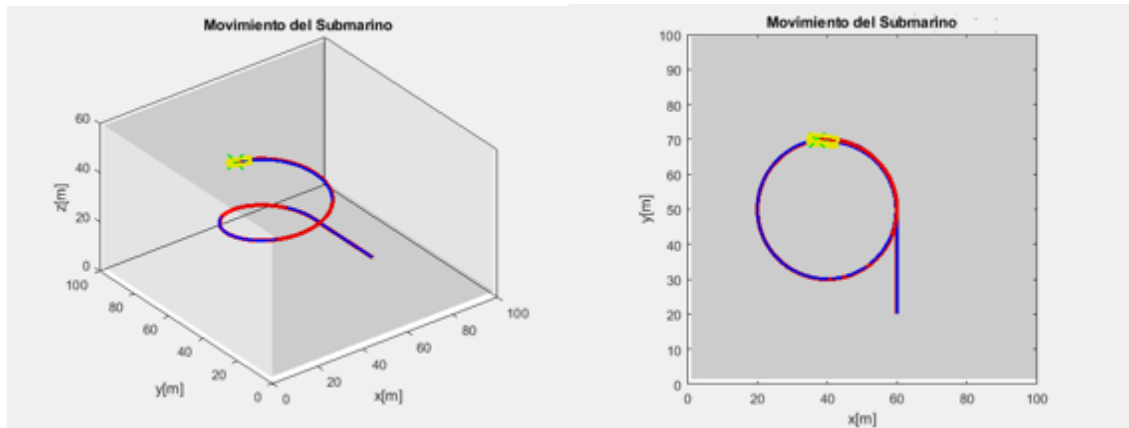


Figura 44: Simulación de espiral con parámetro LA de 10.

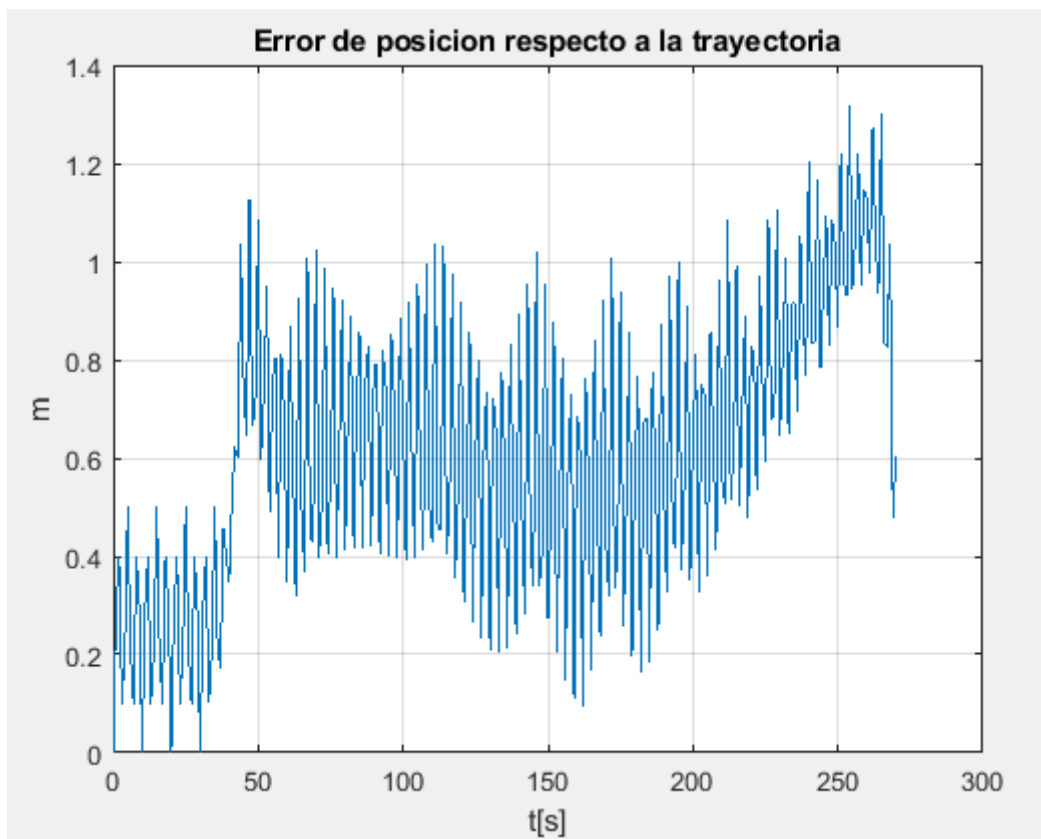


Figura 45: Error de posición respecto a la trayectoria en espiral.

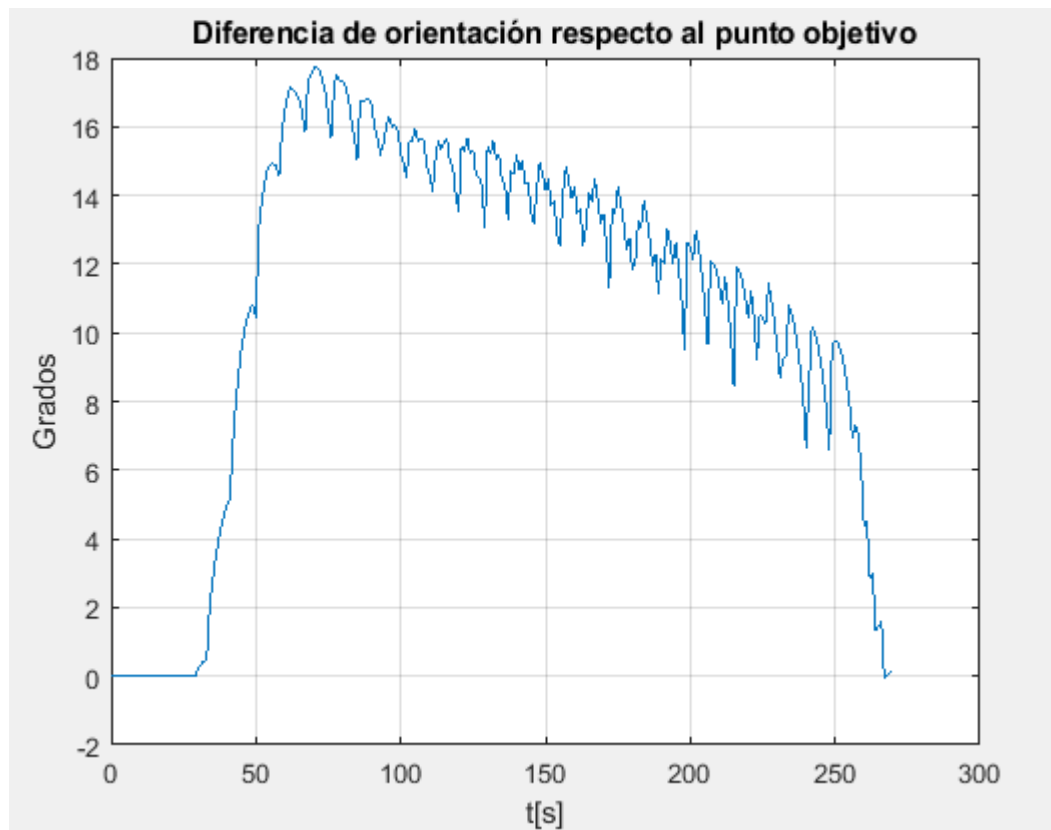


Figura 46: Error de orientación respecto al punto objetivo.

En esta simulación se puede comprobar como el error de posición de la trayectoria es bueno, rondando el valor máximo en torno a 1.2 metros y quedando casi en la posición exacta del punto objetivo. En cuanto al error de orientación, alcanza un máximo de 18° de error, pero se puede apreciar como el AUV va corrigiendo poco a poco dicho error hasta quedando al final de la trayectoria con la posición y orientación adecuada.

