



UNIVERSIDAD NACIONAL DE COLOMBIA

Planificación de trayectorias óptimas de robots móviles empleando el mapeo de celdas

Efraín Grisales Ramírez

Universidad Nacional de Colombia
Facultad de Ingeniería y Arquitectura
Departamento de Ingeniería Eléctrica, Electrónica y Computación
Manizales, Colombia
2023



UNIVERSIDAD NACIONAL DE COLOMBIA

Optimal Trajectory Planning for Mobile Robots Using Cell Mapping

Efraín Grisales Ramírez

Universidad Nacional de Colombia
Facultad de Ingeniería y Arquitectura
Departamento de Ingeniería Eléctrica, Electrónica y Computación
Manizales, Colombia
2023

Planificación de trayectorias óptimas de robots móviles empleando el mapeo de celdas

Efraín Grisales Ramírez

Tesis presentada como requisito parcial para optar al título de:
Doctor en Ingeniería - Línea Automática

Director:
Ph.D. Gustavo Adolfo Osorio Londoño

Línea de Investigación:
Sistemas de Control y Robótica
Grupo de Investigación:
Percepción y Control Inteligente (PCI)

Universidad Nacional de Colombia
Facultad de Ingeniería y Arquitectura
Departamento de Ingeniería Eléctrica, Electrónica y Computación
Manizales, Colombia
2023

A mi familia y mi novia, los cuales contribuyeron en gran parte para conseguir este logro.

“Cuanto mayor sea el esfuerzo, mayor es la gloria”.

Pierre Corneille

Agradecimientos

En primer lugar quiero agradecer a mi familia por brindarme el apoyo y la oportunidad de estudiar a pesar de las dificultades. A mi novia Alexandra Paola por apoyarme en este último paso para culminar mi formación. A mi Director Gustavo Adolfo Osorio Londoño, por sus invaluable contribuciones, su incondicional apoyo durante el desarrollo de esta investigación, su continua motivación y paciencia en estos años. Al sistema nacional de becas de la Universidad Nacional de Colombia por el apoyo en mi formación doctoral. A la Universidad Nacional de Colombia Sede Manizales por su valioso aporte en mi formación profesional y posgradual. A los integrantes de los Grupos de Trabajo Académico PCI y SHAC, por su apoyo y contribuciones durante este proceso.

Resumen

El problema de planificación de trayectorias es de importancia dentro de la robótica móvil, puesto que es indispensable generar caminos de referencia que eviten obstáculos y de fácil seguimiento por los sistemas autónomos. La solución dada por los algoritmos de planificación puede ser de diferente naturaleza, aumentando o disminuyendo la complejidad computacional. En este documento se propone un método para la planificación de trayectorias de robots móviles empleando el algoritmo de mapeo de celdas (AMC), el cual contiene las restricciones cinemáticas y dinámicas del robot, así como las del entorno, permitiendo generar trayectos de fácil seguimiento por el sistema. También contiene la información necesaria con el fin de dar solución al problema de optimización multiobjetivo.

El algoritmo consiste en dividir el espacio de estados en celdas y luego simular la evolución del sistema dinámico del móvil para cada condición inicial; con esto se generan grafos que contienen la información de las conexiones entre celdas en términos de distancia, tiempo, esfuerzo de control y energía. Con esta información se solucionan los problemas de optimización simple o multiobjetivo, según sea la función de costo empleada para encontrar el camino más corto entre dos puntos con el algoritmo de Dijkstra.

Además, en este documento se introducen algunos conceptos de control óptimo que serán empleados en la planificación de trayectorias de robots móviles. Estos conceptos se aplican inicialmente al doble integrador (DI) en tiempo continuo, debido a que éste emula el comportamiento de un robot sin masa. Posteriormente se implementa el AMC para solucionar los mismos problemas de optimización en forma discreta para el movimiento en 1 y 2 dimensiones, y se extiende el concepto a la aplicación en robots móviles de guiado diferencial, así como para el Robotino de FESTO.

Palabras clave: Algoritmo de Dijkstra, Algoritmo de Mapeo de Celdas, doble integrador, modelo dinámico, optimización combinatoria, optimización continua, optimización discreta, robot móvil de guiado diferencial, teoría de control óptimo.

Abstract

The trajectory planning problem is of importance in mobile robotics since it is essential to generate reference paths that avoid obstacles and are easy to follow for autonomous systems. The solutions provided by planning algorithms can vary in nature, either increasing or decreasing computational complexity. This paper presents a method for planning mobile robot trajectories using the Cell Mapping Algorithm (CMA), which incorporates the kinematic and dynamic constraints of the robot as well as those of the environment. This approach enables the generation of trajectories that are easy for the system to follow and includes the necessary information to address multi-objective optimization problems.

The algorithm involves dividing the state space into cells and then simulating the evolution of the mobile robot's dynamic system for each initial condition. This process generates graphs that contain information about the connections between cells in terms of distance, time, control effort, and energy. With this information, simple or multi-objective optimization problems can be solved, depending on the cost function used to find the shortest path between two points, typically implemented with Dijkstra's algorithm.

Also, this thesis introduces some optimal control concepts that will be used in path planning of mobile robots. These concepts are initially applied to continuous-time double integrator (DI), because it emulates behavior of a massless robot. Subsequently, the AMC is implemented to solve same optimization problems in discrete form for motion in 1 and 2 dimensions, and concept is extended to the application in differential guided mobile robots, as well as for FESTO Robotino.

Keywords: Dijkstra Algorithm, Cell Mapping Algorithm, discrete optimization, double integrator, dynamic model, combinatorial optimization, continuous optimization, differential guided mobile robot, optimal control theory.

Contenido

| | |
|--|-------------|
| Agradecimientos | IX |
| Resumen | XI |
| Contenido | XIII |
| Lista de Figuras | XVII |
| Lista de Tablas | XXI |
| Lista de símbolos | XXII |
| 1. Introducción | 1 |
| 1.1. Estructura del documento | 5 |
| I. Optimización continua y discreta del DI en 1D | 6 |
| 2. Control óptimo para el DI | 7 |
| 2.1. Problema de optimización simple (POS) | 7 |
| 2.2. POS para el DI | 10 |
| 2.2.1. Mínimo esfuerzo de control | 11 |
| 2.2.2. Minimización de la variación total | 13 |
| 2.2.3. Mínimo tiempo de navegación | 15 |
| 2.2.4. Problema de Fuller | 19 |
| 2.2.5. Minimización del gasto energético | 21 |
| 2.3. Problema de optimización multiobjetivo (POM) | 21 |
| 2.3.1. POM para el DI | 24 |
| 2.4. Conclusión | 26 |
| 3. Optimización combinatoria empleando el AMC para el DI en 1D | 27 |
| 3.1. Mapeo de Celdas | 27 |
| 3.2. AMC para el DI | 29 |
| 3.2.1. Vecinos de una celda | 29 |
| 3.2.2. Generación del grafo a partir de los vecinos de celda | 30 |

| | | |
|--------|---|----|
| 3.2.3. | Algoritmo de búsqueda del camino más corto | 32 |
| 3.3. | Solución al problema de optimización combinatoria para el DI por medio del AMC | 34 |
| 3.3.1. | Generación de trayectoria con menor esfuerzo de control | 34 |
| 3.3.2. | Generación de trayectoria con menor tiempo de navegación | 37 |
| 3.3.3. | Generación de trayectoria para el desplazamiento con menor energía cinética | 38 |
| 3.3.4. | Generación de trayectoria para solucionar el POM con parámetros de tiempo y esfuerzo de control | 39 |
| 3.3.5. | Generación de trayectoria para solucionar el POM con parámetros de tiempo, energía cinética y esfuerzo de control | 40 |
| 3.4. | Análisis de las diferentes trayectorias generadas por el AMC | 41 |
| 3.5. | Conclusión | 42 |

II. Planificación de trayectorias del DI en 2D 43

4. Planificación de trayectorias 44

| | | |
|--------|---|----|
| 4.1. | Planificadores de trayectorias | 44 |
| 4.1.1. | Computación reactiva | 45 |
| 4.1.2. | Soft Computing | 46 |
| 4.1.3. | Búsqueda del C-Space | 46 |
| 4.1.4. | Control Óptimo | 46 |
| 4.2. | Algoritmos de planificación de trayectorias | 47 |
| 4.2.1. | Grafo de visibilidad | 47 |
| 4.2.2. | Diagrama de Voronoi | 47 |
| 4.2.3. | Algoritmos de búsqueda del camino más corto | 47 |
| 4.2.4. | Algoritmo A* (A star) | 48 |
| 4.2.5. | Algoritmo D* Lite (D star Lite) | 49 |
| 4.2.6. | Campos potenciales artificiales (CPA) | 49 |
| 4.2.7. | Algoritmo de Bug | 50 |
| 4.2.8. | Algoritmos bioinspirados | 50 |
| 4.2.9. | Árboles aleatorios de búsqueda rápida | 51 |
| 4.3. | Planteamiento del problema de planificación de trayectorias | 51 |
| 4.4. | Implementación del planificador por CPA | 51 |
| 4.5. | Implementación del planificador A* | 52 |
| 4.6. | Implementación del planificador RRT* | 53 |
| 4.7. | AMC para el DI en 2D | 54 |
| 4.7.1. | Vecinos de una celda con $\dot{x}_0 = 0$ y $\dot{y}_0 = 0$ | 55 |
| 4.7.2. | Vecinos de una celda con $\dot{x}_0 > 0$ y $\dot{y}_0 > 0$ | 56 |
| 4.7.3. | Vecinos de una celda con $\dot{x}_0 < 0$ y $\dot{y}_0 < 0$ | 57 |
| 4.7.4. | Vecinos de una celda con $\dot{x}_0 > 0$ y $\dot{y}_0 < 0$ | 57 |

