



Universidad Autónoma de Querétaro
Facultad de Ingeniería
Maestría en Ciencias en Inteligencia Artificial

**Adaptación del algoritmo 'Ant System' para evasión de obstáculos con
restricciones de posicionamiento local y global.**

Tesis

Que como parte de los requisitos para obtener el grado de
Maestro en Ciencias en Inteligencia Artificial

Presenta:

Emmanuel Rodríguez Díaz

Dirigido por:

Dr. Juan Manuel Ramos Arreguín

Dr. Juan Manuel Ramos Arreguín
Presidente

Dr. Jesús Carlos Pedraza Ortega
Secretario

Dr. Marco Antonio Aceves Fernández
Vocal

Dr. Efrén Gorrostieta Hurtado
Suplente

Dr. Saúl Tovar Arriaga
Suplente

Centro Universitario Querétaro, QRO
México.
Septiembre 2019

Dirección General de Bibliotecas UAQ

Esta tesis esta dedicada a mi familia que siempre me ha apoyado en cada uno de mis proyectos personales, al igual que a la comunidad científica, que con cada pequeño esfuerzo procura mejorar nuestra sociedad.

Dirección General de Bibliotecas UAQ

Dirección General de Bibliotecas UAQ

Agradecimientos

Agradezco al CONACYT por el apoyo económico que me brindó durante los dos años de la maestría, mismo apoyo que dignificó mi estancia y me ayudó a concentrarme 100% en la maestría. Así mismo agradezco a la Universidad Autónoma de Querétaro por el apoyo e importancia que le otorga a la ciencia y a los posgrados en ciencias.

Dirección General de Bibliotecas UNQ

Dirección General de Bibliotecas UAQ

Resumen

La tesis que se muestra a continuación es un trabajo de investigación y desarrollo sobre planeación de trayectorias para evasión de obstáculos con un enfoque a vehículos autónomos, la investigación surge a partir de la problemática que existe actualmente sobre la gran cantidad de información que obtienen los vehículos autónomos y la dificultad de procesarla en poco tiempo.

En el desarrollo del trabajo se implementaron dos enfoques para solucionar el problema de planeación de trayectorias con restricciones locales y globales, el primer enfoque se basó en la simulación de campos magnéticos virtuales (CMV), mismo enfoque que a pesar de tener una buena efectividad al momento de generar trayectorias, el tiempo de solución fue demasiado grande por lo que el enfoque se consideró inviable.

Debido a lo anterior, se desarrolló el algoritmo *Magnetic Ants* (MA), un algoritmo que toma conceptos de CMV y *Ant System*, el algoritmo MA presentó mejoras considerablemente superiores a CMV en cuanto al tiempo de solución, por lo que los resultados finales se consideraron viables para su futura implementación en un vehículo autónomo de pruebas.

Dirección General de Bibliotecas UAQ

Abstract

The thesis shown below is a development and investigation work about path planning for obstacle avoidance with an approach to autonomous vehicles, the investigation comes from the problematic that exist nowadays related with the big amount of information that autonomous cars receive from its instrumentation and how hard it is to process all that information in a short period of time.

Two main approaches to solve path planning with global and local restrictions were implemented during the work development, the first approach was based on the simulation of Virtual Magnetic Fields (VMF), this approach had good results solving trajectories, but solution time was too big to be considerate feasible.

Given the problems with VMF, the algorithm 'Magnetic Ant'(MA) was developed, an algorithm that takes concepts from 'Ant System' and VMF. The algorithm MA show considerably superior improvements against VMF in time solution, therefore, the final results were considerate feasible for a future implementation in autonomous vehicles.

Índice general

Dedicación	
Agradecimientos	
Resumen	
Abstract	I
Contenido	II
Índice de Figuras	IV
Índice de Tablas	VIII
1. Introducción	1
1.1. Antecedentes	1
1.2. Descripción del problema	6
1.3. Justificación	7
1.4. Hipótesis y objetivos	7
1.4.1. Hipótesis	7
1.4.2. Objetivo general	8
1.4.3. Objetivos específicos	8
1.5. Estado del arte	8
2. Marco Teórico	12
2.1. Ingeniería de un vehículo autónomo	12
2.2. Planeación de trayectorias	13

2.3.	Algoritmo <i>Ant System (AS)</i>	15
2.4.	Campos Magnéticos Virtuales (CMV)	18
2.4.1.	Partícula de atracción	19
2.4.2.	Partícula de repulsión	20
2.4.3.	Sumatoria de campos magnéticos virtuales	22
3.	Metodología	26
3.1.	Solución mediante CMV's	26
3.1.1.	Proceso de solución	27
3.1.2.	Explicación técnica de Algoritmo Genético	34
3.1.3.	Limitaciones del algoritmo CMV con enfoque a planeación de trayectorias.	38
3.2.	Solución mediante algoritmo <i>Ant System</i> y CMV's	41
3.2.1.	Obtención de espacio muestreado en su forma discreta	42
3.2.2.	Definición de condiciones iniciales	44
3.2.3.	Creación de Vector <i>Tau</i> (τ)	48
3.2.4.	Ejecución de algoritmo MA	52
3.2.5.	Suavizado de la trayectoria	59
4.	Resultados y conclusiones	61
4.1.	Tiempos de solución y casos de estudio evaluados	61
4.2.	Resultados con <i>Hyundai i10 (HatchBack)</i>	62
4.3.	Resultados con (<i>SUV</i>) <i>Mazda CX-5</i>	64
4.4.	Resultados con (<i>Pick-Up</i>) <i>Mitsubishi L200</i>	66
4.5.	Conclusiones	68
	Referencias	71
	Apéndice	72

Índice de figuras

1.1. Estacionamiento automático (Paromtchik y Laugier, 1996).	2
1.2. Vehículo participante de <i>Grand Challenge</i> 2007.	3
1.3. Implementación de algoritmo ACS para planeación de trayectorias global (Mei et al., 2006).	3
1.4. Grafo inicial y posibles movimientos para el robot móvil (Guan-Zheng et al., 2007)	4
1.5. Optimización de resultados de Dijkstra mediante <i>Ant Colony</i> (Guan- Zheng et al., 2007).	5
1.6. Restricciones de algoritmo <i>Ant Colony</i> para planeación de trayecto- rias (Garcia et al., 2009).	5
1.7. Restricciones locales tomadas por imágenes satelitales (Marin-Plaza et al., 2018).	9
1.8. Proceso de planeación global (Chamzas et al., 2019).	10
1.9. Grafo de posibles movimientos para vehículo autónomo (Huang et al., 2019).	10
1.10. Espacio discretizado por vehículo autónomo (Huang et al., 2019). . .	11
2.1. Ingeniería de un vehículo autónomo.	12
2.2. Mapa de información binaria (Lee et al., 2014).	13
2.3. Ejemplo de planeación de trayectorias. (a) Con restricciones globales. (b) Con restricciones locales.	14
2.4. Solución con restricciones locales y globales.	15
2.5. Ejemplo de TSP con Ant System.	16

2.6. Solución al problema del Agente Viajero.	18
2.7. Partícula de atracción.	19
2.8. Partícula de repulsión.	21
2.9. Evasión de CMV.	24
2.10. Partícula de repulsión sin efecto sobre el robot móvil virtual.	24
3.1. Metodología para la implementación del algoritmo CMV.	26
3.2. Metodología para la asignación de casos de estudio.	27
3.3. Creación de caso de estudio.	28
3.4. Mapa para la asignación de campos magnéticos virtuales.	28
3.5. Asignación de campos magnéticos virtuales en forma gráfica.	28
3.6. Algoritmo para planeación de trayectorias mediante CMV's.	30
3.7. Trayectoria resultante con condiciones iniciales.	31
3.8. Algoritmo genético para búsqueda de mejor configuración.	32
3.9. Mejor trayectoria tomada de la población inicial.	33
3.10. Generación N°1.	33
3.11. Ejemplo para explicación técnica de algoritmo genético.	34
3.12. Tiempo de solución en función de número de obstáculos.	39
3.13. Solución con 1 obstáculo	40
3.14. Solución con 2 obstáculos	40
3.15. Solución con 3 obstáculos	40
3.16. Solución con 4 obstáculos	40
3.17. Metodología de solución mediante MA.	42
3.18. Ejemplo de entorno físico muestreado (Lee et al., 2014).	43
3.19. Interpretación y manejo de la información binaria.	43
3.20. Caso de estudio virtual en su forma discreta.	44
3.21. Vehículo virtual en espacio discreto.	45

3.22. Línea de objetivo.	46
3.23. Espacio muestreado con espacio de seguridad (Conceptual).	47
3.24. Espacio muestreado con espacio de seguridad (Caso de estudio).	48
3.25. Posibles movimientos.	49
3.26. Ejemplo de obtención de valor de posición en vector Tau	51
3.27. Algoritmo MA	53
3.28. Trayectoria generada por algoritmo MA	54
3.29. Valor de cada movimiento en MA	55
3.30. Constante Alpha = 30	56
3.31. Constante Alpha = 32.5	56
3.32. Constante Alpha = 33.855	57
3.33. Caso de colisión por elección de movimiento.	57
3.34. Trayectoria suavizada mediante solución a matriz Vandermonde.	60
4.1. Tiempos de solución en diferentes casos de estudio.	62
4.2. Caso de estudio 1 con (<i>HatchBack</i>) <i>Hyundai i10</i>	63
4.3. Caso de estudio 2 con (<i>HatchBack</i>) <i>Hyundai i10</i>	63
4.4. Caso de estudio 3 con (<i>HatchBack</i>) <i>Hyundai i10</i>	63
4.5. Caso de estudio 4 con (<i>HatchBack</i>) <i>Hyundai i10</i>	63
4.6. Caso de estudio 5 con (<i>HatchBack</i>) <i>Hyundai i10</i>	64
4.7. Caso de estudio 1 con (<i>SUV</i>) <i>Mazda CX-5</i>	64
4.8. Caso de estudio 2 con (<i>SUV</i>) <i>Mazda CX-5</i>	65
4.9. Caso de estudio 3 con (<i>SUV</i>) <i>Mazda CX-5</i>	65
4.10. Caso de estudio 4 con (<i>SUV</i>) <i>Mazda CX-5</i>	65
4.11. Caso de estudio 5 con (<i>SUV</i>) <i>Mazda CX-5</i>	65
4.12. Caso de estudio 1 con (<i>Pick-Up</i>) <i>Mitsubishi L200</i>	66
4.13. Caso de estudio 2 con (<i>Pick-Up</i>) <i>Mitsubishi L200</i>	66

4.14. Caso de estudio 3 con <i>(Pick-Up) Mitsubishi L200</i>	67
4.15. Caso de estudio 4 con <i>(Pick-Up) Mitsubishi L200</i>	67
4.16. Caso de estudio 5 con <i>(Pick-Up) Mitsubishi L200</i>	67

Dirección General de Bibliotecas UAQ

Índice de tablas

1.1. Antecedentes sobre planeación de trayectorias	6
1.2. Estado del arte sobre planeación de trayectorias	11
2.1. Constantes aplicadas a algoritmo AS para la solución del Agente Via- jero.	17
3.1. Restricciones locales y globales tomadas en la solución a planeación de trayectorias con enfoque de CMV's	27
3.2. Valores iniciales para variables de obstáculos y objetivo.	29
3.3. Valores finales para variables de obstáculos y objetivo.	33
3.4. Valores iniciales para variables del obstáculo y el objetivo.	34
3.5. Población inicial de caso de estudio.	35
3.6. Aptitud de población inicial de caso de estudio de forma ordenada. . .	36

Introducción

1.1. Antecedentes

La planeación de trayectorias se ha utilizado para resolver diferentes problemas, tales como 'El agente viajero' en donde el objetivo es encontrar la ruta más corta para visitar múltiples puntos en un mapa, diseño de trayectorias para robots manipuladores en donde el objetivo es reducir el consumo de energía del robot y en el caso particular de este trabajo, planeación de trayectorias en robots móviles para la evasión de obstáculos.

A inicios de los 90's la planeación de trayectorias en robots móviles se realizaba dentro de entornos controlados y estáticos, en aplicaciones como aspiradoras autónomas, tal y como se muestra en el trabajo realizado por (Hofner y Schmidt, 1995).

El avance en tecnologías de instrumentación para reconocimiento bidimensional y tridimensional a partir de 1995 permitió el desarrollo de aplicaciones para vehículos autónomos. Como ejemplo, tenemos la aplicación para asistir al conductor en la tarea de estacionar un vehículo desarrollado por (Paromtchik y Laugier, 1996), tal y como se muestra en la figura 1.1.

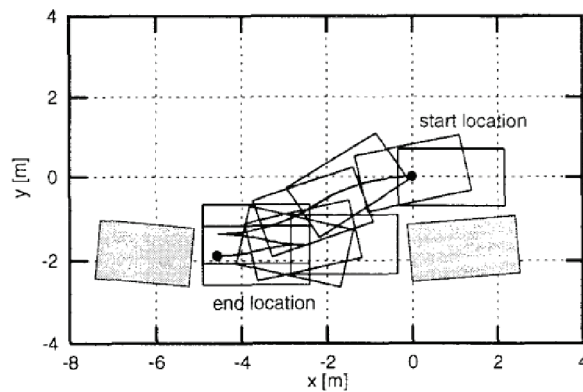


Figura 1.1: Estacionamiento automático (Paromtchik y Laugier, 1996).

Hasta este punto, la planeación de trayectorias no otorgaba una serie de puntos por los cuales dirigir al robot móvil o vehículo autónomo. Esto quiere decir, que en lugar de generar una trayectoria de un punto de origen a un punto destino, la trayectoria se creaba en función de la orientación que debía seguir el robot móvil y la cercanía con los obstáculos que lo rodeaban.

Los avances tecnológicos no han sido los únicos responsables del desarrollo científico en vehículos autónomos y de algoritmos para planeación de trayectorias.

En el año 2004 la agencia de investigación en proyectos avanzados (DARPA¹ por sus siglas en inglés) de los Estados Unidos de América organizó el primer *Grand Challenge* y el *Urban Challenge* en el año 2007.

Ambos concursos fueron diseñados para impulsar el desarrollo de vehículos autónomos. Desde el año 2004 el *Grand Challenge* ha impulsado el desarrollo de diferentes técnicas de instrumentación, al igual que técnicas de autonomía. En la figura 1.2 se puede observar un vehículo participante del *Grand Challenge* 2007.

¹Defense Advanced Research Projects Agency



Figura 1.2: Vehículo participante de *Grand Challenge 2007*.

A partir del año 2006, algoritmos metaheurísticos como *Ant Colony* y Algoritmos Genéticos comenzaron a aportar al problema de planeación de trayectorias de forma exitosa.

La primera implementación de *Ant Colony* al problema de planeación de trayectorias fue introducido por (Mei, Tian, y Zu, 2006). En el trabajo realizado se utiliza el algoritmo *Ant Colony System* (ACS) para realizar la planeación global dentro de un espacio discretizado con diferentes niveles de complejidad y sin restricciones locales como se muestra en la figura 1.3.



Figura 1.3: Implementación de algoritmo ACS para planeación de trayectorias global (Mei et al., 2006).

Más adelante, en el trabajo de (Guan-Zheng, Huan, y Sloman, 2007), el algo-

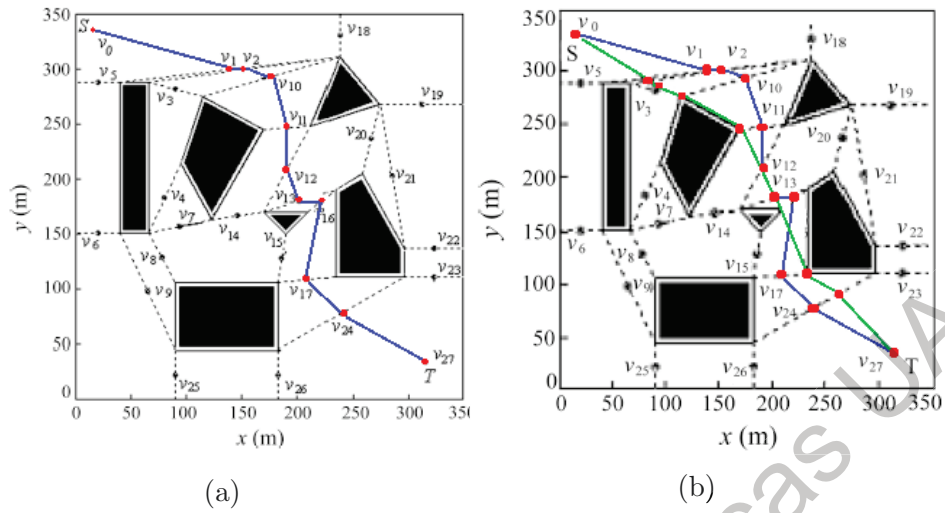


Figura 1.5: Optimización de resultados de Dijkstra mediante *Ant Colony* (Guan-Zheng et al., 2007).

En el trabajo realizado por (García, Montiel, Castillo, Sepúlveda, y Melin, 2009) se logró adaptar el algoritmo ACS con la variante de que en lugar de dejar un rastro de feromonas, como tradicionalmente se realiza en *Ant Colony*, el movimiento de las hormigas es influenciado por reglas difusas basadas en la cercanía con los obstáculos. En la figura 1.6 se muestran gráficamente las restricciones del algoritmo.

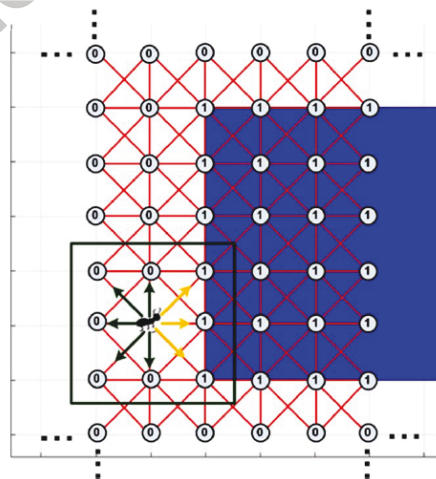


Figura 1.6: Restricciones de algoritmo *Ant Colony* para planeación de trayectorias (García et al., 2009).

Hasta aquí el estado del arte, como se puede observar las mateheurísticas como *Ant Colony* se han ido adaptando al problema de planeación de trayectorias obteniendo en cada nueva investigación mejores resultados. En la tabla 1.1 se resume la información sobre los antecedentes.

Resumen de antecedentes	
Autor/Organización-Año	Aportación
(Hofner y Schmidt, 1995)	Planeación de trayectorias en entornos controlados
(Paromtchik y Laugier, 1996)	Estacionamiento automático
DARPA (2004)	<i>Grand Challenge</i>
DARPA (2007)	<i>Urban Challenge</i>
(Mei et al., 2006)	Primer adaptación de <i>Ant Colony</i> a planeación de trayectorias.
(Guan-Zheng et al., 2007)	Optimización de Dijkstra mediante ACS
(Garcia et al., 2009)	Adaptación de CS con lógica difusa

Tabla 1.1: Antecedentes sobre planeación de trayectorias

1.2. Descripción del problema

Actualmente la investigación enfocada a desarrollo de algoritmos para evasión de obstáculos se encamina a procesar la mayor cantidad de información posible en el menor tiempo posible.

Sin embargo los algoritmo que existen hoy en día no han sido adaptados para aprovechar toda la información otorgada por la instrumentación de los vehículos autónomos.

A partir de esta problemática surge el trabajo que se muestra a continuación En donde el objetivo es adaptar un algoritmo que sea capaz de procesar información proveniente de restricciones locales y globales en un lapso de tiempo que sea viable para un vehículo autónomo o robot móvil.

1.3. Justificación

Una integración entre planeación de trayectorias local y global enfocada a vehículos autónomos puede aumentar la seguridad con la que circula un vehículo autónomo en situaciones de riesgo.

Un vehículo autónomo se encuentra constantemente con situaciones en las que evadir un obstáculo puede representar un riesgo de colisión, o desvió de su carril. Mientras más información de su entorno sea capaz de procesar, menor será el riesgo de colisión.

En los trabajos más actuales, mismos que se mencionan en el estado del arte, no existe una integración de restricciones locales y globales en los algoritmos de planeación de trayectorias. El trabajo que presento contribuye a la base de conocimiento en una adaptación de algoritmos de inteligencia artificial para la integración de restricciones locales y globales en la planeación de trayectorias.

1.4. Hipótesis y objetivos

1.4.1. Hipótesis

”Un vehículo autónomo es capaz de evadir obstáculos para evitar colisiones con mayor número de casos de éxito si se consideran restricciones de posicionamiento local y global a diferencia de considerar solo restricciones locales dentro del algoritmo que genera las trayectorias para evasión de obstáculos”

1.4.2. Objetivo general

Adaptar el algoritmo *Ant System* para resolver el problema de planeación de trayectorias con restricciones de posicionamiento local y global.

1.4.3. Objetivos específicos

1. Diseñar e implementar un entorno de simulación con el objetivo de realizar pruebas de planeación de trayectorias para evasión de obstáculos.
2. Utilizar el entorno de simulación previamente diseñado para realizar pruebas de simulación con el objetivo de adaptar el algoritmo *Ant System* al problema de planeación de trayectorias con restricciones locales y globales.
3. Obtener resultados de trayectorias generadas con restricciones locales y globales con la adaptación del algoritmo *Ant System* y valorar si los resultados pueden ser implementados en una aplicación real a vehículos autónomos.

1.5. Estado del arte

Los trabajos más actuales en el área de planeación de trayectorias para vehículos autónomos están enfocados en desarrollar nuevas técnicas para planeación de trayectorias que mejoren dos áreas principales: tiempo de solución y cantidad de información procesada.

Para mejorar el tiempo de solución se han explorado metaheurísticas como *Ant Colony* y Algoritmos Genéticos con el objetivo de disminuir el tiempo de convergencia al reducir el universo de soluciones evaluadas.

Para aumentar la cantidad de información procesada, se han desarrollado técnicas que unan la planeación de trayectorias local con la global, con el objetivo de obtener soluciones que cumplan con ambas restricciones (locales y globales), tomando en cuenta que la solución representa una trayectoria o serie de puntos por los cuales un vehículo autónomo debe pasar sin colisionar con su entorno.

El trabajo realizado por (Marin-Plaza, Hussein, Martin, y Escalera, 2018) muestra una técnica que integra la planeación local al usar imágenes satelitales para guiar al robot en su entorno local, y fusiona dicha información con la obtenida por la instrumentación del vehículo de pruebas.

Dicho trabajo se diferencia con el trabajo que se realizó para esta tesis en que las restricciones que se tomaron en cuenta no están enfocadas a vehículos autónomos. El proceso de discretización satelital se muestra en la figura 1.7.

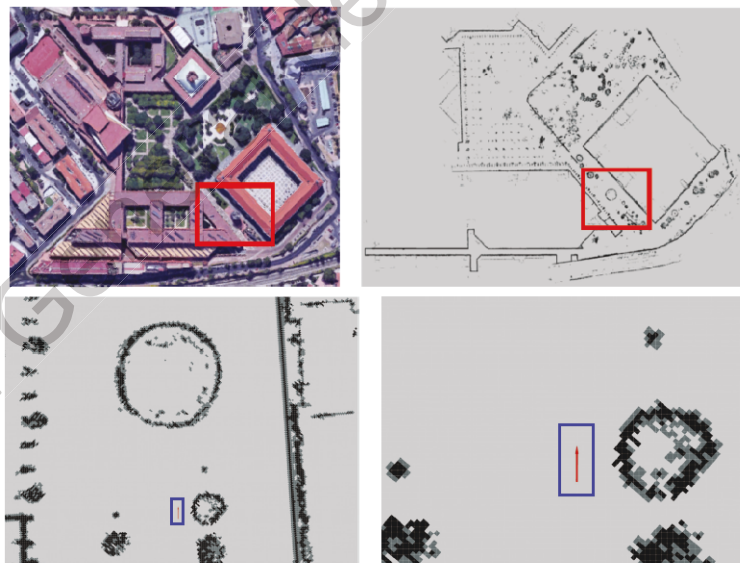


Figura 1.7: Restricciones locales tomadas por imágenes satelitales (Marin-Plaza et al., 2018).

Más recientemente el trabajo de (Chamzas et al., 2019) desarrollo una técnica

de planeación global que divide el mismo entorno global en pequeños sub-entornos locales, mismos que son comparados con una base de datos con soluciones previamente realizadas y de esta manera generar la planeación global al unir las soluciones por partes. Dicho proceso se muestra en la figura 1.8.

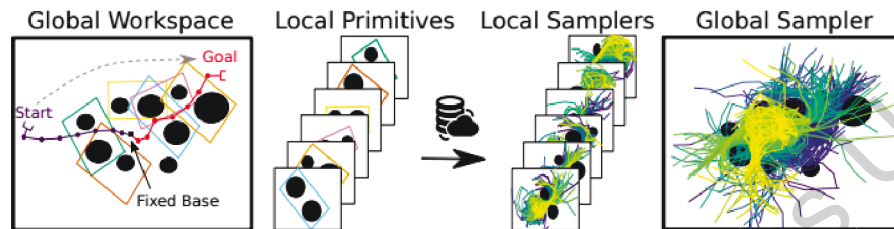


Figura 1.8: Proceso de planeación global (Chamzas et al., 2019).

Aunque el trabajo realizado por (Chamzas et al., 2019) presenta una aplicación diferente a la que se trabajó en este proyecto, el uso de información obtenida previamente por el robot móvil no se descarta en vehículos autónomos ya que estos también circulan por entornos muy similares en la mayoría de los casos.