Capítulo 4

Barrido

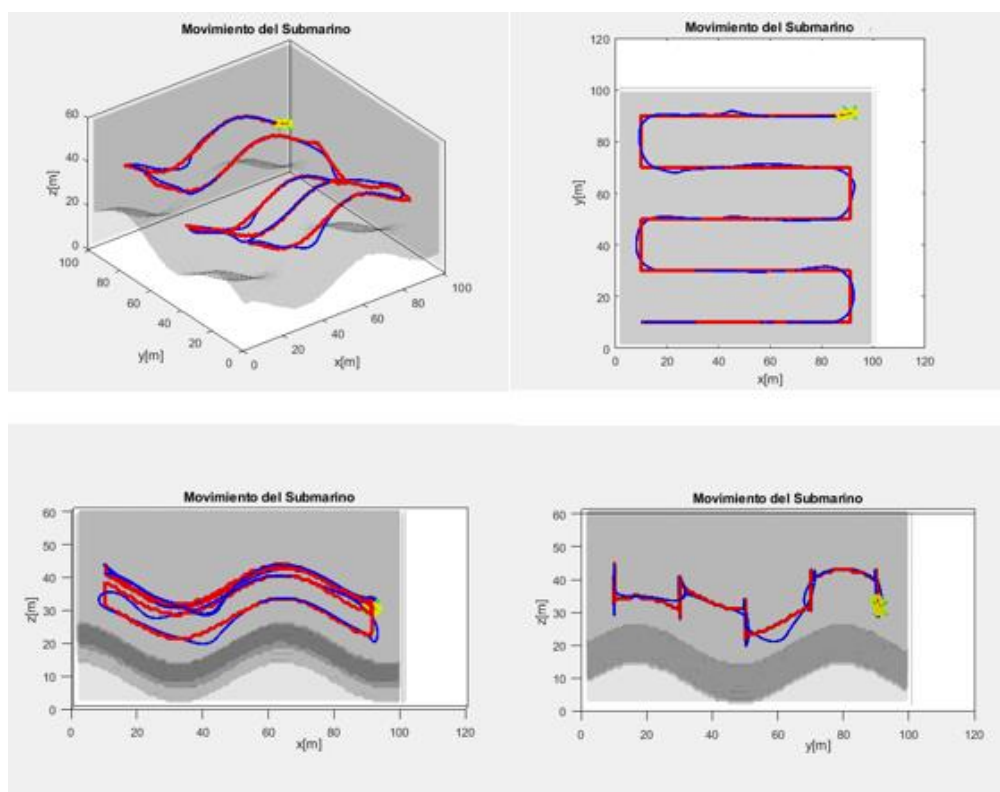


Figura 47: Simulación de barrido con parámetro LA de 10.

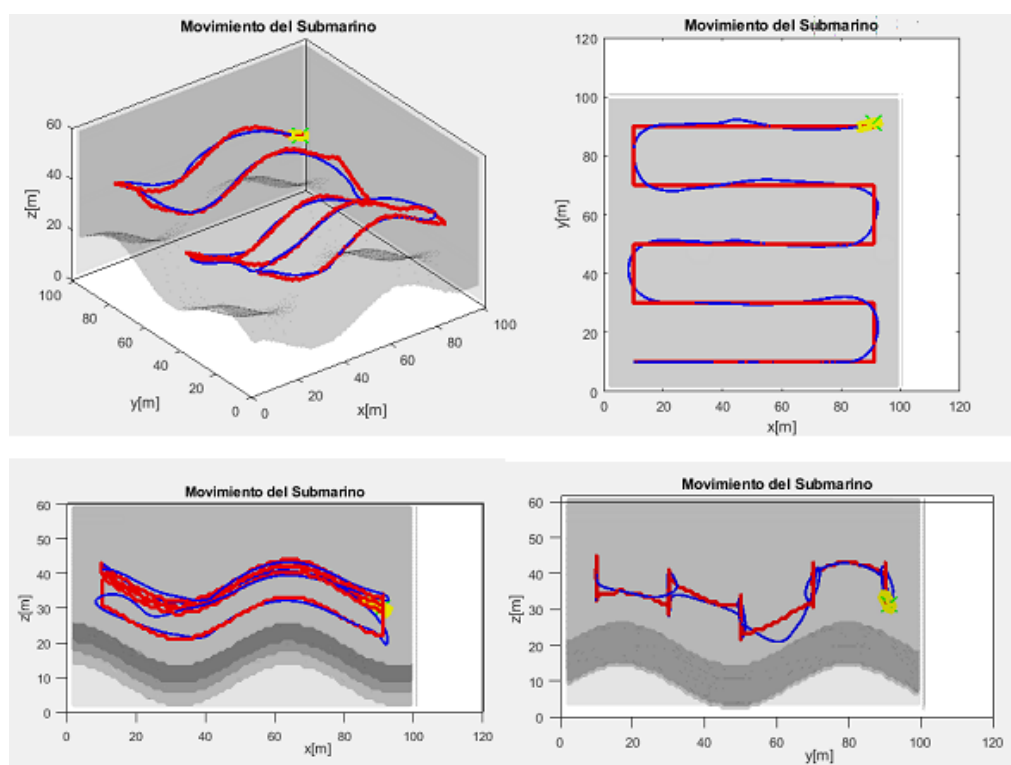


Figura 48: Simulación de barrido con parámetro LA de 10 aplicando un valor proporcional.

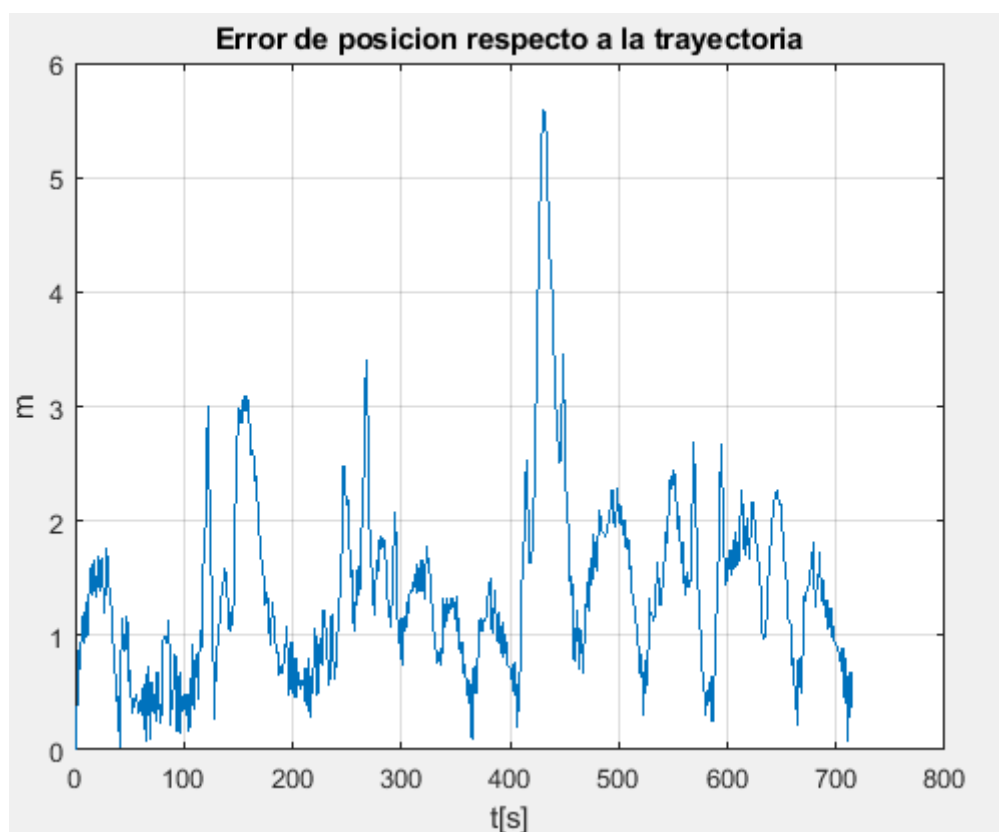


Figura 49: Error de posición respecto a la trayectoria de barrido.

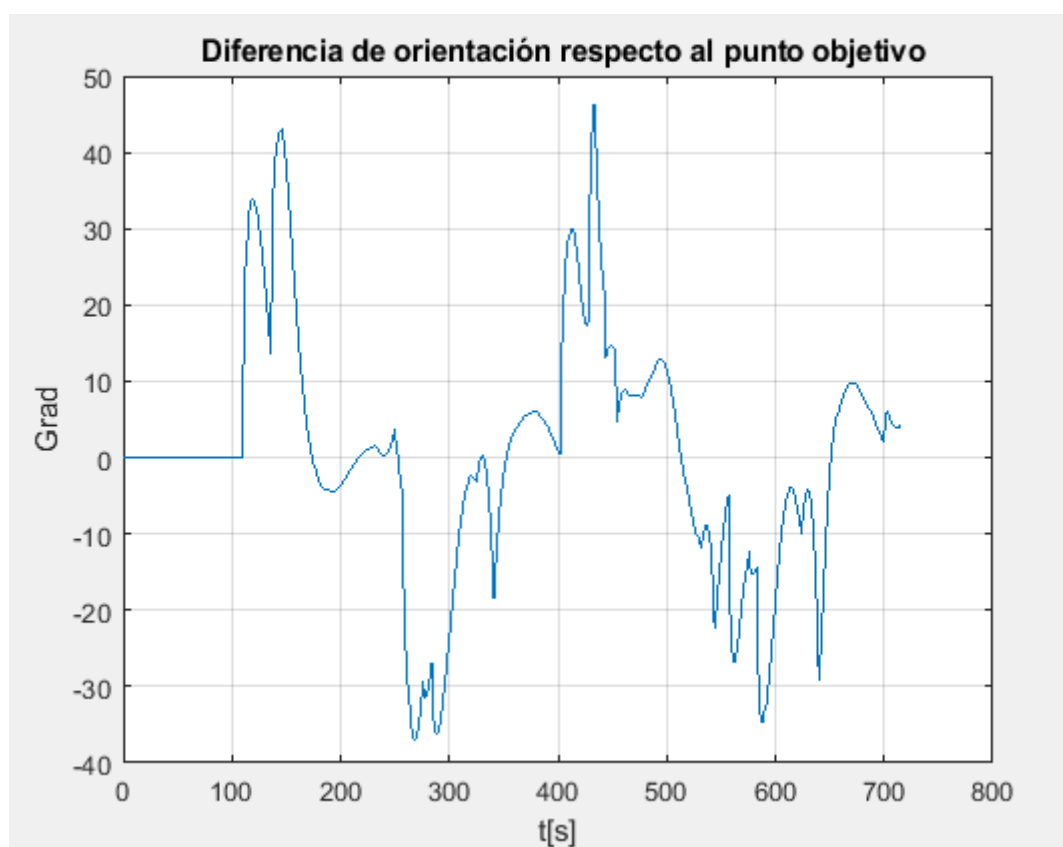


Figura 50: Error de orientación respecto al punto objetivo.

