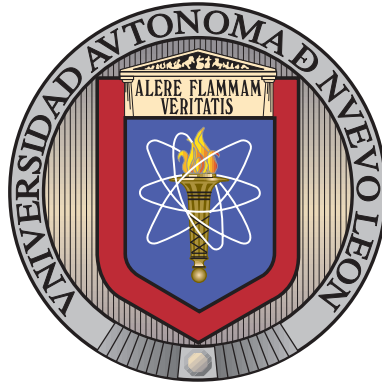


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



DISEÑO DE ALGORITMO GENETICO PARA
PLANEACIÓN DE TRAYECTORIA DE UN UAV

POR

MANUEL ALEJANDRO GUTIÉRREZ MARTÍNEZ

COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE

MAESTRÍA EN INGENIERÍA AERONÁUTICA

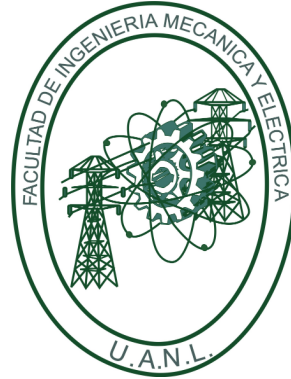
CON ORIENTACIÓN EN ESTRUCTURAS

DICIEMBRE 2020

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



DISEÑO DE ALGORITMO GENETICO PARA
PLANEACIÓN DE TRAYECTORIA DE UN UAV

POR

MANUEL ALEJANDRO GUTIÉRREZ MARTÍNEZ

COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE

MAESTRÍA EN INGENIERÍA AERONÁUTICA

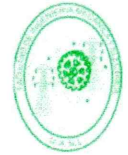
CON ORIENTACIÓN EN ESTRUCTURAS

DICIEMBRE 2020



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

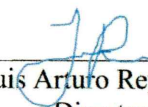



FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA


Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica
Subdirección de Estudios de Posgrado


Los miembros del Comité de Tesis recomendamos que la Tesis “Diseño de algoritmo genético para planeación de trayectoria de un UAV”, realizada por el alumno Manuel Alejandro Gutiérrez Martínez, con número de matrícula 1517068, sea aceptada para su defensa como requisito para obtener el grado de Maestría en Ingeniería Aeronáutica con Orientación en Estructuras.

El Comité de Tesis



Dr. Luis Arturo Reyes Osorio
Director


Dr. Octavio García Salazar
Co-Director


Dr. Filiberto Muñoz Palacios
Revisor


Dr. Eduardo Steed Espinoza Quesada
Revisor

Vo. Bo.


Dr. Simón Martínez Martínez
Subdirector de Estudios de Posgrado



093

San Nicolás de los Garza, Nuevo León, 11 de enero de 2021



Dedico esta tesis a Dios, a mis padres y a todos los que me apoyaron a mis estudios.

ÍNDICE GENERAL

Agradecimientos	XII
Resumen	XIV
1. Introducción	1
1.1. Antecedentes	1
1.2. Estado del arte	3
1.3. Planteamiento de problemática	4
1.4. Justificación	5
1.5. Hipótesis	5
1.6. Objetivos	6
1.7. Metodología	6
2. Preliminares	8
2.1. Planeación de trayectoria	8
2.1.1. Métodos clásicos	9
2.1.2. Métodos Heurísticos.	13

2.2. Algoritmos genéticos	17
2.2.1. Función objetivo.	20
2.2.2. Operadores genéticos.	21
3. Algoritmo genético para planeación de trayectoria.	26
3.1. Inicialización algoritmo genético.	27
3.2. Diseño de función multi-objetivo.	29
3.2.1. Longitud de arco.	29
3.2.2. Distancia entre puntos de navegación a obstáculos.	31
3.2.3. Cruce de rutas con obstáculos.	33
3.3. Diseño de operadores genéticos.	36
3.3.1. Método de selección	36
3.3.2. Operador de Cruce.	37
3.3.3. Operador de Mutación.	39
3.4. Algoritmo para entorno estático y dinámico.	41
3.5. Suavizado de trayectoria.	45
4. Resultados y experimentos.	46
4.1. Entorno estático.	47
4.1.1. Caso A: Un obstáculo entre el punto inicial y el punto final.	48
4.1.2. Caso B: Dos obstáculos entre el punto inicial y el punto final.	48
4.1.3. Caso C: Tres obstáculos que encierran el punto final.	50

4.1.4. Caso D: Cuatro obstáculos colocados de forma aleatoria. . . .	51
4.1.5. Caso E: Cuatro obstáculos en formación de doble barrera. . .	52
4.1.6. Gráficas de peso.	53
4.1.7. Algoritmo estático 3D	54
4.1.8. Algoritmo dinámico 3D	58
5. Conclusiones y trabajos futuros.	73
A. Apéndice A	75
A.1. Ecuaciones de movimiento de quadrtor	75

ÍNDICE DE FIGURAS

2.1. Ejemplo descomposición de celdas.	9
2.2. Ejemplo descomposición de celdas.	11
2.3. Ejemplo de campos potenciales.	12
2.4. Ejemplo de métodos gráficos.	13
2.5. Ejemplo de estructura de red neuronal.	16
2.6. Ejemplo de optimización de colonia de hormigas.	18
2.7. Tipos de parámetros en el algoritmo genético.	19
2.8. Analogía entre naturaleza y computación evolutiva.	20
2.9. Tipos de criterios de Selección.	23
2.10. Criterio de cruce.	24
2.11. Criterio de cruce.	25
3.1. Diagrama de flujo de metodología.	26
3.2. Evaluación de largo de trayectorias.	30
3.3. Evaluación de puntos de navegación.	32
3.4. Evaluación de cruce de trayectoria por obstáculos.	35

3.5. Operador genético de selección.	37
3.6. Operador genético de cruce.	39
3.7. Evolución de proceso de cruce:a)iteración 1, b) iteración 5, c) iteración 8, d) iteración 10, e) iteración 14, f) iteración 29	39
3.8. Operador genético de mutación.	40
3.9. Algoritmo en entorno estático.	44
3.10. Algoritmo en entorno dinámico.	44
4.1. Pruebas en entorno estático implementadas.	47
4.2. Trayectoria generada 2D para un obstáculo.	49
4.3. Trayectoria generada 2D para dos obstáculos.	50
4.4. Trayectoria generada 2D para tres obstáculos.	51
4.5. Trayectoria generada 2D para cuatro obstáculos aleatorios.	52
4.6. Trayectoria generada 2D para cuatro obstáculos en formación barreras.	53
4.7. Gráficas de peso para entorno estático.	55
4.8. Trayectoria generada 3D para un obstáculo.	56
4.9. Gráfica de peso para entorno 3D para un obstáculo.	56
4.10. Trayectoria generada 3D para tres obstáculos.	57
4.11. Gráfica de peso para entorno 3D para tres obstáculos.	57
4.12. Trayectoria dinámica un obstáculo movimiento diagonal	59
4.13. Gráfica de peso en entorno 3D dinámico para un obstáculo en movi- miento diagonal.	60

4.14. Gráfica de peso en entorno 3D dinámico para un obstáculo en movimiento diagonal.	61
4.15. Trayectoria dinámica un obstáculo movimiento vertical.	63
4.16. Grafica de peso en entorno 3D dinámico para un obstáculo en movimiento diagonal.	64
4.17. Gráfica de peso en entorno 3D dinámico para un obstáculo en movimiento diagonal.	64
4.18. Trayectoria dinámica tres obstáculos.	67
4.19. Gráfica de peso en entorno 3D dinámico para un obstáculo en movimiento diagonal.	68
4.20. Gráfica de peso en entorno 3D dinámico para un obstáculo en movimiento diagonal.	68
4.21. Movimiento en el eje X sin y con procesamiento caso 1.	69
4.22. Movimiento en el eje Y sin y con procesamiento caso 1.	69
4.23. Movimiento en el eje Z sin y con procesamiento caso 1.	69
4.24. Trayectoria dinamica tres obstaculos caso 1	70
4.25. Movimiento en el eje X sin y con procesamiento caso 2.	71
4.26. Movimiento en el eje Y sin y con procesamiento caso 2.	71
4.27. Movimiento en el eje Z sin y con procesamiento caso 2.	71
4.28. Trayectoria dinamica tres obstaculos caso 2	72

ÍNDICE DE TABLAS

3.1. Función objetivo 1	30
3.2. Función objetivo 2	33
3.3. Función objetivo 3.	34
3.4. algoritmo de alta mutación fuera de la población de convergencia . .	42
3.5. Mutación para convergencia a punto final.	43
3.6. Mutación para generar nueva población aleatoria.	44

AGRADECIMIENTOS

Quiero agradecer especialmente a Dios por el regalo de la vida y por todas las gracias que me ha regalado, también especialmente a mis padres Juan Manuel Gutierrez Izaguirre, María del Carmen Martinez Montemayor por todo su amor y espero que esta tesis sea una pequeña retribución a todo lo que me han dado.

También agradezco a mis asesores por la paciencia y las correcciones hechas, por todo el tiempo dedicado y por creer en mi por todo lo que me han enseñado en estos años, a mi director de tesis el Dr. Luis Arturo Reyes Salazar a mi codirector de tesis el Dr. Octavio García Salazar, y a mis revisores el Dr. Eduardo Steed Espinoza Quesada y el Dr. Filiberto Muñoz Palacios.

Agradezco a mis amigos y compañeros de generación por apoyarme y hacer más pasable mi estadía en esta maestría Luis Enrique Cabriales Ramírez e Israel de Santiago Talavera, a todos mis compañeros en el Laboratorio de Navegación especialmente a mi amigo Erik Gilberto Rojo Rodríguez que me ayudo y me enseñó muchas cosas sin las cuales no habría podido realizar este trabajo.

También quiere agradecer a la UANL y a la FIME por su profesionalismo y calidad en sus programas educativos e instalaciones, por darme la oportunidad de estudiar una maestría y así seguir preparándome y ser un mejor profesionista, también al CIIA que ha sido una casa para mi desde hace ya unos años, a todos sus maestros, becarios y trabajadores que siempre están en la mejor disposición de ayudar.

Por último agradezco al CONACyT por apoyarme en mis estudios por la beca que me otorgaron "2019-000002-01NACF-08878" que fue de gran ayuda para mis estudios y por creer en la ciencia.

RESUMEN

Manuel Alejandro Gutiérrez Martínez.

Candidato para obtener el grado de Maestría en Ingeniería Aeronáutica con Orientación en Estructuras.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio: DISEÑO DE ALGORITMO GENETICO PARA PLANEACIÓN DE TRAYECTORIA DE UN UAV.

Número de páginas: 83.

OBJETIVOS: Los objetivos de esta tesis son el diseño de un algoritmo de planeación de trayectoria que cumpla las condiciones de trayectoria corta y evasión de obstáculos para entornos estáticos y dinámicos.

CONTRIBUCIONES Y CONCLUSIONES: La contribución principal de este trabajo es el diseño de un algoritmo genético para entornos estáticos y dinámicos. Se puede concluir que las técnicas heurísticas como los algoritmos genéticos son buenas alternativas para la planeación de trayectoria eficiente.

Firma del asesor: _____
Dr.Luis Arturo Reyes Osorio

CAPÍTULO 1

INTRODUCCIÓN

1.1 ANTECEDENTES

En recientes años muchos investigadores han propuesto un gran número de nuevos métodos e investigaciones sobre la planificación de trayectorias aplicados a la robótica. La navegación, es un factor importante en la robótica móvil, se define como el proceso de identificación precisa de la posición del robot móvil, la planificación de la trayectoria y el seguimiento de la trayectoria[39]. La planificación de trayectorias puede verse como la problemática de encontrar la trayectoria más eficiente que por lo general suele ser la que tiene menos consumo de energía en el menor tiempo posible. Para esto los algoritmos de planeación de trayectoria se encargan de encontrar la trayectoria más corta entre un punto inicial y un punto final evitando colisionar con algún obstáculo, de esta manera la planeación de trayectoria también se puede ver como un problema de optimización [20]. Actualmente existen muchas técnicas de planeación de trayectoria que han sido investigadas y han sido utilizadas en VANTs(Vehículo aéreo no tripulado UAV por sus siglas en ingles), estas técnicas se pueden dividir en basada en espacio (métodos clásicos) y métodos coordinados (métodos heurísticos). Entre algunos de los métodos clásicos más utilizados en la planeación de trayectoria están los algoritmos de mapas de ruta, como los gráficos de visibilidad y los diagramas de Voronio, estos permiten pasar lo más cerca posible

de los obstáculos, mientras que los diagramas de Voronio generan caminos que están más alejados de los obstáculos; cada uno de los métodos tiene ventajas en cuestión de trayectorias cortas o trayectorias seguras, sin embargo aunque estas son alternativas interés suelen ser técnicas poco flexibles [14]. Otra técnica utilizada son los campos potenciales los cuales son eficiente para aplicaciones de evasión de obstáculos y tienen respuestas satisfactorias en entornos dinámicos, pero dado que estos no dependen de una función objetivo para minimizar, estos suelen caer en mínimos locales [11]. Aunque los métodos clásicos obtienen soluciones eficientes tienen un alcance limitado, así fue que surgió la necesidad de nuevas técnicas como los métodos o algoritmos heurísticos cuya principal característica es que tienen la capacidad de aprender y auto mejorarse, además estos métodos suelen no solo depender de modelos matemáticos, sino que suelen ser inspirados en procesos de la vida real. Cabe destacar que los métodos heurísticos no encuentran una solución óptima, sino que encuentran soluciones cercanas a las óptimas. Estos algoritmos se pueden clasificar como de solución única aquí se encuentran algunos como, simulated Annealing, local search, iterated local search, entre otros, la otra clasificación suele ser los métodos evolutivos y los métodos bioinspirados como pueden ser, OPH (optimización por colonia de hormigas o ACO por sus siglas en ingles de ant colony), OEP (optimización por enjambre de partículas o PSO por sus siglas en ingles de particle swarm optimization) y los algoritmos genéticos, también se pueden encontrar las redes neuronales y la lógica difusa [5], [32] que también son métodos muy conocidos. Los algoritmos genéticos comparados con otros métodos como las redes neuronales que suelen tener largo tiempo de aprendizaje lo que puede hacer complicado sus aplicaciones o la lógica difusa suele ser altamente robusta y por lo tanto con gran costo computacional, los algoritmos genéticos suelen tener una buena relación entre costo computacional y tiempos de convergencia. Los algoritmos genéticos fueron originalmente desarrollados por John holland en 1975 [23]. Más tarde Goldberg [16], que fue uno de sus estudiantes termino de desarrollar los algoritmos genéticos y hacerlos populares, estos algoritmos están inspirados en los procesos de evolución natural de las especies, reproducción sexual, y supervivencia de especies. Estos son de los algoritmos

heurísticos más utilizados y se han utilizado en una gran variedad de áreas como robótica, mecatrónica automatización aeronáutica entre otras, además de que estas son ampliamente usadas para encontrar soluciones eficientes donde hay aplicaciones con muchas posibles soluciones como es el caso de planeación de trayectoria.

1.2 ESTADO DEL ARTE

En los últimos años, la planificación de trayectorias y la evasión de obstáculos han ganado el interés de ingenieros y científicos, principalmente de los campos de la robótica y la mecatrónica. Por lo que han surgido la necesidad de nuevas técnicas más sofisticadas para VANTs de planeación de trayectorias y de control [13]. Los algoritmos genéticos son una herramienta eficiente que permiten hacer planeación de trayectorias, este se basa en generar puntos de navegación (waypoints en inglés) en un espacio libre, el algoritmo hace una búsqueda de aquellos que cumplen las restricciones que por lo general es procurar que evite colisionar con obstáculos y que encuentre una ruta corta para ahorro de energía y de recursos. Algunos investigadores como Bandala et al [12], han usado los algoritmos genéticos para planeación de trayectorias para VANTs en entornos tridimensionales estáticos para guardar tiempo y energía de forma óptima, los resultados de este algoritmo demostraron que se logró minimizar la distancia recorrida basado en criterios de distancia y puntos de navegación en un espacio muestreado, lograron implementar este tipo de algoritmos para entornos más parecidos a lo que trabajan los VANTs es decir en entornos en 3D, para este caso se obtuvieron soluciones en 500 iteraciones mostrando que la tercera dimensión incrementa considerablemente el número de cálculos posibles. Otros autores como en [41], presenta un nuevo método de planificación de trayectoria para VANTs mediante un algoritmo genético, su algoritmo funciona añadiendo conocimiento previo a las funciones objetivos logrando reducir el costo computacional para así encontrar una trayectoria eficiente. Jihua Tao et al [37], en su artículo trabaja con un criterio de codificación de números reales basado el cambio de curso, el algoritmo

trabaja con una ruta local que después basándose en el cambio de curso, el algoritmo construye la ruta temporal por un vector real de codificación individual y sobre esta base, los operadores relacionados diseñan mediante el nuevo método de codificación una trayectoria mejorada, las simulaciones en este método mostraron mejorar la capacidad de búsqueda global y mejorar la calidad de la trayectoria de vuelo para los VANTs. Otros estudios sugieren métodos con operadores genéticos mejorados, autores como Chaymaa Lamini et al [28], utilizaron un operador de cruce mejorado que trabaja con cromosomas variables para resolver los problemas de planificación de trayectoria en entorno estático, este tipo de enfoque muestra que se puede evitar una convergencia prematura en un mínimo local ofreciendo rutas factibles, pero a su vez estos enfoques con operadores mejorados ayudan a converger en el mínimo global de manera más rápida. También este tipo de algoritmos ha demostrado ser factible para aplicaciones reales considerando condiciones de vuelo en entorno estático y entorno dinámico en 3D para poder aplicar cambios de trayectoria en tiempo real durante vuelo [25]. Otros investigadores [1], utilizaron algoritmos genéticos para encontrar soluciones en un entorno estático en 2D en este artículo se logró encontrar soluciones en 70 iteraciones en tiempos de 30 segundos. Otras investigaciones como en [3], mostraron este tipo de algoritmos para entornos dinámicos donde encontraron soluciones de entre 10 a 40 iteraciones, entre 30 a 140 segundos, estos algoritmos suelen ser eficientes ya que comparado con los tiempos de aprendizaje de redes neuronales suelen ser rápidos y comparados con lógica difusa suelen ser menos robustos. En este trabajo se propone un algoritmo genético que sea capaz de generar una ruta que sea corta y que evite colisionar con obstáculos.

1.3 PLANTEAMIENTO DE PROBLEMÁTICA

En este trabajo de investigación se trata la problemática de encontrar una ruta que sea segura y que sea la más corta posible para un VANT que vuela desde un punto de inicio a un punto final, en un campo de obstáculos. Para encontrar

la trayectoria más corta posible, se utiliza un algoritmo genético en donde los cromosomas representan posibles puntos de navegación. Cada uno de estos puntos es entonces evaluado por funciones objetivas diseñadas, con el fin de clasificarlos por su posibilidad de formar una ruta viable. Este proceso repite lo anterior y se reproduce un porcentaje de la población inicial, reemplazando los individuos menos fuertes. Este proceso se repite hasta que todos los individuos convergen en un solo punto de navegación, lo que significa que este es el mejor camino formado posible. Si no se encuentra una ruta posterior con un solo punto de navegación, el algoritmo inicia otra capa, iterando con pares de puntos de ruta, y así sucesivamente, hasta encontrar una trayectoria posterior.

1.4 JUSTIFICACIÓN

Debido al incremento de la necesidad de aeronaves no tripuladas cada vez más independientes y autónomas en aplicaciones civiles y militares como vigilancia, reconocimiento, topología, paquetería entre otros. Es necesario algoritmos de planeación de trayectoria que permitan a los VANTs aprender a moverse con libertad, de manera autónoma y con interacción inteligente en su entorno de trabajo, además de tener trayectorias eficientes para resguardar energía y recursos computacionales. Debido a estas razones este trabajo de investigación resuelve la planificación de trayectoria proponiendo un algoritmo genético con una función multiobjetivo para poder encontrar soluciones eficientes, para diferentes entornos de trabajo estático y dinámico en 3 dimensiones espaciales.

1.5 HIPÓTESIS

El algoritmo genético diseñado será capaz de realizar la tarea de planeación de trayectoria en un espacio de trabajo y cumplirá las condiciones de evasión de obstáculos

y trayectoria mínima consiguiendo la mejor trayectoria posible.

1.6 OBJETIVOS

Diseñar un algoritmo genético de planeación de trayectoria que cumpla con las condiciones de trayectoria de distancia mínima y evasión de obstáculos para entorno estático y entorno dinámico.

OE1. Desarrollar un algoritmo con base genética de planeación de trayectorias en entorno estáticos.

OE2. Expandir el algoritmo para entornos dinámicos.

OE3. Expandir el algoritmo en un entorno 3D.

OE4. Implementar el algoritmo en un VANT y realizar pruebas.

1.7 METODOLOGÍA

La metodología desarrollada en este trabajo de investigación fue la siguiente, primero se diseñaron los criterios de la función multiobjetivo del algoritmo genético. Después de tener los criterios de la función objetivo se diseñaron y seleccionaron los parámetros del algoritmo genético es decir el tamaño de la población inicial y la longitud del cromosoma para la resolución del espacio, además que se determinó un criterio de codificación para traducir las cadenas de bits a coordenadas en el espacio. También se determinó y se eligió un criterio generacional del algoritmo para tener un constante cambio de los individuos en cada generación. Se evalúa el funcionamiento de la función objetivo para determinar si los resultados eran lógicos para la aplicación. Luego se crearon los operadores genéticos para generar la diversidad de población y finalmente poder converger. Para el operador de selección se escogió un criterio determinístico para tener una convergencia rápida sin caer en un mínimo local, el operador de cruce se utilizó un punto único de cruce que fue más que suficiente para

la aplicación y para el criterio de mutación se utilizó un criterio clásico lo cual sirvió bien para el entorno estático, pero para el entorno dinámico se tuvieron que diseñar criterios mejorados de mutación. Finalmente se obtuvo el algoritmo para entorno estático en 2D, y posteriormente se agregó la tercera dimensión, luego se implementaron las condiciones mejoradas de mutación para realizar correr el algoritmo en un entorno dinámico tridimensional. Por último, realizo un procesamiento de suavizado de trayectoria mediante un filtro paso bajas. Para el entorno estático se realizaron pruebas en un VANT para diferentes condiciones de trabajo.