Finalmente el trabajo realizado por (Huang et al., 2019) propone un nuevo método de planeación de trayectorias basado en la asignación de Campos Magnéticos Virtuales (CMV) a los obstáculos y a los límites del camino, la información otorgada por los campos magnéticos virtuales es asignada a cada arista dentro del grafo de posibles movimientos, mismo grafo que se muestra en la figura 1.9.

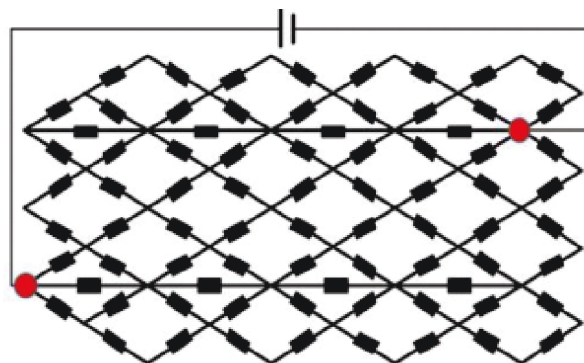


Figura 1.9: Grafo de posibles movimientos para vehículo autónomo (Huang et al., 2019).

En donde los puntos marcados en rojo en la figura 1.9 representan el punto de origen y el punto de destino, dicho grafo representa un espacio discretizado del camino, como se muestra en la figura 1.10.

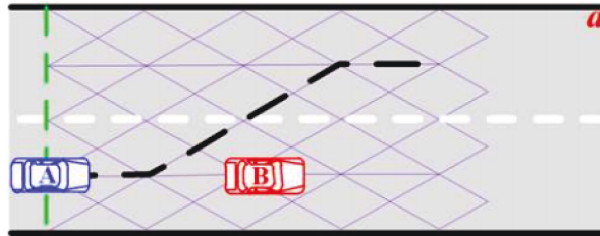


Figura 1.10: Espacio discretizado por vehículo autónomo (Huang et al., 2019).

A pesar de que el trabajo realizado por (Huang et al., 2019) muestra resultados positivos existen un par de problemas que no se tratan dentro del trabajo realizado, los cuáles son la asignación de campos CMV a geometrías complejas y la caída en mínimos locales debido al los pocos movimientos posibles en el grafo, mismos problemas que son tratados en el trabajo presentado en esta tesis.

Los trabajos mostrados en el estado de la arte se resumen en la tabla 1.2.

Resumen de estado del arte	
Autor/Organización-Año	Aportación
(Marin-Plaza et al., 2018)	Planeación de trayectorias global mediante imágenes satelitales
(Chamzas et al., 2019)	Planeación de trayectorias global con ayuda de entornos locales previamente resueltos
(Huang et al., 2019)	Asignación de campos CMV a obstáculos y límites del camino para planeación de trayectorias local.

Tabla 1.2: Estado del arte sobre planeación de trayectorias

Marco Teórico

2.1. Ingeniería de un vehículo autónomo

Actualmente la ingeniería de un vehículo autónomo se divide en tres partes principales, tal y como se muestra en la figura 2.1.

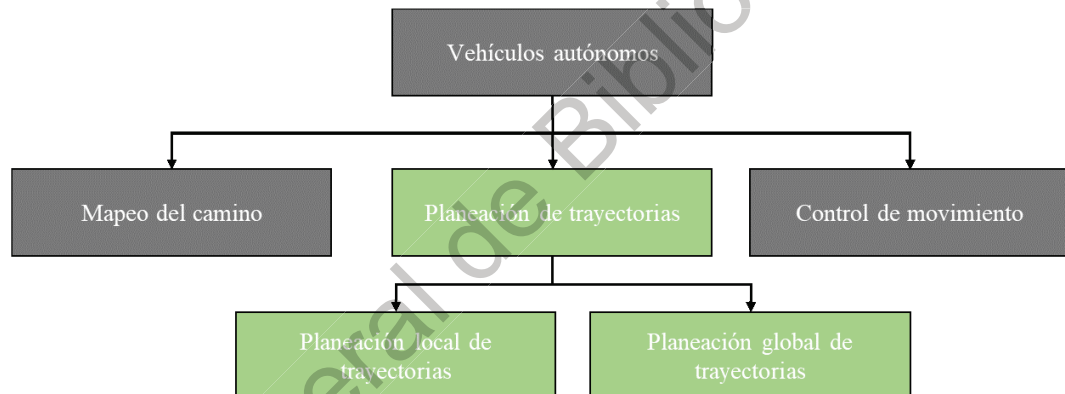


Figura 2.1: Ingeniería de un vehículo autónomo.

Como se puede observar en la figura (2.1) el proceso con el que un vehículo autónomo empieza su función de moverse de un punto de origen a un punto de llegada comienza con el reconocimiento de su entorno, dicho proceso se realiza mediante sensores ultrasónicos, cámaras o sensores LIDAR¹, en la figura 2.2 se muestra un mapa de información binaria obtenido con tecnología LIDAR.

¹Light Detection and Ranging o Laser Imaging Detection and Ranging

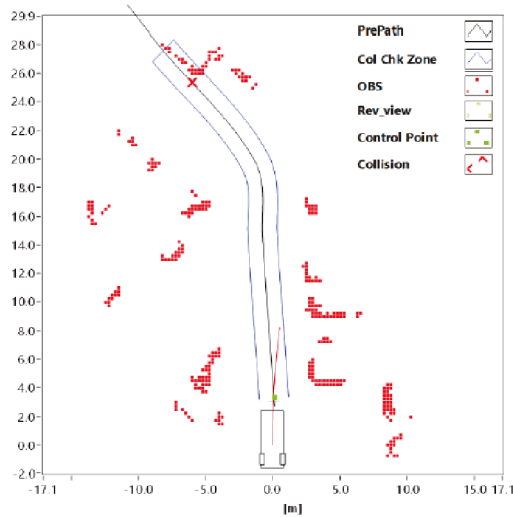


Figura 2.2: Mapa de información binaria (Lee et al., 2014).

2.2. Planeación de trayectorias

Una vez que se tiene una imagen del entorno que rodea al vehículo autónomo se pasa a la etapa de planeación de trayectorias, misma etapa que representa la parte central de dicha tesis.

La planeación de trayectorias se divide a su vez en dos partes, planeación de trayectorias local y global

De acuerdo a la definición de (Lee, Yoon, Shim, Vasseur, y Demonceaux, 2014), la planeación de trayectorias local incluye la información obtenida por la instrumentación del vehículo autónomo, información como los obstáculos alrededor del vehículo o las líneas del carril que tiene que seguir. Por otro lado, la planeación de trayectorias global incluye información del trayecto previamente definido, información como la ruta a seguir y la orientación que debe tener el vehículo para seguir dicha ruta.

La gran mayoría de técnicas de planeación de trayectorias se adapta solamente a la planeación de trayectorias global ya que este tipo de planeación de trayectorias no requiere que se encuentre un solución en poco tiempo como es el caso de la planeación de trayectorias local.

La planeación de trayectorias local se tiene que realizar en poco tiempo ya que la solución se tiene que adaptar a cambios en tiempo real en el entorno, por lo tanto, la evasión de obstáculos es una parte central de la planeación de trayectorias local.

La figura 2.3 muestra gráficamente la diferencia entre planeación de trayectorias local y global, en la figura 2.3a se muestra una trayectoria solo considerando restricciones globales, mientras que en la figura 2.3b se muestra una trayectoria considerando solo restricciones locales.

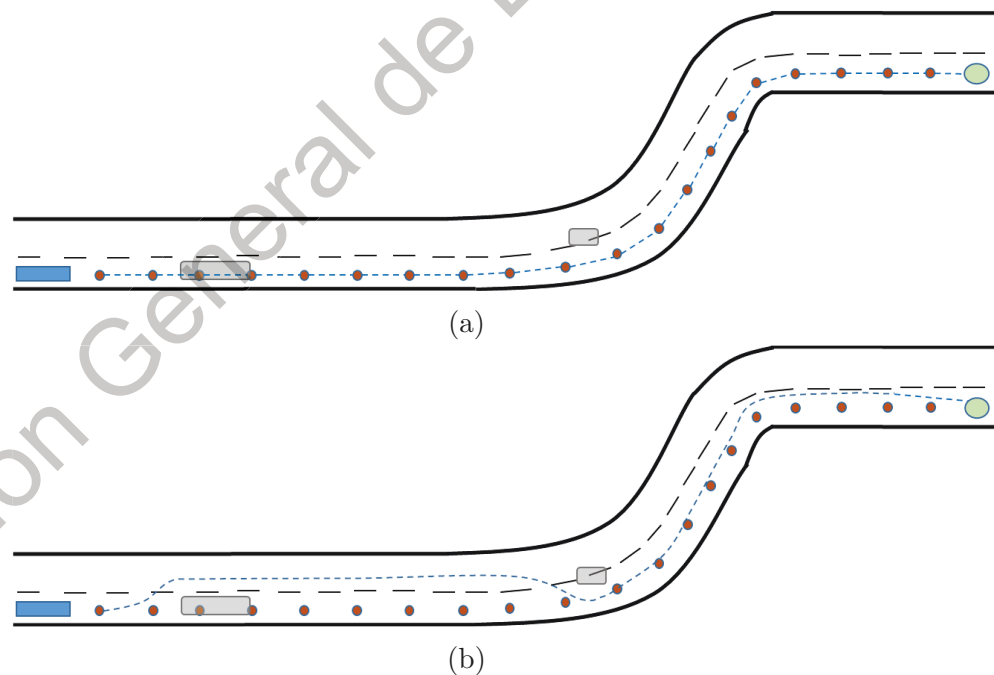


Figura 2.3: Ejemplo de planeación de trayectorias. (a) Con restricciones globales. (b) Con restricciones locales.

Mientras que el objetivo de la tesis es desarrollar una técnica que adapte la

solución a ambas restricciones, tanto locales como globales, tal y como se muestra en la figura 2.4.

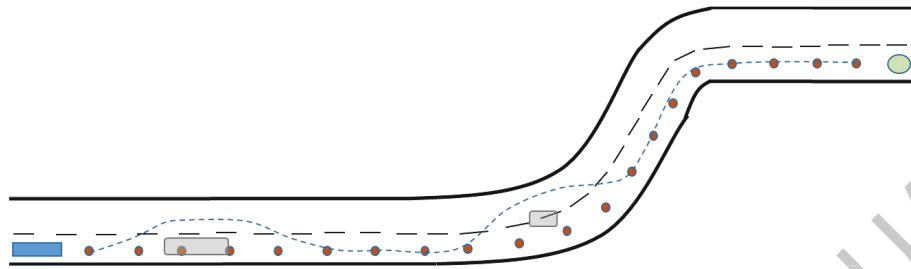


Figura 2.4: Solución con restricciones locales y globales.

2.3. Algoritmo *Ant System* (AS)

Debido a que la parte central de la tesis esta basada en la adaptación del algoritmo AS al problema de planeación de trayectorias local y global, se dará un repaso sobre dicho algoritmo, al mencionar como funciona el algoritmo de forma conceptual y las matemáticas que forman el algoritmo.

El algoritmo *Ant System* fue introducido por (Dorigo, Maniezzo, y Coloni, 1996), dicho algoritmo simula el comportamiento de las hormigas al buscar comida.

Para explicar el algoritmo *Ant System* se tomará como ejemplo el problema del Agente Viajero, misma introducción que se muestra en (Dorigo y Socha, 2006), un problema en el que hay que encontrar la ruta más corta para recorrer un número n de ciudades tal y como se muestra en la figura 2.5.

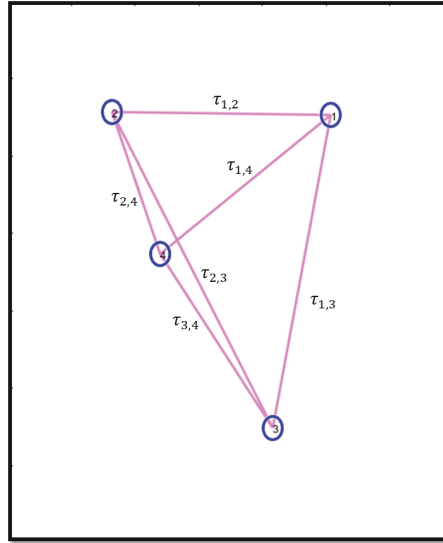


Figura 2.5: Ejemplo de TSP con Ant System.

La principal característica del algoritmo *AS* es que el valor de la feromona es actualizado por cada una de las hormigas que completan el trayecto. La actualización de la feromona para τ_{ij} , que es, el valor de la feromona asignado a la arista que une a las ciudades i y j , se realiza de la siguiente manera:

$$\tau_{ij} \leftarrow (1 - p) \cdot \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (2.1)$$

en donde p es la velocidad de evaporación, m es el número de hormigas, y $\Delta\tau_{ij}^k$ es la cantidad de feromona por unidad de longitud que se deja sobre la arista (i, j) por la hormiga número k :

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{si la hormiga } k \text{ recorrió la arista } (i, j) \text{ en su trayecto} \\ 0 & \text{en el caso contrario} \end{cases} \quad (2.2)$$

en donde Q es una constante y L_k es la longitud obtenida por la hormiga número k .

Al momento de que las hormigas realizan su trayecto, es necesario un proceso de toma de decisión basado en la probabilidad de moverse de la ciudad i a la ciudad j , dicha probabilidad P_{ij}^k se obtiene de la siguiente manera:

$$P_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{l \in \text{permitidos}_k} \tau_{ij}^\alpha \cdot \eta_{ij}^\beta} & \text{si } j \in \text{permitidos}_k \\ 0 & \text{en el caso contrario} \end{cases} \quad (2.3)$$

en donde permitidos_k es la lista de las ciudades que aun no has sido visitadas por la hormiga número k , y α y β son parámetros de control que representan la importancia relativa de la feromona contra la información heurística η_{ij}^β dada por:

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (2.4)$$

en donde d_{ij} es la distancia que existe del nodo i al nodo j .

En la figura 2.6 se muestran los resultados obtenidos por el algoritmo AS aplicado al caso mostrado en la figura 2.5, tomando las constantes que se muestran en la tabla

Constante	Valor
Número de ciudades	4
Número de hormigas	10
Valor de feromona inicial	0.5
α	1
β	5

Tabla 2.1: Constantes aplicadas a algoritmo AS para la solución del Agente Viajero.

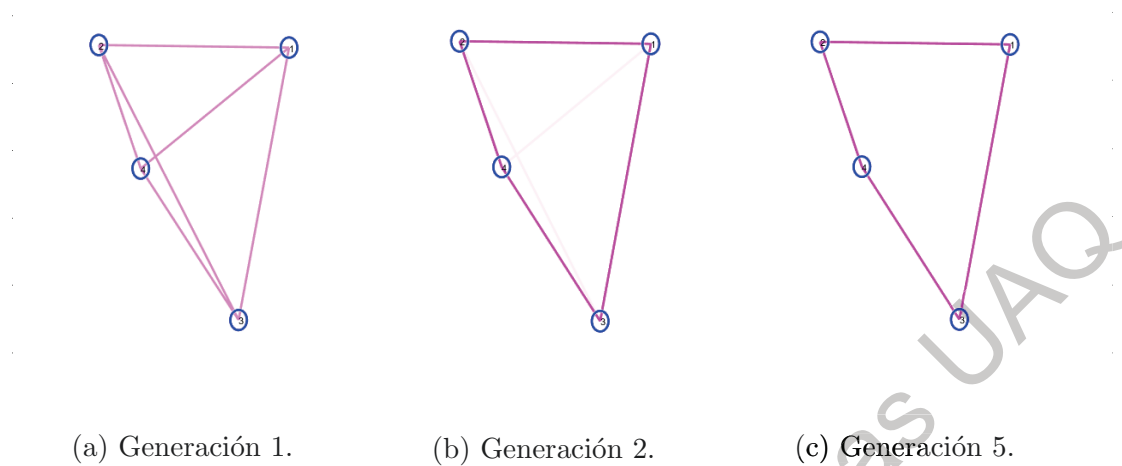


Figura 2.6: Solución al problema del Agente Viajero.

2.4. Campos Magnéticos Virtuales (CMV)

En el trabajo que se desarrollo fue necesario buscar una forma de ayudar al algoritmo AS a converger de la manera mas rápida posible, para lograrlo se utilizo el algoritmo CMV mismo que se utilizo como remplazo de la feromona dejada por la hormigas , de tal forma que las hormigas son guiadas por una fuerza magnética virtual para lograr una convergencia más rápida.

A continuación se explica el algoritmo CMV con la aplicación de planeación de trayectorias.

El algoritmo está basado en la simulación de partículas magnéticas, mismas partículas que se conforman por las siguientes variables: Alcance(α), Intensidad(β) y Radio(γ). Existen dos tipos de partículas magnéticas, de atracción y de repulsión.

2.4.1. Partícula de atracción

La partícula de atracción en el entorno global se define como la coordenada a la que tiene que llegar el robot móvil. Por lo tanto, sin importar en qué punto del plano se encuentre el robot móvil tiene que ser atraído por dicha partícula tal y como se muestra en la figura

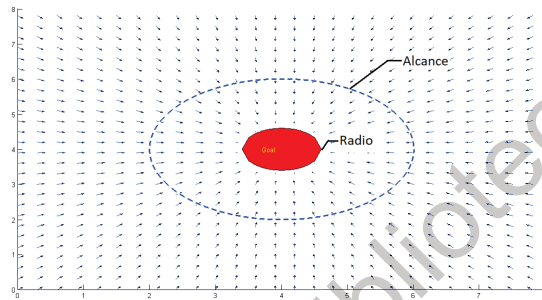


Figura 2.7: Partícula de atracción.

Mediante una ecuación gradiente se determina como afecta la partícula de atracción al robot móvil para así determinar en cada iteración el siguiente punto en el plano $[x, y]$ de tal forma que la posición en la iteración $P_x(i + 1)$ y $P_y(i + 1)$ y (siendo $[P_x, P_y]$ el punto en el que se encuentra el robot móvil) se define de la siguiente manera:

$$\begin{aligned} P_x(i + 1) &= P_x(i) + \Delta A_x \\ P_y(i + 1) &= P_y(i) + \Delta A_y \end{aligned} \tag{2.5}$$

La ecuación gradiente $[\Delta A_x, \Delta A_y]$ se genera con restricciones de acuerdo a la cercanía que se tiene con la partícula de atracción:

1. Gradiente estando fuera del Alcance (α_A)

$$\Delta A_x = \beta_A \alpha_A \cos(\theta_A)$$

$$\Delta A_y = \beta_A \alpha_A \sin(\theta_A)$$

2. Gradiente estando entre el Alcance (α_A) y el Radio (γ_A)

$$\Delta A_x = \beta_A (d_A - \alpha_A) \cos(\theta_A)$$

$$\Delta A_y = \beta_A (d_A - \alpha_A) \sin(\theta_A)$$

3. Gradiente estando dentro del Radio (γ_A)

$$\Delta A_x = 0$$

$$\Delta A_y = 0$$

en donde d_A representa la distancia entre el robot móvil y la partícula de atracción, y θ_A representa el ángulo entre el robot móvil y la partícula de atracción:

$$\begin{aligned} d_A &= \sqrt{(P_x - x_A)^2 + (P_y - y_A)^2} \\ \theta_A &= \arctan\left(\frac{y_A - P_y}{x_A - P_x}\right) \end{aligned} \quad (2.6)$$

En donde (x_A, y_A) representan las coordenadas de la partícula de atracción en el plano (x, y) .

2.4.2. Partícula de repulsión

La partícula de repulsión en el entorno local se define como un obstáculo que se interpone en el trayecto del robot móvil. Por lo tanto, su objetivo es desviar al robot móvil de su trayectoria si es que el robot está en el alcance de dicha partícula, como se muestra en la figura 2.8.

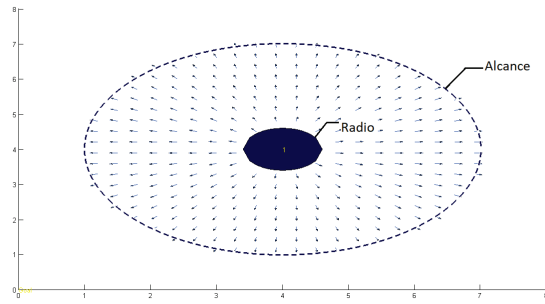


Figura 2.8: Partícula de repulsión.

A diferencia de la partícula de atracción, la partícula de repulsión solo tendrá efecto sobre la posición del robot móvil si es que este entra de su alcance previamente definido.

Mediante una ecuación gradiente se determina como afecta la partícula de repulsión al robot móvil para así determinar en cada iteración el siguiente punto en el plano $[x, y]$ de tal forma que la posición en la iteración $P_x(i + 1)$ y $P_y(i + 1)$ se define de la siguiente manera:

$$\begin{aligned} P_x(i + 1) &= P_x(i) + \Delta R_x \\ P_y(i + 1) &= P_y(i) + \Delta R_y \end{aligned} \tag{2.7}$$

La ecuación gradiente $(\Delta R_x, \Delta R_y)$ se genera con restricciones de acuerdo a la cercanía que se tiene con la partícula de repulsión:

1. Gradiente estando fuera del Alcance (α_R)

$$\Delta R_x = 0$$

$$\Delta R_y = 0$$

2. Gradiente estando entre el Alcance (α_R) y el Radio (γ_R)

$$\Delta R_x = -\beta_A(d_A - \alpha_A)\cos(\theta_A)$$

$$\Delta R_y = -\beta_A(d_A - \alpha_A)\sin(\theta_A)$$

3. Gradiente estando dentro del Radio (γ_R)

$$\Delta R_x = \infty$$

$$\Delta R_y = \infty$$

en donde d_R representa la distancia entre el robot móvil y la partícula de repulsión, y θ_R representa el ángulo entre el robot móvil y la partícula de repulsión:

$$d_R = \sqrt{(P_x - x_R)^2 + (P_y - y_R)^2} \quad (2.8)$$

$$\theta_R = \arctan\left(\frac{y_R - P_y}{x_R - P_x}\right)$$

En donde (x_R, y_R) representan las coordenadas de la partícula de repulsión en el plano (x, y) .

2.4.3. Sumatoria de campos magnéticos virtuales

La orientación correcta para el robot móvil se obtiene mediante la sumatoria del CMV de atracción (destino) y los CMV's de repulsión (obstáculos):

$$\Delta_x = \Delta A_x + \sum_{i=1}^n \Delta R x_i \quad (2.9)$$

$$\Delta_y = \Delta A_y + \sum_{i=1}^n \Delta R y_i$$

La distancia entre la posición actual ($P(i)$) del vehículo virtual y su posición en la iteración $P(i + 1)$ se define de la siguiente manera:

$$d_{P_1P_2} = \sqrt{\Delta_x^2 + \Delta_y^2} \quad (2.10)$$

Mientras que el ángulo que se forma entre el punto $P(i)$ y el punto $P(i + 1)$ se define de la siguiente manera:

$$\theta_{P_1P_2} = \arctan\left(\frac{\Delta_x}{\Delta_y}\right) \quad (2.11)$$

En la mayoría de los casos será necesario definir una distancia máxima (d_{MAX}) entre el punto $P(i)$ y el punto $P(i + 1)$, para estos casos será necesario redimensionar Δ_x y Δ_y de la siguiente manera :

$$\begin{aligned} \Delta_x &= d_{MAX} \sin(\theta) \\ \Delta_y &= d_{MAX} \cos(\theta) \end{aligned} \quad (2.12)$$

Se puede llegar a tener más de una partícula de repulsión, pero no más de una partícula de atracción, ya que puede haber múltiples obstáculos, pero no múltiples objetivos.

Dentro del plano en el que influyen los CMV's, el robot móvil debe de ser capaz de llegar al objetivo (partícula de atracción) sin colisionar con el obstáculo (partícula de repulsión) desde cualquier punto del plano (x, y) tal y como se muestra en la figura 2.9.

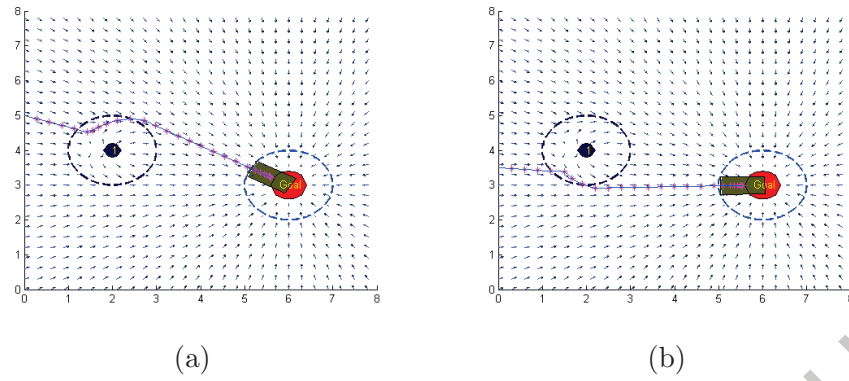


Figura 2.9: Evasión de CMV.

Al igual que puede haber obstáculos que no tengan ningún tipo de efecto sobre la trayectoria del robot móvil tal y como se muestra en la figura 2.10.