Capítulo 4

En primer lugar, se han introducido dos simulaciones para explicar que, el PP es un controlador proporcional y que, ajustando dicho valor proporcional (el valor proporcional es el valor que determina la reacción del error actual), se puede obtener una mejora de los resultados. Esto se puede apreciar en las figuras 46 y 47 donde ambas simulaciones tienen el mismo parámetro LA pero una ha sido ajustada con un valor proporcional de 1.5. Los resultados obtenidos son bastante notorios, se puede apreciar cómo solo viendo las imágenes, la segunda simulación realizada sigue el recorrido con mayor fidelidad que la primera simulación. Los resultados analizados a continuación han sido tomados partiendo de la segunda simulación.

Cogiendo de referencia las gráficas, se puede ver que los resultados obtenidos son los que mayor fluctuación tienen. Esto se debe principalmente a la complejidad de la trayectoria realizada. A pesar de ello, se puede observar que el AUV obtiene un buen resultado realizando esta trayectoria.

En la gráfica del error de posición se puede observar que el error en casi todo el tramo fluctúa entre el valor de 0 y 2 metros habiendo un pico máximo de error de unos 5.5 metros de distancia que se produce a la altura del giro de la tercera curva. Si se observa la segunda gráfica, lo primero a mencionar son los segundos picos bruscos de subida que se producen cuando aumenta el error de posición. Esto se debe a que se está calculando el error de posición respecto al punto de mira, cuando se está realizando el giro, el punto de mira realiza un cambio brusco de la línea vertical a la horizontal, aumentando la diferencia de ángulo que existe entre el punto y el robot. Todos los picos de error que se producen después de una bajada ocurren por este acontecimiento. Sabiendo esto, los errores de la gráfica son bastante grandes debido a que se están realizando giros muy bruscos, pero lo importante a analizar aquí es la capacidad que tiene el robot de reaccionar a estos cambios bruscos de orientación, que es bastante corto.

Se puede apreciar también como los picos de error en ambas gráficas coinciden. Existe una relación entre los dos errores, de cómo cuando se realiza un giro brusco, el error de orientación aumenta al igual que el error de posición haciendo que el robot tenga que corregir con mayor rapidez la trayectoria para no desviarse del recorrido.

Capítulo 4

4.2.3. CASO 3: VALOR DE DISTANCIA LA 21.

Curva

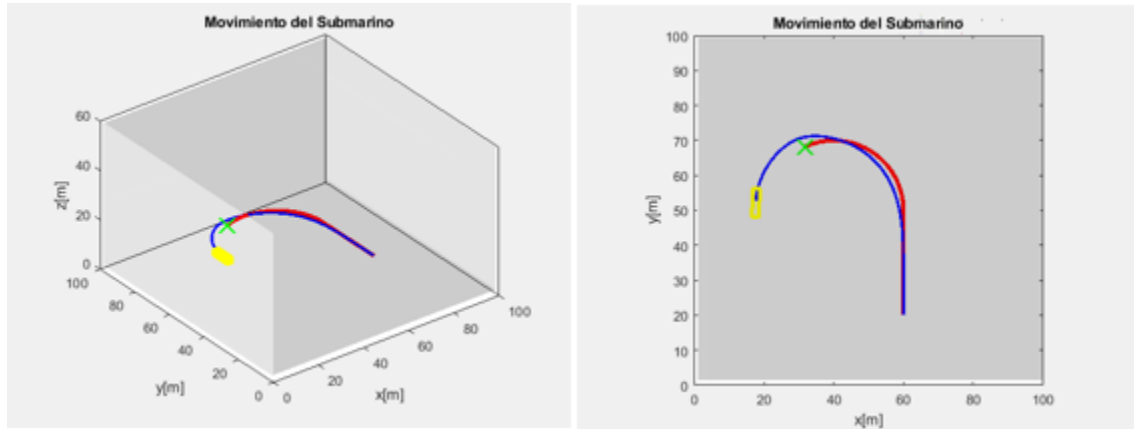


Figura 51: Simulación de curva 2D con parámetro LA de 21.

Como se puede observar en esta simulación, con una distancia muy grande del valor LA, se comete un mayor error de posición con respecto a las otras simulaciones realizadas. En este caso, el robot no consigue girar a tiempo para alcanzar el punto final generando un giro muy abierto hasta causar un fallo de simulación. Como no se ha finalizado la simulación correctamente, no se han podido proporcionar gráficas de la trayectoria.

Espiral

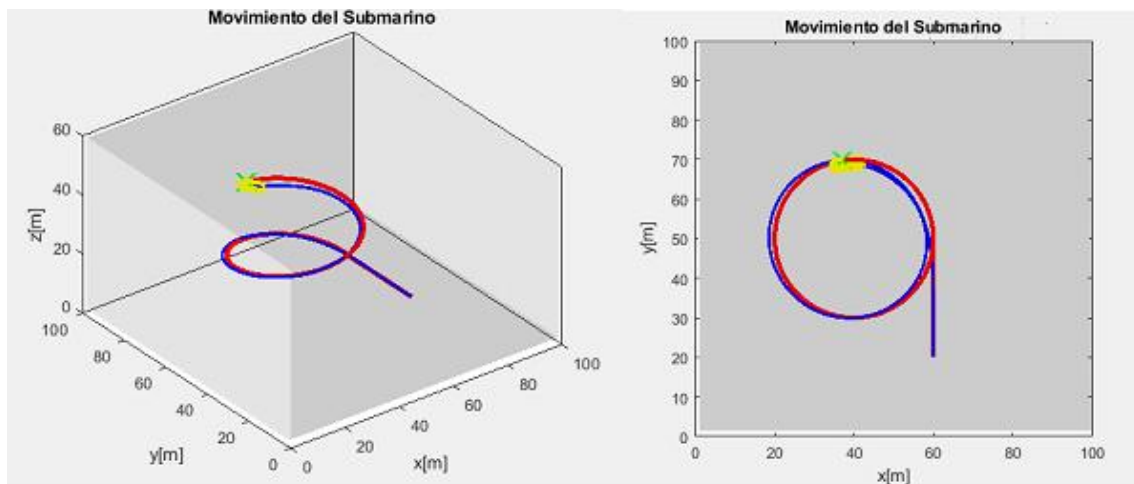


Figura 52: Simulación de espiral con parámetro LA de 21.

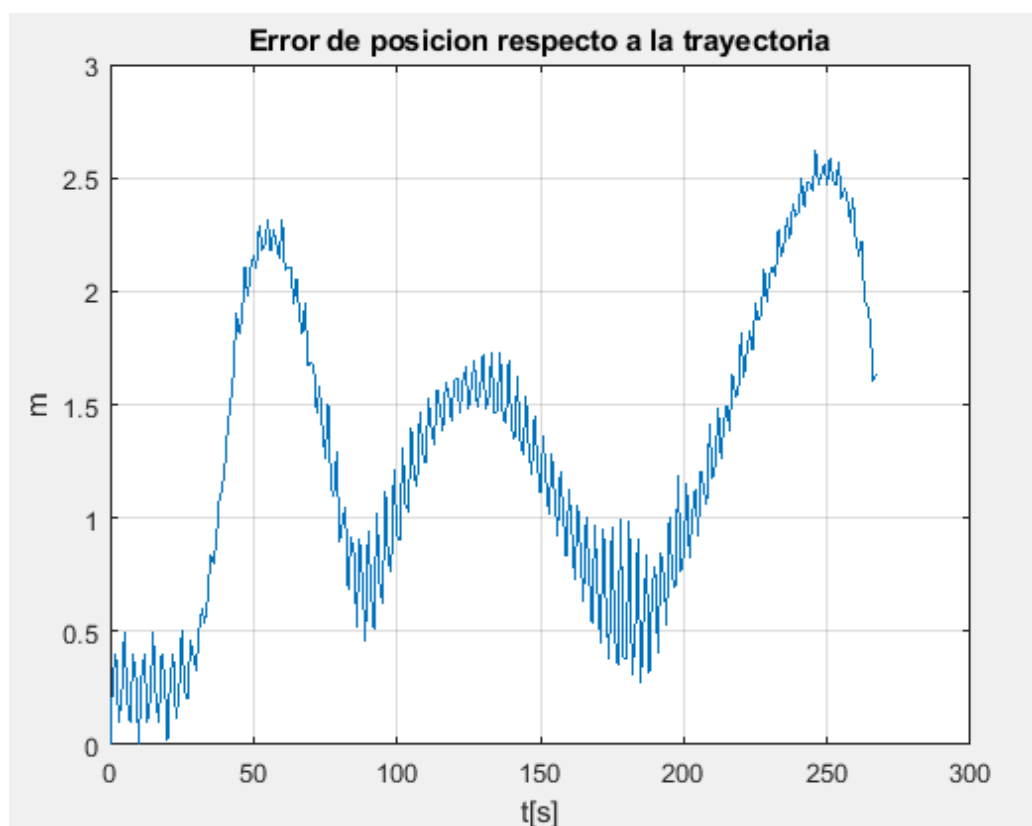


Figura 53: Error de posición respecto a la trayectoria en espiral.

