



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN ELECTRÓNICA E INSTRUMENTACIÓN**

**TEMA: SISTEMA MULTICAPA DE PLANIFICACIÓN EN 3D MEDIANTE
LA APLICACIÓN DE ALGORITMOS GENÉTICOS, PARA OPTIMIZAR
TRAYECTORIAS DE ROBOTS AÉREOS NO TRIPULADOS EN
AMBIENTES PARCIALMENTE ESTRUCTURADOS**

AUTORES:

MOLINA PALACIOS JOSÉ IGNACIO.

NARANJO ERAZO MAURICIO XAVIER.

DIRECTOR: ING. JULIO ACOSTA N. M.Sc.

LATACUNGA

ENERO 2019



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA E INSTRUMENTACIÓN

CERTIFICACIÓN

Certifico que el trabajo de titulación, “**SISTEMA MULTICAPA DE PLANIFICACIÓN EN 3D MEDIANTE LA APLICACIÓN DE ALGORITMOS GENÉTICOS, PARA OPTIMIZAR TRAYECTORIAS DE ROBOTS AÉREOS NO TRIPULADOS EN AMBIENTES PARCIALMENTE ESTRUCTURADOS**” realizado por los señores **MOLINA PALACIOS JOSÉ IGNACIO, NARANJO ERAZO MAURICIO XAVIER** ha sido revisado en su totalidad y analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Latacunga, 23 de enero de 2019



JULIO FRANCISCO ACOSTA NÚÑEZ
C.C: 050151949-0
DIRECTOR



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA E INSTRUMENTACIÓN

AUTORÍA DE RESPONSABILIDAD

Nosotros, **MOLINA PALACIOS JOSÉ IGNACIO, NARANJO ERAZO MAURICIO XAVIER** declaramos que el contenido, ideas y criterios del trabajo de titulación : ***Sistema multicapa de planificación en 3D Mediante la aplicación de algoritmos genéticos, para optimizar trayectorias de robots aéreos no tripulados en ambientes parcialmente estructurados*** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas - ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz

Latacunga, 23 de enero de 2019



MOLINA PALACIOS JOSÉ IGNACIO

C.C: 050345601-4



NARANJO ERAZO MAURICIO XAVIER

C.C:050365453-5



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA E INSTRUMENTACIÓN

AUTORIZACIÓN

Nosotros, **MOLINA PALACIOS JOSÉ IGNACIO, NARANJO ERAZO MAURICIO XAVIER**, autorizamos a la Universidad de las Fuerzas Armadas – ESPE publicar el trabajo de titulación: ***ANÁLISIS SISTEMA MULTICAPA DE PLANIFICACIÓN EN 3D MEDIANTE LA APLICACIÓN DE ALGORITMOS GENÉTICOS, PARA OPTIMIZAR TRAYECTORIAS DE ROBOTS AÉREOS NO TRIPULADOS EN AMBIENTES PARCIALMENTE ESTRUCTURADOS*** en el Repositorio Digital Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Latacunga, 23 de enero de 2019

MOLINA PALACIOS JOSÉ IGNACIO

C.C: 050345601-4

NARANJO ERAZO MAURICIO XAVIER

C.C:050365453-5

DEDICATORIA

*Dedico principalmente este trabajo a mi madre **LARAINÉ PALACIOS** quien me supo apoyar durante todas y cada una de mis decisiones sean buenas o malas, hasta este punto en el cual culmino mis estudios de ingeniería. A mi papá, mi hermana, mi hermano, mis sobrinos y al resto de mi familia quienes son el engranaje de mi vida y con quienes comparto día a día momentos especiales.*

JOSÉ MOLINA

Este presente trabajo se lo dedico a Dios por haber puesto a las personas adecuadas en mi camino tanto a familiares como amigos para alcanzar este logro así como también por brindarme la sabiduría día a día para lograrlo, a mis dos madres Hilda Laverde y Leonor Erazo que nunca dejaron de creer en mi pese a los problemas que se presentaron en todo este trayecto, a mi padre German Erazo por su apoyo incondicional en mi vida guiándome para seguir sus pasos tanto de vida como profesionales, a mis hermanos Edison Naranjo y Estefanía Naranjo quienes pese a sus carreras me alientan y comparten su tiempo en donde sea que se encuentren les quiero mucho.

A mis tíos: Mario, Narcisa E., Julio, Patricia, Xavier, Tere, Narcisa R., Pepe, Pauli, Alexandra y Janeth quienes siempre me impulsaron a seguir adelante en los momentos difíciles para cumplir esta meta y al resto de mi gran familia que son promotores también de esta etapa de mi vida.

A Rosita Holguín, Liliana Delgado, Blanca Delgado, Efrén Delgado, Maritza Delgado, Laura Macías, Eduardo Luces y toda su familia entera por haberme brindado su hogar en aquella época difícil que pasé.

MAURICIO NARANJO

AGRADECIMIENTO

El presente trabajo agradece de manera especial a cada integrante de sus familias y seres amados respectivamente A la Universidad de las Fuerzas Armadas - ESPE sede Latacunga por las oportunidades brindadas y conocimientos adquiridos, de igual manera a su mentor y director de proyecto al Ing. Julio Acosta M.Sc., además de ser quien nos brindó la oportunidad de conocer otro país, otra universidad y otra cultura.

José Molina P.

Mauricio Naranjo E.

ÍNDICE DE CONTENIDO

CARATULA

CERTIFICACIÓN	i
AUTORÍA DE RESPONSABILIDAD.....	ii
AUTORIZACIÓN.....	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
ÍNDICE DE CONTENIDO	vi
ÍNDICE DE FIGURAS.....	viii
RESUMEN.....	x
ABSTRACT.....	xi

CAPITULO 1

1 PROBLEMA	1
1.1 Tema de investigación.....	1
1.2 Antecedentes.....	1
1.3 Planteamiento del Problema	2
1.4 Justificación	3
1.5 Objetivos	4
1.5.1 Objetivo general.....	4
1.5.2 Objetivos específicos	4
1.6 Variables de la Investigación.....	4

CAPITULO 2

2 MARCO TEÓRICO	5
2.1 Robots aéreos no tripulados.....	5
2.1.1 Clasificación de los UAV.....	5
2.1.2 Aplicación de los UAV's	13
2.2 Path Planning RRT	15
2.2.1 Path Planning RRT 2D.....	17

2.2.2 Path Planning RRT 3D.....	25
2.3 Algoritmos Genéticos	27
2.3.1 Definición de Inteligencia Artificial.....	27
2.3.2 Técnicas mediante Algoritmo Genéticos	30
CAPITULO 3	
3 DESARROLLO DEL ALGORITMO	36
3.1 Sistema de planificación.....	36
3.1.1 Capa de Mapeo	37
3.1.2 Capa RRT 3D.....	39
3.1.3 Capa de Optimización y Control	45
A. Algoritmos Genéticos	46
B. Operaciones de los algoritmos genéticos	54
C. Optimización por distancias euclidianas	59
3.1.4 Controlador de UAV's.....	62
CAPITULO 4	
4 RESULTADOS EXPERIMENTALES	67
4.1 Prueba del algoritmo RRT 3D y generación de trayectoria de probabilidad.....	67
4.2 Técnicas de algoritmos genéticos y Optimización.....	72
4.3 Pruebas de trayectorias en controlador de caminos.....	82
4.3.1 Evaluación de trayectorias en el controlador	85
CAPITULO 5	
5 CONCLUSIONES Y RECOMENDACIONES.....	98
5.1 Conclusiones	98
5.2 Recomendaciones	99
REFERENCIAS BIBLIOGRÁFICAS	101

ÍNDICE DE FIGURAS

<i>Figura 2.1:</i> Drone de rotor único.....	6
<i>Figura 2.2:</i> Drone tricóptero en forma de "Y"	7
<i>Figura 2.3:</i> Drone Quadróptero tipo "X".....	7
<i>Figura 2.4:</i> Drone Hexacóptero.....	8
<i>Figura 2.5:</i> Drone Octocóptero	9
<i>Figura 2.6:</i> Drone tipo Ala Fija	9
<i>Figura 2.7:</i> Micro Drone	10
<i>Figura 2.8:</i> Mini Drone Tello	11
<i>Figura 2.9:</i> Drone de Tamaño Medio Phantom	11
<i>Figura 2.10:</i> Drone de Gran Tamaño MQ-1 Predator de General Atomics	12
<i>Figura 2.11:</i> Algoritmo RRT básico.....	18
<i>Figura 2.12:</i> Algoritmo RRT función Extiende.....	19
<i>Figura 2.13:</i> Esquema de expansión del RRT	20
<i>Figura 2.14:</i> Evolución del Algoritmo RRT en un entorno circular.	22
<i>Figura 2.15:</i> Evolución del Algoritmo RRT en un entorno asimétrico	24
<i>Figura 2.16:</i> Algoritmo RRT en 3D	26
<i>Figura 2.17:</i> Darwin, Charles Robert (1809 - 1882).....	28
<i>Figura 3.1:</i> Descripción del sistema planificación.....	36
<i>Figura 3.2:</i> Entorno 2D en imagen vista superior.....	37
<i>Figura 3.3:</i> Entorno 2D en escala de grises invertida.	38
<i>Figura 3.4:</i> Valor en pixeles del relieve de materia a partir de RGB.	38
<i>Figura 3.5:</i> Entorno 3D reconstruido a partir de imagen en 2D.....	39
<i>Figura 3.6:</i> Nodos generados y punto de llegada.....	40
<i>Figura 3.7:</i> Generación del nuevo nodo a partir del nuevo punto padre.	41
<i>Figura 3.8:</i> Trayectoria de menor distancia por la función costo.....	42
<i>Figura 3.9:</i> Proyecciones de las partículas en el entorno.	42
<i>Figura 3.10:</i> Trayectorias determinadas por la función costo.	43
<i>Figura 3.11:</i> Generación trayectoria RRT 3D.	44
<i>Figura 3.12:</i> Diagrama de un algoritmo genético simple.....	47
<i>Figura 3.13:</i> Muestras y probabilidades de la población.	51
<i>Figura 3.14:</i> Muestras y probabilidades de evolución.	52
<i>Figura 3.15:</i> Muestras y probabilidades de mutación.	54
<i>Figura 3.16:</i> Interpretación del vector probabilidad para decisión del algoritmo genético.	55
<i>Figura 3.17:</i> Interpretación del vector población para decisión del algoritmo genético.....	57
<i>Figura 3.18:</i> Interpretación del vector probabilidad para decisión del algoritmo genético.	58
<i>Figura 3.19:</i> Trayectorias del algoritmo genético.....	59
<i>Figura 3.20:</i> Distancias euclidianas en las trayectorias	61
<i>Figura 3.21:</i> Selección de la trayectoria (<i>path</i>) óptima para la navegación	62

Figura 3.22: Representación en el plano de un UAV	63
Figura 3.23: Diagrama de bloques del control	64
Figura 4.1: Mapeado de materia (Obstáculos).....	69
Figura 4.2: Trayectoria reproducción de probabilidad mediante algoritmo RRT 3D	71
Figura 4.3: Algoritmo Genético por mutación y evolución de probabilidad.	74
Figura 4.4: Trayectorias probables generadas en el entorno 1.....	75
Figura 4.6: Trayectorias probables generadas en el entorno 2.....	77
Figura 4.7: Trayectoria óptima para navegación entorno 2	78
Figura 4.8: Trayectorias probables generadas en el entorno 3.....	79
Figura 4.9: Trayectoria óptima para navegación entorno 3.	80
Figura 4.10: Trayectorias probables generadas en el entorno 4.....	81
Figura 4.11: Trayectoria óptima para navegación entorno 4	82
Figura 4.12: Suavizado de Trayectorias generadas.	84
Figura 4.13: Resultados evaluación de Trayectoria por mutación entorno 1	86
Figura 4.14: Resultados evaluación de Trayectoria por evolución entorno 1	87
Figura 4.15: Resultados evaluación de Trayectoria por reproducción natural entorno 1	88
Figura 4.16: Resultados evaluación de Trayectoria por mutación entorno 2	89
Figura 4.17: Resultados evaluación de Trayectoria por evolución entorno 2	90
Figura 4.18: Resultados evaluación de Trayectoria por reproducción natural entorno 2	91
Figura 4.19: Resultados evaluación de Trayectoria por mutación entorno 3	92
Figura 4.20: Resultados evaluación de Trayectoria por evolución entorno 3	93
Figura 4.21: Resultados evaluación de Trayectoria por reproducción natural entorno 3	94
Figura 4.22: Resultados evaluación de Trayectoria por mutación entorno 4	95
Figura 4.23: Resultados evaluación de Trayectoria por evolución entorno 4	96
Figura 4.24: Resultados evaluación de Trayectoria por reproducción natural entorno 4	97

RESUMEN

La nueva era tecnológica en la que los robots aéreos no tripulados tienen múltiples aplicaciones y han evolucionado sus habilidades, así como sus capacidades hace posible que en la actualidad posean diferentes funcionalidades una de las más importantes es su autonomía de navegación, lográndose esto mediante inteligencia artificial. En el presente proyecto se implementa un sistema capaz de planificar trayectorias como las condiciones del entorno parcialmente estructurado, para esto se desarrollará un sistema multicapas la cual consta de *i) capa de mapeo* mediante nuevas técnicas de reconstrucción de entornos en 3D, se usará un lenguaje de programación matricial mediante gráficas de 2D en formato RGB de esta manera la información dentro de las matrices reconstruirán las tres dimensiones del entorno semiestructurado; *ii) capa RRT 3D* está orientada al desarrollo de una arquitectura de planificación en 3D, la cual evita obstáculos a través de algoritmos genéticos; *iii) capa de optimización y control* aquí se puede optimizar varios criterios de la navegación del UAV como son distancia, tiempo, consumo de energía, etc. También se muestra el control de las variables más importantes para darle autonomía al sistema además de una comunicación TCP-IP a la capa de comunicación la cual es transparente al usuario y sirve para medir la posición actual del UAV y por lo tanto tomar las acciones de control pertinentes para que los robots lleguen a destino.

PALABRAS CLAVE

- **VEHÍCULO AÉREO NO TRIPULADO (UAV)**
- **PLANIFICACIÓN DE CAMINOS**
- **ALGORITMOS GENÉTICOS**

ABSTRACT

The new technological era in which unmanned aerial robots have multiple applications and have evolved their abilities as well as their capabilities make it possible that at present they have different functions for the most important ones in their navigational autonomy, achieving this through artificial intelligence. In the present project, a system capable of planning trajectories is implemented, as well as the conditions of the partially structured environment, for this a multilayer system will be developed, which consists of *i) mapping layer* through new reconstruction techniques in 3D environments, a language will be used of matrix programming through 2D graphics in RGB format in this way the information within the matrices will reconstruct the three dimensions of the semi-structured environment; *ii) RRT 3D layer* is oriented to the development of a 3D planning architecture, which avoids obstacles through genetic algorithms; *iii) Optimization and control layer* here you can optimize several UAV navigation criteria such as distance, time, energy consumption, etc. It also shows the control of the most important variables to give autonomy to the system as well as a TCP-IP communication to the communication layer which is transparent to the user and serves to measure the current position of the UAV and therefore take the actions of control relevant to the robots reach their destination.

KEYWORDS

- **UNMANNED AERIAL VEHICLE (UAV)**
- **PATH PLANNING**
- **GENETIC ALGORITHMS**

CAPITULO 1

1 PROBLEMA

1.1 Tema de investigación

“Sistema Multicapa de Planificación en 3D mediante la aplicación de algoritmos genéticos, para optimizar trayectorias de robots aéreos no tripulados en ambientes parcialmente estructurados”

1.2 Antecedentes

La investigación científica a nivel mundial tiende a realizarse a través de grupos de investigación, siendo uno de los campos más activos la robótica, debido a esto se han realizado varias investigaciones en el área de ubicación, mapeo y planificación para robots autónomos que se dirigen a entornos de 3D, adicionalmente casi todas las investigaciones toman dirección a algoritmos que facilitan la locomoción para un robot entre estos: el mapeo, la exploración y la ubicación en entornos cada vez más cercanos a la realidad, habiendo pasado por entornos de simulación. Como en distintos casos el enfoque de un robot con herramientas sencillas para localización, mapeo y exploración de ambientes en 3D con robots aéreos no tripulados, todo esto se investiga en forma que puede describirse con características basadas en el uso de puntos de referencia (Dedeoglu & Sukhatme, 2000) y/o grafos (Rekleitis, Dudek, & Milios, 2000), los cuales usan sensores de rango (láser) y sensores de visión (Rao, 2000).

A esto nos queda mucho por ser determinado por la variedad de enfoques disponibles para ubicación, mapeo y planificación. Los escenarios de aplicación de la robótica han evolucionado en los últimos años, desde ambientes relativamente simples y controlados a ambientes dinámicos en exteriores. Al mismo tiempo para afrontar ciertas aplicaciones, la optimización en la planificación se ha convertido en una necesidad. Una tendencia en la actualidad es la investigación en sistemas que consideren la planificación de robots aéreos no tripulados y algoritmos de inteligencia artificial como los genéticos que permitan optimizar recursos presentes en el entorno para multitud de aplicaciones, como robótica de servicio en entornos urbanos o monitorización de desastres naturales y vigilancia de procesos industriales.

1.3 Planteamiento del Problema

En la última década la creciente demanda de avances tecnológicos alrededor de la robótica autónoma a nivel mundial, genera avances en algoritmos capaces de determinar trayectorias que consideren las restricciones de robots, como también la dinámica del entorno de trabajo.

Debido a la complejidad para que los robots aéreos no tripulados eviten obstáculos en entornos inexplorados y además de problemas en procesamiento computacional, algoritmos de planificación de trayectorias que usan funciones en las que se debe priorizar el mínimo tiempo en que se cumple el algoritmo, además del uso mínimo de energía que gasta el robot, para desplazarse, así como deficientes sistemas de navegación que brinden estructuras adecuadas para cumplir tareas específicas. Por lo

mencionado anteriormente, la planificación de una ruta automática en entornos de 3D es un problema crítico en la creación de robots autónomos, pero no es el único problema. Otros problemas importantes incluyen el control de movimiento en tiempo real, detección y evasión de obstáculos, siendo el problema más importante la optimización de dichas trayectorias planificadas en entornos de 3D.

1.4 Justificación

Este proyecto de investigación se realiza para su presentación como Proyecto de Fin de carrera de Ingeniería Electrónica e Instrumentación en la Universidad de las Fuerzas Armadas ESPE extensión Latacunga.

Debido a la problemática anteriormente expuesta, La planificación de caminos se direcciona a soluciones geométricas sin tomar en cuenta a las limitaciones impuestas por la no holonomía de los sistemas. Lozano-Pérez trabajó en la búsqueda de caminos en términos geométricos e introdujo conceptos de espacio de configuraciones que define el espacio físico de trabajo del sistema y dentro del cual está localizado por su posición y orientación (LOZANO-PEREZ, 1990).

Así como implementar algoritmos y sistemas de planificación que superen hitos de entornos urbanos, los cuales aumentan su dificultad por encontrarse en constante interacción con el hombre, haciendo que los sistemas ligados a este tipo de robots con autonomía se conviertan en máquinas capaces de servir de manera eficiente y óptima para mejorar la calidad de vida junto con el desarrollo de tecnología basada en inteligencia artificial.

1.5 Objetivos

1.5.1 Objetivo general

Implementar un sistema multicapa de planificación en 3D mediante la aplicación de algoritmos genéticos, para optimizar trayectorias de robots aéreos no tripulados en ambientes parcialmente estructurados.

1.5.2 Objetivos específicos

- Investigar los diferentes algoritmos para la planificación RRT, para robots aéreos no tripulados en entornos parcialmente estructurados.
- Proponer un algoritmo para la planificación RRT en 3D de robots aéreos no tripulados, para tareas de locomoción en entornos de trabajo parcialmente estructurados.
- Desarrollar un algoritmo genético como optimización de trayectorias junto con un ambiente virtual para simular el comportamiento del robot con el ambiente.
- Evaluar el desempeño del algoritmo de planificación propuesto mediante la interacción con un entorno virtual parcialmente estructurado para robots aéreos no tripulados.

1.6 Variables de la Investigación

- **Variable independiente.**

Planificación RRT en 3D para robots aéreos no tripulados.

- **Variable dependiente.**

Algoritmo genético de optimización para trayectorias en 3D.

CAPITULO 2

2 MARCO TEÓRICO

2.1 Robots aéreos no tripulados

Un vehículo aéreo no tripulado, del inglés *Unmanned Aerial Vehicle* (UAV), o dron comúnmente llamado, es una clase de aeronave con la capacidad de realizar tareas u operaciones aéreas, capaz de mantener de forma autónoma un nivel de vuelo sostenido y controlado sin la necesidad de tripulación.

El control de los UAV's se lo puede realizar de dos maneras, a través de un operario humano situado en una estación de control en tierra o mediante un sistema autónomo de control de vuelo (Díaz Moreno, 2016).

La autonomía en los UAV's está relacionada con los diferentes algoritmos de control que estos poseen para responder de manera eficiente a los eventos inesperados, durante las misiones de vuelo (Escareño Castro, J. A., 2008).

Los UAV tienen una gran variedad de formas, tamaños, configuraciones y características dependiendo de la aplicación o uso que le desee dar el operario.

2.1.1 Clasificación de los UAV

Al momento de establecer una clasificación en los UAV se llega a diferentes criterios pues existe una clasificación dependiendo su misión principal, a su rango y distancia, al uso y tamaño, al número de hélices, al tipo de ala y al equipamiento; para una mejor

facilidad de entendimiento se clasificará en tres secciones básicas (Miguel González, 2017):

A. Drones rotatorios

1. Drones de rotor único: cuya funcionalidad y característica es la de cargar con ciertos pesos determinados y a su vez que sean resistentes como se muestra en la Figura 2.1; otra opción puede ser la de tener un rotor en la cola para proporcionar control al encabezado.



Figura 2.1: Drone de rotor único

Fuente: (xdrones.es, 2018)

2. Tricóptero: configurado típicamente en forma de “Y” o en “T” como muestra la Figura 2.2, equipado con tres potentes motores, tres controladores, cuatro giroscopios y un servo. Cada brazo de los tres del dron sostiene un sensor de localización y a su vez los motores se sitúan en cada extremo de los tres brazos. El tricóptero se mantiene estable en todo su camino, pero al momento de elevación tiene menos fuerza por el número de motores.



Figura 2.2: Drone tricóptero en forma de "Y"

Fuente: (Miguel Gonzáles, 2017)

3. Quadróptero: Tiene 4 motores DC sin escobillas colocados en una estructura simétrica, en tres configuraciones típicas: en "X", en "+" o en "H", dos motores giran en sentido horario y los otros dos en sentido anti horario para facilitar el aterrizaje y mantener el equilibrio en el vuelo tal como se ve en la Figura 2.3, usado comúnmente para las carreras de drones o para tomas de video o fotografía.



Figura 2.3: Drone Quadróptero tipo "X"

Fuente: (dji.com, 2018)

4. Hexacóptero: Equipado con 3 pares motores montados en los 6 brazos del dron típicamente separados a 60 grados en un marco simétrico como se ve en la Figura 2.4, son similares a los quadcópteros, pero los motores adicionales dan más capacidad de elevación y mayor seguridad si un motor deja de funcionar, pues el dron es lo suficientemente estable para aterrizar de mejor manera.



Figura 2.4: Drone Hexacóptero

Fuente: (Jonathan, 2016)

5. Octocóptero: Con ocho potentes motores montados cada uno en los ocho brazos como se muestra en la Figura 2.5, tienen muchas capacidades de vuelo y de estabilización en comparación a otros modelos discutidos anteriormente; tienen además mayor capacidad de carga, pero al tener muchos motores consumen mucha energía de las baterías; son utilizados comúnmente para fotografía aérea y para topografía.



Figura 2.5: Drone Octocóptero

Fuente: (Miguel Gonzáles, 2017)

6. Drones de ala fija: estos drones son totalmente diferentes a los drones de tipo multirotor, estos tienen alas que le dan la apariencia de aviones tradicionales como se muestra en la Figura 2.6. No son capaces de mantener estabilidad en el aire durante el vuelo, sus aplicaciones se encuentran en la grabación en movimiento y acciones de rescate.



Figura 2.6: Drone tipo Ala Fija

Fuente: (Miguel Gonzáles, 2017)

B. Según el tamaño

1. Drones muy pequeños: suelen ser diseñados desde el tamaño de un insecto de gran tamaño hasta los 50 cm de largo; los diseños más comunes son: mini drones y nano/micro drones. Son utilizados ampliamente en el área de espionaje debido a su construcción pequeña y peso ligero (Figura 2.7).



Figura 2.7: Micro Drone

Fuente: (MICRODRONE, 2014)

2. Mini drones: un poco más grande que los micro drones, es decir que sobrepasan los 50 cm pero no exceden los 2 metros de dimensión. Se suelen diseñar en mayor forma con alas fijas y pocos con alas rotatorias (Figura 2.8).



Figura 2.8: Mini Drone Tello

Fuente: (dji.com, 2018)

3. Drones de tamaño medio: tienen prototipos más pesados, pero más ligeros y son más pequeñas que las aeronaves. Pueden soportar peso de carga hasta los 200 kg. con una autonomía de vuelo entre 5 a 10 minutos (Figura 2.9).



Figura 2.9: Drone de Tamaño Medio Phantom

Fuente: (Phantom, 2018)

4. Drones de gran tamaño: tienen una estructura comparable al de los aviones y se los usa comúnmente en aplicaciones militares, para cubrir áreas en las cuales los aviones normales no pueden acceder (Figura 2.10).



Figura 2.10: Drone de Gran Tamaño MQ-1 Predator de General Atomics

Fuente: (CESPE, 2015)

C. Según el equipamiento

1. Con cámara: utilizados ampliamente por los aficionados a las cámaras de acción y expertos en filmación cinematográfica para la grabación de películas.
2. Con FPV: Visión en Primera Persona en español, usados comúnmente para las competencias de drones, ya que el usuario que los controla con la ayuda de unos binoculares puede ver en tiempo real el recorrido que hace el dron.
3. Con GPS: son capaces de obtener información sobre su ubicación estando en marcha, son capaces de grabar escenas en cualquier localidad y regresar a su punto de despegue luego de completar su tarea.

2.1.2 Aplicación de los UAV's

Los UAV's se encuentran en dos grandes categorías: aplicaciones militares y aplicaciones civiles.

A) Aplicaciones Militares.

La cualidad más importante de los UAV's como ya se mencionó es el vuelo no tripulado, esta característica permite evitar cualquier tipo de riesgo posible para los pilotos tanto en prácticas de dominio militar como en zonas hostiles (Díaz Moreno, 2016).

Las aplicaciones militares de los UAV's pueden ser:

1. Patrullaje y reconocimiento: realizan vigilancia o barrido fotográfico de la zona de interés para recolectar información sobre objetivos militares.
2. Combate: tipo de UAV equipado con misiles para acertar objetivos situados en tierra.
3. Entrenamiento militar: se usan UAV's como objetivos aéreos para el entrenamiento de futuros pilotos humanos.

B) Aplicaciones Civiles.

Tienen un buen potencial gracias a su versatilidad y flexibilidad de operación, pero no tienen el mismo desarrollo que han tenido los UAV militares. Existe un rango amplio de aplicaciones para los UAV's civiles, especialmente en tareas

peligrosas o de rutina (Díaz Moreno, 2016), en las cuales se puede mencionar las siguientes aplicaciones:

1. Vigilancia aérea: para la seguridad de grandes espacios como: control de incendios, seguridad en zonas públicas o privadas y vigilancia en autopistas además para la vigilancia con fines comerciales, todo esto se puede realizar con el uso de UAV's de bajo costo.
2. Producciones comerciales: gracias a las cámaras integradas en el sistema del UAV, la capacidad de maniobrar y conservar su estabilidad se han empleado drones para la grabación y filmación en tiempo real de sucesos, puesto que es mucho mejor que los sistemas convencionales de cámaras con cableado.
3. Búsqueda y rescate: usados en entornos de difícil acceso para la identificación de víctimas y su pronto rescate, esto gracias a los radares que llevan integrados; pueden ser usados además para la detección de fuego en incendios, tornados y erupciones, etc.
4. Transporte de carga: el uso de drones en compañías de transporte permitiría una entrega rápida, segura y económica hacia los clientes. Actualmente *Amazon* está realizando pruebas para comprobar la viabilidad de uso de este nuevo método de entrega de paquetes.

Existen aún muchas otras aplicaciones civiles de los UAV's, usados en la

investigación científica como: estudio de la atmosfera, el océano y la tierra, estudio del agua, cartografía además para cuidado de cultivos, preservación de entornos naturales, vigilancia urbana, etc.

2.2 Path Planning RRT

La planificación de trayectorias tiene por objetivo crear algoritmos que permitan determinar caminos que respeten las restricciones propias de los robots; desde que la autonomía de un robot móvil se hizo evidente surgieron problemas como colisiones con objetos fijos y móviles tanto en ambientes estructurados y no estructurados, así como la importancia de llegar desde un punto inicial a un punto final. Así se crearon dos métodos principales de path planning, en el primer caso están los métodos de planeación de path que consiste en encontrar en forma automática los path's libres de colisión, y en el segundo caso, el método de evasión de obstáculos que consiste en evadir los obstáculos que se presentan en la trayectoria del robot, mientras éste consigue alcanzar la meta (Parkin & Gonzáles, 2005).

No es relevante toda la información disponible para la planeación de trayectorias; es decir, mientras más datos se ingresen al sistema muchas más variables tendrá que controlar, volviéndose difícil dirigir al robot hacia el objetivo, un algoritmo de planificación de trayectorias será óptimo siempre y cuando encuentre un camino recomendable (Santana, Ortiz, Acosta, & Andaluz, 2018). La dificultad de generar trayectorias para un robot móvil convencionalmente es que, dado un robot y un ambiente, se puede generar una trayectoria entre dos puntos específicos si el robot

está libre de colisiones y si cumple con ciertos criterios de desempeño (Eduardo H., 2011) .

Existen varios algoritmos de planificación en el campo de la robótica, mayor parte de estos se puede agrupar o considerar cierta técnica desarrollada para la planificación de trayectorias, tales como:

- Algoritmos Determinísticos
 - Algoritmos Bug.
 - Basados en grafos.
 - Campos potenciales artificiales.
- Algoritmos Probabilísticos y Aleatorios
 - Mapas probabilísticos (Probabilistic Road Maps - PRM).
 - Árboles de exploración rápida (Rapidly Exploring Random Tree - RRT).
 - Basados en optimización.

A base del método de campos potenciales surgió una primera aproximación a los métodos de planificación aleatoria (Latombe, 2012), eliminando la cantidad de procesamiento computacional para el cálculo del potencial. Los algoritmos de planificación aleatoria debían de ser simples para aumentar la velocidad y reemplazar la falta de una inteligencia para buscar caminos (Lopez, Gomez-Bravo, Cuesta, & Ollero, 2006). Uno de estos métodos es el nombrado “Rapidly Expliring Random Trees”, RRT (LaValle, 1998).

2.2.1 Path Planning RRT 2D

El RRT enfrenta el problema de búsqueda de caminos usando para ello aproximaciones aleatorias que facilitan la rápida exploración de toda el área. La idea principal es seleccionar un punto aleatoriamente dentro de todo el entorno y conectarlo continuamente al árbol de búsqueda que se va generando. Los puntos generados aleatoriamente se van conectando y así formar lo que comúnmente se llaman “ramas” las cuales van expandiendo al árbol de búsqueda aleatorio (Fernández-Peteiro Belmonte, 2016).