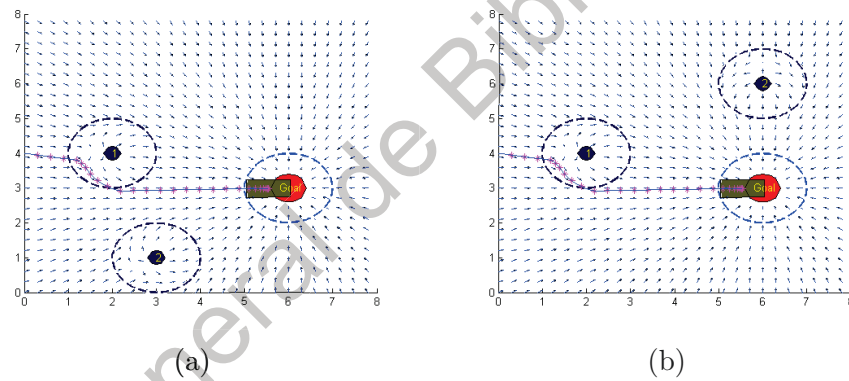


Figura 2.10: Partícula de repulsión sin efecto sobre el robot móvil virtual.

La tesis desarrollada utiliza fundamentos teóricos de la información que se ha mostrado en este capítulo, el siguiente capítulo, mismo describe la metodología y el desarrollo de la tesis, se muestran referencias a este capítulo para mostrar como es que se adaptaron los algoritmos CMV y *Ant System* al algoritmo final para planeación de trayectorias con restricciones locales y globales.

El tesis paso por dos etapas principales, la primera se basó en resolver el problema a planeación de trayectorias con restricciones locales y globales mediante

CMV's. Al analizar los resultados y ver la limitaciones del algoritmo se decidió realizar un nuevo enfoque, mismo que se baso en la adaptación del algoritmo 'Ant System' con la diferencia de que el rastro de feromona sería depositado por una partícula magnética virtual en lugar de ser depositado por el trayecto de las hormigas como lo indica el algoritmo 'Ant System'.

Dirección General de Bibliotecas UAQ

Metodología

3.1. Solución mediante CMV's

El primer enfoque de solución al problema de planeación de trayectorias con restricciones locales y globales fue mediante CMV's, mismo algoritmo que es explicado en el Capítulo 2, la metodología de solución se muestra en la figura 3.1.

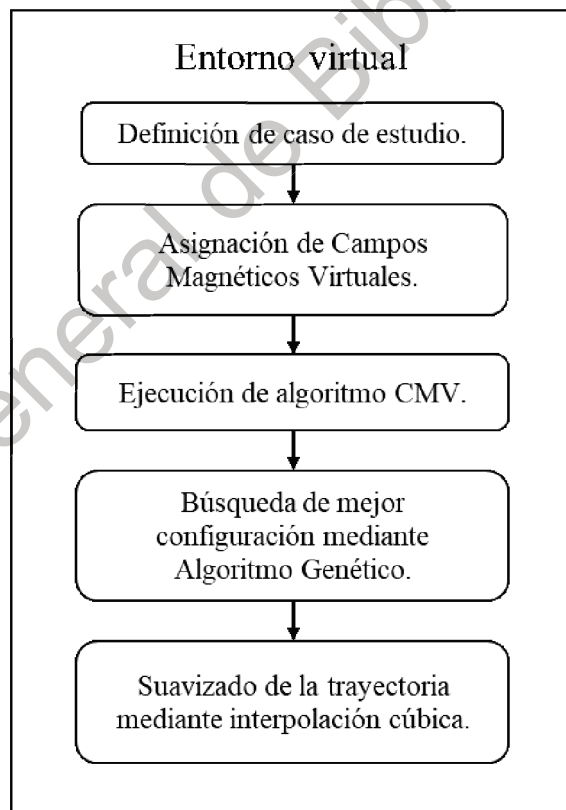


Figura 3.1: Metodología para la implementación del algoritmo CMV.

Mientras que en la tabla 3.1 se muestran las restricciones locales y globales

tomadas para el caso de estudio.

Restricciones locales	Restricciones globales
Geometría de obstáculos	Geometría del vehículo
Límites del camino	Orientación del camino
	Número de carriles

Tabla 3.1: Restricciones locales y globales tomadas en la solución a planeación de trayectorias con enfoque de CMV's

3.1.1. Proceso de solución

En el mapa conceptual que se muestra en la figura 3.2 se explica a detalle el proceso para definir un caso de estudio.

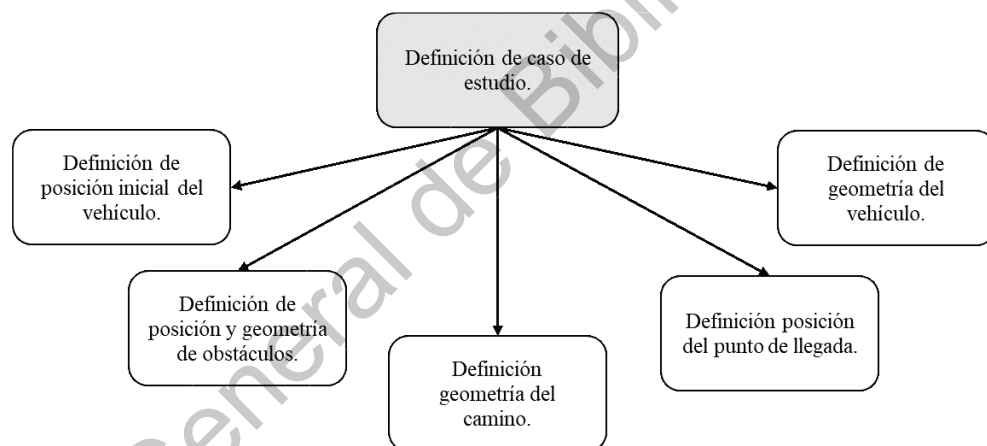


Figura 3.2: Metodología para la asignación de casos de estudio.

En la figura 3.3 se muestra gráficamente el resultado de la creación de un caso de estudio, en donde los cuadros amarillo y naranja representan obstáculos y el recuadro azul representa el objetivo.

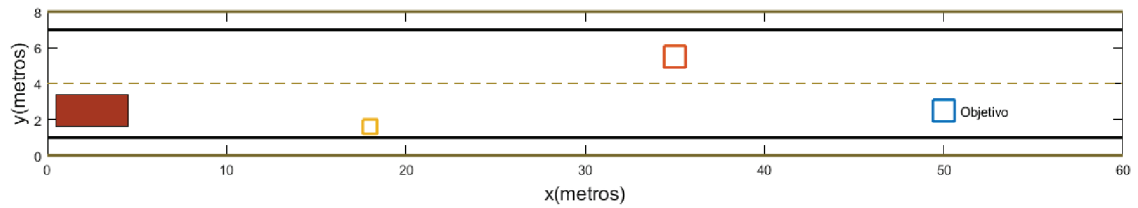


Figura 3.3: Creación de caso de estudio.

Una vez que se tiene un caso de estudio, es posible pasar a la asignación de campos magnéticos virtuales. Se sigue el proceso mostrado en la figura 3.4.

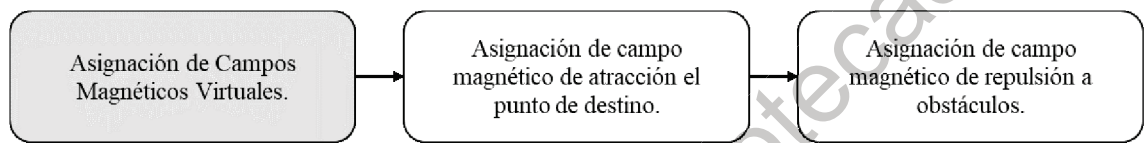


Figura 3.4: Mapa para la asignación de campos magnéticos virtuales.

En la figura 3.5 se muestra gráficamente la asignación de Campos Magnéticos Virtuales. Cabe destacar que no se muestra el alcance del objetivo, ya que este tiene un efecto de atracción magnética virtual sobre todo el entorno muestreado.

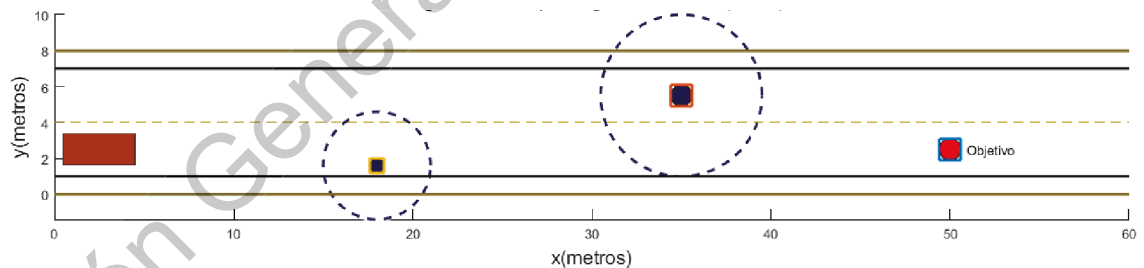


Figura 3.5: Asignación de campos magnéticos virtuales en forma gráfica.

Al momento de asignar los CMV's se crean condiciones iniciales, mismas que corresponden a las variables del objetivo y los obstáculos. Las condiciones iniciales en el caso de los obstáculos se determinan en función del tamaño del obstáculo tal y como se muestra en la tabla 3.2.

Valores iniciales en metros		
Obstáculo 1	Obstáculo 2	Objetivo
$\gamma = 0.4m$	$\gamma = 0.6m$	$\gamma = 0.6m$
$\alpha = 7.5^* \cdot \gamma = 3m$	$\alpha = 7.5^* \cdot \gamma = 4.5m$	$\alpha = 10m^*$
$\beta = 12^* \cdot \gamma = 4.8u$	$\beta = 12^* \cdot \gamma = 7.2u$	$\beta = 4u^*$

Tabla 3.2: Valores iniciales para variables de obstáculos y objetivo.

Recordando que $\gamma = \text{Radio}(m)$, $\alpha = \text{Alcance}(m)$ y $\beta = \text{Intensidad}(u)$. Las constantes marcadas con un asterisco(*) representan constantes que fueron determinadas mediante un proceso de prueba y error. Mientras que los datos marcados en negritas son las variables que modificará el algoritmo genético para encontrar la mejor configuración.

Una vez que se tiene definido un caso de estudio y se ha realizado la correcta asignación de campos magnéticos virtuales, se procesa a la ejecución del algoritmo para obtener una trayectoria libre de colisión. El algoritmo se muestra en la figura 3.6.

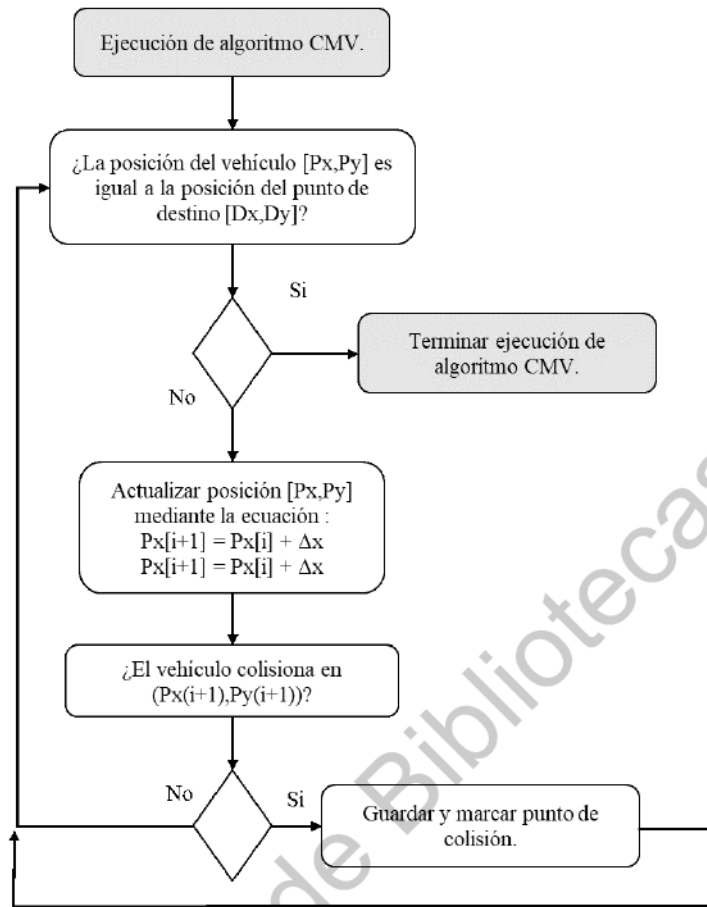


Figura 3.6: Algoritmo para planeación de trayectorias mediante CMV's.

Una vez que el algoritmo termina su ejecución se obtiene una trayectoria marcada por un serie de puntos, en donde la distancia entre cada punto depende de un valor definido previamente, el cuál define un valor máximo obtenido por la ecuación (2.12), para el caso particular del caso de estudio mostrado se definió una distancia máxima(d_{MAX}) de un metro.

La ejecución del algoritmo CMV crea una trayectoria dadas las condiciones iniciales mostradas en la tabla 3.2, y la trayectoria creada con las condiciones iniciales se muestra en la figura 3.7.

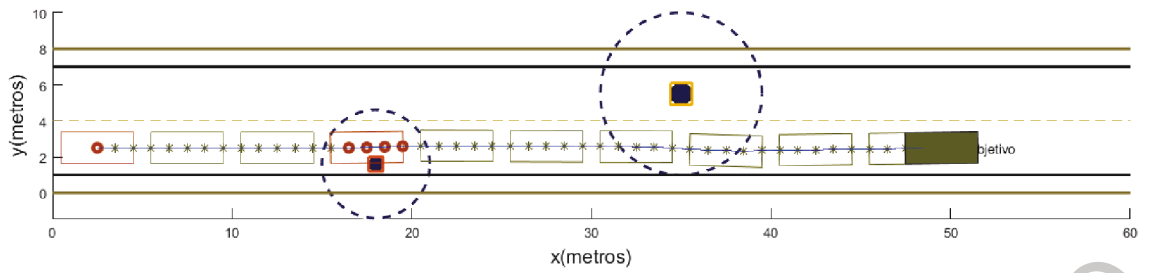


Figura 3.7: Trayectoria resultante con condiciones iniciales.

El resultado mostrado en la figura 3.6, se utiliza como base para crear la población inicial del algoritmo genético, mismo algoritmo que se explica en la figura 3.8.

Dirección General de Bibliotecas UAQ

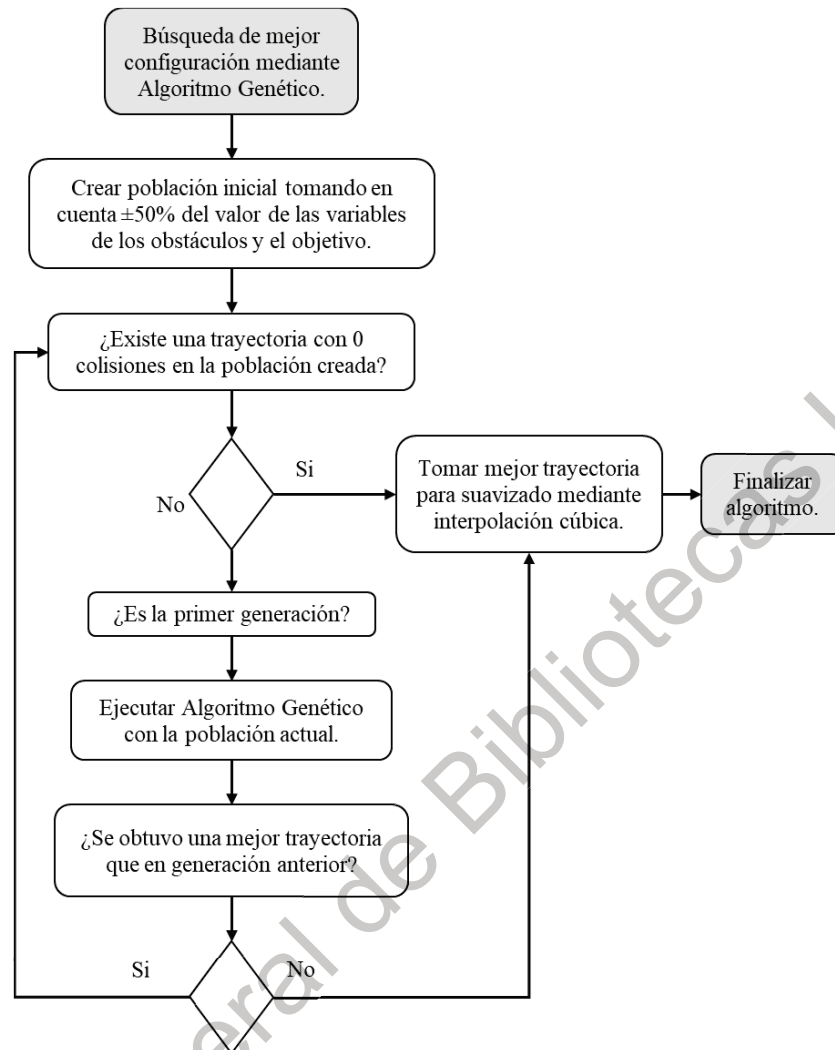


Figura 3.8: Algoritmo genético para búsqueda de mejor configuración.

Tal y como se describe en el algoritmo mostrado en la figura 3.6, se comienza tomando la mejor trayectoria de la población inicial, en este caso, la trayectoria mostrada en la figura 3.9.

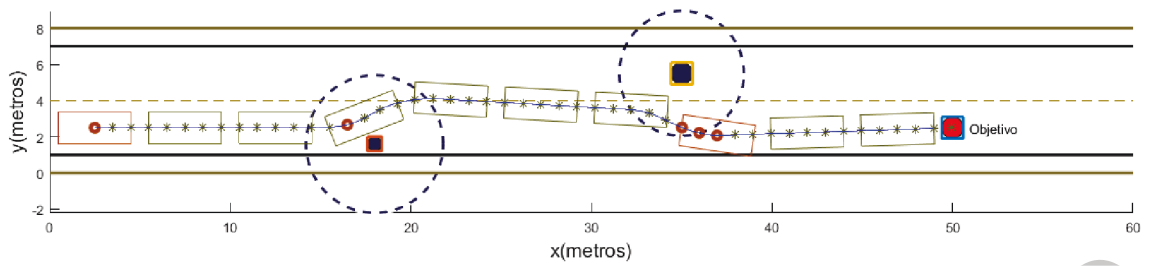


Figura 3.9: Mejor trayectoria tomada de la población inicial.

El algoritmo continuará ejecutándose hasta el punto en el que se detecte que la nueva población no registra una mejor trayectoria que la anterior. Dicho proceso se muestra a continuación, en el cual solo se requiere de una generación para llegar a un caso de 0 colisiones.

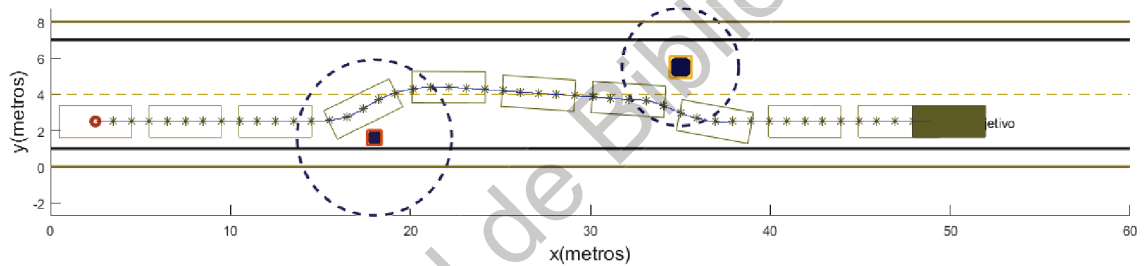


Figura 3.10: Generación N°1.

El valor final de las variables modificadas por el algoritmo genético de muestra en la tabla 3.3.

Valores finales en metros		
Obstáculo 1	Obstáculo 2	Objetivo
$\gamma = 0.4m$	$\gamma = 0.6m$	$\gamma = 0.6m$
$\alpha = 4.1m$	$\alpha = 2.35m$	$\alpha = 10m$
$\beta = 6u$	$\beta = 10.5u$	$\beta = 2.2u$

Tabla 3.3: Valores finales para variables de obstáculos y objetivo.

A pesar de los buenos resultados obtenidos con el enfoque mediante CMV, existen escenarios que limitan las capacidades del algoritmo para crear una trayectoria en casos especiales.

3.1.2. Explicación técnica de Algoritmo Genético

Para explicar la parte técnica del algoritmo genético paso a paso, se usa el ejemplo de la figura 3.11, que solo consta de un obstáculo, considerando los valores iniciales que se muestran en la tabla 3.4.

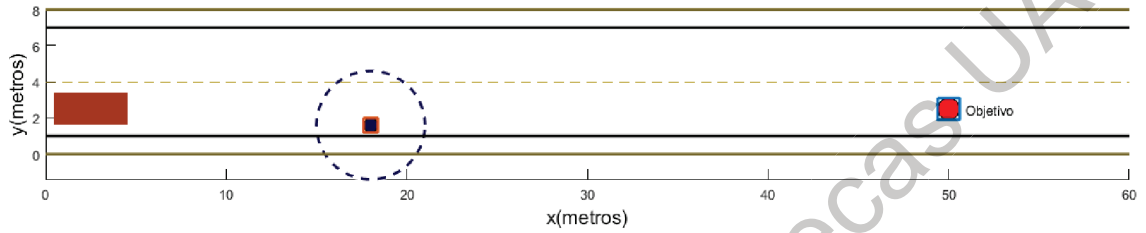


Figura 3.11: Ejemplo para explicación técnica de algoritmo genético.

Valores iniciales en metros	
Obstáculo 1	Objetivo
$\gamma = 0.4m$	$\gamma = 0.6m$
$\alpha = 7.5^* \cdot \gamma = 3m$	$\alpha = 10m$
$\beta = 12^* \cdot \gamma = 4.8u$	$\beta = 2.2u$

Tabla 3.4: Valores iniciales para variables del obstáculo y el objetivo.

Una vez que se define el caso de estudio, el primer paso es crear el Cromosoma que define el caso de estudio. El Cromosoma se crea de la siguiente manera:

$$Cromosoma = [\beta_{O1}, \alpha_{O1}, \dots, \beta_{On}, \alpha_{On}, \beta_{Objetivo}]$$

Para el caso particular que se muestra en la figura 3.11, el Cromosoma que se generó fue el siguiente:

$$Cromosoma = [4.8, 3.0, 4.0]$$

Una vez que se tiene el Cromosoma, se crea una matriz que define la población inicial, misma que se forma usando el $\pm 50\%$ de cada uno de los Genes que conforma el Cromosoma, de tal forma que se obtiene la siguiente población inicial. Esto se muestra en la tabla 3.5.

Valores de población inicial								
β_{O1}	α_{O1}	$\beta_{Objetivo}$	β_{O1}	α_{O1}	$\beta_{Objetivo}$	β_{O1}	α_{O1}	$\beta_{Objetivo}$
2.4	1.5	2	4.1	3.2	3.7	5.8	1.9	5.4
2.5	1.6	2.1	4.2	3.3	3.8	5.9	2.7	5.5
2.6	1.7	2.2	4.3	3.4	3.9	6	4.5	5.6
2.7	1.8	2.3	4.4	3.5	4	6.1	3.4	5.7
2.8	1.9	2.4	4.5	3.6	4.1	6.2	4.2	5.8
2.9	2	2.5	4.6	3.7	4.2	6.3	2.9	5.9
3	2.1	2.6	4.7	3.8	4.3	6.4	1.5	6
3.1	2.2	2.7	4.8	3.9	4.4	6.5	3.4	3.7
3.2	2.3	2.8	4.9	4	4.5	6.6	2.2	2
3.3	2.4	2.9	5	4.1	4.6	6.7	3.1	4.4
3.4	2.5	3	5.1	4.2	4.7	6.8	3.7	5.9
3.5	2.6	3.1	5.2	4.3	4.8	6.9	3.3	2.3
3.6	2.7	3.2	5.3	4.4	4.9	7	3.3	2.1
3.7	2.8	3.3	5.4	4.5	5	7.1	2.8	5.6
3.8	2.9	3.4	5.5	3.7	5.1	7.2	2.2	3
3.9	3	3.5	5.6	4.2	5.2	N/A	N/A	N/A
4	3.1	3.6	5.7	4.4	5.3	N/A	N/A	N/A

Tabla 3.5: Población inicial de caso de estudio.

Una vez que se tiene la población inicial, se obtiene la aptitud de cada cromosoma. En este caso, la aptitud consta de dos valores, los cuales son: el número de colisiones y la inversa de la suma de la distancia mínima a cada obstáculo en cada iteración. La fórmula (3.1) describe la aptitud de cada cromosoma.

$$AC_i = NC + \frac{n}{\sum_{j=1}^n MinDis} \quad (3.1)$$

1. AC_i = Aptitud del cromosoma i .

2. NC = Número de colisiones a lo largo de la trayectoria.
3. n = número de puntos en la trayectoria.
4. $MinDis$ = Distancia mínima a cada obstáculo en cada punto de la trayectoria.

Una vez que se obtiene la población inicial, se obtiene la aptitud de cada cromosoma y se ordena la población en forma ascendente en función de la aptitud. Dicho proceso otorga los resultados mostrados en la tabla 3.6.