La estructura para este trabajo de investigación es la siguiente: capítulo 2 preliminares sobre métodos de planeación de trayectoria y conceptos sobre algoritmos genéticos, en el capítulo 3 se encuentra el desarrollo o metodología utilizados para el desarrollo del algoritmo genético para planeación de trayectoria, en el capítulo 4 se presentan los resultados para los diferentes casos y entornos de trabajo, el capítulo 5 se presentan las conclusiones y los trabajos futuros, por último en el capítulo 6 se encuentran las referencias utilizadas en el trabajo de investigación. La contribución principal de este trabajo es el diseño de un algoritmo genético para la tarea de planeación de trayectoria y generar conocimiento sobre como los métodos heurísticos se pueden implementar en este tipo de aplicaciones para VANTs.

CAPÍTULO 2

PRELIMINARES

2.1 PLANEACIÓN DE TRAYECTORIA

Una de las tareas más importantes en la robótica es la navegación donde existen muchas técnicas en la literatura [2]. Una de las áreas pilares de la navegación es la planeación de trayectorias, donde podemos entender el problema de planeación de trayectoria como encontrar una ruta o camino a seguir que sea eficiente, esto es; un camino que sea seguro desde un punto de partida a un punto final deseado, algunos autores lo definen como encontrar una trayectoria a través de un mapa o entorno de trabajo, en el que un vehículo puede llegar a una ubicación objetivo. La planeación de trayectoria se considera un problema complejo en la computación debido a que puede aumentar su complejidad exponencialmente con la dimensión del espacio, el factor del espacio es también es de suma importancia debido a que es el lugar donde se tiene que generar las trayectorias, algunos autores [7] lo definen como espacio de configuración debido a que como su nombre lo dice es el espacio donde todas las configuraciones posibles pueden suceder. Este tipo de problemáticas de planificación de trayectoria se puede realizar en dos enfoques, el enfoque estático o global, en este enfoque los vehículos tienen acceso o conocimiento del entorno en el que se mueve, es decir que el robot conoce completamente sobre todo el entorno desde las posiciones de los obstáculos y de los objetivos. El otro enfoque es el entorno dinámico

o local, donde no se tienen información previa sobre su medio o ambiente además que la trayectoria se puede ir recalculando conforme los obstáculos se mueven [4]. Un esquema de los diferentes métodos y clasificación de las técnicas utilizadas para planeación de trayectoria se observa en la Figura 2.1.

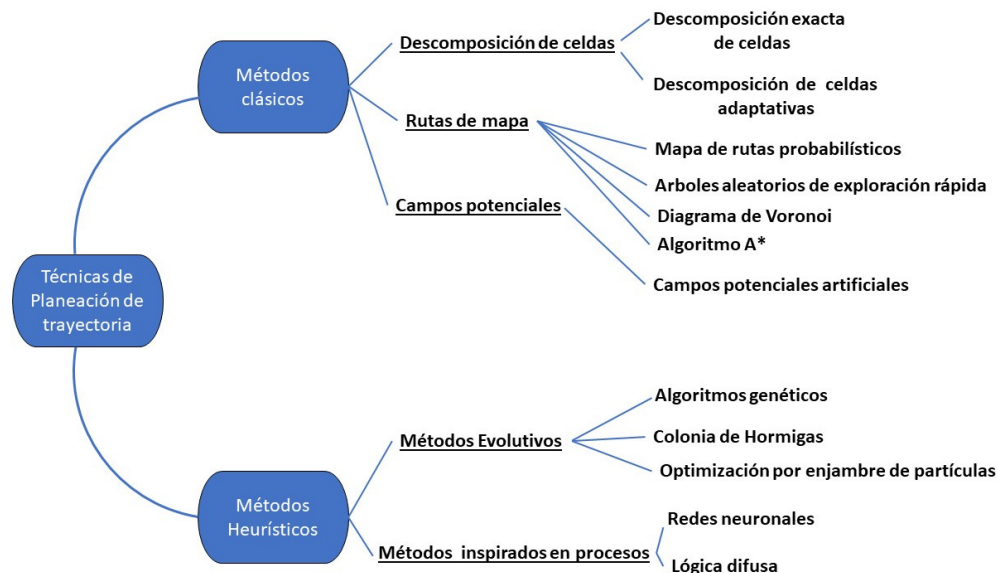


Figura 2.1: Ejemplo descomposición de celdas.

2.1.1 MÉTODOS CLÁSICOS

Los métodos clásicos fueron las primeras técnicas propuestas para resolver el problema de planeación de trayectoria por lo tanto son métodos más antiguos. Por métodos clásicos se entienden aquellos métodos que solo suelen depender de algún modelo matemático, de cuestiones geométricas, o de métodos de grafos. Estos métodos suelen tener resultados eficientes y ya ha sido publicados muchos trabajos relacionados con ellos para la robótica y en la implementación de VANTs [15], [9]. Las conclusiones que se han obtenido es que suelen tener resultados rígidos es decir o logran encontrar soluciones o no hay solución, esto a su vez suele ser una desventaja ya que significa

que suelen caer en mínimos locales, otra de las desventajas que suelen tener este tipo de enfoques es su intensidad computacional y su incapacidad para hacer frente a elementos desconocidos o parcialmente desconocidos y sobre todo suelen tener problemas para implementación en entornos dinámicos [39].

2.1.1.1 DESCOMPOSICIÓN DE CELDAS

Uno de los primeros métodos clásicos y de los más antiguos es el método de descomposición de celdas el cual es muy utilizado en la literatura en problemas de planificación de rutas, sin embargo aunque son métodos antiguos siguen siendo ampliamente utilizados y mejorado para aplicaciones con VANTs [33]. Esta técnica surgió de la necesidad de separar espacios de muestreo grandes en varios espacios más pequeños para tener mejor exploración del espacio, por lo que este método consiste en reducir el espacio de trabajo o de búsqueda mediante el uso de una representación de celdas y a través de la búsqueda de celdas libres de colisiones y así crear una secuencia de celdas que servirán para crear la trayectoria desde un punto inicial hasta el punto final, en el caso de que se encontrara una celda corrupta o inviable este método sigue haciendo más divisiones de celda hasta que exista una celda que no esté en la zona del obstáculo, de esta manera se garantiza el criterio de evasión de obstáculos. En la Figura 2.2 se muestra una secuencia de pasos de cómo funciona la técnica de descomposición de celdas, primero definir el espacio c (espacio de configuración) de trabajo, así como un punto de inicio y un punto final. Segundo se divide el espacio en regiones de búsqueda llamadas celdas y se construye un mapa de grafos a través de las celdas y los vértices creados a partir de los obstáculos. Tercero se determinan las celdas de inicio y final y se proporcionan secuencias de celdas libre de colisiones uniendo las celdas de inicio a final y se crea la trayectoria basándose en la secuencia de celdas obtenidas.

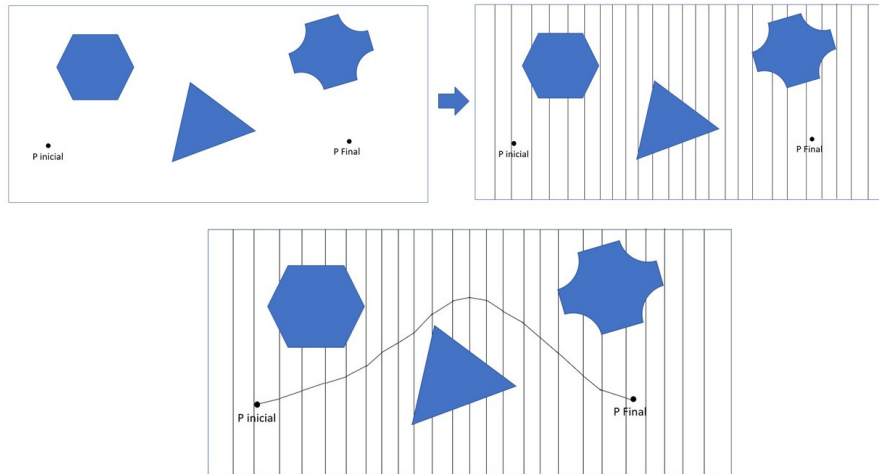


Figura 2.2: Ejemplo descomposición de celdas.

2.1.1.2 CAMPOS POTENCIALES.

Los campos potenciales son una de las primeras técnicas utilizadas para planeación de trayectorias, este método fue originalmente concebido por Khatib [26] y posteriormente desarrollado por Volpe [38]. Aunque tienen varias ventajas en cuestión de aplicación en tiempo real y evasión de obstáculos suele tener algunas desventajas al caer en mínimos locales, aunque algunos autores muestran métodos para superar sus limitaciones [31], [43].

La técnica de campos potenciales funciona mediante la interacción de los efectos de los diferentes campos, donde la ruta de los VANTs se calcula sobre la base de los campos resultantes desde un punto inicial a un punto final. Para obstáculos normalmente los campos se configuran como campos de repulsión, para de esta manera evitarlos, para metas u objetivos se configuran como campos de atracción para que el vehículo siempre esté dirigido hacia un objetivo como se observa en la Figura 2.3. Esta técnica suele ser muy efectiva en cuestión de implementación y tiempo real, una de las ventajas de este método es que tienen cierta versatilidad,

ya que podemos proporcionar la libertad hacia como determinar la función de los campos potenciales, sin embargo, una de las desventajas que tienen y una de sus mayores problemáticas es que suelen caer en mínimos locales. En este método, dos fuerzas opuestas se definirán de una manera que el objetivo tendrá un campo que atraerá a un vehículo hacia sí mismo, mientras que los obstáculos o cualquier otro objeto, incluyendo paredes o restricciones tienen fuerzas repulsivas, esta combinación de resultados de fuerzas de atracción y de fuerzas de repulsión conducen al vehículo hacia el destino.

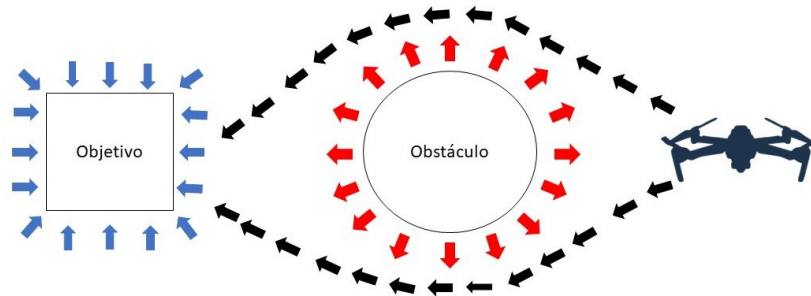


Figura 2.3: Ejemplo de campos potenciales.

2.1.1.3 RUTA DE MAPA

Otra de las técnicas clásicas más utilizadas es el método de hoja de ruta o también como gráfico de visibilidad, este concepto de gráfico de visibilidad relacionado con la planeación de trayectoria fue introducido por primera vez por Lozano-Perez [30], este método trabaja mediante un conjunto de rutas, donde cada ruta consta de conexiones de áreas libres de colisiones como se puede observar en la Figura 2.4, donde los vértices de los obstáculos pueden formar la trayectoria. Después de tener la red construida o mapa vial, este se utiliza para la planificación del movimiento.

Es así como este método puede verse como encontrar el camino más corto libre de colisiones desde un punto inicial a un punto final utilizando la red de mapas de ruta. Los gráficos de visibilidad y de Voroni son de los métodos más populares para desarrollar mapas de rutas, los gráficos de visibilidad se crean basándose en los vértices y a los nodos de los obstáculos y estos vértices suelen crear formas poligonales. La trayectoria generada por los diagramas de Voroni siempre es lo más segura posible debido a que los bordes de la trayectoria están lo más lejos posibles de los obstáculos, por otro lado, mientras que los diagramas visibilidad pasan lo más cerca posible de los obstáculos garantizando así la ruta más corta esto con la desventaja del gráfico de visibilidad al ser el camino más corto pasa lo más cerca posible a los obstáculos en los vértices comprometiendo la seguridad de la ruta.

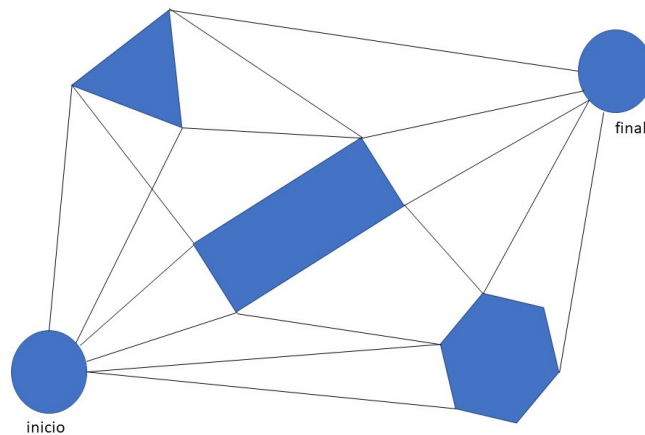


Figura 2.4: Ejemplo de métodos gráficos.

2.1.2 MÉTODOS HEURÍSTICOS.

Uno de los problemas de la navegación está influenciado por la precisión del mapa o de la técnica de búsqueda, además de que en enfoques de navegación dinámica

existe cierta incertidumbre en la precisión del mapa. Los métodos heurísticos ofrecen ciertas ventajas sobre los métodos clásicos en cuestión de búsqueda y se utilizan para una planificación óptima de trayectoria. Algunos estudios han demostrado que algunos métodos convencionales comparados con métodos heurísticos solo resguardan el sistema de colisiones un 12 % mientras que algunos métodos heurísticos han demostrado resguardar de colisiones hasta 70 % [39]. Estos métodos por lo general suelen estar inspirados en procesos de la naturaleza o basándose en población de individuos. En los enfoques basados en el comportamiento, se pueden encontrar algoritmos como redes neuronales los cuales asemejan el proceso de sinopsis del cerebro o la lógica difusa que se basa en los procesos de mundo real. Por otro lado, los algoritmos basados en población están bioinspirados en procesos de optimización de la naturaleza, entre estos métodos están los algoritmos genéticos, de optimización por enjambre de partículas y algoritmo de optimización de colonia de hormigas, estos métodos comparten la característica que en cada iteración, la población cambia o se desarrolla a una nueva población, por lo mismo estos métodos se conocen como métodos evolutivos ya que tiene la capacidad de optimizar [21]. Los algoritmos heurísticos, aunque son muy eficientes no garantizan encontrar la mejor solución, pero si una respuesta basándose en las reglas y restricciones en las que están programados, una de las ventajas que tienen estos métodos es que pueden encontrar soluciones en menor tiempo y menor tiempo computacional.

2.1.2.1 REDES NEURONALES.

Las redes neuronales están inspiradas en la capacidad de hacer sinopsis del cerebro, estos se basan en la capacidad de resolver problemas de optimización, búsqueda, aprendizaje y reconocimiento de patrones. Las redes neuronales se definen como un sistema donde el procesamiento está interconectado por elementos formando una red. Los elementos de una red se definen como nodos y tiene la capacidad de ser adaptables y que se ajustan para resolver repetidamente problemas basados en el

conocimiento empírico obtenido del proceso de aprendizaje. Posteriormente, estos elementos se conectan entre si formando una red.

Algunos investigadores definen el funcionamiento de estos métodos en capas, donde la capa 1 procesa los elementos de entrada, la capa 2 es la encargada de realizar operaciones con la información recibida y la tercera capa devuelve los resultados y salidas, ver Figura 2.5. Estos métodos basados en redes neuronales también se pueden clasificar en tres grupos: su patrón de interconexión de neuronas, la metodología que se utilizó para determinar los pesos de conexiones de las neuronas, y la activación de las neuronas. Algunos autores categorizan el uso de redes neuronales en las siguientes tres categorías, interpolación de los datos sensoriales, evasión de obstáculos y planificación de trayectorias.

Investigadores como Hong [24], han demostrado que usar redes neuronales para planeación de trayectorias es bastante viable, además que otros autores han logrado combinar redes neuronales con técnicas como los algoritmos genéticos obteniendo lo mejor de cada método, demostrando que las redes neuronales aparte de funcionar por su cuenta se pueden complementar con otros métodos trayendo mejores resultados [40].

2.1.2.2 OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS.

El algoritmo de optimización por enjambres de partículas fue presentado por primera vez por Kennedy y Eberhart en 1995 [10], han sido usados en algunas investigaciones para la planificación de trayectorias en tiempos mínimos, derivando la expresión a optimizar y así evitando la colisión de obstáculos [19]. Estos tipos de algoritmos son considerados algoritmos bioinspirados, en este caso está basado en partículas inspiradas en los comportamientos de las relaciones sociales de los animales, por ejemplo, bandadas de aves y bancos de peces, este método define a los miembros de una población como posibles partículas a una solución. Este tipo de comportamiento

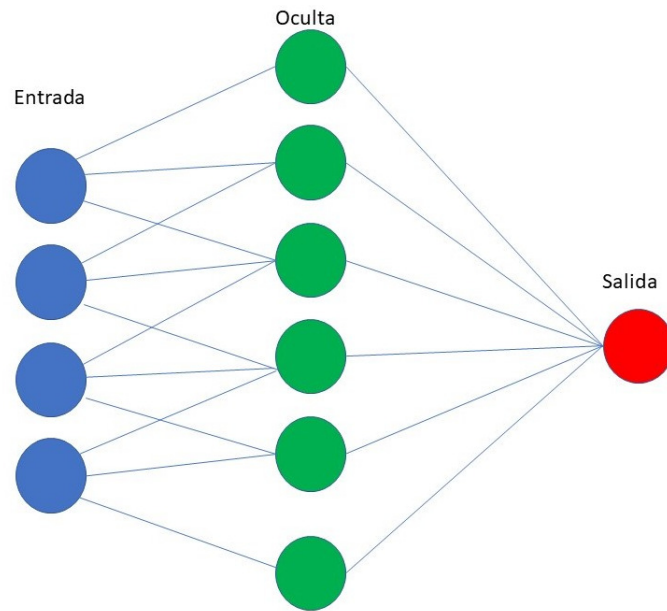


Figura 2.5: Ejemplo de estructura de red neuronal.

del algoritmo encuentra el óptimo global debido al uso de partículas que se ven influenciadas por el comportamiento cognitivo del enjambre. En PSO las partículas se definen en función de su posición y su velocidad en el espacio de búsqueda. Las partículas se atraen hacia posiciones en el espacio de búsqueda que representan su mejor hallazgo personal y el mejor hallazgo del enjambre. La formulación de la velocidad de las partículas utiliza dos coeficientes de aceleración para controlar la influencia de mejores resultados personales y resultados del enjambre sobre la nueva decisión. Además, un coeficiente llamado el peso de inercia se utiliza para controlar el impacto de la última velocidad de la nueva versión actualizada.

2.1.2.3 COLONIA DE HORMIGAS.

El método de optimización de colonia de hormigas por su eficiencia de búsqueda ha sido utilizado para aplicaciones de planeación de trayectoria para tareas de reconocimiento de VANTs [6]. Estos algoritmos fueron introducidos por Marco y sus colegas [8], están inspirados en los comportamientos sociales de las hormigas y como trabajan de forma colectiva en búsqueda de alimento para su colonia, se basan en encontrar un óptimo en el espacio de búsqueda como se observa en la Figura 2.6. Como en la naturaleza las hormigas usan una sustancia química llamada feromona para marcar las rutas tomadas, esto les ayuda a seguir el camino a las demás hormigas. La calidad de un camino se evaluaría en función de la cantidad de feromonas dejadas por las hormigas que pasaron a partir de esa ruta utilizando factores como concentración y proporción. Las hormigas eligen las rutas con mayor probabilidad de proporción a la concentración de la feromona, parecidos con otros métodos heurísticos se basan en el azar y la diversidad de una población inicial de hormigas, la evolución optima de la trayectoria se basa en la evaluación de la cantidad de feromonas depositadas por las hormigas, entre más hormigas dejen sus feromonas mejor es el camino.

2.2 ALGORITMOS GENÉTICOS

Los algoritmos genéticos se pueden categorizar como métodos heurísticos adaptativos de optimización, este tipo de algoritmos esta inspirados en los procesos naturales de reproducción sexual y en el principio de supervivencia de especies, donde los individuos mejor adaptados a su entorno natural son los que tendrán mayor probabilidad de transmitir sus genes. El primero en introducir los algoritmos genéticos fue Holland en 1975 [23] y después fueron publicados por primera vez por uno de sus estudiantes Goldberg en el año de 1989 [16]. Esta técnica utiliza una población de individuos generalmente aleatorios y binarios, donde en esta población se representan la información o características de los individuos. Los algoritmos genéticos al ser un método

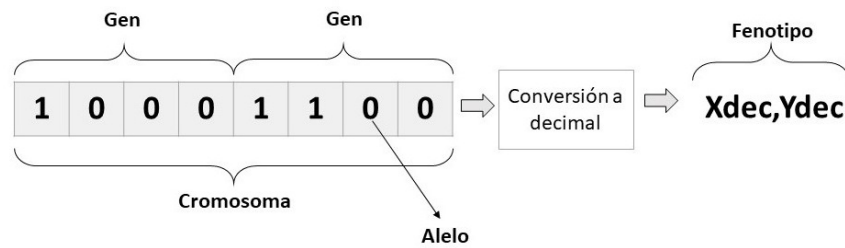


Figura 2.7: Tipos de parámetros en el algoritmo genético.

Una analogía computacional entre el algoritmo genético y el proceso naturaleza se puede observar en la Figura 2.8 primero tenemos en una primera instancia un entorno con individuos representados por los círculos grises, en computación es equivalente a un problema con muchas soluciones posibles, donde los individuos mejor adaptados a su entorno natural tendrán mejores posibilidades de sobrevivir y reproducirse, en computación esto es equivalente a un criterio de selección con base en las restricciones de la aplicación, donde los individuos más fuertes, en este caso podemos decir que son los círculos azules, tendrán más descendencia, mientras que los individuos más débiles, que en este caso serán los círculos verdes, quedarán obsoletos. En la naturaleza a este proceso se conoce como reproducción sexual, en el algoritmo genético este proceso se llama crossover y se encarga mediante la combinación de las cadenas de bits de crear nuevas soluciones al problema para tener una mejor búsqueda local. Finalmente, como en la naturaleza, de vez en cuando un individuo muta trayendo más diversidad a la especie, en el algoritmo genético tener una mutación nos ayuda a no estancarnos en un mínimo local y así llegar a tener la mejor solución posible, después este proceso se repite un gran número de veces, hasta que una descendencia que representa lo mejor de las viejas generaciones es quien sobrevive. Esto en el algoritmo genético equivale a encontrar una solución que converja y esta solución cumpla de manera eficiente los criterios del problema.