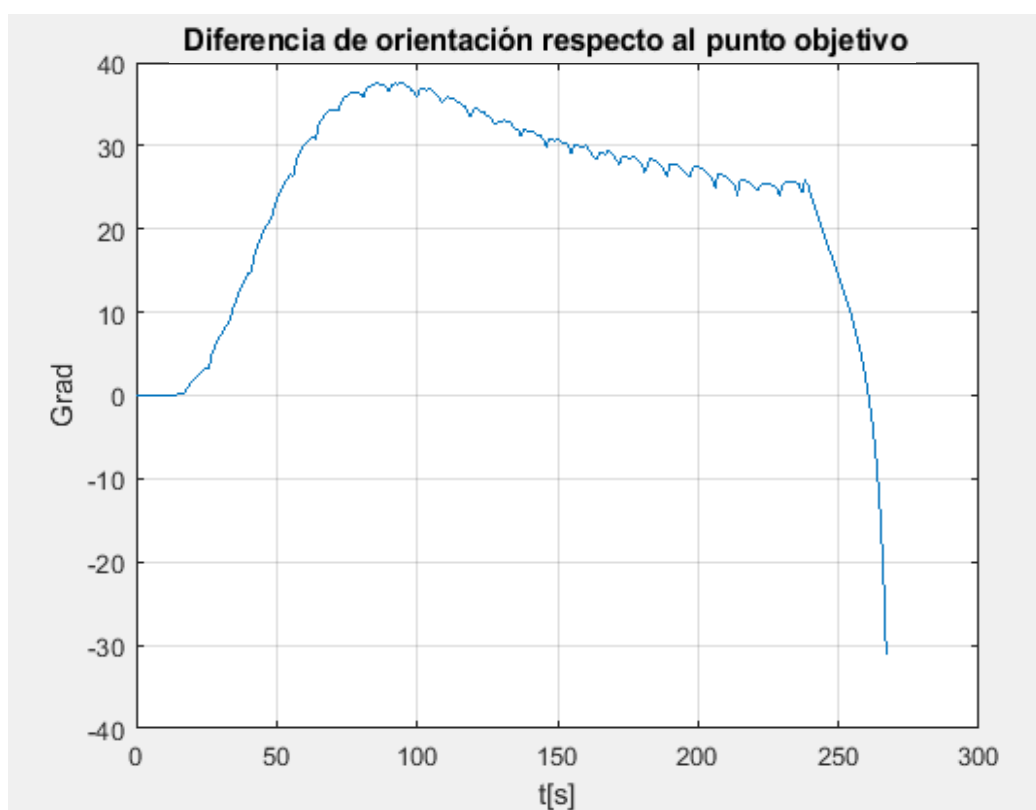


Figura 54: Error de orientación respecto al punto objetivo.

Capítulo 4

Esta segunda simulación con resultados gráficos refuerza la teoría vista en el apartado 2.3.2. Se puede observar que el AUV realiza giros más abiertos generando un mayor error de posición con respecto a la trayectoria. En el error de orientación, se puede ver como el giro es muy abierto con error máximo de aproximadamente 38° que disminuye levemente a medida que transcurre el tiempo donde el robot consigue acortar un poco dicho error. Finalmente, llega al punto final con una orientación poco adecuada de unos 30° de error.

Barrido

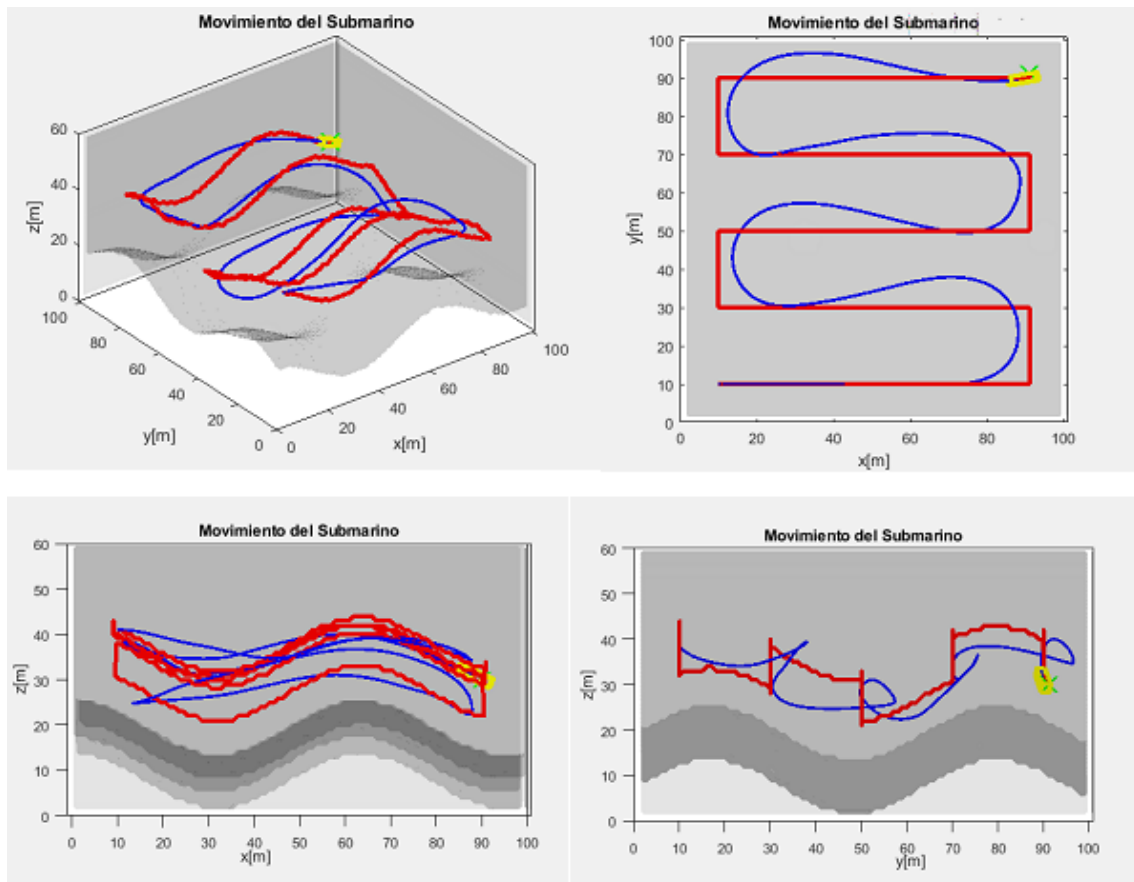


Figura 55: Simulación de barrido con parámetro LA de 21.



Figura 56: Error de posición respecto a la trayectoria en espiral.



Figura 57: Error de orientación respecto al punto objetivo.

Capítulo 4

En esta simulación final de barrido se puede determinar como el error de posición con respecto a la trayectoria es muy dispar, generando errores muy grandes de hasta 10 metros de error. Si comprobamos las imágenes de la simulación se puede ver como el AUV realiza los giros bruscos muy abiertos generando dichos picos en las gráficas anteriormente mencionados. Como era de esperar, el error de orientación también es muy elevado produciendo sus máximos en las curvas cerradas de la trayectoria donde alcanzan un valor de hasta 90° de error.

4.3. CONTROL DE FORMACIONES

A continuación, se adjuntan las imágenes de los resultados del control de formaciones de la estrategia líder-seguidor.

Como se ha visto en el apartado 2.4.2, se puede determinar el control de la formación a través del análisis de la distancia que ejercen los robots seguidores y el ángulo con respecto al líder, o a través de las distancias que ejercen los robots seguidores con el líder y entre ellos. Por ello, se ha decidido representar una primera gráfica donde se dibuja la distancia entre los robots seguidores y el líder, y una segunda donde se representa la distancia entre los robots seguidores.

La distancia deseada es que los robots seguidores se encuentren a una distancia de 8 metros respecto al líder y que mantengan una distancia de 12 metros entre ellos. Cabe mencionar que la distancia LA elegida para esta simulación ha sido de 10.