Valores con aptitud de población inicial											
Aptitud	β_{O1}	α_{O1}	β_{Obj}	Aptitud	β_{O1}	α_{O1}	β_{Obj}	Aptitud	β_{O1}	α_{O1}	β_{Obj}
0.0733	5.9	3.9	2.2	4.0748	6.2	2.9	4	4.0748	3	3.3	4.5
1.0727	5.5	4.4	2.6	4.0748	2.6	3	3.3	4.0748	3.2	2.3	3.9
1.0727	2.4	4.5	2.1	4.0748	7	2.2	3.7	4.0748	3.3	2.8	4.4
1.0732	6.6	2.9	2.1	4.0748	5.8	2.1	2.4	4.0748	3.4	1.6	3.2
2.0728	2.8	3.5	2	4.0748	5.6	4.3	5	4.0748	3.7	4.4	5.6
3.0749	4.5	3.3	2.3	4.0748	5.2	1.8	2.3	4.0748	3.9	3.4	5.3
4.0748	4.9	3.7	3.4	4.0748	6.5	1.5	2.5	4.0748	5.4	2.4	5.2
4.0748	7.2	3.4	3	4.0748	4.3	1.7	2.8	4.0748	3.5	3.4	5.5
4.0748	4.6	2.7	2.7	4.0748	6.8	1.9	3.7	4.0748	4.1	3.1	5.7
4.0748	6.9	3.6	2.9	4.0748	5	3.8	4.9	4.0748	2.9	4.1	5.9
4.0748	2.7	4.5	3.5	4.0748	4.4	3.7	4.7	4.0748	2.5	4	5.8
4.0748	4.8	3.2	3.6	4.0748	6.1	2.5	4.3	4.0748	6.4	1.5	5.1
4.0748	6	3.1	3.8	4.0748	3.6	3.7	4.6	4.0748	5.7	2	6
4.0748	5.1	2.2	3	4.0748	4.7	2.8	4.4	4.0748	3.1	2.2	5.9
4.0748	4.2	4.2	4.1	4.0748	7.1	2.7	4.8	4.075	6.3	1.9	2
4.0748	3.8	2.6	3.1	4.0748	5.3	4.2	5.6	NA	NA	NA	NA
4.0748	4	4.2	4.2	4.0748	6.7	3.3	5.4	NA	NA	NA	NA

Tabla 3.6: Aptitud de población inicial de caso de estudio de forma ordenada.

En este caso, la población inicial crea un cromosoma en el que su aptitud indica que tiene 0 colisiones, por lo que para este caso no se realizaría el cruce para encontrar una mejor configuración. En un caso en el que la población inicial no otorgue una configuración de 0 colisiones, se realizaría el siguiente proceso.

Suponiendo que tenemos dos obstáculos, el cromosoma tendría la siguiente configuración:

$$Cromosoma = [\beta_{O1}, \alpha_{O1}, \beta_{O2}, \alpha_{O2}, \beta_{Objetivo}]$$

Al momento de hacer el cruce se tomarían de dos en dos cromosomas de la población inicial de forma descendente, los cuales serían los padres a cruzar para generar dos hijos.

$$Padre1 = [\beta_{O1}^1, \alpha_{O1}^1, \beta_{O2}^1, \alpha_{O2}^1, \beta_{Objetivo}^1]$$

$$Padre2 = [\beta_{O1}^2, \alpha_{O1}^2, \beta_{O2}^2, \alpha_{O2}^2, \beta_{Objetivo}^2]$$

Una vez que se tienen los padres, se selecciona un número aleatorio de genes a cruzar, en este caso de 1 a 4 ya que el número de genes es de 5, por lo que seleccionar los 5 genes implicaría hacer una copia del cromosoma.

Suponiendo que se seleccionan 2 genes en posiciones aleatorias, mismas posiciones que son las mismas para el padre 1 y para el padre 2:

$$Padre1 = [\beta_{O1}^1, \alpha_{O1}^1, \beta_{O2}^1, \alpha_{O2}^1, \beta_{Objetivo}^1]$$

$$Padre2 = [\beta_{O1}^2, \alpha_{O1}^2, \beta_{O2}^2, \alpha_{O2}^2, \beta_{Objetivo}^2]$$

de tal forma que el Padre 1 y el Padre 2 generarían los siguiente hijos :

$$Hijo1 = [\beta_{O1}^2, \alpha_{O1}^1, \beta_{O2}^1, \alpha_{O2}^2, \beta_{Objetivo}^1]$$

$$Hijo2 = [\beta_{O1}^1, \alpha_{O1}^2, \beta_{O2}^2, \alpha_{O2}^1, \beta_{Objetivo}^2]$$

Posteriormente, se obtendría la aptitud de los dos hijos, y se eliminarían las dos peores aptitudes tomando en cuenta los padres y los hijos, para así incluir a la nueva generación las mejores dos aptitudes.

Dicho proceso se repite hasta terminar con todos los cromosomas de la población inicial. Si al ordenar la nueva población no mejoró el mejor cromosoma comparado con el mejor cromosoma de la generación anterior, se detiene el algoritmo, de lo contrario continuará ejecutándose.

3.1.3. Limitaciones del algoritmo CMV con enfoque a planeación de trayectorias.

La solución al problema de planeación de trayectorias con restricciones de posicionamiento local y global con el enfoque de CMV presenta limitaciones que no pueden ser corregidas solo con la ayuda del algoritmo CMV.

Una de las principales limitaciones del algoritmo es la geometría con la que se asignan los CMV's a los obstáculos, misma geometría que en cualquier caso tiene que ser circular. Idealmente se podrían manejar todos los obstáculos como cuadrados, lo cual permitiría que un círculo cubra prácticamente toda la superficie del cuadrado, tal y como se realizó en el caso mostrado en la figura 3.10.

Sin embargo, en un caso real con un robot móvil circulando por un entorno desconocido o un vehículo autónomo andando por calles transitadas, en rara ocasión se tendrán entornos en los cuales los obstáculos puedan ser representados solo por

cuadrados.

Otra de las limitaciones es que aumentar el número de los obstáculos incrementa significativamente el tiempo de solución, ya que a mayor número de obstáculos mayor número de generaciones para obtener la mejor trayectoria.

A continuación se pone el ejemplo de soluciones con diferente número de obstáculos para ejemplificar el problema de tiempo de solución en función del número de obstáculos.

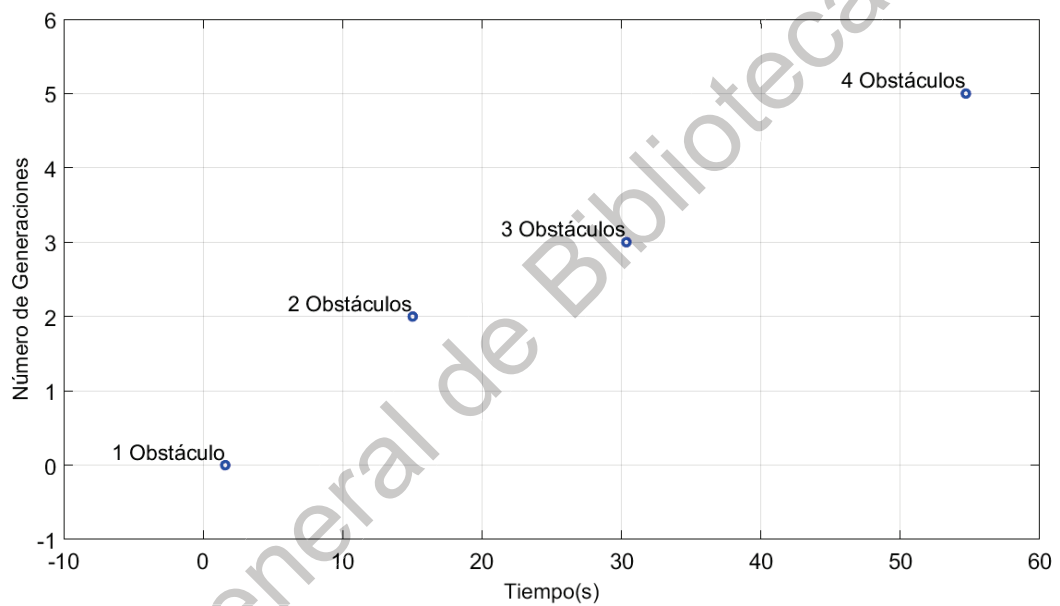


Figura 3.12: Tiempo de solución en función de número de obstáculos.

La figura 3.12 muestra el tiempo de solución y el número de generaciones que toma la solución en función del número de obstáculos, mientras que las figuras 3.13, 3.14, 3.15 y 3.16 muestran las trayectorias resultantes e casos con diferente número de obstáculos.

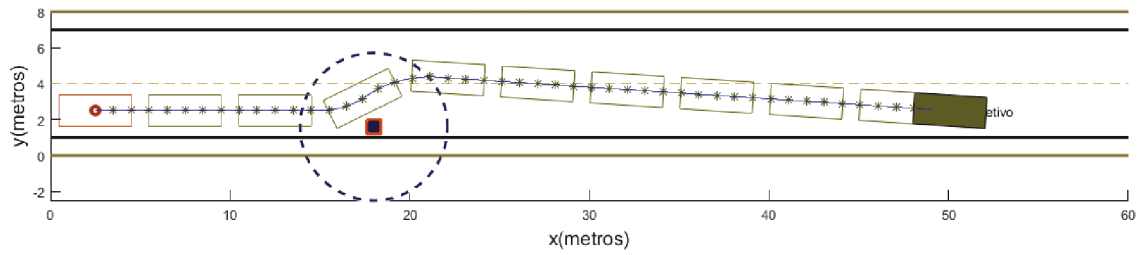


Figura 3.13: Solución con 1 obstáculo

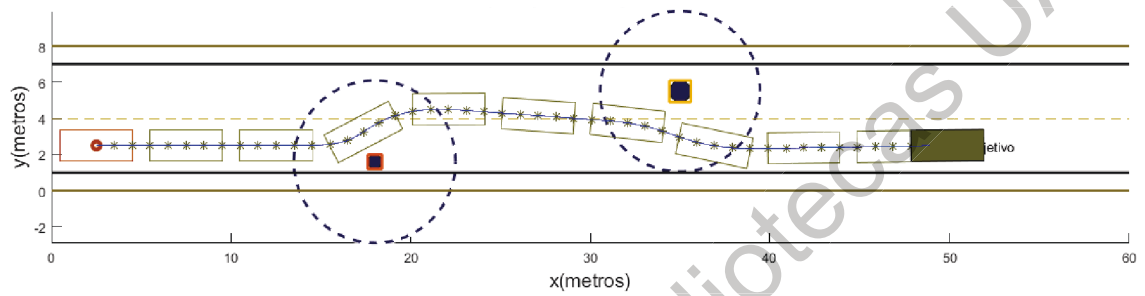


Figura 3.14: Solución con 2 obstáculos

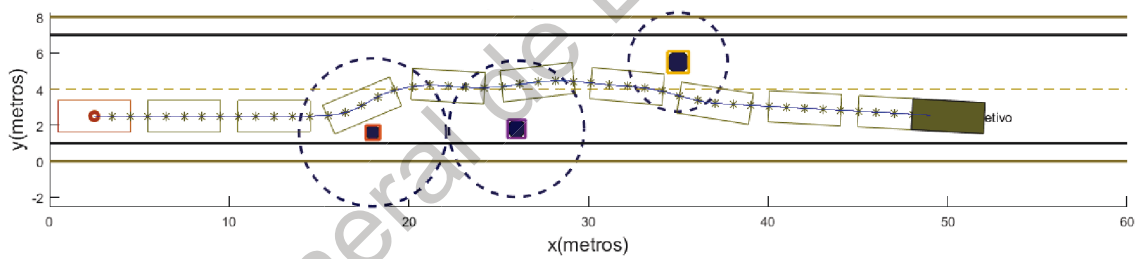


Figura 3.15: Solución con 3 obstáculos

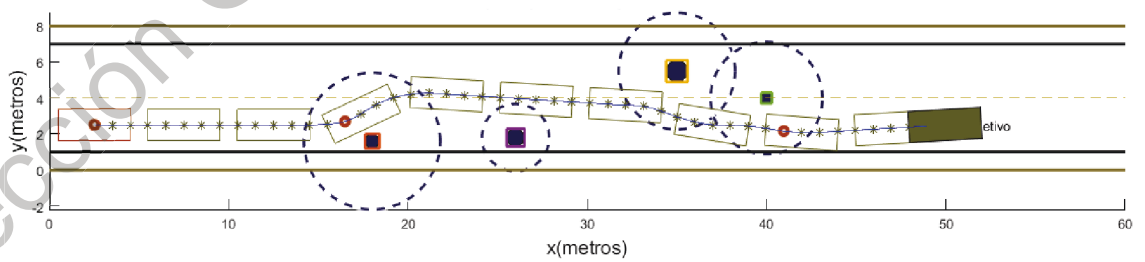


Figura 3.16: Solución con 4 obstáculos

Por último, y como se puede observar en el último caso de la figura 3.16, el hecho de que el algoritmo genético haya encontrado la mejor configuración no

garantiza que se haya encontrado una ruta con 0 colisiones, lo cuál se presenta como otra limitación.

Dichas limitaciones llevaron a buscar un nuevo enfoque, y la solución se encontró en una combinación del algoritmo *Ant System* con el algoritmo CMV. Esto se presenta a continuación.

3.2. Solución mediante algoritmo *Ant System* y CMV's

El enfoque de solución mediante *Ant System* y CMV resolvió dos problemas que el enfoque con CMV no pudo resolver. El primer problema fue el tiempo de solución y el segundo fue el manejo de cualquier geometría.

A este nuevo enfoque lo llamo *Magnetic Ants*(MA) debido que las hormigas son guiadas por una fuerza magnética virtual.

La metodología que se implementó para el enfoque de MA se muestra en la figura 3.17.

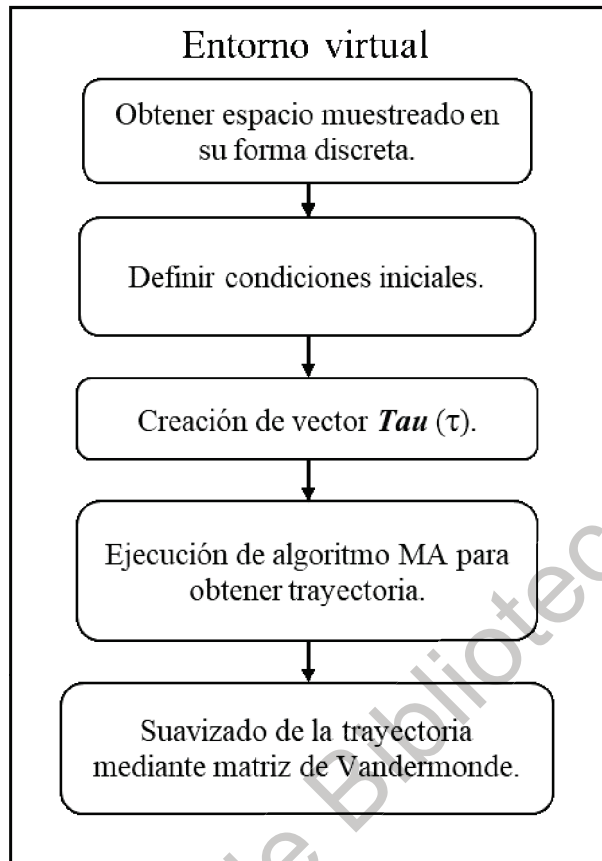


Figura 3.17: Metodología de solución mediante MA.

A continuación se explica a detalle cada parte del proceso de solución mostrado en la figura 3.17.

3.2.1. Obtención de espacio muestreado en su forma discreta

Debido a que en robótica móvil o en vehículos autónomos se maneja la información del entorno muestreado de forma discreta, como se muestra en la figura 3.18, es necesario manejar la información en forma discreta en lugar de manejarla en un espacio continuo como se realizó en el enfoque con CMV.

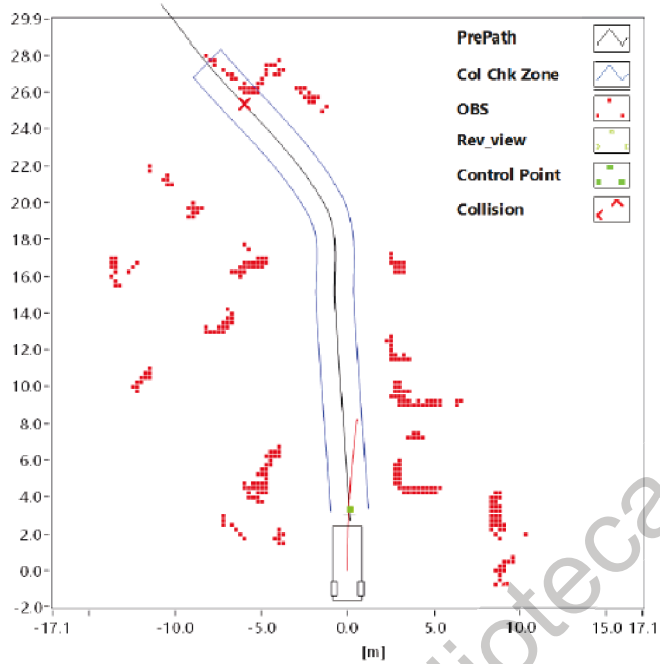


Figura 3.18: Ejemplo de entorno físico muestreado (Lee et al., 2014).

La forma en que se decidió manejar la información fue la siguiente: valor de 0 para detección de un objeto y valor de 1 para camino libre, en donde un objeto puede ser cualquier cosa detectada por la instrumentación de un vehículo autónomo o un robot móvil dependiendo de la aplicación.

La forma en que se maneja la información es mediante una matriz binaria, la cuál contiene la información del entorno muestreado, la figura 3.19 representa la interpretación y el manejo de la información.

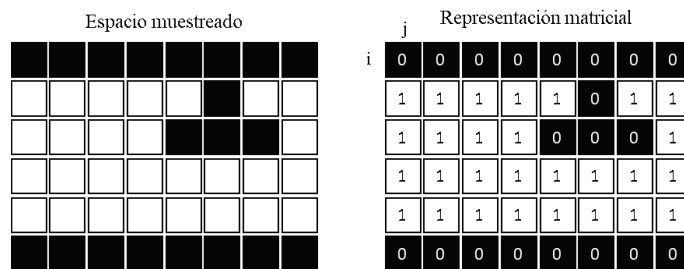


Figura 3.19: Interpretación y manejo de la información binaria.

Para la simulación del algoritmo MA se simula la detección de obstáculos y de los límites del camino, de tal forma que un caso de estudio se puede observar como se muestra en la figura 3.20, en donde el punto en color azul representa la posición inicial del vehículo en su forma virtual y el resto del entorno representa la información del caso de estudio en su forma discreta.

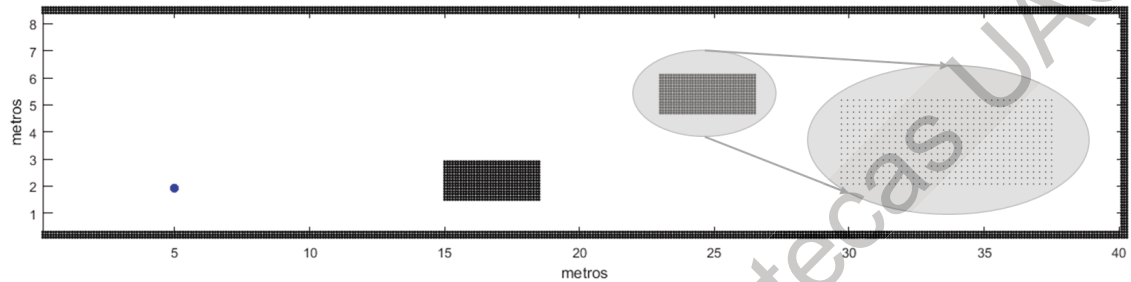


Figura 3.20: Caso de estudio virtual en su forma discreta.

La distancia entre cada punto del espacio discreto dependerá de la precisión de la instrumentación dedicada a la detección de obstáculos. A mayor precisión de la instrumentación, mayor será la cantidad de información a procesar.

Para el caso particular de la simulación de este enfoque, se simuló una precisión de 0.1m, lo cual implica que en la figura 3.20 la distancia entre cada punto representa una distancia de 0.1m.

3.2.2. Definición de condiciones iniciales

Existen tres condiciones iniciales que se deben tomar sin importar cual sea el caso de estudio:

1. Geometría y posición inicial del vehículo.
2. Posición del objetivo.

3. Espacio de seguridad.

Una vez que se obtiene el espacio muestreado como se muestra en la figura 3.20, se procede a insertar las condiciones iniciales en el espacio muestreado.

Para el caso de la geometría del vehículo se consideraron las medidas de una *Hyundai i10*, la cuál tiene dimensiones de 3.6m de largo por 1.6m de ancho. Con las dimensiones del vehículo se procesa a discretizarlo y ubicarlo en el espacio muestreado como se muestra en la figura 3.21.

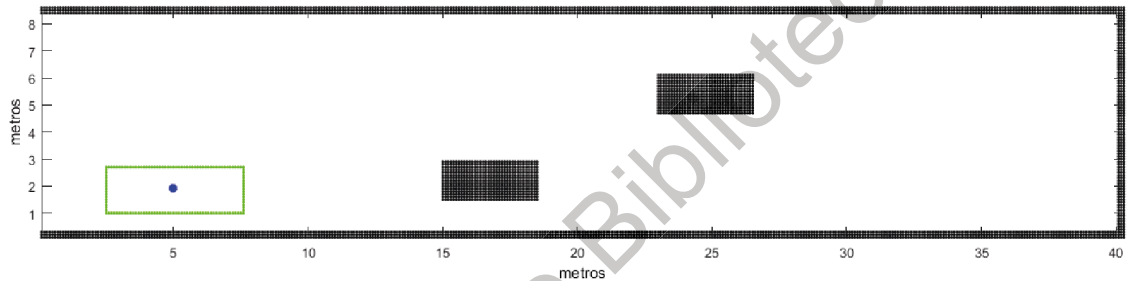


Figura 3.21: Vehículo virtual en espacio discreto.

El objetivo se define como la siguiente coordenada por la que tiene que pasar el vehículo de acuerdo al trayecto que sigue. Idealmente, el algoritmo debería converger en el punto exacto que indica el objetivo. Sin embargo, en las pruebas realizadas con el algoritmo, se encontró que forzar al algoritmo a converger en un punto exacto no es posible de acuerdo al tipo de movimientos permitidos, por lo que se definió una línea continua a partir del objetivo, de forma que el algoritmo puede converger a lo largo de toda la línea como se muestra en la figura 3.22.

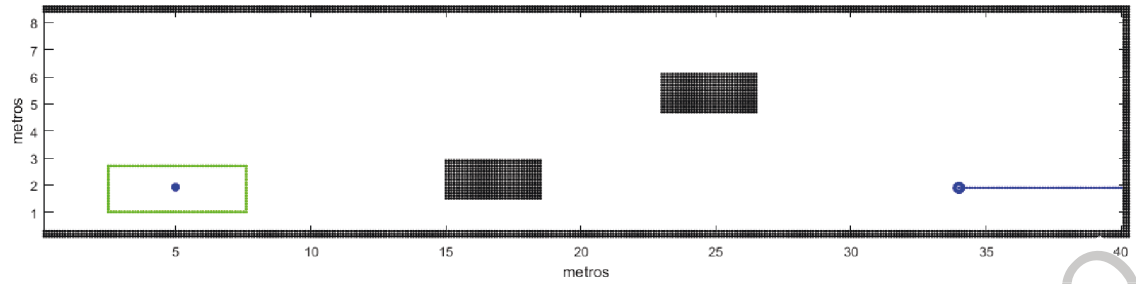


Figura 3.22: Línea de objetivo.

Finalmente, la última condición inicial es la distancia máxima a la cuál se puede acercar el vehículo a los obstáculos. Para realizar dicho proceso se hace un barrido de la matriz binaria de abajo hacia arriba y de izquierda a derecha. En los puntos en los que se detecta un obstáculo se añade la distancia de seguridad correspondiente, para crear una nueva matriz binaria que contenga el espacio de seguridad.

El algoritmo que se encarga de insertar el espacio de seguridad se muestra en el código 3.1.

```

% BinaryVec = Matriz con datos binarios del espacio muestreado
% NVec      = Nueva matriz binaria del espacio muestreado con espacio de seguridad
% MoveSteps = Columnas o renglones ha ser ocupados por espacio de seguridad

% Barrido de izquierda a derecha
for i=1:xLen -1
    for j=1:yLen
        D1 = BinaryVec(j,i);
        D2 = BinaryVec(j,i+1);
        if D1 == 1 && D2 == 0
            NVec(j,(i-MoveSteps+1):i) = 0;
        end
        if D1 == 0 && D2 == 1
            NVec(j,(i+1):(i+MoveSteps)) = 0;
        end
    end
end
end

```

```

% Barrido de abajo hacia arriba
for i=1:xLen
    for j=1:yLen -1
        D1 = BinaryVec(j,i);
        D2 = BinaryVec(j+1,i);
        if D1 == 1 && D2 == 0
            NVec((j-MoveSteps+1):j,i) = 0;
        end
        if D1 == 0 && D2 == 1
            NVec((j+1):(j+MoveSteps),i) = 0;
        end
    end
end
end

BinaryVec = NVec;

```

Código 3.1: Configuraciones iniciales

El resultado de la ejecución del código 3.1 se muestra conceptualmente en la figura 3.23, y al implementarlo en el mismo caso de estudio, da el resultado de la figura 3.24, en el cuál se implementó un espacio de seguridad de 0.4m.