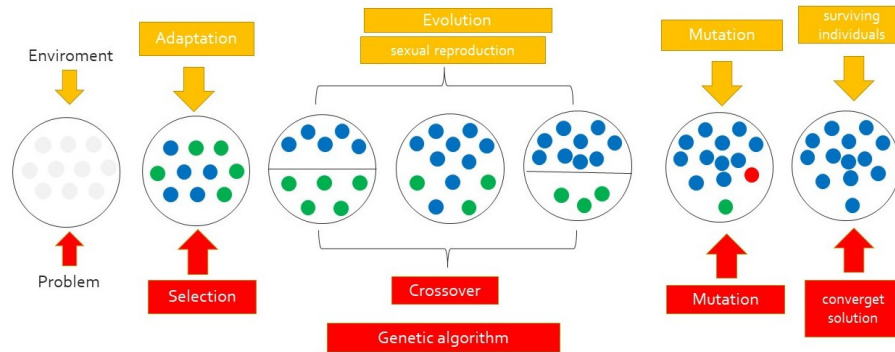


Figura 2.8: Analogía entre naturaleza y computación evolutiva.

2.2.1 FUNCIÓN OBJETIVO.

Debido a la naturaleza de múltiples variables los algoritmos genéticos trabajan con funciones objetivo, esta función objetivo es lo que se conoce en otros métodos como condiciones de frontera. Haciendo una analogía con la naturaleza se puede entender como un criterio de adaptación natural donde los individuos que mejor se adaptan a su entorno natural, son los que tendrán más probabilidad de reproducirse y transmitir sus genes. Se puede decir que el núcleo del algoritmo genético es la función objetivo ya que es la encargada de interpretar los individuos en función de su aplicación y dar a cada uno un valor traducible a la calidad de la solución, lo único malo de trabajar con funciones objetivos y con tantas variables como trabajan los algoritmos genéticos es que pueden tener muchos mínimos locales o que el mínimo global sea muy aislado, otra cosa que se debe considerar cuando se trabaja con funciones objetivos es cuidar la relación de aleatoriedad del algoritmo y la velocidad de convergencia se deben cuidar estos aspectos ya que de una forma con mucha aleatoriedad se puede nunca llegar a converger en el mínimo global o se puede tener una convergencia prematura, cualquiera de los dos casos nunca es bueno.

A continuación se pueden definir 4 tipos de formas de utilizar las funciones objetivo.

Valor aptitud puro: En este tipo de interpretación de la función objetivo, se utiliza el mismo valor de ajuste basado en la terminología natural del propio problema, es decir sin ninguna estandarización o normalización, por ejemplo, en planeación de trayectoria puede ser tal cual la distancia de la trayectoria.

Valor aptitud estandarizado: Este tipo de soluciones es utilizada para los problemas de maximización y minimización, donde en el caso de minimización se utiliza el valor aptitud puro mientras que para maximización se resta de su valor superior, como se observa ec (2.1).

$$fvalue_s = \begin{cases} fvalue_p, & \text{minimizacion} \\ fvalue_{max} - fvalue_p, & \text{maximizacion} \end{cases} \quad (2.1)$$

Valor aptitud ajustado: Este es utilizado agregando una la siguiente transformación al valor aptitud estandarizado ec (2.2). De esta manera, el valor aptitud ajustado siempre tiene valores de intervalo entre $[0,1]$, cuando más se aproxime el valor aptitud ajustado de un individuo a 1 mayor será su bondad.

$$fvalue_a = \frac{1}{1 + fvalue_s} \quad (2.2)$$

Valor aptitud normalizado: El valor aptitud normalizado introduce un nuevo aspecto, esto indica la bondad de una solución con respecto al resto de soluciones representadas en la población. Donde al igual que el valor aptitud ajustado, siempre tomará valores del intervalo $[0,1]$, con mejores individuos cuanto más próximo este a la unidad.

2.2.2 OPERADORES GENÉTICOS.

Se puede decir que los operadores genéticos son la estructura del algoritmo, estos son los encargados de mantener la diversidad genética en la población esta es la

función de crear nuevos individuos y así evitar el estancamiento de las soluciones, además de hacer que el algoritmo evolucione y se adapte a la mejor solución. Los operadores genéticos más comunes son selección, cruce y mutación, estos suelen trabajar mutuamente y están relacionados entre sí.

2.2.2.1 OPERADOR GENÉTICO DE SELECCIÓN.

El operador genético de selección es el encargado de elegir a los individuos que pasaran a reproducirse y cuáles no, se puede decir que como en la naturaleza existe un criterio que selecciona a las especies a través de las condiciones del medio ambiente, así el operador genético de selección basándose en criterios de la aplicación selecciona a los individuos genéticos. El operador genético de selección se puede dividir en dos criterios, en criterios determinísticos y probabilísticos ambos criterios, aunque funcionan de manera diferente su objetivo es el mismo seleccionar la mayor cantidad de veces los mejores individuos. Los criterios determinísticos son basados en su valor aptitud de los individuos, en función de su valor son fundamentadas las operaciones para selección del número de veces que se elige un individuo, por otro lado, en los criterios probabilísticos como su nombre dice son escogidos bajo criterios aleatorios, esto a su vez tiene ventajas y desventajas, como que evita la caída en el mínimo local, pero tiene la desventaja que tiene mayores tiempos de convergencia. Entre los métodos probabilísticos más utilizados se encuentran los criterios de selección por ruleta, selección por torneo que son de los métodos más populares, la Figura 2.9 muestra el funcionamiento general de los criterios, que como se aprecia en la Figura se tiene una tabla con individuos y su valor aptitud, en una primera instancia como el criterio de tipo ruleta este asigna un porcentaje de ser seleccionados a los individuos basándose en su valor aptitud, por otro lado el método de torneo hace competir dos individuos y basándose en su valor determina un ganador, que será escogido para reproducirse y por último el método determinístico ordena de menor a mayor los individuos basándose en su valor aptitud y escoge la mejor parte de la

población. Es importante también aclarar que la literatura menciona que no es bueno eliminar los individuos de los padres, sino que es bueno dejar algo de los individuos de las poblaciones pasadas, para que la población no se vuelva homogénea en pocas iteraciones [22].

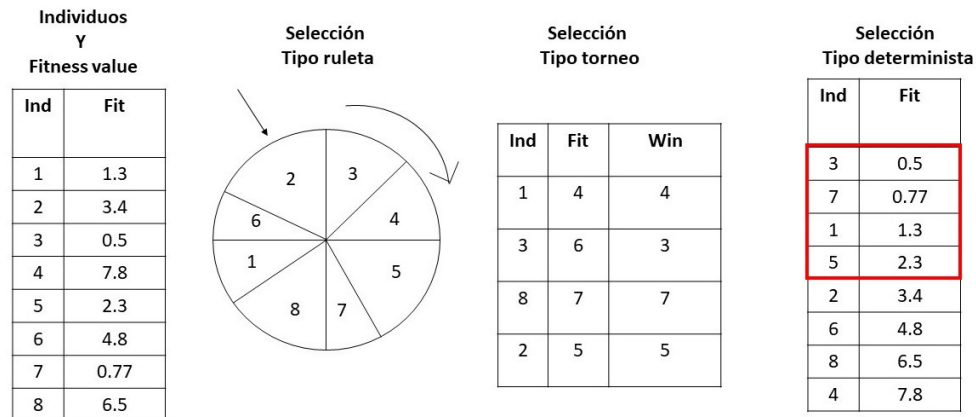


Figura 2.9: Tipos de criterios de Selección.

2.2.2.2 OPERADOR GENÉTICO DE CRUCE.

Una vez que el operador genético de selección elige los individuos más aptos, estos pasan al siguiente operador que es el operador de cruce, en la naturaleza esto es equivalente a la reproducción sexual donde los padres mejor adaptados a su ambiente o los más fuertes se encargan de transmitir sus genes a las siguientes generaciones. Su importancia en el algoritmo es vital ya que este es el operador encargado de hacer combinaciones para producir nuevos individuos o lo que se conoce en el algoritmo genético como descendientes. Los porcentajes de cruce son importantes ya que algunos autores tienen porcentajes altos esto para aumentar la diversidad de soluciones mientras que otros autores prefieren trabajar con porcentajes bajos [22]. El operador de cruce puede trabajar con diferentes criterios, estos pueden ser mediante una

estrategia destructiva o una estrategia no destructiva. Una estrategia destructiva trabaja con una población temporal, donde, aunque los padres pudieran tener un mejor valor aptitud los descendientes se insertaran en la población temporal, por otro lado, con una estrategia no destructiva la descendencia de los padres pasara a la siguiente generación solamente si cumple con un criterio de ajuste. La idea básica del operador de cruce se basa en tomar dos individuos y combinar los genes o bits si se trabaja en binario. Existen diferentes criterios de cruce entre los más conocidos están el cruce de un punto único, el cruce de 2 puntos y cruce de punto uniforme como se puede ver en la Figura 2.10, estos criterios funcionan con puntos de cruce donde este puede variar aleatoriamente o se pueden tener más de un punto de cruce, los algoritmos genéticos también tiene un proceso de elitismo que protege el cromosoma de mejor rendimiento, la solución más apta encontrada, en esta configuración no existe diversidad pero si rápida convergencia [27].

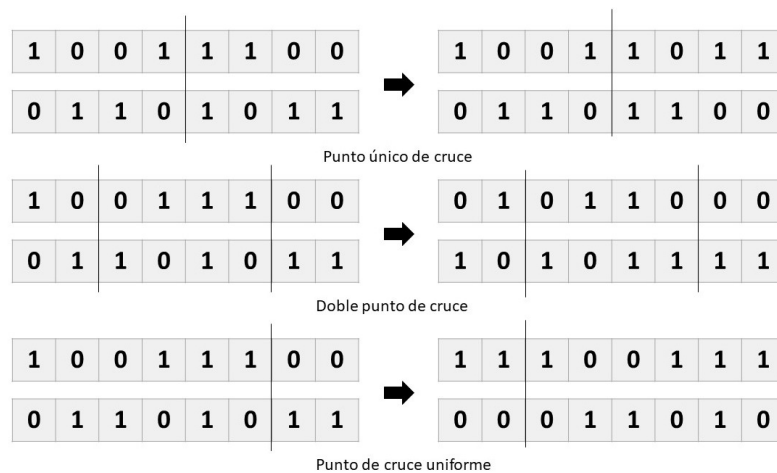


Figura 2.10: Criterio de cruce.

2.2.2.3 OPERADOR GENÉTICO DE MUTACIÓN.

El último operador genético es el operador de mutación, este ayuda a la diversidad de la población y evita al estancamiento de la población es decir caer en un mínimo local, aunque se puede seleccionar los individuos directamente en la población, es mutada aleatoriamente como se observa en Figura 2.11 este operador altera la población antes de remplazarse por una nueva población. Se puede decir que mientras que el operador de cruce se encarga de hacer una búsqueda global en todo instante, el operador de mutación funciona mejor conforme la población va convergiendo. Algunos investigadores [17] han mostrado los beneficios de la mutación sobre otros métodos solo con selección y cruce.



Figura 2.11: Criterio de cruce.

CAPÍTULO 3

ALGORITMO GENÉTICO PARA PLANEACIÓN DE TRAYECTORIA.

A continuación, se describirá la metodología desarrollada para diseñar el algoritmo genético, desde los criterios utilizados para el desarrollo de la función multi-objetivo, la cual se basó en tres criterios llevados secuencialmente. También se describe el diseño de los operadores genéticos y se describe los criterios de porcentajes utilizados, además que se describe la lógica de porque se optó por un algoritmo generacional con un criterio determinístico para los operadores genéticos un diagrama de flujo de la metodología se observa en la figura 3.1.



Figura 3.1: Diagrama de flujo de metodología.

3.1 INICIALIZACIÓN ALGORITMO GENÉTICO.

Uno de los aspectos más esenciales del algoritmo genético como se discutió en capítulos pasados es la inicialización, Para este caso el algoritmo fue establecido con una población inicial aleatoria de 24 cromosomas y 500 individuos para el entorno estático, para entornos dinámicos solo se utilizaron 100 individuos esto para reducir el costo computacional en entorno dinámico, sin embargo aunque se redujo la cantidad de individuos no se pierde mucha resolución de búsqueda debido a los criterios de alta mutación que se presentan más adelante. Los cromosomas y los individuos se pueden colocar en un arreglo matricial como se observa en (3.1), esto para tener una mejor manipulación de la información con la que trabajara el algoritmo, donde $a \in \mathcal{A}$ y \mathcal{A} está formado por el conjunto binario $\{0, 1\}$, es decir $\mathcal{A} = \{0, 1\}$.

$$Po_i = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,24} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,24} \\ \vdots & \vdots & \ddots & \vdots \\ a_{500,1} & a_{500,2} & \cdots & a_{500,24} \end{pmatrix} \quad (3.1)$$

Después del proceso de inicialización es necesario un proceso de codificación que transforme los datos binarios a un valor traducible a la aplicación, en este caso nosotros trabajamos con submatrices, donde cada submatriz contiene la información de las coordenadas en el espacio para cada eje, sub_1 contiene la información de la coordenada X en el espacio, sub_2 contiene la información de coordinar Y en el espacio y sub_3 la información de la coordenada Z espacio, como se observa en (3.2).

$$Po_i = \left([sub_1] \quad [sub_2] \quad [sub_3] \right) \quad (3.2)$$

$$[sub_1] = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,8} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,8} \\ \vdots & \vdots & \ddots & \vdots \\ a_{500,1} & a_{500,2} & \cdots & a_{500,8} \end{pmatrix} \quad (3.3)$$

$$[sub_2] = \begin{pmatrix} a_{1,9} & a_{1,10} & \cdots & a_{1,16} \\ a_{2,9} & a_{2,10} & \cdots & a_{2,16} \\ \vdots & \vdots & \ddots & \vdots \\ a_{500,9} & a_{500,10} & \cdots & a_{500,16} \end{pmatrix} \quad (3.4)$$

$$[sub_3] = \begin{pmatrix} a_{1,17} & a_{1,18} & \cdots & a_{1,24} \\ a_{2,17} & a_{2,18} & \cdots & a_{2,24} \\ \vdots & \vdots & \ddots & \vdots \\ a_{500,17} & a_{500,18} & \cdots & a_{500,24} \end{pmatrix} \quad (3.5)$$

Esto significa que cada valor de coordenada tiene un correspondiente a 8 bits (3.6), donde al hacer una conversión de binario a decimal cuando $\{a_{1,1} \dots a_{1,8} = 0\}$ significa que $x_{dec} = 0$ es decir el mínimo valor es 0, mientras que cuando $\{a_{1,1} \dots a_{1,8} = 1\}$ significa que el máximo valor que puede llegar a tener una cadena es de $x_{dec} = 255$ (3.7), por último se traduce a un valor de espacio real en este caso se supone un espacio tridimensional de 15 X 15 X 15, para esto se divide la matriz con los valores binarios entre un factor de conversión de 17 (3.8).

$$x_b = [a_{1,1}, a_{1,2}, a_{1,3}, a_{1,4}, a_{1,5}, a_{1,6}, a_{1,7}, a_{1,8}] \quad (3.6)$$

$$x_{dec11} \in B, B = \{0 \dots 255\} \quad (3.7)$$

$$P_{O_{real}} = \left([x_{dec}]/17 \quad [y_{dec}]/17 \quad [z_{dec}]/17 \right) \quad (3.8)$$

Debido al rango de 0 a 255 de la cadena de 8 bits, tenemos una resolución de espacio milésima, sin embargo, la resolución está determinada por la longitud de la cadena de bits o número de cromosomas y el número de individuos. Esto significa que, aunque el algoritmo tiene un muestreo espacial muy fino, todavía es limitado y se necesitan operadores genéticos para generar nuevos individuos.

3.2 DISEÑO DE FUNCIÓN MULTI-OBJETIVO.

La función objetivo es la responsable de determinar el valor de aptitud de cada individuo y asignar un valor que refleje su calidad para resolver el problema, en este caso los individuos representan las coordenadas en el espacio del punto de navegación y dependiendo si el punto de navegación sirve para generar una trayectoria libre de colisiones y de mínimo consumo de energía la trayectoria tendrá un valor aptitud pequeño si por otro lado la trayectoria es muy larga y cruza varias veces obstáculos la trayectoria tendrá un valor aptitud muy elevado. Se utilizó una función multi-objetivo con tres criterios; el criterio de longitud de arco para determinar la distancias de la trayectoria, el criterio de distancia entre puntos de navegación a obstáculos para determinar los puntos de navegación que están dentro de una zona prohibida de un obstáculo y así eliminar de manera parcial los puntos de navegación indeseables y el ultimo criterio para determinar el cruce de trayectoria con obstáculos el cual a través de la suma de ángulos determina si la ruta pasa por un obstáculo.

3.2.1 LONGITUD DE ARCO.

El primer punto de la función multi-objetivo es una función en estado puro donde se utiliza la distancia de la trayectoria como criterio. La distancia se calcula entre el punto inicial el punto de navegación o WP1 y el punto final como se muestra en la Figura 3.1, se evalúan todas las trayectorias realizadas, por lo que la trayectoria

que tiene la distancia más larga también tiene el valor de aptitud más grande y por lo tanto menos probabilidades de reproducirse. En la Tabla 3.1 se puede ver el pseudocódigo del algoritmo 1 correspondiente a la función objetivo, las ec (3.9) y ec (3.10) se utilizan para calcular la distancia entre el punto de partida, punto de navegación y el punto final.

Algoritmo 1: Longitud de arco.

```

Iniciar Variable P correspondiente al tamaño de población
Iniciar contador  $h = 1$ ;
mientras  $h \leq P$  hacer
    Calcular la distancia  $f_1$  de punto de inicio
    al punto de navegación mediante Ec (3.9);
    Calcular la distancia  $f_2$  del punto de navegación
    al punto final mediante Ec (3.10);
     $h = h + 1$ ;
end
    
```

Tabla 3.1: Función objetivo 1

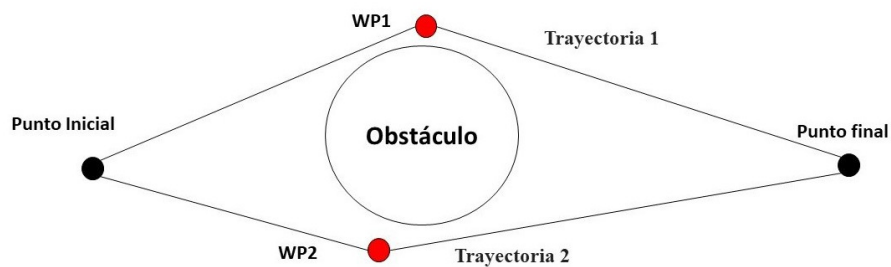


Figura 3.2: Evaluación de largo de trayectorias.

$$f_1(h) = \sqrt{(x_i - x_1(h))^2 + (y_i - y_1(h))^2 + (z_i - z_1(h))^2} \quad (3.9)$$

$$f_2(h) = \sqrt{(x_1(h) - x_f)^2 + (y_1(h) - y_f)^2 + (z_1 - z_f(h))^2} \quad (3.10)$$

3.2.2 DISTANCIA ENTRE PUNTOS DE NAVEGACIÓN A OBSTÁCULOS.

En la evasión de obstáculos entendemos como trayectoria inviable un camino que atraviesa un obstáculo, sin embargo antes de diseñar un criterio para evitar el cruce de la trayectoria por una zona prohibida, se diseña una función objetivo que penaliza el camino donde se encuentran los puntos de navegación adentro de un obstáculo, esto con el fin de reducir el tiempo de procesamiento computacional y evitar que los operadores genéticos pierdan tiempo haciendo cálculos en iteraciones posteriores, lo cual ayuda a tener una convergencia rápida y eficiente. El segundo criterio de la función multi-objetivo se muestra en la Tabla 3.2, este criterio es el responsable de eliminar los puntos de navegación inviables, para esto se calcula la distancia entre el punto de navegación y el obstáculo como se observa en la Figura 3.3, esto se hace para todos los puntos de navegación contra todos los obstáculos. Si la distancia es menor o igual al radio de los obstáculos, significa que los puntos de navegación están dentro del obstáculo y, por lo tanto, se agrega una penalización a su valor aptitud. Cabe mencionar que a diferencia del criterio anterior este se puede decir que es discreto; es decir o cumple o no cumple, o tiene penalización o no tiene penalización y la penalización no es la distancia, sino que es una penalización constante.

$$do_{x,y}(h, m) = \sqrt{(x_1(h) - x_o(m))^2 + (y_1(h) - y_o(m))^2} \quad (3.11)$$

$$do_{x,z}(h, m) = \sqrt{(x_1(h) - x_o(m))^2 + (z_1(h) - z_o(m))^2} \quad (3.12)$$

$$do_{y,z}(h, m) = \sqrt{(y_1(h) - y_o(m))^2 + (z_1(h) - z_o(m))^2} \quad (3.13)$$

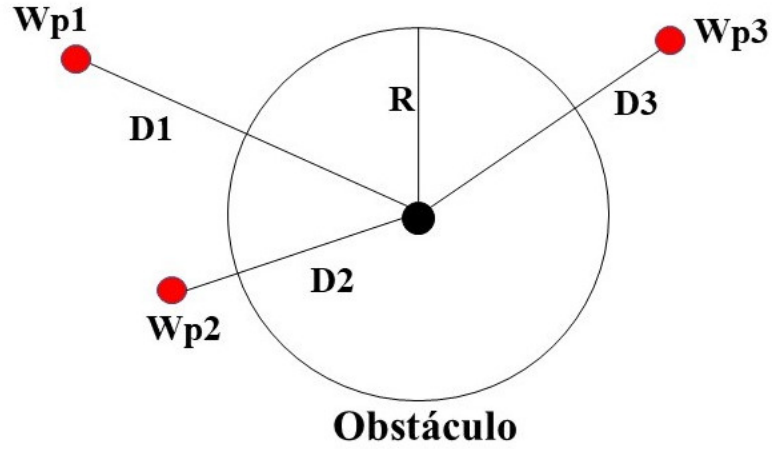


Figura 3.3: Evaluación de puntos de navegación.

$$f(h) = \begin{cases} f(h) + p, & do_{x,y}(h, m) \text{ y } do_{x,z}(h, m) \text{ y } do_{y,z}(h, m) \leq radius \\ f(h), & do_{x,y}(h, m) \text{ y } do_{x,z}(h, m) \text{ y } do_{y,z}(h, m) > radius, \end{cases} \quad (3.14)$$

El proceso anteriormente descrito es para 2D en un solo plano, para los cálculos en 3D se optó por una estrategia de planos bidimensionales, de esta manera se utilizó esta estrategia para que los cálculos no fueran robustos y así poder reducir el costo y el tiempo computacionales. La estrategia de planos bidimensionales utiliza las ec (3.11) - (3.13), para calcular la distancia del obstáculo en los diferentes planos. Si se cumplen las tres condiciones se agrega una penalización constante como se muestra en la ec (3.14), ya que esto significa que el punto de navegación está dentro de la zona prohibida. En este caso en el entorno 3D las zonas prohibidas son esferas.