4.3.1. CURVA

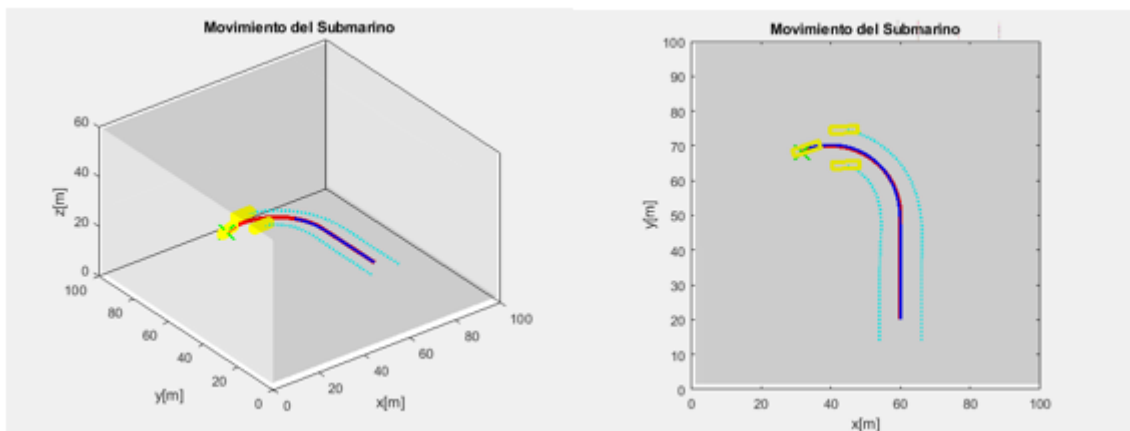


Figura 58: Simulación de curva 2D con robots seguidores.

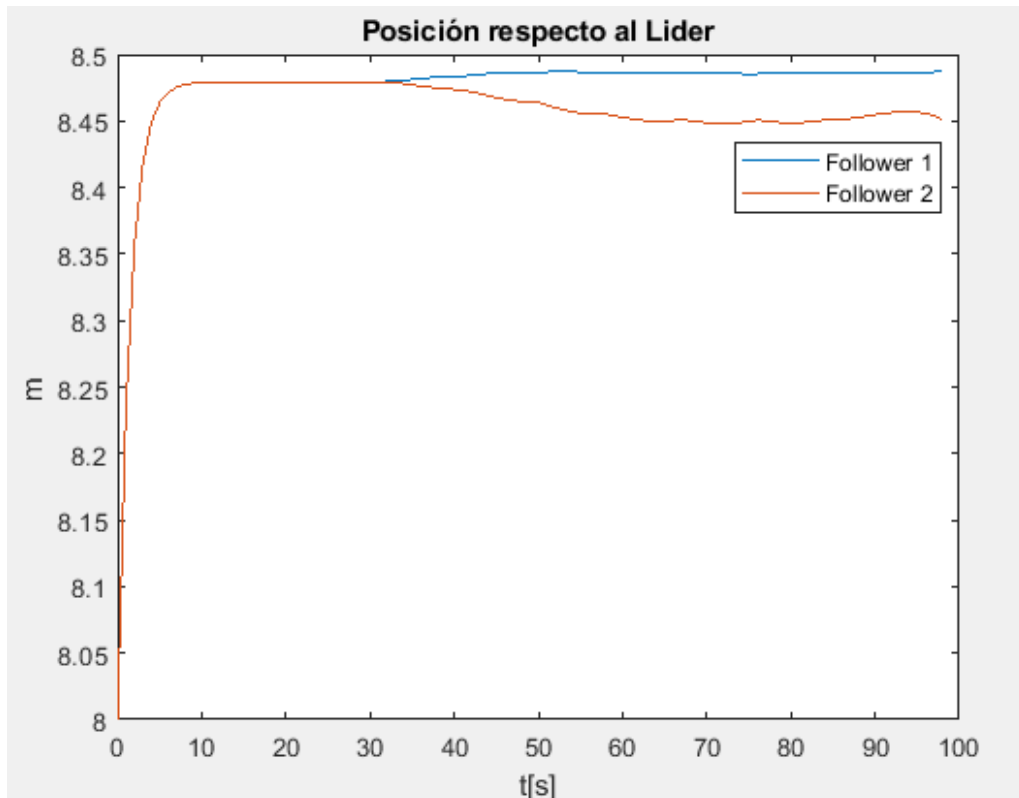


Figura 59: Posición de los robots seguidores con respecto al líder.

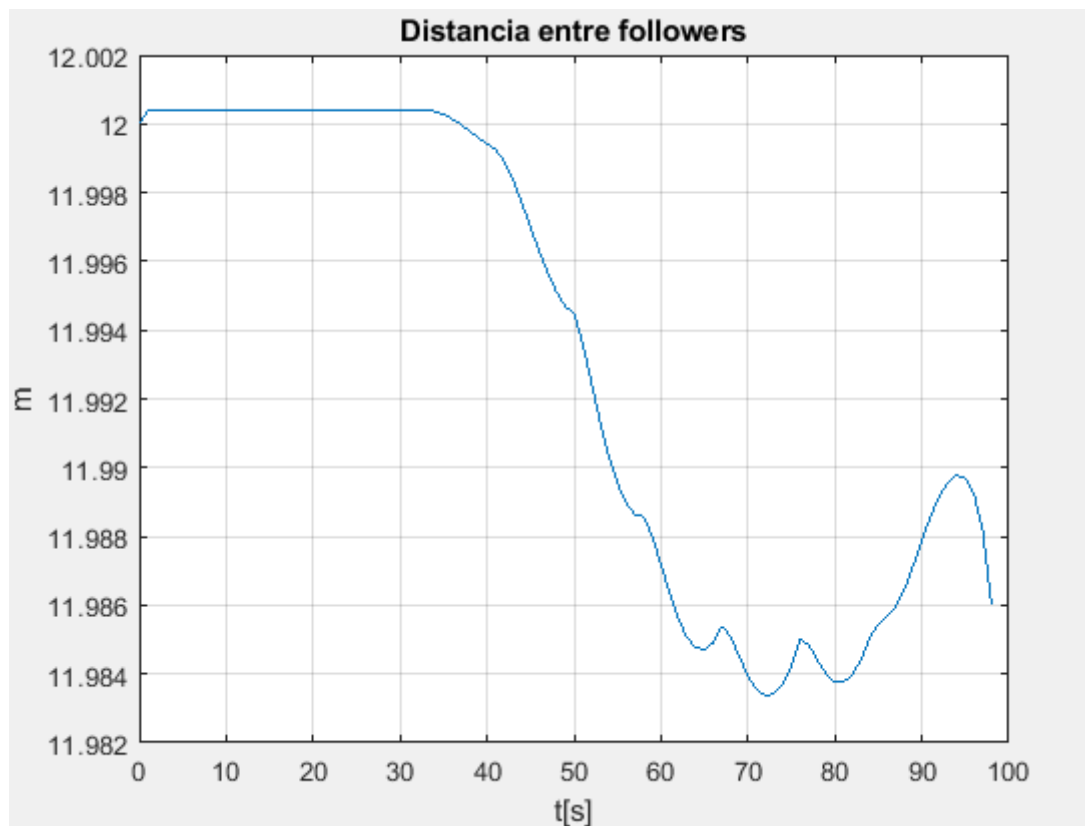


Figura 60: Distancia mantenida entre los robots seguidores en la trayectoria.

Capítulo 4

Como se puede observar en la primera gráfica, hay un error casi inicial de 0.45 metros donde perdura hasta finalizar la simulación. Este error inicial es normal dado que el controlador no es perfecto y el robot líder es el que inicia el movimiento. Se han realizado numerosas simulaciones y este es el mejor resultado que se ha obtenido. Como se puede ver el error inicial es bastante pequeño manteniéndose a lo largo del giro. Cuando esto sucede, el robot situado a la izquierda del líder (representado en azul en la primera gráfica) debe decrecer su velocidad para no acercarse en exceso al líder y el robot situado a la derecha del líder (representado en rojo en la primera gráfica) debe aumentar su velocidad para no quedarse atrás. Este resultado se puede ver a lo largo de la simulación donde el primer robot seguidor queda a una distancia más alejada que el segundo robot seguidor dado que el giro se realiza a favor de este.

Como queda representado en la segunda gráfica, que representa la distancia entre los dos robots seguidores. Esta permanece constante en 12 metros hasta que se realiza el giro, donde se puede apreciar que los robots seguidores se acercan mínimamente habiendo un error casi nulo.