Para entender el algoritmo se usarán los siguientes conceptos (López García, 2012):

C = Es el conjunto de todas las configuraciones posibles del robot en un espacio dado.

C_{free} = Es el subconjunto de C de las configuraciones libres de colisión en el entorno.

ρ = Métrica definida dentro de C . Puede ser distancia euclidiana u otra ponderación de proximidad que pueda interesar.

q_{ini} = Es la configuración inicial

q_{fin} = Es la configuración final que se desea alcanzar.

q_{rand} = Es una configuración aleatoria que genera el algoritmo dentro del espacio de configuraciones ($q_{rand} \in C$).

q_{near} = Es la configuración más próxima a q_{rand} , en el sentido definido por ρ de

entre las existentes en el árbol.

q_{new} = Es la configuración que se va a añadir al árbol.

ε = Longitud del segmento de crecimiento. Es la distancia entre un punto del árbol y el siguiente con que está conectado.

Árbol = Estructura de datos.

El objetivo del método RRT consiste en elaborar un árbol de exploración que cubra uniformemente todo el espacio de configuraciones libre de colisión (C_{free}), se desarrolla el algoritmo que se muestra en la Figura 2.11 (LaValle, 1998); el algoritmo tienen la misión de escoger un punto (q_{rand}) de forma aleatoria y extender hacia el punto el árbol de configuraciones.

```

Algoritmo_basico RRT(  $q_{ini}$  )
Arbol[0]=  $q_{ini}$ 
Para  $k = 1$  hasta  $Kmax$ 
     $q_{rand}$  ← Configuración_Aleatoria();
    Extiende(Arbol,  $q_{rand}$ );
Siguiente  $k$ 
Devuelve Arbol
  
```

Figura 2.11: Algoritmo RRT básico

Fuente: (López García, 2012)

El uso de la función “*Extiende*”, tiene el objetivo de ampliar el árbol en el sentido que indica (q_{rand}). El esquema de dicha función se muestra en la Figura 2.12.

El algoritmo comienza inicializando la tabla asociada al árbol como la configuración

origen. Seguidamente, entra a un bucle limitado por un valor $Kmax$, cuya función es finalizar la ejecución del algoritmo una vez que se haya realizado el número previamente fijado de iteraciones. Este valor se usará después para detener el algoritmo en caso de que no se alcance la configuración final. Cabe resaltar que la determinación de dicho valor dependerá de las características del problema (número de obstáculos, tiempo límite del algoritmo, etc.).

```

Extiende(Arbol,  $q_{rand}$ )
 $q_{near}$  = VecinoMásPróximo( $q_{rand}$ , Arbol);
    Si NuevaConfiguración( $q_{rand}$ ,  $q_{near}$ ,  $q_{new}$ )
Entonces
    AñadeVértice(Arbol,  $q_{new}$ );
    Si  $q_{new} = q_{rand}$  Entonces
        Devuelve "alcanzado"
    Si no
        Devuelve "avanzado"
    Si no
        Devuelve "rechazado"
  
```

Figura 2.12: Algoritmo RRT función Extiende

Fuente: (López García, 2012)

Dentro del bucle del algoritmo RRT hay dos instrucciones; con la primera se obtiene un punto al azar dentro del espacio de configuraciones (C); la segunda permite crecer al árbol en dirección a la configuración aleatoria anteriormente obtenida (Asqui, Andaluz, Sánchez, & Acosta, 2018).

El crecimiento del árbol se consigue mediante la función *Exiende*. La estructura de esta función comienza con el cálculo de q_{near} . Esto se realiza gracias a la función *VecinoMásPróximo* que aplica la métrica ρ definida anteriormente a todos los vértices del árbol, obteniendo el punto más cercano a q_{rand} . Seguidamente, la función *NuevaConfiguración* calcula q_{new} , el nuevo punto a agregar, mediante un salto de tamaño ε partiendo de q_{near} en dirección a q_{rand} , como muestra la Figura 2.13. Para la obtención de q_{new} se tiene en cuenta si hay alguna colisión en dicho desplazamiento, devolviendo “verdadero” o “falso” según un movimiento posible o, por lo contrario, colisione con algún obstáculo.

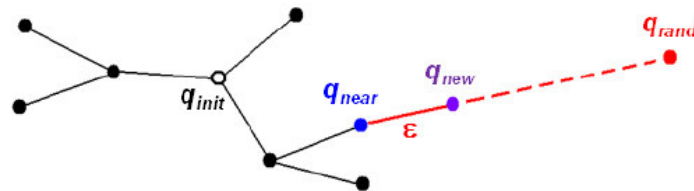


Figura 2.13: Esquema de expansión del RRT

Fuente: (López García, 2012)

De no detectarse colisión se agrega el nuevo punto al árbol distinguiendo entre dos casos. Si el punto aleatorio pertenecía a un círculo de centro q_{near} y radio ε , entonces q_{new} coincidirá con q_{rand} y, por tanto, dicho punto ha sido “alcanzado”, el algoritmo notificará tal circunstancia. Por el contrario (caso más general), no se ha dado el alcance, se devolverá el valor “avanzado”. Por último, en caso de que la función *NuevaConfiguración* haya advertido de la existencia de algún obstáculo en el camino que une q_{near} con q_{new} , se informa de que no ha habido nuevas ramas y el punto de es incluido al árbol.

El comportamiento de este algoritmo con respecto a otros es mejor en cuanto a la homogeneidad del espacio explorado. La naturaleza del RRT le exige al principio continuar con más rapidez hacia zonas inexploradas del entorno, pues es allí donde existe más posibilidad de que se defina q_{rand} . Esto se puede comprender observando el crecimiento del árbol en distintos entornos. Por ejemplo, la progresión del algoritmo representado a través de la Figura 2.14.

Para que un punto determinado del árbol agregue una rama, es necesario que q_{rand} quede más cerca al punto que a ningún otro. Es decir, que q_{rand} pertenezca a la región de Voronoi adjunto a dicho punto. Si q_{rand} se genera de forma equiprobable en todo el entorno de configuraciones, y éste se divide en regiones de Voronoi, la probabilidad de que q_{rand} pertenezca a una de estas regiones será igual al cociente entre el tamaño de la región y el del espacio de configuraciones.

En la Figura 2.14, el espacio libre de colisión consiste en un círculo, esto significa que, desde el punto central, no existe ninguna preferencia en cuanto a la dirección de crecimiento. Se puede observar que las ramas iniciales del árbol inician en direcciones aleatorias y crecen sin influencia entre ellas.

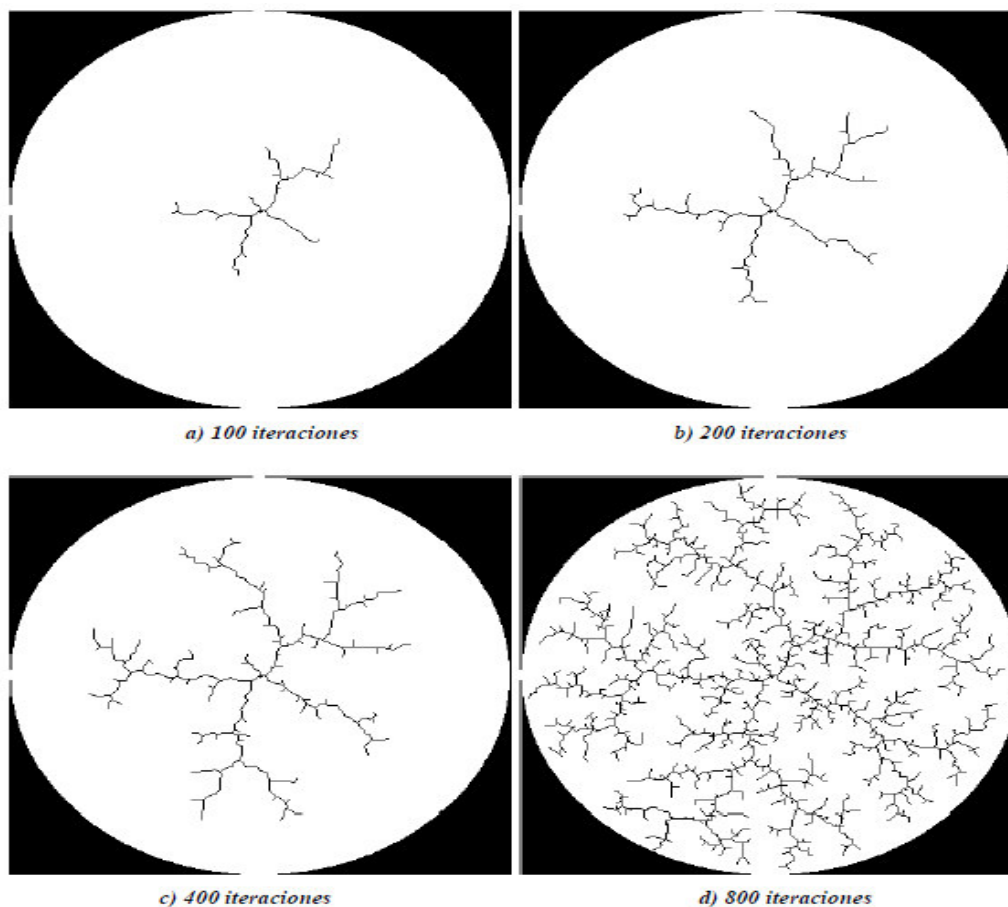


Figura 2.14: Evolución del Algoritmo RRT en un entorno circular.

Fuente: (López García, 2012)

La naturaleza del RRT evita la existencia de una alta densidad de ramas en el origen. Las regiones de Voronoi de los puntos periféricos del árbol al principio son las más extensas, por lo que el crecimiento se invierte en la prolongación de unas pocas ramas. Conforme el árbol va llenando el espacio libre, el proceso decae y las ramas aumentan en número surgiendo por doquier; se puede observar que el árbol logra una densidad uniforme sobre el espacio libre del entorno.

Ahora las Figuras 2.15 a, b, y c. El entorno en este caso es cuadrado, y se muestra tanto el árbol RRT como las regiones de Voronoi asociadas a sus puntos. En este caso

las ramas tienden a crecer hacia las esquinas, dado que en estas direcciones hay más configuraciones que en otras, y ellos se traducen en una mayor probabilidad de establecer q_{rand} .

En las Figuras 2.15 d, e, f, y g se representa la progresión del algoritmo RRT sobre un espacio donde el desequilibrio en regiones de Voronoi es más acusado. Por lo que se sitúa el origen del árbol RRT a un lado del escenario. En la evolución se aprecia que, desde el punto de origen, colocado dentro del rectángulo pequeño, parte una rama que crece más con respecto a las demás, debido a la alta frecuencia con lo que aparecen puntos aleatorios y estos puntos dirigen el crecimiento.

Al igual que en la Figura 2.14, tras las suficientes iteraciones, se puede comprobar que la densidad de las ramas resulta razonablemente homogénea a pesar del desequilibrio inicial existente entre los dos rectángulos.

A continuación se expone una lista de características de los algoritmos basados en RRT (LaValle, 1998), cuyos dos primeros elementos se acaban de explicar:

- La expansión de los algoritmos basados en RRT se inclina decididamente hacia los espacios inexplorados.
- Son simples, facilitando el análisis de comportamiento y la implementación en cualquier escenario.
- Un RRT es un módulo que puede ser implementado en otros planificadores.

- No requieren una definición explícita de C_{free} sólo utiliza una función que compruebe la existencia de colisión.
- Es probabilísticamente completo, es decir, la probabilidad de encontrar un camino si ha de existir tiende a 1 exponencialmente con el número de nodos.

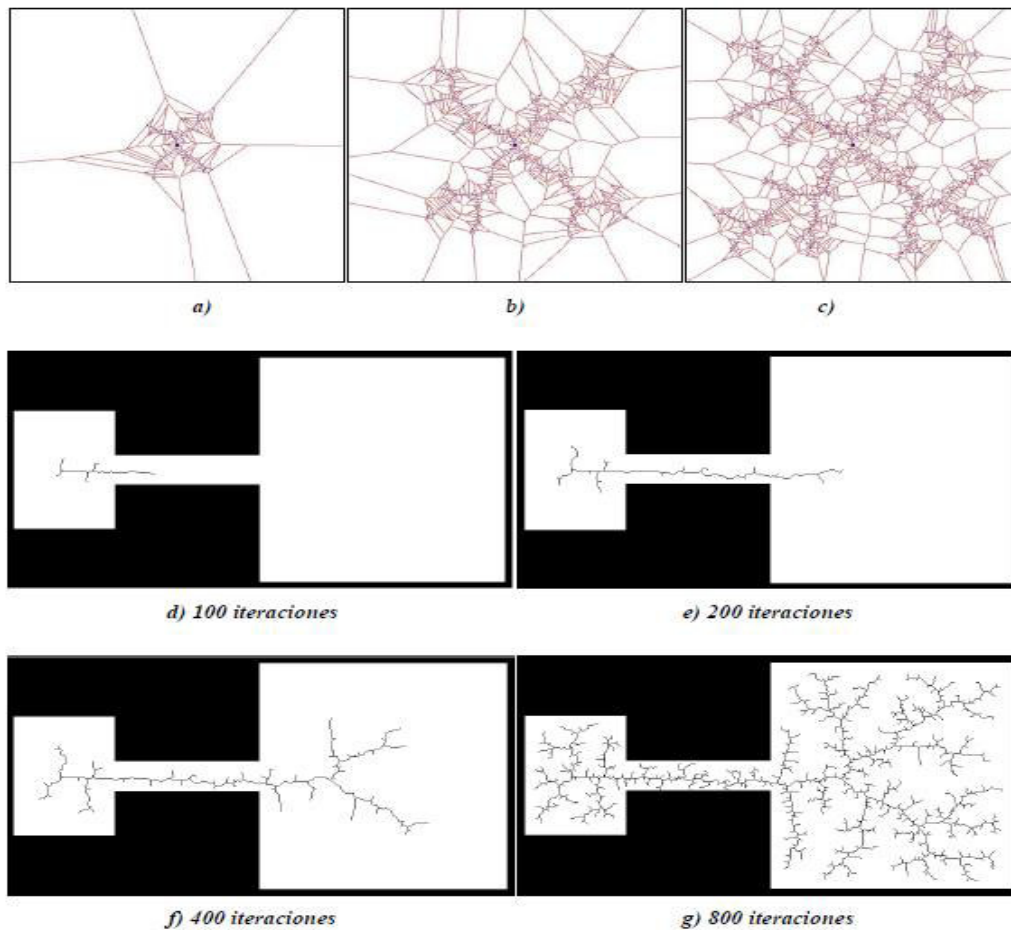


Figura 2.15: Evolución del Algoritmo RRT en un entorno asimétrico

Fuente: (López García, 2012)

Por último, se advierte que el resultado de la función “*Extiende*”, no se utilizan en este algoritmo que sólo se ocupa para generar un árbol capaz de explorar de modo

equiprobable todo el espacio libre. Esta peculiaridad atiende al diseño inicial del RRT como estructura de datos a implementar sobre otros algoritmos de planificación (López García, 2012).

Existen varias extensiones, así como otras mejoras del RRT que permite establecer un camino entre un origen y un destino, una de ellas es la adaptación del algoritmo RRT – Bidireccional (LaValle & Kuffner, 2000), que se basan en la generación simultanea de dos árboles cuyas raíces son las configuraciones inicial y final. Dichos árboles crecen explorando el espacio vacío y buscándose entre sí hasta conectar, obteniendo como resultado una trayectoria.

2.2.2 Path Planning RRT 3D

El algoritmo RRT es un algoritmo probabilístico utilizado para explorar entornos rápidamente y planear caminos en estos. Los entornos pueden estar o no con obstáculos, el algoritmo tiene la capacidad de trabajar en 2 dimensiones o en múltiples dimensiones (Figura 2.16). Se puede especificar la cinemática para hacer crecer el árbol y planear el camino con obstáculos estáticos o en movimiento. Los inconvenientes de este algoritmo son (Abbadi, Matousek, Jancik, & Roupec, 2012):

1. No es óptimo, ya que depende mucho de la probabilidad y el azar para elegir el siguiente punto de crecimiento y los puntos de camino a seguir.
2. La cantidad de puntos redundantes que se usan para explorar el camino desde el punto de origen hasta el punto de destino es muy grande.

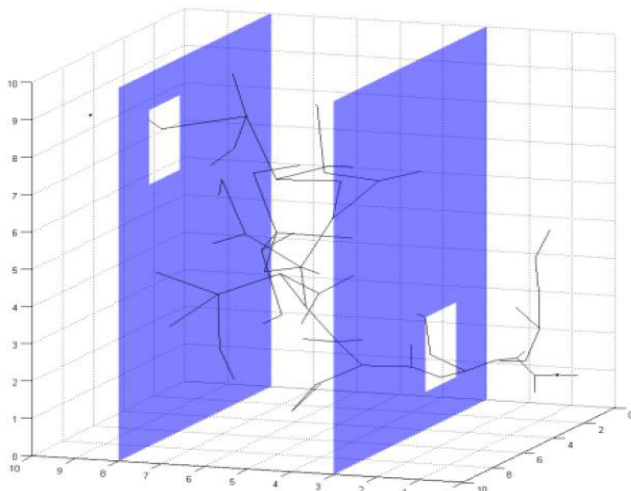


Figura 2.16: Algoritmo RRT en 3D

Fuente: (Abbadi et al., 2012)

Algunas variaciones fueron desarrolladas para luego ser introducidas al algoritmo básico del RRT; con el propósito de hacer más veloz al algoritmo y a incrementar la posibilidad de alcanzar el destino tan rápido como sea posible. Se puede en general clasificar las variaciones del RRT en algunas clases basadas en la función principal del RRT (Abbadi, Matousek, Minar, & Soustek, 2011).

La primera clase de estas variaciones depende en aumentar la velocidad de exploración al incrementar el número de árboles para explorar el entorno, por ejemplo, usar dos árboles en vez de uno, así el primer árbol empezaría en el punto de inicio y el segundo árbol en el punto destino y cuando estos dos árboles se encuentren cerca se encontraría el camino deseado.

La segunda clase de estas variaciones depende en la forma de extender las ramas del árbol. Es decir, extender el valor de la rama por la distancia de ε . En otras variaciones en lugar de usar el valor de ε se usa un método para conectar

completamente la distancia desde q_{near} a q_{rand} usando una rama o varias ramas entre estos dos puntos.

La tercera clase de estas variaciones depende en escoger q_{rand} , algunas variaciones usan libremente la opción de escoger un punto aleatorio en el entorno, otras variaciones muestrean el entorno primero y luego escogen un punto aleatorio de estas muestras. Esta opción sirve en gran mayoría para que el propio algoritmo tenga la decisión de que punto escoger y así considerar de igual manera los obstáculos presentes en todo el entorno.

Para el crecimiento del árbol de acuerdo a la tercera clase de las variaciones del algoritmo RRT se compone en 4 fases (Luan & Fang, 2016). Primero, las partículas son distribuidas por un muestreo uniforme en todo el espacio tridimensional. Segundo, el árbol aleatorio se construye rápidamente entre estas partículas. Tercero, un cilindro es usado para simplificar la función de comprobación de colisión en cada iteración. Cuarto, se realiza un suavizado en el camino encontrado para obtener un camino óptimo.

2.3 Algoritmos Genéticos

2.3.1 Definición de Inteligencia Artificial

Antes de definir el término Inteligencia Artificial (IA) deberíamos definir lo que se entiende por inteligencia. La Real Academia de la Lengua Española nos define la inteligencia como “Potencia intelectual facultad de conocer, de entender o comprender”. Una vez definido esto. La inteligencia artificial se podría definir de la misma forma, pero

aplicado a las maquinas. De esta forma, cuando nos dicen que una maquina es inteligente tendríamos que preguntar que conocimiento o comprensión tiene dicha lavadora del proceso que realiza. Como podemos observar, este término es muy ambiguo y se han aducido múltiples argumentos a favor y en contra de la inteligencia en las maquinas(Galipienso, Quevedo, Pardo, Ruiz, & Ortega, 2003).

Marvin Minski fue uno de los primeros desarrolladores en la inteligencia artificial y la define de la siguiente manera: “La inteligencia Artificial es la ciencia de construir máquinas para que hagan cosas que, si las hicieran los humanos requerirían inteligencia” («EN INGENIERÍA ELECTRÓNICA», 2014). Entonces se puede entender como IA una ciencia la cual involucra compartir conocimientos en las actividades o acciones con un objetivo específico y que estas tengan éxito en la tarea asignada.

Charles Darwin y la Teoría de la evolución.

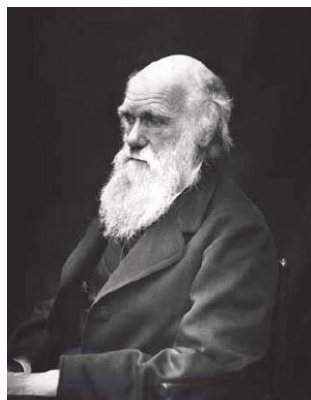


Figura 2.17: Darwin, Charles Robert (1809 - 1882)

Fuente: (Cruz, 2011)

Científico británico que sentó las bases de la moderna teoría evolutiva (Figura 2.17), al plantear el concepto de que todas las formas de vida se han desarrollado a lo

largo de un proceso de selección natural que involucra millones de años. Su trabajo tiene una influencia decisiva en varias disciplinas científicas, la religión y el pensamiento moderno en general (Cruz, 2011).

En 1831 inició una expedición científica alrededor del mundo en el barco de reconocimiento HMS Beagle, que duró cinco años durante los cuales recopiló una cantidad inmensa de datos y anotaciones sobre geografía, geología, botánica y zoología, así como un gran número de muestras de las que obtendría información suficiente para escribir varios libros. En 1838 ya había perfilado su teoría de la evolución del mecanismo de la selección natural, que presentó 20 años después, el 1 julio de 1858 en la Linnean Society de Londres. El 24 de noviembre de 1859 se puso a la venta la primera edición de *El origen de las especies por selección natural* (Cruz, 2011).

La teoría de Darwin sostiene que la variación entre las especies ocurre al azar y que la supervivencia o extinción de cada organismo está determinado por la capacidad de dicho organismo a adaptarse a su medio ambiente (Darwin, 2009), con los siguientes puntos clave:

- Las especies tienen gran fertilidad.
- Las poblaciones permanecen aproximadamente del mismo tamaño, con fluctuaciones muy pequeñas.
- Los alimentos son limitados, pero relativamente constantes la mayor parte del tiempo.
- En ciertas condiciones habrá lucha por la supervivencia entre individuos.

- En la reproducción sexual, dos individuos no serán idénticos.
- La variación es extensa, muchas de estas variaciones serán heredadas

2.3.2 Técnicas mediante Algoritmo Genéticos

Un algoritmo genético se define como una estructura de datos que admita las posibles soluciones a un problema. Cada uno de los posible conjunto de datos admitidos por esa estructura será una solución al problema. Unas soluciones serán mejores que otras. Solucionar un problema consiste en hallar una solución óptima, y por tanto, los algoritmos genéticos son en realidad un método de búsqueda para encontrar una posible solución (Lahoz-Beltra, 2010).

A finales de los 1950s y principios de los 1960s, el biólogo inglés Alex S. Fraser (1923-2002) publicó una serie de trabajos sobre la evolución de sistemas biológicos en una computadora digital, sirviendo de inspiración para los algoritmos genéticos denominado, Simulación de sistemas genéticos por computadoras digitales automáticas (Fraser, 1957).

Hans-Joachim Bremermann (1926-1996) fue el primero en ver la evolución como un proceso de optimización, además de realizar una de las primeras simulaciones con cadenas binarias que se procesaban por medio de reproducción, selección y mutación (Bremermann, 1962).

John Henry Holland desarrolla a principios de los 1960s los “planes reproductivos” y “adaptativos” en un intento por hacer que las computadoras aprendan imitando el

proceso de la evolución lo cual da como auge el desarrollo de algoritmos genéticos en varios campos (Holland, 1962).

Como indica su nombre es un algoritmo que surge de ideas de evolución natural en un proceso de optimización el cual tiene características específicas por tener operación empleadas en la naturaleza como son:

- Selección
- Cruzamiento
- Mutación

Para poder desarrollar el algoritmo estas operaciones deben ser evaluadas de forma numérica que se puede encontrar en cadena de bits (0,1), esta representación sea la información de cada individuo que conforma una población, puede haber varias formas de representación, pero en forma binaria es la más utilizada. Al hablar de algoritmos genéticos se trata de métodos de optimización, al hablar de optimización el problema se plantea de la siguiente manera:

$x_o \in X$ tal que f es un máximo en x_o , donde $f: X \rightarrow R$, por lo tanto:

$$f(x_o) = \max_{x \in X} f(x) \quad (1)$$

Es casi imposible obtener una solución en sentido estricto. Dependiendo del problema planteado, puede ser suficiente encontrar el máximo valor o el más cercano al valor máximo. f es una función asignada para definir el valor de “aptitud” para cada

individuo. Se asumiría que S es un arreglo de cadenas (en un caso no trivial en algunos aspectos de la gramática).(Cruz, 2011) Proponiendo que X es el espacio deseado para la optimización, se tendrá la siguiente función:

$$c : X \rightarrow S$$

$$x \rightarrow c(x)$$

Esta función se llama *función codificación*. Inversamente, la función

$$\tilde{c} : S \rightarrow X$$

$$s \rightarrow \tilde{c}(s)$$

Se llama *función de decodificación*. Estas funciones deben ser especificadas dependiendo de las necesidades del problema que se plantea. Para un mejor apreciamiento de las diferentes expresiones que se usa en AG a relación con la genética de evolución sus definiciones son equivalentes a:

- Genotipo= AG (código de cadena)
- Fenotipo= AG (punto sin codificar)
- Cromosoma= AG (cadena)
- Gen= AG (posición de cadena)
- Alelo= AG (valor en una posición determinada)
- Función de aptitud o aptitud= AG (valor de la función objetivo)

Genotipo: se refiere a la estructura genética de un organismo la cual contiene la información existente en el genoma.

Fenotipo: características físicas de un organismo, atribuibles a la expresión de su fenotipo. Contiene tanto los rasgos físicos como los conductuales. Es el resultado de la interacción entre el genotipo y el ambiente; se interpreta como la suma de los caracteres observables en un individuo. Es la manifestación externa del genotipo (Garamszegi, 2014).

Cromosoma: es la molécula única de ADN unida a histonas (proteínas básicas) y otras proteínas que se condensa durante la mitosis (proceso de división celular - reparto equitativo del material hereditario) y la meiosis (proceso de fragmentación - divisiones pequeñas), formando una estructura compacta (Kluger, 2015).

Gen: especifica la herencia de un carácter; está formado por una secuencia de aminoácidos de una o más cadenas de ARN (ácido ribonucleico - interviene en diferentes neuronas, en la expresión de la información genética), que realizan diferentes funciones en la cadena (Garamszegi, 2014).

Alelo: el valor de un gen. Una de las dos o más formas alternativas de un gen; determina el carácter controlado por el gen. Un ejemplo es el *diploide* que contiene dos juegos de cromosomas, por lo tanto, tiene dos copias de cada gen (Kluger, 2015).

Función de aptitud: es un tipo especial de función que cuantifica la optimalidad de una solución. Se traduce en un cromosoma óptimo para que sus bases sean

combinadas con cualquier otra técnica para la producción de una nueva generación que sea mejor a las anteriores(Lahoz-Beltra, 2010).

La evolución natural promueve la copia de información genética entre varias generaciones de individuos la cual consta en las estructuras básicas de un AG como son:

Selección: mecanismo de selección individual (cadena) para la reproducción acorde con la *función de aptitud* (valor de la función objetivo). Los algoritmos de selección serán los encargados de escoger qué individuos van a disponer de oportunidades de reproducirse y cuáles no. La idea básica de selección está asociada con la *función de aptitud* y el sistema original; para implementación es comúnmente conocida como *roulette-wheel* (RWS); ésta utiliza una distribución de probabilidad, donde la probabilidad de selección de una cadena es directamente proporcional a su *aptitud* (Nilsson, 2014).

Cruzamiento: método de fusión sobre la información genética de dos individuos; si la codificación se elige apropiadamente, dos progenitores saludables producirán descendientes sanos. Es el principal operador genético; provee un mecanismo para heredar características a su descendencia; interviene en ambos progenitores (Steels, Brooks, & Brooks, 2018).

Mutación: en la evolución real, el material genético puede ser alterado en forma aleatoria debido a un error en la reproducción o la deformación de genes; un ejemplo es la radiación de los rayos gama. En los AG, la mutación se realiza, con gran

probabilidad, como una deformación aleatoria de las cadenas. Produce cambios incrementales al azar en la descendencia, efectuando cambios aleatorios en los valores del alelo en algunos genes. En el caso de cromosomas binarios, le corresponde hacer los cambios de posiciones en cada bit (Cohen & Feigenbaum, 2014). No afecta a toda la población, pero es probable que dañe a algunos también tiene el efecto de perturbar de manera segura a los cromosomas a fin de introducir nuevas características que no estaban presentes en ningún elemento de los progenitores.

Reemplazo: procedimiento para calcular (crear) una nueva generación de la anterior y sus descendientes. Se crea un espacio a la descendencia en la población eliminando de ella a los padres (Kluger, 2015).

CAPITULO 3

3 DESARROLLO DEL ALGORITMO

3.1 Sistema de planificación

Como se explicó en el capítulo anterior, el objetivo final de la planificación de trayectorias es que el robot cumpla con la obtención de trayectorias seguras, adecuadas y con una gran eficiencia en desplazamiento en el entorno. Cabe recalcar que es importante tomar en cuenta el modelo matemático (cinemático y dinámico) del robot que se va a utilizar para poder tener una óptima planificación de trayectorias.

En forma general el sistema de planificación mediante un sistema multicapas que consta de *i) capa de mapeo* mediante el cual se realiza la reconstrucción de entornos en 3D; *ii) capa RRT 3D* está orientada al desarrollo de una arquitectura de planificación en 3D, la cual evita obstáculos; *iii) capa de optimización y control* donde se muestra el control de cada una de las variables más importantes para darle autonomía al sistema como se muestra en la Figura 3.1.

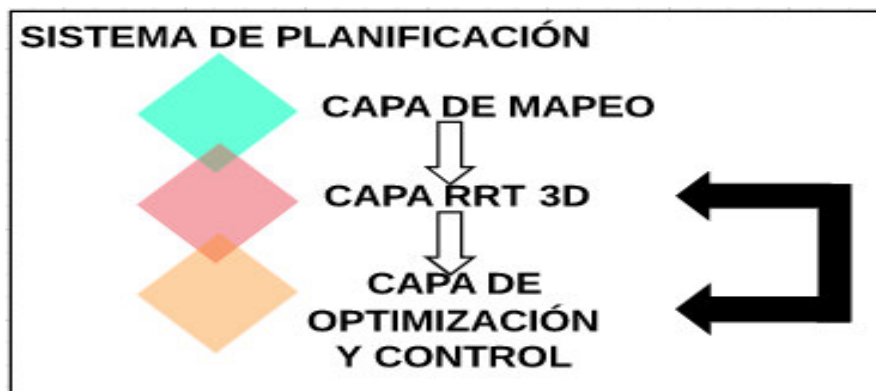


Figura 3.1: Descripción del sistema planificación.