| | | |
|-------------|--|------------|
| 4.7.5. | Construcción del grafo | 59 |
| 4.7.6. | Trayectoria con mínima distancia (AMC_d) | 59 |
| 4.8. | Análisis de las rutas planificadas | 62 |
| 4.9. | Conclusión | 63 |
| 5. | Planificación de trayectorias óptimas para el DI en 2D empleando el AMC | 64 |
| 5.1. | AMC para la planificación de trayectorias del DI en 2D | 64 |
| 5.1.1. | Generación de trayectoria para minimizar el esfuerzo de control (AMC_e) | 64 |
| 5.1.2. | Generación de trayectoria con menor tiempo de navegación (AMC_t) | 67 |
| 5.1.3. | Generación de trayectoria con POM de distancia y tiempo (AMC_{dt}) | 69 |
| 5.1.4. | Generación de trayectoria con POM de distancia y esfuerzo de control (AMC_{de}) | 72 |
| 5.1.5. | Generación de trayectoria con POM de tiempo, esfuerzo de control y distancia (AMC_{POM}) | 74 |
| 5.1.6. | Análisis de las diferentes trayectorias generadas por el AMC | 77 |
| 5.2. | Complejidad computacional del AMC | 78 |
| 5.3. | Conclusión | 80 |
| III. | Planificación de trayectorias de Robots móviles | 81 |
| 6. | Modelo dinámico de robots móviles de guiado diferencial | 82 |
| 6.1. | Realización en el espacio de estados | 86 |
| 6.2. | Control Look-Ahead | 88 |
| 6.3. | Dinámica interna | 89 |
| 6.4. | Simulaciones de la evolución del modelo dinámico linealizado | 94 |
| 6.4.1. | Movimiento sobre el eje x con condiciones iniciales nulas | 94 |
| 6.4.2. | Movimiento sobre el eje y con condiciones iniciales nulas | 97 |
| 6.4.3. | Movimiento en dos dimensiones con $\phi = 0^\circ$ | 100 |
| 6.4.4. | Movimiento en dos dimensiones con $\phi = 135^\circ$ | 103 |
| 6.5. | Conclusión | 106 |
| 7. | Planificación de trayectorias de robots móviles empleando el AMC | 107 |
| 7.1. | AMC para robots móviles | 107 |
| 7.1.1. | Generación de trayectoria con optimización de distancia | 108 |
| 7.1.2. | Generación de trayectoria con optimización de esfuerzo de control | 109 |
| 7.1.3. | Generación de trayectoria con menor tiempo de navegación | 111 |
| 7.1.4. | Generación de trayectoria con POM de esfuerzo de control, distancia y tiempo | 112 |
| 7.1.5. | Análisis de resultados de los planificadores de trayectorias | 114 |

| | |
|--|------------|
| 7.2. AMC para generar trayectorias con mínimo gasto energético en robots móviles de guiado diferencial | 115 |
| 7.2.1. Cálculo de energía para el movimiento en una dimensión con AMC | 117 |
| 7.2.2. Generación de trayectoria para el desplazamiento con menor consumo de energía por medio del AMC | 119 |
| 7.3. Aplicación del AMC para la planificación de trayectorias de robots omnidireccionales con 3 ruedas | 120 |
| 7.4. Conclusión | 123 |
| Bibliografía | 125 |

Lista de Figuras

| | | |
|-------------|---|----|
| 2-1. | Partícula en 1D. | 7 |
| 2-2. | Comportamiento de la partícula en el tiempo para minimizar esfuerzo de control. . . | 13 |
| 2-3. | Diagrama de estados y comportamiento de la partícula en el tiempo para minimizar la variación total. | 15 |
| 2-4. | Diagrama de estado y superficie de conmutación para trayectoria óptima en tiempo. | 18 |
| 2-5. | Comportamiento de la partícula en el tiempo para trayectoria óptima en tiempo. . . | 18 |
| 2-6. | Comportamiento de la partícula en el tiempo para el problema de Fuller. | 20 |
| 2-7. | Espacio de estados de la partícula para el problema de Fuller. | 20 |
| 2-8. | Conjunto de Pareto y señal de control para el POM. | 25 |
| 2-9. | Señales en el tiempo del POM para $\alpha = 0.8$ | 26 |
| 3-1. | Celdas vecinas con diferentes condiciones iniciales de velocidad. El asterisco azul representa la celda inicial y los rojos las celdas finales o vecinas. | 30 |
| 3-2. | Conexiones entre celdas para el movimiento del DI en 1D. | 31 |
| 3-3. | Posición, velocidad y aceleración para minimizar esfuerzo de control. Los puntos rojos muestra la solución del AMC, la línea negra es una interpolación de los valores del AMC y la línea azul es la solución continua. | 35 |
| 3-4. | Error del AMC en la minimización del esfuerzo de control. | 36 |
| 3-5. | Variables de estado para minimizar esfuerzo de control. Los puntos rojos muestra la solución del AMC con la segunda discretización, la línea negra es una interpolación de los valores del AMC y la línea azul es la solución continua. | 36 |
| 3-6. | Posición, velocidad y aceleración para minimizar el tiempo de navegación. Los puntos rojos muestra la solución del AMC, la línea negra es una interpolación de los valores del AMC y la línea azul es la solución continua. | 38 |
| 3-7. | Posición, velocidad y aceleración para minimizar el consumo energético. Los puntos rojos muestra la solución del AMC y la línea negra es una interpolación de estos. | 39 |
| 3-8. | Posición, velocidad y aceleración para minimizar tiempo y esfuerzo de control. Los puntos rojos muestra la solución del AMC y la línea negra es una interpolación de estos. | 40 |
| 3-9. | Posición, velocidad y aceleración para minimizar tiempo, energía y esfuerzo de control. Los puntos rojos muestra la solución del AMC y la línea negra es una interpolación de los valores del AMC. | 41 |

| | | |
|--------------|---|----|
| 4-1. | Clasificación general de métodos de planificación de trayectorias. | 45 |
| 4-2. | Planificación de trayectorias con el algoritmo de campos potenciales artificiales. . . | 52 |
| 4-3. | Planificación de trayectorias con el algoritmo A*. | 52 |
| 4-4. | Algoritmo RRT*. (a) distancia de 0.3 m, (b) distancia de 0.5 m, (c) distancia de 0.8 m | 54 |
| 4-5. | Análisis de celdas vecinas en posición y velocidad para $\dot{x}_0 = 0$ y $\dot{y}_0 = 0$ | 56 |
| 4-6. | Análisis de celdas vecinas en posición y velocidad para $\dot{x}_0 > 0$ y $\dot{y}_0 > 0$ | 56 |
| 4-7. | Análisis de celdas vecinas en posición y velocidad para $\dot{x}_0 < 0$ y $\dot{y}_0 < 0$ | 57 |
| 4-8. | Análisis de celdas vecinas en posición y velocidad para $\dot{x}_0 > 0$ y $\dot{y}_0 < 0$ | 58 |
| 4-9. | Señales de control. | 58 |
| 4-10. | Conexiones entre celda y posición en plano para optimización de distancia. | 60 |
| 4-11. | Posición en el tiempo para optimización de distancia. | 60 |
| 4-12. | Posición y velocidad en el tiempo para optimización de distancia. | 61 |
| 4-13. | Señales de control para optimización de distancia. | 61 |
| 4-14. | Rutas generadas. | 62 |
| 5-1. | Conexiones entre celda y posición en plano para optimización de esfuerzo de control. | 65 |
| 5-2. | Posición en el tiempo para optimización de esfuerzo de control. | 66 |
| 5-3. | Posición y velocidad en el tiempo para optimización de esfuerzo de control. | 66 |
| 5-4. | Señales de control para optimización de esfuerzo de control. | 67 |
| 5-5. | Conexiones entre celda y posición en plano para menor tiempo de navegación. | 68 |
| 5-6. | Posición en el tiempo para menor tiempo de navegación. | 68 |
| 5-7. | Velocidad en el tiempo para menor tiempo de navegación. | 69 |
| 5-8. | Señales de control para menor tiempo de navegación. | 69 |
| 5-9. | Posición en el plano para optimización de distancia y tiempo. | 70 |
| 5-10. | Posición en el tiempo para optimización de distancia y tiempo. | 71 |
| 5-11. | Velocidad en el tiempo para optimización de distancia y tiempo. | 71 |
| 5-12. | Señales de control para optimización de distancia y tiempo. | 72 |
| 5-13. | Posición en el plano para optimización de distancia y esfuerzo de control. | 73 |
| 5-14. | Posición en el tiempo para optimización de distancia y esfuerzo de control. | 73 |
| 5-15. | Velocidad en el tiempo para optimización de distancia y esfuerzo de control. | 74 |
| 5-16. | Señales de control para optimización de distancia y esfuerzo de control. | 74 |
| 5-17. | Posición en el plano para optimización de tiempo, esfuerzo de control y distancia. . | 75 |
| 5-18. | Posición en el tiempo para optimización de tiempo, esfuerzo de control y distancia. | 76 |
| 5-19. | Velocidad en el tiempo para optimización de tiempo, esfuerzo de control y distancia. | 76 |
| 5-20. | Señales de control para optimización de tiempo, esfuerzo de control y distancia. . . | 77 |
| 5-21. | Comparación de planificadores con AMC | 77 |
| 6-1. | Robot móvil de guiado diferencial. | 82 |
| 6-2. | Señales de control en el tiempo para movimiento sobre el eje x. | 94 |
| 6-3. | Posición en el plano del robot para movimiento sobre el eje x. | 95 |
| 6-4. | Posición del robot en el tiempo para movimiento sobre el eje x. | 95 |