4.3.2. ESPIRAL

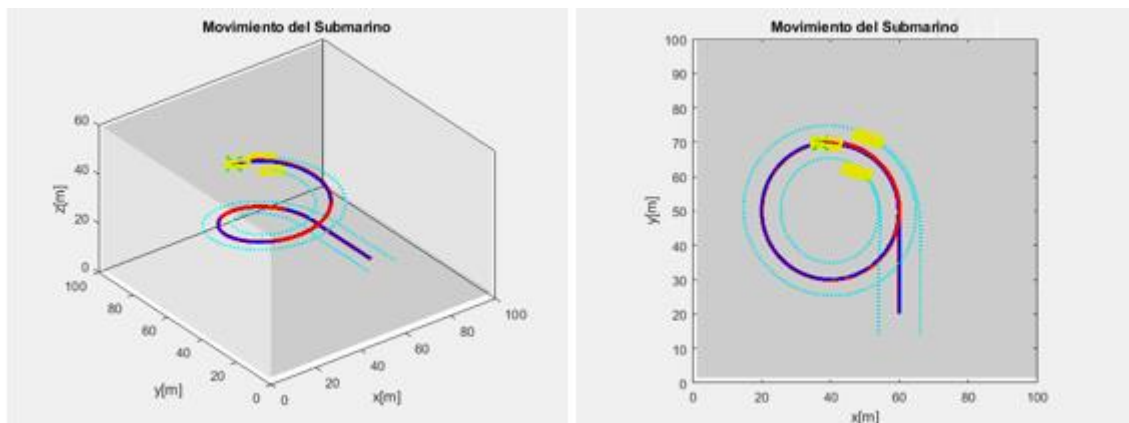


Figura 61: Simulación de espiral con robots seguidores.

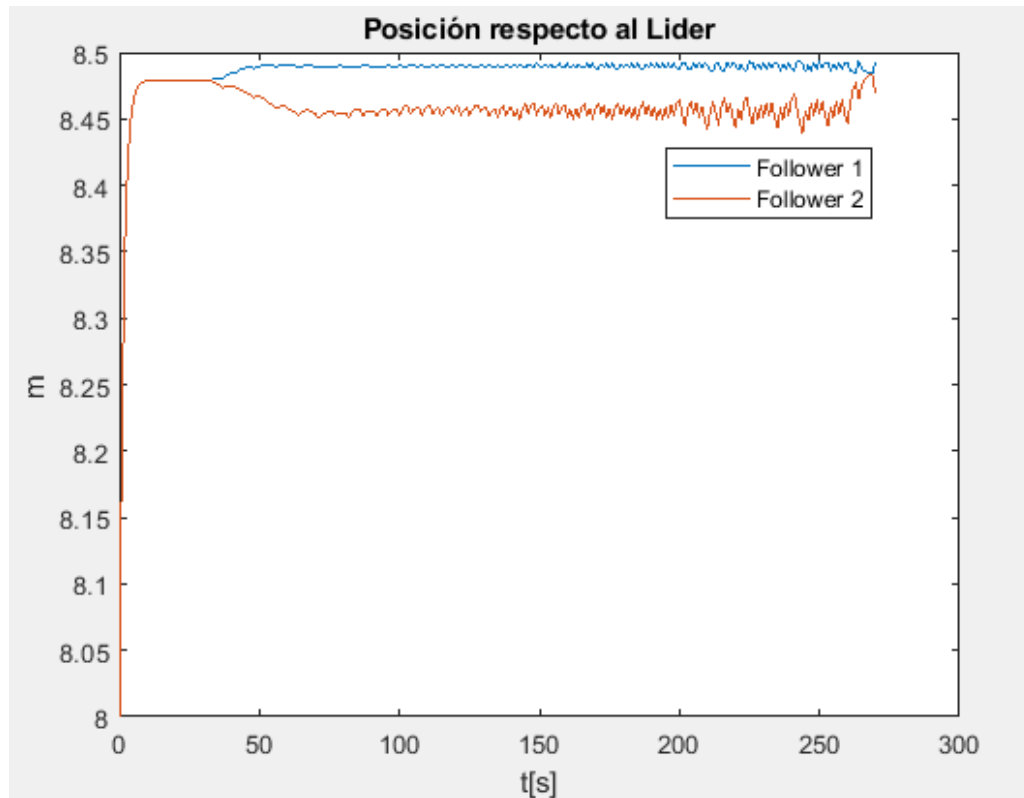


Figura 62: Posición de los robots seguidores con respecto al líder.

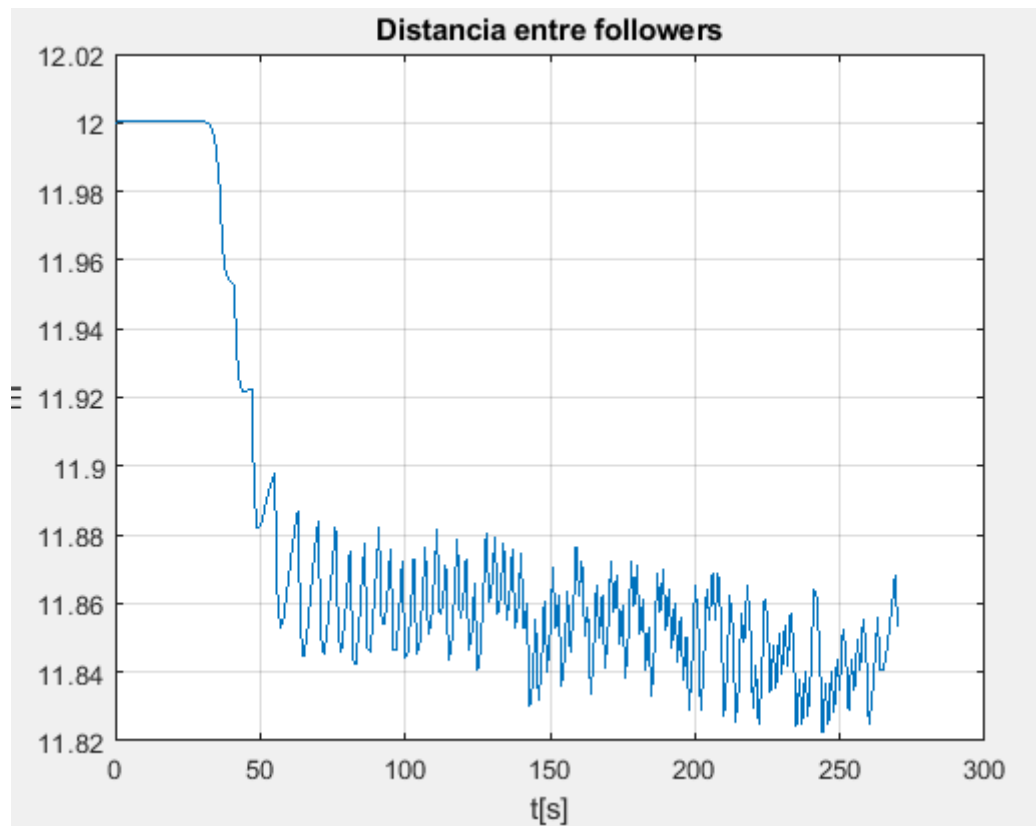


Figura 63: Distancia mantenida entre los robots seguidores en la trayectoria.

Capítulo 4

En esta trayectoria se puede apreciar que los errores de posición aumentan levemente, donde se realiza un giro y medio completo a una altura creciente. De la primera gráfica se puede determinar que el valor de establecimiento es de 8.455 metros para el primer follower y de 8.45 para el segundo. Los errores de establecimiento son muy pequeños por lo que se puede considerar que la distancia que se ejerce es muy buena.

En la segunda gráfica se puede observar que el valor de establecimiento es de aproximadamente 11.86 metros, oscilando los valores con unos 0.02 metros de diferencia. A pesar de los pequeños errores de establecimiento, los resultados de la formación son bastantes notables.

4.3.3. BARRIDO

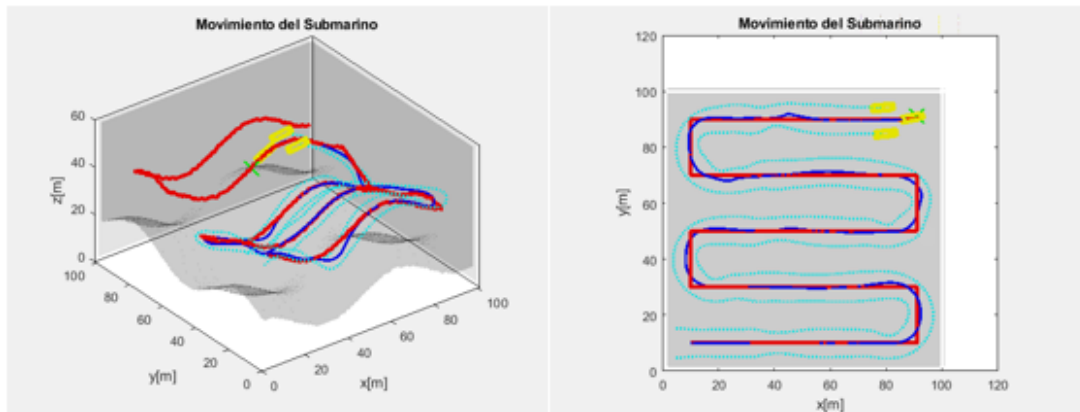


Figura 64: Simulación de barrido con robots seguidores.