3.1.1 Capa de Mapeo

El ambiente necesario para el funcionamiento del algoritmo de planeación, empieza con una imagen en dos dimensiones (2D) denominada también imagen en vista superior o vista de águila, en este caso la será una imagen en RGB (H. Wang et al., 2016) que cada color respectivo indicara el relieve de altura que posee la materia como se muestra en la Figura 3.2, la cual contiene información en el plano $R(X,Y)$ tanto de espacio libre de navegación y de edificios que son obstáculos a evadir, para luego transformarla en una imagen en tres dimensiones (3D).

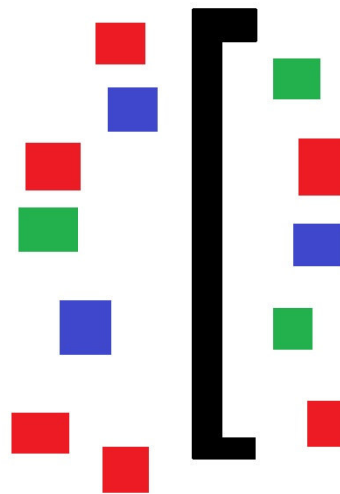


Figura 3.2: Entorno 2D en imagen vista superior.

La reconstrucción del ambiente en 3D, inicia con la imagen en RGB y procesando la información se la transforma en una imagen en escala de grises invertida, se toma las dos entornos y se realiza una interpolación lineal en el software Matlab para así obtener una matriz con información sobre la superficie y la altura de los

obstáculos en el plano Z, esta depende de la tonalidad de la escala de gris siendo el color blanco la máxima altura y el color negro el suelo del entorno (Schotland, 2000) indicada en la Figura 3.3.

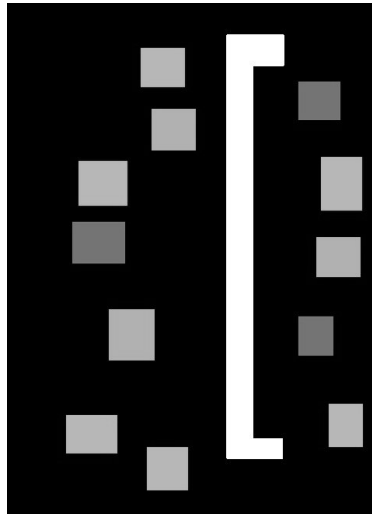


Figura 3.3: Entorno 2D en escala de grises invertida.

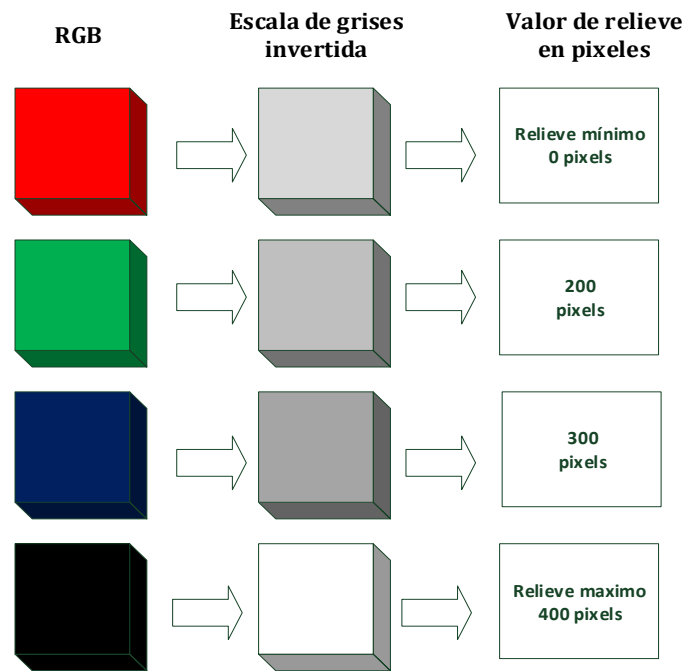


Figura 3.4: Valor en pixeles del relieve de materia a partir de RGB.

Finalmente con la información en 2D posición, dimensión y altura de los obstáculos presentes, se genera un entorno con todas las estructuras que están presentes en la imagen en 2D (Yiqun, Jun, Bin, Youmin, & Jianliang, 2015), se reconstruye el entorno en 3D haciendo un barrido en cada coordenada existente a partir de la imagen en 2D y tomando el valor de relieve correspondiente al que se genera en la imagen a escala de grises invertida a partir de la coloración RGB como muestra la Figura 3.4. La materia en cada posición existente es representada con un círculo en color magenta en el plano $R(X,Y,Z)$ correspondiente tal como se indica en la Figura 3.5. Los entornos reconstruidos en Matlab, tienen como unidad de medida el pixel, con el cual se puede escalar a cualquier magnitud de longitud,

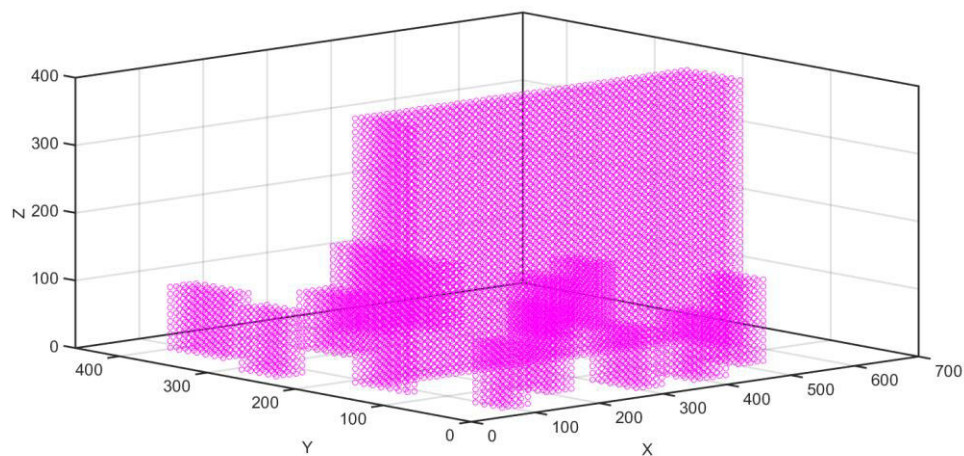


Figura 3.5: Entorno 3D reconstruido a partir de imagen en 2D.

3.1.2 Capa RRT 3D

Este algoritmo se enfoca en la reconstrucción de un árbol randómico de posiciones que incrementa a partir de un punto de origen (Dong, Fu, & Kayacan, 2016). El presente

algoritmo es una variación del RRT en 2D enfocado a un ambiente en el espacio que posee los valores límites de navegación en el entorno, donde el UAV representa una coordenada de partida en el espacio denominada ($qstart$) con condiciones iniciales en el plano 3D, así como también se define la coordenada de llegada ($qgoal$) (Luan & Fang, 2016), el correcto funcionamiento del algoritmo RRT 3D posee un número adecuado de nodos a generar ($qnewnode$) para encontrar las coordenadas suficientes hasta encontrar un punto de llegada, como muestra la Figura 3.6, caso contrario de no existir suficientes nodos no habría un alcance para un punto de llegada.

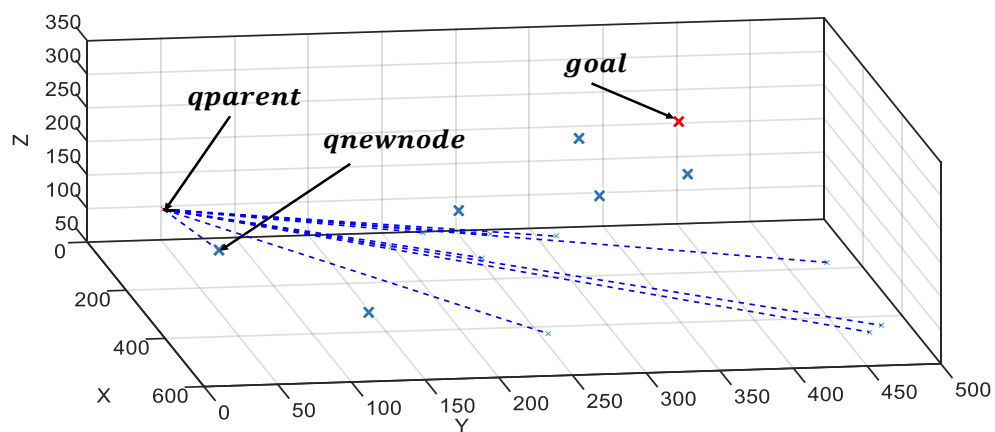


Figura 3.6: Nodos generados y punto de llegada.

Con todos los parámetros necesarios el algoritmo RRT 3D inicia generando partículas aleatorias en el espacio libre desde el punto de partida ($qstart$), con sus respectivas proyecciones, como se muestra en la Figura 3.7; luego de las proyecciones generadas se realiza una búsqueda dentro del radio de una esfera, para encontrar todos los nodos existentes siendo el origen la coordenada del punto padre ($qparent$) y la posición del nuevo nodo ($qnewnode$) es la coordenada más cercana al centro de la esfera.

Posteriormente la iteración continúa en el espacio libre siendo ahora el punto padre la coordenada del punto hijo y generando nuevamente las proyecciones de las partículas, encontrando un nuevo punto hijo como se muestra en la Figura 3.7.

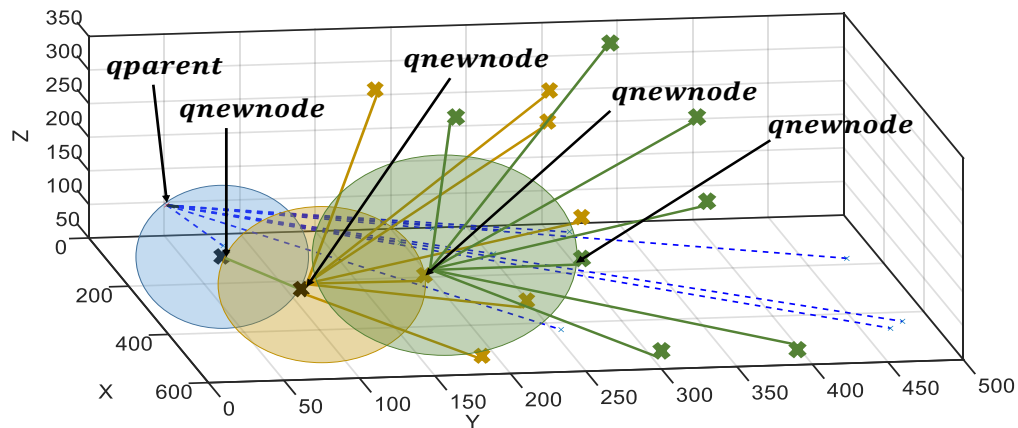


Figura 3.7: Generación del nuevo nodo a partir del nuevo punto padre.

Al tener generadas estas proyecciones de las partículas, el algoritmo a través de todos los vecinos más cercanos encuentra rutas alternativas de menor distancia y la posición del nuevo punto padre es la coordenada final de la función costo (Li, Li, Cheng, & Song, 2012); la función costo (*cost value*) es una pre-optimización y determina la longitud entre la suma de distancias de dos proyecciones adyacentes, y la compara con la distancia entre el nodo inicial y el nodo final de la iteración, de esta manera se obtiene la menor magnitud de longitud entre los nodos, como indica en la Figura 3.8, este proceso se realiza cuantas veces se haya establecido el número de nodos hasta encontrar el punto de llegada.

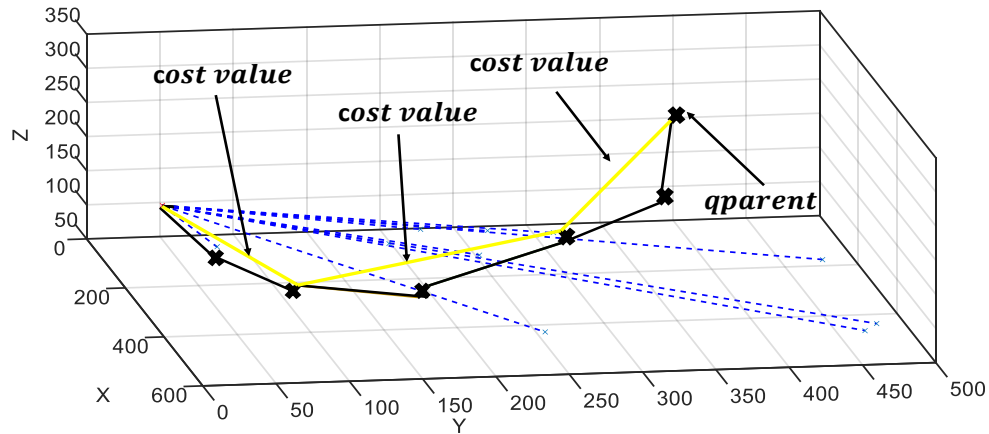


Figura 3.8: Trayectoria de menor distancia por la función costo.

En el entorno generado se aplica el algoritmo de planeación RRT 3D desarrollado, generando en el espacio libre las partículas aleatorias con sus respectivas proyecciones como se muestra en la Figura 3.9 y trayectorias determinadas por la función costo (*cost value*) mostrada en la Figura 3.10.

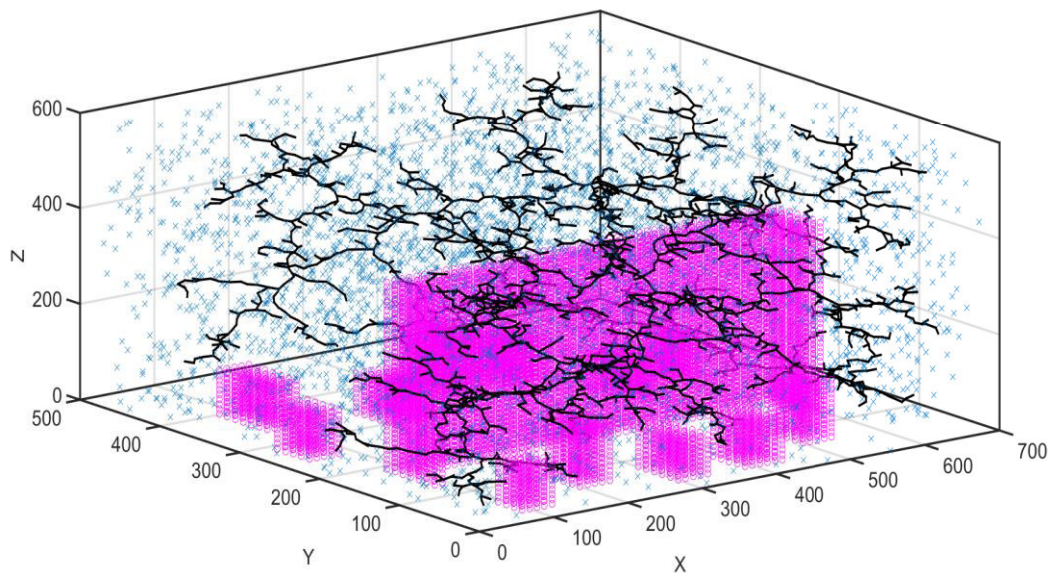


Figura 3.9: Proyecciones de las partículas en el entorno.

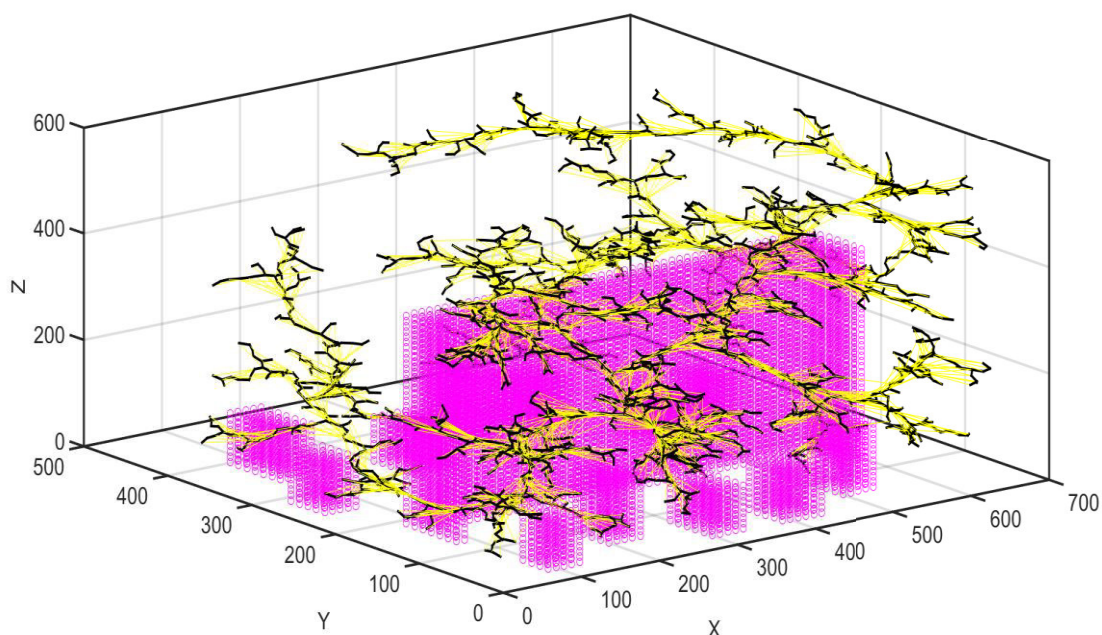


Figura 3.10: Trayectorias determinadas por la función costo.

Posteriormente, el algoritmo RRT 3D finaliza de hacer proyecciones cuando la nueva coordenada padre ($q_{newnode}$) es igual al punto de llegada (q_{goal}) establecido en las condiciones iniciales como se observa en la Figura 3.11. El algoritmo RRT 3D planifica seleccionando todos los nodos con continuidad desde el punto inicial hasta el final, cada una de las posiciones en $R(X,Y,Z)$ de los nodos seleccionados forman un camino que tiene como origen la coordenada inicial (q_{start}) y su finalización es la coordenada de llegada (q_{goal}) la trayectoria formada ($path$) se muestra de color rojo en la Figura 3.11 y la misma evita el contacto con la materia en el entorno (Samaniego, Sanchis, Garcia-Nieto, & Simarro, 2017).

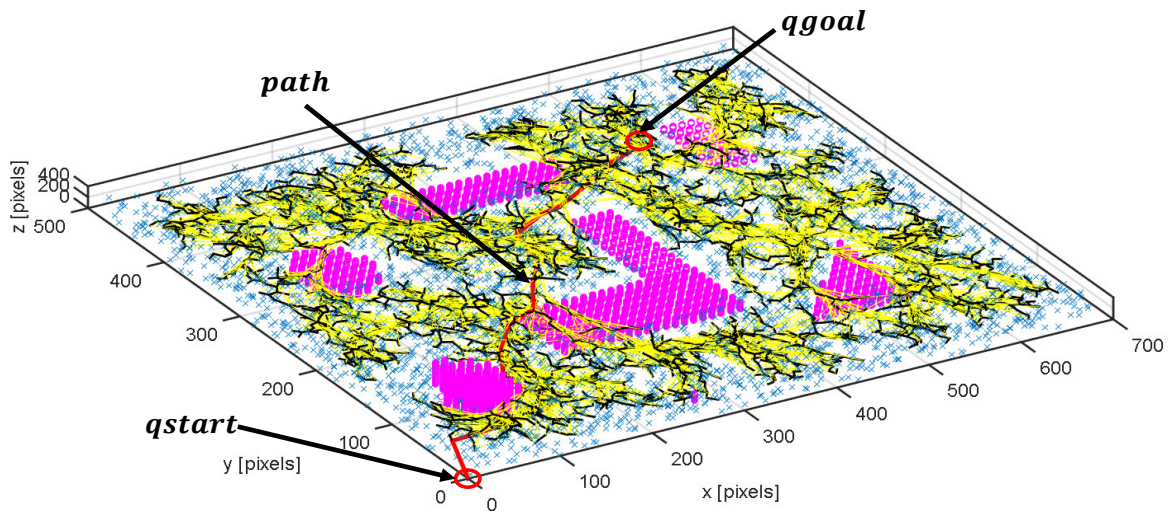


Figura 3.11: Generación trayectoria RRT 3D.

El desarrollo del algoritmo RRT 3D se programó bajo el pseudocódigo descrito a continuación en el PseudoCode1:

PseudoCode1:

Procedimiento RRT 3D

```

1  BEGIN
2  Initial conditions(xqmax, yqmax, zqmax, qstart, qgoal)
3  Mapping();
4  Nodes();
5  For i=1:#nodes
6  BEGIN
7    While(qfree==true)
8    qrand =Randomicposition();
9  END
10 If qgoal==true Then
11   Break;
12 END
13   Nearest_node(map);

```

```
14   Neighbor_parent(radius);
15   Costvalue(qnear);
16   Newparent(costvalue);
17   Appendnodes(E);
18   END
19   While(qendparent==true)
20     BEGIN
21     Lastfreeposition();
22     E=Appendnewtrajectory(qnew);
23   If qnew==true Then
24     Break;
25   END
26     Path Seleccction();
27     Smoothing Layer
28   END
```

3.1.3 Capa de Optimización y Control

La optimización es el proceso de hacer algo mejor en ingeniería, los algoritmos de optimización han sido ampliamente desarrollados y bien utilizados en todos los aspectos durante mucho tiempo. Un ingeniero o un científico evoca una nueva idea y una optimización para mejorar esa idea. La optimización consiste en intentar variaciones en un concepto inicial y el uso de la información. Muchos problemas de optimización del mundo de la ingeniería, en particular los sistemas de fabricación, son de naturaleza muy compleja y bastante difícil de resolver mediante técnicas de optimización convencionales para lo cual el uso del ordenador y con las premisas

respectivas se lo puede realizar en este caso no se tiene una función objetivo para evaluarla, pero se parte de la misma premisa que es una población inicial para poder aplicar dichas técnicas de optimización.

A. Algoritmos Genéticos

Los algoritmos genéticos son procesos de optimización en donde se busca interpretar y recrear eventos de reproducción natural. Generando un cálculo de exploración mediante técnicas estocásticas que reproducen los criterios de selección natural y genética aplicadas (Olguín, 2008). En una formación muy general el algoritmo genético depende de una población que representa un conjunto inicial de posibles soluciones de donde cada elemento dentro de la población se denomina cromosoma. Estos pueden reproducirse acorde los procesos de iteración la cual representa una solución aplicable a los sistemas, esta secuencia de iteraciones sucesivas se llaman generaciones. En estas generaciones los cromosomas pueden evaluar el progreso mediante medidas de aptitud. (Arranz de La Peña & Parra Truyol, 2017).

Todo el proceso de evolución biológica se hace de forma natural, pero para aplicar el algoritmo genético en la resolución de problemas se sigue una serie de pasos. Una de las ideas principales es conseguir que el tamaño de la población sea lo suficientemente grande para garantizar una diversidad de soluciones. La población preferiblemente debe ser generada de forma aleatoria para obtener diversidad, en el caso de que la población no sea generada de forma aleatoria,

habrá que tener en cuenta que se garantice una cierta diversidad en la población de descendientes generada como que se muestra a continuación en la Figura 3.12.



Figura 3.12: Diagrama de un algoritmo genético simple.

Los algoritmos genéticos se consideran procesos robustos de búsqueda evolucionada, los cuales para este problema específico donde el problema principal es encontrar rutas libres de colisión, de donde se presentan posibles vectores de solución que son tratados como parámetros de conjuntos soluciones.

En donde:

- σ Factor de selección
- S Vector de probabilidades
- h Identificador de espacios dentro de un espacio de referencias \mathfrak{R}^3
- k Identificador de posiciones en el vector
- e Factor de evolución
- r Factor de reproducción
- m Factor de mutación
- A Vector de posición

i Posición de inicio del vector

f Posición final del vector

Dichos algoritmos operan con poblaciones de elementos posición en:

$$A_h(k) = \{A_h^k, \dots, A_n^k\}; \quad \forall h = x, y, z \in \mathfrak{R}^3; \quad \forall k = 1, \dots, n \quad (2)$$

donde cada iteración k , para los elementos x_k, y_k, z_k , representan un punto de búsqueda en el espacio de configuraciones de soluciones probables. La aptitud de cada individuo se evalúa a partir de una función de probabilidad $S(S_x, S_y, S_z)$, la cual le asigna una adaptación al proceso de identificación adecuado, mostrado en el PseudoCode2:

PseudoCode2:

Procedimiento Algoritmo Genético

1 Inicio

2 Inicialización de parámetros (map, Xi, Yi, Zi)

3 inicializar población A (Ax, Ay, Az);

4 construir población PA (Px, Py, Pz);

5 Mientras (no se cumpla la existencia de población probable) hacer

6 Inicio(2)

7 $j = j + 1$

8 elección PM(j) hasta PM(j-1)

9 interacción PM(j)

10 mutación PM(j)

11 valorar PM(j)

12 Final(2)

13 Inicio(3)

14 $j = j + 1$

- 15 elección PV(j) hasta PA(j-1)
 - 16 interacción PV(j)
 - 17 evolución PV(j)
 - 18 valorar PV(j)
 - 19 **Final(3)**
 - 20 **Final(1)**
-

Dentro del proceso la población inicial se interactúa sucesivamente hacia lugares adecuados dentro del espacio de configuración donde se realiza la exploración mediante procesos estocásticos, tales como: a) selección de población inicial más adaptada a la solución, b) mayor grado de probabilidad de mutar, c) mayor grado de probabilidad de evolucionar, de todos los elementos identificados a una población apta para solucionar las trayectorias principales libres de colisión.

Definición Tamaño de la Población Inicial o Reproducción natural

$$\forall S_x, S_y, S_z \in \mathfrak{R} \Rightarrow S_{hk} = \frac{A_{hi} + A_{hf}}{\sum A_{hk}} \quad (3)$$

$$, m < S_{nk} < e ,$$

$$\forall k = 1, \dots, n ; \forall h = x, y, z \in \mathfrak{R}^3$$

Las coordenadas de búsqueda $\vec{x}_k = (x_{k1}, x_{k2}, \dots, x_{kn})$; $\vec{y}_k = (y_{k1}, y_{k2}, \dots, y_{kn})$; $\vec{z}_k = (z_{k1}, z_{k2}, \dots, z_{kn})$, donde se simula el número de cromosomas que tenemos en nuestra población inicial, determinando una generación (Figura 3.13). De no ser una medida con soluciones probables, el algoritmo genético no tendrá las suficientes

posibilidades de realizar reproducciones de esta manera se realiza una exploración de resultados de aplicabilidad baja y escasos.

Existencia del elemento de reproducción natural

$$\exists!(r) \in \mathcal{R} ; \forall \sigma \in \mathcal{R} / m < \sigma < e \quad (4)$$

Al comprobarse la existencia de un elemento de reproducción natural válido, (4) esto quiere decir una posición tanto de probabilidad que cumpla ser un valor entre los parámetros m , e , todas estas probabilidades calculadas de la ecuación (3), así como de coordenadas que se encuentren en el rango de pixeles que forman el mapa y que cumplan con estar en una posición libre de colisión. Si al tener una población excesiva, el algoritmo genético se vuelve excesivamente lento. Existen estudios que revelan la existencia de un límite a partir del cual es ineficiente aumentar el tamaño inicial de población ya que no se produce una mayor velocidad al solucionar el problema (A. M. Andaluz, 2016).

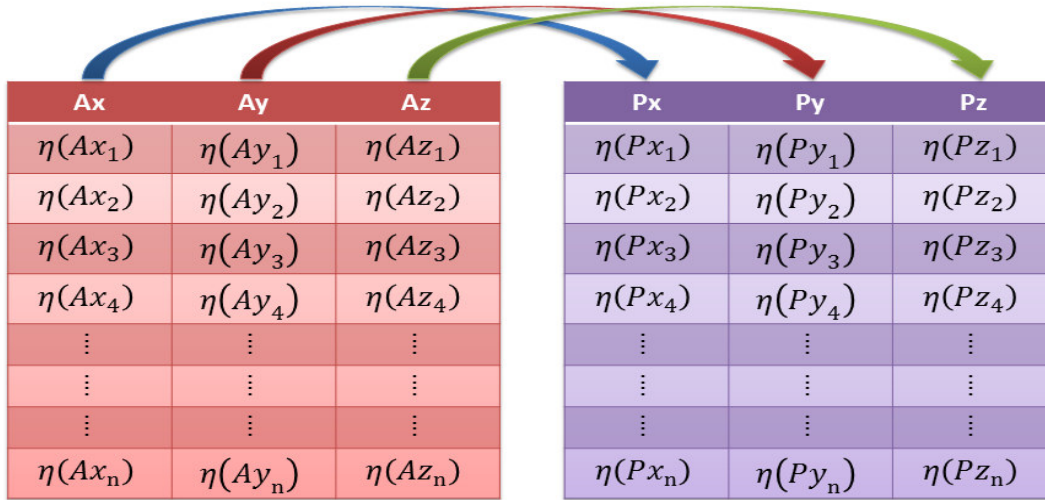


Figura 3.13: Muestras y probabilidades de la población.

Definición Probabilidad de evolución

$$\forall S_x, S_y, S_z \in \mathfrak{R} \Rightarrow S_{hk} = \frac{A_{hi} + A_{hf}}{\sum A_{hk}} \quad (5)$$

$$, S_{hx} \geq e ,$$

$$\forall k = 1, \dots, n ; \forall h = x, y, z \in \mathfrak{R}^3$$

Identifica con qué frecuencia se producen cruces entre los cromosomas considerados padre, donde existe la probabilidad para reproducirse entre los mismos (Figura 3.13). La probabilidad de que no exista evolución, entonces los hijos se consideran copias exactas de los padres (A. M. Andaluz, 2016).

Existencia del elemento de evolución

$$\exists !(e) \in \mathfrak{R} ; \forall \sigma \in \mathfrak{R} / \sigma \geq e \quad (6)$$

Al comprobarse la existencia de un elemento de evolución válido (6) esto quiere decir, una posición tanto de probabilidad que cumplan ser valores mayores e iguales al parámetro ℓ , todas estas probabilidades calculadas de la ecuación (5), así como de coordenadas que se encuentren en el rango de pixeles que forman el mapa y que cumplan con estar en una posición libre de colisión. Cuando este factor exista, los hijos tienden a tener partes de los cromosomas de los padres. Al tener una probabilidad de evolución del 100%, el hijo se crea totalmente por evolución (Jorge Arranz de la Peña & Parra Truyol, 2007).

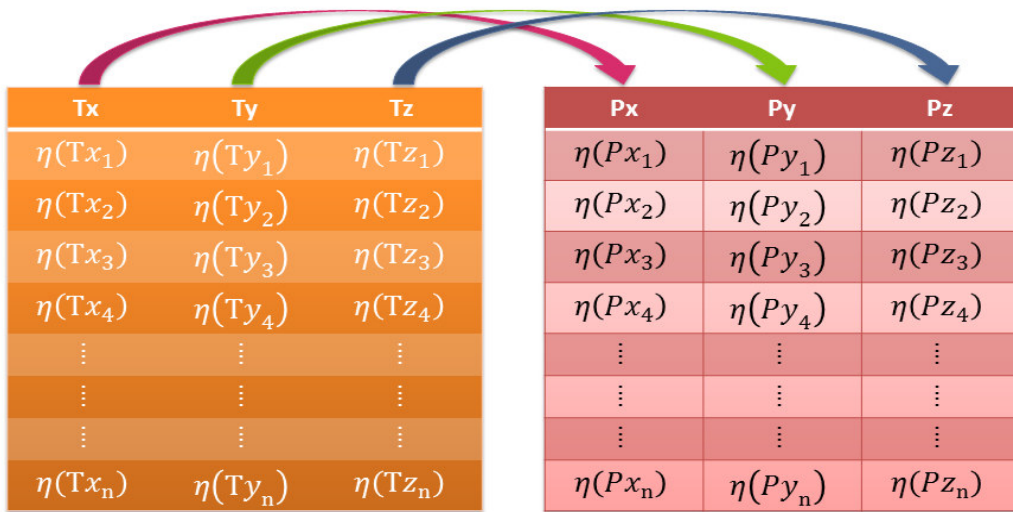


Figura 3.14: Muestras y probabilidades de evolución.