| | | |
|-------|---|-----|
| 6-5. | Velocidad del robot en el tiempo para movimiento sobre el eje x. | 96 |
| 6-6. | Posición y velocidad angular del móvil para movimiento sobre el eje x. | 96 |
| 6-7. | Posición y velocidad angular de las ruedas para movimiento sobre el eje x. | 97 |
| 6-8. | Señales de control en el tiempo para movimiento sobre el eje y. | 97 |
| 6-9. | Posición en el plano del robot para movimiento sobre el eje y. | 98 |
| 6-10. | Posición del robot en el tiempo para movimiento sobre el eje y. | 98 |
| 6-11. | Velocidad del robot en el tiempo para movimiento sobre el eje y. | 99 |
| 6-12. | Posición y velocidad angular del móvil para movimiento sobre el eje y. | 99 |
| 6-13. | Posición y velocidad angular de las ruedas para movimiento sobre el eje y. | 100 |
| 6-14. | Señales de control en el tiempo para movimiento en el plano con $\phi = 0^\circ$ | 100 |
| 6-15. | Posición del robot para movimiento en el plano con $\phi = 0^\circ$ | 101 |
| 6-16. | Posición del robot en el tiempo para movimiento en el plano con $\phi = 0^\circ$ | 101 |
| 6-17. | Velocidad del robot en el tiempo para movimiento en el plano con $\phi = 0^\circ$ | 102 |
| 6-18. | Posición y velocidad angular del móvil para movimiento en el plano con $\phi = 0^\circ$ | 102 |
| 6-19. | Posición y velocidad angular de las ruedas para movimiento en el plano con $\phi = 0^\circ$ | 103 |
| 6-20. | Señales de control en el tiempo para movimiento en el plano con $\phi = 135^\circ$ | 103 |
| 6-21. | Posición del robot para movimiento en el plano con $\phi = 135^\circ$ | 104 |
| 6-22. | Posición del robot en el tiempo para movimiento en el plano con $\phi = 135^\circ$ | 104 |
| 6-23. | Velocidad del robot en el tiempo para movimiento en el plano con $\phi = 135^\circ$ | 105 |
| 6-24. | Posición y velocidad angular del móvil para movimiento en el plano con $\phi = 135^\circ$ | 105 |
| 6-25. | Posición y velocidad angular de las ruedas para movimiento con $\phi = 135^\circ$ | 106 |
| | | |
| 7-1. | Ruta para menor distancia. | 108 |
| 7-2. | Comportamiento de las ruedas para trayectoria optimizada en distancia. | 108 |
| 7-3. | Comportamiento del robot para trayectoria optimizada en distancia. | 109 |
| 7-4. | Ruta para menor esfuerzo de control. | 110 |
| 7-5. | Comportamiento de las ruedas para trayectoria optimizada en esfuerzo de control. | 110 |
| 7-6. | Comportamiento del robot para trayectoria optimizada en esfuerzo de control. | 111 |
| 7-7. | Ruta para menor tiempo de navegación. | 111 |
| 7-8. | Comportamiento de las ruedas para trayectoria optimizada en tiempo. | 112 |
| 7-9. | Comportamiento del robot para trayectoria optimizada en tiempo. | 112 |
| 7-10. | Ruta para POM | 113 |
| 7-11. | Comportamiento de las ruedas para trayectoria con optimización multiobjetivo. | 113 |
| 7-12. | Comportamiento del robot para trayectoria con optimización multiobjetivo. | 114 |
| 7-13. | Comparación planificadores de trayectorias. | 114 |
| 7-14. | Posición y velocidad. | 117 |
| 7-15. | Aceleración del móvil. | 117 |
| 7-16. | Posición y velocidad angular de las ruedas. | 118 |
| 7-17. | Aceleración angular de las ruedas. | 118 |
| 7-18. | Energía y potencia consumida por cada rueda. | 118 |

| | |
|---|-----|
| 7-19. Desplazamiento del robot en el plano. | 119 |
| 7-20. Posición, velocidad y aceleración para minimizar consumo energético. | 120 |
| 7-21. Sistemas coordenados del robot móvil. | 121 |
| 7-22. Ruta planificada por el AMC y posición en el plano del Robotino [®] | 121 |
| 7-23. Posiciones y velocidades dadas por el AMC y el Robotino [®] | 122 |
| 7-24. Velocidad angular de cada rueda. | 123 |

Lista de Tablas

| | | |
|-------------|---|-----|
| 3-1. | Comparación de trayectorias del movimiento de una partícula en 1D. | 41 |
| 4-1. | Comparación en distancia de las rutas generadas. | 63 |
| 5-1. | Comparación de trayectorias del movimiento de una partícula en 2D. | 78 |
| 7-1. | Comparación de trayectorias del movimiento de un robot en el plano. | 115 |

Lista de símbolos

Símbolos con letras latinas

| Símbolo | Término | Unidad SI |
|-------------|--|-----------------------------|
| b | Distancia entre cada rueda y el eje de simetría | m |
| c | Constante igual a $\frac{r}{2b}$ | $\frac{\text{m}}{\text{m}}$ |
| d | Distancia entre el punto P_0 y el centro de masa del robot | m |
| I_c | Momento de inercia del robot sin ruedas ni motor | kg.m ² |
| I_m | Momento de inercia de cada rueda con motor cerca al diámetro de la rueda | kg.m ² |
| I_ω | Momento de inercia de cada rueda con motor | kg.m ² |
| m_c | Masa del robot sin ruedas ni motor | kg |
| m_ω | Masa de cada rueda con motor | kg |
| r | Radio de las ruedas | m |
| x_1 | Variable de estado 1 | |
| x_2 | Variable de estado 2 | |
| \dot{x}_1 | Derivada de variable estado 1 | |
| \dot{x}_2 | Derivada de variable estado 2 | |
| J | Función objetivo del POS en tiempo continuo | |
| L | Lagrangiano | |
| H | Función Hamiltoniana | |
| G_t | Grafo con los costos de tiempo para movimiento entre celdas | |
| G_E | Grafo con los costos de energía para movimiento entre celdas | |
| G_{E_k} | Grafo con los costos de energía cinética para movimiento entre celdas | |
| G_u | Grafo con los costos de esfuerzo de control para movimiento entre celdas | |
| G_d | Grafo con los costos de distancia para movimiento entre celdas | |
| f_k | Funciones objetivo del problema de optimización | |

| Símbolo | Término | Unidad SI |
|----------------|---|------------------|
| F | Conjunto de las funciones objetivo del problema de optimización | |
| n_{cellsp} | Cantidad de particiones en la dimensión de posición | |
| n_{cellsv} | Cantidad de particiones en la dimensión de velocidad | |

Símbolos con letras griegas

| Símbolo | Término | Unidad SI |
|------------------|--|-------------------------------|
| θ_1 | Posición angular rueda 1 | rad |
| θ_2 | Posición angular rueda 2 | rad |
| $\dot{\theta}_1$ | Velocidad angular rueda 1 | $\frac{\text{rad}}{\text{s}}$ |
| $\dot{\theta}_2$ | Velocidad angular rueda 2 | $\frac{\text{rad}}{\text{s}}$ |
| ϕ | Orientación del móvil con respecto a x_1 | rad |
| $\dot{\phi}$ | Velocidad angular del móvil en se | $\frac{\text{rad}}{\text{s}}$ |
| λ | Multiplicadores de Lagrange | |

Abreviaturas

| Abreviatura | Término |
|--------------------|--|
| AMC | Algoritmo de Mapeo de Celdas |
| AMC _d | Algoritmo de Mapeo de Celdas con optimización en distancia |
| AMC _e | Algoritmo de Mapeo de Celdas con optimización en esfuerzo de control |
| AMC _t | Algoritmo de Mapeo de Celdas con optimización en tiempo |
| AMC _{de} | Algoritmo de Mapeo de Celdas con optimización en distancia y esfuerzo de control |
| AMC _{dt} | Algoritmo de Mapeo de Celdas con optimización en distancia y tiempo |
| AMC _{POM} | Algoritmo de Mapeo de Celdas para POM |
| CPA | Campos Potenciales Artificiales |
| DI | Doble Integrador |
| LQG | Linear Quadratic Gaussian Control |

Abreviatura Término

| | |
|-----|--|
| LQR | Linear Quadratic Regulator |
| MCS | Mapeo de Celdas Simple |
| POM | Problema de Optimización Multiobjetivo |
| POS | Problema de Optimización Simple |
| UAV | Unmanned Aerial Vehicle |

1. Introducción

La robótica es una rama de la ingeniería que tiene importancia en el desarrollo tecnológico de la sociedad, debido a la necesidad de construir dispositivos para realizar trabajos pesados y peligrosos, así como tareas en las que se requieren un estándar de calidad alto, difícil de lograr con procesos manuales. La implementación de robots en la industria aumenta la eficiencia, optimiza tiempos de producción y maximiza los estándares de calidad [1]. El concepto de robot comenzó a utilizarse en 1920 en la novela dramática *Rossum's Universal Robots*, en la cual surgió la palabra “*Robota*” que traduce servidumbre, hasta el día de hoy en donde se tienen robots sofisticados como Sophia de Hanson Robotics, el cual es diseñado para interactuar con seres humanos y aprender de su comportamiento.

Los robots autónomos son de interés dentro de la comunidad científica por varios motivos [2]. Primero, debido a la inteligencia que se debe dotar al sistema y su implementación. Es por ello que la robótica requiere la aplicación de varias disciplinas tales como: ingeniería mecánica, electrónica, sistemas, el control automático, entre otras. Segundo, por el grado de inteligencia que debe tener el sistema para recrear comportamientos humanos o interacciones con ellos. Tercero, por la cantidad de aplicaciones que se pueden realizar con los agentes robóticos, tales como: seguridad, exploración, búsqueda de minas, riego, limpieza, manipulación de materiales radioactivos, logística, despacho y distribución de paquetería, búsqueda y rescate, automatización industrial, transporte por medio de vehículos autónomos, entre otras.

Debido a la necesidad de que los sistemas robóticos sean autónomos, en la literatura se encuentran diferentes técnicas para generar algoritmos de planificación de trayectorias que pueden tener características acordes a la tarea a realizar. Los algoritmos de planificación son ampliamente estudiados en la comunidad científica y se construyen para generar rutas de referencia de diferente naturaleza y complejidad, en la que se eviten obstáculos de forma automática como se muestra en [3–5]. Los planificadores se pueden implementar para entornos dinámicos o estáticos, con el fin de encontrar una ruta entre dos puntos o cubrir todo el espacio libre. Estos se clasifican como globales o locales, dependiendo de la cantidad de información obtenida del espacio de trabajo.

Algunos trabajos de planificadores tienen en cuenta que los robots móviles pueden presentar restricciones cinemáticas o dinámicas [6–9]. Estas dependen del tipo de robot: marinos, aéreos, de desplazamiento sobre otros fluidos, de locomoción por medio de patas, de locomoción por bandas y vehículos con ruedas. Los móviles sobre ruedas pueden tener las siguientes configuraciones: Ackerman, triciclo clásico, tracción diferencial, síncrona, tracción omnidireccional, entre otras [2]. Los vehículos terrestres se pueden modelar en función de las coordenadas globales $p=(x,y,\phi)$ (posición y orientación) y las velocidades $\dot{p}=(\dot{x},\dot{y},\dot{\phi})$.

Las restricciones cinemáticas son aquellas que dependen del movimiento, se clasifican en holónomas y no holónomas. Las holónomas son las que no incluyen la velocidad. Las configuraciones síncrona y omnidireccional son algunos ejemplos de robots holónomos. Las restricciones no holónomas son las que tienen implícitamente la velocidad y no se obtienen por derivación total de una holónoma. Las configuraciones Ackerman, triciclo clásico, tracción diferencial son algunos ejemplos de vehículos que presentan restricciones cinemáticas del tipo no holónomas. En [10] se observa un algoritmo de optimización por colonia de hormigas (ACO) para encontrar trayectos que cumplan con las restricciones cinemáticas de tipo no holónomas y que sean óptimos en distancia. En [11] se muestra un planificador por campos potenciales para un humanoide con restricciones no holónomas, que construye caminos factibles de seguir por el robot.