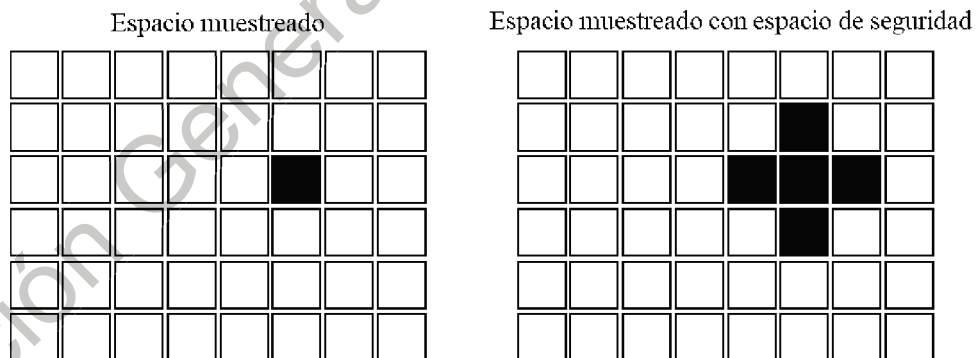


Figura 3.23: Espacio muestreado con espacio de seguridad (Conceptual).

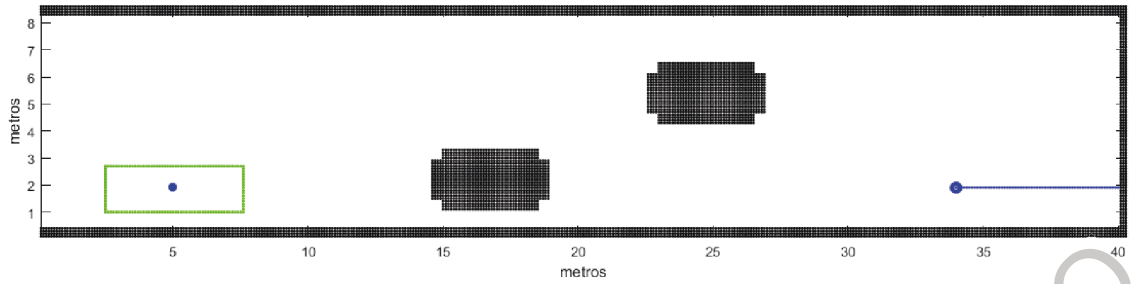


Figura 3.24: Espacio muestreado con espacio de seguridad (Caso de estudio).

Con el espacio de seguridad incluido en la matriz binaria, se concluyen las condiciones iniciales para pasar a la creación del vector $Tau(\tau)$, misma que describe la feromona que hay en cada una de las aristas que determinan los posibles movimientos del vehículo virtual.

3.2.3. Creación de Vector Tau (τ)

El vector Tau es una parte muy importante en el algoritmo MA, ya que su rápida creación, así como el rápido acceso al vector Tau , es lo que determina en gran parte la velocidad de respuesta del algoritmo.

El primer paso es determinar cuales serán los posibles movimientos para las hormigas virtuales o el vehículo virtual, y en base a dichos movimientos crear el vector Tau . Los posibles movimientos se muestran la imagen 3.25.

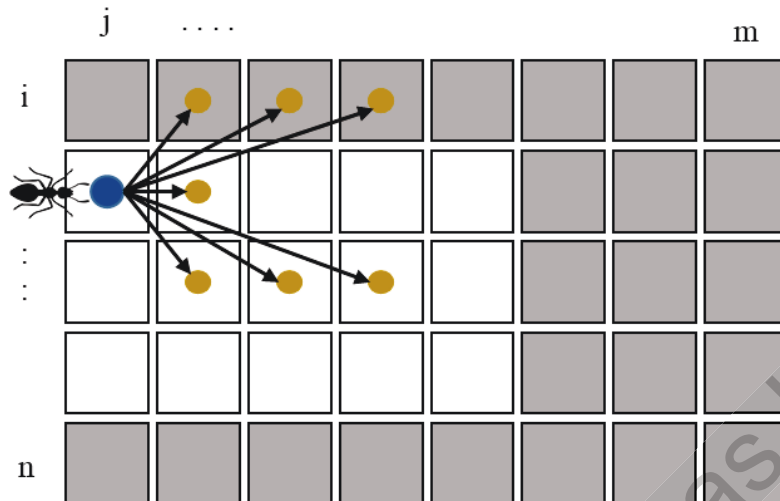


Figura 3.25: Posibles movimientos.

Como se puede observar en la figura 3.25, la hormiga puede tomar 7 diferentes movimientos, los cuales son de: 45° , 18.4° y 14° hacia arriba y hacia abajo, y un movimiento en línea recta. Los movimientos de 45° hacia arriba y hacia abajo ayudan al algoritmo MA a encontrar más rápido una trayectoria para evadir un obstáculo, mientras que los movimientos de 18.4° y 14° ayudan a obtener una trayectoria más suave.

La hormiga virtual (punto azul) mostrada en la figura 3.25 se puede mover por todos los recuadros blancos, mismos que representan el espacio muestreado, en este caso un espacio muestreado sin obstáculos, mientras que los recuadros grises representan renglones y columnas agregados de forma que la hormiga pueda tener los 7 posibles movimientos desde cualquier punto del espacio muestreado.

De acuerdo a la configuración de movimientos mostrada en la figura 3.25, la longitud del vector Tau estará por lo tanto determinada por la fórmula (3.2).

$$TauLength = 7(m - 3)(n - 2) \quad (3.2)$$

Inicialmente, todos los valores del vector Tau son iguales a 1, lo cuál se puede interpretar que se puede pasar por cualquier parte del espacio muestreado.

Mientras que la fórmula para encontrar la posición del vector Tau que corresponde al valor de la feromona para moverse del punto (i_1, j_1) al punto (i_2, j_2) es determinada por la fórmula (3.3).

$$TauPos(i_1, j_1, i_2, j_2) = 7(i_1 - 2)(m - 3) + 7(j_1 - 1) + Val \quad (3.3)$$

$$Val = \begin{cases} (j_2 - j_1) & \text{si } (i_2 - i_1) = -1 \\ 4 & \text{si } (i_2 - i_1) = 0 \\ 4 + (j_2 - j_1) & \text{si } (i_2 - i_1) = 1 \end{cases}$$

Mientras que la función para obtener la posición de Tau en donde se encuentra el valor deseado de la feromona se describe en el código 3.2.

```

function [TauValue, Val] = GetTVal(i_1, j_1, i_2, j_2)
    %T = Vector Tau
    %m = Longitud m de la matriz binaria
    global T xLen

    Val = 7*(i_1 - 2)*(m-3)+7*(j_1 - 1);
    V = i_2 - i_1;
    switch V
        case -1, Val = Val + (j_2 - j_1); % Movimiento hacia arriba
        case 0, Val = Val + 4; % Movimiento recto
        case 1, Val = Val + 4 + (j_2 - j_1); % Movimiento hacia abajo
    end
    TauValue = T(Val);
end

```

Código 3.2: Función para obtener valor de *Tau* en función de coordenadas.

Para ejemplificar el funcionamiento de la fórmula (3.3), se muestra el ejemplo de la figura 3.26.

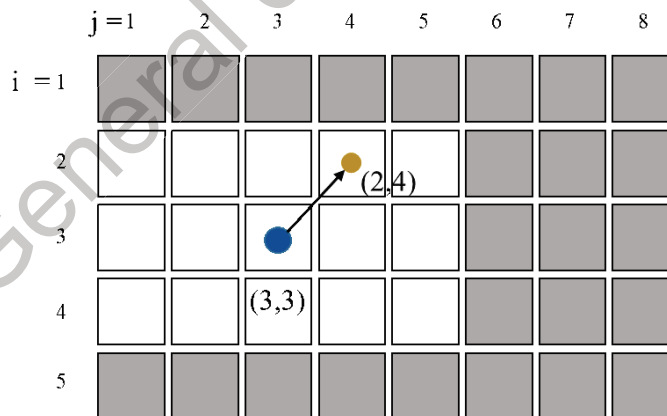


Figura 3.26: Ejemplo de obtención de valor de posición en vector *Tau*.

Al substituir los valores correspondientes del caso mostrado en la figura 3.26 en la fórmula (3.3) se obtiene el siguiente resultado:

$$TauPos(3, 3, 2, 4) = 7(3 - 2)(8 - 3) + 7(3 - 1) + Val$$

$$\text{Al ser } i_2 - i_1 = 2 - 3 = -1 \therefore Val = j_2 - j_1 = 4 - 3 = 1$$

$$TauPos(3, 3, 2, 4) = 7(1)(5) + 7(2) + 1 = 50$$

Por lo tanto, el valor buscado de la feromona se encuentra en T(50).

3.2.4. Ejecución de algoritmo MA

La ejecución del algoritmo MA dará como resultado una trayectoria entre el punto de inicio y el objetivo. La secuencia del algoritmo se muestra en la figura 3.27.

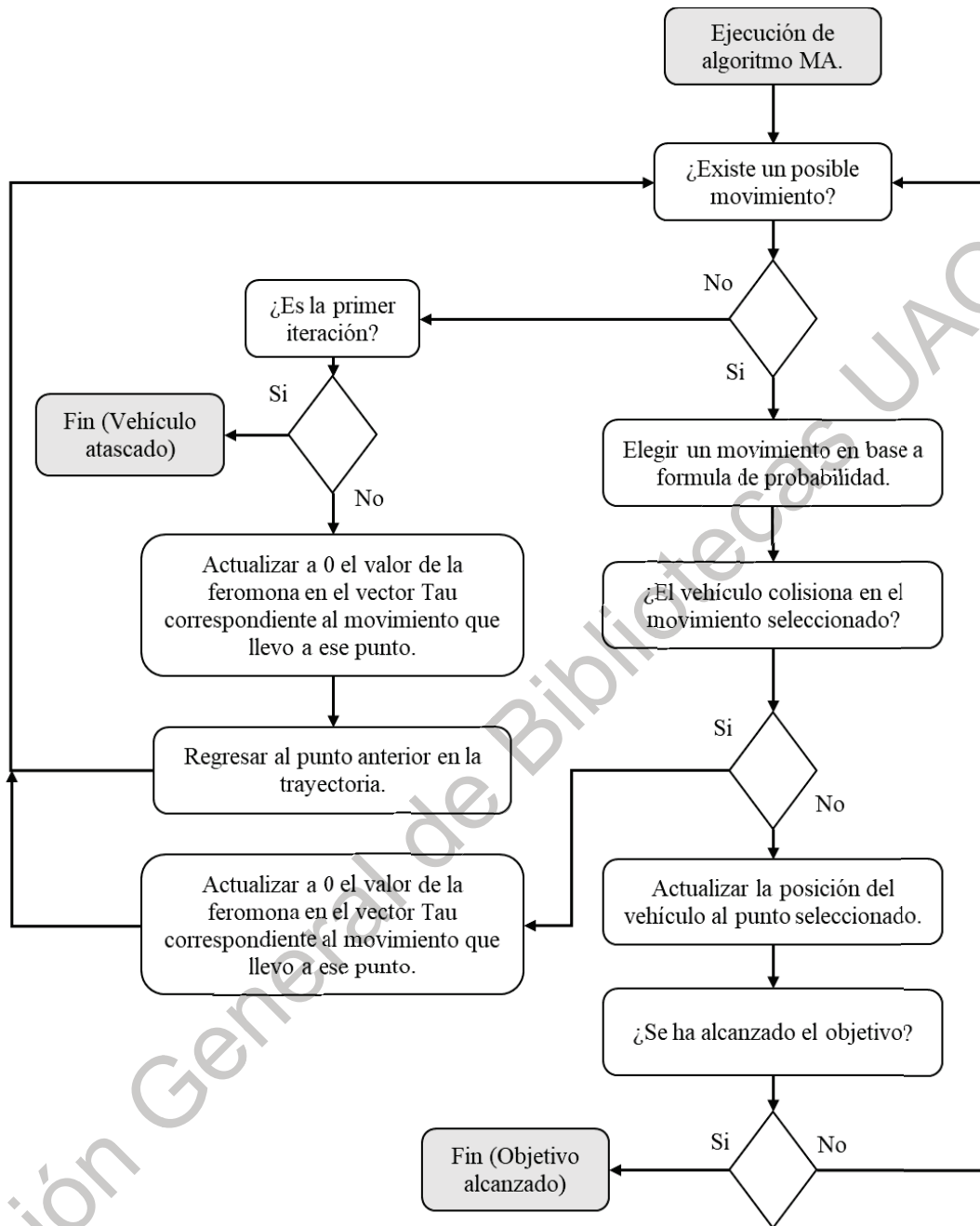


Figura 3.27: Algoritmo MA

Al terminar la ejecución del algoritmo que se muestra en la figura 3.27, se obtiene una serie de puntos que van desde el origen hasta el objetivo, tal y como se muestra en la figura 3.28. Los puntos marcados con rojo representan puntos en los que el vehículo se quedó sin posibles movimientos, y los puntos marcados en color

magenta representan los puntos en los que se detectó un cambio de dirección. Los puntos de color magenta son utilizados para el suavizado de la trayectoria.

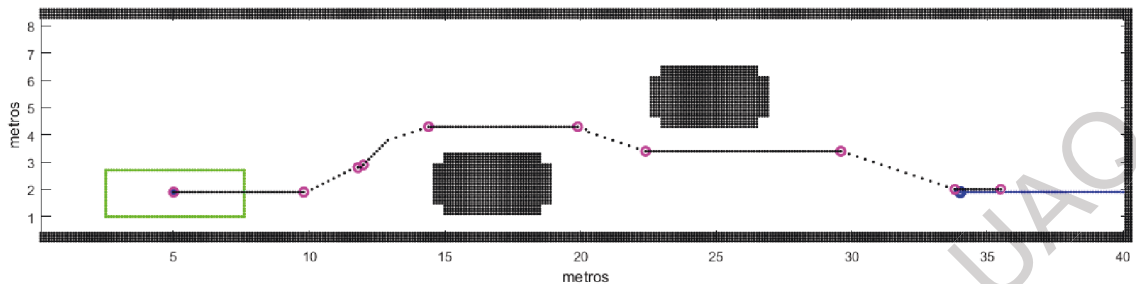


Figura 3.28: Trayectoria generada por algoritmo MA

Hay dos pasos en el algoritmo MA que lo definen. El primero es la fórmula de probabilidad, que describe la probabilidad en todo momento de desviarse por uno de los 7 movimientos que tiene el algoritmo. El segundo paso es la detección de colisión.

La fórmula de probabilidad (3.4) se conforma de dos partes. La primera es la probabilidad en función de la importancia que tiene cada movimiento, mismo que se ejemplifica en la figura 3.29. La segunda es la fuerza de atracción que recibe cada movimiento.

$$Prob(k) = Tval(k) \cdot (Val(k) + AForce(k)) \quad (3.4)$$

en donde :

1. $Prob(k)$ es la probabilidad de moverse a la posición i .
2. $Tval(k)$ es al valor de Tau asignado al movimiento a la posición i .
3. $Val(k)$ es el valor asignado al movimiento k .

4. $AForce(k)$ es la fuerza de atracción que recibe el movimiento en la posición k .

Si una de las aristas que conectan uno de los 7 posibles movimientos ha sido actualizada a 0, debido a que ese movimiento produce una colisión, la sumatoria del valor del movimiento más la fuerza de atracción se anula, y por lo tanto no será posible moverse en esa dirección.

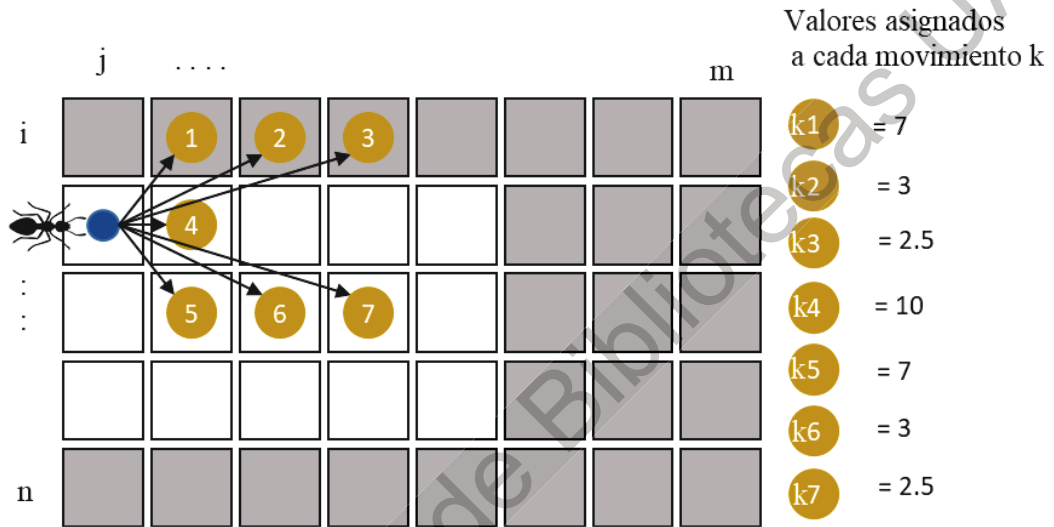


Figura 3.29: Valor de cada movimiento en MA

Los valores asignados a cada movimiento que se muestran en la figura 3.29, están en función de la importancia que tiene cada movimiento. El objetivo siempre es avanzar, por lo que el movimiento recto es al que se le asigna el mayor valor, y le siguen los giros de mayor a menor ángulo. Como se puede observar, los ángulos mas grandes son los más importantes, ya que son lo que ayudan al vehículo virtual a evadir los obstáculos con mayor velocidad.

La fuerza de atracción ($AForce$) que recibe cada movimiento, está en función de la cercanía que tiene con el objetivo y se determina como se muestra en (3.5).

$$AForce(k) = \alpha \cdot (IDis - d(k)) \quad (3.5)$$

en donde :

1. $AForce(k)$ es la fuerza de atracción que recibe el movimiento en la posición k .
2. α es una constante que intensifica la fuerza de atracción y de determino en 33.855.
3. $IDis$ es la distancia que existe entre la posición inicial y el objetivo, dicha distancia se mantiene constante hasta determinar un nuevo objetivo.
4. $d(k)$ es la distancia que existe entre la posición k y el objetivo.

La constante *Alpha* se determino en función de la distancia a la que se deseaba que la partícula magnética empezará a tener efecto sobre los movimientos el vehículo.

El proceso de ajuste de la constante *Alpha* se muestra a continuación:

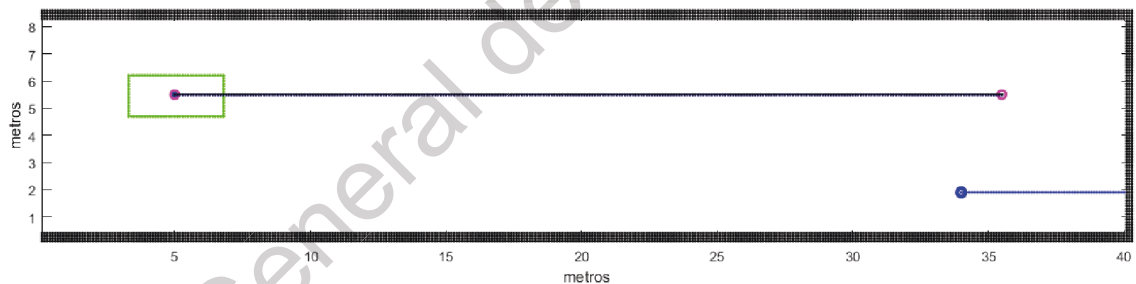


Figura 3.30: Constante Alpha = 30

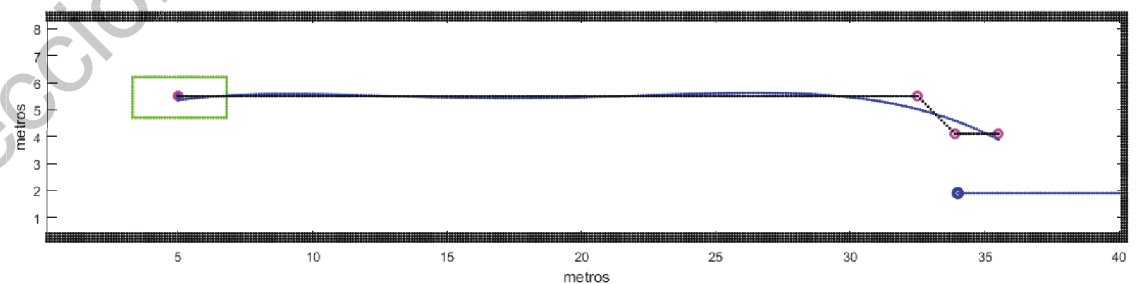


Figura 3.31: Constante Alpha = 32.5

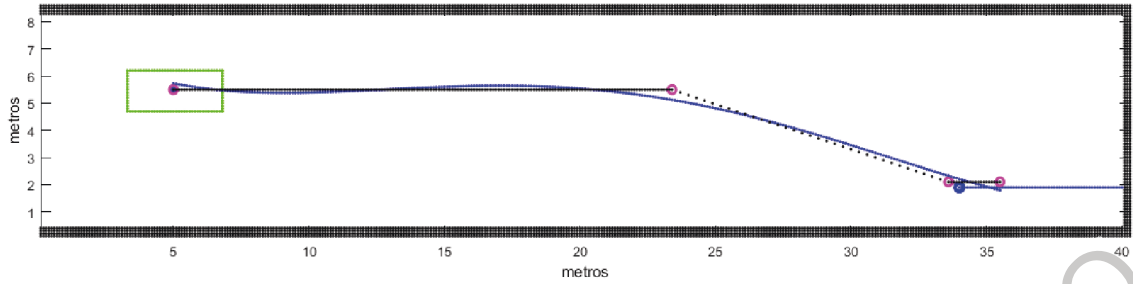


Figura 3.32: Constante Alpha = 33.855

Lo que se puede observar en las figura 3.30, 3.31, 3.32 es un proceso en el cual a mayor es el valor de *Alpha* mayor es la intensidad que tiene la partícula magnética virtual, desde pasar de no ser lo suficientemente fuerte para atraer al vehículo virtual como se muestra en la figura 3.30 hasta tener la suficiente fuerza como para atraer al vehículo virtual a 11 metro de distancia.

La detección de colisión se realiza de forma discreta, de forma que el vehículo discreto se gira de acuerdo al movimiento *k* elegido, y se obtienen los puntos en los que se ubica el vehículo. En el momento en el que se detecta que uno de lo puntos del vehículo discreto coincide con un obstáculo o uno de los limites del camino, se marca como un punto de colisión.

En la figura 3.33 se muestra un caso en el que el movimiento elegido por la fórmula de probabilidad posiciona al vehículo en un punto de colisión.

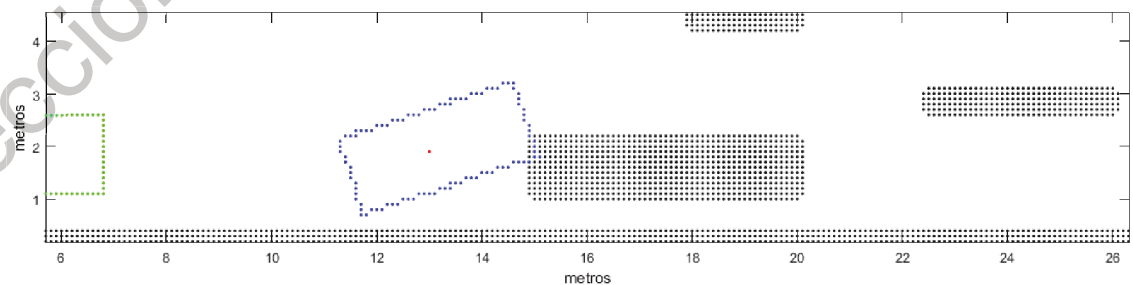


Figura 3.33: Caso de colisión por elección de movimiento.

Para optimizar la velocidad de ejecución se guardó la posición relativa de cada punto del vehículo discretizado respecto a su centro geométrico en cada uno de los posibles giros. El proceso de detección de colisión se muestra en el código 3.3.

```

function RotateCar(Dir)
    % CarIndex = Matriz con valores relativos del carro

    NCar = CarMeters;
    Len = length(NCar(1,:));

    if Dir == 4 % Movimiento recto
        NCar(1,:) = CarIndex(1,:) + 1;
        NCar(2,:) = CarIndex(2,:);
    else % Movimiento giratorio
        NCar(1,:) = CarIndex(1,:) - CarIndexMatrix(2*Dir-1,:);
        NCar(2,:) = CarIndex(2,:) - CarIndexMatrix(2*Dir,:);
    end

    i=1;
    % Verificar choque con el entorno
    while (flag == 0 && i<=Len)
        if NCar(2,i) <= yLen && NCar(2,i) > 0 && NCar(1,i) <= xLen && NCar(1,i) > 0
            if BinaryVec(NCar(2,i),NCar(1,i)) == 0
                flag = 1;
            end
            i=i+1;
        else
            flag = 1;
        end
    end
end

```

Código 3.3: Configuraciones iniciales

3.2.5. Suavizado de la trayectoria

El suavizado de la trayectoria se realizó mediante el despeje de la matriz Vandermonde (3.6), método que realiza un ajuste polinomial tomando como datos de entrada los puntos en $[x,y]$ y el grado del polinomio deseado, en donde se cuenta con los datos $[x, y]$, y el objetivo es despejar el vector P que contiene los datos del polinomio. En este caso el grado del polinomio se determino en base al número de cambios de dirección mismos cambios de dirección que se pueden apreciar en color magenta en la imagen 3.34.

$$\begin{pmatrix} x_1^n & x_1^{n-1} & \cdots & 1 \\ x_2^n & x_2^{n-1} & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ x_m^n & x_m^{n-1} & \cdots & 1 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_{n+1} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} \quad (3.6)$$

La solución a la matriz de Vandermonde se puede escribir también de la siguiente manera:

$$XP = Y \rightarrow P = X^{-1}Y$$

Finalmente se realiza el suavizado de la trayectoria como se muestra en la figura 3.34, en donde la línea color azul representa la trayectoria suavizada.