Definición Probabilidad de Mutación

$$\forall S_x, S_y, S_z \in \mathfrak{R} \Rightarrow S_{hk} = \frac{A_{hi} + A_{hf}}{\sum A_{hk}} \quad (7)$$

$$, S_{hx} \leq m ,$$

$$\forall k = 1, \dots, n ; \forall h = x, y, z \in \mathfrak{R}^3$$

Muestra la frecuencia con la que genes en un cromosoma se mutan. Cuando no exista mutación, la descendencia se da por los mismos que tenían tras la reproducción (Figura 3.14). Si se identifican mutaciones, parte del cromosoma descendiente se ve modificado, así también si la probabilidad de mutar equivale al 100%, el cromosoma es cambiado en su totalidad (A. M. Andaluz, 2016) .

Existencia del elemento de mutación

$$\exists ! (m) \in \mathfrak{R} ; \forall \sigma \in \mathfrak{R} / \sigma \leq m \quad (8)$$

La existencia de un elemento de evolución valido (8) esto quiere decir, una posición tanto de probabilidad que cumplan ser valores menores e iguales al parámetro m , todas estas probabilidades calculadas de la ecuación (7), así como de coordenadas que se encuentren en el rango de pixeles que forman el mapa y que cumplan con estar en una posición libre de colisión. Al determinarse la mutación el cambio del número de bits del cromosoma se produce en su totalidad, lo que se indica una transformación del cromosoma, resultando una rápida degeneración de la población resultante (Jorge Arranz de la Peña & Parra Truyol, 2007).

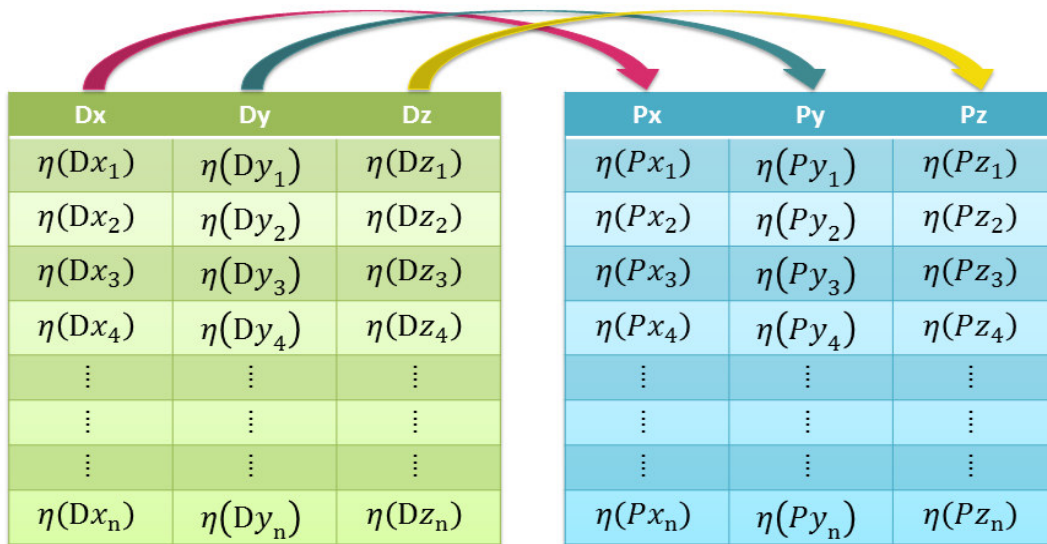


Figura 3.15: Muestras y probabilidades de mutación.

B. Operaciones de los algoritmos genéticos

Para resolver el problema de planificación mediante los algoritmos genéticos se debe parametrizar el proceso hacia la solución mediante en una serie de variables, que se codifican en un cromosoma, de acuerdo a esto tenemos un factor de selección de reproducción natural σ el cual nos indica si nos encontramos en una evolución o mutación al estar entre valores de los factores ℓ, m , respectivamente.

Estos operadores son utilizados en algoritmos genéticos los cuales se aplicarán sobre dichos cromosomas que representan poblaciones de ellos en posibles soluciones de coordenadas para la planificación. Dentro del algoritmo genético se implica el método para resolver el problema de planificación, así también son algoritmos independientes del problema, haciéndolo un algoritmo

robusto, que puede ser usado en cualquier ámbito de acción, y a la vez débil, por no ser un proceso de especialización.

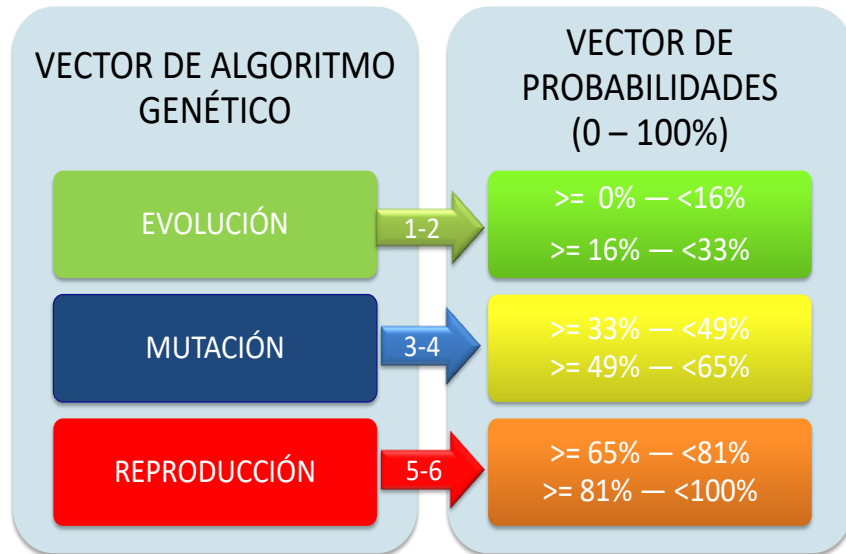


Figura 3.16: Interpretación del vector probabilidad para decisión del algoritmo genético.

Cada una de las componentes del vector $(\vec{x}_k, \vec{y}_k, \vec{z}_k)$, se codifica en probabilidades usando porcentajes. Luego las representaciones de probabilidad de cada parámetro se concatenan en una sola pila, obteniéndose individuos de población (S_x, S_y, S_z) (9).

$$Sx = \frac{Ax_i + Ax_f}{\sum_{k=1}^{k=n} Ax_k} \quad Sy = \frac{Ay_i + Ay_f}{\sum_{k=1}^{k=n} Ay_k} \quad (9)$$

$$Sz = \frac{Az_i + Az_f}{\sum_{k=1}^{k=n} Az_k}$$

El factor de selección, simula los procesos genéticos de tres grupos ordenados por pares, desde la probabilidad menor valor hacia el mayor valor las que se representan por posiciones en un vector; donde las misma representan: 1-2 (Evolución 0-33%), 3-4 (Mutación 34-65%), 5-6 (Reproducción 66-100%) según la tabla de la Figura 3.16. Cada dato de curva se representa por colores tales como: Reproducción Natural (*), Mutación (*), Evolución (*).

$$P_{k,t} = \frac{f_i}{\sum_{k=1}^n f_k} \quad (10)$$

La probabilidad de selección $P_{k,t}$ del i-ésimo individuo (10) en la población $P(t)$ depende de la adaptación relativa de éste con respecto a la población (Figura 3.17)

$$x_k \in |Px_k - Sx_k| \in \mathfrak{R}, Px_k > Sx_k, \forall k = 1, \dots, n;$$

$$y_k \in |Py_k - Sy_k| \in \mathfrak{R}, Py_k > Sy_k, \forall k = 1, \dots, n; \quad (11)$$

$$z_k \in |Pz_k - Sz_k| \in \mathfrak{R}, Pz_k > Sz_k, \forall k = 1, \dots, n;$$

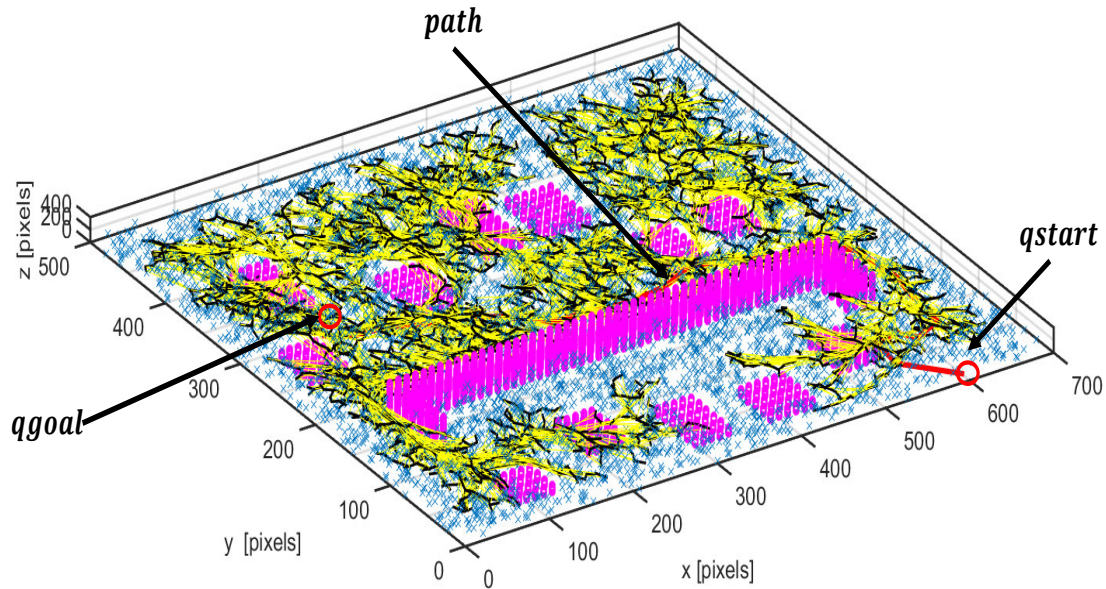


Figura 3.17: Interpretación del vector población para decisión del algoritmo genético.

La Figura 3.18 muestra una ilustración equivalente a la posición del vector de probabilidades, donde el algoritmo genético evalúa el valor de selección junto con los valores del vector probabilidad, esto quiere decir que cuenta las posibles soluciones de mutación y evolución de la función de probabilidad entregada, la cual para este caso define una trayectoria diferente. En este caso, los puntos de mutación y evolución (color azul, verde) se superponen a los puntos de trayectoria identificados en color magenta.

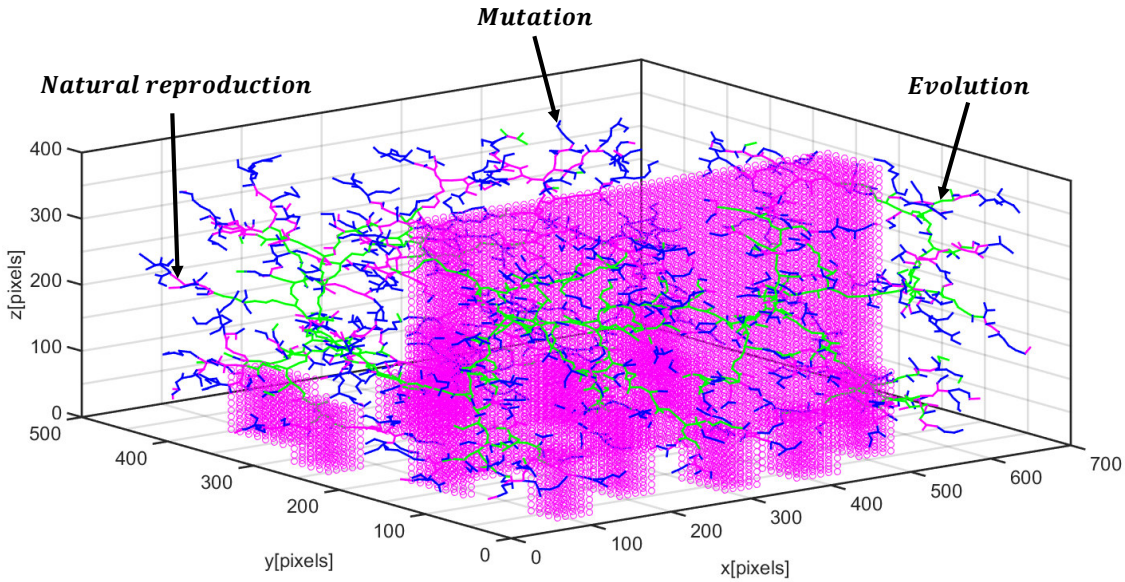


Figura 3.18: Interpretación del vector probabilidad para decisión del algoritmo genético.

Una vez calculadas todas las posibles soluciones, se realiza una búsqueda de mínimos locales por método de descenso de gradiente para encontrar posibles soluciones (12),

$$x_k \min f(x_{k1}, y_{k1}, \dots, x_{kn}, y_{kn}), \quad (12)$$

$$x_k \in [m_k, e_k] \in \mathfrak{R}, m_k \leq o \leq e_k, \forall k = 1, \dots, n$$

$$y_k \in [m_k, e_k] \in \mathfrak{R}, m_k \leq o \leq e_k, \forall k = 1, \dots, n,$$

$$z_k \in [m_k, e_k] \in \mathfrak{R}, m_k \leq o \leq e_k, \forall k = 1, \dots, n,$$

En la Figura 3.19, se muestra las trayectorias libres de colisión en donde poseen una probabilidad de evolución, probabilidad de mutación y probabilidad de reproducción

equivalentes al vector de respuestas posibles, donde el algoritmo genético evaluó previamente los valores cercanos a la población inicial, esto quiere decir que su trayectoria equivalente (12) de la función probabilidad entregada define tres trayectorias probables libres de colisión.

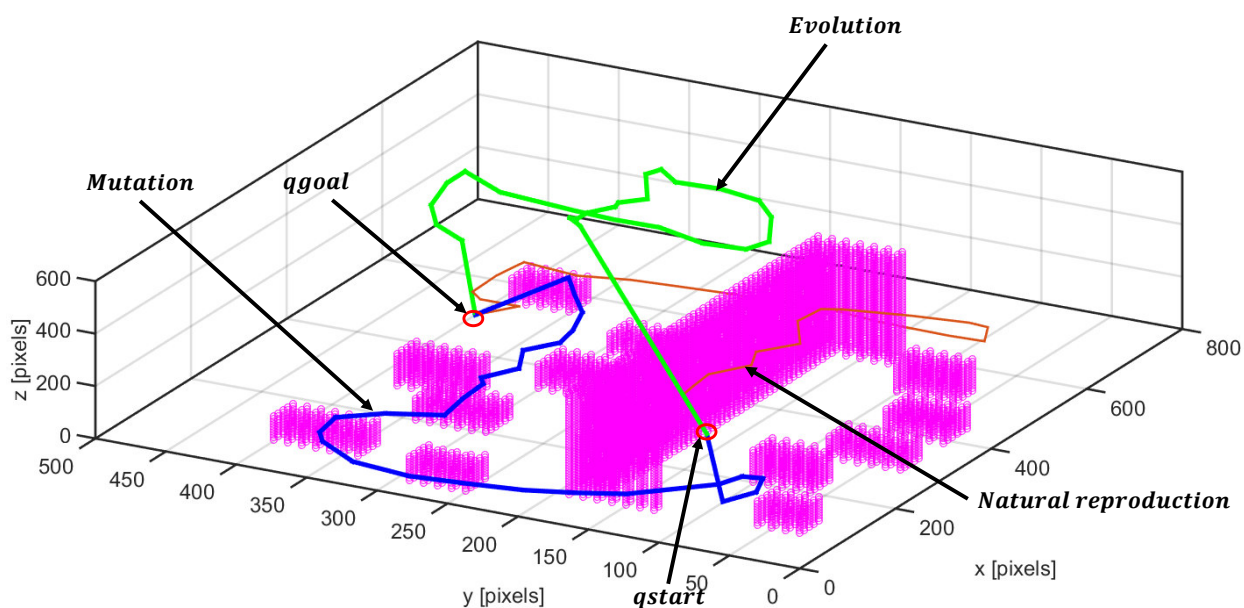


Figura 3.19: Trayectorias del algoritmo genético.

C. Optimización por distancias euclidianas

En este caso no se posee una función definida al proceso debido a que es un proceso randómico que genera coordenadas el espacio cartesiano de tres dimensiones. Algunos de los factores de optimización para la navegación en robots aéreos no tripulados son: distancia, tiempo, eficiencia energética, entre otros; en este caso el algoritmo realiza la optimización mediante el factor de distancia, la cual será directamente proporcional al tiempo de navegación al suponer que el UAV se desplace

con una velocidad casi constante sin tomar en cuenta efectos de la dinámica en su desplazamiento.

Para comprender el proceso que realiza el algoritmo partiremos desde la premisa entre la distancia entre dos puntos los puntos A y B tienen tres coordenadas cada uno. Para encontrar la distancia euclidiana $A(x_1, y_1, z_1)$ y $B(x_2, y_2, z_2)$, se utiliza la fórmula:

$$AB = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (13)$$

El factor de decisión τ indica la distancia más corta de las trayectorias probables y consiste en obtener el valor mínimo de la sumatoria de distancias euclidianas desde el punto inicial ($qstart$) hasta cada uno de los puntos que forma toda la trayectoria terminando la sumatoria en la coordenada final ($qgoal$) como indica en la Figura 3.20 y definida en la fórmula (17):

$$\sum_{i=1}^n (Px_i, Py_i, Pz_i) \quad (14)$$

$$\sum_{i=1}^n (Vx_i, Vy_i, Vz_i) \quad (15)$$

$$\sum_{i=1}^n (Dx_i, Dy_i, Dz_i) \quad (16)$$

$$\tau = \min(P, V, D) \quad (17)$$

En donde P (14), V (15), D (16) son las coordenadas de las trayectorias probables.

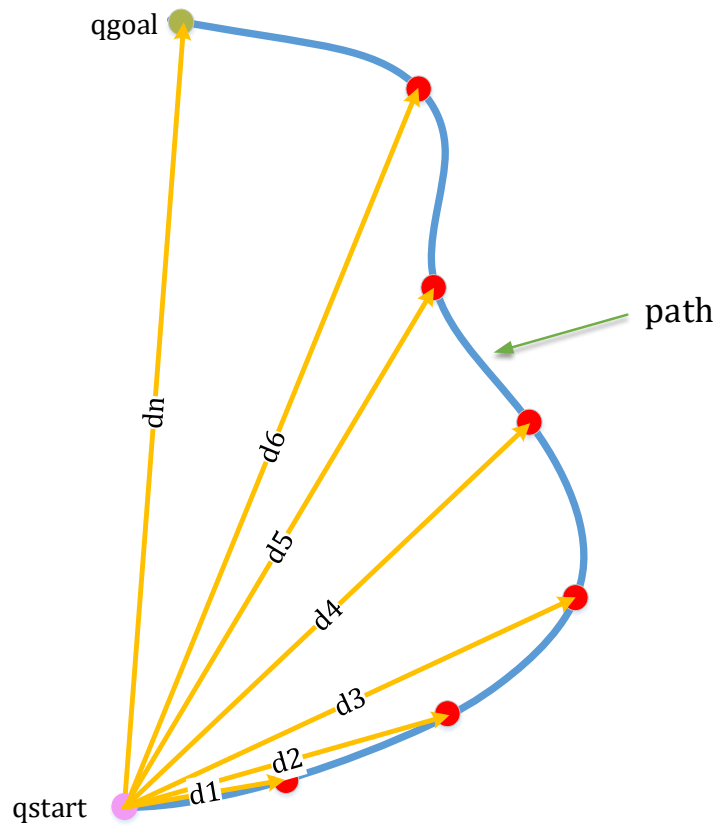


Figura 3.20: Distancias euclidianas en las trayectorias

A continuación (Figura 3.21), se indica una de las tres trayectorias, equivalente al vector de probabilidades, donde el cálculo de distancias euclidianas (17) y gasto de tiempo en bucles usados con el procesador son los más reducidos elegidos por el algoritmo, resultan en una trayectoria final libre de colisiones que corresponde a la más óptima para la navegación tanto en distancia como en tiempo.

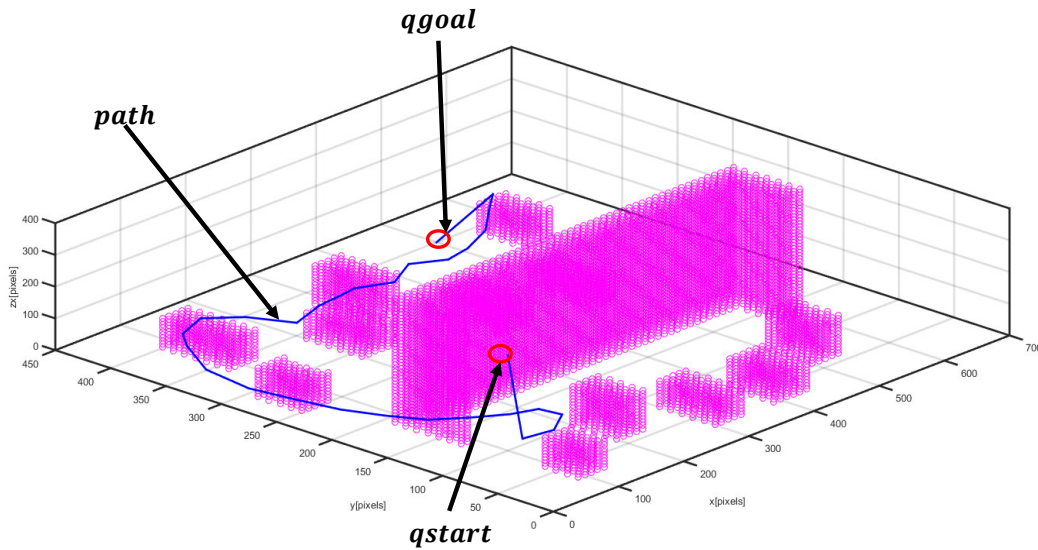


Figura 3.21: Selección de la trayectoria (*path*) óptima para la navegación

3.1.4 Controlador de UAV's

Modelo Cinemático de un UAV

El modelo cinemático del UAV está conformado por un conjunto de cuatro velocidades representadas en el espacio $R(x, y, z)$ que rotan sobre un ángulo φ respecto al eje z . Las velocidades lineales son u_1, u_m y u_n en el marco del espacio $R(x, y, z)$ así como su velocidad angular ω . En tal caso, cada velocidad lineal se proyecta desde el centro de gravedad del UAV hacia arriba u_n con el eje z , hacia adelante u_1 con el eje x y u_m hacia la izquierda con el eje y . La velocidad angular ω gira el marco del plano espacial en sentido anti horario alrededor de su eje z (V. H. Andaluz, Ortiz, & Sánchez, 2015) así como se muestra en la Figura 3.22.

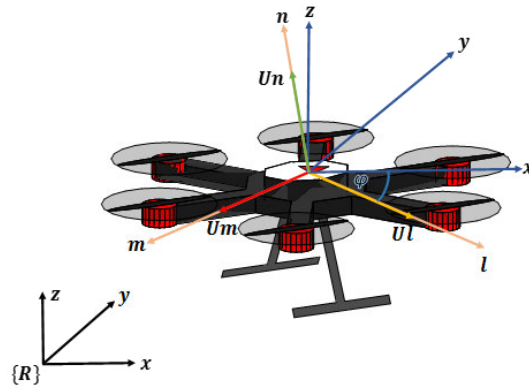


Figura 3.22: Representación en el plano de un UAV

El comportamiento cinemático del robot UAV en el marco inercial $R(x, y, z)$ puede ser representado matricialmente como (Hongpeng Wang, Li, Zhang, He, & Liu, 2017):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) & 0 & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{u}_l \\ \mathbf{u}_m \\ \mathbf{u}_n \\ \boldsymbol{\omega} \end{bmatrix} \quad (16)$$

Expresado vectorialmente como $\dot{\mathbf{h}} = \mathbf{J}\boldsymbol{\mu}$, donde: $\dot{\mathbf{h}}$ es el vector de posiciones en x, y, z y φ ; \mathbf{J} es la matriz Jacobiana que contiene la información cinemática del UAV y $\boldsymbol{\mu}$ es el vector de velocidades en x, y, z y φ .

Tener en cuenta que el punto de interés o también llamado punto de control, se encuentra en el centro de masa del UAV.

Controlador de Seguimiento de Caminos

El diseño de los controladores cinemáticos para el UAV, permite alcanzar en cualquier instante de tiempo su objetivo de navegación, el funcionamiento de control del sistema se indica en la Figura 3.23.

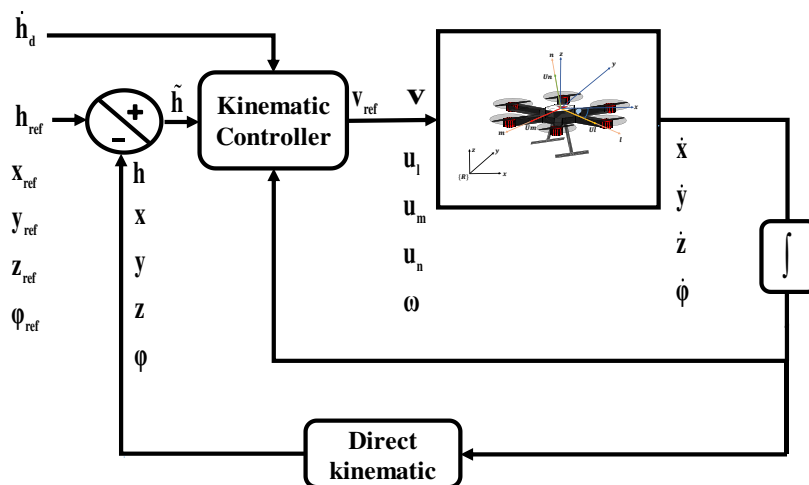


Figura 3.23: Diagrama de bloques del control

La posición deseada del vehículo sigue la tangente a una curva designada en una tarea, se propone una velocidad v_d para el desplazamiento del punto de referencia entre dos posiciones consecutivas a lo largo del camino. La velocidad v_d define qué tan rápido la posición deseada varía a lo largo del camino. Generalmente, un valor constante v_{\max} es adoptado para v_d .

Sin embargo, un enfoque de movimiento se implanta, teniendo en cuenta la distancia ρ más corta entre el camino y el vehículo. Es decir $v_d = \frac{v_{\max}}{1+k\rho}$, donde k es

una constante positiva, lo que impide grandes valores de la señal de control siempre que el vehículo esté lejos de la posición deseada, también permitiendo que el vehículo alcance el camino.

Las posiciones del camino a seguir en un cierto instante de tiempo están definidas por el vector, $\mathbf{h}_{ref} = [x_{ref} \ y_{ref} \ z_{ref} \ \phi_{ref}]^T$, obteniendo la primera derivada descrita por (Brandao, Andaluz, Sarcinelli-Filho, & Carelli, 2011):

$$\begin{bmatrix} \dot{x}_{ref} \\ \dot{y}_{ref} \\ \dot{z}_{ref} \\ \dot{\phi}_{ref} \end{bmatrix} = \begin{bmatrix} \frac{v_{max}}{1+k\rho} \cos \alpha \cos \beta \\ \frac{v_{max}}{1+k\rho} \cos \alpha \sin \beta \\ \frac{v_{max}}{1+k\rho} \sin \alpha \\ \omega_d \end{bmatrix} \quad (18)$$

donde la distancia más corta entre el camino y el UAV es

$$\rho = \sqrt{(x_{ref} - x)^2 + (y_{ref} - y)^2 + (z_{ref} - z)^2}$$

la proyección del movimiento del ángulo en el plano xy es

$$\alpha = \tan^{-1} \left(\frac{\dot{z}_{ref}}{\sqrt{\dot{x}_{ref}^2 + \dot{y}_{ref}^2}} \right)$$

y la proyección de movimiento en el eje Z

$$\beta = \tan^{-1} \left(\frac{\dot{y}_{ref}}{\dot{x}_{ref}} \right)$$

El diseño del controlador cinemático del UAV se basa en el modelo cinemático del Dron. Se propone la siguiente ley de control (V. H. Andaluz et al., 2015):

$$\mathbf{u} = \mathbf{J}^{-1} \left(\dot{\mathbf{h}}_d + \mathbf{K}(\tilde{\mathbf{h}}) \right) \quad (19)$$

donde \mathbf{J}^{-1} contiene los parámetros cinemáticos del UAV, $\dot{\mathbf{h}}_d$ es el vector de velocidades deseadas, $\tilde{\mathbf{h}}$ es el vector de errores de control donde $\tilde{x} = x_{ref} - x$, $\tilde{y} = y_{ref} - y$, $\tilde{z} = z_{ref} - z$, $\tilde{\varphi} = \varphi_{ref} - \varphi$, definido como $\tilde{\mathbf{h}} = \mathbf{h}_{ref} - \mathbf{h}$, los vectores \mathbf{h}_{ref} y \mathbf{h} poseen las pociões deseadas y reales en una tarea otorgada respectivamente, por cada tiempo de muestreo. Para incluir una saturación analítica de las velocidades en el UAV, se propone el uso de la expresión $\mathbf{K}(\tilde{\mathbf{h}}) = \frac{\mathbf{K}_{max}}{1 + |\tilde{\mathbf{h}}|}$, que limita el error en $\tilde{\mathbf{h}}$, donde \mathbf{K}_{max} es una matriz definida positiva de ganancia, para la corrección de errores.