Las restricciones dinámicas de robots móviles se presentan cuando se tienen limitaciones en velocidad y aceleración generadas por masas, fricciones y/o pares. Estas restricciones tienen incidencia en el tiempo de navegación y el gasto energético del móvil. En [12] se resuelve el problema de planeación de trayectorias para el movimiento de un robot sujeto a restricciones dinámicas, encontrando caminos óptimos en tiempo de navegación o seguridad, desde una posición inicial hasta una final evitando obstáculos y respetando las limitaciones. Conociendo los perfiles de velocidad y aceleración que presenta el sistema robótico, se puede generar rutas que no sobrepasen los valores máximos de estas variables y optimicen los parámetros deseados. El gasto energético en robótica esta directamente asociado al esfuerzo de control necesario para mantener al móvil en el trayecto de referencia.

Adicional a las restricciones del móvil se deben considerar las del entorno que aportan información para encontrar rutas de referencia evitando obstáculos. El espacio puede presentar restricciones de seguridad, tales como: distancia entre el móvil y los obstáculos, ruta menos escarpada, velocidad máxima. Estas condiciones ayudan al sistema a construir caminos de referencia sin riesgos de colisión por la distancia adecuada entre el sistema y los obstáculos o por la velocidad que lleva. También aporta información acerca del tipo de terreno en el que se esta desplazando, con el fin de que el móvil sea capaz de elegir una ruta que sea factible de seguir según su forma de locomoción.

Las trayectorias encontradas por los planificadores de trayectorias pueden ser de diferente naturaleza, aumentando o disminuyendo la complejidad del algoritmo. La mayoría de trabajos que aplican optimización en la construcción de rutas de referencia, lo hacen para tener caminos con menor distancia, suaves con respecto a la posición y seguros. Una forma de abordar el problema de planificación se basa en descomponer el espacio de trabajo en celdas o nodos y evaluar las distancias entre cada elemento del espacio, así como distinguir entre celdas ocupadas y libres. En este documento se pretende desarrollar un planificador de trayectorias que optimice no solo distancia de la ruta, sino otros parámetros mediante la minimización de una función de costo.

La descomposición del espacio de trabajo en celdas es ampliamente empleada debido a su buen desempeño. En [13–16] se muestran algunos métodos para la planificación que utilizan ésta como estrategia para representar el área de trabajo. Para encontrar rutas que optimicen un parámetro, es necesario definir una función de costo que contenga los criterios a minimizar o maximizar. En la literatura se encuentran algunos trabajos en donde el planificador entrega rutas de referencia con mínima distancia. En [17] se observa un algoritmo genético que optimiza la trayectoria en distancia, suavidad y seguridad (distancia entre la ruta de referencia y los obstáculos).

En la planificación de trayectorias siguen estando presentes algunas dificultades, tales como: la generalización de métodos en entornos conocidos y parcialmente conocidos o desconocidos, falta de adaptación y comportamiento no robusto de las técnicas convencionales, costo computacional alto [18, 19], rutas largas, generación de trayectorias sin tener en cuenta las restricciones cinemáticas y dinámicas del vehículo [20–23], pocos estudios para ambientes complejos [24]. En [25] se presenta un análisis comparativo de algunos métodos con inteligencia artificial y algoritmos convencionales para planificación de trayectorias.

Las trayectorias resultantes en los algoritmos de planificación, se emplean como referencia para el movimiento del robot. Por tanto, es necesario implementar una acción de control en la que se requieren sistemas de navegación. Estos sistemas, ayudan a localizar el robot a través del tiempo en coordenadas relativas o absolutas, proporcionando la información necesaria al controlador. Al final, el sistema de control está construido de forma que el robot sea capaz de seguir la trayectoria de referencia proporcionada por el planificador [26–33]. En [34–38] se observa el modelado y control de un péndulo invertido sobre robots móviles, mientras que en [31, 39–53] se observan algunos trabajos con diferentes técnicas de control empleadas para robots.

En este trabajo se utiliza un Algoritmo de Mapeo de Celdas (AMC) para obtener soluciones subóptimas. El AMC fue introducido por primera vez como una técnica para analizar el comportamiento global de los sistemas no lineales [54]. La técnica proporciona una discretización del espacio de estados, generando grafos cuyas conexiones representan las posibles trayectorias entre estados, y los vínculos están relacionados con el costo de cada transición entre celdas. Las mediciones comunes incluyen el tiempo de navegación, esfuerzo de control y distancia. En [55, 56], los autores muestran una aplicación del mapeo de celdas en problemas de optimización multiobjetivo. Un ejemplo del uso del mapeo de celdas en la planificación de trayectorias para los brazos robóticos se puede ver en [57]. En [58] se muestra una aplicación para realizar el control óptimo de la velocidad en el desplazamiento de una partícula en una dimensión y la orientación de un bote en dos dimensiones, por lo que esta aplicación puede extenderse a sistemas de n dimensiones.

En esta tesis se presenta el diseño de un planificador de trayectorias para robots móviles que se desplazan en el plano, y que su modelo dinámico es equivalente al del DI luego de ser linealizado por medio de métodos analíticos o aprendizaje de máquina. El algoritmo propuesto se fundamenta sobre el método de Mapeo de Celdas, agregando las restricciones cinemáticas y dinámicas del sistema, así como las limitaciones del entorno. Estas características, hacen que el planificador implementado genere rutas optimizadas en diferentes parámetros, tales como: tiempo de navegación, distancia, esfuerzo de control y energía. También se muestra la aplicación del AMC para la solución de problemas de optimización de forma numérica, en los que no hay una solución analítica o su desarrollo es difícil de efectuar. A lo largo del documento, se muestran algunos problemas de optimización continua, así como sus equivalentes en optimización combinatoria para el movimiento del DI en 1D. Luego se plantea el problema de planificación de trayectorias para el DI sobre el plano. Por último, se presenta el modelado matemático de robots móviles de guiado diferencial para ver su equivalencia con el DI en 2D y encontrar las dinámicas seguidas por las ruedas y la plataforma, para posteriormente emplearlas en la generación de rutas optimizadas.

1.1. Estructura del documento

Este documento se desarrolla en tres partes, cada una con dos capítulos. La primera parte trata el problema de optimización del doble integrador (móvil más sencillo) en tiempo continuo y discreto, la segunda parte habla de la planificación de trayectorias y la aplicación en tiempo discreto para el movimiento del DI en el plano y la tercera parte describe el modelado matemático de robots móviles de guiado diferencial que se desplazan en el plano y el problema de planificación de robots. En el **Capítulo 2** se muestra el desarrollo matemático y la solución de algunos problemas de optimización continua para el movimiento del DI en 1D. En el **Capítulo 3** se observa la implementación del AMC para la solución del problema de optimización de forma discreta del movimiento del DI en 1D, así como la comparación del entre la forma continua y discreta de algunos casos. En el **Capítulo 4** se mencionan e implementan algunos algoritmos de planificación de trayectorias existentes en la literatura, así como el AMC para minimizar distancia y se comparan los resultados obtenidos por el AMC con otros métodos. En el **Capítulo 5** se desarrolla el AMC para la planificación de trayectorias óptimas del DI en el plano. En el **Capítulo 6** se describe el modelo dinámico de un robot móvil de guiado diferencial, se muestra el proceso de linealización por realimentación, se observa como el movimiento del punto de control P_r es análogo a la dinámica del DI y se identifican las funciones matemáticas que rigen el movimiento de las ruedas del móvil. En el **Capítulo 7** se emplea el AMC para la generación de trayectorias, perfiles de velocidad y dinámica de las ruedas de robots móviles que se desplazan en el plano, posteriormente se comparan las distancias de las rutas generadas por el AMC con las de dos algoritmos clásicos. También se muestra la aplicación para minimizar el gasto energético y la implementación del planificador en robots móviles omnidireccionales de 3 ruedas. Por último, en el **Capítulo 8** se muestran las conclusiones, recomendaciones y trabajo futuro de este trabajo de investigación.

Parte I.

**Optimización continua y discreta del DI en
1D**

2. Control óptimo para el DI

En este capítulo se desarrollan analíticamente algunos problemas de optimización continua para el movimiento del DI en 1D. Para esto, se presenta el modelo matemático del DI, que es ampliamente estudiado en la literatura y se utiliza para analizar el movimiento de una partícula en 1D. En algunos casos se puede asumir que el movimiento de un vehículo, robot u otro objeto sobre una superficie plana tiene un modelo simplificado idéntico al del doble integrador, por lo que con este sistema simplificado se pueden estudiar múltiples problemas. En [59] se muestra la aplicación del DI en problemas de optimización continua.

El DI es mostrado en la Figura 2-1 y representado por la ecuación diferencial Ec. (2-1).

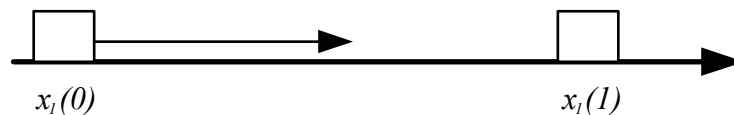


Figura 2-1.: Partícula en 1D.

$$\ddot{x} = u \quad (2-1)$$

En el espacio de estados el DI es representado por la Ec. (2-2), donde x_1 es la posición, x_2 la velocidad y u la aceleración de la partícula.

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u \end{cases} \quad (2-2)$$

2.1. Problema de optimización simple (POS)

El problema de optimización puede entenderse como el proceso de encontrar la mejor alternativa entre un conjunto de soluciones. Este conjunto se puede formar con las posibles soluciones de la evolución del proceso y de acuerdo con la minimización de un criterio de desempeño se escoge la “mejor”. Por tanto, se puede formalizar matemáticamente, si se consigue definir una función objetivo J que esté asociada al desempeño del sistema dinámico [60].

En ingeniería, la teoría de control óptimo ha sido popular desde la década de los 50, época en la que se llevó la carrera espacial y en la que era de gran importancia llevar un vehículo espacial desde la tierra hasta un punto en el espacio en el menor tiempo y consumiendo la menor cantidad de combustible posible. Actualmente, la creciente demanda de bienes y servicios ha llevado a los ingenieros a buscar métodos óptimos para la toma de decisiones, con el fin de disminuir costos y esfuerzos, y maximizar recursos [61].