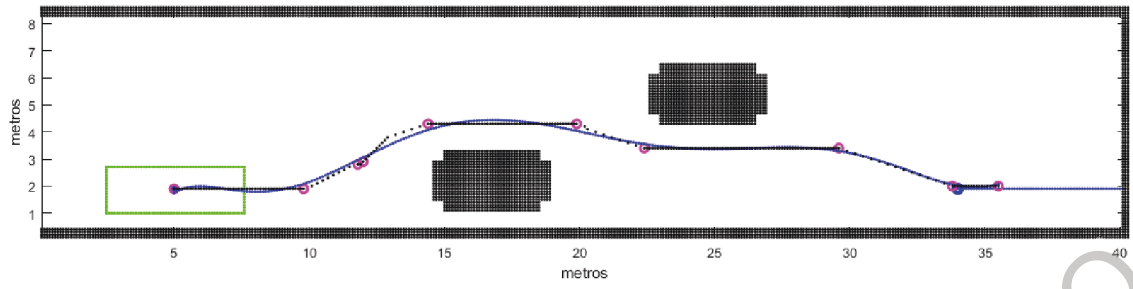


Figura 3.34: Trayectoria suavizada mediante solución a matriz Vandermonde.

Dirección General de Bibliotecas UAQ

Resultados y conclusiones

4.1. Tiempos de solución y casos de estudio evaluados

Se probó el algoritmo MA con tres dimensiones de tres vehículos diferentes en 5 casos de estudio diferentes, cada caso con una complejidad mayor, siendo el caso número 1 el caso con la menor complejidad y el caso número 5 con la mayor complejidad.

Las dimensiones de los vehículos que se simularon fueron los siguientes:

1. (*HatchBack*) *Hyundai i10*..... Largo = 3.6m, Ancho = 1.6m.
2. (*SUV*¹) *Mazda CX-5*..... Largo = 4.5m, Ancho = 1.9m.
3. (*Pick-Up*) *Mitsubishi L200*..... Largo = 5.2m, Ancho = 1.8m.

En la gráfica 4.1 se muestran los resultados de los tiempos de solución con el algoritmo MA. En la cuál se puede observar tiempos de solución significativamente menores a los tiempos de solución obtenidos con el enfoque CVM mismos resultados que se muestran en la gráfica 3.12.

El peor tiempo obtenido fue de poco menos de 0.4s lo cuál es un tiempo aceptable ya que nos permite en teoría obtener al menos dos trayectorias por segundo.

¹Sport utility vehicle

Mientras que los mejores tiempos se encuentran por debajo de 0.2s, mismos tiempos que se obtuvieron en los casos de simulación que presentan los escenarios mas comunes para un vehículo convencional que transita en una zona urbana.

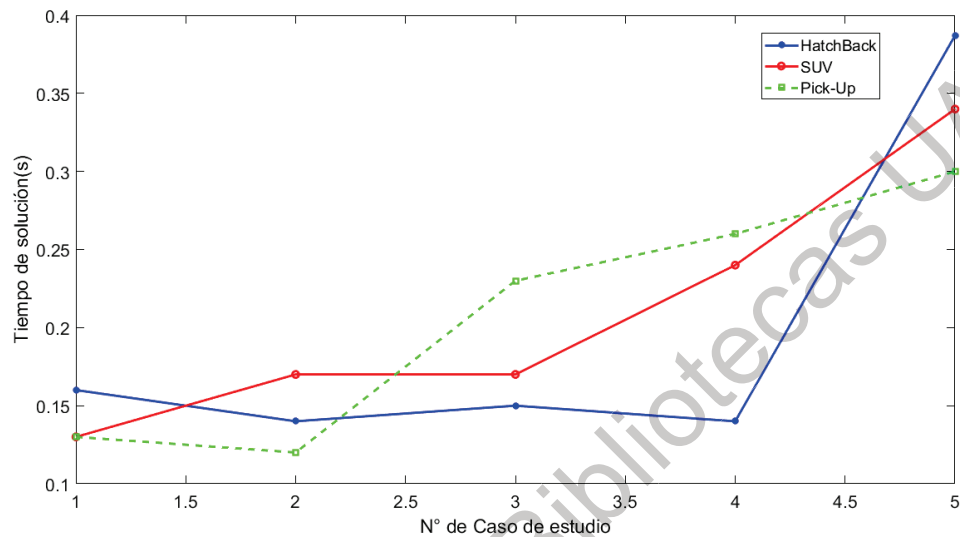


Figura 4.1: Tiempos de solución en diferentes casos de estudio.

Los resultados que se muestran en la figura 4.1 son el resultado de la simulación con un procesador *Intel Core i7 de 2.6 GHz*.

4.2. Resultados con *Hyundai i10 (HatchBack)*

Se muestran los resultados de la simulación para el vehículo *Hyundai i10 (HatchBack)*), considerando 5 diferentes casos de simulación.

La línea punteada en color negro representa la trayectoria discretizada otorgada por el algoritmo MA, mientras que la línea color azul representa la trayectoria suavizada.

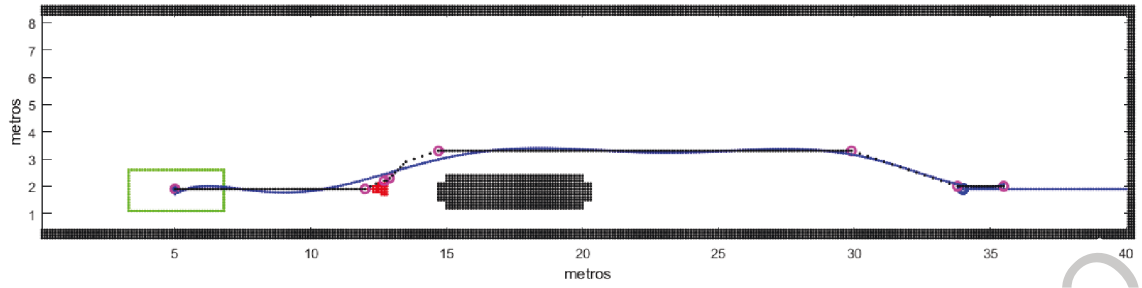


Figura 4.2: Caso de estudio 1 con *(HatchBack) Hyundai i10*.

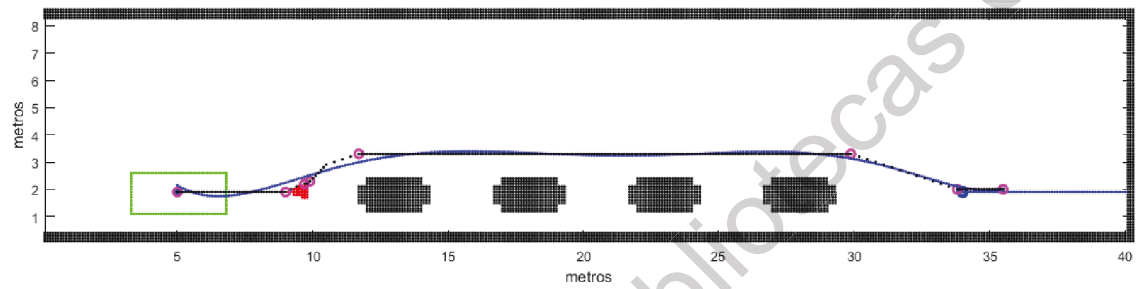


Figura 4.3: Caso de estudio 2 con *(HatchBack) Hyundai i10*.

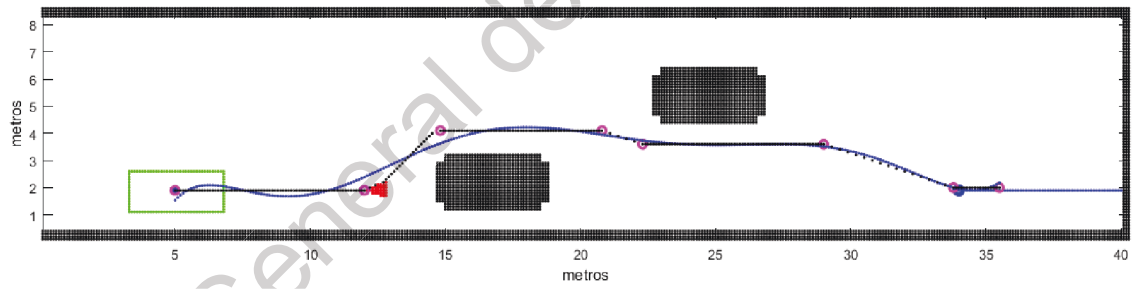


Figura 4.4: Caso de estudio 3 con *(HatchBack) Hyundai i10*.

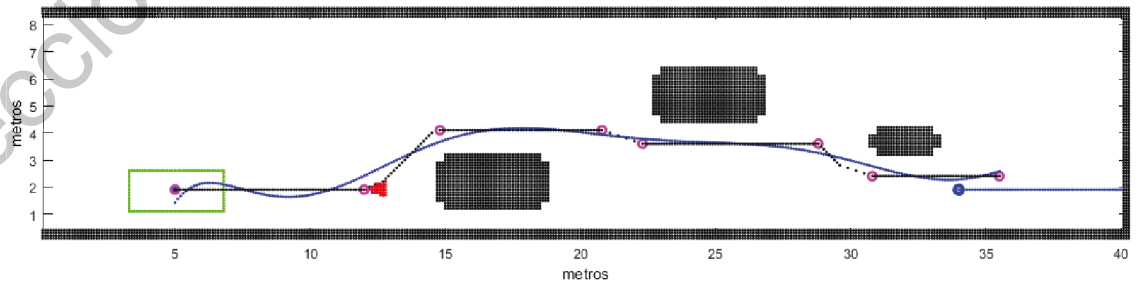


Figura 4.5: Caso de estudio 4 con *(HatchBack) Hyundai i10*.

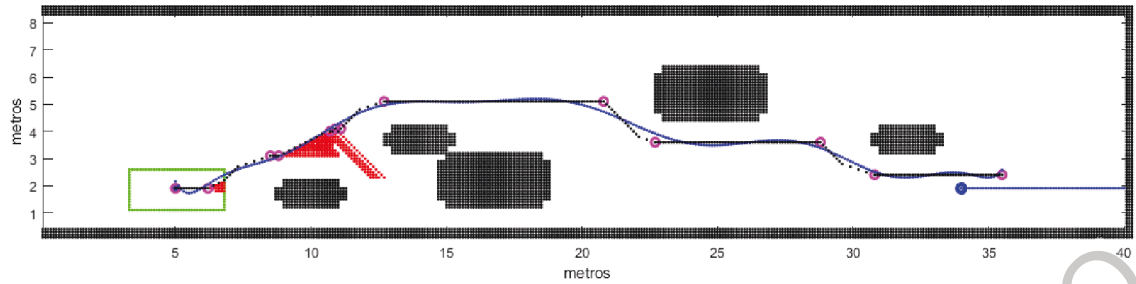


Figura 4.6: Caso de estudio 5 con *(HatchBack) Hyundai i10*.

De los resultados obtenidos por el vehículo *(HatchBack) Hyundai i10* en los 5 diferentes casos de estudio se puede concluir que un vehículo pequeño puede encontrar fácilmente una solución a entornos complejos por su versatilidad al moverse por espacios reducidos

Sin embargo, como se puede observar en la figura 4.6, el hecho de que el vehículo sea tan pequeño produce que explore soluciones que lo llevan a situaciones en la que se encuentra con un camino sin retorno lo cual aumenta el tiempo de solución.

4.3. Resultados con *(SUV) Mazda CX-5*

A continuación se muestran los resultados obtenidos por el primer vehículo (*(SUV) Mazda CX-5*) en los 5 diferentes casos de simulación.

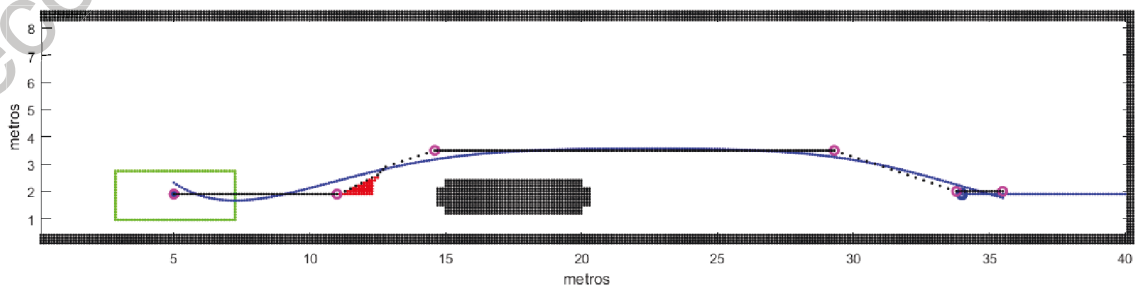


Figura 4.7: Caso de estudio 1 con *(SUV) Mazda CX-5*.

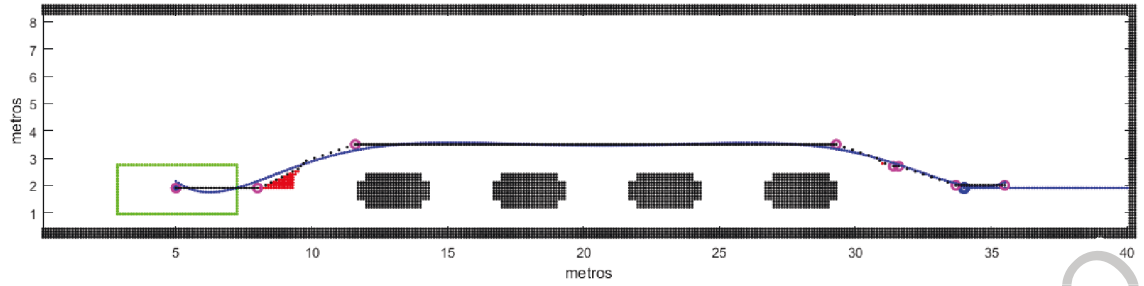


Figura 4.8: Caso de estudio 2 con (*SUV*) Mazda CX-5.

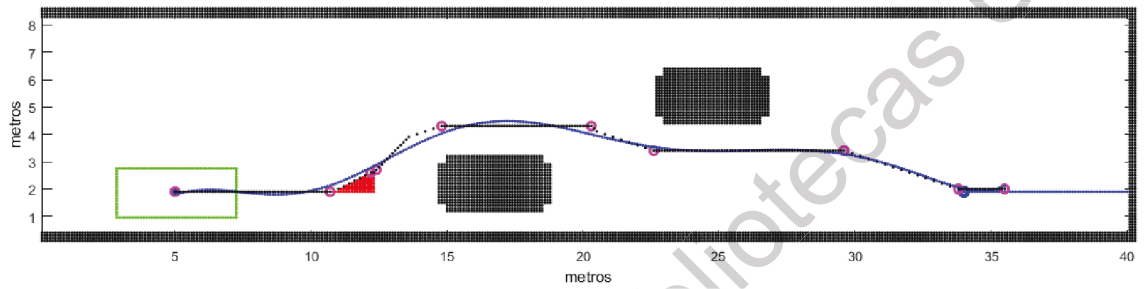


Figura 4.9: Caso de estudio 3 con (*SUV*) Mazda CX-5.

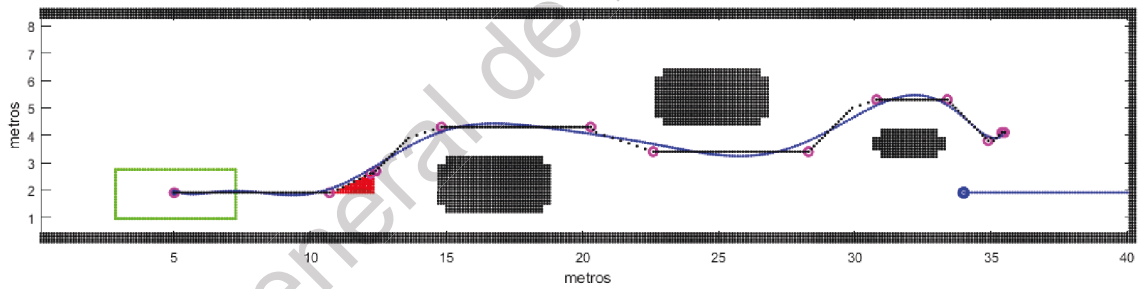


Figura 4.10: Caso de estudio 4 con (*SUV*) Mazda CX-5.

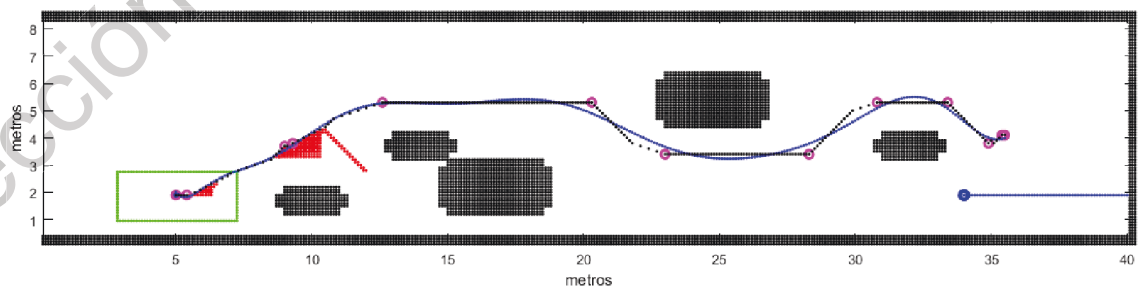


Figura 4.11: Caso de estudio 5 con (*SUV*) Mazda CX-5.

La simulación de un vehículo con las dimensiones de la (*SUV*) Mazda CX-5

produce que el espacio de soluciones explorado sea menor en comparación con el (*HatchBack*) *Hyundai i10* debido a que sus dimensiones son mayores.

Un problema que se genera al tener un vehículo más grande es que es necesario buscar caminos con mayor libertad de movimiento, lo cuál como se muestra en las figuras 4.10 y 4.11, lleva a caminos que no garantizan su llegada al objetivo, sin embargo este resultado no se considera como un fracaso ya que el vehículo termina su trayectoria con la orientación correcta.

4.4. Resultados con (*Pick-Up*) *Mitsubishi L200*

A continuación se muestran los resultados obtenidos por el primer vehículo (*Pick-Up*) *Mitsubishi L200*) en los 5 diferentes casos de simulación.

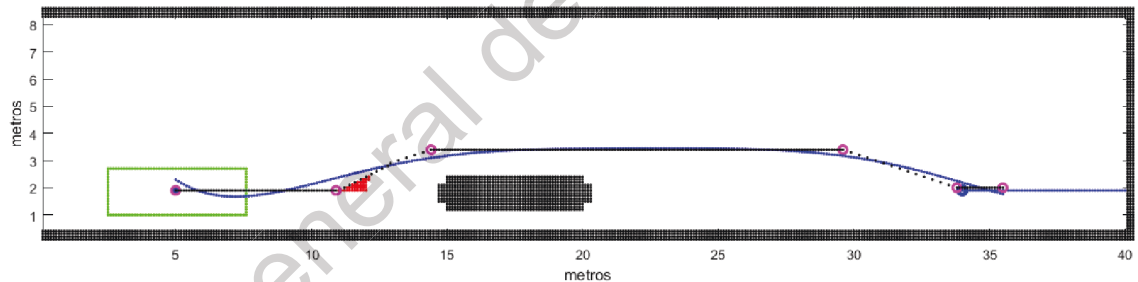


Figura 4.12: Caso de estudio 1 con (*Pick-Up*) *Mitsubishi L200*.

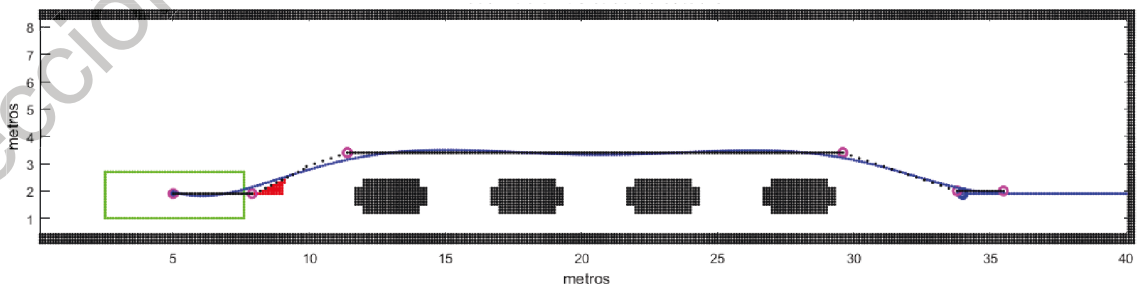


Figura 4.13: Caso de estudio 2 con (*Pick-Up*) *Mitsubishi L200*.

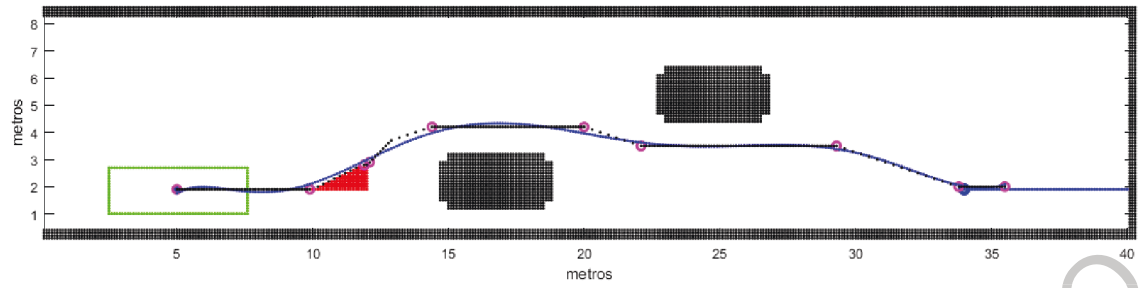


Figura 4.14: Caso de estudio 3 con (*Pick-Up*) Mitsubishi L200.

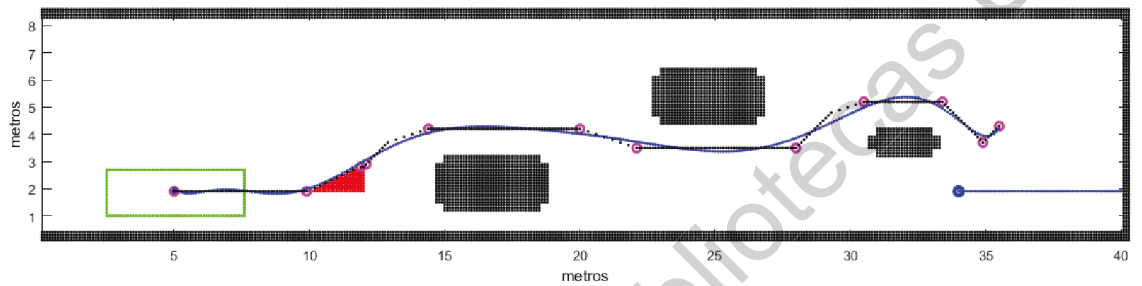


Figura 4.15: Caso de estudio 4 con (*Pick-Up*) Mitsubishi L200.

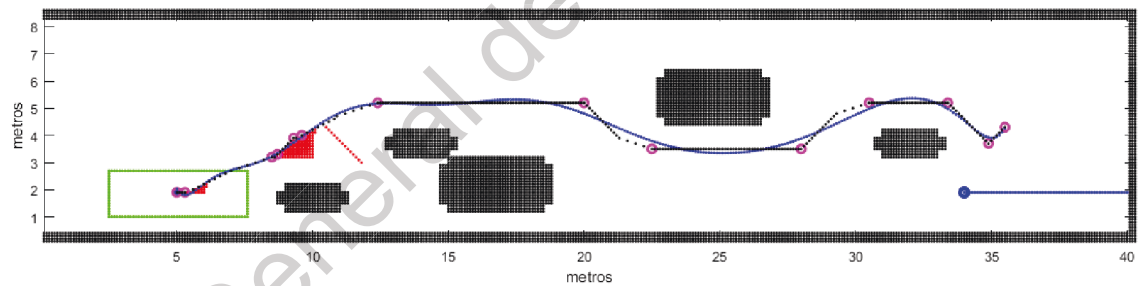


Figura 4.16: Caso de estudio 5 con (*Pick-Up*) Mitsubishi L200.

A pesar de que un vehículo *Pick-Up* como es el caso de la (*Pick-Up*) Mitsubishi L200 es mucho más larga que una *SUV*, los resultados fueron bastante similares, lo cual nos lleva a deducir que el ancho de un vehículo limita más sus movimientos el largo del mismo.

4.5. Conclusiones

En el marco del desarrollo de algoritmos para vehículos autónomos, la evasión de obstáculos representa un reto bastante importante. Una de las características más importantes es tomar en cuenta las restricciones locales y globales. Actualmente el proceso de solución de restricciones locales y globales se realiza por separado.

Debido a esto, se han desarrollado una gran variedad de trabajos desarrollando algoritmos que ayuden a realizar la función de evasión de obstáculos. Entre estos algoritmos, podemos encontrar algoritmos clásicos como CMV y Dijkstra. También podemos encontrar algoritmos metaheurísticos como Ant System y genéticos, entre otros.