Algoritmo 2:Distancia entre puntos y obstáculos.

```

iniciar contador  $m = 1$ ;
iniciar variable  $numobs$  como numero de obstáculos;
mientras  $m \leq numobs$  hacer
    Calcular la distancia de punto de navegación al obstaculo
    en los diferentes planos mediante Ec (3.11) - (3.13) ;
    si  $do_{x,y}(h, m)$  y  $do_{x,z}(h, m)$  y  $do_{y,z}(h, m) \leq radius$ 
        añade penalización mediante Ec (3.14);
    si no
        no añadas penalización
    end;
 $m = m + 1$ ;

```

Tabla 3.2: Función objetivo 2

3.2.3 CRUCE DE RUTAS CON OBSTÁCULOS.

Aunque en el criterio pasado ya se calculó si el algoritmo coloca un punto navegación dentro del radio de un obstáculo aún puede poner un punto de navegación en una zona en la que eventualmente la trayectoria cruzara un obstáculo, para esto se optó por una estrategia de suma de ángulos de esta manera se puede decir que esta operación se determina de una forma continua a diferencia de otros métodos que tienen que discretizar la trayectoria. El criterio final se muestra en la Tabla 3.3 este determina si el camino cruza o no un obstáculo; para esto se calcula el ángulo entre los vectores virtuales de los puntos tangentes del obstáculo con el punto de inicio o con el punto de navegación, también se obtiene el ángulo que se forma con la trayectoria y el vector virtual del punto tangente de la izquierda y también con la trayectoria y el vector virtual de la derecha como se observa en la Figura 3.4. Se calcula el ángulo entre los puntos tangentes con la Ec (3.15) este ángulo es nombrado "*anguloDI*". Luego, siguiendo la misma lógica, se calcula el ángulo entre la trayectoria y el vector

virtual de la derecha mediante la Ec (3.16) a este ángulo se le nombro "*anguloLI*" y por último se calcula el ángulo con el vector virtual de la izquierda mediante (3.17) este angulo se le nombro "*anguloLD*".

Algorithm 3:Cruce de trayectoria por obstáculos.

```

Iniciar contador  $j = 1$ ;
Iniciar variable numway como número de puntos de navegación;
Iniciar contador  $f = 1$ ;
mientras  $j \leq numway + 1$  hacer
    mientras  $f \leq P$  hacer
        obtener  $\alpha_{DI}(f)$  mediante ec. (3.15);
        obtener  $\alpha_{LI}(f)$  mediante ec.(3.16);
        obtener  $\alpha_{LD}(f)$  medainte eq.(3.17);
        AngleSum =angleLD+angleLI;
        si  $angleDI = AngleSum$ 
            peso(h)=1;
        si no
            peso(h)=0;
        end
    end
     $j = j + 1$ 
end

```

Tabla 3.3: Función objetivo 3.

$$\alpha_{DI}(f) = \arccos \frac{\vec{U} \cdot \vec{V}}{|\vec{U}| |\vec{V}|} \quad (3.15)$$

$$\alpha_{LI}(f) = \arccos \frac{\vec{U} \cdot \vec{W}}{|\vec{U}| |\vec{W}|} \quad (3.16)$$

$$\alpha_{LD}(f) = \arccos \frac{\vec{W} \cdot \vec{V}}{|\vec{W}| |\vec{V}|} \quad (3.17)$$

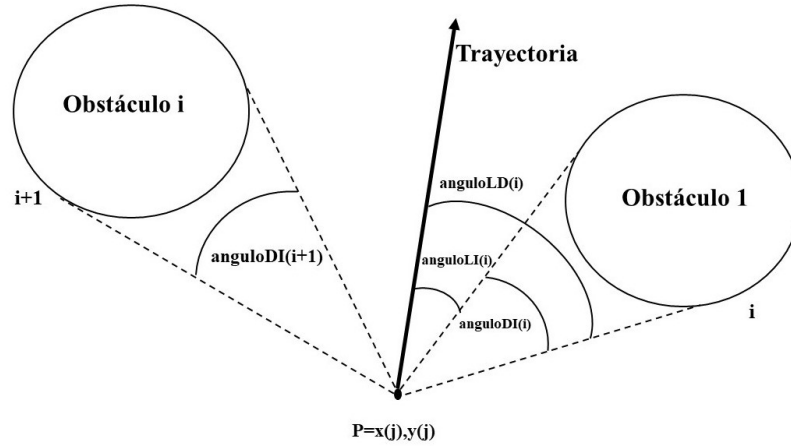


Figura 3.4: Evaluación de cruce de trayectoria por obstáculos.

El proceso anteriormente descrito igual que con el criterio anterior es para un plano o para 2D esto para poder ahorrar tiempo de procesamiento y computo. Igualmente que con el criterio anterior se realiza este proceso en los tres plano x, y, x, z y y, z . Si una trayectoria cruza un obstáculo en una variable $peso_{x,y}$, $peso_{x,z}$, $peso_{y,z}$ se guarda el número de veces que una trayectoria cruza un obstáculo en cada plano, para que una trayectoria sea eficiente su valor de peso debe ser 0 y tener la mínima distancia de la trayectoria. Después mediante la ecuación (3.18) se multiplica por una constante los valores de peso y se suman a su valor aptitud.

$$f(h) = f(h) + peso_{x,y}(h) * 50 + peso_{x,z}(h) * 50 + peso_{y,z}(h) * 50 \quad (3.18)$$

3.3 DISEÑO DE OPERADORES GENÉTICOS.

Después de definir los criterios de la función multi-objetivo se diseñaron los parámetros de los operadores genéticos, estos fueron elegidos para tener una cuasi convergencia rápida sin caer en una solución ineficiente, esto para que el algoritmo genético pueda tomar decisiones rápidamente ante cambios repentinos los cual será necesario en el ambiente dinámico. Para esto se utilizó una estrategia de selección determinista en lugar de una aleatoria, en la cual si bien tenemos una rápida convergencia no caemos en un mínimo local debido a la amplia resolución de muestreo espacial mencionada anteriormente y además gracias a los altos porcentajes de cruce y mutación que aumentaron aún más la resolución del muestreo espacial. Debido a tener un alto porcentaje de alteración en la población, tuvimos que trabajar con lo que hemos denominado poblaciones de convergencia, que es una parte de la población que no se ven afectadas por la mutación ni por el operador de cruce, esto para garantizar la convergencia con criterios de mutación y cruzamiento muy elevados.

3.3.1 MÉTODO DE SELECCIÓN

El primer operador genético después de ser evaluados los individuos por la función objetivo es el operador de selección, esto puede variar dependiendo del diseñador o el programador, algunos autores suelen primero poner los operadores genéticos y después la función objetivo o incluso algunos suelen poner dos criterios de evaluación. El operador genético de selección cuya tarea es como su nombre lo dice seleccionar o escoger basándose en el valor aptitud de la población los individuos que cumplan las restricciones previamente establecidas, que en este caso es aquel cuyo vector valor de cruce sea 0 y su longitud de trayectoria sea la más corta. Para este caso se utilizó un operador de selección natural con criterios deterministas; esto permite, dependiendo de su valor aptitud, ordenar a los individuos de menor a mayor. Los puntos de navegación con menor valor de aptitud son elegidos para reproducirse, por

ejemplo en la Figura 3.5, en el paso 1 se tienen las coordenadas X,Y,Z de los puntos de navegación con su valor correspondiente de valor de aptitud, en el paso 2 se acomodan los valores de las coordenadas basándose en su valor aptitud de menor a mayor, en un paso 3 se seleccionan solo los individuos que cumplen con los requisitos, que en este caso son los de menor valor aptitud o la mitad de la población con el valor más pequeño son seleccionados, los individuos no seleccionados o los que son menos aptos son desechados, después en el paso 4 los individuos seleccionados son duplicados. De esta manera los individuos mejor adaptados al problema son preservados conforme avanzan las iteraciones, así es como funciona un criterio determinista, también cabe destacar que como se mencionó antes es bueno no reemplazar toda la población así que solo se selecciona la mitad de la población, la población seleccionada se duplica y reemplaza a la población obsoleta, de esta forma cuando la población seleccionada pase al siguiente criterio de cruce y se inserten en la población, los descendientes y los padres estarán en una misma generación.

No	x	y	z	Fitness value
1	5.11	2.88	3.02	22.72
2	1.82	5.41	3.7	24.62
3	4.11	13.52	6.05	25.14
4	1.94	8.76	8.92	25.37
5	14.35	0.058	3.69	33.25
6	12.88	14.88	13.12	25.08
7	6.70	4	5.66	21.34
8	3.47	8.64	2.85	342.99
9	1.64	4.11	1.03	26.77
10	0.11	12.4	10.10	30.8

No	x	y	z	Fitness value
7	6.70	4	5.66	21.34
1	5.11	2.88	3.02	22.72
2	1.82	5.41	3.7	24.62
6	12.88	14.88	13.12	25.08
3	4.11	13.52	6.05	25.14
4	1.94	8.76	8.92	25.37
9	1.64	4.11	1.03	26.77
10	0.11	12.4	10.10	30.8
5	14.35	0.058	3.69	33.25
8	3.47	8.64	2.85	342.99

No	x	y	z	Fitness value
7	6.70	4	5.66	21.34
1	5.11	2.88	3.02	22.72
2	1.82	5.41	3.7	24.62
6	12.88	14.88	13.12	25.08
3	4.11	13.52	6.05	25.14
7	6.70	4	5.66	21.34
1	5.11	2.88	3.02	22.72
2	1.82	5.41	3.7	24.62
6	12.88	14.88	13.12	25.08
3	4.11	13.52	6.05	25.14

Figura 3.5: Operador genético de selección.

3.3.2 OPERADOR DE CRUCE.

Los individuos elegidos previamente por el método de selección determinista proceden a reproducirse. Este operador genético es responsable de aumentar la resolución

de búsqueda. Como mencionamos anteriormente, la resolución del espacio de búsqueda depende del tamaño de la población y del cromosoma y aunque puede ser muy fina, está limitada por las combinaciones de bits de la población inicial. El operador de cruce se encarga de combinar la información de los individuos previamente seleccionada y crear nuevas soluciones inexploradas. La descendencia para este caso se obtiene a partir del criterio de cruce de un punto único, como se muestra en la Figura 3.6 donde el punto cambia al azar y este proceso se realiza para todos los individuos seleccionados. El porcentaje de cruce utilizado en esta aplicación fue del 50 por ciento. Los descendientes se obtienen a partir del criterio de cruce de un solo punto, como se muestra en la figura donde el punto cambia al azar, este proceso se realiza para todos los individuos seleccionados. Según la literatura siempre es beneficioso mantener a algunos de los padres para que tengan mayor aleatoriedad y no caer en un mínimo local, en esto ayuda el criterio determinista que ya mantienen los padres mejor adaptados [22]. En la Figura 3.7 se muestra un caso de cómo funciona el operador de cruce, donde los puntos azules representan la localización de los obstáculos y los puntos rojos representan los puntos de navegación de las posibles trayectorias que evolucionan a través de las iteraciones. Se puede ver cómo afecta el operador de cruce en cada iteración creando nuevos individuos. Por ejemplo, en la Figura 3.7 a) esto corresponde a la iteración 1, los puntos de navegación se distribuyen aleatoriamente por todo el espacio sin ningún orden, en la Figura 3.7 b) que corresponden a 5 iteraciones, los puntos de navegación comienzan a concentrarse en dos áreas en la parte superior izquierda y la parte inferior central, luego en la Figura 3.7 c) y d) correspondientes a 5 y 8 iteraciones, predomina un área hasta que convergen en un solo punto en la Figura 3.7 f), se puede decir que el algoritmo aunque trabaja con los puntos de navegación, optimiza las áreas donde ubica el punto de navegación como se observa en la Figura 3.7 anterior, es decir entre más avanza el algoritmo la zona donde puede colocar el punto de navegación se hace infinitesimal, claro que esto solo es una forma de decir que el área se hace más pequeña hasta converger en un punto.

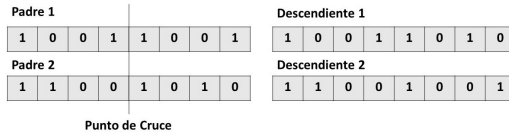


Figura 3.6: Operador genético de cruce.

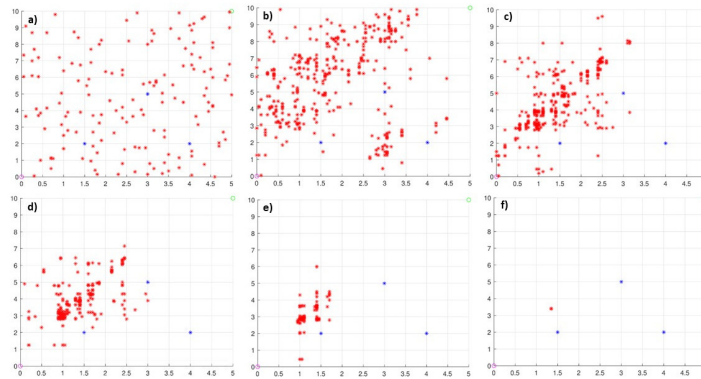


Figura 3.7: Evolución de proceso de cruce: a) iteración 1, b) iteración 5, c) iteración 8, d) iteración 10, e) iteración 14, f) iteración 29

3.3.3 OPERADOR DE MUTACIÓN.

La mutación ayuda preservar la diversidad de la población y evita la convergencia prematura. De la misma manera que el operador de cruce, la mutación se produce al azar alterando la cadena de bits en la matriz de población como se observa en la Figura 3.8, esto es reemplazando un 1 por un 0. Para calcular la operación de genes afectados por la mutación es una simple operación se multiplica $R * C$ donde R representa el número de individuos y C representa el número de cromosomas (3.19) y con la ecuación (3.20) calculamos el número de individuos que se verán afectados en cada iteración, para esto en entorno estático se utilizó una población de 500 individuos con una $prob = 0.2$, de esta manera se obtienen 5 individuos afectados

por la mutación, que para el caso de mutación en entorno estático 0.2 porciento por mutación es más que suficiente, para entorno dinámico se utilizó una población de 100 individuos un porcentaje de mutación de 0.2 que es equivalente a 5 individuos por iteración sin embargo el criterio de mutación para entorno dinámico tiene más criterios por lo que se considera un criterio de alta mutación.

$$Tgenes = R * C \tag{3.19}$$

$$Genesmut = Tgenes * prob/100 \tag{3.20}$$

$$Genesmut = (12000) * (0.2)/100 \approx 24 \tag{3.21}$$



Figura 3.8: Operador genético de mutación.

Para entorno dinámico uno de los problemas que se tenían que resolver fue el poder recalcular la trayectoria, esto debido a la convergencia prematura, ya que cuando el punto de navegación ha convergido y ocupa debido a un obstáculo recalculer la trayectoria esto no es posible debido a que no hay diversidad en la población. Para resolver este problema se utilizó una estrategia de alta mutación, primero para evitar estancarnos en el punto de navegación, cuando la distancia entre el punto

de inicio y el punto de navegación es menor a una distancia establecida, el algoritmo introduce una nueva población aleatoria, de esta manera el 25 por ciento de la población es creada de nuevo, esto ayuda a tener diversidad, sin embargo la alta mutación puede afectar y mover el mismo punto de navegación de convergencia y hacer un cambio brusco de dirección para esto se trabajó con una estrategia donde cierto por ciento de la población no será afectado por la mutación. A esta parte de la población le hemos llamado población de convergencia y por lo que se debe encontrar un equilibrio entre la relación de alta mutación y la población de convergencia. Para ambos casos con los que hemos trabajado de población de 100 individuos y 500 individuos, se han seleccionado los primeros 20 individuos de la población total como la población de convergencia, esto porque estos individuos ya son los mejores para cumplir la población, en caso de ya no ser útiles para la aplicación simplemente son reemplazados por los individuos de alta mutación que cumplen la función. Por último, aunque con alta mutación se tiene diversidad y con la población de convergencia evitas que el punto de navegación esté haciendo cambios bruscos y con esto perder la dirección, esto no garantiza que el punto de navegación converja al punto final. Para resolver el problema de convergencia al punto final se creó un último criterio de mutación dentro de la población de convergencia se mete una mutación, que es el punto final de esta manera si no existe una interacción con los obstáculos el objetivo directo será el punto final. El funcionamiento del nuevo operador de mutación se observa en las Tablas 3.4, 3.5, 3.6.

3.4 ALGORITMO PARA ENTORNO ESTÁTICO Y DINÁMICO.

Una de las ventajas del algoritmo genético de planeación trayectoria, es que funciona para dos estrategias en entorno estático y para entorno dinámico. Por entornos estáticos entendemos que son aquellos donde en la aplicación tiene un conocimiento completo de su entorno de trabajo además que los obstáculos no cambian en el tiem-

Algoritmo 4:Criterio de mutación 1.

```

Iniciar Variable  $R$  como número de individuos;
Iniciar Variable  $C$  como número de cromosomas ;
Iniciar constante  $p_{safe}$  como individuos;
de población de convergencia
   $Tgenes = R * C$ ;
   $GenesMut = Tgenes * prob/100$ 
  ciclo  $i = GenesMut$ 
     $fila = randi([1 R])$ 
     $column = randi([1 C])$ 
    si  $fila = < P_{safe}$ 
      hacer
    else
       $Pob(fila, column) = 1 - Pob(fila, column)$ ;
    final si
  final ciclo
  
```

Tabla 3.4: algoritmo de alta mutación fuera de la población de convergencia

po y la trayectoria se genera antes de comenzar el VANT a moverse. Los entornos dinámicos son aquellos de los cuales solo se tiene una información parcial del entorno de trabajo y los obstáculos pueden cambiar su posición en el tiempo, lo que significa que la trayectoria global puede cambiar. Un ejemplo para entorno estático se observa en la Fig 3.9, se tiene un punto de inicio, en este caso dos puntos de navegación utilizados para generar la trayectoria y un punto final, además de un entorno de obstáculos donde estos están fijos y la trayectoria no cambia conforme avanza las iteraciones. En la Figura 3.10, está el caso de cómo funciona el algoritmo para casos dinámicos y como el punto de inicio que representa la posición del VANT se mueve hacia el punto final conforme avanzan las iteraciones, además se puede observar que

Algoritmo 5:Criterio de mutación 2.

```

cin=1; re=mod(c,cin) para obtener el residuo ;
    si re=0
num1=stop.x/0.058823529;
num2=stop.y/0.058823529;
num3=stop.z/0.058823529;
nnum1=dec2bin(num1), nnum2=dec2bin(num2), nnum3=dec2bin(num3);
str1 =nnum1, str2 =nnum2, str3 =nnum3 ;
l1=length(str1), l2=length(str2), l3=length(str3);
resta1=8-l1, resta2=8-l2, resta3=8-l3;
    x1=zeros(1,8) para iniciar variable de 0 x1;
    x2=zeros(1,8)para iniciar variable de 0 x2 ;
    x3=zeros(1,8) para iniciar variable de 0 x3;
        ciclo k=l1
            x1(k+resta1)=str2num(str1(k));
        final ciclo
        ciclo k=l2
            x2(k+resta2)=str2num(str2(k));
        final ciclo
        ciclo k=l3
            x2(k+resta3)=str2num(str3(k));
        final ciclo
Pob(5,1:C)=[x1,x2,x3];
    final si

```

Tabla 3.5: Mutación para convergencia a punto final.

los obstáculos se mueven por lo que se tiene que recalculer la trayectoria dependiendo de la posición de los obstáculos.

Algoritmo 6: Criterio de mutación 3.

```

Iniciar Variable dist como distancia entre
punto inicial y punto de navegación
    si dist <= distr
        Pob(R/4+1:R,1:C) = round(rand(R/4,C));
    final si
    
```

Tabla 3.6: Mutación para generar nueva población aleatoria.

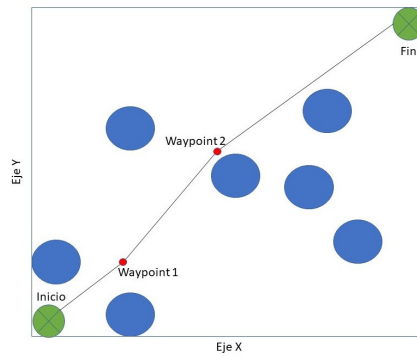


Figura 3.9: Algoritmo en entorno estático.

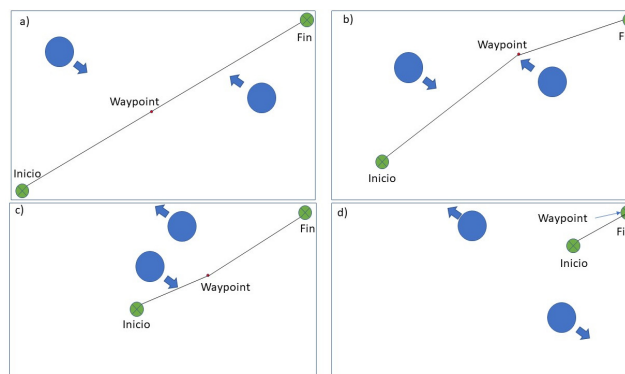


Figura 3.10: Algoritmo en entorno dinámico.