La realimentación de estados es de gran importancia dentro de la teoría de control moderna. Para determinar la ley de control óptima se emplea el Principio del Mínimo de Pontryagin, el cual minimiza la función Hamiltoniana sujeta a la evolución del sistema. Más adelante en este capítulo se muestra el procedimiento para la aplicación de este principio. Este se puede emplear en sistemas continuos cuando se conocen las condiciones de frontera y se cuenta con una función objetivo a minimizar.

El control óptimo es una extensión del cálculo variacional y se enfoca en determinar las señales de control que hacen que un proceso satisfaga las restricciones físicas minimizando una función de costo. Las restricciones físicas son definidas para cada sistema, así como el criterio de desempeño a optimizar como se menciona en [62]. En el POS se considera una función de costo con solo un parámetro a minimizar.

También existen diversos trabajos para desarrollar técnicas de ajuste de los parámetros de control óptimo. El LQG es empleado en sistemas lineales con información de estado incompleta y en sistemas no lineales perturbados. El LQR es empleado para minimizar una función de costo cuadrática y es una forma automatizada de encontrar el controlador adecuado. En [63] se muestra la implementación de un controlador LQG y de un LQR para el control del vuelo de un UAV. En [64] se muestra la aplicación de LQR en sistemas con restricciones.

A continuación se formula el problema de optimización de forma generalizada.

Sea el sistema descrito por la Ec. (2-3)

$$\dot{x} = f(x, u, t), \quad (2-3)$$

donde, x es la variable de estado, u la acción de control y t el tiempo. El problema consiste en encontrar la trayectoria desde $x(t_0)$ hasta $x(t_1)$, donde es necesario determinar la señal de control $u(t)$ que debe aplicarse en el intervalo de tiempo $[t_0, t_1]$ para que su evolución minimice el funcional dado en la Ec. (2-4).

$$J = \int_{t_0}^{t_1} L(x, u, t) dt, \quad (2-4)$$

donde J es la función de costo, L es el Lagrangiano del sistema y $u(t) \in U$. U es el conjunto de señales de control que se pueden aplicar al sistema descrito por (2-3) y existe una $\tilde{u}(t)$ tal que:

$$J(\tilde{u}(t)) \leq J(u(t)) \quad \forall u(t) \in U \quad (2-5)$$

donde, $\tilde{u}(t)$ es la señal de control óptima en el intervalo de tiempo $[t_0, t_1]$. Para determinar la señal $\tilde{u}(t)$ se puede emplear el principio del mínimo de Pontryagin o la ecuación de Hamilton-Jacobi-Bellman. En este documento se presenta el desarrollo matemático para encontrar la señal de control mínima por medio del mínimo de Pontryagin, que usa el método de los multiplicadores de Lagrange reformulado como se muestra en [65] para minimizar la función Hamiltoniana dada en (2-6).

$$H(x, \lambda, u, t) = L(x, u, t) + \lambda f(x, u, t) \quad (2-6)$$

donde, λ son los multiplicadores de Lagrange y H la función Hamiltoniana. El problema de optimización se puede plantear de la forma mostrada en la Ec. (2-7).

$$\left\{ \begin{array}{l} \text{mín } J \\ \text{sujeto a: } \dot{x} = f(x, u, t), \quad x(t_0) = x_0, \quad x(t_1) = x_1 \end{array} \right. \quad (2-7)$$

Para solucionar el problema se resuelve primero la Ec. (2-8), con el fin de obtener la señal de control $\tilde{u}(t) = \tilde{u}(x, \lambda, t)$.

$$\frac{\partial H(x, \lambda, u, t)}{\partial u} = 0, \quad (2-8)$$

Ahora se encuentra la función $H(x, \lambda, \tilde{u}, t)$ minimizada y se procede a resolver el conjunto de ecuaciones diferenciales (2-9) sujetas a las condiciones de contorno dadas en $x(t_0)$ y $x(t_1)$.

$$\left\{ \begin{array}{l} \dot{\lambda} = -\frac{\partial H(x, \lambda, \tilde{u}, t)}{\partial x} \\ \dot{x} = \frac{\partial H(x, \lambda, \tilde{u}, t)}{\partial \lambda} \end{array} \right. , \quad (2-9)$$

por último, se deben sustituir los valores de \tilde{x} y $\tilde{\lambda}$ hallados en la Ec. (2-9) en la señal de control $\tilde{u}(t) = \tilde{u}(\tilde{x}, \tilde{\lambda}, t)$ para obtener el vector de control que optimiza el sistema.

2.2. POS para el DI

En el movimiento de una partícula como la mostrada en la Figura 2-1 y que es representada por el modelo matemático del DI se pueden presentar diferentes problemas de optimización, los cuales se analizan a continuación:

El modelo matemático del DI de forma vectorial es representado como se muestra en la Ec. (2-10).

$$\dot{X} = f(X, u, t) = \begin{bmatrix} x_2 \\ u \end{bmatrix} \quad (2-10)$$

Las condiciones iniciales y finales de la partícula son mostradas en las Ec. (2-11) y (2-12).

$$X(t_0) = X_0 = \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} \quad (2-11)$$

$$X(t_1) = X_1 = \begin{bmatrix} x_1(1) \\ x_2(1) \end{bmatrix} \quad (2-12)$$

El vector de estado se muestra en la Ec. (2-13) y en la Ec. (2-14) se observa el vector de los Multiplicadores de Lagrange del problema de optimización.

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (2-13)$$

$$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} \quad (2-14)$$

La definición general de la función Hamiltoniana para un problema vectorial se muestra en la Ec. (2-15).

$$H(X, \lambda, u, t) = L(X, u, t) + \lambda^T f(X, u, t) \quad (2-15)$$

A continuación se muestran los problemas de optimización encontrados en sistemas continuos y su aplicación para el modelo matemático del DI:

2.2.1. Mínimo esfuerzo de control

El problema de mínimo esfuerzo de control está relacionado con la capacidad o restricciones de los actuadores del sistema. Por tal motivo, surge la necesidad de generar señales de control que tengan la menor energía posible. La función de costo empleada en la literatura para solucionar el problema de minimización del esfuerzo de control se muestra en la Ec. (2-16).

$$J = \frac{1}{2} \int_{t_0}^{t_1} u^2(t) dt \quad (2-16)$$

El problema de optimización de mínimo esfuerzo de control se puede plantear de la forma mostrada en la Ec. (2-17).

$$\begin{cases} \text{mín} & \frac{1}{2} \int_{t_0}^{t_1} u^2(t) dt \\ \text{sujeto a:} & \dot{X} = f(X, u, t), \quad X(t_0) = X_0, \quad X(t_1) = X_1 \end{cases} \quad (2-17)$$

En la Ec. (2-18) se define la función Hamiltoniana para el problema de minimización del esfuerzo de control del DI.

$$H(X, \lambda, u, t) = \frac{1}{2} u^2 + \lambda_1 x_2 + \lambda_2 u \quad (2-18)$$

Al derivar la función Hamiltoniana con respecto a λ y X , se obtiene respectivamente la Ec. (2-19) y Ec. (2-20).

$$\frac{\partial H(X, \lambda, u, t)}{\partial \lambda} = \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u \end{cases} \quad (2-19)$$

$$\frac{\partial H(X, \lambda, u, t)}{\partial X} = \begin{cases} \dot{\lambda}_1 = 0 \\ \dot{\lambda}_2 = -\lambda_1 \end{cases} \quad (2-20)$$

Al derivar el Hamiltoniano con respecto a u como en la Ec. (2-21), se obtiene la señal óptima de u que se llama \tilde{u} y se muestra en la Ec. (2-22).

$$\frac{\partial H(X, \lambda, u, t)}{\partial u} = 0 \quad (2-21)$$

$$\tilde{u} = -\lambda_2 \quad (2-22)$$

La forma de λ_1 y λ_2 según la Ec. (2-20) es la que se observa en la Ec. (2-23).

$$\begin{cases} \lambda_1 = c_1 \\ \lambda_2 = c_2 t + c_3 \end{cases}, \quad (2-23)$$

Por tanto, la señal de control óptima será $\tilde{u} = \tilde{c}_2 t + \tilde{c}_3$, donde \tilde{c}_2 y \tilde{c}_3 son los negativos de c_2 y c_3 respectivamente.

Sustituyendo \tilde{u} en el sistema de la Ec. (2-2) se obtiene la Ec. (2-24).

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \tilde{c}_2 t + \tilde{c}_3 \end{aligned} \quad (2-24)$$

La solución del sistema de la Ec. (2-24) se muestra en la Ec. (2-25).

$$\begin{aligned} x_1(t) &= \frac{1}{6} \tilde{c}_2 t^3 + \frac{1}{2} \tilde{c}_3 t^2 + x_2(0)t + x_1(0) \\ x_2(t) &= \frac{1}{2} \tilde{c}_2 t^2 + \tilde{c}_3 t + x_2(0) \end{aligned} \quad (2-25)$$

Para encontrar las señales óptimas en esfuerzo de control es necesario tomar las condiciones de frontera dadas $X_0 = [x_1(0), x_2(0)]^T$ y $X_1 = [x_1(1), x_2(1)]^T$, luego definir el tiempo necesario para llevar la partícula desde X_0 a X_1 . Para el ejemplo, se toma $t \in [0, 1]$ y se encuentran los valores de las constantes \tilde{c}_2 y \tilde{c}_3 que satisfacen las condiciones de contorno.

$$\begin{aligned} \tilde{c}_2 &= -12(x_1(1) - x_1(0)) + 6(x_2(1) + x_2(0)) \\ \tilde{c}_3 &= 6(x_1(1) - x_1(0)) - 2(x_2(1) + 2x_2(0)) \end{aligned}$$

En la Figura 2-2 se observa la posición, velocidad y señal de control con mínima energía en el tiempo, que lleva la partícula desde una condición inicial X_0 hasta una final X_1 en un $t \in [0, 1]$. Para el ejemplo se toma $X_0 = [-10, -4]^T$ y $X_1 = [0, 0]^T$.