Sin embargo, estos algoritmos tienen ciertas limitaciones en su aplicación. Los algoritmos clásicos evalúan todo el universo de soluciones, lo que los hace muy lentos. Los metaheurísticos no exploran todo el universo de soluciones, pero hasta ahora no se había realizado alguna adaptación que considere restricciones locales y globales.

Para este trabajo se usó el algoritmo CMV como punto de partida en la búsqueda de un algoritmo que facilite la evasión de obstáculos. En las pruebas realizadas con CMV, se encontraron dos limitaciones importantes. Una es la geometría que se asigna a los obstáculos, misma que en todos los casos debe ser circular. Otra limitación es que el número de configuraciones posibles es muy grande, por lo que encontrar la configuración ideal toma mucho tiempo.

Debido a las limitaciones del CMV, se desarrolló un nuevo enfoque que combina conceptos de Ant System y CMV, al cual se dio el nombre de *Magnetic Ants* (MA). Al realizar esta combinación se lograron las siguientes ventajas: Se puede

manejar cualquier geometría y trabaja en un espacio discreto. Como limitación se encontró que en ocasiones la aproximación polinomial produce curvas no deseadas en los extremos de la trayectoria.

Al comparar los resultados obtenidos con CMV y MA, se encontró que el tiempo de solución es mucho menor en MA que en CMV. Además, en el algoritmo CMV, cuando el vehículo se encuentra en una ruta de la que no tiene salida, el algoritmo no tiene forma de resolver el problema y el vehículo deja de moverse, por lo que la trayectoria se queda incompleta. Por el contrario, en el caso del algoritmo MA, cuando se presenta la situación de no encontrar una salida, el algoritmo hace que el vehículo regrese sobre su trayectoria hasta encontrar un nuevo camino. Otra diferencia importante es que el algoritmo CMV solamente puede usar obstáculos con geometría circular, mientras que el algoritmo MA puede usar cualquier tipo de geometría.

Referencias

- Chamzas, C., Shrivastava, A., y Kavraki, L. E. (2019). Using local experiences for global motion planning. *arXiv preprint arXiv:1903.08693*.
- Dorigo, M., Maniezzo, V., y Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, man, and cybernetics, Part B: Cybernetics*, 26(1), 29–41.
- Dorigo, M., y Socha, K. (2006). An introduction to ant colony optimization. *Universit de Libre de Bruxelles, CP*, 194(6).
- Garcia, M. P., Montiel, O., Castillo, O., Sepúlveda, R., y Melin, P. (2009). Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. *Applied Soft Computing*, 9(3), 1102–1110.
- Guan-Zheng, T., Huan, H., y Sloman, A. (2007). Ant colony system algorithm for real-time globally optimal path planning of mobile robots. *Acta automatica sinica*, 33(3), 279–285.
- Hofner, C., y Schmidt, G. (1995). Path planning and guidance techniques for an autonomous mobile cleaning robot. *Robotics and autonomous systems*, 14(2-3), 199–212.
- Huang, Y., Ding, H., Zhang, Y., Wang, H., Cao, D., Xu, N., y Hu, C. (2019). A motion planning and tracking framework for autonomous vehicles based on artificial potential field-elaborated resistance network (apfe-rn) approach. *IEEE Transactions on Industrial Electronics*.
- Lee, U., Yoon, S., Shim, H., Vasseur, P., y Demonceaux, C. (2014). Local path planning in a complex environment for self-driving car. En *Cyber technology in*

automation, control, and intelligent systems (cyber), 2014 ieee 4th annual international conference on (pp. 445–450).

Marin-Plaza, P., Hussein, A., Martin, D., y Escalera, A. d. l. (2018). Global and local path planning study in a ros-based research platform for autonomous vehicles. *Journal of Advanced Transportation, 2018*.

Mei, H., Tian, Y., y Zu, L. (2006). A hybrid ant colony optimization algorithm for path planning of robot in dynamic environment. *International Journal of Information Technology, 12(3)*, 78–88.

Paromtchik, I. E., y Laugier, C. (1996). Motion generation and control for parking an autonomous vehicle. En *Proceedings of ieee international conference on robotics and automation* (Vol. 4, pp. 3117–3122).

La Mecatrónica en México



DICTÁMEN DE EVALUACIÓN

4 de abril de 2019

OFICIO LMEM/20190404-1

Por este medio, el comité editorial de la revista LMEM se les informa a los autores que el trabajo titulado

Planeación de trayectorias mediante campos magnéticos virtuales y sus limitaciones

Con número de registro **LMEM20181205-1**, cuyos autores son

Rodríguez Díaz Emmanuel, Juan Manuel Ramos Arreguín, Marco Antonio Aceves Fernández

Ha tenido como dictamen **APROBADO**. Por lo que el artículo será publicado en la edición de Mayo del presente año.

ATENTAMENTE,
Comité Evaluador de los Artículos de la revista
“La Mecatrónica en México”



ASOCIACIÓN MEXICANA
DE MECATRÓNICA A.C.



Planeación de trayectorias mediante campos magnéticos virtuales y sus limitaciones.

Rodríguez Díaz Emmanuel, Juan-Manuel Ramos-Arreguín✉, Marco-Antonio Aceves-Fernández

Universidad Autónoma de Querétaro, Facultad de Ingeniería
emmanuel.rodriguez.diaz94@gmail.com, jsistdig@yahoo.com.mx✉, marco.aceves@gmail.com

Resumen

El tema que se trata a continuación muestra una técnica para evasión de obstáculos basada en la simulación de campos magnéticos virtuales (CMV) con un enfoque hacia vehículos autónomos. El trabajo demuestra que al momento de generar una trayectoria mediante CMV se pueden incluir tanto restricciones locales como globales en el mismo algoritmo para garantizar soluciones que aseguren una ruta libre de colisiones a un vehículo autónomo que circula dentro de entornos físicos complejos. Lo anterior, debido a que actualmente los vehículos autónomos generan trayectorias sin tomar en cuenta las restricciones locales y globales en un mismo algoritmo. Esto produce en ciertos casos riesgos de colisión por falta de restricciones en el algoritmo de planeación de trayectorias. En consecuencia, la principal contribución de este trabajo es la inclusión de ambas restricciones en el algoritmo de planeación de trayectorias. Así mismo, se muestran los resultados obtenidos al realizar la simulación del algoritmo bajo diferentes circunstancias al igual que sus limitaciones.

Palabras clave: Planeación de trayectorias, Robótica Móvil, Vehículos Autónomos.

Abstract

This paper shows an obstacle avoidance technique based on the simulation of virtual magnetic fields (VMF) with an autonomous vehicles approach. The work probes that, at the time to generate a trajectory with VMF, it can include both local and global restrictions within the same algorithm to guarantee solutions with a free collision path for an autonomous vehicle, that circulates in a physical complex road. Nowadays the autonomous vehicles generate trajectories without considering local and global restrictions in the same algorithm. It produces, in certain cases, a risk of collision given the leak of restrictions in the path planning algorithm. Therefore, the main contribution of this work is the covering of both restrictions for the path planning algorithm. Likewise, the obtained results of the simulation are shown, and also the algorithm limitations.

Keywords: Path Planning, Mobil Robotics, Autonomous Vehicles.

1. Introducción

Dentro del campo de la robótica móvil existe un área dedicada a la planeación de trayectorias, misma que se divide en dos partes: planeación de trayectorias local y global [1].

La diferencia entre la planeación de trayectorias local y global radica en las restricciones que se toman en cuenta al momento de generar la trayectoria. En la planeación de trayectorias local se toma

en cuenta la información que se puede obtener del entorno que rodea al robot móvil o vehículo autónomo. Esta información se puede obtener mediante cámaras, sensores ultrasónicos, entre otros. Por otro lado, la planeación de trayectorias global incluye información adicional que restringe los movimientos del robot móvil a lo largo de la ruta. Por ejemplo, una restricción global es la orientación que debe seguir el robot para llegar a su objetivo, y los puntos por los que debe pasar independientemente de lo que la instrumentación del robot móvil pueda detectar.

La planeación de trayectorias ha utilizado a lo largo del tiempo dos variantes de algoritmos: algoritmos clásicos y algoritmos metaheurísticos. Los algoritmos clásicos tales como: Rapid Random Tree [2], Potential Fields [3], Descomposición por Celdas [4], entre otros, se utilizaron en el campo de la robótica desde los 80's entregando resultados positivos en aplicaciones con robots manipuladores y robótica de inspección. Sin embargo, la robótica móvil representa un reto que los algoritmos clásicos no buscan solucionar.

La robótica móvil (vehículos autónomos y robótica colaborativa) implica procesar una gran cantidad de información en poco tiempo, debido a que se mueven en entornos dinámicos, por lo que es necesario obtener resultados de la trayectoria de la forma más rápida posible, sin necesidad de que la solución otorgada sea la solución óptima. Los algoritmos clásicos buscan entregar la ruta óptima, sin importar el tiempo de cómputo. Por el contrario, los algoritmos metaheurísticos, entregan resultados en un menor tiempo de cómputo sin que necesariamente sea la ruta óptima.

Los algoritmos clásicos más utilizados para la planeación de trayectorias son: Rapid Random Tree (RRT) [2], Potential Fields (PF) [3] y Mapeo del camino mediante descomposición por celdas [4]. Mientras que los algoritmos metaheurísticos más utilizados son: Ant Colony [5], Redes Neuronales [6] y Algoritmos Genéticos [6].

El proyecto utiliza una variación del algoritmo PF [3] de forma que permite implementar restricciones locales y globales dentro del mismo algoritmo. Para esto se consideró un campo magnético virtual [7] que permite modificar fácilmente sus parámetros de atracción y repulsión magnética virtual de acuerdo a las restricciones del entorno.

Los campos magnéticos virtuales se basan en la simulación de partículas magnéticas de atracción y repulsión, siendo las partículas de atracción el punto de destino y las partículas de repulsión los obstáculos, tal como se muestra en la Figura 1, donde el destino es el círculo de color rojo y el obstáculo es el círculo de color azul.

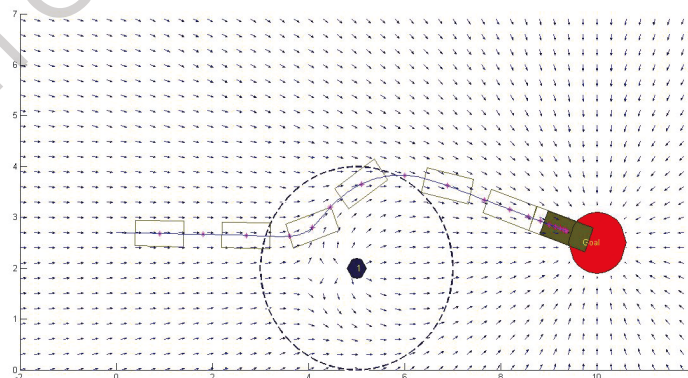


Figura 1. Campo magnético virtual.

Al momento de realizar la planeación de una trayectoria con un enfoque hacia vehículos autónomos, es necesario considerar restricciones que disminuyan al máximo el riesgo de colisión con

obstáculos u otros vehículos que estén circulando dentro del mismo entorno. Las restricciones que se consideran dentro de este proyecto son: geometría del camino por el cual circula el vehículo autónomo, número de carriles, geometría del vehículo y geometría del obstáculo.

Para el caso particular de la generación de trayectorias para vehículos autónomos, una trayectoria que se genera tomando en cuenta restricciones globales, buscará pasar por todos los puntos generados por la ruta planeada entre los puntos de origen y destino. También es necesario conocer la orientación del vehículo como se muestra en la Figura 2.

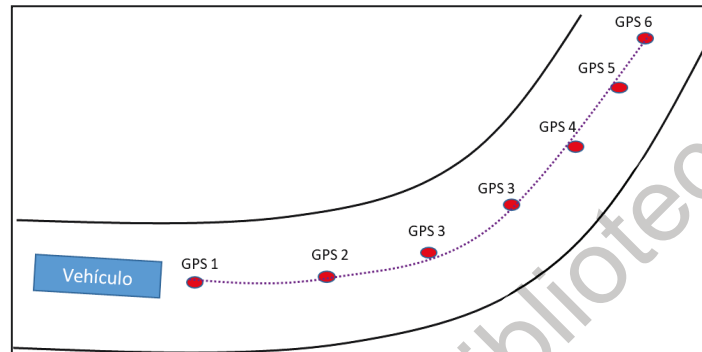


Figura 2. Trayectoria con restricciones globales.

Sin embargo, en el caso de que existan obstáculos en la trayectoria, como se muestra en la Figura 3, dichos obstáculos no serán considerados en el momento de generar la trayectoria global. Por otro lado, al momento de generar una trayectoria solo considerando restricciones locales no tomará en cuenta las coordenadas de la ruta planeada que se tienen que seguir, por lo tanto, el vehículo puede carecer de la orientación correcta lo que en el caso de vehículos autónomos implica una invasión de carriles ya que su única referencia es no salir del camino por el cual circula el vehículo autónomo.

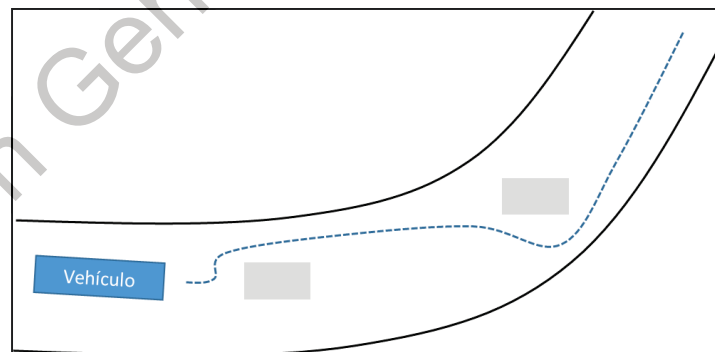


Figura 3. Trayectoria con restricciones locales.

El objetivo final del trabajo es adaptar el algoritmo CMV, de forma que permita una interacción constante entre las restricciones globales y locales para así obtener una trayectoria como se muestra en la Figura 4.

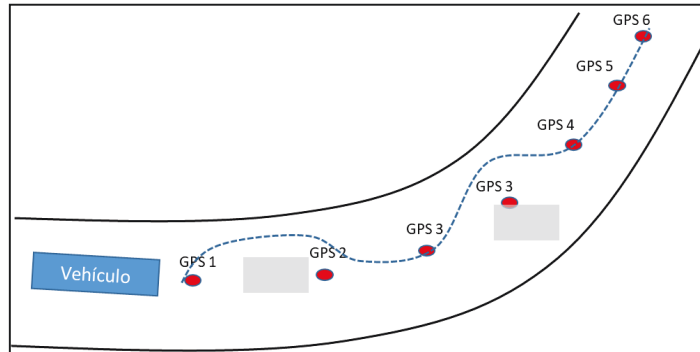


Figura 4. Trayectoria con restricciones locales y globales.

2. Algoritmo para Campos Magnéticos Virtuales (CMV)

El algoritmo está basado en la simulación de partículas magnéticas, mismas partículas que se conforman por las siguientes variables: Alcance (α), Intensidad (β) y Radio (γ). Existen dos tipos de partículas magnéticas, de atracción y de repulsión, mismas partículas magnéticas que se definen a continuación.

2.1 Partícula de atracción

La partícula de atracción en el entorno global se define como la coordenada a la que tiene que llegar el robot móvil. Por lo tanto, sin importar en qué punto del plano se encuentre el robot móvil tiene que ser atraído por dicha partícula tal y como se muestra en la Figura 5.

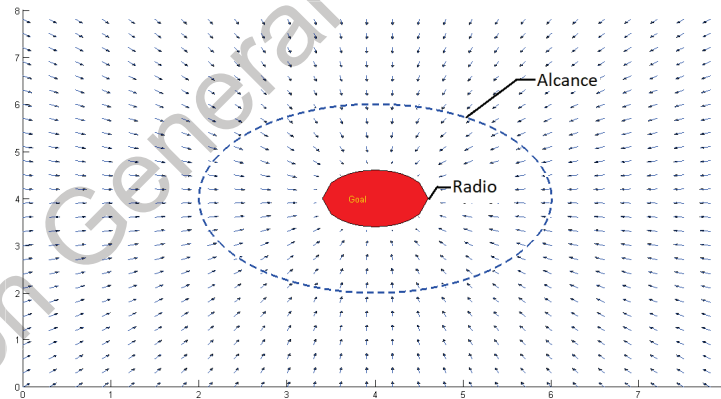


Figura 5. Partícula de atracción.

Mediante una ecuación gradiente se determina como afecta la partícula de atracción al robot móvil para así determinar en cada iteración el siguiente punto en el plano $[x, y]$ de tal forma que la posición en la iteración $P_x(i+1)$ y $P_y(i+1)$ (siendo $[P_x, P_y]$ el punto en el que se encuentra el robot móvil) se define de la siguiente manera:

$$P_x(i+1) = P_x(i) + \Delta A_x \tag{1}$$

$$P_y(i+1) = P_y(i) + \Delta A_y \tag{2}$$



La ecuación gradiente $(\Delta A_x, \Delta A_y)$ se genera con restricciones de acuerdo a la cercanía que se tiene con la partícula de atracción:

1. Gradiente estando fuera del Alcance (α_A)

$$\Delta A_x = \beta_A \alpha_A \cos(\theta_A) \quad (3)$$

$$\Delta A_y = \beta_A \alpha_A \sin(\theta_A) \quad (4)$$

2. Gradiente estando entre el Alcance (α_A) y el Radio (γ_A)

$$\Delta A_x = \beta_A (d_A - \alpha_A) \cos(\theta_A) \quad (5)$$

$$\Delta A_y = \beta_A (d_A - \alpha_A) \sin(\theta_A) \quad (6)$$

3. Gradiente estando dentro del Radio (γ_A)

$$\Delta A_x = 0 \quad (7)$$

$$\Delta A_y = 0 \quad (8)$$

En donde d_A representa la distancia entre el robot móvil y la partícula de atracción y θ_A se define como el ángulo entre el robot móvil y la partícula de atracción:

$$d_A = \sqrt{(P_x - x_A)^2 + (P_y - y_A)^2} \quad (9)$$

$$\theta_A = \arctan((y_A - P_y)/(x_A - P_x)) \quad (10)$$

En donde (x_A, y_A) representan las coordenadas de la partícula de atracción en el plano (x, y)

2.2 Partícula de repulsión

La partícula de repulsión en el entorno local se define como un obstáculo que se interpone en el trayecto del robot móvil. Por lo tanto, su objetivo es desviar al robot móvil de su trayectoria, si es que el robot está al alcance de esta partícula, como se muestra en la Figura 6.

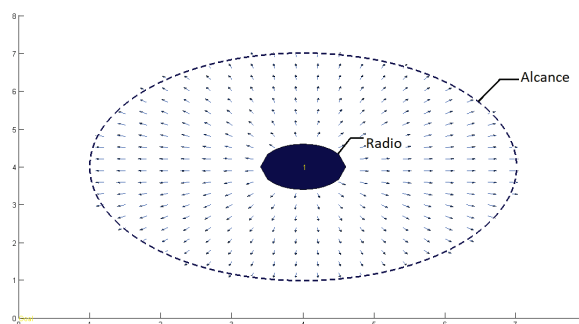


Figura 6. Partícula de repulsión.



A diferencia de la partícula de atracción, la partícula de repulsión solo tendrá efecto sobre la posición del robot móvil si es que este entra de su alcance previamente definido.

Mediante una ecuación gradiente se determina como afecta la partícula de repulsión al robot móvil para así determinar en cada iteración el siguiente punto en el plano $[x, y]$ de tal forma que la posición en la iteración $P_x(i+1)$ y $P_y(i+1)$ se define de la siguiente manera:

$$P_x(i+1) = P_x(i) + \Delta R_x \quad (11)$$

$$P_y(i+1) = P_y(i) + \Delta R_y \quad (12)$$

La ecuación gradiente $(\Delta R_x, \Delta R_y)$ se genera con restricciones de acuerdo a la cercanía que se tiene con la partícula de repulsión:

1. Gradiente estando fuera del Alcance (α_R)

$$\Delta R_x = 0 \quad (13)$$

$$\Delta R_y = 0 \quad (14)$$

2. Gradiente estando entre el Alcance (α_R) y el Radio (γ_R)

$$\Delta R_x = -\beta_R (d_R - \alpha_R) \cos(\theta_R) \quad (15)$$

$$\Delta R_y = -\beta_R (d_R - \alpha_R) \sin(\theta_R) \quad (16)$$

3. Gradiente estando dentro del Radio (γ_A)

$$\Delta R_x = \infty \quad (17)$$

$$\Delta R_y = \infty \quad (18)$$

En donde d_R representa la distancia entre el robot móvil y la partícula de repulsión y θ_R se define como el ángulo entre el robot móvil y la partícula de atracción:

$$d_R = \sqrt{(P_x - x_R)^2 + (P_y - y_R)^2} \quad (19)$$

$$\theta_R = \arctan((y_R - P_y)/(x_R - P_x)) \quad (20)$$

En donde (x_R, y_R) representan las coordenadas de la partícula de repulsión en (x, y) .

2.3 Sumatoria de campos magnéticos virtuales

La orientación correcta para el robot móvil se obtiene mediante la sumatoria del CMV de atracción (destino) y los CMV's de repulsión (obstáculos):

$$\Delta x = \Delta Ax + \sum \Delta Rx \quad (21)$$

$$\Delta y = \Delta Ay + \sum \Delta Ry \quad (22)$$

Se pueden llegar a tener más de una partícula de repulsión, pero no más de una partícula de atracción, ya que puede haber múltiples obstáculos, pero no múltiples objetivos.

Dentro del plano en el que influyen los CMV's, el robot móvil debe de ser capaz de llegar al objetivo (partícula de atracción) sin colisionar con el obstáculo (partícula de repulsión) desde cualquier punto del plano (x, y) tal y como se muestra en la Figura 7.

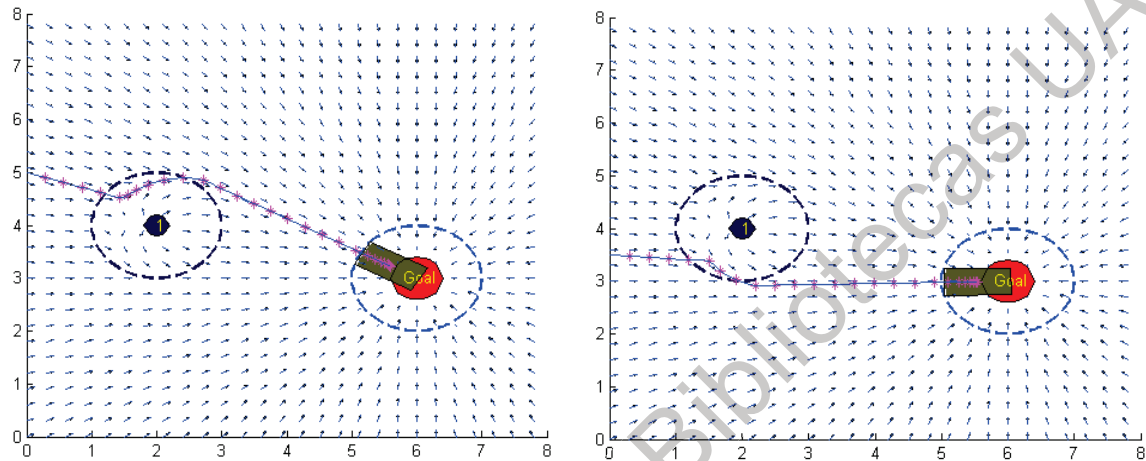


Figura 7. Alcance de Objetivo.

Al igual que puede haber obstáculos que no tengan ningún tipo de efecto sobre la trayectoria del robot móvil tal y como se muestra en la Figura 8.

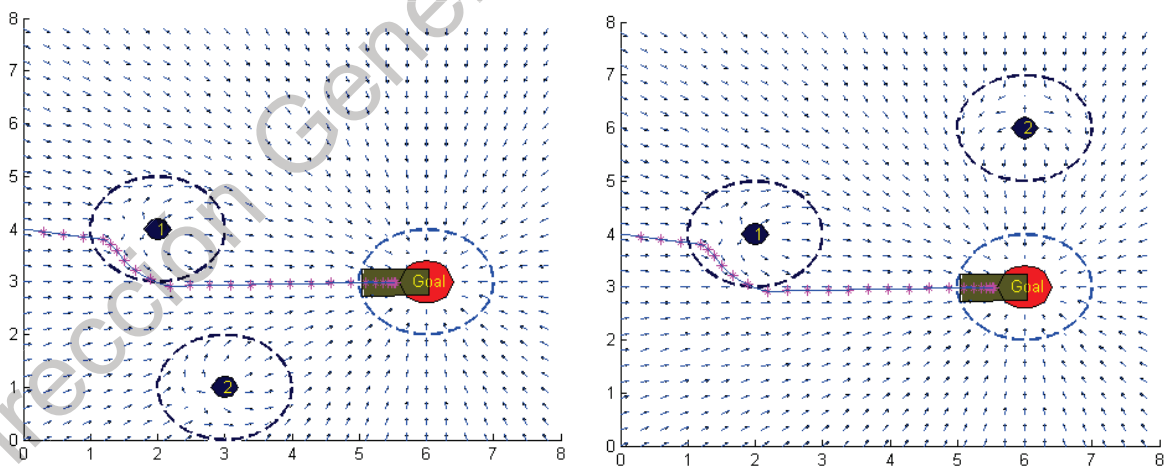


Figura 8. Partícula de repulsión sin efecto.



3. Metodología

La metodología que se utilizó para realizar la simulación del algoritmo CMV se muestra en la Figura 9.