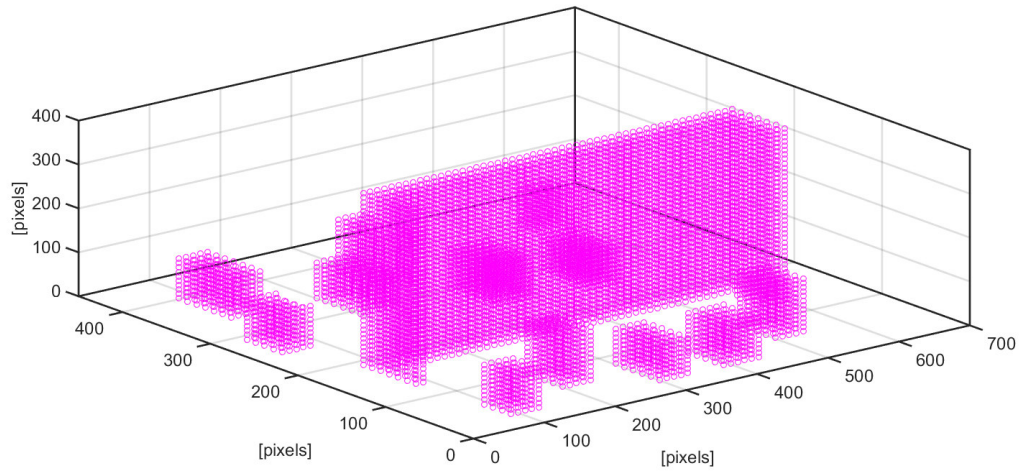
CAPITULO 4

4 RESULTADOS EXPERIMENTALES

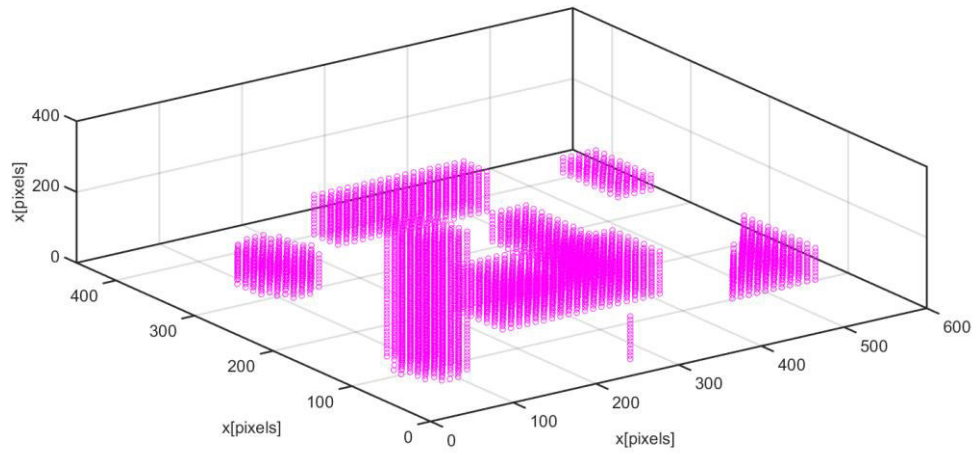
4.1 Prueba del algoritmo RRT 3D y generación de trayectoria por reproducción de probabilidad.

Con la finalidad de evaluar el correcto funcionamiento del algoritmo, se realizó un ambiente propio de simulación que cumple con las características adecuadas para el desarrollo de planificación de trayectorias y así también sea factible implementarlo en un sistema físico con la funcionalidad que posea una navegación libre de obstáculos en tiempo real.

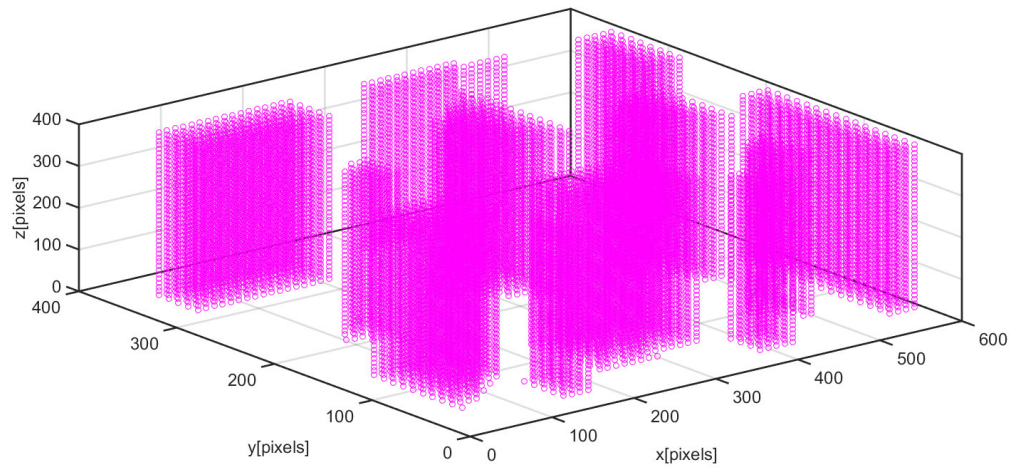
Las pruebas en su totalidad se realizaron en un computador utilizando ambientes virtuales autogenerados a través de un software que maneja matrices matemáticas. La misma que fue reconstruida teniendo en cuenta la funcionalidad del algoritmo en la generación del entorno, así como de la trayectoria de población. Como se explicó en capítulos anteriores los ambientes se basan en una imagen RGB que al ser convertida a una escala de grises nos proporciona el relieve de los entornos que son los obstáculos a evitar, representadas por los gráficos de color magenta para lo cual se tomó cuatro posibles escenarios para realizar las pruebas como indica en la Figura 4.1 a, b, c, d.



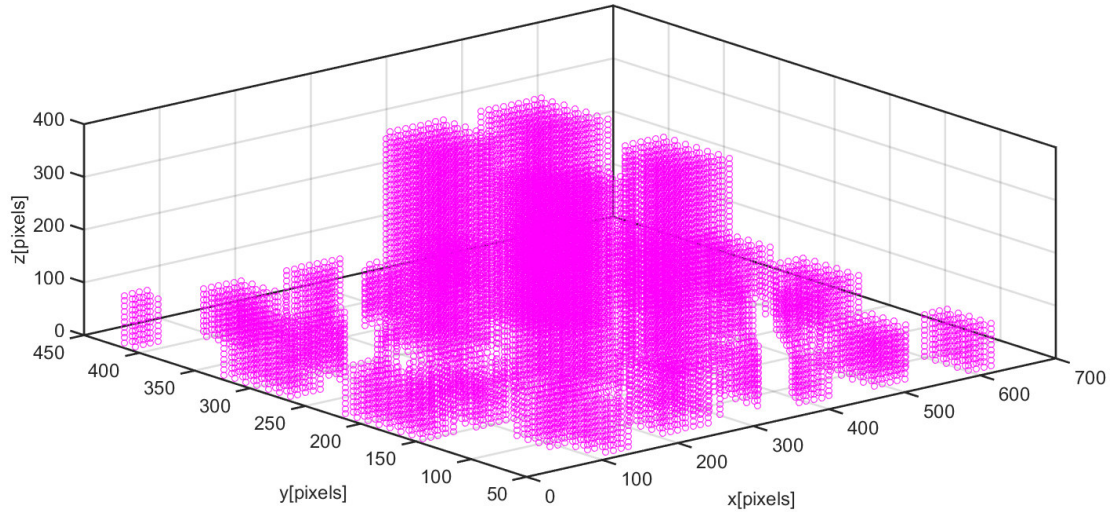
4.1 a) Mapeado de obstáculos en el entorno 1



4.1 b) Mapeado de obstáculos en el entorno 2



4.1 c) Mapeado de obstáculos en el entorno 3



4.1 d) Mapeados obstáculos en el entorno 4

Figura 4.1: Mapeado de materia (Obstáculos)

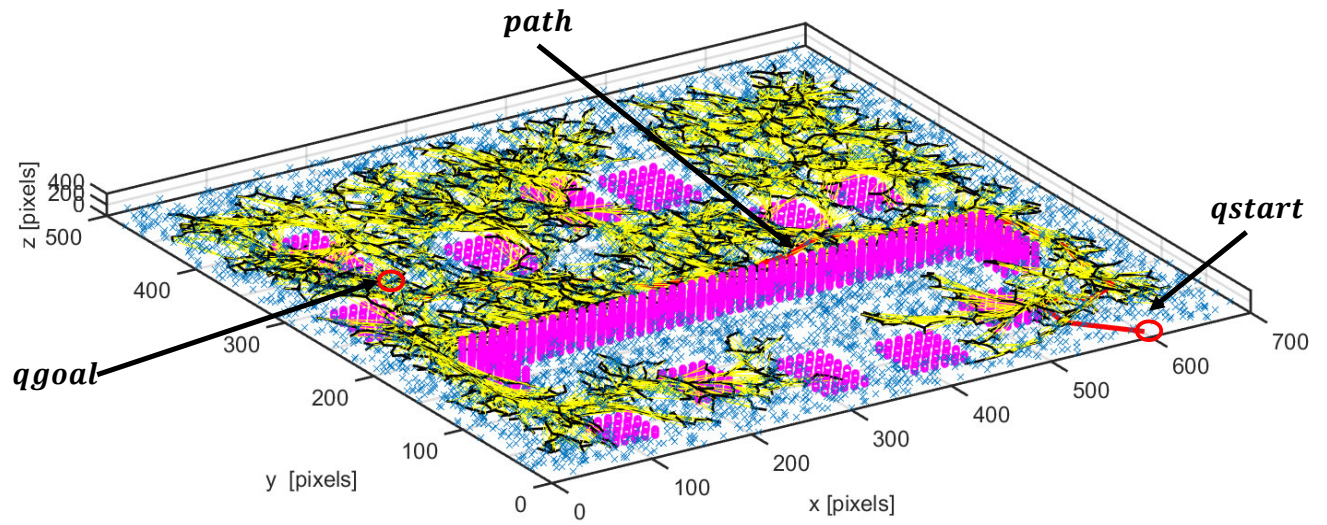
Posteriormente el algoritmo RRT 3D es ejecutado en cada uno de los ambientes creando alrededor de 4500 iteraciones en todo el espacio libre y generando una trayectoria libre de colisiones por reproducción de probabilidad de color rojo como se muestra en las Figura 4.2. a, b, c, d con coordenadas de navegación en el marco inercial $R(x, y, z)$ para cada entorno respectivo:

Entorno 1: qstart (600, 10, 310); qgoal (100, 300, 10)

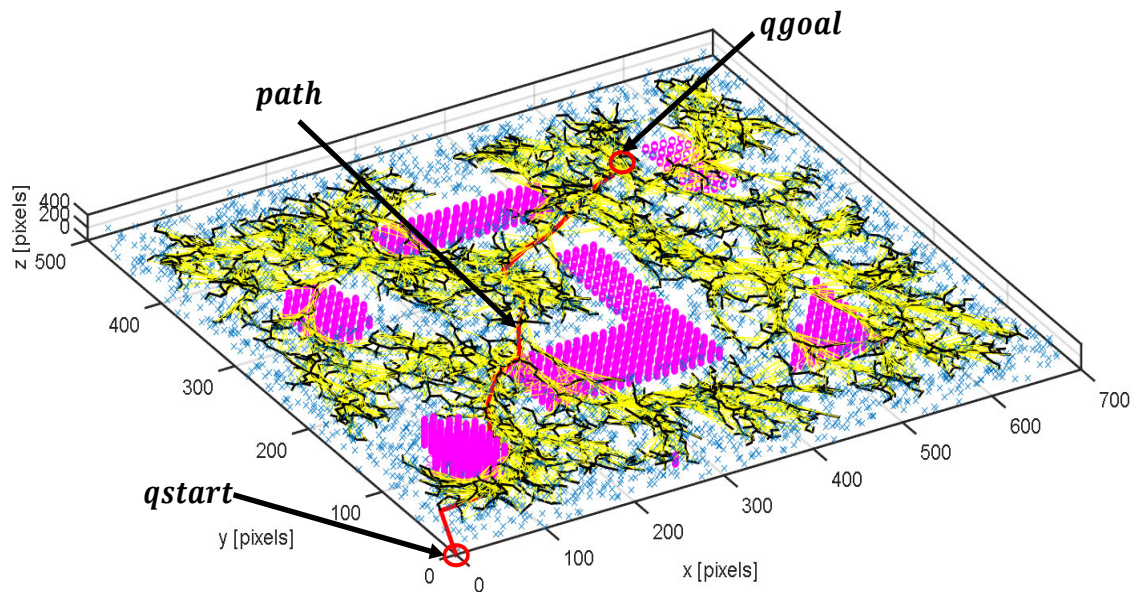
Entorno 2: qstart (100, 400, 10); qgoal (650, 50, 10)

Entorno 3: qstart (550, 10, 10); qgoal (0, 250, 10)

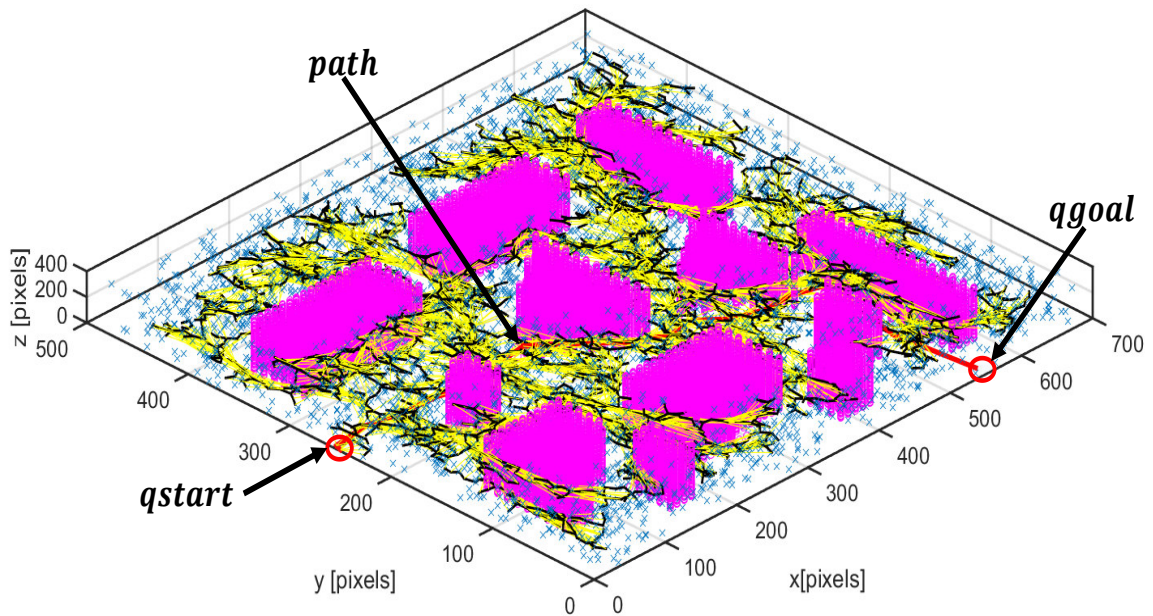
Entorno 4: qstart (80, 400, 10); qgoal (500, 130, 245)



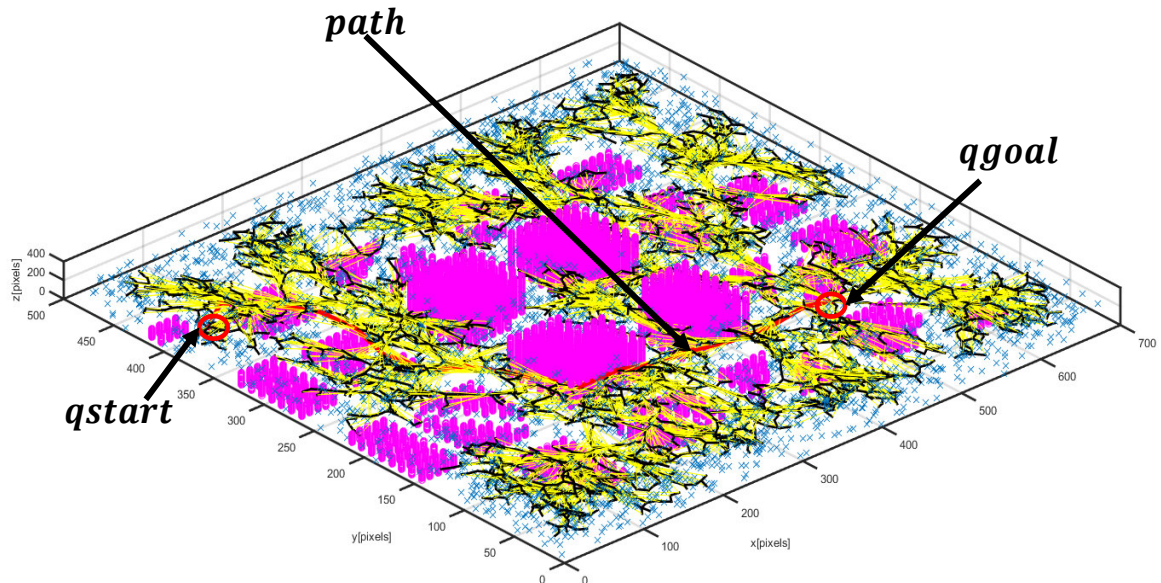
4.2 a) Trayectoria reproducción de probabilidad del entorno 1



4.2 b) Trayectoria reproducción de probabilidad del entorno 2



4.2 c) Trayectoria reproducción de probabilidad del entorno 3

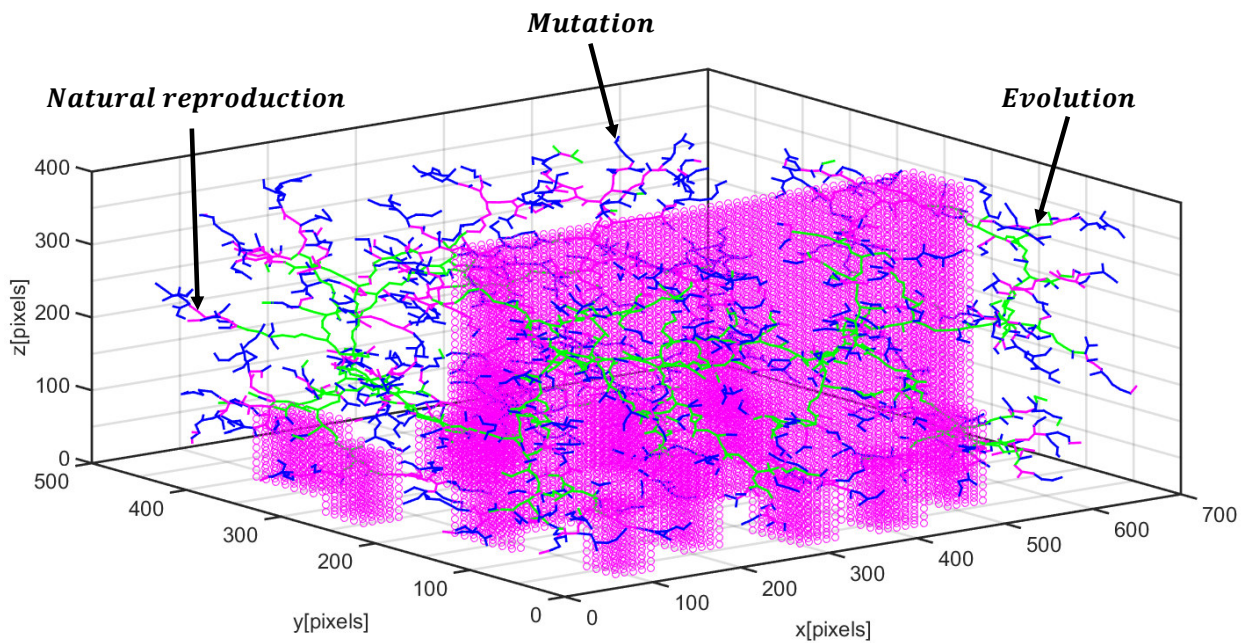


4.2 d) Trayectoria reproducción de probabilidad del entorno 4.

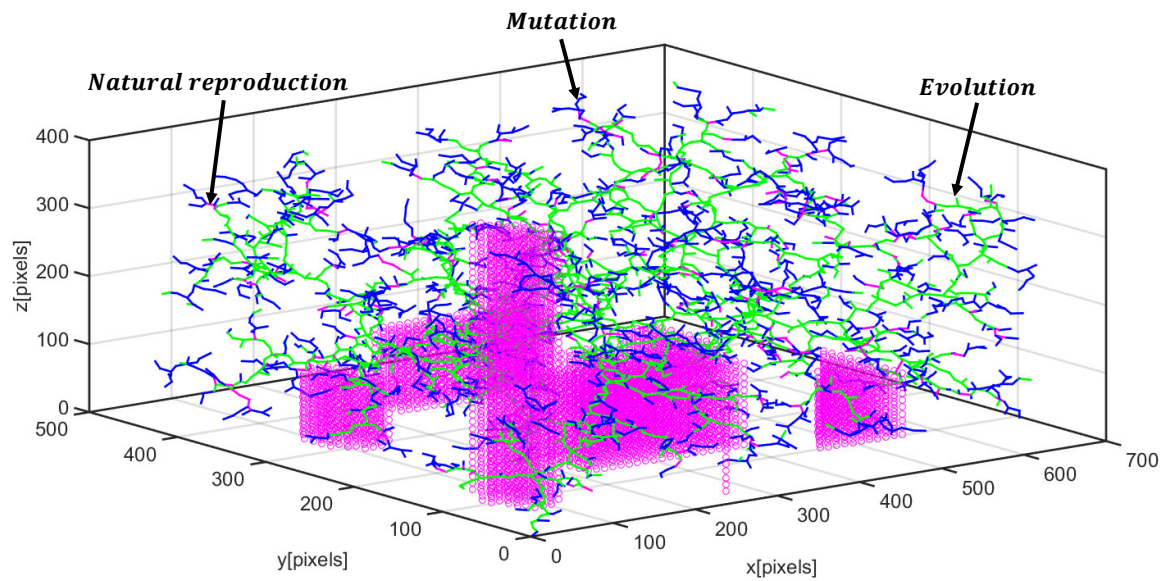
Figura 4.2: Trayectoria reproducción de probabilidad mediante algoritmo RRT 3D

4.2 Técnicas de algoritmos genéticos y Optimización

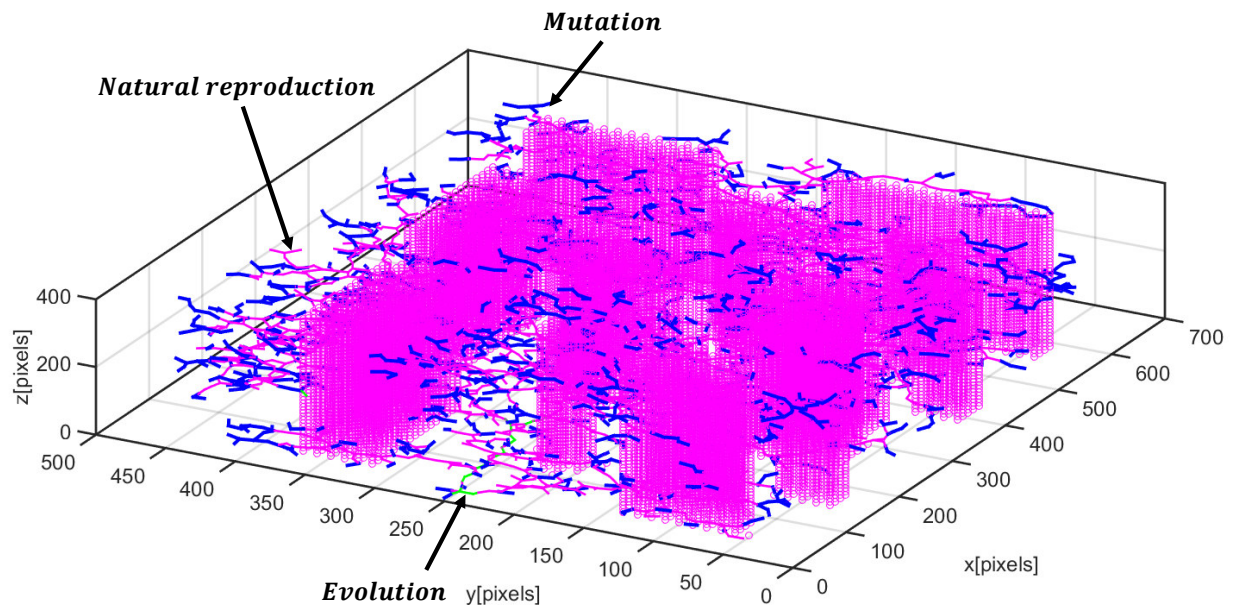
En secciones anteriores se ha expuesto que a partir de generar la trayectoria de reproducción de probabilidad en base a la misma con una población de coordenadas se puede reconstruir las trayectorias libres de colisión utilizando técnicas de algoritmos genéticos como mutación y evolución, las diversas combinaciones aleatorias a partir de la población inicial que genera el algoritmo se muestran recreadas en su respectivo entorno en la Figura 4.3 a, b, c, d.



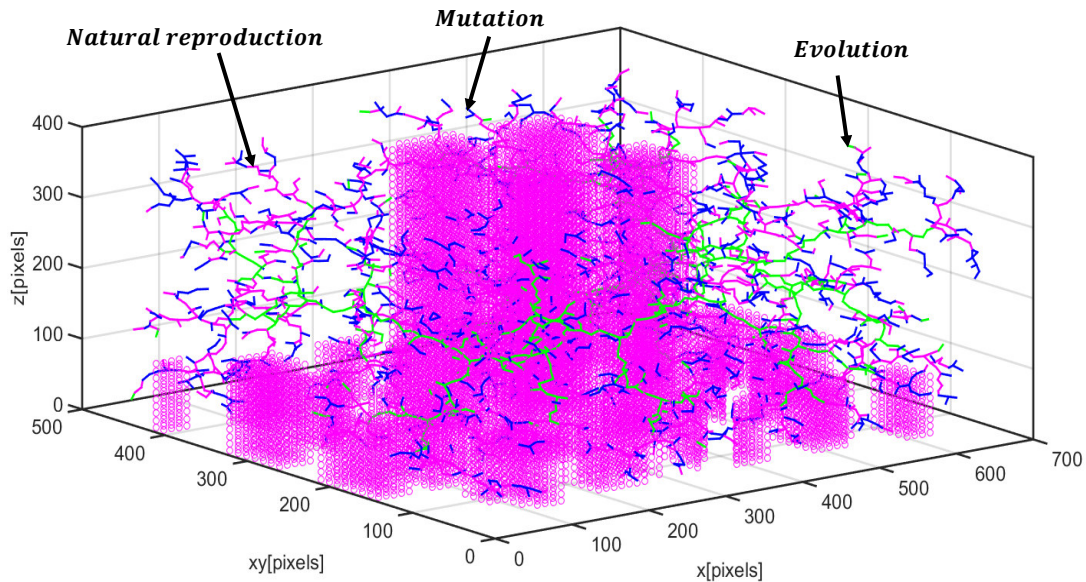
4.3 a) Técnicas de algoritmos genéticos en entorno 1



4.3 b) Técnicas de algoritmos genéticos en entorno 2



4.3 c) Técnicas de algoritmos genéticos en entorno 3



4.3 d) Técnicas de algoritmos genéticos en entorno 4

Figura 4.3: Algoritmo Genético por mutación y evolución de probabilidad.

Los resultados que indica el algoritmo en el entorno 1 son los siguientes: con una evolución de probabilidad de 28.8%, no equivalente a la posición 1 o 2 del vector de respuestas posibles, una mutación de probabilidad de 34.5%, equivalente a la posición 3 del vector de respuestas posibles y por último una reproducción de probabilidad de 36.6%, no equivalente a la posición 5 o 6 del vector de respuestas posibles donde el algoritmo genético evaluó previamente los valores cercanos a la población inicial, esto quiere decir que su trayectoria equivalente (17) de la función probabilidad entregada define tres trayectorias probables como se ilustra en la Figura 4.4.

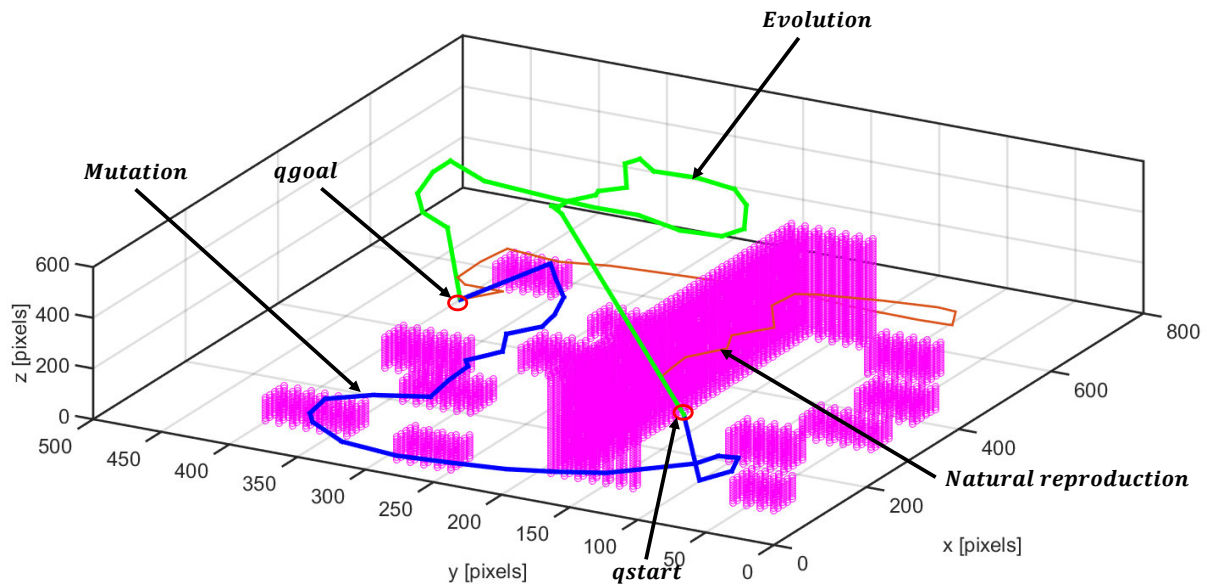


Figura 4.4: Trayectorias probables generadas en el entorno 1.

A continuación (Figura 4.5), se indica una trayectoria de probabilidad 34.5%, equivalente a la posición 3 del vector de probabilidades, donde el cálculo de distancias euclidianas (17) y gasto de tiempo de bucles usados en el controlador dan como resultado una trayectoria final libre de colisiones la cual corresponde a la técnica por mutación en este caso, la misma que es la más óptima para la navegación.

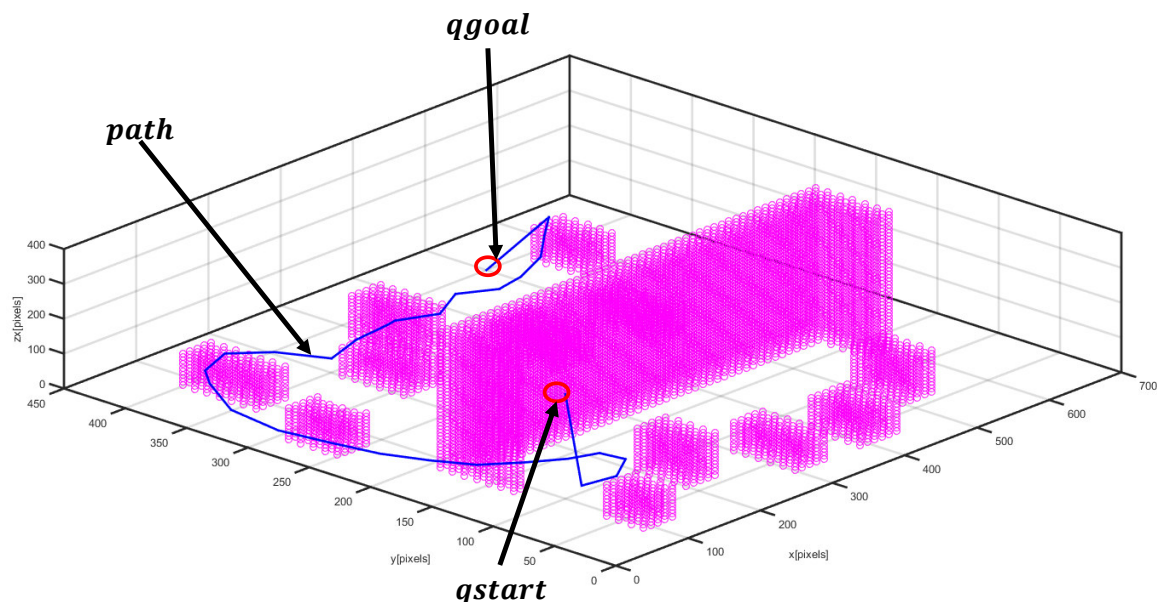


Figura 4.5: Trayectoria óptima para navegación entorno 1.