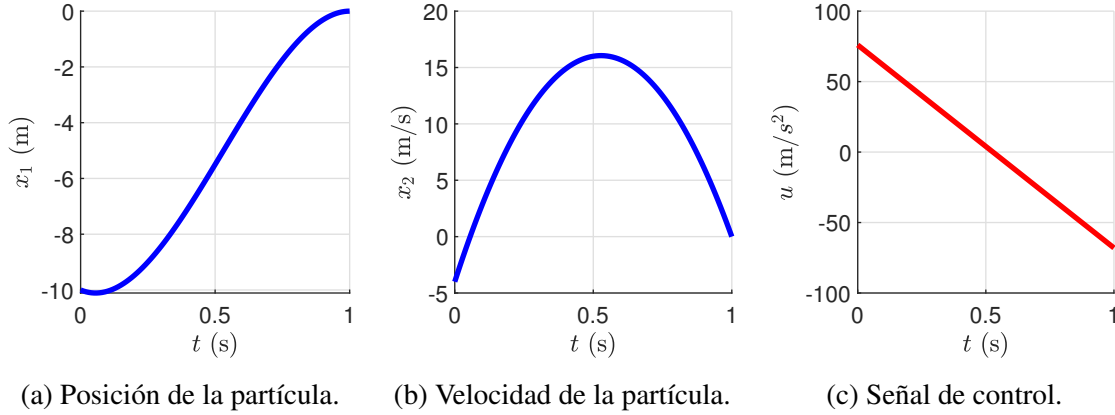


Figura 2-2.: Comportamiento de la partícula en el tiempo para minimizar esfuerzo de control.

2.2.2. Minimización de la variación total

La variación total de una señal en \mathbb{R}^2 puede describirse como la medida acumulada del cambio vertical de la función. En ingeniería, una variación total pequeña de la señal de control es deseable para facilitar el diseño e implementación del sistema. Si esta variación es baja, se requieren en el proceso elementos de baja capacidad, que tendrán menor consumo de energía y tamaño. En [66] se observa una discretización del problema de optimización para evaluar la variación total de la señal de control.

En la Ec. (2-26) se muestra el cálculo de la variación total de la función $u : [t_0, t_1] \rightarrow \mathbb{R}$.

$$VT(u) = \int_{t_0}^{t_1} |\dot{u}(t)| dt, \quad (2-26)$$

se asume u como una función continua en el intervalo $[t_0, t_1]$. Se define una nueva variable de control $v(t) = \dot{u}(t)$ para $t \in [t_0, t_1]$ y se obtiene la función de costo empleada para minimizar la variación total de la señal mostrada en la Ec. (2-27).

$$J = \int_{t_0}^{t_1} |v(t)| dt, \quad (2-27)$$

El problema de minimización de la variación total de la señal de control se puede plantear de la forma mostrada en la Ec. (2-28).

$$\begin{cases} \text{mín} & \int_{t_0}^{t_1} |v(t)| dt, \\ \text{sujeto a:} & \dot{X} = f(X, u, t), \quad \dot{u}(t) = v(t), \quad X(t_0) = X_0, \quad X(t_1) = X_1 \end{cases} \quad (2-28)$$

En la Ec. (2-29) se define la función Hamiltoniana para el problema de minimización de la función de costo dada en la Ec. (2-27) para el DI.

$$H(X, u, \lambda, \eta, v, t) = |v| + \lambda_1 x_2 + \lambda_2 u + \eta v, \quad (2-29)$$

donde $\lambda \in \mathbb{R}^2$ y $\eta \in \mathbb{R}$ son variables adjuntas definidas en la función Hamiltoniana.

En [67] se muestra el procedimiento para hallar las señales necesarias para minimizar la variación total de la señal que lleva la partícula desde una condición inicial a una final en $t = 1$ s. Las expresiones se muestran en la Ec. (2-30).

$$u(t) = \begin{cases} 4(x_1(1) - x_1(0)) - x_2(1) - 3x_2(0), & \text{Si } 0 \leq t < 0.5 \\ 3x_2(1) + x_2(0) - 4(x_1(1) - x_1(0)), & \text{Si } 0.5 \leq t \leq 1 \end{cases} \quad (2-30)$$

Para el ejemplo, se aplica una señal positiva de control durante la mitad del tiempo total y luego una negativa con la misma duración, en este caso 0.5 s. Las señales de control que cumplen la condición de tiempo anterior serán las que tengan mínima variación total, cualquier otra combinación que solucione el problema tendrá una variación total mayor. Si se toma $X_0 = [-10, -4]^T$ y $X_1 = [0, 0]^T$ reemplazando en la Ec. (2-30) da como resultado lo mostrado en la Ec. (2-31).

$$u(t) = \begin{cases} 52, & \text{Si } 0 \leq t < 0.5 \\ -44, & \text{Si } 0.5 \leq t \leq 1 \end{cases} \quad (2-31)$$

La variación total de la señal de control se encuentra con la Ec. (2-32).

$$VT(u) = 4|2(x_1(1) - x_1(0)) - x_2(1) - x_2(0)| = 96 \quad (2-32)$$

En la Figura 2-3a se muestra el comportamiento de las variables de estado para el problema de optimización de variación total de la señal de control. Con estas dos señales de control se llega al estado final en un tiempo de 1 s. En la Figuras 2-3b, 2-3c y 2-3d, se observan la posición, velocidad y señal de control del DI con optimización de variación total. Para poder solucionar el problema de optimización en un tiempo pequeño es necesario tener aceleraciones muy grandes que en la práctica no son realizables. Para este ejemplo no se tuvo en cuenta la saturación de la señal de control, con el fin de observar la solución con restricción de tiempo. Otra forma de solucionar el problema de optimización de variación total es teniendo señales constantes fijas y dejando libre el tiempo, caso que es estudiado en el problema de mínimo tiempo de navegación.

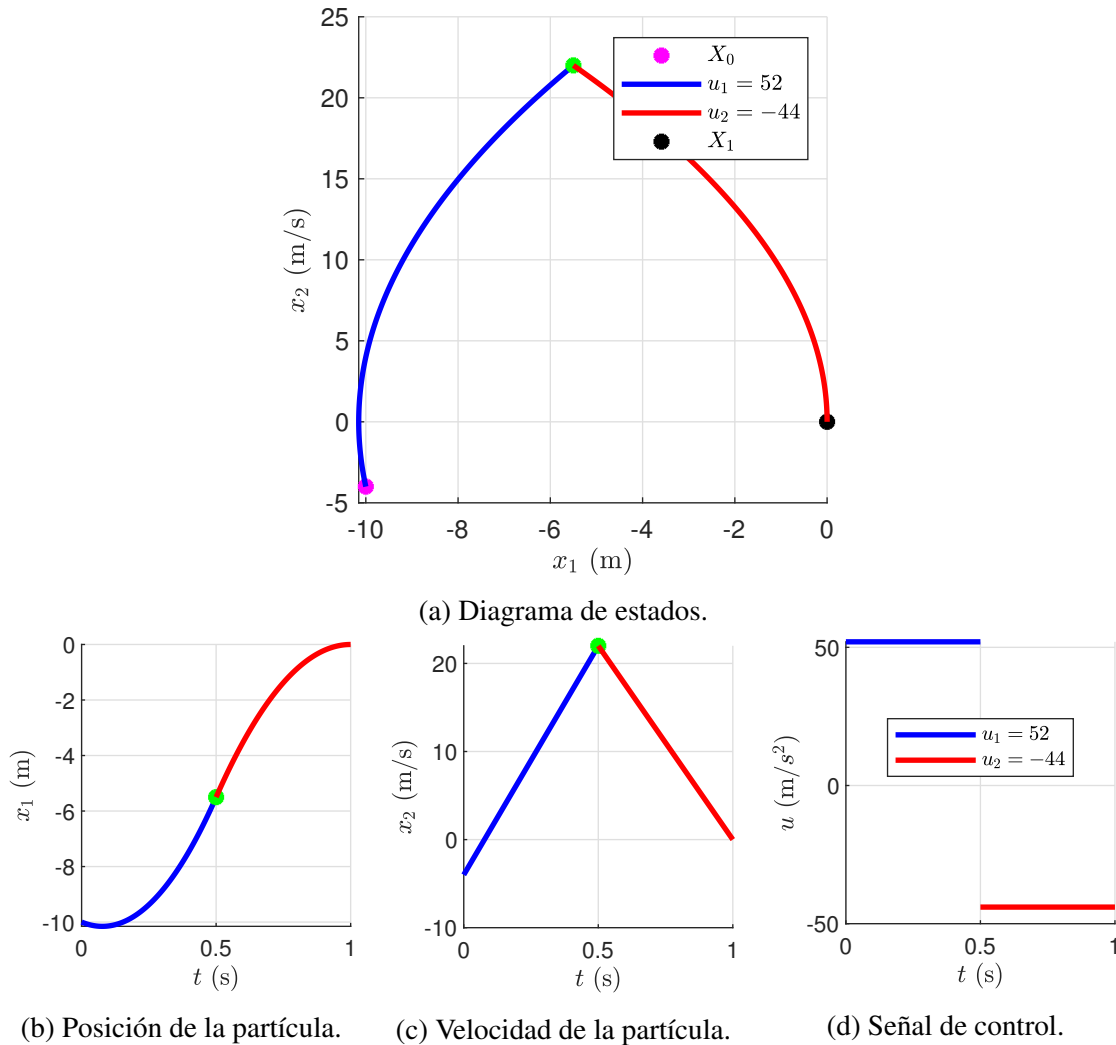


Figura 2-3.: Diagrama de estados y comportamiento de la partícula en el tiempo para minimizar la variación total.

2.2.3. Mínimo tiempo de navegación

Para solucionar el problema de minimización del tiempo de desplazamiento de una partícula se puede emplear una ley de control con $n - 1$ conmutaciones como se menciona en [68], donde n es el grado del sistema. Por tanto se toma $u \in \{-1, 1\}$ para el ejemplo.