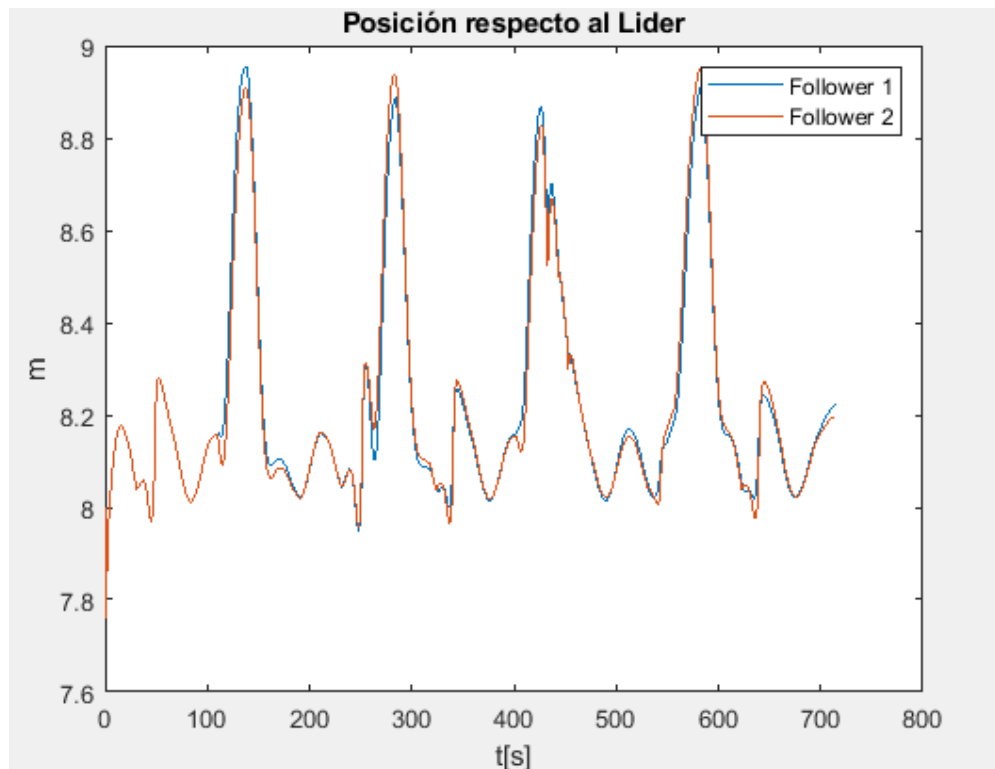


Figura 65: Posición de los robots seguidores con respecto al líder.

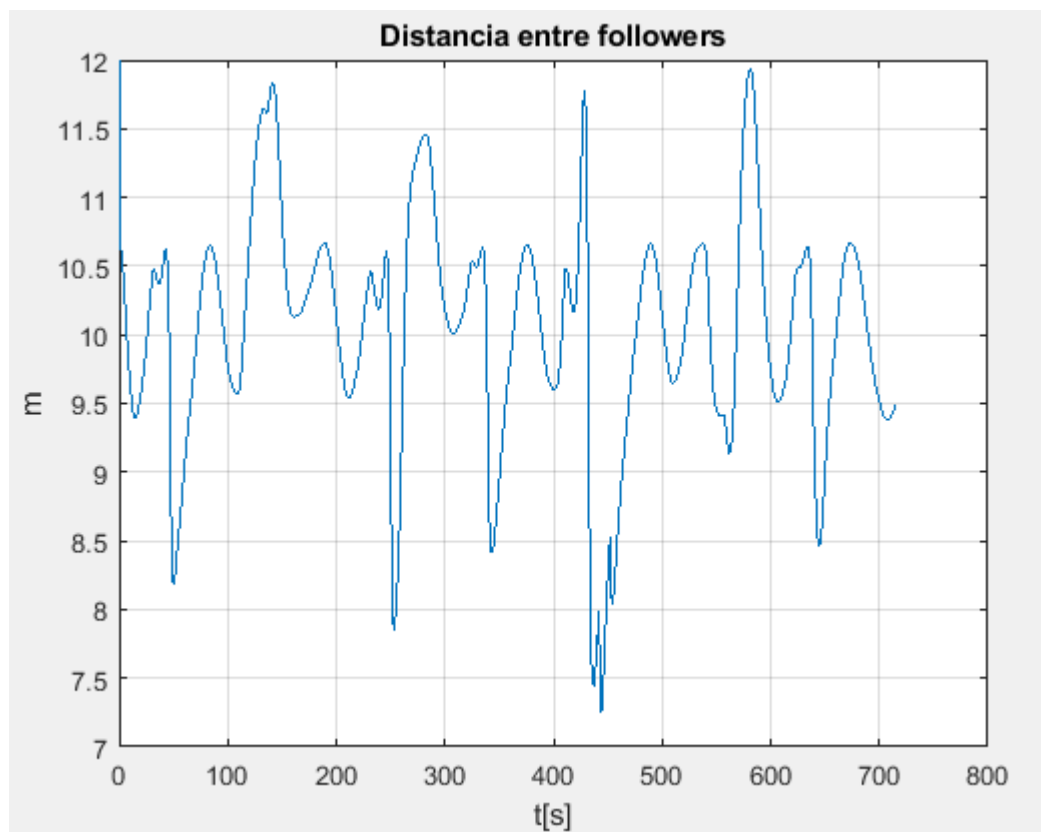


Figura 66: Distancia mantenida entre los robots seguidores en la trayectoria.

Capítulo 4

En esta trayectoria se puede ver claramente que se obtienen los resultados menos estables de las tres simulaciones realizadas ajustando un valor proporcional distinto que a las otras simulaciones. Esto se debe a que esta trayectoria es más compleja y ambiciosa que las otras y donde se realizan giros más agresivos. En la primera gráfica se puede ver cómo hay 4 picos con un error máximo de alrededor 0.9 metros. También es interesante observar cómo, dependiendo de la orientación a favor del giro, los robots seguidores tienen menor error que el otro. A pesar de esto, comparando los resultados con las otras simulaciones se puede apreciar que el valor de establecimiento es mucho mejor

Por otro lado, en la segunda gráfica se puede apreciar que el valor de establecimiento decrece a 10 metros, dejando un error de establecimiento de 2 metros. El error máximo se produce al final del tercer giro en la simulación, cuando se realiza un giro ascendente, dejando una distancia entre los submarinos de unos 7.5 metros.

Capítulo 5. CONCLUSIONES

5.1. CONCLUSIONES

Como conclusión de los resultados obtenidos se puede corroborar que el controlador PP es un algoritmo de seguimiento de trayectoria proporcional que, ajustando dicho parámetro, puede afectar satisfactoriamente en los resultados que se obtienen. Se ha comprobado y verificado que la distancia LA es otro parámetro muy influyente para que el algoritmo funcione correctamente. Si se aproxima mucho dicho valor, hará que el recorrido del AUV sea inestable generando oscilaciones bruscas en su orientación aumentando el peligro de que el robot pierda el control. Por contrapartida, si se aleja en exceso el punto de mira, hará que el robot realice giros muy abiertos aumentando el tiempo en el que el robot consigue corregir su orientación incrementando proporcionalmente el error de posición de este. Solo ajustando dichos valores se puede conseguir un buen resultado de seguimiento de trayectoria.

Analizando los resultados obtenidos en el control de formación, se puede concluir que el sistema de control empleado es capaz de mantener la formación de manera satisfactoria. Cabe mencionar que el error de la distancia que mantienen los robots seguidores entre ellos es mayor que los errores obtenidos entre la distancia que mantiene el robot líder con los seguidores. Estos errores se acentuaban sobre todo cuando se estaban realizando giros, donde los robots seguidores tienden a unirse. Observando los resultados de la trayectoria, se puede ver como cuanto más pronunciada sea el giro, mayor tendencia tienen los seguidores a juntarse. Parte de estos errores también se ven influenciados por el controlador PP. Es decir, como el seguimiento de trayectoria del líder no es perfecto, este acumula errores que se acaban

Capítulo 5

viendo reflejados también en la trayectoria y control de formación de los robots seguidores.

5.2. RESUMEN DE OBJETIVOS

Finalmente, al término de este proyecto, se presentan el resumen de objetivos de este proyecto.

- Uno de los requerimientos del proyecto era establecer una cinemática fiel para un robot submarino. Finalmente, tras investigar entre numerosos trabajos de investigación, se ha conseguido programar una cinemática para un AUV de manera satisfactoria.
- Se ha conseguido diseñar un mapa submarino compuesto por voxels donde se ha desarrollado la simulación. También se ha diseñado una figura de un AUV donde poder aplicar la cinemática en vez de a un punto en el espacio.
- Para el seguimiento de trayectoria, se ha investigado sobre los diversos algoritmos posibles y se ha seleccionado y programado con éxito el algoritmo de persecución pura. Además, se ha modificado este para que pueda operar en un espacio de tres dimensiones.
- Se ha investigado acerca las distintas formaciones posibles en entornos submarinos y se ha seleccionado y programado con éxito la más adecuada para el barrido de un suelo submarino.
- Se ha elegido y programado con éxito la estrategia de control de formaciones Líder-Seguidor.