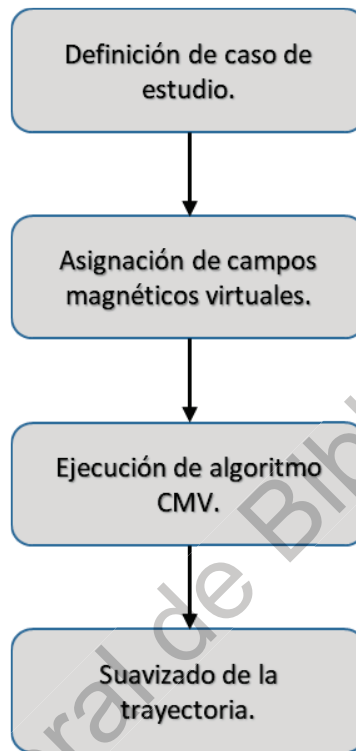


Figura 9. Metodología.

Para poder generar una trayectoria que garantice que no existe colisión es necesario definir casos de estudio para la implementación del algoritmo. Esto se muestra en el mapa conceptual de la Figura 10. Un caso del mapa conceptual se muestra en la Figura 11.

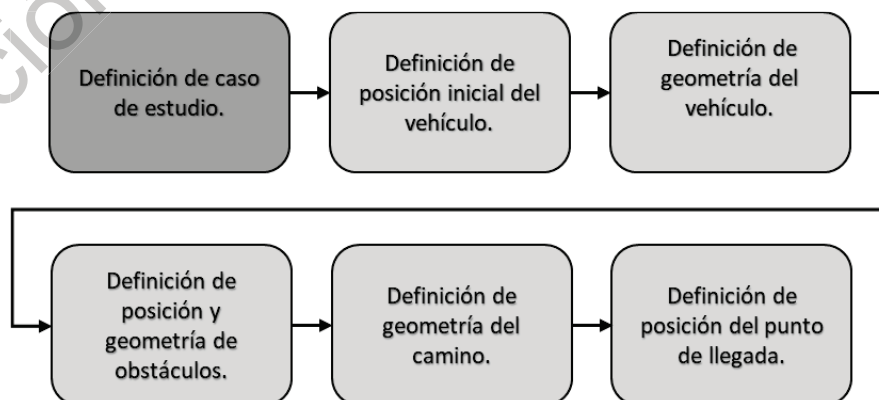


Figura 10. Mapa para la definición de caso de estudio.

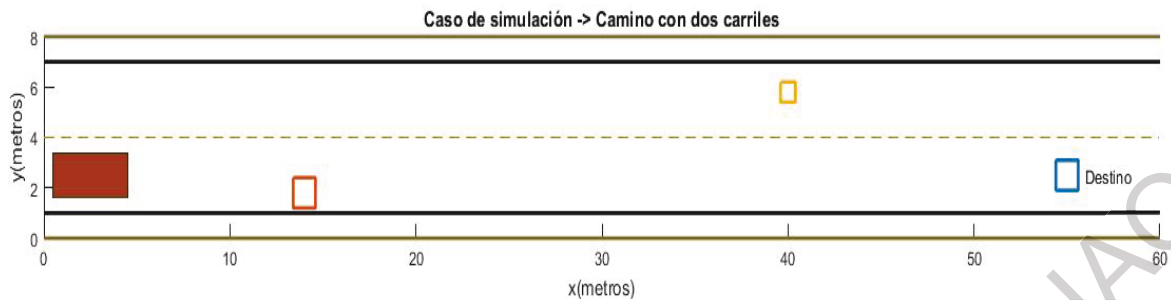


Figura 11. Definición de caso de estudio.

Una vez que se tiene definido el caso de estudio se asignan los campos magnéticos virtuales al destino y a los obstáculos. La Figura 12 muestra un mapa conceptual de la asignación de campos magnéticos virtuales, y la Figura 13 muestra un ejemplo de asignación de campos.

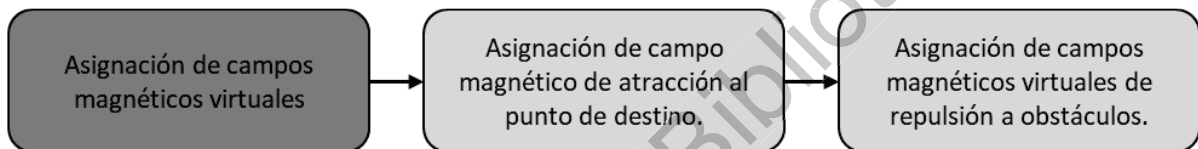


Figura 12. Mapa para la asignación de campos magnéticos virtuales.

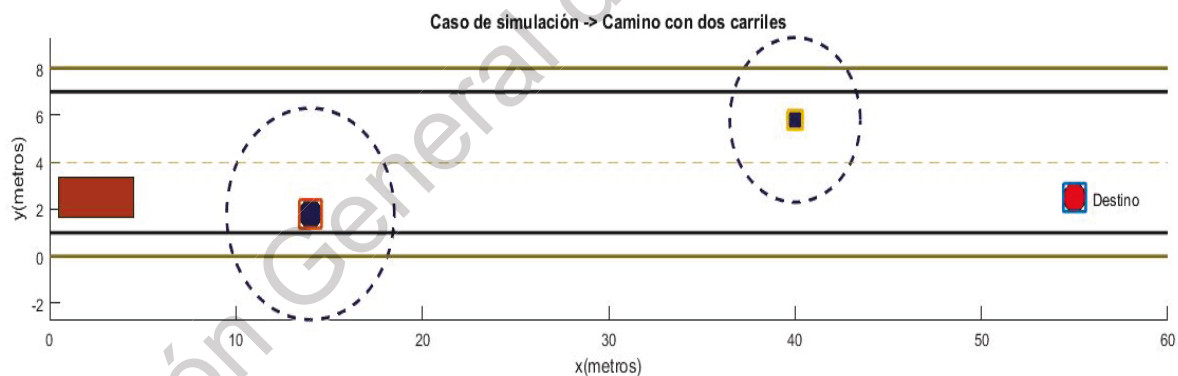


Figura 13. Asignación de campos magnéticos virtuales.

Una vez que se cuenta con la definición del caso de estudio y la asignación de los CMV se procede a la ejecución del algoritmo, de acuerdo al proceso de la Figura 14.

Una vez que el algoritmo termina su ejecución se obtiene una serie de puntos por los cuales tiene que pasar el vehículo, con advertencia en los puntos de una colisión en caso de existir, lo que se observa en la figura 15.

Finalmente se procede a realizar un suavizado de la trayectoria mediante una interpolación cúbica entre los puntos obtenidos por el algoritmo CMV. El resultado de la modificación a la trayectoria se muestra con la línea azul en la Figura 16.

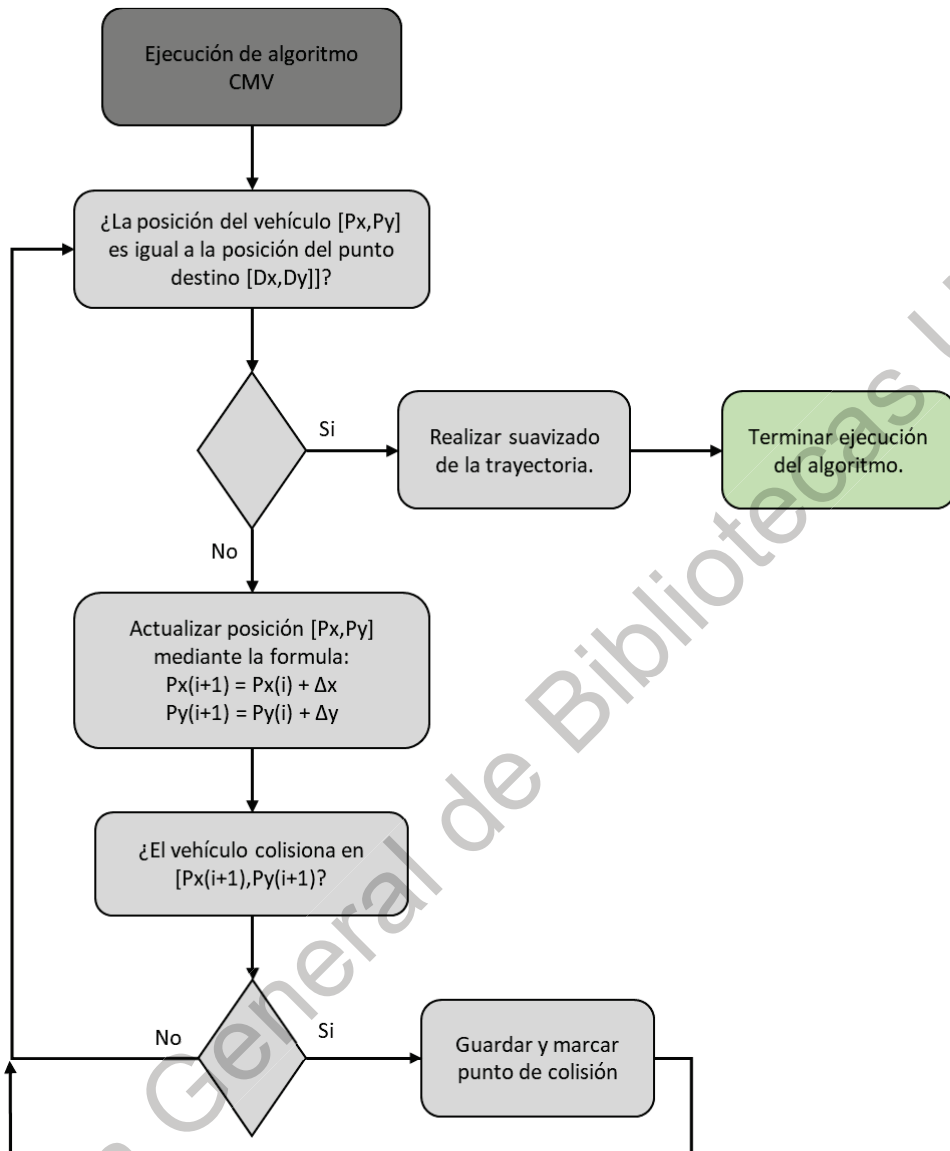


Figura 14. Descripción de algoritmo CMV.

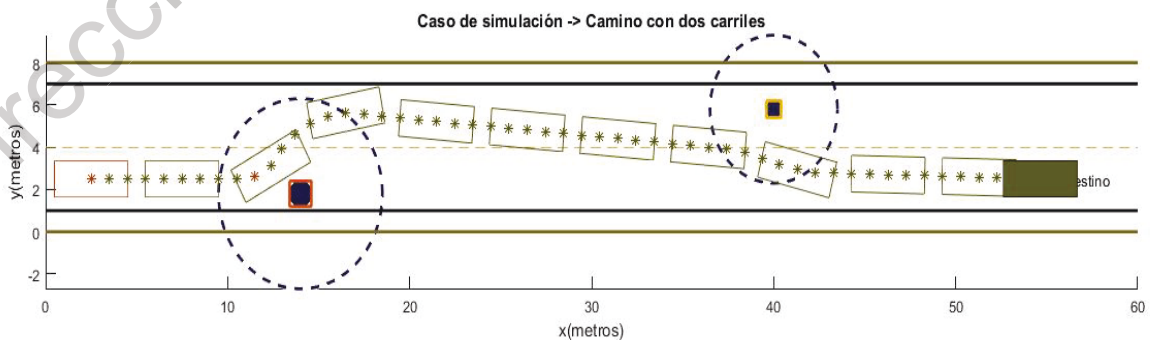


Figura 15. Ejecución de algoritmo CMV.

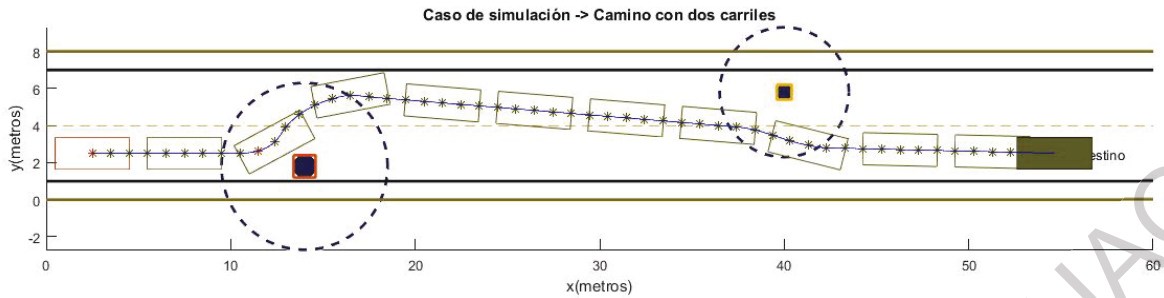


Figura 16. Suavizado de trayectoria.

4. Resultados

Como resultado del presente trabajo, se obtuvo la adaptación del algoritmo CMV para la solución al problema de planeación de trayectorias con restricciones globales y locales con las siguientes características:

1. Detección de colisión con obstáculos y límites del camino.
2. Definición de geometría del vehículo, obstáculos y camino.
3. Suavizado de trayectoria final.
4. Manipulación de constantes para la definición de campos de atracción y repulsión.

Dichas características se ilustran en la Figura 17.

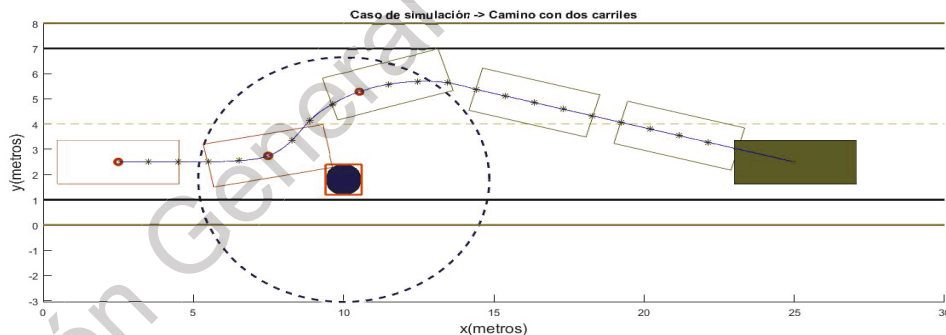


Figura 17. Características finales de algoritmo CMV.

El trabajo desarrollado permite el estudio de casos con obstáculos estáticos y cualquier geometría posible para el camino y los obstáculos. La asignación de las variables de alcance e intensidad se realiza en función de la colisión, ya sea con el obstáculo o con el camino, por lo que se pueden observar dos escenarios posibles. El primero, es un escenario en el cual no se toman en cuenta restricciones locales, como la geometría de los obstáculos o los límites del camino, lo cual produce múltiples colisiones a lo largo del trayecto, como se muestra en la Figura 18.

Sin embargo, al momento de tomar en cuenta las colisiones que se generan a lo largo del trayecto, y modificar las variables de alcance e intensidad de los obstáculos y el objetivo, se obtiene el resultado de la Figura 19.

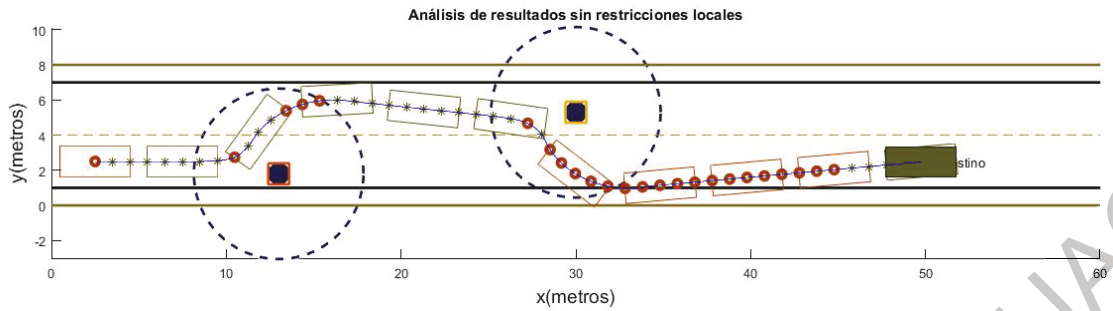


Figura 18. Análisis de resultados sin restricciones locales.

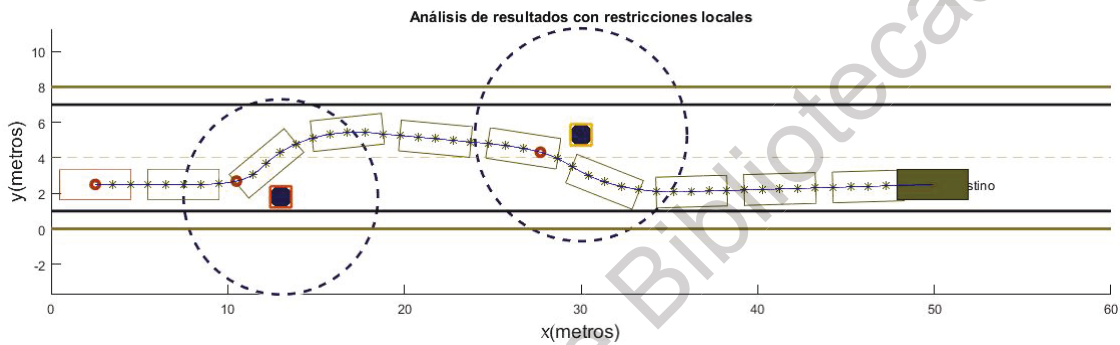


Figura 19. Análisis de resultados con restricciones locales y globales.

Al ajustar las variables de atracción y repulsión del objetivo y los obstáculos, respectivamente, se optimiza el algoritmo al pasar de 23 puntos de colisión a solo 2 puntos de colisión, mismos que se pueden evitar al definir una zona de seguridad entre el vehículo y los obstáculos.

4.1 Limitaciones y casos no favorables.

La ubicación de los obstáculos influye directamente en la dirección que tomará el vehículo autónomo para desviarse, por lo que es necesario modificar en ocasiones la posición del obstáculo ligeramente, para hacer que la desviación del vehículo no sea desfavorable. Es decir, que el camino que toma para la evasión del obstáculo sea el correcto. En la Figura 20 se ilustra el caso en que la posición del obstáculo es favorable para la desviación del vehículo, y la Figura 21 ilustra el caso de la posición desfavorable del obstáculo para el vehículo.

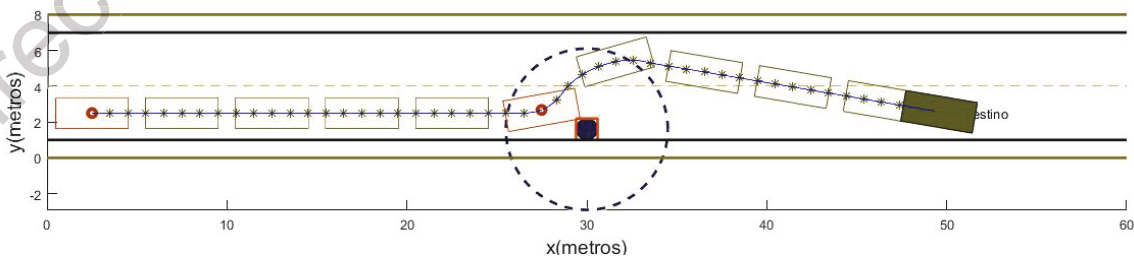


Figura 20. Caso favorable por posición del obstáculo.

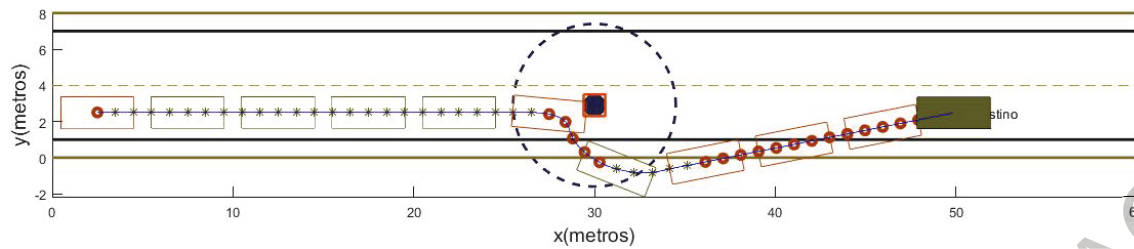


Figura 21. Caso desfavorable por posición del obstáculo.

También existe el caso en el que dos obstáculos crean un mínimo local que evita que la trayectoria se genere a través de dos obstáculos. Esto se muestra en la Figura 22, donde los campos magnéticos de repulsión de los obstáculos impiden que el vehículo pase en medio de ellos.

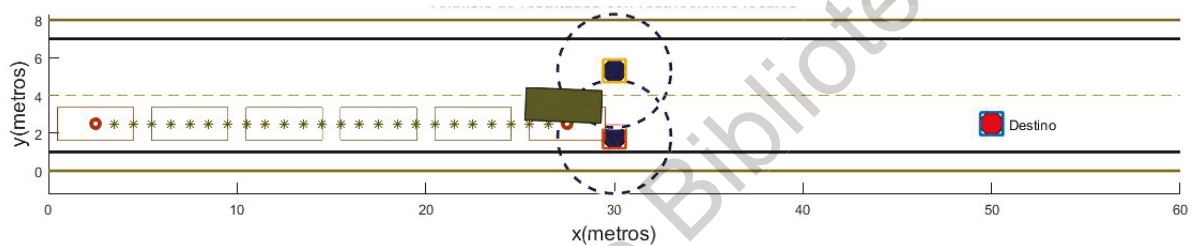


Figura 22. Caída en mínimo local.

Dicho caso se puede resolver al aumentar la intensidad del campo magnético de atracción, provocando que el vehículo pase por en medio de los obstáculos aun cuando al pasar existe colisión. Esto se muestra en la Figura 23.

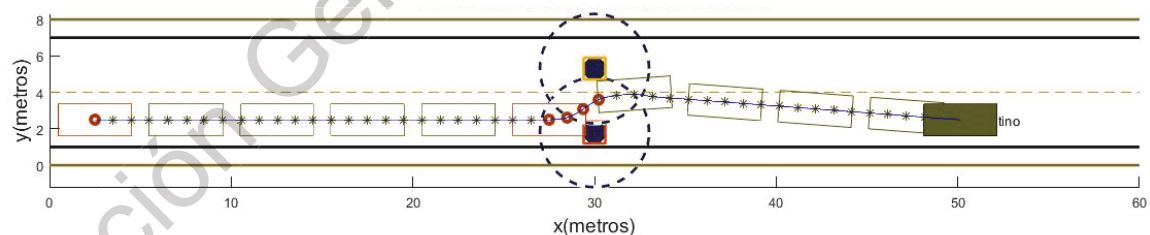


Figura 23. Salida de mínimo local.

5. Conclusiones

El algoritmo CMV se adapta bien al problema de planeación de trayectorias, debido a su rápida respuesta. Sin embargo, se requiere de una técnica para apoyar al algoritmo a definir sus variables de atracción y repulsión, para garantizar que no se ocasionará una colisión entre el robot móvil y su entorno. Actualmente, el algoritmo muestra limitaciones al no tener una variante para salir de casos en donde se producen caídas en mínimos locales al igual que una auto-asignación del alcance de los obstáculos en función de su geometría.



En la sección de resultados, se puede apreciar que cuando se utiliza un algoritmo sin considerar restricciones locales, la trayectoria que se genera crea un gran número de casos de colisión ya que las variables de alcance e intensidad magnética no están considerando las restricciones locales, como se observa en la Figura 18.

La adaptación del algoritmo CMV para la solución al problema de planeación de trayectorias con restricciones locales y globales, permite modificar las variables de atracción y repulsión para obtener la mejor trayectoria posible. Esto se puede apreciar en la Figura 20. Con esto, queda comprobado que el incluir las restricciones locales y globales en el algoritmo CMV existe una combinación de parámetros para las partículas de repulsión (obstáculos) y la partícula de atracción (objetivo) que generen una trayectoria libre de colisión.

Como trabajo futuro, está la posibilidad de adaptar el algoritmo de forma que pueda auto-asignar las variables de alcance e intensidad a las partículas de atracción y repulsión para de esa manera evitar colisiones, así como su implementación en un robot móvil a escala para la realización de pruebas experimentales.

En conclusión, los resultados son positivos, sin embargo, se requieren mejoras como las mencionadas en el análisis de resultados para su posterior implementación en vehículos de prueba.

Referencias

- [1] P. Marin-Plaza, A. Hussein, D. Martin y A. d. I. Escalera, "Global and Local Path Planning Study in a ROS-Based Research Platform for Autonomous Vehicles", Journal of Advanced Transportation, vol. 2018, 2018 (vid. pág. 1).
- [2] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning", 1998 (vid. pág. 1).
- [3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots", en Robotics and Automation. Proceedings. 1985 IEEE International Conference on, IEEE, vol. 2, 1985, págs. 500-505 (vid. pág. 1).
- [4] L. Zhang, Y. J. Kim y D. Manocha, "A hybrid approach for complete motion planning", en Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on, IEEE, 2007, págs. 7-14 (vid. pág. 1).
- [5] Guan-Zheng, T. A. N., Huan, H., & Sloman, A. "Ant colony system algorithm for real-time globally optimal path planning of mobile robots", 2007, págs 279-285.
- [6] Xin, D., Hua-hua, C., & Wei-kang, G. "Neural network and genetic algorithm based global path planning in a static environment". Journal of Zhejiang University-Science A, 2005, 6(6), 549-554.
- [7] M. A. Goodrich, "Potential fields tutorial", Class Notes, vol. 157, 2002 (vid. pág. 1)