Los resultados que indica algoritmo en el entorno 2 son los siguientes: con una evolución de probabilidad de 29.6%, equivalente a la posición 2 del vector de respuestas posibles, una mutación de probabilidad de 43.01%, equivalente a la posición 4 del vector de respuestas posibles y por último una reproducción de probabilidad de 75.3%, equivalente a la posición 5 del vector de respuestas posibles donde el algoritmo genético evaluó previamente los valores cercanos a la población inicial, esto quiere decir que su trayectoria equivalente (17) de la función probabilidad entregada define tres trayectorias probables como se ilustra en la Figura 4.6.

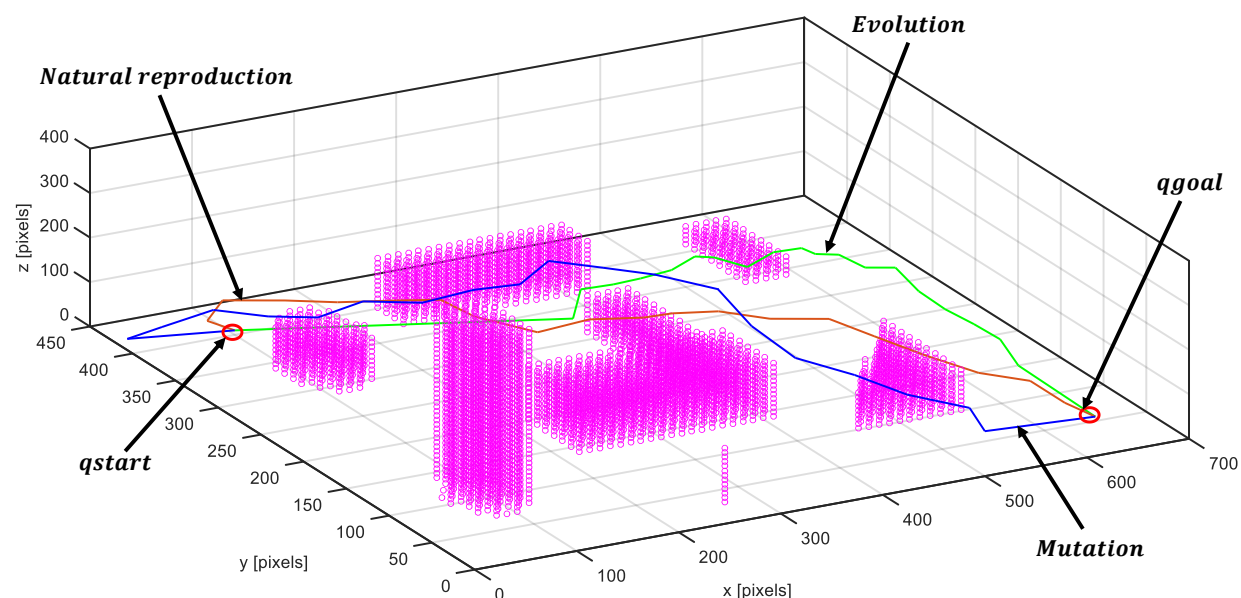


Figura 4.6: Trayectorias probables generadas en el entorno 2.

A continuación (Figura 4.7), se indica una trayectoria de probabilidad 75.3%, equivalente a la posición 5 del vector de probabilidades, donde el cálculo de distancias euclidianas (17) y gasto de tiempo de bucles usados en el controlador dan como resultado una trayectoria final libre de colisiones la cual corresponde a la técnica por reproducción natural en este caso, la misma que es la más óptima para la navegación.

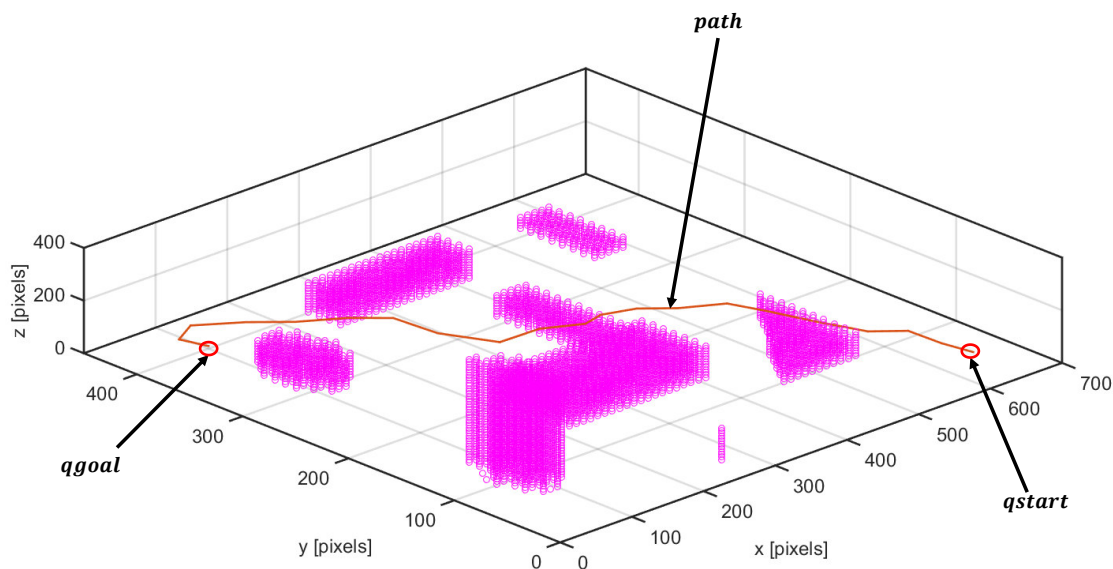


Figura 4.7: Trayectoria óptima para navegación entorno 2

Los resultados que indica algoritmo en el entorno 3 son los siguientes: con una evolución de probabilidad de 30.7%, equivalente a la posición 2 del vector de respuestas posibles, una mutación de probabilidad de 35.5%, equivalente a la posición 3 del vector de respuestas posibles y por último una reproducción de probabilidad de 33.7%, no equivalente a la posición 5 o 6 del vector de respuestas posibles donde el algoritmo genético evaluó previamente los valores cercanos a la población inicial, esto quiere decir que su trayectoria equivalente (17) de la función probabilidad entregada define tres trayectorias probables como se ilustra en la Figura 4.8.

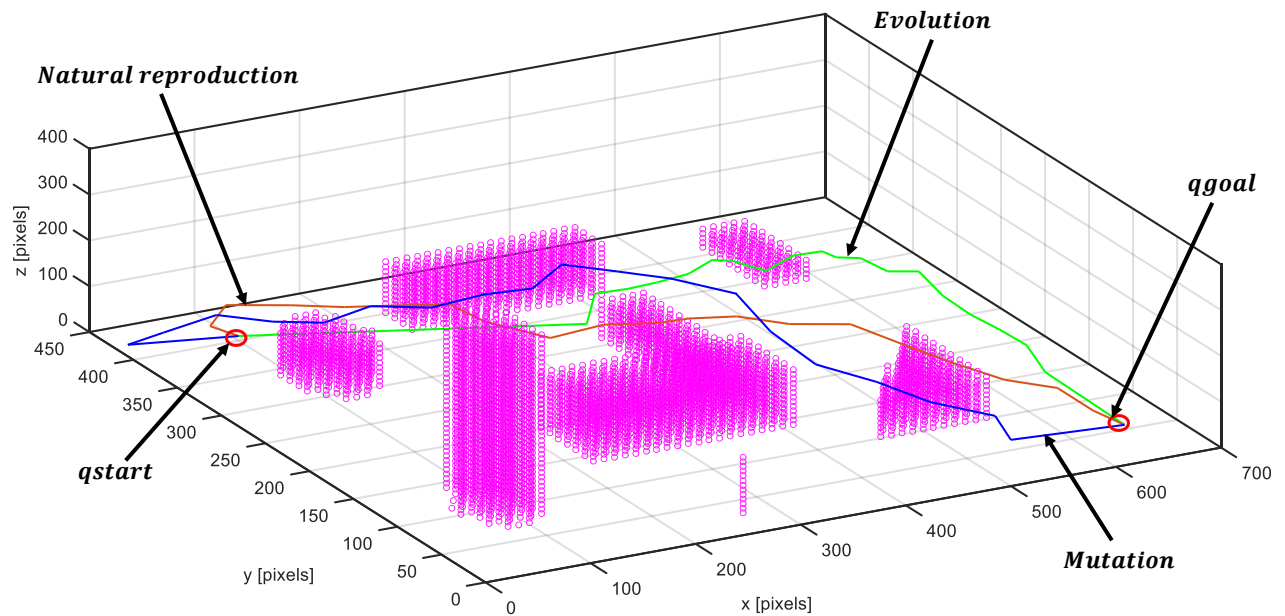


Figura 4.8: Trayectorias probables generadas en el entorno 3.

A continuación (Figura 4.9), se indica una trayectoria de probabilidad 35.5%, equivalente a la posición 3 del vector de probabilidades, donde el cálculo de distancias euclidianas (17) y gasto de tiempo de bucles usados en el controlador dan como resultado una trayectoria final libre de colisiones la cual corresponde a la técnica por mutación en este caso, la misma que es la más óptima para la navegación.

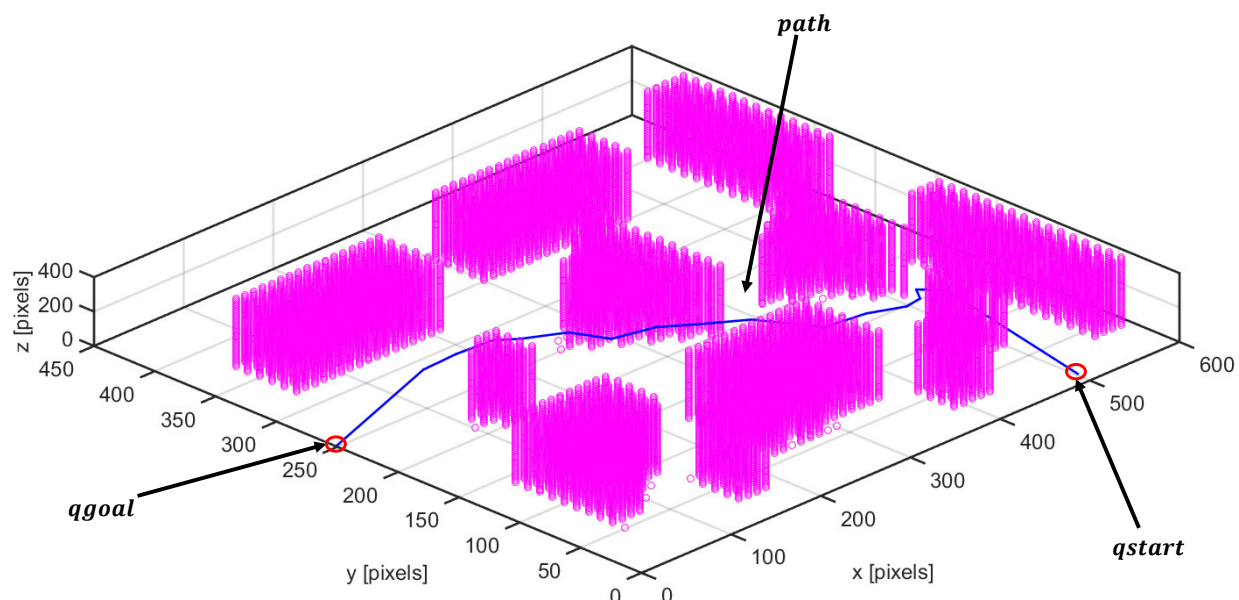


Figura 4.9: Trayectoria óptima para navegación entorno 3.

Los resultados que indica algoritmo en el entorno 4 son los siguientes: con una evolución de probabilidad de 41.3%, no equivalente a la posición 1 o 2 del vector de respuestas posibles, una mutación de probabilidad de 31.1%, no equivalente a la posición 3 o 4 del vector de respuestas posibles y por último una reproducción de probabilidad de 75.6%, equivalente a la posición 5 del vector de respuestas posibles donde el algoritmo genético evaluó previamente los valores cercanos a la población inicial, esto quiere decir que su trayectoria equivalente (17) de la función probabilidad entregada define tres trayectorias probables como se ilustra en la Figura 4.10.

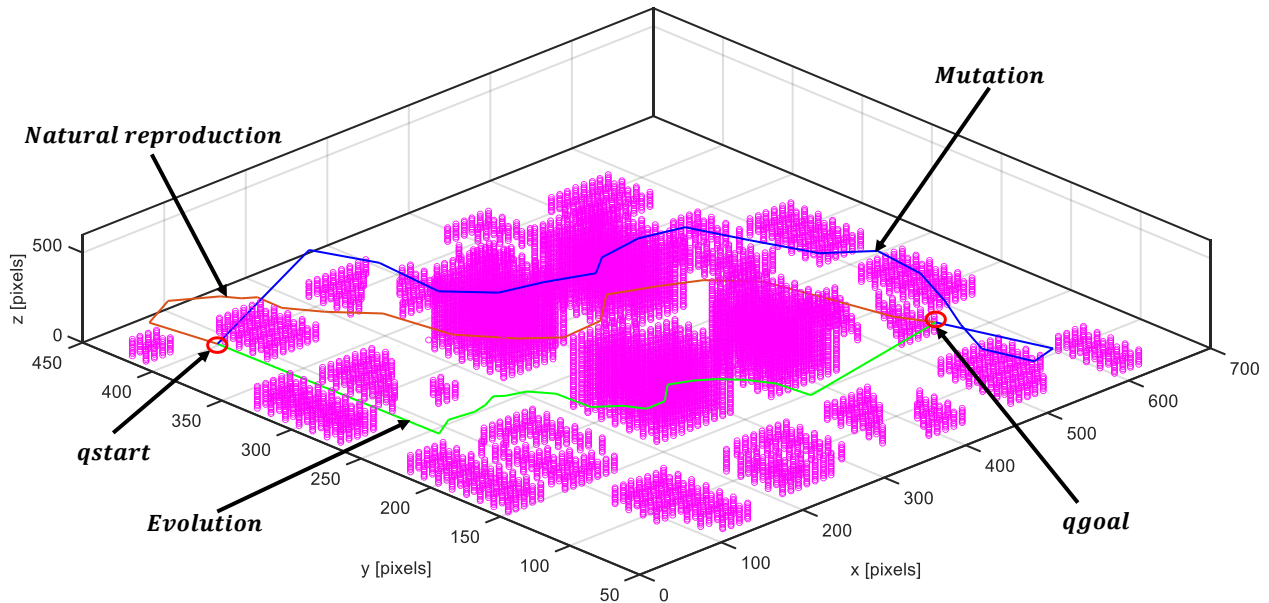


Figura 4.10: Trayectorias probables generadas en el entorno 4.

A continuación (Figura 4.11), se indica una trayectoria de probabilidad 75.6%, equivalente a la posición 5 del vector de probabilidades, donde el cálculo de distancias euclidianas (17) y gasto de tiempo de bucles usados en el controlador dan como resultado una trayectoria final libre de colisiones la cual corresponde a la técnica por evolución en este caso, la misma que es la más óptima para la navegación.

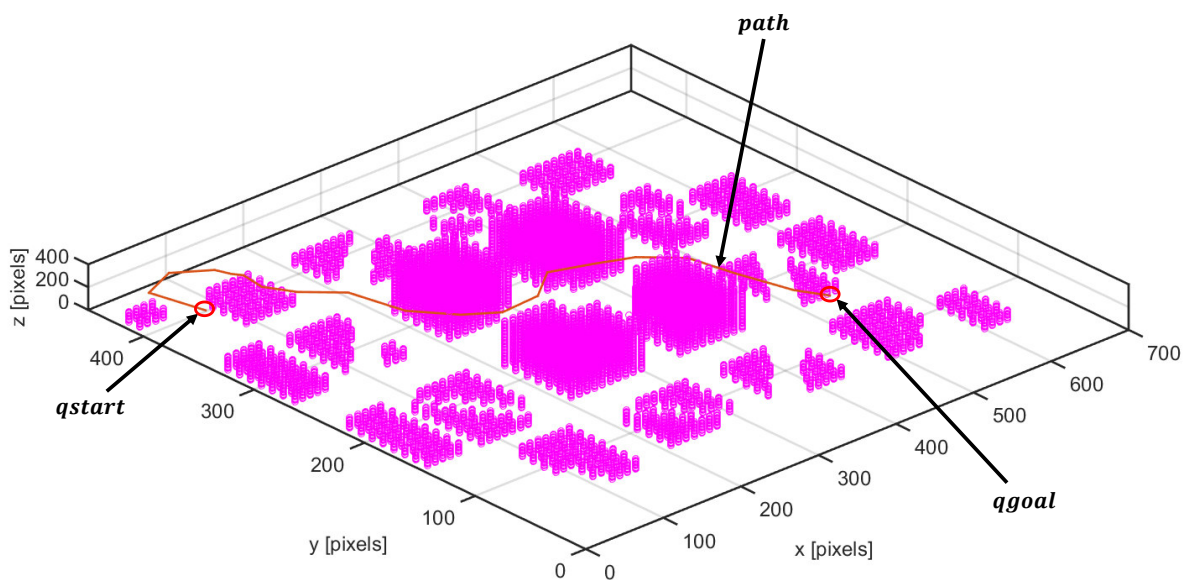
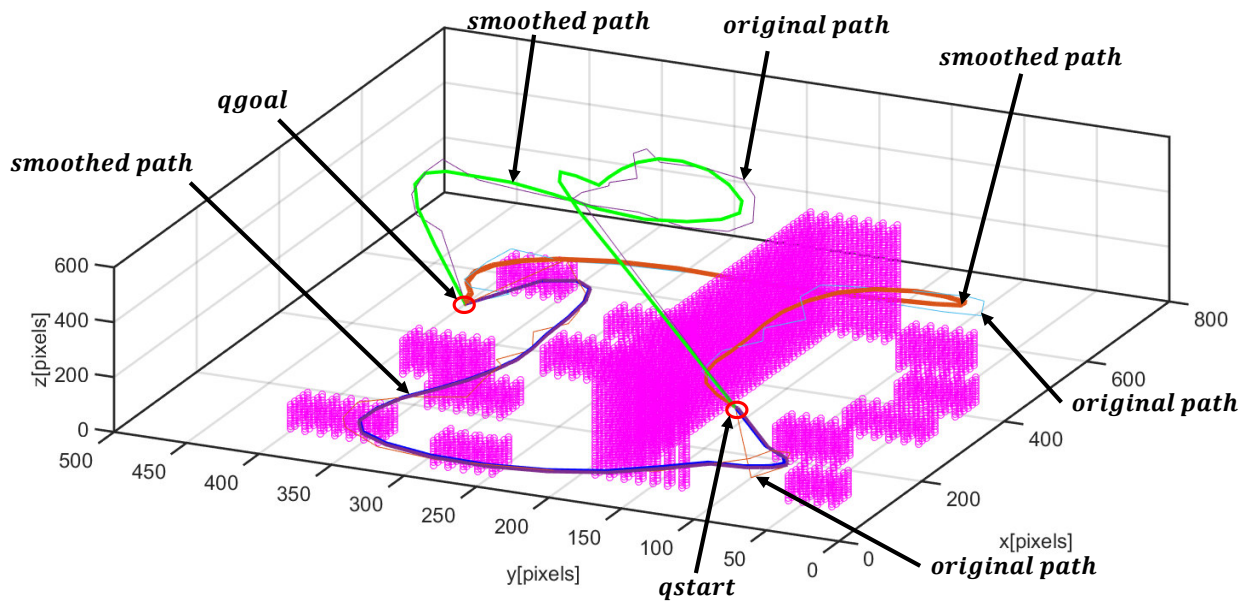


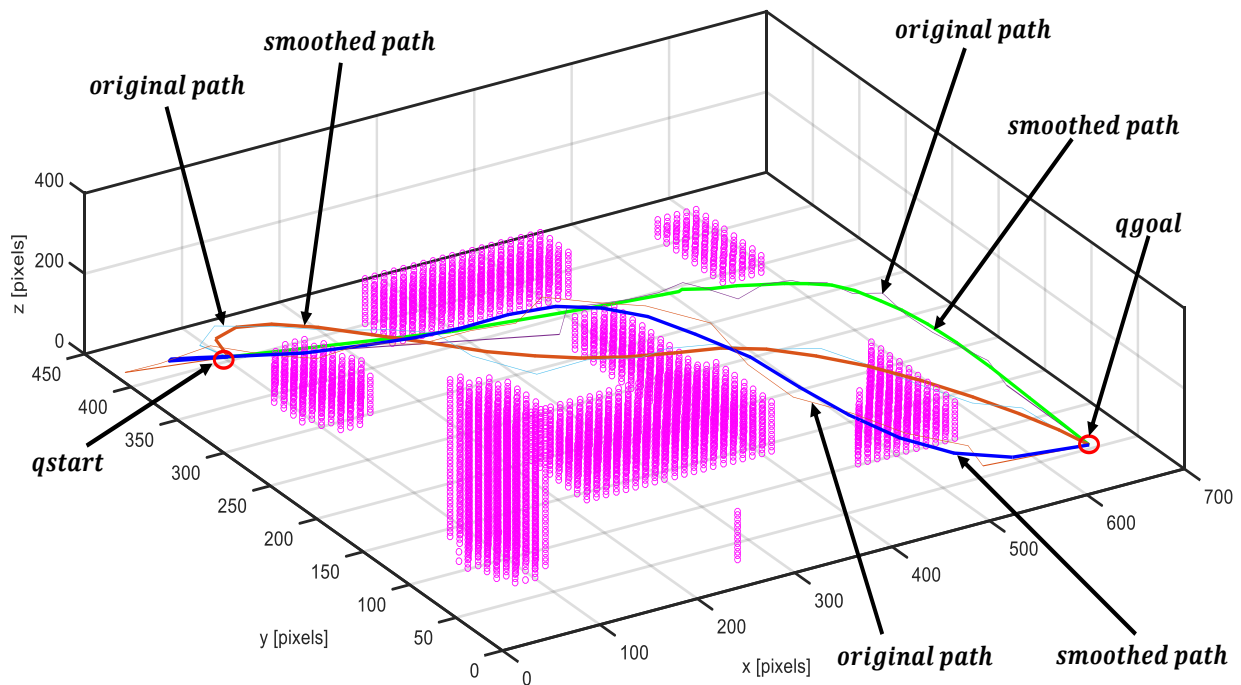
Figura 4.11: Trayectoria óptima para navegación entorno 4

4.3 Pruebas de trayectorias en controlador de caminos

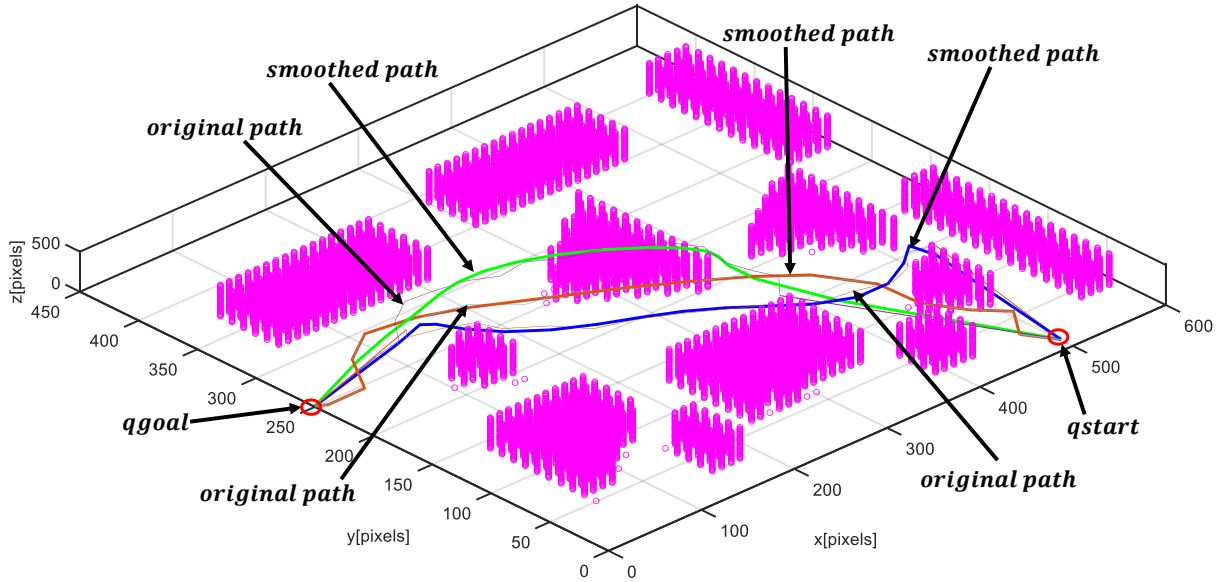
Se utilizó un Filtro de Savitzky–Golay de manera directa en las trayectorias generadas que permite dar un suavizado en los caminos generados en los diferentes entornos como indica la Figura 4.12 a, b, c, d, para mejorar el desplazamiento del robot en el espacio libre debido a que si la ruta original generada posee curvas demasiadas estrechas que formen vértices esto aumentaría los errores de manera considerable e imposibilita la correcta navegación del mismo en la trayectoria a seguir.



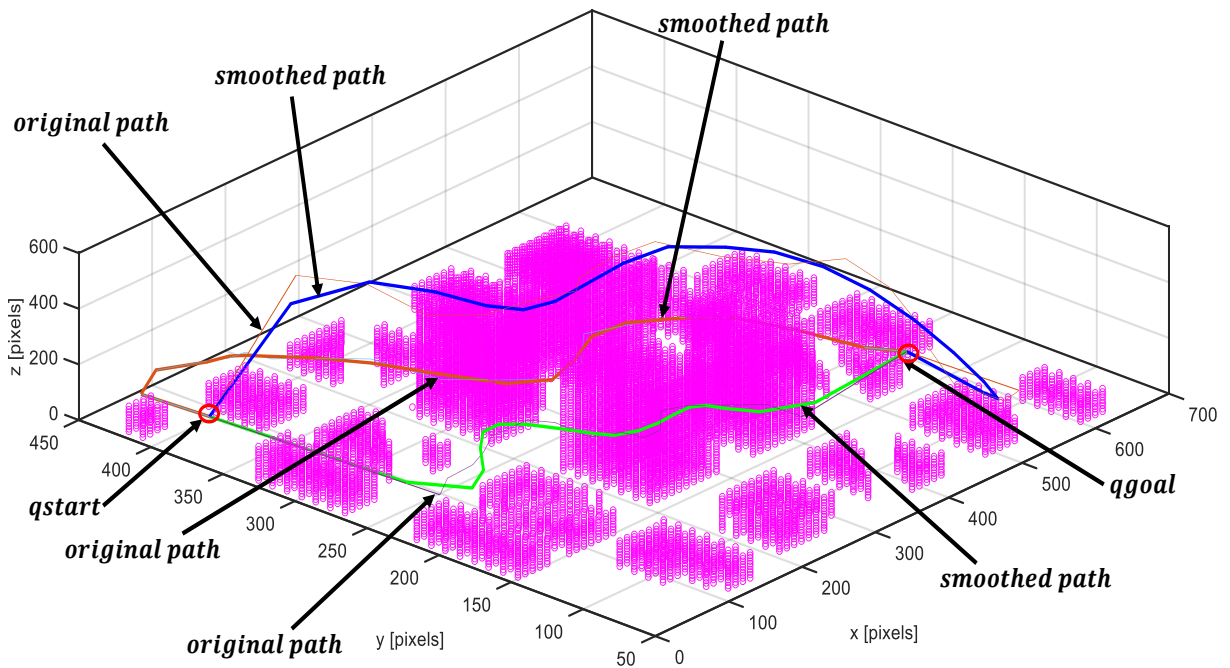
4.12 a) Suavizado de trayectorias entorno 1



4.12 b) Suavizado de trayectorias entorno 2



4.12 c) Suavizado de trayectorias entorno 3



4.12 d) Suavizado de trayectorias entorno 4

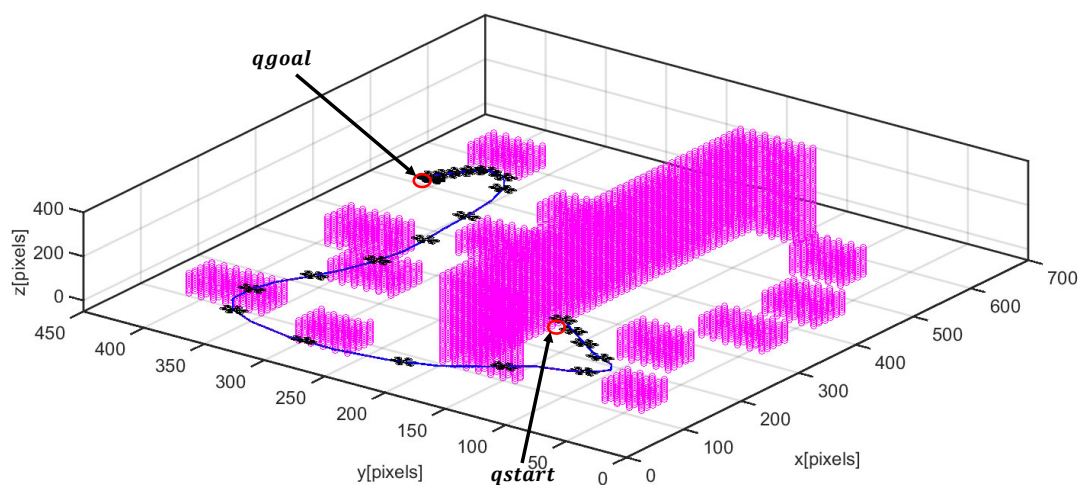
Figura 4.12: Suavizado de Trayectorias generadas.

4.3.1 Evaluación de trayectorias en el controlador

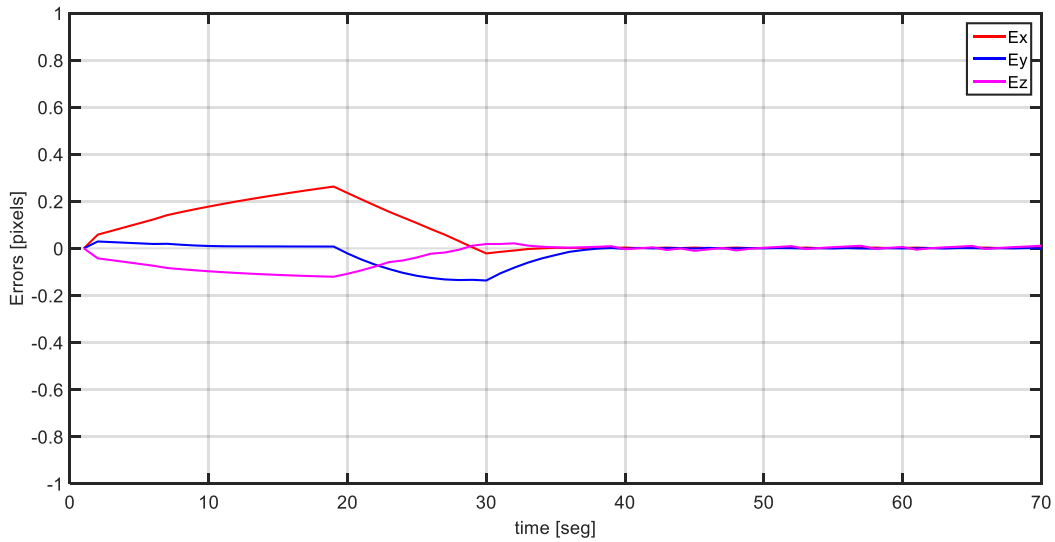
Para el correcto funcionamiento del algoritmo y de que sus rutas son navegables por un UAV se las evalúa en un controlador de caminos expuesto anteriormente donde se indica el tiempo que utiliza el mismo al recorrer toda la trayectoria, así como también los errores en sus coordenadas en el plano inercial para cada entorno respectivo.