5.3. PERSPECTIVAS FUTURAS

Una vez expuestos los resultados y sacado las conclusiones, se procede a exponer las líneas futuras del proyecto:

- Diseñar y simular sensores submarinos de detección de objetos tales como “ping-sonars” y sonares de barrido lateral para poder detectar y esquivar obstáculos en el camino y realizar un mapeo del suelo submarino adaptando el mapa a uno con obstáculos.

Capítulo 5

- Implementar una estrategia de barrido *en línea* siendo el mapa parcialmente conocido o no conocido para los AUV.
- Realizar un algoritmo que resuelva el problema del viajero para poder barrer zonas con obstáculos con mayor eficiencia.
- Programar un algoritmo de control que permita esquivar obstáculos sin perder la formación mientras se sigue la trayectoria.
- Desarrollar un algoritmo que sea capaz de generar una trayectoria desde la posición de los AUV hasta un punto de interés en el espacio libre evitando obstáculos.

BIBLIOGRAFÍA

- [1] Behnaz Hadi, Alireza Khosravi, Pouria Sarhadi, "A Review of the Path Planning and Formation Control for Multiple Autonomous Underwater Vehicles," *Journal of Intelligent & Robotic Systems*, 2021.
- [2] David Ribas, Narcís Palomeras, Pere Ridao, Marc Carreras, Angelos Mallios, "Girona 500 AUV: From Survey to Intervention," *IEEE/ASME Transactions on mechatronics*, vol. 17, NO. 1, 2012.
- [3] M. Prats, D. Ribas, N. Palomeras, J. C. García, V. Nannen, J. J. Fernandez, J. P. Beltrán, R. Campos, P. Ridao, P. J. Sanz, G. Oliver, M. Carreras, N. Gracias, R. Marín, and A. Ortiz, "Reconfigurable AUV for intervention missions: A case study on underwater object recovery," *J. Intell. Serv. Robot.*, to be published. [Online]. Available: <http://dx.doi.org/10.1007/s11370-011-0101-z>.
- [4] Marc Carreras, Juan David Hernández, Eduardo Vidal, Narcís Palomeras, David Ribas, Pere Ridao, "Sparus II AUV – A Hovering Vehicle for Seabed Inspection," *IEEE Journal of oceanic engineering*, vol. 43, NO. 2, 2018.
- [5] Marc Carreras, "Design of SPARUS II AUV," *Underwater Robotics Research Centre (CIRS), Computer Vision and Robotics Institute Universitat de Girona, 17003, Girona, Spain*, 2014.

Bibliografía

- [6] Mohammad Rokonuzzaman, Navid Mohajer, Saeid Nahavandi, Shandy Mohamed, "Review and performance evaluation of path tracking controllers autonomous vehicles," *IET Intelligent Transport Systems*, 2021.
- [7] Martin Lundgren, "Path Tracking for a Miniature Robot," *Department of Computing Science, Umeå University, Sweden*, 2003.
- [8] "Geometric lateral Control – Pure Pursuit," [Online]. Available: <https://www.coursera.org/lecture/intro-self-driving-cars/lesson-2-geometric-lateral-control-pure-pursuit-44N7x>.
- [9] R. Craig Coulter, "Implementation of the Pure Pursuit Path Tracking Algorithm," *The Robotic Institute Carnegie Mellon University, Pittsburgh, Pennsylvania*, 1992.
- [10] E. Minguzzi, "A geometrical introduction to screw theory," *Dipartimento di Matematica Applicata, Università degli Studi di Firenze*, 2012.
- [11] Enric Galceran, Marc Carreras, "A Survey on Coverage Path Planning for Robotics," *University of Girona Underwater Robotics Research Center (CIRS)*, 2013.
- [12] Enric Galceran, "Coverage path planning for autonomous underwater vehicles," *Universitat de Girona*, 2014.
- [13] Yuanchang Liu, Richard Bucknall, "A survey of formation control and motion planning of multiple unmanned vehicles," *Department of Mechanical Engineering, University College London, University College London, Torrington place, London*, 2018.
- [14] Behnaz Hadi, Alireza Khosravi, Pouria Sarhadi, "A Review of the Path Planning and Formation Control for Multiple Autonomous Underwater Vehicles," *Journal of Intelligent & Robotic Systems*, 2021.
- [15] T. Prestero, "Verification of a six-degree of freedom simulation model for the remus autonomous underwater vehicle," *Master's thesis, Massachusetts Institute of Technology; Woods Hole Oceanographic Institution, Cambridge, MA, USA; Woods Hole, MA, USA*, September 2001.
- [16] Outaiba GHERAIRI, Mouna BEN HAMED, "Kinematic model of underwater vehicle," *University of Gabès, Tunisia*, 2017.
- [17] Hector Alonso Moreno Ávalos, "Modelado, Control y Diseño de Robots Submarinos de Estructura Paralela con Impulsores Vectorizados," *Departamento de Automática, Ingeniería Electrónica e Informática Industrial, Escuela Técnica Superior de Ingenieros Industriales*, 2013.

Bibliografía

- [18] "Controlador de Pure Pursuit," *MathWorks*, [Online]. Available: <https://es.mathworks.com/help/robotics/ug/pure-pursuit-controller.html>.
- [19] Yi Yao, Ye Sun, "AGV vehicle differential drive model," *Journal of Physics*, 2021.
- [20] Ben Allen, Roger Stokey, Tom Austin, Ned Forrester, Rob Goldsborough, Mike Purcell, Chris von Alt, "REMUS: A small, low cost AUV; System Description, Field Trials and performance Results", *Oceanographic Systems Laboratory*, 1997.
- [21] Ben Allen, William S. Vorus, Timothy Prestero, "Propulsion System Enhancements on REMUS AUVs", *IEEE*, 2000.

Anexo A. REPOSITORIO DE GITHUB

A.1. CÓDIGO DEL MODELO DE SUBMARINO Y LAS SIMULACIONES

El código realizado para el desarrollo del trabajo fin de grado se encuentra en el siguiente repositorio de Github:

[Planificación y seguimiento de trayectorias para formación de robots submarinos.](#)

Dicho repositorio está formado por una carpeta donde se encuentra todo el código en Matlab y un fichero README que conviene leer para saber cómo manejar la simulación. Además de esta breve descripción que se va a aportar acerca de los ficheros que conforman el código, dentro de este, cada función viene detallada y comentada explicando también los valores de entrada y salida que tienen para hacer más legible el código.

En la carpeta se encuentra el código con los siguientes ficheros:

- Barrer_1: Esta función se encarga de generar el barrido de la zona desde la posición del submarino en el primer cuadrante.

Anexo A

- C-Altura: Esta función se encarga de detectar el relieve del mapa en la posición de la coordenada x e y devolviendo una altura de 20 unidades por encima del relieve.
- CinematicalnvSub: Esta función se encarga de calcular la cinemática inversa del robot. Calcular el error de posición y orientación con la diferencia de la posición/orientación actual con la deseada. Aplicando la ley de control se calcula la velocidad lineal y angular necesaria para alcanzar dicho estado en un tiempo finito. Se calcula la cinemática para recalculer la posición del robot con las velocidades.
- CinematicaSub: Esta función se encarga de calcular la cinemática del robot.
- CorrigeAngulo: Esta función esta empleada para calcular el error de orientación en torno al punto objetivo.
- CurvaXY2D: Esta función se encarga de generar una curva 2D como trayectoria del robot.
- Errpos: Esta función se encarga de calcular el error de posición del AUV con respecto a la trayectoria.
- Espiral: Esta función se encarga de generar una espiral ascendente como trayectoria del robot.
- Fmapa: Función que se encarga de generar el mapa submarino. Se genera un plano tridimensional que será el relieve de la superficie submarina.
- Formación: Esta función se encarga de calcular la formación de los robots. La formación elegida es una formación triangular.
- GestorBarrido: Esta función se encarga de generar el barrido del robot. Pensada también para introducir nuevos barridos de manera más fácil.
- GestorFunciones: Esta función se encarga de seleccionar la trayectoria deseada llamando a la función que se genera dicha trayectoria. La posición inicial del robot líder se define en esta función.
- GetAltura: Esta función se encarga de calcular la altura del punto objetivo actual en la trayectoria para que pueda ser seguido por el robot.
- Main: Fichero main del código, en él se encuentra el bucle de simulación y muestra las gráficas de resultados.

Anexo A

- Plot_map3d: Esta función se encarga de realizar el plot del mapa mostrando el entorno de simulación.
- PosLF: Esta función se encarga de calcular la distancia que existe entre el líder y los followers. Se emplea para obtener datos de cómo se mantiene la formación.
- PuntoCuadrante: Esta función se encarga de ajustar el punto inicial adecuado si no está ajustado.
- PurePursuit: Esta función se encarga de realizar el sistema de control purepursuit devolviendo las velocidades angulares necesarias para realizar la curva y la siguiente posición del punto objetivo.
- Recta: Esta función se encarga de generar una línea recta como trayectoria del robot.
- Rectángulo: Esta función se encarga de dibujar el submarino en función de su posición y orientación.