3.5 SUAVIZADO DE TRAYECTORIA.

Para el suavizado de la trayectoria se utilizó un filtro paso bajas, esto ya que el funcionamiento del filtro paso bajas tiene la característica de eliminar las altas frecuencias además de arrojar una respuesta lenta lo cual es útil para el dinamismo de la trayectoria, en la generación de la trayectoria lo que hará es eliminar esas pequeñas discontinuidades. Las funciones de transferencia, para un filtro de Butterworth de primer orden, se dan a continuación:

$$TL(S) = \frac{1}{S+1} \quad (3.22)$$

$$TH(S) = \frac{S}{S+1} \quad (3.23)$$

En donde, $TL(S)$ es la función de transferencia paso-bajo y $TH(S)$ es la función de transferencias paso-alto donde

$$TL(S) + TH(S) = 1 \quad (3.24)$$

CAPÍTULO 4

RESULTADOS Y EXPERIMENTOS.

Resultados y experimentos. Para la creación de las pruebas se trató de recrear entornos donde normalmente vuelan los VANTs donde los obstáculos tienden a disponerse en formas comunes. Se puede decir que las configuraciones más típicas de obstáculos son en formación de barreras, cerramiento de objetivos o también se conoce como tipo fortaleza, además se realizaron pruebas en entorno aleatorio y sin ningún orden para probar las capacidades y el más simple de los casos intersección directa con la trayectoria de vuelo [34], [42]. Para el apartado de los experimentos en entorno estático se realizó una implementación en tiempo real, las pruebas se ejecutaron en el Laboratorio de Navegación del Centro de Investigación e Innovación en Ingeniería Aeronáutica, de la FIME en la UANL. Todos los algoritmos se llevaron a cabo en una computadora compuesta por un procesador Core i7-8750H, con una GPU Nvidia GTX 1060, y 16 GB de memoria RAM DDR4 a 2400 MHz además de un sistema de captura de movimiento y un vehículo aéreo no tripulado de firmware modificado. Esto se puede observar en el siguiente link de YouTube <https://www.youtube.com/watch?v=iKbr3mcZawk> Figura 4.1. Pruebas en entorno estático en implementadas. El modelo implementado para el control se puede encontrar en el Apéndice A.

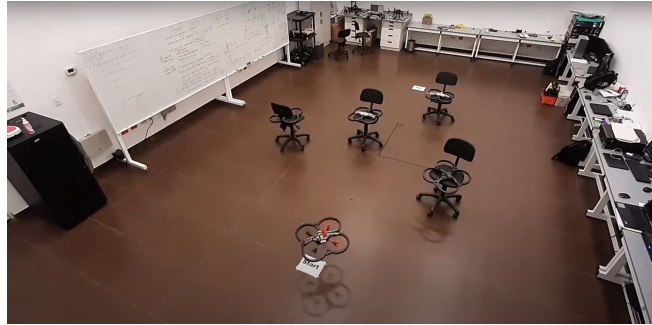


Figura 4.1: Pruebas en entorno estático implementadas.

4.1 ENTORNO ESTÁTICO.

Para validar las capacidades y el rendimiento del algoritmo general, se diseñan una serie de pruebas. Se proponen cuatro escenarios de prueba, donde se utilizan diferentes arreglos de obstáculos para evaluar la trayectoria generada; así, se concluye si el algoritmo es capaz de funcionar en un caso de la vida real. El número y la disposición de los obstáculos seleccionados se describe a continuación.

Caso A: Un solo obstáculo entre el punto inicial y el destino final.

Caso B: Dos obstáculos entre el punto inicial y el destino final.

Caso C: Tres obstáculos que encierran el punto de destino.

Caso D: Cuatro obstáculos colocados de forma aleatoria.

Caso E: Cuatro obstáculos en formación de doble barrera.

Además, como se mencionó antes se realizó un trabajo de implementación, para esto se utilizó una plataforma experimental compuesta por un sistema de captura de movimiento y un vehículo aéreo no tripulado. El VANT utilizado fue un tipo quadrotor modelo Parrot A.R. Drone 2.0 con firmware personalizado integrado, configurado para alcanzar velocidades de guiñada de hasta $100 \text{ grados}/s$, que en la práctica es la máxima velocidad de giro que puede tomar en una trayectoria cur-

va. En este sentido, se selecciona el tiempo de finalización de cada trayectoria para cumplir con estos criterios. Para el seguimiento de las coordenadas de las posiciones, un sistema de captura de movimiento que consta de 16 cámaras T-40 VICON, con una precisión de 0,1 mm con hasta 340 fps [13], para verificar la estrategia que se utilizó crear la trayectoria continua y el control utilizado se puede profundizar en la publicación [18].

4.1.1 CASO A: UN OBSTÁCULO ENTRE EL PUNTO INICIAL Y EL PUNTO FINAL.

Los resultados del algoritmo genético basado en evasión de obstáculos para el caso A se muestran en la Figura 4.2 donde la línea continua negra es la trayectoria genética y la línea naranja discontinua es la trayectoria continua, el punto naranja es el punto de navegación, el pequeño círculo naranja es la meta y el pequeño círculo azul en el punto de inicio, el asterisco morado es la localización del obstáculo, la línea punteada roja es la circunferencia del obstáculo o zona prohibida. Se puede observar que el punto de navegación generado se coloca de tal manera que la trayectoria es casi tangente al círculo que encierra la zona prohibida del obstáculo; por lo tanto, la trayectoria es la distancia mínima, se puede observar también una pequeña desviación en cuanto la localización del punto de navegación la cual es debido al reducido número de población que se utilizó en esta prueba.

4.1.2 CASO B: DOS OBSTÁCULOS ENTRE EL PUNTO INICIAL Y EL PUNTO FINAL.

Para esta prueba, se colocan dos obstáculos entre el punto inicial y el punto final como se observa en la Figura 4.3. Se puede observar que en este caso ya no hay una desviación en el punto de navegación, además que el punto de navegación for-

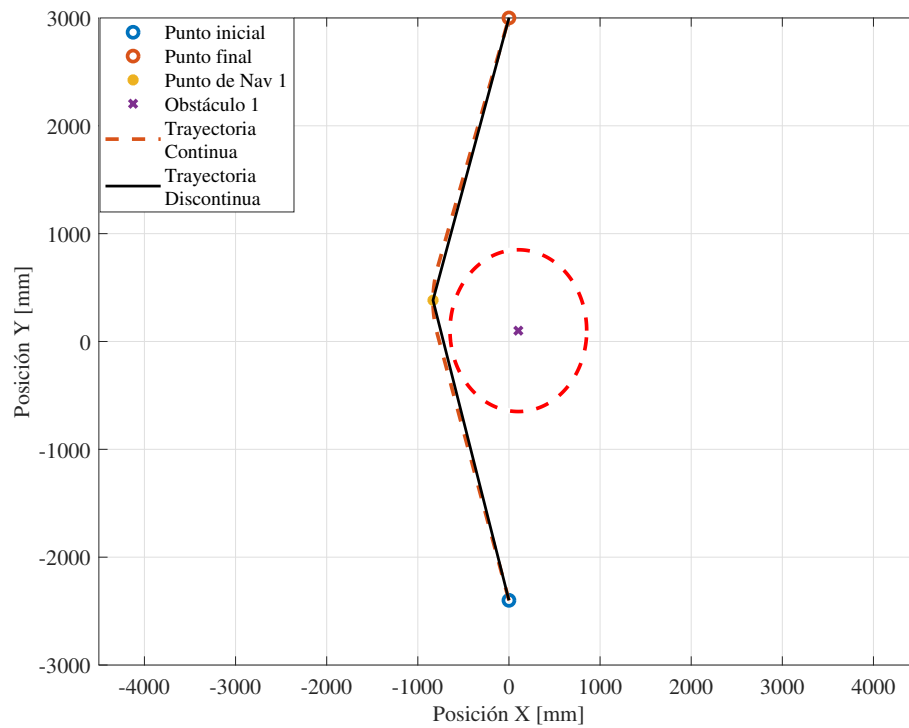


Figura 4.2: Trayectoria generada 2D para un obstáculo.

ma una trayectoria tangente a las dos circunferencias de las zonas prohibidas de los obstáculos, así demostrando la eficiencia de la trayectoria. Se demuestra que la trayectoria cumple las condiciones establecidas anteriormente en las funciones objetivo, estas son la trayectoria es la más corta posible y se observa claramente que se evita generar trayectorias que colisionen con los obstáculos. La línea negra representa la trayectoria genética, y la línea punteada naranja representa la trayectoria continúa realizada para las pruebas.

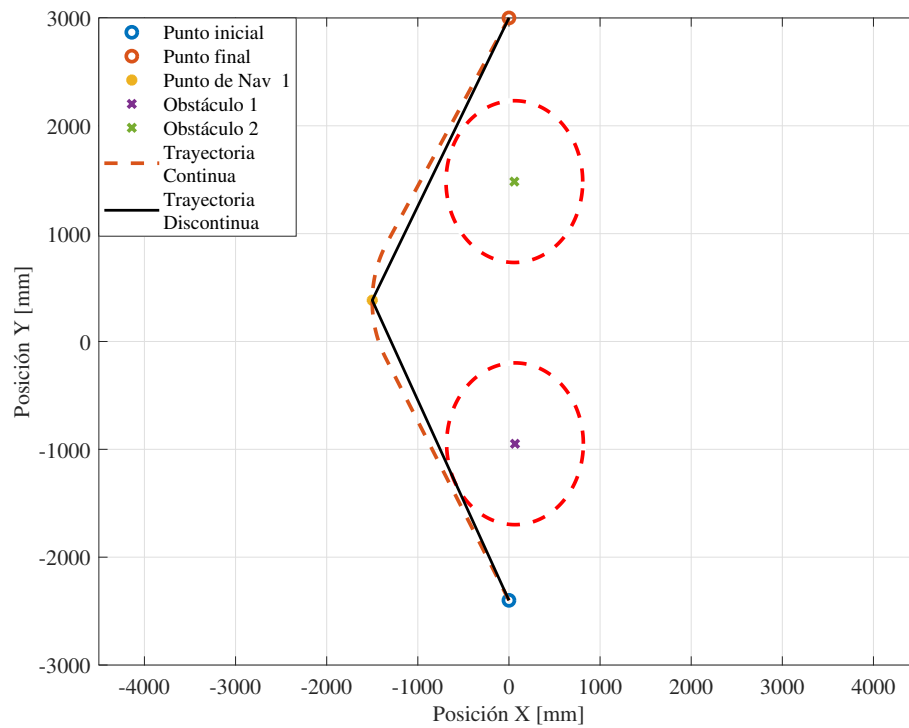


Figura 4.3: Trayectoria generada 2D para dos obstáculos.

4.1.3 CASO C: TRES OBSTÁCULOS QUE ENCIERRAN EL PUNTO FINAL.

Uno de los escenarios más comunes de evasión de obstáculos es cuando un grupo de obstáculos encierra o protege un objetivo esto es una formación tipo fortaleza. Para el caso de esta prueba, se colocaron tres obstáculos que protegen el punto de destino esto como se observa en la Figura 4.4. Los obstáculos colocados en formación fortaleza obligaron a la trayectoria genética a tener que rodear los obstáculos por la zona superior derecha para poder formar la trayectoria, además que la trayectoria genética es realmente tangente a los obstáculos, esto es bueno porque garantiza la trayectoria mínima y el mínimo consumo de energía. Para la trayectoria continua se tuvo que realizar un giro brusco en la esquina para llegar al punto final; sin embargo, la trayectoria aun en estas condiciones puede ser resuelto matemáticamente con un

solo punto de navegación.

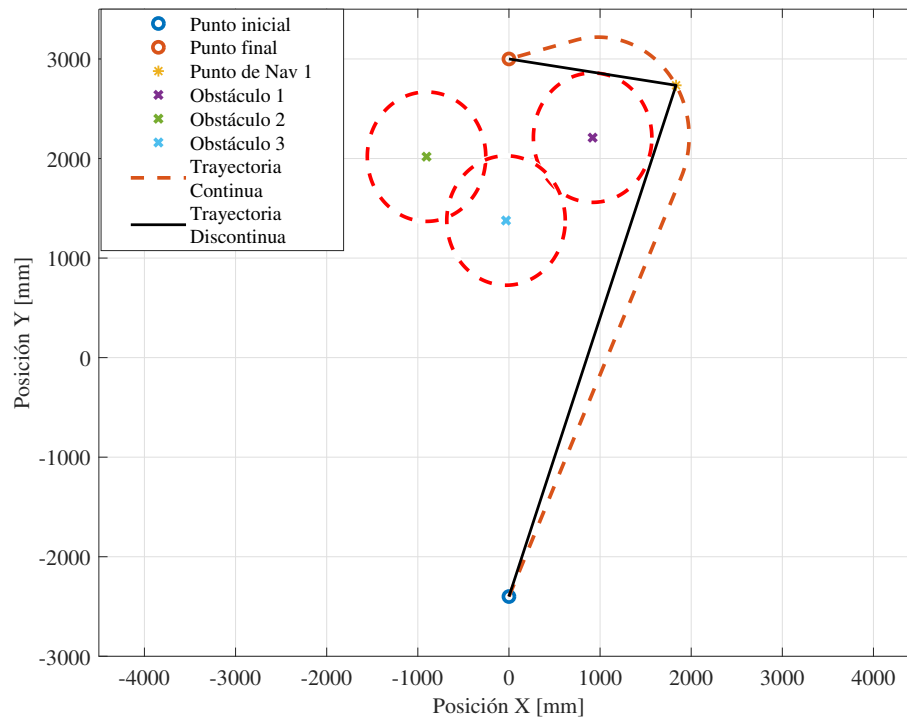


Figura 4.4: Trayectoria generada 2D para tres obstáculos.

4.1.4 CASO D: CUATRO OBSTÁCULOS COLOCADOS DE FORMA ALEATORIA.

Para la siguiente prueba se colocaron cuatro obstáculos de forma aleatoria para probar las capacidades del algoritmo genético, esto como se aprecia en la Figura 4.5. Se puede observar que estos obstáculos no permiten resolver la trayectoria con solo un punto de navegación o que no es matemática mente posible resolver con solo un punto de navegación sin cruzar los límites del Laboratorio o tener que rodear los obstáculos lo cual no es eficiente porque sería una trayectoria más larga, por lo tanto, se conduce a la generación de dos puntos de navegación. Sin embargo, aunque se tuvieron que usar dos puntos de navegación la trayectoria sigue siendo lo más

posible tangente a los obstáculos, lo cual significa que la trayectoria sigue siendo la de distancia mínima posible y sigue evitando la colisión de obstáculos.

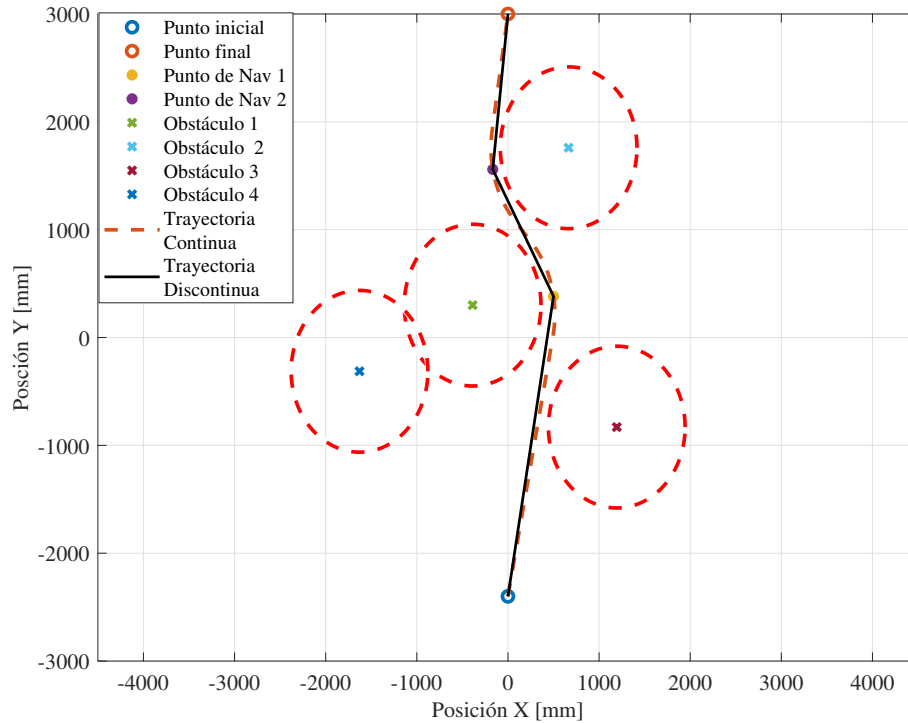


Figura 4.5: Trayectoria generada 2D para cuatro obstáculos aleatorios.

4.1.5 CASO E: CUATRO OBSTÁCULOS EN FORMACIÓN DE DOBLE BARRERA.

Algunos escenarios de evasión se pueden modelar como un grupo de barreras, como muros, esquinas o incluso barreras físicas. Por este motivo, un grupo de cuatro obstáculos se disponen por parejas formando dos barreras espaciadas, superpuestas en el eje Y cómo se puede observar en la Figura 4.6. Esta configuración al igual que en el caso anterior obliga a una trayectoria a que debe tener, al menos, dos vueltas, esto significa que el algoritmo debe calcular dos puntos de navegación para poder resolver la trayectoria, además, aunque era más simple solo rodear los obstáculos, se

puede observar que el algoritmo hace la trayectoria en entre las barreras sin atravesar los obstáculos buscando siempre la trayectoria más corta, aunque tenga que colocar más de un punto de navegación.

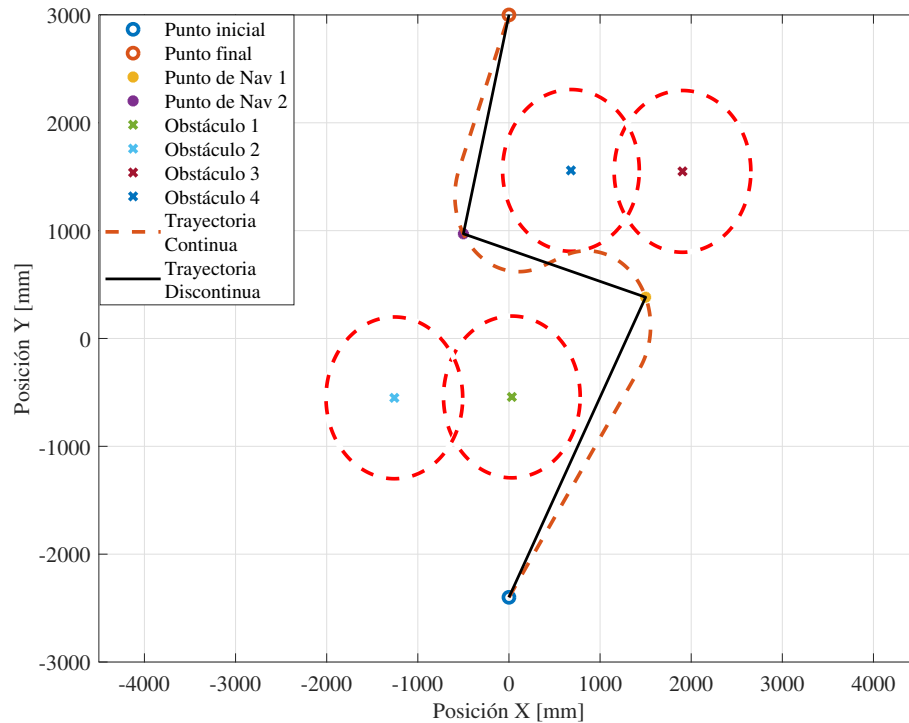


Figura 4.6: Trayectoria generada 2D para cuatro obstáculos en formación barreras.

4.1.6 GRÁFICAS DE PESO.

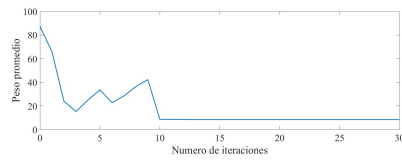
Las gráficas de peso son muy importantes porque nos describen cómo se comporta el algoritmo por iteraciones y como llega a la solución convergente, por ejemplo, en la Figura 4.10(b) que corresponde a un obstáculo, se puede observar que, al no ser un caso muy complicado, el algoritmo converge en una solución rápidamente, aunque existió una convergencia rápida es posible ver algunos picos entre las internaciones 5 y 10, estos son causadas debido al proceso de mutación. En la Figura 4.11 que corresponde al caso de dos obstáculos, el algoritmo tampoco tiene problemas para

converger, pero al ser un caso un poco más complicado que el anterior tiende a demorar más tiempo para converger, demostrando que entre más obstáculos más costo computacional existirá. En la Figura 4.12(a) que corresponde al caso 3, que corresponde a los tres obstáculos en formación de barrera, ocurre de manera muy similar al caso dos, que al ser un caso más complicado tiende a tardar más en converger hasta converger cerca de la iteración 20. En la Figura 4.12(b), que corresponde al caso de 4 obstáculos en la formación de barreras, se puede observar que el sistema convergerá cerca de la iteración 10, después de que ocurran algunos picos de mutación que parecen alterar la solución, pero el algoritmo termina convergiendo cerca de la iteración 20. Por último, en la Figura 4.12(c) se observa la gráfica de peso para el caso de 4 obstáculos en formación de barrera, aquí ocurre algo extraño, ya que al ser un caso más complicado debería tardar más en converger, pero como se puede observar existe en la iteración 8 un pico de mutación que ayuda a tener una respuesta de convergencia más rápida.

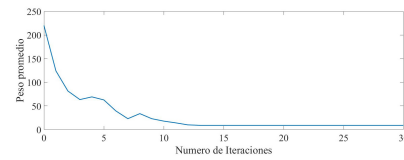
4.1.7 ALGORITMO ESTÁTICO 3D

Además, se probó el algoritmo para un entorno en 3D con un obstáculo y con tres obstáculos. En la Figura 4.12(d) se observa la gráfica de la primera prueba, donde tenemos un punto de inicio y un punto final y que justo en medio de los puntos hay un obstáculo que se interpone entre ellos. Se puede observar que la trayectoria generada para el entorno en 3D en caso estático, al igual que en los casos 2D realizados, la localización del punto de navegación sigue siendo lo más tangente posible al obstáculo, que en este caso es una esfera, en media del espacio tridimensional, de esta manera se puede observar que se cumple la condición de trayectoria corta y la evasión de obstáculos.