Evaluación trayectoria por Mutación Entorno 1

La evaluación de la trayectoria por mutación nos indica un tiempo utilizado por el controlador es de $t=48$ [seg] y la distancia entre la posición desde el punto de interés del UAV y el punto más cercano de la ruta ilustrado en la Figura 4.13 a, b.



4.13 a) Test de navegación del UAV en trayectoria por mutación entorno 1

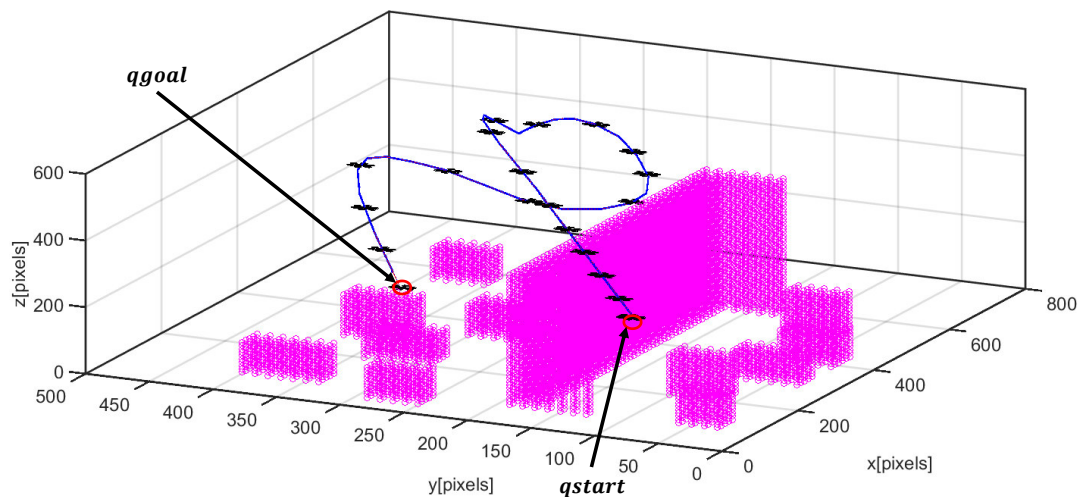


4.13 b) Errores de navegación

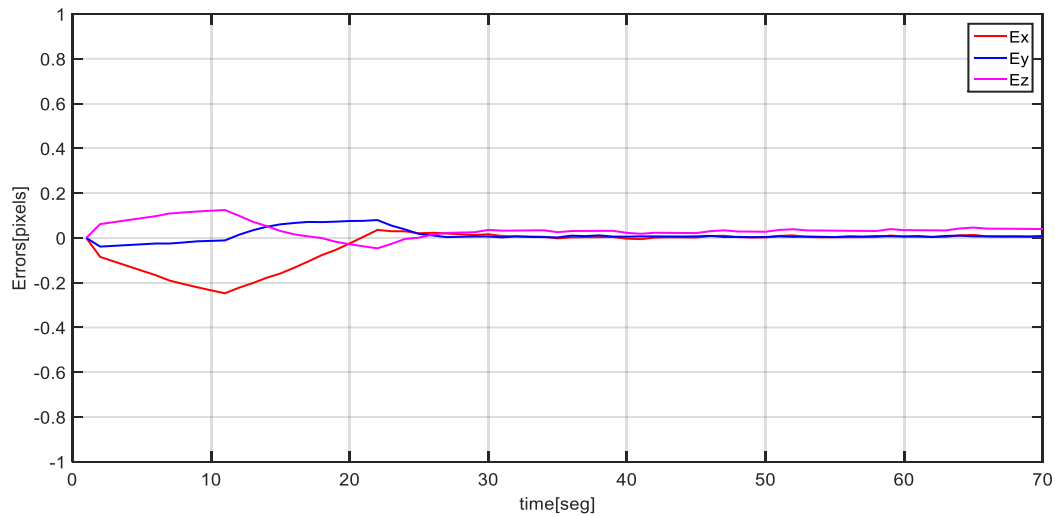
Figura 4.13: Resultados evaluación de Trayectoria por mutación entorno 1

Evaluación de trayectoria por Evolución entorno 1

La evaluación de la trayectoria por evolución nos indica un tiempo utilizado por el controlador es de $t=65$ [seg] y la distancia entre la posición desde el punto de interés del UAV y el punto más cercano de la ruta ilustrado en la Figura 4.14 a, b.



4.14 a) Test de navegación del UAV en trayectoria por evolución entorno 1

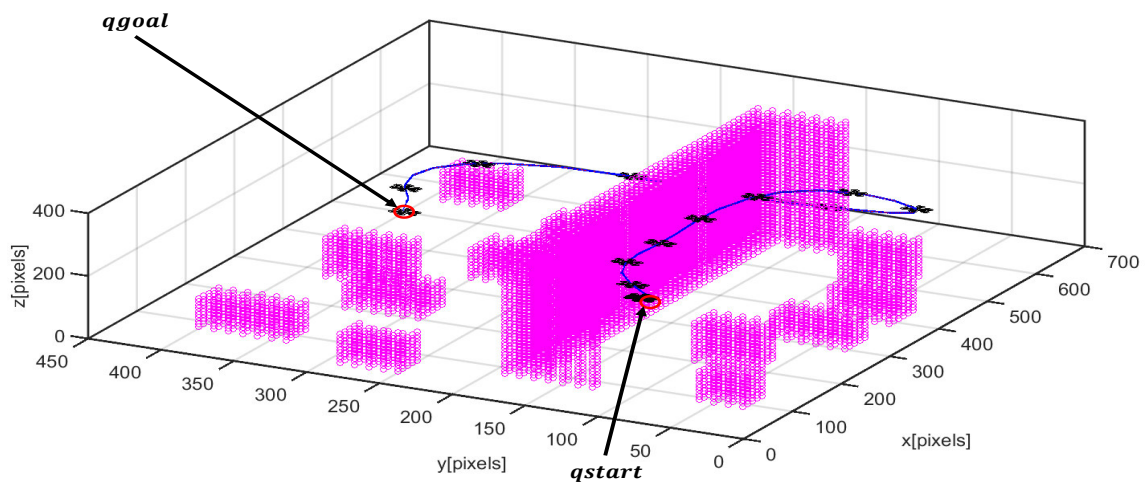


4.14 b) Errores de navegación

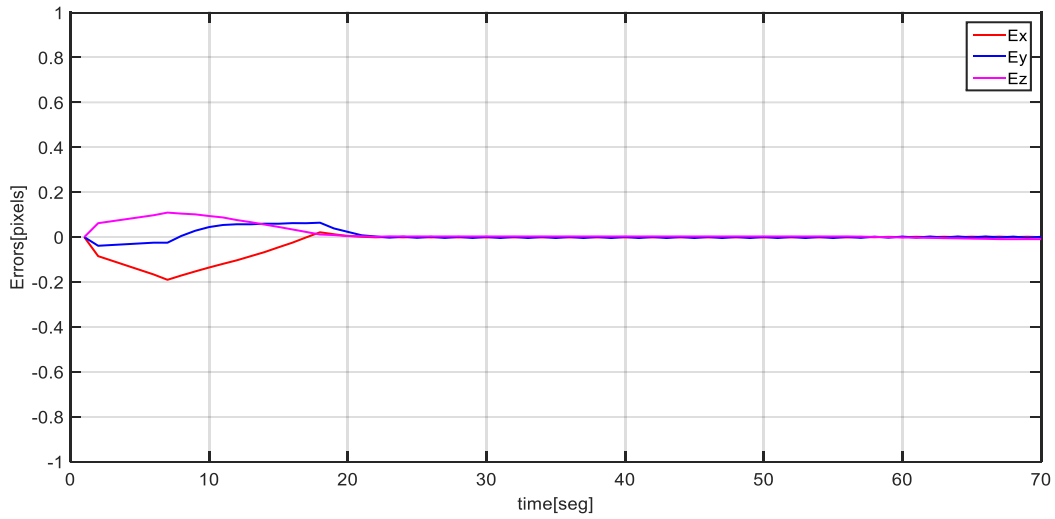
Figura 4.14: Resultados evaluación de Trayectoria por evolución entorno 1

Evaluación de trayectoria por Reproducción Natural entorno 1

La evaluación de la trayectoria por reproducción natural nos indica un tiempo utilizado por el controlador es de $t=53$ [seg] y la distancia entre la posición desde el punto de interés del UAV y el punto más cercano de la ruta ilustrado en la Figura 4.15 a, b.



4.15 a) Test de navegación del UAV en trayectoria por reproducción natural entorno 1

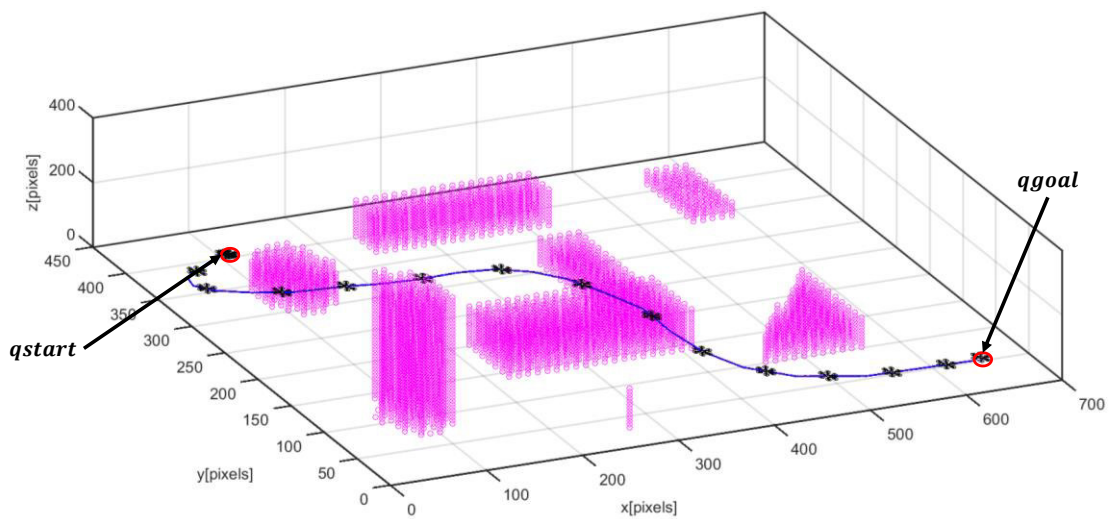


4.15 b) Errores de navegación

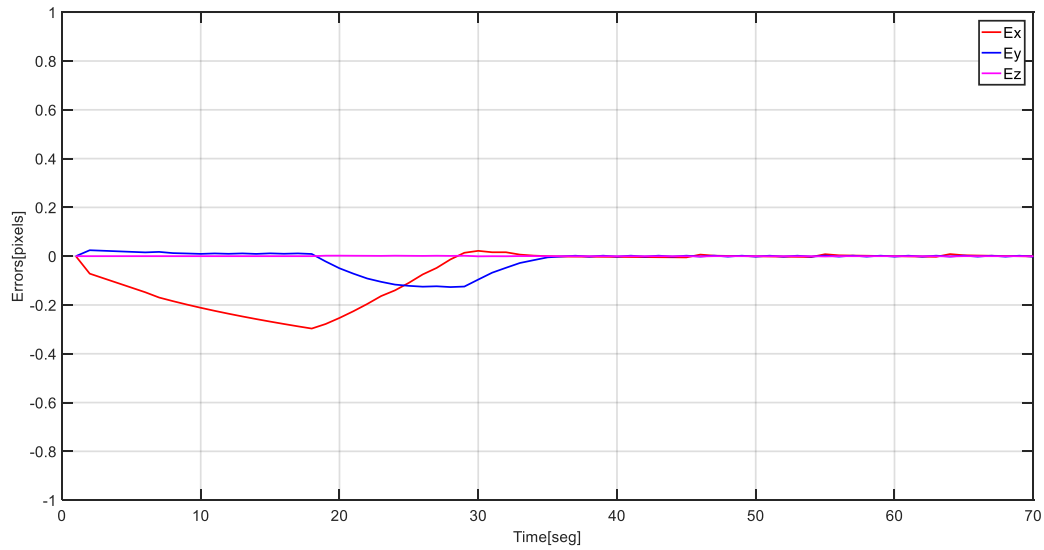
Figura 4.15: Resultados evaluación de Trayectoria por reproducción natural entorno 1

Evaluación trayectoria por Mutación entorno 2

La evaluación de la trayectoria por mutación nos indica un tiempo utilizado por el controlador es de $t=62$ [seg] y la distancia entre la posición desde el punto de interés del UAV y el punto más cercano de la ruta ilustrado en la Figura 4.16 a, b.



4.16 a) Test de navegación del UAV en trayectoria por mutación entorno 2

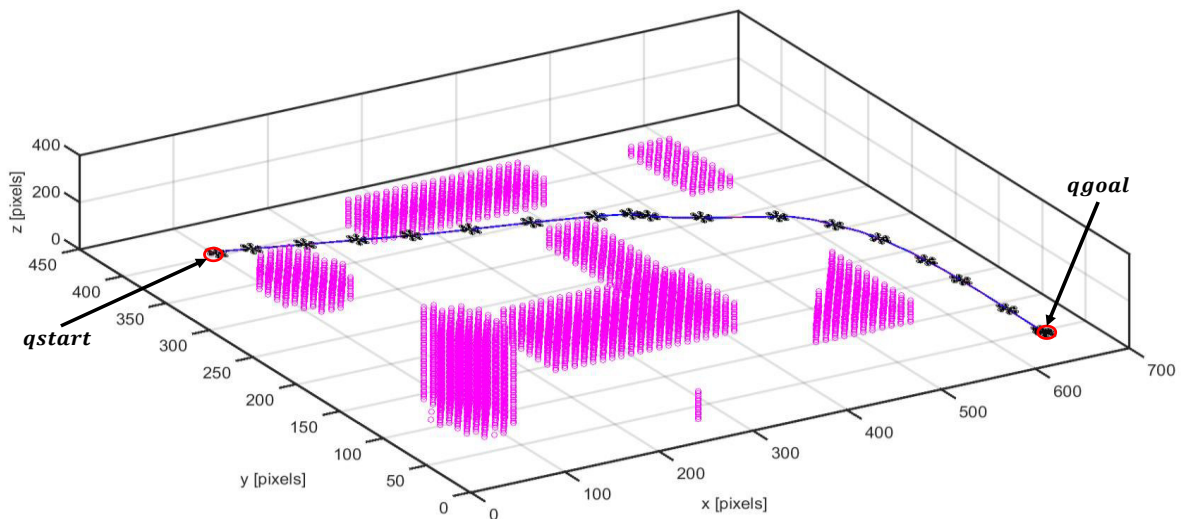


4.16 b) Errores de navegación

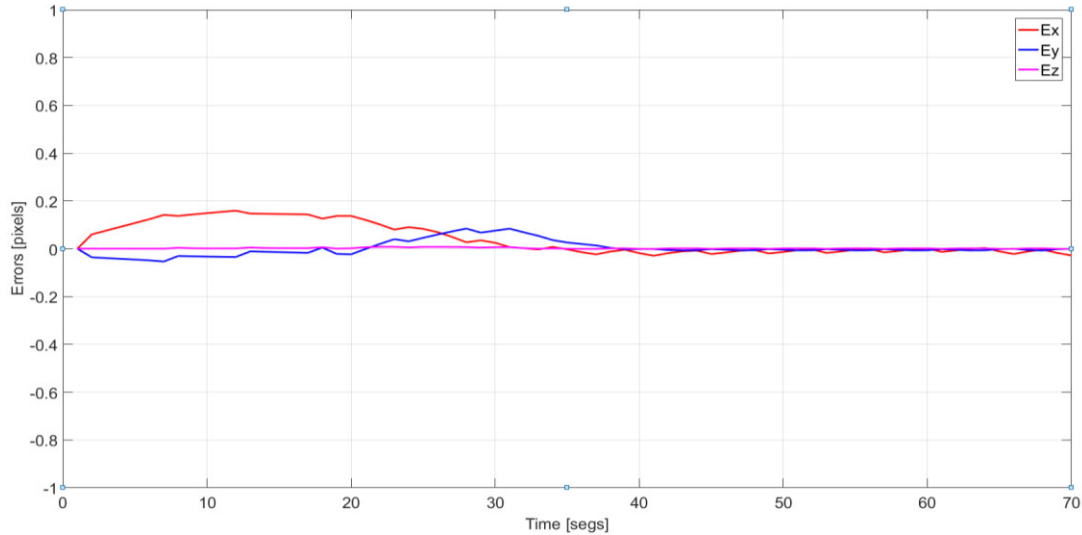
Figura 4.16: Resultados evaluación de Trayectoria por mutación entorno 2

Evaluación de trayectoria por Evolución entorno 2

La evaluación de la trayectoria por evolución nos indica un tiempo utilizado por el controlador es de $t=58$ [seg] y la distancia entre la posición desde el punto de interés del UAV y el punto más cercano de la ruta ilustrado en la Figura 4.17 a, b.



4.17 a) Test de navegación del UAV en trayectoria por evolución entorno 2

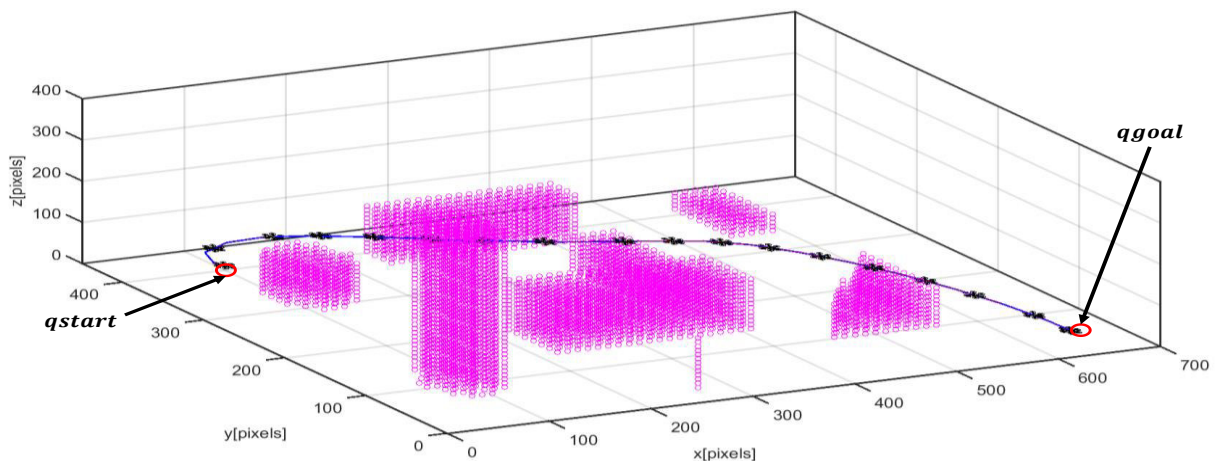


4.17 b) Errores de navegación

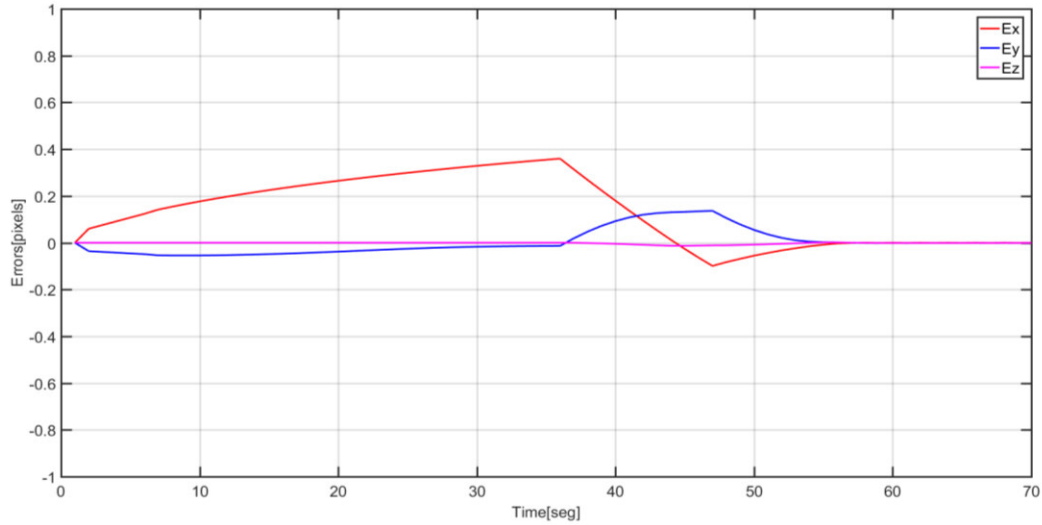
Figura 4.17: Resultados evaluación de Trayectoria por evolución entorno 2

Evaluación de trayectoria por Reproducción Natural entorno 2

La evaluación de la trayectoria por reproducción natural nos indica un tiempo utilizado por el controlador es de $t=53$ [seg] y la distancia entre la posición desde el punto de interés del UAV y el punto más cercano de la ruta ilustrado en la Figura 4.18 a, b.



4.18 a) Test de navegación del UAV en trayectoria por reproducción natural entorno 2

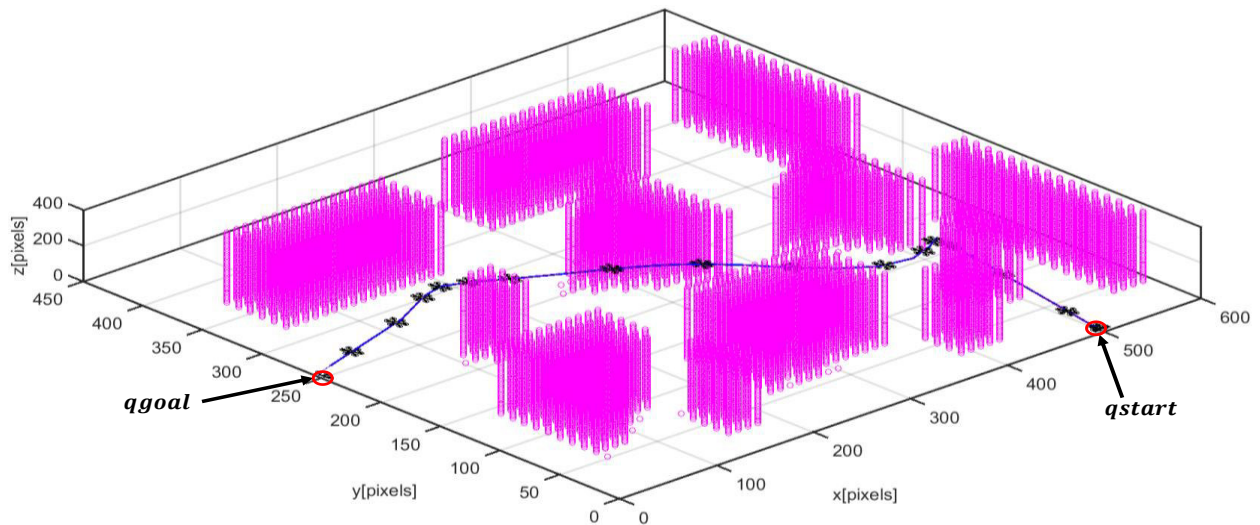


4.18 b) Errores de navegación

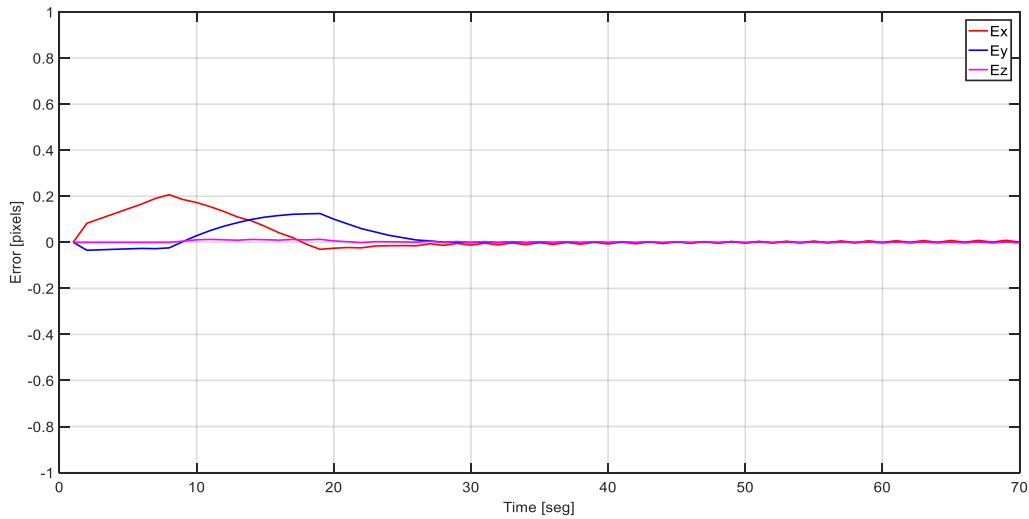
Figura 4.18: Resultados evaluación de Trayectoria por reproducción natural entorno 2

Evaluación trayectoria por Mutación entorno 3

La evaluación de la trayectoria por mutación nos indica un tiempo utilizado por el controlador es de $t=45$ [seg] y la distancia entre la posición desde el punto de interés del UAV y el punto más cercano de la ruta ilustrado en la Figura 4.19 a, b.



4.19 a) Test de navegación del UAV en trayectoria por mutación entorno 3

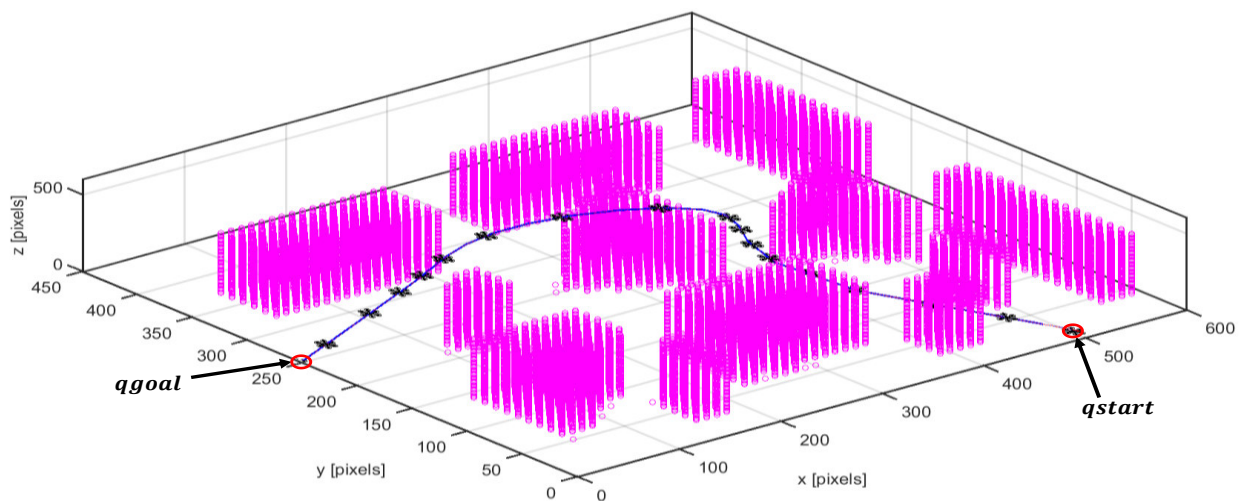


4.19 b) Errores de navegación

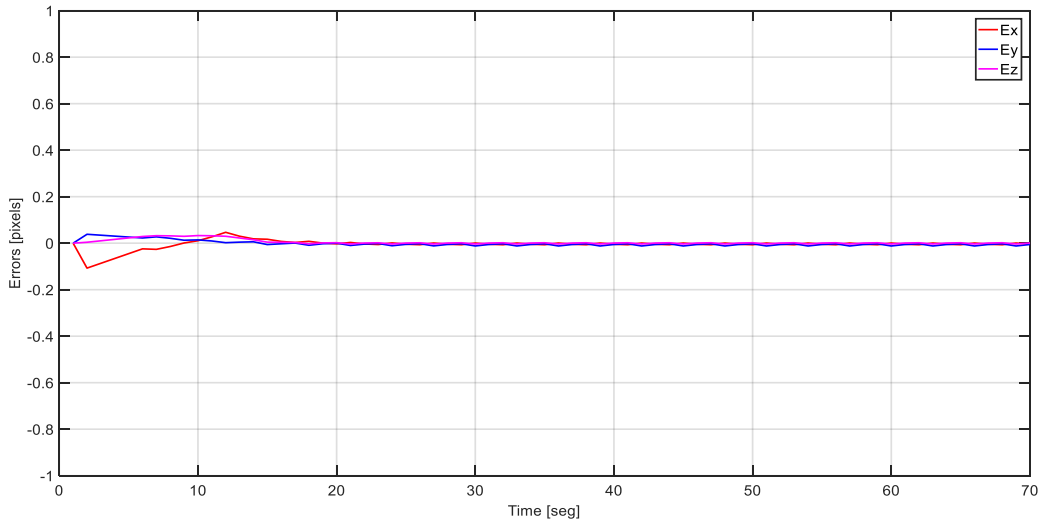
Figura 4.19: Resultados evaluación de Trayectoria por mutación entorno 3

Evaluación de trayectoria por Evolución entorno 3

La evaluación de la trayectoria por evolución nos indica un tiempo utilizado por el controlador es de $t=55$ [seg] y la distancia entre la posición desde el punto de interés del UAV y el punto más cercano de la ruta ilustrado en la Figura 4.20 a, b.