La función de costo empleada en la literatura para solucionar el problema de minimización de tiempo es la que se muestra en la Ec. (2-33), por lo que el Lagrangiano del problema es $L = 1$.

$$J = \int_{t_0}^{t_1} 1 dt \quad (2-33)$$

En la Ec. (2-34) se muestra el planteamiento del problema de optimización de mínimo tiempo.

$$\begin{cases} \text{mín} & \int_{t_0}^{t_1} 1 dt \\ \text{sujeto a:} & \dot{X} = f(X, u, t), \quad X(t_0) = X_0, \quad X(t_1) = X_1 \end{cases} \quad (2-34)$$

En la Ec. (2-35) se define la función Hamiltoniana para el problema de minimización de tiempo de navegación del DI.

$$H(X, \lambda, u, t) = 1 + \lambda_1 x_2 + \lambda_2 u \quad (2-35)$$

Si se deriva la función Hamiltoniana con respecto a λ , se obtiene el sistema de nuevo, como se observa en la Ec. (2-36).

$$\frac{\partial H(X, \lambda, u, t)}{\partial \lambda} = \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u \end{cases} \quad (2-36)$$

Al derivar la función Hamiltoniana con respecto al vector X se obtiene la Ec. (2-37).

$$\frac{\partial H(X, \lambda, u, t)}{\partial X} = \begin{cases} \dot{\lambda}_1 = 0 \\ \dot{\lambda}_2 = -\lambda_1 \end{cases} \quad (2-37)$$

Si \tilde{u} es la señal de control óptima en tiempo, entonces el sistema tiene una solución no trivial $(\lambda(t), x(t))$ y cumple la condición de la Ec. (2-38).

$$H(X, \lambda, \tilde{u}, t) = \text{mín}_{|u| \leq 1} H(X, \lambda, u, t) \quad (2-38)$$

De la Ec. (2-35), si $\lambda_2 \neq 0$, el valor de u que cumple la condición de Ec. (2-38) es $\tilde{u} = \text{sgn}(\lambda_2)$. Por tanto, la función Hamiltoniana que cumple el criterio de minimización se observa en la Ec. (2-39).

$$\text{mín}_{|u| \leq 1} H(X, \lambda, u, t) = 1 + \lambda_1 x_2 + |\lambda_2| \quad (2-39)$$

La forma de λ_1 y λ_2 según la Ec. (2-37) es la que se observa en la Ec. (2-40).

$$\begin{cases} \lambda_1 = c_1 \\ \lambda_2 = c_2 t + c_3 \end{cases}, \quad (2-40)$$

donde c_1 , c_2 y c_3 son constantes. La señal de control óptima será $\tilde{u} = \text{sgn}(c_2 t + c_3)$ y se concluye que \tilde{u} es constante y toma valores de $\tilde{u} = \pm 1$.

El sistema resultante con las señales de control $u = \pm 1$ se observa en la Ec. (2-41).

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \pm 1 \end{cases} \quad (2-41)$$

Las trayectorias dadas por la evolución de la Ec. (2-41) satisfacen el sistema de la Ec. (2-42) que al integrar da como resultado la Ec. (2-43).

$$\frac{dx_1}{dx_2} = \pm x_2 \quad (2-42)$$

$$x_1 = \pm \frac{x_2^2}{2} + C \quad (2-43)$$

En la Figura 2-4a se observan las curvas de la evolución del sistema dado por la Ec. (2-41) ante diferentes condiciones iniciales de posición y velocidad. En la Figura 2-4b se muestra la superficie de conmutación, la cual indica el momento en que debe cambiar la señal de control para llegar desde una condición inicial X_0 hasta una final X_1 en el menor tiempo posible. En la Figura 2-4b se toman dos condiciones iniciales $X_0 = [-10, -4]^T$ y $X_0 = [10, 4]^T$ con condición final $X_1 = [0, 0]^T$.

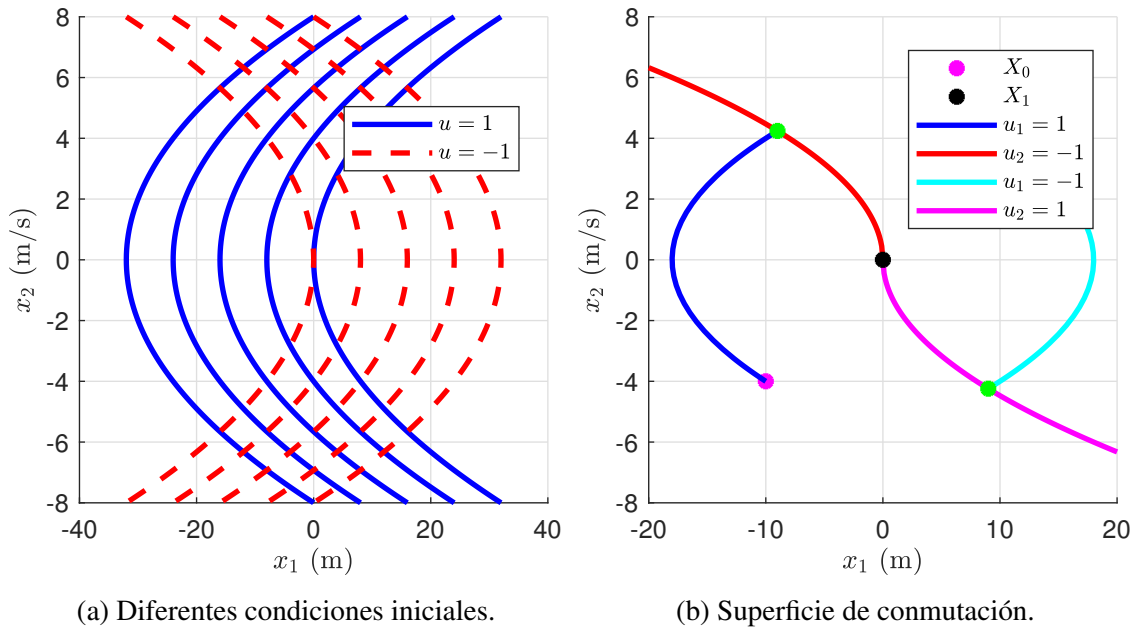


Figura 2-4.: Diagrama de estado y superficie de conmutación para trayectoria óptima en tiempo.

En la Figura 2-5a se observa la posición, en la 2-5b la velocidad y en la 2-5c la señal de control del DI con optimización de tiempo ante una entrada de control $u = \pm 1$ que lleva la partícula desde una condición inicial X_0 hasta una final X_1 en un $t = 12.4853$ s. Para el ejemplo se toma $X_0 = [-10, -4]^T$ y $X_1 = [0, 0]^T$.

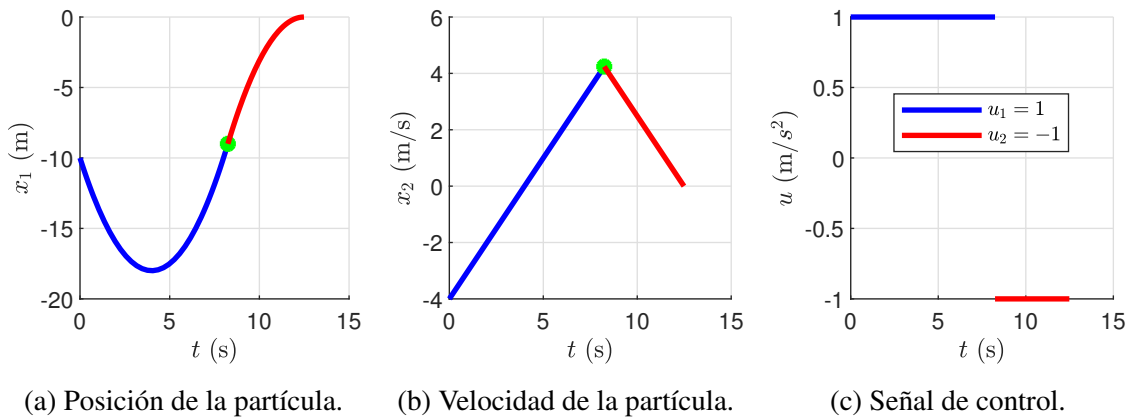


Figura 2-5.: Comportamiento de la partícula en el tiempo para trayectoria óptima en tiempo.

La posición y el perfil de velocidad del DI son generados a partir de la evolución del sistema ante las señales de control obtenidas para solucionar el problema de llevar la partícula desde una condición inicial hasta una final en el menor tiempo posible, tomando únicamente las señales de control $u = \pm 1$. La solución encontrada además de ser la trayectoria con mínimo tiempo de navegación es la que tiene menor variación total de la señal de control.

2.2.4. Problema de Fuller

En la teoría de control, el término de chattering es empleado para referirse a oscilaciones de frecuencia y amplitud finita en un intervalo de tiempo que aparecen debido a la rápida conmutación del controlador. Fuller en [69] describe el problema inicial de optimización de una variable de estado elevada a una potencia positiva. En 2-44 se muestra la función de costo estudiada en [70–72] para el problema de encontrar una solución al problema de optimización en tiempo finito con un número infinito de conmutaciones.

$$J = \int_{t_0}^{t_1} |x_1(t)|^v dt, \quad (2-44)$$

donde, v es un valor positiva. Si $v = 0$ se tiene el problema de optimización del tiempo de navegación. Si $v = 1$ se optimiza la variación total de la señal de control. Si $v = 2$ se describe el problema de Fuller. En la Ec. (2-45) se muestra el planteamiento del problema de optimización para el problema de Fuller.

$$\left\{ \begin{array}{l} \text{mín} \int_{t_0}^{t_1} |x_1(t)|^2 dt \\ \text{sujeto a: } \dot{X} = f(X, u, t), \quad X(t_0) = X_0, \quad X(t_1) = X_1 \end{array} \right. \quad (2-45)$$

En [73] se muestra el procedimiento para encontrar la señal de control (2-46) que minimiza la función de costo que se presenta en (2-44). En esta se introduce g , que para el problema de Fuller es igual a 0.444623560. La constante g depende del valor de v .

$$u = -\text{sgn}(x_1 + gx_2 |x_2|) \quad (2-46)$$

En la Ec. (2-46) se puede observar que la señal de control toma los valores de $u \in [-1, 1]$. Al aplicar u al DI con las condiciones iniciales $X_0 = [-10, -4]^T$ y finales $X_1 = [0, 0]^T$ se obtiene la posición de la partícula en la Figura 2-6a. En la Figura 2-6b se muestra la velocidad en el tiempo. La porción de gráfica verde es la obtenida con la primera señal de control, en este caso $u_1 = 1$. Luego la señal de control conmuta a $u_2 = -1$ fracción azul, posteriormente la acción de control cambia de nuevo generando la gráfica roja y por último vuelve a conmutar para generar la sección de color negro.