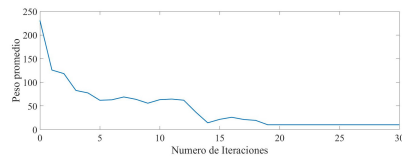
En la gráfica de peso se puede observar el comportamiento de la trayectoria contra las iteraciones en la Figura 4.12(e), se pueden observar unos cuantos picos de mínimos locales en las iteraciones 3 y 15 hasta que en la iteración 16 llega al



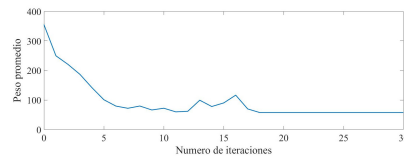
(a) Grafica de peso 1 obstáculo



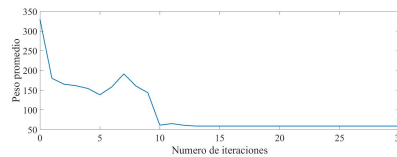
(b) Grafica de peso 2 obstáculos



(c) Grafica de peso 3 obstáculos



(d) Grafica de peso 4 obstáculos aleatorios



(e) Gráfica de peso 4 obstáculos en formación barrera

Figura 4.7: Gráficas de peso para entorno estático.

mínimo global que es el caso de la trayectoria más corta sin colisiones, además se puede observar que entre la iteración 5 a la 14 existen bastantes picos debidos a los procesos de mutación y cruce que incrementan el peso promedio de la gráfica antes de llegar a una convergencia. Además, se puede observar que la tercera dimensión incrementa el proceso computacional aumentando el número de iteración en la cual alcanza la convergencia, esto comparado con su caso en 2D.

En un segundo caso para probar el algoritmo se diseñó en un entorno en 3D con tres obstáculos, estos tres obstáculos fueron colocados de manera que no pudiera existir la trayectoria directa entre el punto inicial y el punto final. La gráfica de la trayectoria generada se observa en la Figura 4.12(e) donde se puede observar que el algoritmo es capaz de encontrar una trayectoria de manera eficiente a un en este tipo de entornos, además que en la Figura 4.12(e) b se observa que en el plano Z,

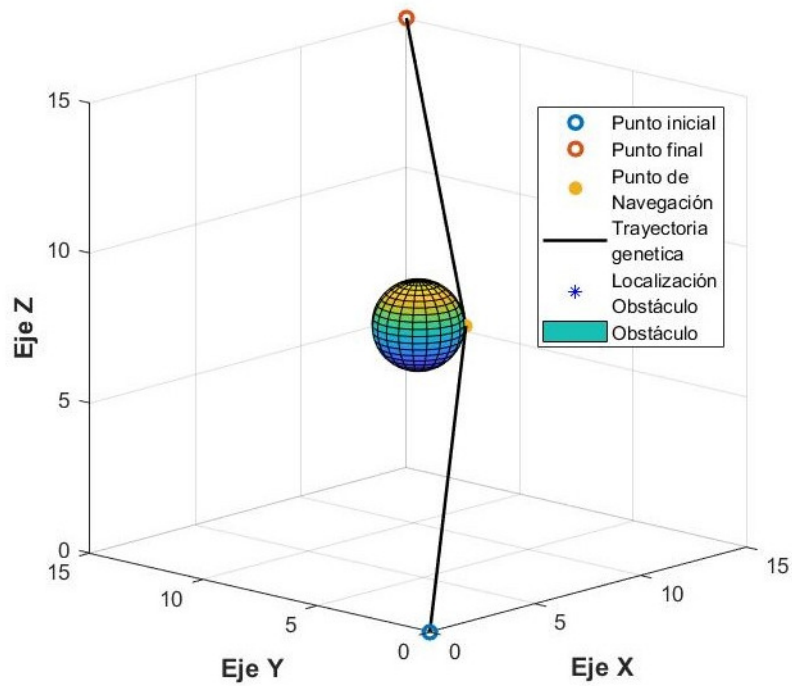


Figura 4.8: Trayectoria generada 3D para un obstáculo.

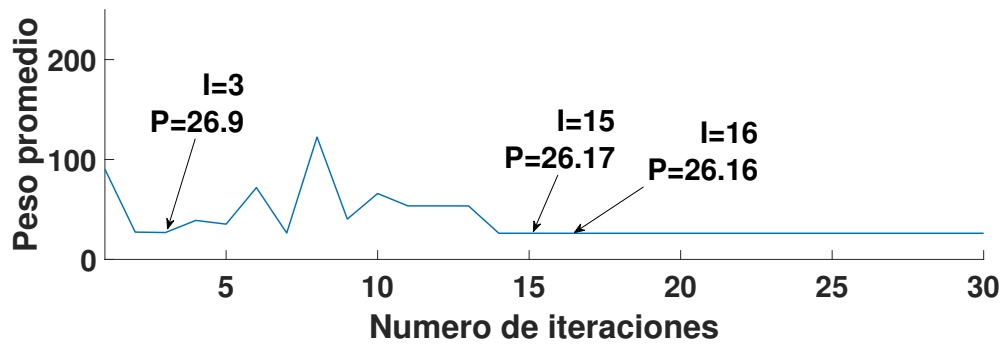
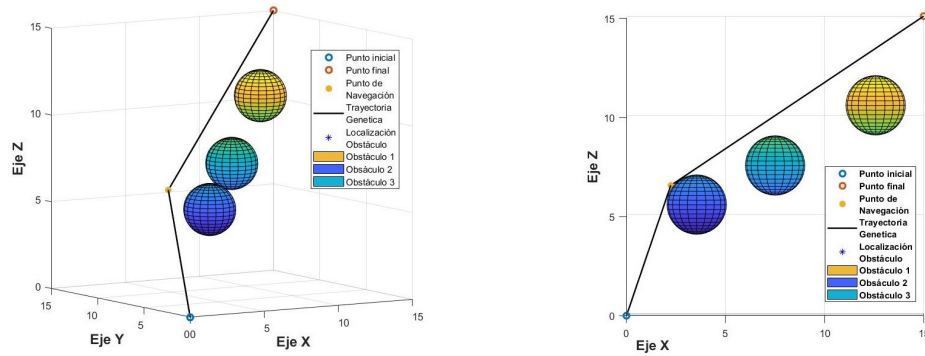


Figura 4.9: Gráfica de peso para entorno 3D para un obstáculo.

X, la trayectoria sigue buscando ser lo más tangente posible y de esta manera ser la trayectoria mínima y así cumplir los criterios de evasión de obstáculos y trayectoria con mínima distancia.



(a) Trayectoria generada 3D para tres obstáculos (b) Trayectoria generada 3D para tres obstáculos en plano z, x.

Figura 4.10: Trayectoria generada 3D para tres obstáculos.

La gráfica de peso de la trayectoria 3D con tres obstáculos se observa en la Figura 4.11, se pueden observar algunos picos de mínimos locales en la iteración 3 y 8 hasta que llega al punto de convergencia en la iteración 21. Después, aunque el algoritmo ya ha convergido se puede observar que entre la interacción 20 a la 40 aparecen algunos picos de mutación que afectaron el promedio de la convergencia, pero debido a que ya había convergido el algoritmo, la trayectoria final no se vio afectada.

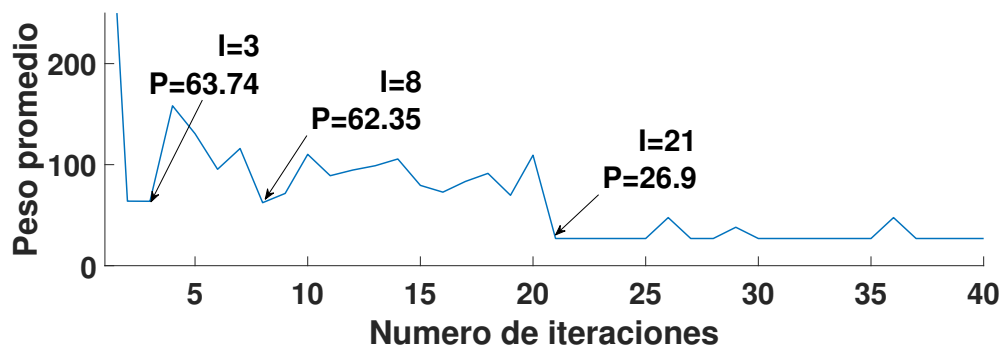


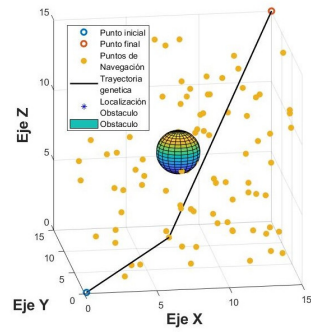
Figura 4.11: Gráfica de peso para entorno 3D para tres obstáculos.

4.1.8 ALGORITMO DINÁMICO 3D

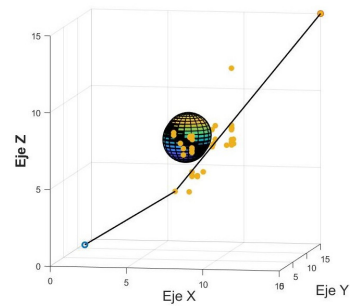
Para probar el algoritmo en entorno dinámico es decir un entorno con obstáculos en movimiento y que sea capaz de recalculando su trayectoria, se diseñaron dos pruebas con solo un obstáculo, uno con movimiento en el eje x y el eje z creando un movimiento en diagonal en el espacio, y otra con un solo movimiento en el eje Z provocando un movimiento vertical por parte del obstáculo. Por último, para probar las capacidades del algoritmo se diseñó una prueba con tres obstáculos en medio de la trayectoria y con movimiento aleatorio.

En la Figura 4.12(a) se puede observar un obstáculo que se interpone entre el punto inicial y el punto final, se puede observar que la población está dispersada, esto debido a que se encuentra en la iteración 1 y no ha convergido sino que prácticamente es la población inicial generada por el algoritmo. En la Figura 4.12(b) se puede observar la evolución de la trayectoria, primero podemos ver que la población ya ha comenzado a converger y que el punto de inicio que en este caso representa la localización del vehículo, se dirige al punto de navegación que mejor cumple con la aplicación, además se puede observar que el obstáculo ha comenzado a moverse, esto se aprecia por el efecto de barrido que se ve en la Figura. Después en la 4.12(c), se observa que el obstáculo por su movimiento está fuera del rango de la trayectoria y que el algoritmo ha encontrado una trayectoria sin interferencia del obstáculo por lo que la trayectoria es directa entre el punto de navegación y el punto final, además se aprecia que la población ha vuelto a dispersarse en lugar de converger, esto debido al proceso de mutación que ocurre cuando el punto de inicio está cerca del punto de navegación. En la 4.12(d) se puede observar que al ya no existir un obstáculo que intervenga en la trayectoria, se cumple el criterio de mutación que hace que el punto de navegación converja en el punto final. Por último, en la 4.12(e) y 4.12(f) que corresponden a 75 iteraciones y a 90 iteraciones, se observa que el punto de inicio llega al punto final y que a su vez se vuelve a cumplir el criterio de mutación que cuando está cerca el punto de inicio al punto de navegación se genera una población

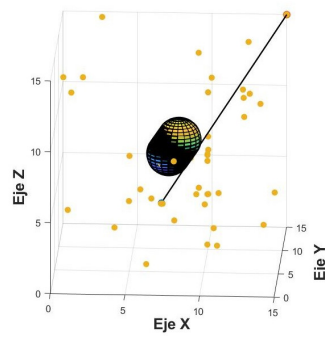
aleatoria, pero debido a la población de convergencia la trayectoria no se ve afectada.



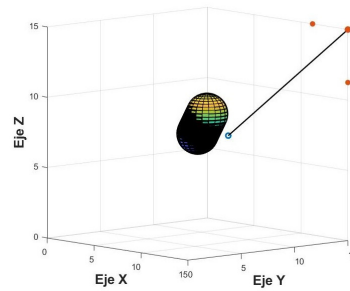
(a) Trayectoria dinámica I=1



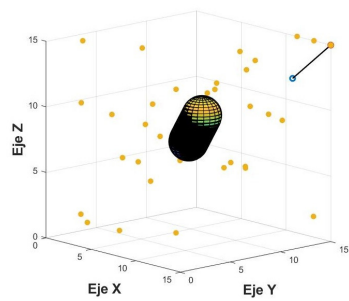
(b) Trayectoria dinámica I=10



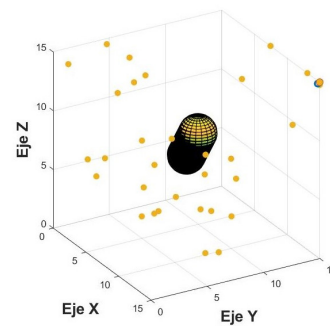
(c) Trayectoria dinámica I=35



(d) Trayectoria dinámica I=50



(e) Trayectoria dinámica I=75



(f) Trayectoria dinámica I=90

Figura 4.12: Trayectoria dinámica un obstáculo movimiento diagonal

La gráfica de peso de la trayectoria dinámica se muestra en la Figura 4.13, se

puede observar que a diferencia de las gráficas de peso para entorno estático esta gráfica disminuye a cero, esto debido a que entre más se acerca el punto de inicio al punto final la trayectoria se hace menor y por lo tanto el peso de la función objetivo también disminuye. Entre las iteraciones 10 a la 35 se puede observar picos ocasionados por el proceso de cruce, esto debido a que en las primeras iteraciones cuando se hacen los procesos de cruce se crean algunos individuos que no son eficientes e incrementan el peso promedio, esto también es debido al proceso de alta mutación. Después de la iteración 50 se observa que el punto de navegación ha convergido en el punto final y además debido al proceso de población de convergencia y al porcentaje de cruce seleccionado la gráfica de peso se ve muy poco afectada la gráfica, estos cambios se aprecian en mayor manera entre las iteraciones 15 a la 35 y entre las iteraciones 75 a la 90. La trayectoria seguida por el vehículo se observa en la Figura 4.14.

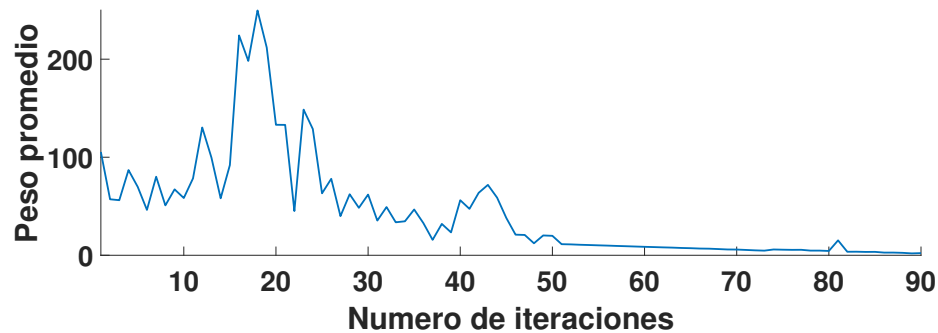


Figura 4.13: Gráfica de peso en entorno 3D dinámico para un obstáculo en movimiento diagonal.

Para el segundo ejemplo diseñado para probar el algoritmo en entorno dinámico se creó una prueba con un obstáculo en medio del punto de inicio y el punto final, pero en este caso el obstáculo se mueve en el eje z. Se puede observar en la Figura 4.15(a) que corresponde para la iteración 1 que la población está dispersa y que la trayectoria seleccionada no es eficiente. Después en la Figura 4.15(b) la cual corresponde al algoritmo en la iteración número 10, se aprecia que la población ha convergido y los posibles puntos de navegación tratan de ser lo más tangente posible

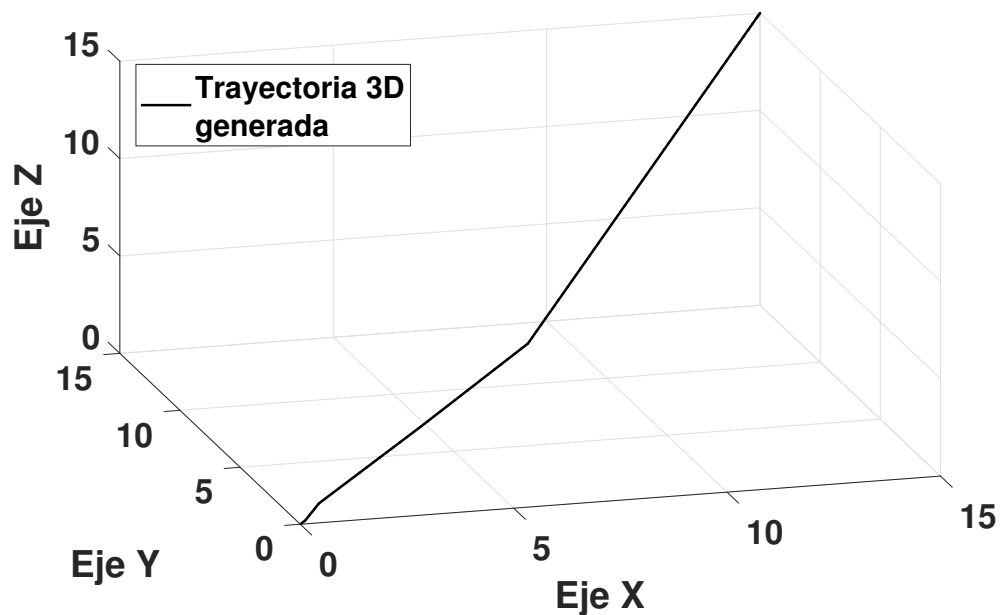


Figura 4.14: Gráfica de peso en entorno 3D dinámico para un obstáculo en movimiento diagonal.

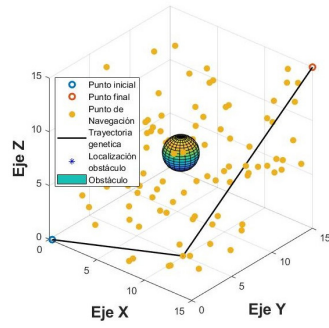
al obstáculo, además que el punto de inicio ha avanzado al punto de navegación que mejor cumple la trayectoria deseada. En la Figura 4.15(c) que corresponde para la iteración número 35 pareciera que la trayectoria cruza el obstáculo por la parte inferior del obstáculo, pero esto es debido solo al efecto de barrido para poder ver el movimiento del obstáculo, en realidad el obstáculo no se encuentra ahí y la trayectoria nunca cruza un obstáculo, además se puede apreciar que también debido a que el punto de inicio se encuentra cerca del punto de navegación se cumple el criterio de mutación y se crea una población aleatoria en la cuarta parte de la población para que exista diversidad y no se estanque en el punto de navegación, también se puede observar que cerca de la iteración 50 el algoritmo ya no encuentra nada que se interponga entre el punto de inicio y punto final por lo que se cumple el criterio de mutación que hace que el punto de navegación converja en el punto final así asegurando que se cumpla el objetivo, esto se observa en la Figura 4.15(d). Por último, se observa en la Figura 4.15(e) y la Figura 4.15(f) como el vehículo llega al su objetivo y cuando el punto de inicio y el punto de navegación están muy cerca el criterio de

mutación se vuelve a cumplir y vuelve a haber diversidad en la población, pero debido a la población de convergencia y al porcentaje utilizado de cruce la trayectoria no se ve afectada.

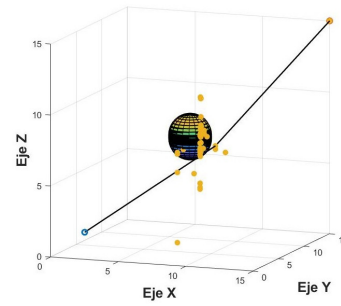
La gráfica de peso para el caso dinámico de un obstáculo en movimiento vertical se presenta en la Figura 4.14, que al igual que en el caso anterior se observan picos entre las iteraciones 0 a la 50, esto debido al constante proceso de recalcular la trayectoria, esto ocasiona que los operadores genéticos creen mucha diversidad de población. Después cuando ya no se encuentra ningún obstáculo entre la trayectoria el punto de navegación converge al punto final y la gráfica de peso desciende hasta cero debido a que sin un obstáculo que interfiera con la trayectoria el único criterio de la función objetivo es la distancia de la trayectoria y esta va disminuyendo conforme el punto de inicio llega al punto final. La trayectoria realizada por el vehículo se presenta en la Figura 4.17.

En la última prueba se utilizaron tres obstáculos con movimiento aleatorio. En la Figura 4.18(a) se observa la trayectoria para 1 iteración, se puede observar como en los casos anteriores la diversidad de la población. La Figura 4.18(b) muestra el movimiento de los tres obstáculos y la evolución del movimiento del punto de inicio, se puede apreciar también que la población ha empezado a converger. Después en la Figura 4.18(c)- 4.18(f), se puede observar la evolución de como el punto inicial llega al punto final, además se aprecia que en la Figura 4.12(b), debido al tercer obstáculo que entre las iteraciones 40 a 50 la trayectoria tuvo que cambiar debido a la interacción con el tercer obstáculos, sin embargo en la Figura 4.12(c) vuelve el punto de navegación a converger con el punto final debido a que el obstáculo por su movimiento sale de la trayectoria, esto demuestra que el algoritmo es capaz de recalcular la trayectoria cuando un obstáculo se interpone y la convergencia con el punto final ya existía.

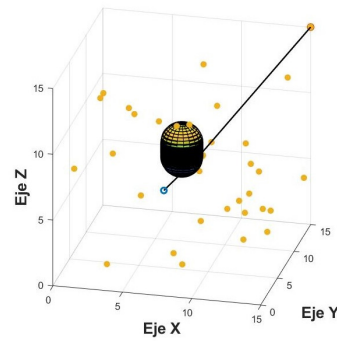
Por último, en la Figura 4.15(b) se puede observar la gráfica de peso, igual que en los casos pasados en las primeras iteraciones se encuentran los picos más altos



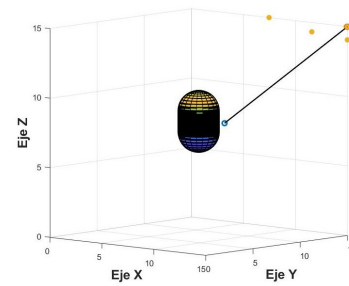
(a) Trayectoria dinámica I=1



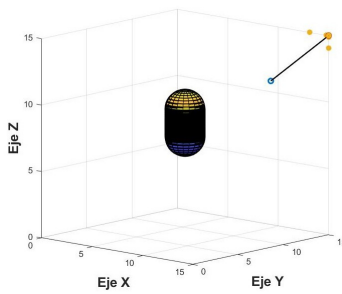
(b) Trayectoria dinámica I=10



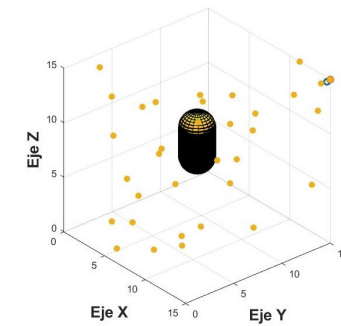
(c) Trayectoria dinámica I=35



(d) Trayectoria dinámica I=50



(e) Trayectoria dinámica I=75



(f) Trayectoria dinámica I=90

Figura 4.15: Trayectoria dinámica un obstáculo movimiento vertical.