4.20 a) Test de navegación del UAV en trayectoria por evolución entorno 3

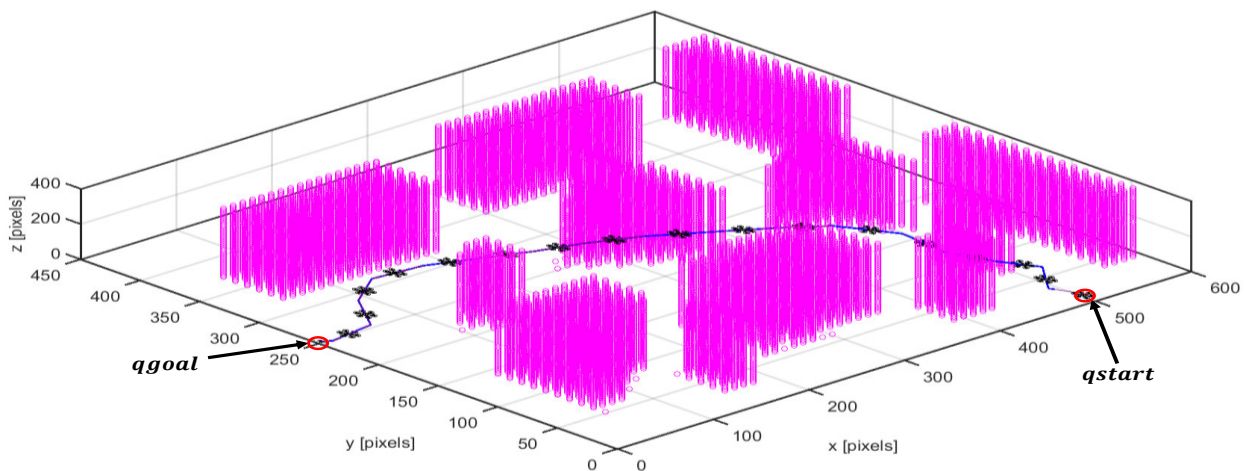


4.20 b) Errores de navegación

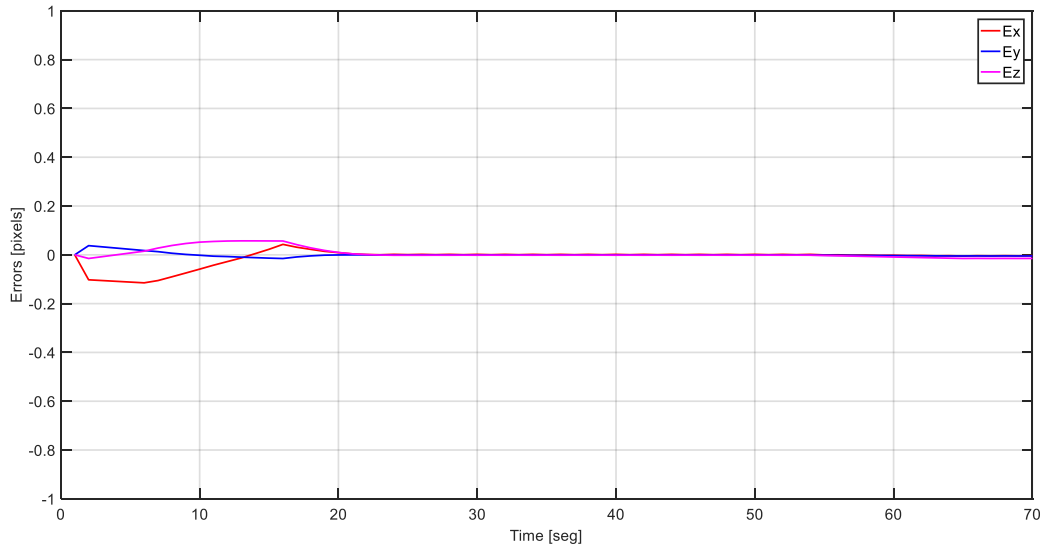
Figura 4.20: Resultados evaluación de Trayectoria por evolución entorno 3

Evaluación de trayectoria por Reproducción Natural entorno 3

La evaluación de la trayectoria por reproducción natural nos indica un tiempo utilizado por el controlador es de $t=49$ [seg] y la distancia entre la posición desde el punto de interés del UAV y el punto más cercano de la ruta ilustrado en la Figura 4.21 a, b.



4.21 a) Test de navegación del UAV en trayectoria por reproducción natural entorno 3

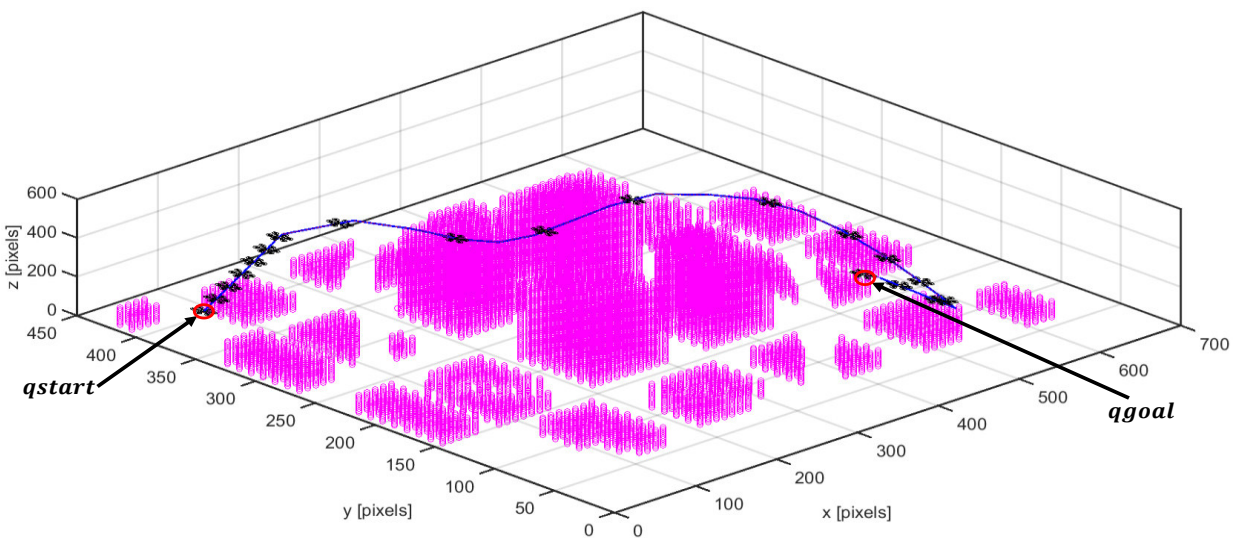


4.21 b) Errores de navegación

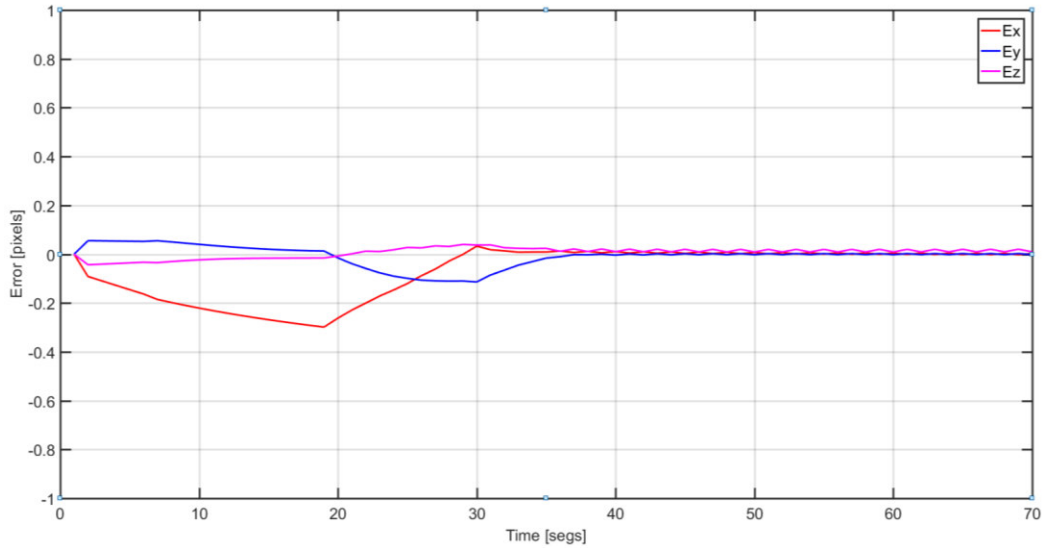
Figura 4.21: Resultados evaluación de Trayectoria por reproducción natural entorno 3

Evaluación trayectoria por Mutación entorno 4

La evaluación de la trayectoria por mutación nos indica un tiempo utilizado por el controlador es de $t=77$ [seg] y la distancia entre la posición desde el punto de interés del UAV y el punto más cercano de la ruta ilustrado en la Figura 4.22 a, b.



4.22 a) Test de navegación del UAV en trayectoria por mutación entorno 4

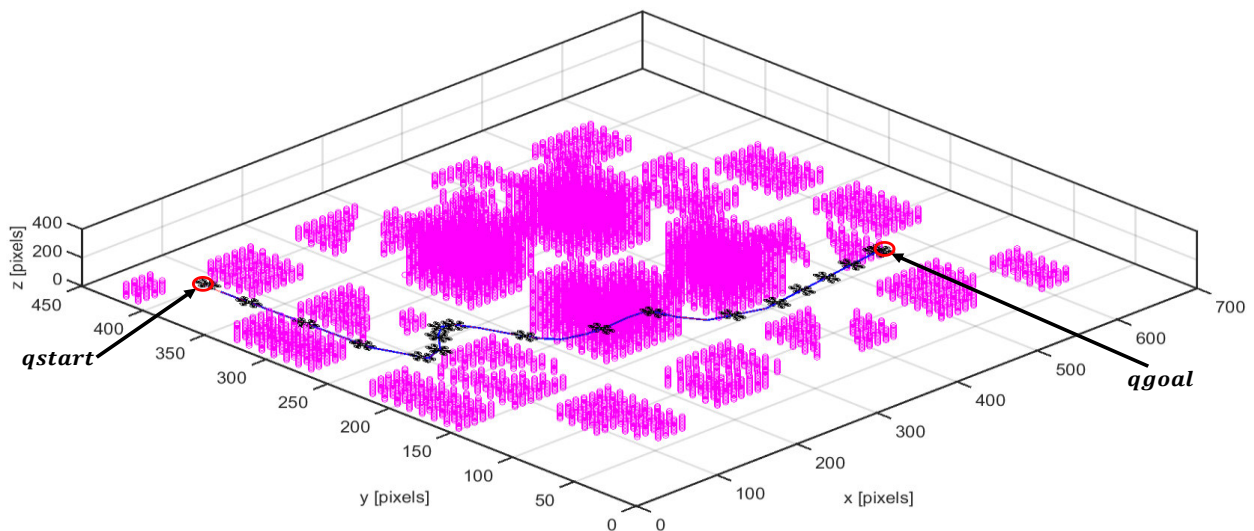


4.22 b) Errores de navegación

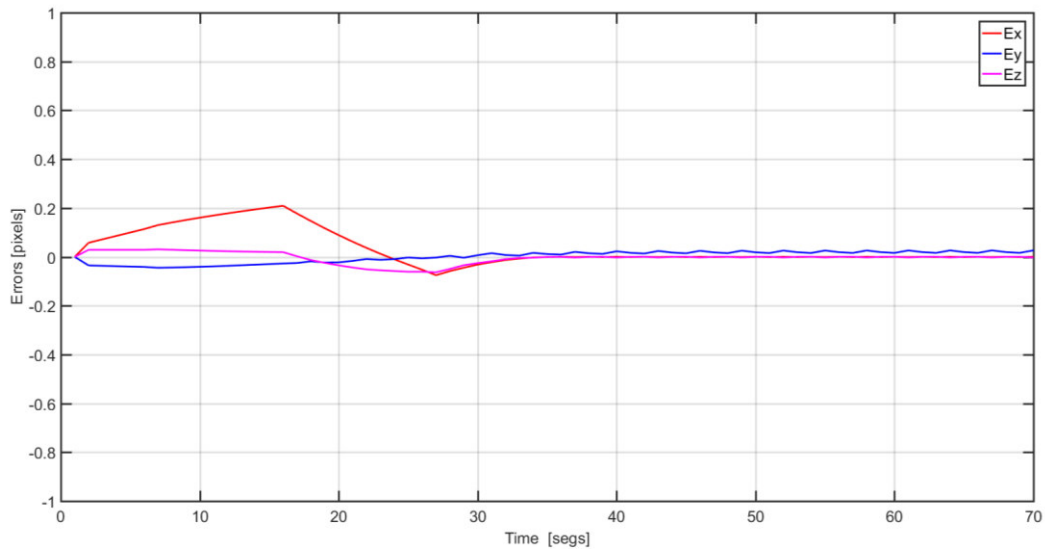
Figura 4.22: Resultados evaluación de Trayectoria por mutación entorno 4

Evaluación de trayectoria por Evolución entorno 4

La evaluación de la trayectoria por evolución nos indica un tiempo utilizado por el controlador es de $t=68$ [seg] y la distancia entre la posición desde el punto de interés del UAV y el punto más cercano de la ruta ilustrado en la Figura 4.23 a, b.



4.23 a) Test de navegación del UAV en trayectoria por evolución entorno 4

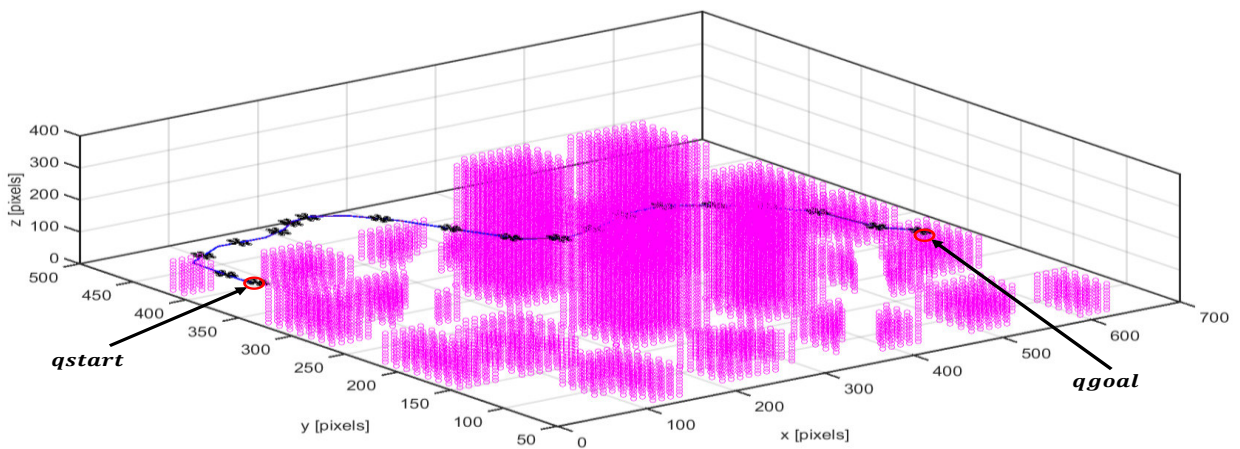


4.23 b) Errores de navegación

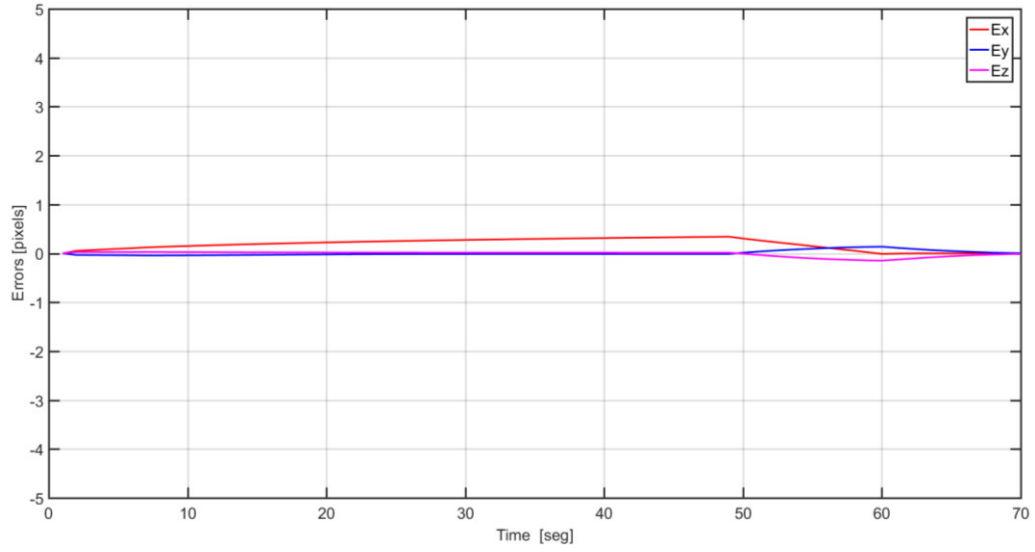
Figura 4.23: Resultados evaluación de Trayectoria por evolución entorno 4

Evaluación de trayectoria por Reproducción Natural entorno 4

La evaluación de la trayectoria por reproducción natural nos indica un tiempo utilizado por el controlador es de $t=66$ [seg] y la distancia entre la posición desde el punto de interés del UAV y el punto más cercano de la ruta ilustrado en la Figura 4.24 a, b.



4.24 a) Test de navegación del UAV en trayectoria por reproducción natural entorno 4



4.24 b) Errores de navegación

Figura 4.24: Resultados evaluación de Trayectoria por reproducción natural entorno 4

CAPITULO 5

5 CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Los algoritmos de planificación de caminos (Path Planning) son una herramienta funcional que su primer objetivo es generar caminos seguros libres de colisiones en la navegación autónoma de un robot, tomando en cuenta las dimensiones del entorno y de los obstáculos presentes en el mismo. Además permiten la integración de cualquier tipo de configuración de robot que tenga capacidad de locomoción autónoma, como pueden ser robots terrestres o aéreos, en nuestro caso un robot UAV (Vehículo Aéreo no Tripulado).

El uso del algoritmo de planificación RRT 3D, al momento de realizar cada ejecución y al ser un procedimiento de selección randomico, todos los posibles caminos a obtener en cada entorno de simulación son diferentes cada uno. Esto ayuda de forma significativa a obtener una población de probabilidades aleatoria por las cuales con la implementación de las técnicas de algoritmos genéticos se pueden obtener las diferentes trayectorias, estas trayectorias generadas corresponden a cierto porcentaje del vector de probabilidad para decisión las cuales están libres de colisión en el ambiente.

Con las trayectorias obtenidas mediante la optimización con algoritmos genéticos y de un filtro de suavizado (filtro Sgolay) da como resultado unas trayectorias con menos

cambios bruscos en su dirección, con el cual al momento de implementar el control de caminos, el UAV sigue de manera correcta el camino propuesto.

De acuerdo con las simulaciones ya realizadas respectivamente en cada uno de los ambientes parcialmente estructurados y a su vez realizando las acciones de control para el UAV, se concluye que para cada ambiente se tiene una trayectoria óptima de las tres que se generan, la cual tiene los factores de optimización como distancia y tiempo de menor valor que las otras trayectorias; esto quiere decir que la optimización por algoritmos genéticos para algoritmos de planificación RRT 3D cumple con su principal funcionalidad para este trabajo.

Queda planteado el tema de optimización con la utilización de algoritmos genéticos para dar continuidad a futuros trabajos en cualquier campo de investigación, pues no se pretende reemplazar técnicas ya existentes y probadas sino la de visualizar estas técnicas en lo probabilístico con lo determinístico.

5.2 Recomendaciones

Como recomendación principal es la de tomar en cuenta los requisitos computacionales para el sistema multicapa planteado, tanto en mapeado, planificación, optimización y simulación, puesto que desde la primera capa el gasto computacional en memoria, tiempo y energía es cuantioso, para ello un computador de altas prestaciones ayuda de manera significativa a la rápida ejecución de cada una de las capas y así poder visualizar los resultados en menos tiempo.

Profundizar en el estudio y continuar con el desarrollo de algoritmos genéticos para ser aplicados a otros problemas de optimización, usando técnicas combinadas para mejorar los operadores de evolución y mutación; además de realizar ejecuciones de simulación en prueba y error en el trabajo con algoritmos en paralelo los cuales pueden reducir el costo en tiempo significativamente.

REFERENCIAS BIBLIOGRÁFICAS

- Abbadi, A., Matousek, R., Jancik, S., & Roupec, J. (2012). RAPIDLY-EXPLORING RANDOM TREES: 3D PLANNING, 6. <https://doi.org/10.13140/2.1.3632.3848>
- Abbadi, A., Matousek, R., Minar, P., & Soustek, P. (2011). RRTs Review and Options. *Computational Engineering in Systems Applications*, 2, 7.
- Andaluz, A. M. (2016). *Algoritmos Evolutivos y Algoritmos Genéticos*. Universidad Carlos III de Madrid.
- Andaluz, V. H., Ortiz, J. S., & Sánchez, J. S. (2015). Bilateral Control of a Robotic Arm Through Brain Signals. En L. T. De Paolis & A. Mongelli (Eds.), *Augmented and Virtual Reality* (Vol. 9254, pp. 355-368). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-22888-4_26
- Asqui, L., Andaluz, V., Sánchez, J., & Acosta, J. (2018). Path Planning Based in Algorithm Rapidly-Exploring Random Tree RRT. *Advanced Science Letters*, 24(11), 8831–8836.
- Brandao, A. S., Andaluz, V. H., Sarcinelli-Filho, M., & Carelli, R. (2011). 3-D path-following with a miniature helicopter using a high-level nonlinear underactuated controller. En *2011 9th IEEE International Conference on Control and Automation (ICCA)* (pp. 434-439). Santiago, Chile: IEEE. <https://doi.org/10.1109/ICCA.2011.6138018>
- Bremermann, H. J. (1962). Optimization Through Evolution and Recombination. *BioSystems*, 12.
- CESPE, C. de E. E. (2015, marzo 2). Los cinco drones más letales de todos los tiempos. Recuperado 7 de agosto de 2018, de <http://cespe.espe.edu.ec/los-cinco-drones-mas-letales-de-todos-los-tiempos/>
- Cohen, P. R., & Feigenbaum, E. A. (2014). *The Handbook of Artificial Intelligence*. Butterworth-Heinemann.

- Cruz, P. P. (2011). *Inteligencia Artificial. Con Aplicaciones a la Ingeniería* (Vol. 1). Barcelona: Marcombo.
- Darwin, C. (2009). *El origen de las especies por medio de la selección natural*. Editorial CSIC - CSIC Press.
- Dedeoglu, G., & Sukhatme, G. S. (2000). Landmark-based Matching Algorithm for Cooperative Mapping by Autonomous Robots. En L. E. Parker, G. Bekey, & J. Barhen (Eds.), *Distributed Autonomous Robotic Systems 4* (pp. 251-260). Tokyo: Springer Japan. https://doi.org/10.1007/978-4-431-67919-6_24
- Díaz Moreno, A. (2016, junio 22). Planificador basado en tareas para misiones de vehículos aéreos no tripulados [info:eu-repo/semantics/bachelorThesis]. Recuperado 7 de agosto de 2018, de <http://oa.upm.es/43830/>
- dji.com. (2018). DJI – The Future Of Possible. Recuperado 5 de enero de 2019, de <https://www.dji.com>
- Dong, Y., Fu, C., & Kayacan, E. (2016). RRT-based 3D path planning for formation landing of quadrotor UAVs. En *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)* (pp. 1-6). Phuket, Thailand: IEEE. <https://doi.org/10.1109/ICARCV.2016.7838567>
- Eduardo H., C. E. (2011). PROPUESTA DE UN ALGORITMO PARA LA PLANEACIÓN DE TRAYECTORIAS DE ROBOTS MÓVILES EMPLEANDO CAMPOS POTENCIALES Y ENJAMBRES DE PARTÍCULAS ACTIVAS BROWNIANAS, 137. EN INGENIERÍA ELECTRÓNICA. (2014, mayo 14). Recuperado 24 de enero de 2019, de <https://docplayer.es/28663794-En-ingenieria-electronica.html>
- Escareño Castro, J. A. (2008). Localización de un helicóptero miniatura utilizando visión artificial. *Tesis de Doctorado, CINVESTAV*.

- Fernández-Peteiro Belmonte, M. (2016). Desarrollo de algoritmos de planificación de caminos en 3D para Blender y V-REP. Recuperado de <https://idus.us.es/xmlui/handle/11441/53158>
- Fraser, A. (1957). Simulation of Genetic Systems by Automatic Digital Computers I. Introduction. *Australian Journal of Biological Sciences*, 10(4), 484. <https://doi.org/10.1071/BI9570484>
- Galipienso, M. I. A., Quevedo, M. A. C., Pardo, O. C., Ruiz, F. E., & Ortega, M. A. L. (2003). *Inteligencia artificial: modelos, técnicas y áreas de aplicación*. Editorial Paraninfo.
- Garamszegi, L. Z. (Ed.). (2014). *Modern Phylogenetic Comparative Methods and Their Application in Evolutionary Biology: Concepts and Practice*. Berlin Heidelberg: Springer-Verlag. Recuperado de [//www.springer.com/us/book/9783662435496](http://www.springer.com/us/book/9783662435496)
- Holland, J. H. (1962). Outline for a Logical Theory of Adaptive Systems. *Journal of the ACM*, 9(3), 297-314. <https://doi.org/10.1145/321127.321128>
- Jonathan. (2016, agosto 11). Holy Stone F181 review - best seller for many reasons! Recuperado 5 de enero de 2019, de <http://www.dronesglobe.com/review/holy-stone-f181/>
- Jorge Arranz de la Peña, & Parra Truyol, A. (2007). Algoritmos Genéticos, p. 8.
- Kluger, M. J. (2015). *Fever: Its Biology, Evolution, and Function*. Princeton University Press.
- Lahoz-Beltra, R. (2010). *Bioinformática: Simulación, vida artificial e inteligencia artificial*. Ediciones Díaz de Santos.
- Latombe, J.-C. (2012). *Robot Motion Planning*. Springer Science & Business Media.
- LaValle, S. M. (1998). *Rapidly-Exploring Random Trees: A New Tool for Path Planning*.
- LaValle, S. M., & Kuffner, J. J. (2000). *Rapidly-Exploring Random Trees: Progress and Prospects*.
- Li, D., Li, Q., Cheng, N., & Song, J. (2012). Extended RRT-based path planning for flying robots in complex 3D environments with narrow passages. En *2012 IEEE International*

- Conference on Automation Science and Engineering (CASE)* (pp. 1173-1178). Seoul, Korea (South): IEEE. <https://doi.org/10.1109/CoASE.2012.6386513>
- Lopez, D., Gomez-Bravo, F., Cuesta, F., & Ollero, A. (2006). PLANIFICACIÓN DE TRAYECTORIAS CON EL ALGORITMO RRT. APLICACIÓN A ROBOTS NO HOLÓNOMOS, 3(3), 12.
- López García, D. A. (2012). *Nuevas aportaciones en algoritmos de planificación para la ejecución de maniobras en robots autónomos no holónomos: memoria para optar al grado de doctor*. Universidad de Huelva, Huelva.
- LOZANO-PEREZ, Toma. (1990). Spatial Planning: A Configuration Space Approach, 2. https://doi.org/10.1007/978-1-4613-8997-2_20
- Luan, C., & Fang, Z. (2016). Random particles boosted RRT for complicated 3D environments with narrow passages. En *2016 12th World Congress on Intelligent Control and Automation (WCICA)* (pp. 3271-3276). Guilin, China: IEEE. <https://doi.org/10.1109/WCICA.2016.7578529>
- MICRODRONE. (2014). Micro Drone 3.0 - Combo. Recuperado 7 de agosto de 2018, de <http://www.microdrone.co.za/micro-drone-3-0-combo-pack.html>
- Miguel Gonzáles. (2017, julio 12). Tipos de drones. Recuperado 7 de agosto de 2018, de <https://filmora.wondershare.com/es/drones/types-of-drones.html>
- Nilsson, N. J. (2014). *Principles of Artificial Intelligence*. Morgan Kaufmann.
- Olgún, J. (2008). *ALGORITMO PARA LA OPTIMIZACIÓN DE PARÁMETROS CONTINUOS VÍA SIMULACIÓN BASADO EN UN ALGORITMO GENÉTICO HÍBRIDO*. PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE ESCUELA DE INGENIERIA, Santiago de Chile. Recuperado de <https://repositorio.uc.cl/bitstream/handle/11534/1448/505053.pdf?sequence=1&isAllowed=y>

- Parkin, R. M., & Gonzáles, V. J. (2005). EVADIENDO OBSTÁCULOS CON ROBOTS MÓVILES. *Revista Digital Universitaria*, 6(1), 9.
- Phantom, dji. co. (2018). DJI Phantom Drone. Recuperado 5 de enero de 2019, de <https://www.dji.com/phantom>
- Rao, N. S. V. (2000). Terrain Model Acquisition By Mobile Robot Teams and n-Connectivity. En L. E. Parker, G. Bekey, & J. Barhen (Eds.), *Distributed Autonomous Robotic Systems 4* (pp. 231-240). Tokyo: Springer Japan. https://doi.org/10.1007/978-4-431-67919-6_22
- Rekleitis, I. M., Dudek, G., & Milios, E. E. (2000). Graph-Based Exploration using Multiple Robots. En L. E. Parker, G. Bekey, & J. Barhen (Eds.), *Distributed Autonomous Robotic Systems 4* (pp. 241-250). Tokyo: Springer Japan. https://doi.org/10.1007/978-4-431-67919-6_23
- Samaniego, F., Sanchis, J., Garcia-Nieto, S., & Simarro, R. (2017). UAV motion planning and obstacle avoidance based on adaptive 3D cell decomposition: Continuous space vs discrete space. En *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)* (pp. 1-6). Salinas: IEEE. <https://doi.org/10.1109/ETCM.2017.8247533>
- Santana, A. M., Ortiz, O. P., Acosta, J. F., & Andaluz, V. H. (2018). Autonomous Assistance System for People with Amyotrophic Lateral Sclerosis. En *IT Convergence and Security 2017* (pp. 267-277). Springer, Singapore. https://doi.org/10.1007/978-981-10-6451-7_32
- Schotland, J. C. (2000). Tomography with diffusing photons: a Feynman path integral perspective. En *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)* (Vol. 6, pp. 3791-3794 vol.6). <https://doi.org/10.1109/ICASSP.2000.860228>
- Steels, L., Brooks, R., & Brooks, R. (2018). *The Artificial Life Route to Artificial Intelligence: Building Embodied, Situated Agents*. Routledge. <https://doi.org/10.4324/9781351001885>

- Wang, H., Yan, B., Li, X., Luo, X., Yang, Q., & Yan, W. (2016). On optimal path planning for UAV based patrolling in complex 3D topographies. En *2016 IEEE International Conference on Information and Automation (ICIA)* (pp. 986-990). <https://doi.org/10.1109/ICInfA.2016.7831962>
- Wang, Hongpeng, Li, H., Zhang, C., He, S., & Liu, J. (2017). A 3D coverage path planning approach for flying cameras in nature environment under photogrammetric constraints. En *2017 36th Chinese Control Conference (CCC)* (pp. 6761-6766). Dalian, China: IEEE. <https://doi.org/10.23919/ChiCC.2017.8028424>
- xdrones.es. (2018). Tipos de drones - Conoce todos los tipos de drones que existen. Recuperado 5 de enero de 2019, de <https://www.xdrones.es/tipos-de-drones-clasificacion-de-drones-categorias-de-drones/>
- Yiqun, D., Jun, F., Bin, Y., Youmin, Z., & Jianliang, A. (2015). Position and heading angle control of an unmanned quadrotor helicopter using LQR method. En *2015 34th Chinese Control Conference (CCC)* (pp. 5566-5571). <https://doi.org/10.1109/ChiCC.2015.7260508>



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA E INSTRUMENTACIÓN

CERTIFICACIÓN

Se certifica que el presente trabajo fue desarrollado por los señores **MOLINA PALACIOS JOSÉ IGNACIO** y **NARANJO ERAZO MAURICIO XAVIER**.

En la ciudad de Latacunga, a los días veintitrés del mes de enero del 2019.

Ing. Julio Acosta

DIRECTOR DE PROYECTO

Ing. Amparo Meythaler

DIRECTORA DE CARRERA

Aprobado por:



Dr. Rodrigo Vaca

SECRETARIO ACADEMICO