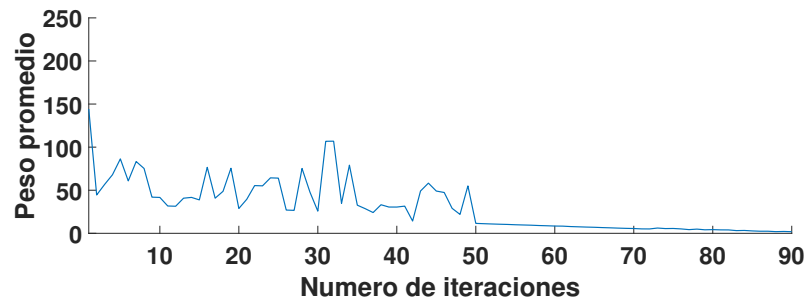


Figura 4.16: Grafica de peso en entorno 3D dinámico para un obstáculo en movimiento diagonal.

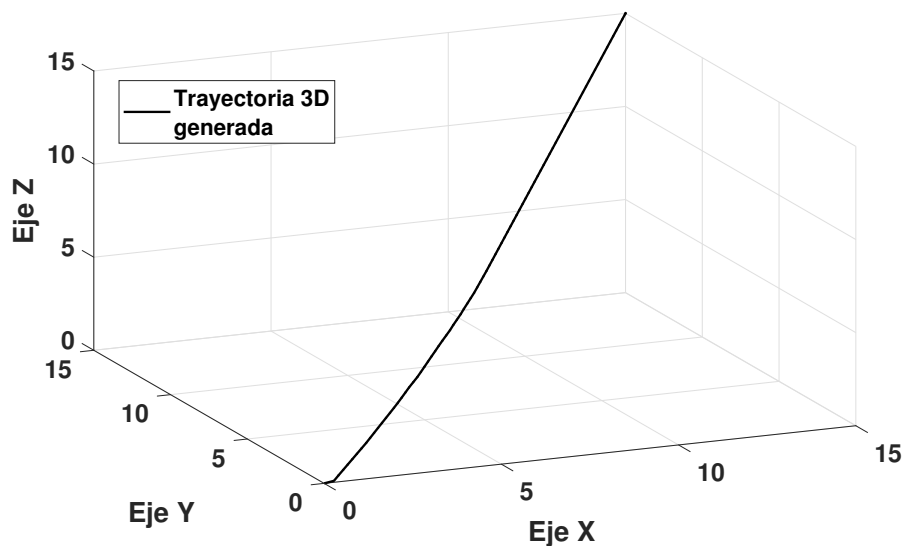


Figura 4.17: Gráfica de peso en entorno 3D dinámico para un obstáculo en movimiento diagonal.

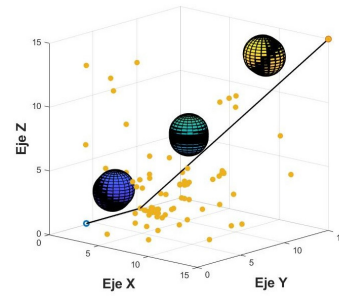
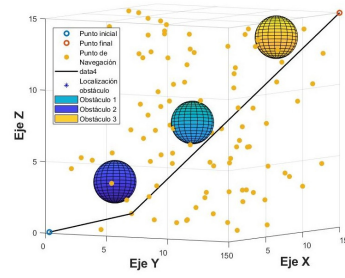
debido a que los operadores genéticos están tratando de converger además que el promedio de puntos de navegación no es eficiente. Después entre las iteraciones 40 a la 50 se puede observar que una gran cantidad de puntos de navegación cruzaron por los obstáculos por lo tanto el peso se elevó mucho, pero después logro salir de esa trayectoria. Hasta la iteración 70 se encontró una trayectoria sin interacción con los obstáculos, así que se cumplió el criterio de mutación y el punto de navegación converge en el punto final. Por último, en la Figura 4.20 se puede observar la trayectoria

realizada por el vehículo.

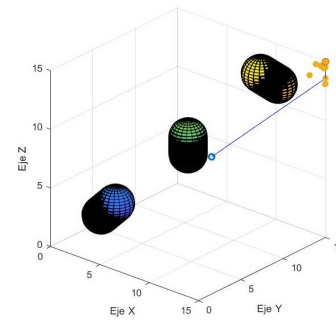
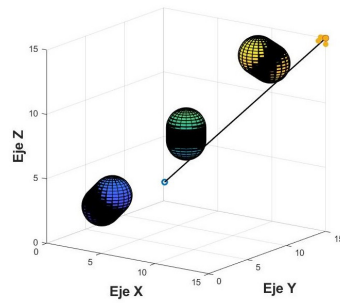
Por último, se hizo un suavizado de la trayectoria con un filtro paso bajas de primer orden con una frecuencia de corte muy baja de 0.03 Hz se obtuvo empíricamente, este filtro se utilizó para que la trayectoria no presentara discontinuidades o cambios muy bruscos en la trayectoria, ya que de manera física un Vant no puede seguir una trayectoria con discontinuidades debido a que sería una derivada infinita. En la Figura 4.21(a) se puede observar la gráfica de la posición X contra las iteraciones sin procesamiento, mientras que en la Figura 4.21(b) se muestra el suavizado de la trayectoria después del procesamiento del filtro paso bajas, se puede observar que donde hay un cambio muy brusco el filtro hace una curva suave. Este se repite para todas las posiciones, la Figura 4.22(a) y Figura 4.22(b) muestran el comportamiento de la posición Y contra las iteraciones sin procesamiento y la posición suavizada por procesamiento del filtro paso bajas, mientras que el comportamiento de la posición en Z contra las iteraciones se puede observar en la Figura 4.23(a) y en la Figura 4.23(b). Por último, la trayectoria discontinua y la trayectoria continua se observa en la Figura 4.24(a) y en la Figura 4.24(b), donde se puede apreciar que el filtro funciona eficientemente y que la trayectoria sigue esquivando obstáculos.

En una segunda prueba se realizó un proceso de ida y vuelta, primero el vehículo llega a un primer objetivo espera un momento y una vez cumple su objetivo regresa al punto de inicio, todo esto sin nunca tocar el obstáculo. Se puede observar en las figuras 4.25(a) el movimiento en el eje X sin procesamiento y en la figura 4.25(b) se tiene la gráfica de los ejes X con procesamiento, donde al comparar las gráficas se puede observar el suavizado del movimiento gracias al filtro paso bajas. La figura 4.26(a) muestra el movimiento en el eje Y sin el filtro mientras que la figura 4.26(b) muestra el movimiento en el eje Y con el filtro paso bajas. Por último La figura 4.27(a) muestra el movimiento sin filtrar del eje Z y por otro lado la figura 4.27(b) muestra el movimiento en el eje Z con el filtro. Al final la figura 4.28(a) muestra la trayectoria generada por el algoritmo genético para este nuevo caso donde se pueden apreciar algunas discontinuidades, después podemos observar en la figura 4.28(b) la

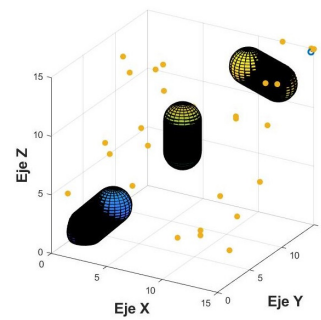
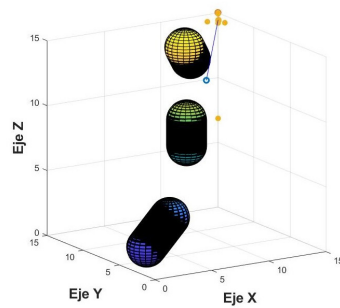
trayectoria suavizada ya sin discontinuidades.



(a) Trayectoria dinamica tres obstáculos I=1 (b) Trayectoria dinamica tres obstáculos I=10



(c) Trayectoria dinamica tres obstáculos I=35 (d) Trayectoria dinamica tres obstáculos I=50



(e) Trayectoria dinamica tres obstáculos I=75 (f) Trayectoria dinamica tres obstáculos I=90

Figura 4.18: Trayectoria dinámica tres obstáculos.

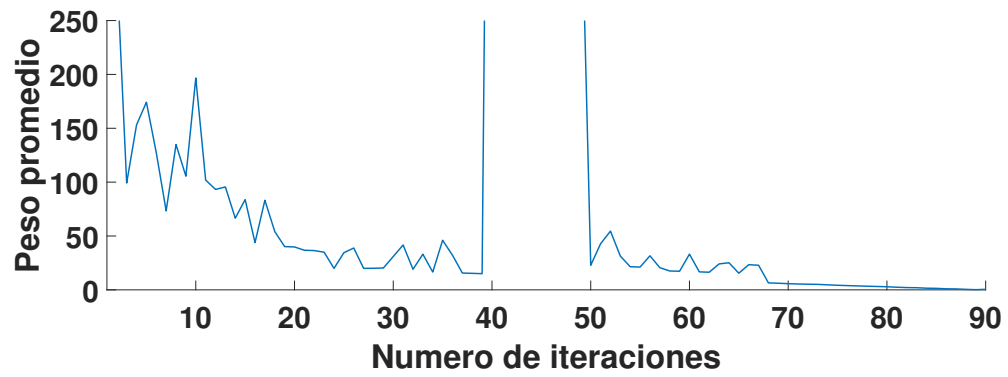


Figura 4.19: Gráfica de peso en entorno 3D dinámico para un obstáculo en movimiento diagonal.

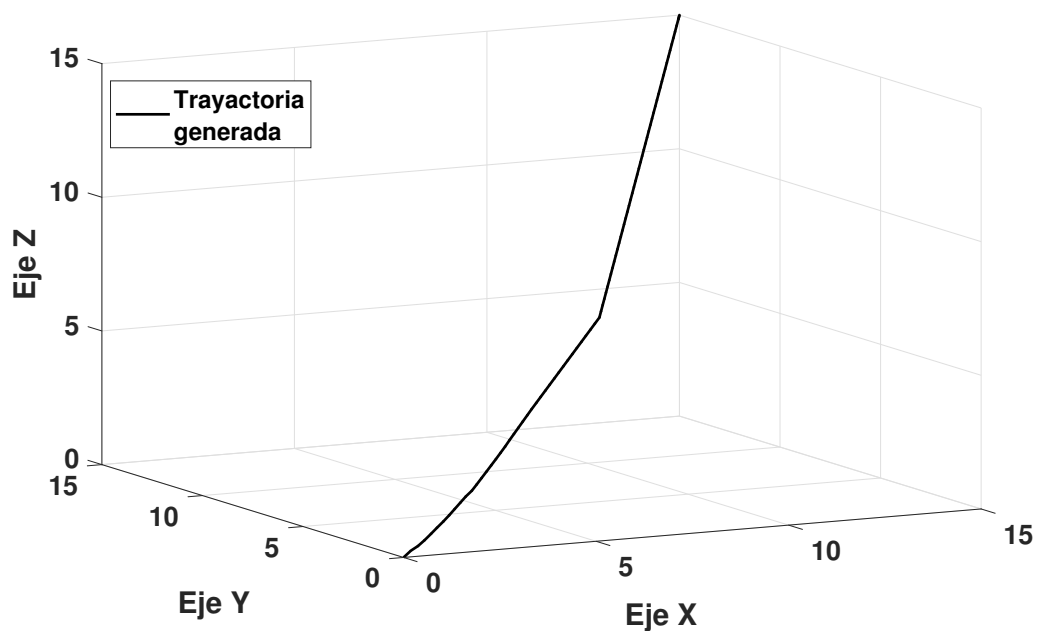
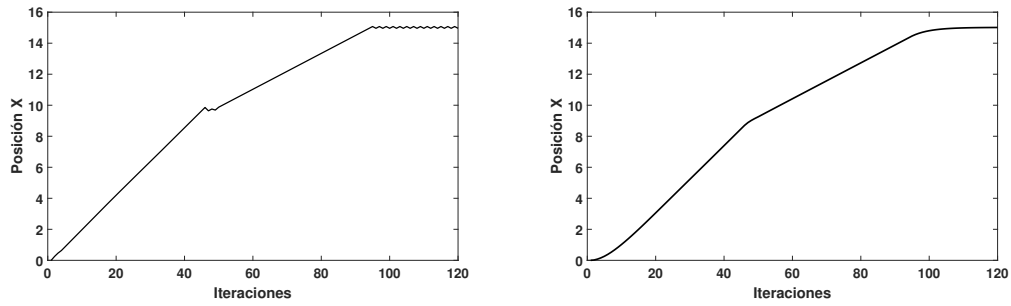
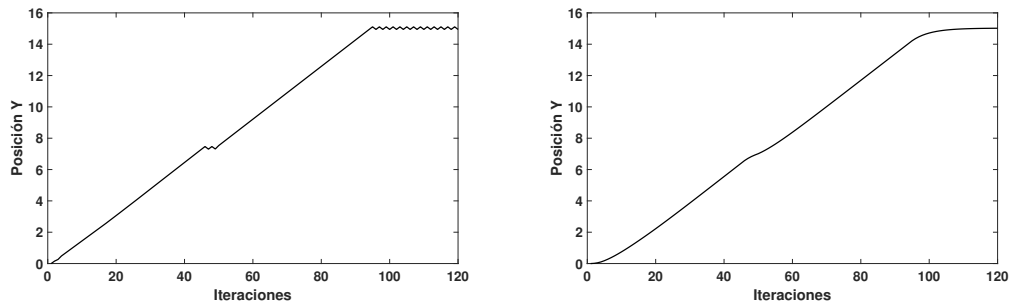


Figura 4.20: Gráfica de peso en entorno 3D dinámico para un obstáculo en movimiento diagonal.



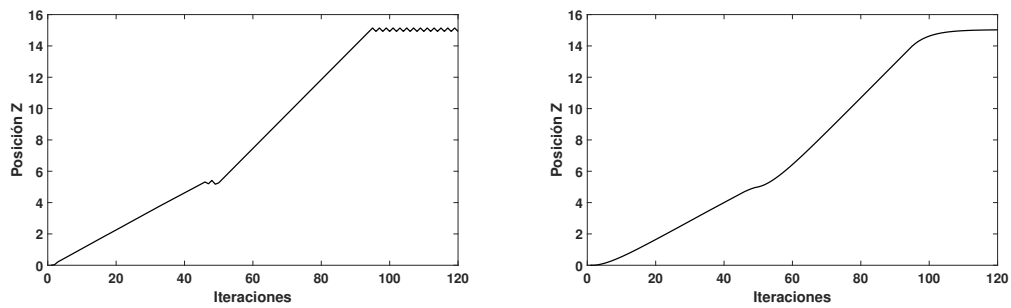
(a) Movimiento en el eje X sin procesamiento caso 1 (b) Movimiento en el eje X con procesamiento caso 1

Figura 4.21: Movimiento en el eje X sin y con procesamiento caso 1.



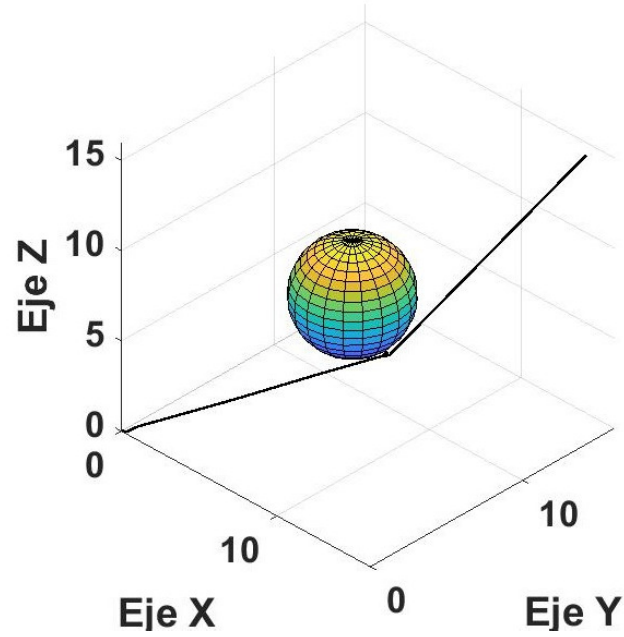
(a) Movimiento en el eje Y sin procesamiento caso 1 (b) Movimiento en el eje Y con procesamiento caso 1

Figura 4.22: Movimiento en el eje Y sin y con procesamiento caso 1.

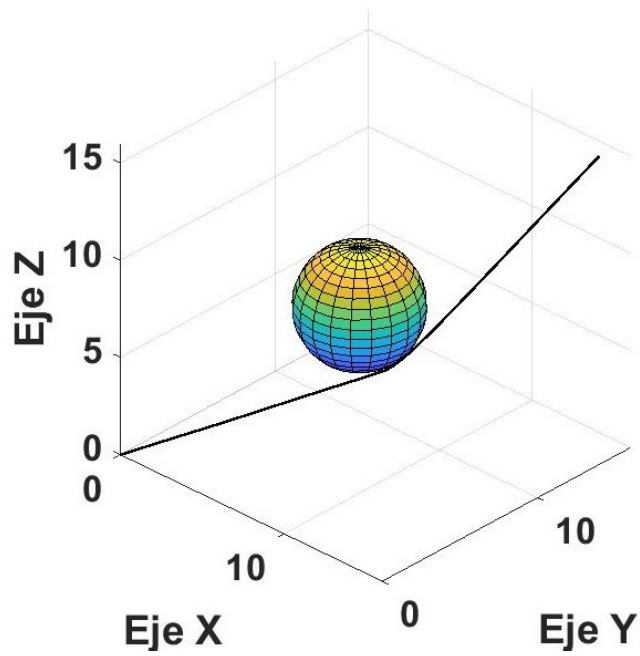


(a) Movimiento en el eje Z sin procesamiento caso 1 (b) Movimiento en el eje Z con procesamiento caso 1

Figura 4.23: Movimiento en el eje Z sin y con procesamiento caso 1.

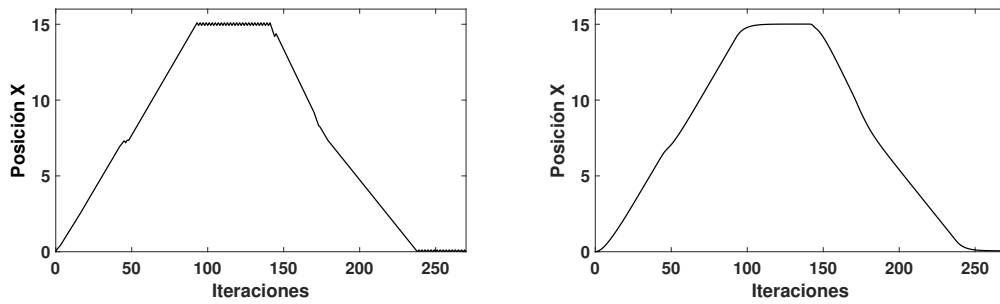


(a) Trayectoria 3D no suavizada caso 1



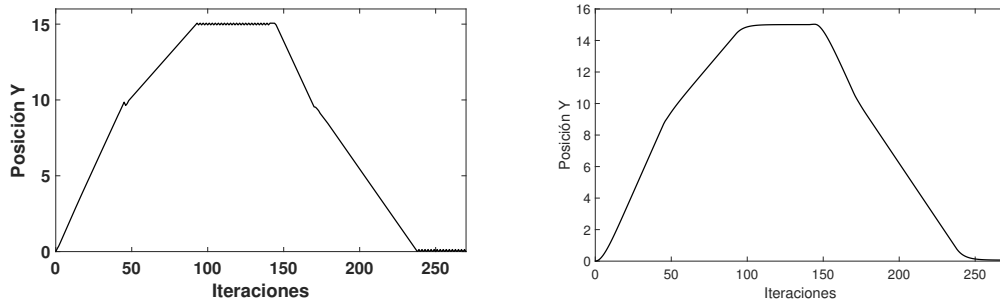
(b) Trayectoria 3D suavizada caso 1

Figura 4.24: Trayectoria dinamica tres obstaculos caso 1



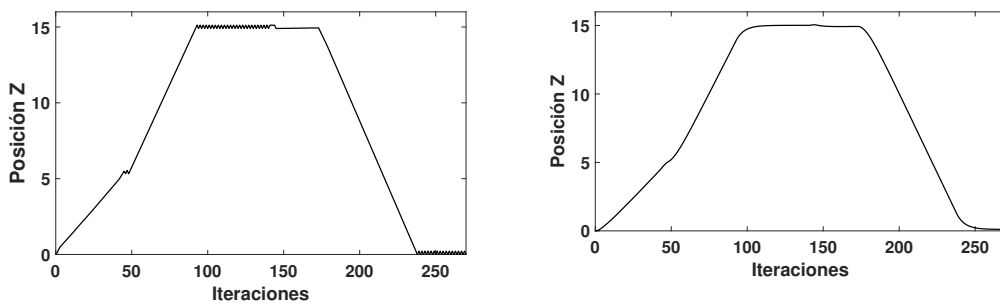
(a) Movimiento en el eje X sin procesamiento caso 2 (b) Movimiento en el eje X con procesamiento caso 2

Figura 4.25: Movimiento en el eje X sin y con procesamiento caso 2.



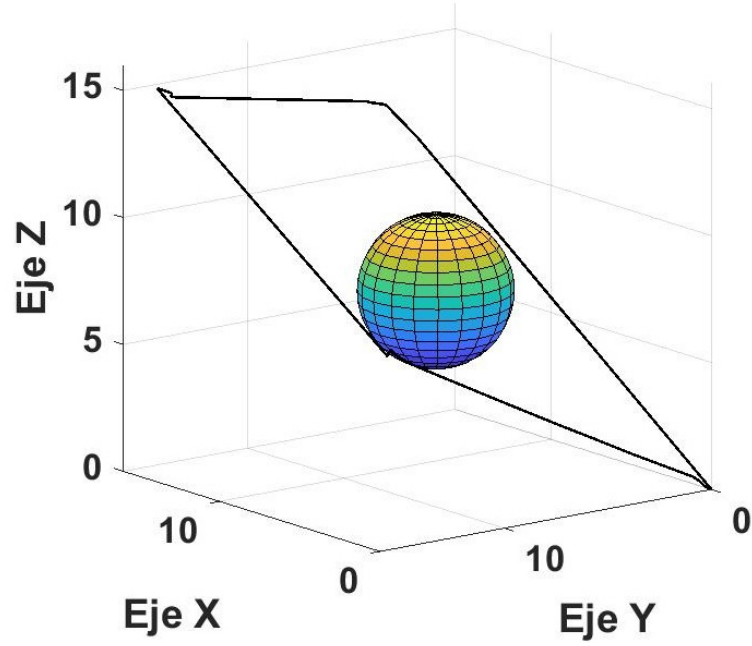
(a) Movimiento en el eje Y sin procesamiento caso 2 (b) Movimiento en el eje Y con procesamiento caso 2

Figura 4.26: Movimiento en el eje Y sin y con procesamiento caso 2.

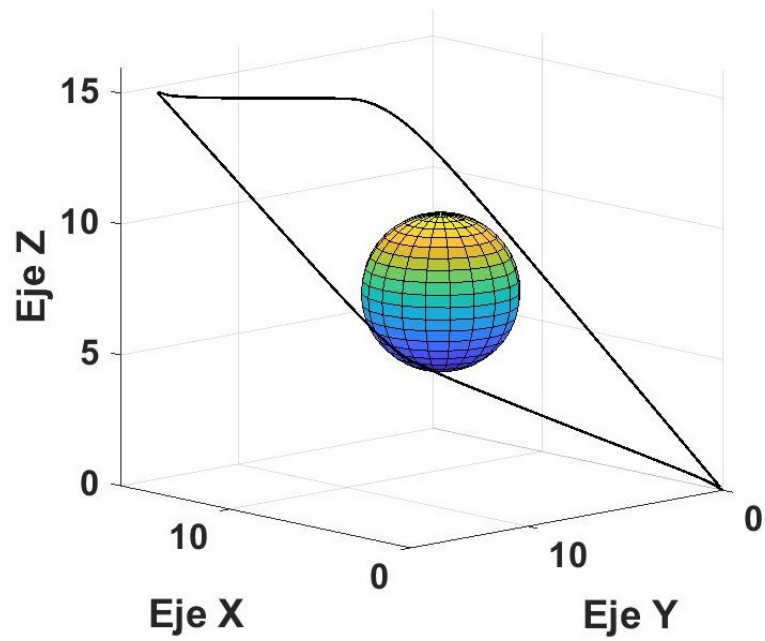


(a) Movimiento en el eje Z sin procesamiento caso 2 (b) Movimiento en el eje Z con procesamiento caso 2

Figura 4.27: Movimiento en el eje Z sin y con procesamiento caso 2.



(a) Trayectoria 3D no suavizada caso 2



(b) Trayectoria 3D suavizada caso 2

Figura 4.28: Trayectoria dinamica tres obstaculos caso 2

CAPÍTULO 5

CONCLUSIONES Y TRABAJOS FUTUROS.

En este trabajo de investigación se diseñó un algoritmo genético de planeación de trayectoria, se demostró que los algoritmos genéticos son eficientes es cuestiones de búsqueda donde existen múltiples opciones, los criterios y metodología utilizada juntos con la función multi-objetivo demostraron que el algoritmo es eficiente para encontrar trayectorias en el espacio.

Se demostró en las pruebas realizadas la eficacia de la trayectoria, donde las trayectorias generadas cuando no había un obstáculo que se interponga entre el punto inicial y el punto final, la trayectoria era la línea recta entre el objetivo y el destino, cuando existía algún obstáculo que se interpusiera entre el objetivo y el destino la trayectoria se hacía lo más tangente posible al obstáculo, es decir la distancia mínima posible sin entrar al obstáculo.

La cadena de cromosomas de 24 bits donde cada gen era de 8 bits para cada coordenada espacial dio la suficiente resolución para tener búsqueda en el espacio y a su vez esto permitió que en cada iteración la trayectoria se volviera cada vez más tangente al obstáculo, para entorno estático la población de 500 y el porcentaje de mutación de 0.2 ayudo a que existiera una convergencia alrededor de la iteración

15 a la 20, esto para entorno estático, ya que para entorno dinámico se usó otra estrategia al no existir una convergencia como tal.

Para el entorno dinámico se utilizaron los criterios de alta mutación, además del criterio de repoblación aleatoria cuando el punto de inicio y el punto de navegación están muy cerca, el último criterio de mutación fue el del punto de navegación al punto final, estos criterios demostraron trabajar de manera eficiente, mientras unos criterios ayudaban a tener diversidad de población otros criterios ayudaban a que el algoritmo convergiera de manera eficiente, de esta manera se concluyó que para los casos dinámicos convergieron de manera eficaz. También se puede decir que se encontró una buena relación entre los porcentajes utilizados y la cantidad de individuos utilizados, que en este caso fueron 100 individuos y un porcentaje de mutación de 0,2, se redujo la población a 100, ya que para entorno dinámico en 3D 500 individuos incrementaba mucho el costo computacional, sin embargo esto resulto beneficioso porque al ser menor la población el porcentaje de mutación afectaba a mayor parte de la población que era lo que se buscaba, aunque al reducir la población fue menor la resolución de búsqueda, los criterios de cruce y de alta mutación ayudaron a tener diversidad de búsqueda. También se puede observar que el algoritmo trabaja bien para entorno 2D y 3D y que el filtro paso bajas utilizado funciona bien para el suavizado de la trayectoria, aunque la frecuencia de corte utilizada suavizo bien la trayectoria este se determinó de manera empírica y hace falta más estudio de la frecuencia de corte más eficiente.

Para trabajos futuros se planea hacer una comparación entre diferentes métodos para comparar tiempos de convergencia, simulaciones con un sistema de control, además de hacer la implementación del algoritmo en el VANT para pruebas en entorno dinámico la cual debido a la pandemia se tuvieron que posponer, también para trabajo futuro expandir el algoritmo a una plataforma de múltiples agentes.

APÉNDICE A

APÉNDICE A

A.1 ECUACIONES DE MOVIMIENTO DE QUADRTOR

El modelo del quadrotor considera un marco fijo inercial como $\mathcal{I}=\{x_{\mathcal{I}}, y_{\mathcal{I}}, z_{\mathcal{I}}\}$ y un marco fijado al cuerpo del vehículo unido al centro de gravedad del como \mathcal{B} $\{x_{\mathcal{B}}, y_{\mathcal{B}}, z_{\mathcal{B}}\}$. El Marco aerodinámico $\mathcal{A}=\{x_{\mathcal{A}}, y_{\mathcal{A}}, z_{\mathcal{A}}\}$ se considera durante el vuelo hacia adelante, [36], [35].

La formulación Newton-Euler, para cuerpos rígidos, es usada en orden para obtener el modelo dinámico del quadrotor

$$\dot{\xi} = V \tag{A.1}$$

$$m\dot{V} = Re_3(-T_T) + mge_3 + D_\xi \tag{A.2}$$

$$\dot{R} = R\hat{\Omega} \tag{A.3}$$

$$J\dot{\Omega} = -\Omega \times J\Omega + \tau_a + D_\eta \tag{A.4}$$

Donde

- $\xi = (x, y, z)^\top \in \mathbb{R}^3$ y $V = (\dot{x}, \dot{y}, \dot{z})^\top \in \mathbb{R}^3$ son las coordenadas de posición y la velocidad traslacional en relación con el marco inercial.

- $\eta = (\phi, \theta, \psi)^\top \in \mathbb{R}^3$ describe las coordenadas de rotación donde ϕ , θ y ψ representan el alabeo, cabeceo y guiñada o dirección, respectivamente.
- La matriz de rotación, $R \in SO(3) : \mathcal{B} \rightarrow \mathcal{I}$, satisface $SO(3) = \{R | R \in \mathbb{R}^{3 \times 3}, \det[R] = 1, RR^\top = R^\top R = I\}$ y está parametrizado por los ángulos de Euler ϕ , θ y ψ . La matriz de rotación es escrita como

$$R = \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix}$$

- $\Omega = (p, q, r)^\top \in \mathbb{R}^3$ es la velocidad angular en \mathcal{B} y $\hat{\Omega}$ denota la matriz simétrica sesgada del vector que viene dada por

$$\hat{\Omega} = \begin{pmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{pmatrix}$$

- e_1 , e_2 , and e_3 son los vectores de la base canónica de \mathbb{R}^3 .
- $J \in \mathbb{R}^{3 \times 3}$ contiene los momentos de inercia y $m \in \mathbb{R}$ denota la masa del quadrotor VANT.
- $T_T \in \mathbb{R}^3$ es el empuje total y $\tau_a \in \mathbb{R}^3$ son los momentos del actuador del quadrotor.
- $D_\xi = (d_{\xi_1}, d_{\xi_2}, d_{\xi_3})^\top \in \mathbb{R}^3$ y $D_\eta = (d_{\eta_1}, d_{\eta_2}, d_{\eta_3})^\top \in \mathbb{R}^3$ son perturbaciones delimitadas suaves, posiblemente variables en el tiempo y dependientes del estado.

Los momentos del actuador τ_a generados por los cuatro rotores se describen a continuación

$$\tau_a = \begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} = \begin{pmatrix} d[(f_2 + f_4) - (f_1 + f_3)] \\ d[(f_3 + f_4) - (f_1 + f_2)] \\ Q_2 + Q_3 - Q_1 - Q_4 \end{pmatrix}$$

Donde d denota la distancia desde el centro de masa hasta el eje del rotor y f_i es la fuerza de sustentación o la fuerza de empuje de la hélice para $i = 1, 2, 3, 4$. $Q_i = \rho_a A_{d_i} \omega_i^2 r_i^3 c_{Q_i}$ es el momento de reacción de los rotores con c_{Q_i} que denota el coeficiente de momento del eje del rotor, ρ_a es la densidad del aire, A_{d_i} denota el área del disco, ω_i denota la velocidad angular del rotor y r_i es el radio del rotor para $i = 1, 2, 3, 4$.

Momento giroscópico El momento del giroscopio generado por la rotación del fuselaje y los cuatro rotores se describe como

$$d_{\eta_{gyro}} = \sum_{i=1}^4 (-1)^{i+1} I_{r_i} [\Omega \times (\omega_i e_{b_3})]$$

Donde I_{r_i} es el momento de inercia del rotor i y ω_i denota la velocidad angular del rotor i , con $i=1, 2, 3, 4$.

A.1.0 PERTURBACIONES DEBIDAS AL VIENTO

Las *fuerzas aerodinámicas* producidas durante el vuelo son escritas como

$$D_{\xi}(t) = \begin{pmatrix} d_{\xi_1} \\ d_{\xi_2} \\ d_{\xi_3} \end{pmatrix} = RW^T \begin{pmatrix} D \\ Y \\ 0 \end{pmatrix}$$

con la matriz aerodinámica de rotación $W : \mathcal{B} \rightarrow \mathcal{A}$ que transforma una fuerza del marco de la carrocería en un marco aerodinámico se describe como

$$W = \begin{pmatrix} c_{\alpha} c_{\beta} & s_{\beta} & s_{\alpha} c_{\beta} \\ -c_{\alpha} s_{\beta} & c_{\beta} & -s_{\alpha} s_{\beta} \\ -s_{\alpha} & 0 & c_{\alpha} \end{pmatrix}$$

donde α es el ángulo de ataque y β son el ángulo de deslizamiento lateral. L , Y , y D son las fuerzas aerodinámicas: sustentación, fuerza lateral y resistencia, respectivamente, [29].

Los *Momentos aerodinámicos*: Los momentos aerodinámicos generados durante el vuelo se escriben como descrito como

$$d_{\eta_{aero}}(t) = \begin{pmatrix} \mathcal{L} \\ \mathcal{M} \\ \mathcal{N} \end{pmatrix}$$

Donde \mathcal{L} , \mathcal{M} y \mathcal{N} son los momentos aerodinámicos de alabeo, cabeceo y guiñada, respectivamente, [?, ?].

Reescribiendo las perturbaciones en los vectores variables en el tiempo $D_{\xi}(t)$ y $D_{\eta}(t)$, resulta

$$D_{\xi}(t) = \begin{pmatrix} d_{\xi_1} \\ d_{\xi_2} \\ d_{\xi_3} \end{pmatrix}, \quad D_{\eta}(t) = d_{\eta_{gyro}}(t) + d_{\eta_{aero}}(t) = \begin{pmatrix} d_{\eta_1} \\ d_{\eta_2} \\ d_{\eta_3} \end{pmatrix}$$

que son continuamente diferenciables.

BIBLIOGRAFÍA

- [1] N. Achour and M. Chaalal, “Mobile robots path planning using genetic algorithms,” *The Seventh International Conference on Autonomic and Autonomous Systems ICAS 2011*, 01 2011.
- [2] P. Anish, “Mobile robot navigation and obstacle avoidance techniques: A review,” *International Robotics and Automation Journal*, vol. 2, pp. 1–12, 05 2017.
- [3] T. Arora, Y. Gigras, and V. Arora, “Robotic path planning using genetic algorithm in dynamic environment,” *International Journal of Computer Applications*, vol. 89, 02 2014.
- [4] A. Atyabi and D. Powers, “Review of classical and heuristic-based navigation and path planning approaches,” *International Journal of Advancements in Computing Technology (IJACT)*, vol. 5, pp. 1–14, 01 2013.
- [5] H. Bharadwaj and V. K. E, “Comparative study of neural networks in path planning for catering robots,” *Procedia Computer Science*, vol. 133, pp. 417 – 423, 2018, international Conference on Robotics and Smart Manufacturing (RoSMa2018). [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1877050918309967>
- [6] X. Chen, R. Xu, and J. Zhao, “Multi-objective route planning for uav,” in *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*, 2017, pp. 1023–1027.

-
- [7] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, “Principles of robot motion: Theory, algorithms, and implementation errata!!!! 1,” 01 2003.
- [8] M. Dorigo, M. Birattari, and T. Stützle, “Ant colony optimization,” *Computational Intelligence Magazine, IEEE*, vol. 1, pp. 28–39, 12 2006.
- [9] R. C. DuToit, M. Holt, and S. Biaz, “Uav collision avoidance using rrt and los maximization technical report csse 1203,” 2012.
- [10] R. Eberhart and J. Kennedy, “New optimizer using particle swarm theory,” in *Proceedings of the International Symposium on Micro Machine and Human Science*, 1995.
- [11] G. E. D. Flores, E. S. E. Quesada, S. S. Cruz, L. R. G. Carrillo, and R. Lozano, “Online UAS local path-planning algorithm for outdoors obstacle avoidance based on attractive and repulsive potential fields,” in *2016 International Conference on Unmanned Aircraft Systems, ICUAS 2016*, 2016.
- [12] R. L. Galvez, E. P. Dadios, and A. A. Bandala, “Path planning for quadrotor UAV using genetic algorithm,” in *2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, HNICEM 2014 - 7th HNICEM 2014 Joint with 6th International Symposium on Computational Intelligence and Intelligent In*, 2014.
- [13] O. Garcia, E. G. Rojo-Rodriguez, A. Sanchez, D. Saucedo, and A. J. Munoz-Vazquez, “Robust geometric navigation of a quadrotor uav on $se(3)$,” *Robotica*, vol. 38, no. 6, p. 1019–1040, 2020.
- [14] S. Garrido and L. Moreno, *Mobile Robot Path Planning using Voronoi Diagram and Fast Marching*, 01 2015.
- [15] R. Geraerts, “Planning short paths with clearance using explicit corridors,” 06 2010, pp. 1997 – 2004.

-
- [16] D. E. Goldberg, “Genetic algorithms in search, optimization, and machine learning,” *Choice Reviews Online*, 1989.
- [17] M. Golub and L. Budin, “An asynchronous model of global parallel genetic algorithms,” 01 2000.
- [18] M. A. Gutierrez-Martinez, E. G. Rojo-Rodriguez, L. E. Cabriales-Ramirez, L. A. Reyes-Osorio, P. Castillo, and O. Garcia-Salazar, “Collision-free path planning based on a genetic algorithm for quadrotor uavs,” in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2020, pp. 948–957.
- [19] A. T. Hafez, M. A. Kamel, P. T. Jardin, and S. N. Givigi, “Task assignment/trajectory planning for unmanned vehicles via hflc and pso,” in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 554–559.
- [20] J. Han and Y. Seo, “Mobile robot path planning with surrounding point set and path improvement,” *Applied Soft Computing Journal*, 2017.
- [21] I. Hasircioglu, H. Topcuoglu, and M. Ermis, “3-d path planning for the navigation of unmanned aerial vehicles by using evolutionary algorithms,” 01 2008, pp. 1499–1506.
- [22] A. B. A. Hassanat, K. Almohammadi, E. Alkafaween, E. Abunawas, A. Hammouri, and V. Prasath, “Choosing mutation and crossover ratios for genetic algorithms - a review with a new dynamic approach,” *Inf.*, vol. 10, p. 390, 2019.
- [23] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975, second edition, 1992.
- [24] S. X. Y. Hong Qu, “Real-time robot path planning based on a modified pulse-coupled neural network model,” *IEEE Transactions on Neural Networks*, 2009.
- [25] V. Jeauneau, L. Jouanneau, and A. Kotenkoff, “Path planner methods for uavs in real environment,” *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 292 – 297, 2018, 12th IFAC Symposium on Robot Control SYROCO 2018.

- [26] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 500–505.
- [27] P. Kora and P. Yadlapalli, “Crossover operators in genetic algorithms: A review,” *International Journal of Computer Applications*, vol. 162, pp. 34–36, 03 2017.
- [28] C. Lamini, S. Benhlima, and A. Elbekri, “Genetic algorithm based approach for autonomous mobile robot path planning,” in *Procedia Computer Science*, 2018.
- [29] J. G. Leishman, *Principles of Helicopter Aerodynamics (Cambridge Aerospace Series), 2nd ed*, 2016.
- [30] T. Lozano-Perez and M. Wesley, “An algorithm for planning collision-free paths among polyhedral obstacles,” *Commun. ACM*, vol. 22, pp. 560–570, 1979.
- [31] T. T. Mac, C. Copot, A. Hernandez, and R. De Keyser, “Improved potential field method for unknown obstacle avoidance using uav in indoor environment,” in *2016 IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (SAMII)*, 2016, pp. 345–350.
- [32] D. A. Pandey, “Path planning navigation of mobile robot with obstacles avoidance using fuzzy logic controller,” 01 2014.
- [33] F. Samaniego, J. Sanchis, S. García-Nieto, and R. Simarro, “Uav motion planning and obstacle avoidance based on adaptive 3d cell decomposition: Continuous space vs discrete space,” in *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, 2017, pp. 1–6.
- [34] A. Sgorbissa and R. Zaccaria, “Planning and obstacle avoidance in mobile robotics,” *Robotics and Autonomous Systems*, vol. 60, no. 4, pp. 628 – 638, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S092188901100234X>

- [35] R. F. Stengel, *Flight dynamics*, 2015.
- [36] B. L. Stevens and F. L. Lewis, “Aircraft Control and Simulation,” *Aircraft Engineering and Aerospace Technology*, 2004.
- [37] J. Tao, C. Zhong, L. Gao, and H. Deng, “A study on path planning of unmanned aerial vehicle based on improved genetic algorithm,” in *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 02, 2016, pp. 392–395.
- [38] R. Volpe, “Real and artificial forces in the control of manipulators: Theory and experiments,” 02 2001.
- [39] M. N. A. Wahab, S. Nefti-Meziani, and A. Atyabi, “A comparative review on mobile robot path planning: Classical or meta-heuristic methods?” *Annual Reviews in Control*, 2020. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1367578820300675>
- [40] D. Xin, C. Hua-hua, and G. Wei-kang, “Neural network and genetic algorithm based global path planning in a static environment,” *Journal of Zhejiang University SCIENCE*, vol. 6, pp. 549–554, 06 2005.
- [41] W. C. Yang Wang, “Path planning and obstacle avoidance of unmanned aerial vehicle based on improved genetic algorithms,” in *Proceedings of the 33rd Chinese Control Conference, CCC 2014*, 2014.
- [42] C. Yin, Z. Xiao, X. Cao, X. Xi, P. Yang, and D. Wu, “Offline and online search: Uav multiobjective path planning under dynamic urban environment,” *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 546–558, 2018.
- [43] Z. Yingkun, “Flight path planning of agriculture uav based on improved artificial potential field method,” in *2018 Chinese Control And Decision Conference (CCDC)*, 2018, pp. 1526–1530.

RESUMEN AUTOBIOGRÁFICO

Manuel Alejandro Gutiérrez Martínez

Candidato para obtener el grado de
Maestría en Ingeniería Aeronáutica
con Orientación en Estructuras

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

DISEÑO DE ALGORITMO GENETICO PARA PLANEACIÓN DE
TRAYECTORIA DE UN UAV

El Ing. Manuel Alejandro Gutierrez Martinez nació el 11 de Julio de 1994 en la ciudad de Monterrey, Nuevo León, México. Hijo de Juan Manuel Gutierrez Izaguirre y María del Carmen martinez Montemayor.

Concluyó sus estudios en la Preparatoria técnica EIAO en el año 2012

Inicio sus estudios de licenciatura en Ingeniería en Mecatrónica, en la Facultad de ingeniería Mecánica y Eléctrica en el año 2013, finalizo satisfactoriamente sus estudios en tiempo y forma en el año 2018, donde se desempeñó en el área de máquinas inteligentes y tuvo la oportunidad de hacer su estancia de investigación en el área en el Centro de Investigación e Innovación en Ingeniería Aeronáutica.

Trabajo en el año 2018 en la empresa Hielmex en el cargo de control de producción.

En el año 2019 inicio sus estudios en la Maestría de aeronáutica, donde tuvo la oportunidad de participar International Conference on Unmanned Aircraft Systems (ICUAS) 2020.