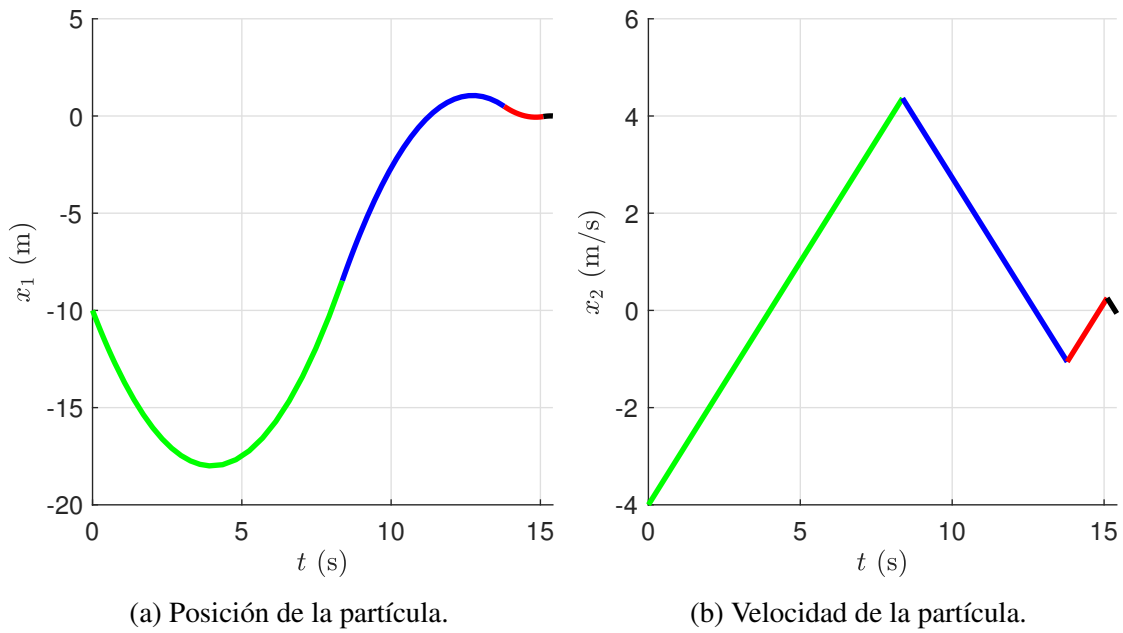


Figura 2-6.: Comportamiento de la partícula en el tiempo para el problema de Fuller.

En la Figura 2-7 se observa el diagrama de estados de la partícula para el problema de Fuller cortando la simulación al llegar al cuarto evento, donde debería haber otra conmutación de la señal de control, pero al ser un valor cercano al objetivo se termina la ejecución.

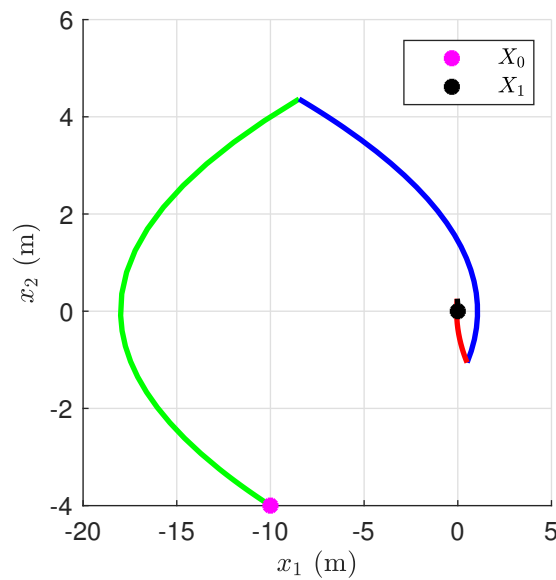


Figura 2-7.: Espacio de estados de la partícula para el problema de Fuller.

2.2.5. Minimización del gasto energético

El gasto energético de sistemas en movimiento es determinado por diferentes factores. En este apartado solo se considera la energía cinética de una partícula $E_k = \frac{1}{2}mv^2$, por lo que la función de costo para el problema de optimización queda como se muestra en la Ec. (2-47). Como se puede apreciar el problema planteado es análogo al de Fuller pero en este caso se minimiza el área bajo la curva de la velocidad al cuadrado.

$$J = \frac{1}{2} \int_{t_0}^{t_1} mx_2(t)^2 dt \quad (2-47)$$

El problema de optimización para la minimización del gasto energético se puede plantear de la forma mostrada en la Ec. (2-48). En el siguiente capítulo se soluciona este problema de forma numérica debido a la complejidad matemática para encontrar la señal de control en tiempo continuo con el teorema del mínimo de Pontryagin.

$$\left\{ \begin{array}{l} \text{mín} \quad \frac{1}{2} \int_{t_0}^{t_1} mx_2(t)^2 dt \\ \text{sujeto a: } \dot{X} = f(X, u, t), \quad X(t_0) = X_0, \quad X(t_1) = X_1 \end{array} \right. \quad (2-48)$$

2.3. Problema de optimización multiobjetivo (POM)

En muchas aplicaciones es necesaria la optimización de varios objetivos simultáneamente. En la literatura estas aplicaciones son llamadas problemas de optimización multiobjetivo (POM). Como ejemplo se puede considerar el problema de navegación de un robot de búsqueda de minas, en el que se deben optimizar los parámetros de gasto energético y mayor cobertura de terreno. Estos objetivos entran en conflicto, puesto que la mayor cobertura posible no es la que tiene menor gasto energético y para tener menor gasto energético no se cubriría gran cantidad del terreno. Por lo que se hace necesario optimizar ambos objetivos de una manera adecuada.

En el POM se encuentra un conjunto de soluciones óptimas, llamado conjunto de Pareto. Este conjunto suele formar un objeto de $(k - 1)$ dimensiones, donde k es el número de objetivos implicados en el problema. A diferencia del problema de optimización simple (POS) donde se consideran un sólo objetivo y en los que se espera una solución óptima, ahora se considera la existencia de un conjunto de soluciones que optimizan los diferentes objetivos dándoles prioridad según el peso expresado en la función de costo. En [74–80] se observan algunos trabajos en los que se aplica la optimización multiobjetivo.

En la literatura se han desarrollado una gran cantidad de trabajos para obtener el conjunto de Pareto, entre ellos están: optimización inspirada en algoritmos genéticos [81], algoritmo de colonia de hormigas [82], algoritmo inmunológico [83], estrategia de búsqueda de soluciones convergentes por medio de la simulación de fuerzas de repulsión entre partículas magnéticas [84] y optimización de enjambres de partículas [85]. Todos fueron aplicados a los diseños de control de realimentación de estados. Fliege y Svaiter propusieron algunos algoritmos basados en el gradiente para convertir el problema de optimización multiobjetivo en optimización simple [86]. Bosman [87] expande el concepto introduciendo transformaciones geométricas y combinándolo con el algoritmo genético [88].

El POM se puede definir como en la Ec. (2-49)

$$\min_{\mathbb{X} \in \mathbb{R}^n} F(\mathbb{X}), \quad (2-49)$$

$$\begin{aligned} \text{con } g_i(\mathbb{X}) &\leq 0, i = 1, \dots, l, \\ h_i(\mathbb{X}) &= 0, i = 1, \dots, m, \end{aligned}$$

donde, F es el conjunto que contiene las funciones objetivo $f_i : Q \rightarrow \mathbb{R}$ a analizar, tal que:

$$F : Q \rightarrow \mathbb{R}^k, F(\mathbb{X}) = \{f_1(\mathbb{X}), \dots, f_k(\mathbb{X})\} \quad (2-50)$$

El conjunto $Q \subset \mathbb{R}^n$ es el dominio de J o el conjunto realizable y es un subconjunto del espacio de decisión \mathbb{R}^n . \mathbb{X} es el vector de decisión y Q esta implícitamente definido por las restricciones del problema.

$$Q = \{\mathbb{X} \in \mathbb{R}^n \mid g_i(\mathbb{X}) \leq 0, i = 1, \dots, l, \text{ y } h_j(\mathbb{X}) = 0, j = 1, \dots, m\}, \quad (2-51)$$

donde, g_i y h_i representan las restricciones de desigualdad y de igualdad respectivamente. Si $Q = \mathbb{R}^n$ (i.e., $l = m = 0$), el POM es un problema sin restricciones. En muchos casos Q es definido por un cajón de restricciones, es decir, el conjunto de valores posibles es igual a una caja n-dimensional.

$$Q = B_{l,u} = \{\mathbb{X} \in \mathbb{R}^n : l_i \leq x_i \leq u_i, i = 1, \dots, n\}, \quad (2-52)$$

donde, $l_i \leq u_i, i = 1, \dots, n$, son los límites inferior y superior, respectivamente.

En el POM no se puede considerar una solución mejor a otra. Para definir las soluciones de un POM es necesario utilizar el concepto de dominancia que se explica en [89].

Definición de dominancia

1. Sea $\mathbf{v}, \mathbf{w} \in \mathbb{R}^k$. El vector \mathbf{v} es menor que \mathbf{w} (forma corta: $\mathbf{v} <_p \mathbf{w}$), si $v_i < w_i \forall i \in \{1, \dots, k\}$. La relación \leq_p se define igual.
2. Un vector $\mathbb{Y} \in \mathbb{Q}$ es dominado por un vector $\mathbb{X} \in \mathbb{Q}$ ($\mathbb{X} < \mathbb{Y}$) con respecto al POM si:

$$F(\mathbb{X}) \leq_p F(\mathbb{Y}) \quad \text{y} \quad F(\mathbb{X}) \neq F(\mathbb{Y})$$

de lo contrario, \mathbb{Y} no es dominado por \mathbb{X} .

3. Un vector $\mathbb{Y} \in \mathbb{Q}$ es dominado débilmente por un vector $\mathbb{X} \in \mathbb{Q}$ ($\mathbb{X} \leq \mathbb{Y}$) con respecto al POM si:

$$F(\mathbb{X}) \leq_p F(\mathbb{Y})$$

Si \mathbb{X} domina a \mathbb{Y} , se puede considerar que \mathbb{X} es un conjunto “mejor” según el POM.

Definición de conjunto de Pareto

1. Un punto $\mathbb{X} \in \mathbb{Q}$ es llamado óptimo de Pareto del POM si no existe $\mathbb{Y} \in \mathbb{Q}$ que domine a \mathbb{X} .
2. Un punto $\mathbb{X} \in \mathbb{Q}$ es llamado óptimo local de Pareto del POM si existe una vecindad N de \mathbb{X} tal que no haya un $\mathbb{Y} \in \mathbb{Q} \cap N$ que domine a \mathbb{X} .
3. Un punto $\mathbb{X} \in \mathbb{Q}$ es llamado óptimo débil de Pareto del POM si no existe $\mathbb{Y} \in \mathbb{Q}$ tal que $F(\mathbb{Y}) <_p F(\mathbb{X})$.
4. El conjunto de todas las soluciones óptimas de Pareto se llama conjunto de Pareto, es decir:

$$P = P_Q = \{\mathbb{X} \in \mathbb{Q} : \mathbb{X} \text{ es el punto de Pareto del POM}\} \quad (2-53)$$

5. La imagen $F(P)$ de P es llamada frontera de Pareto.

2.3.1. POM para el DI

En este apartado se encuentra la solución del POM para el DI con los dos primeros parámetros de optimización estudiados en este capítulo (esfuerzo de control y variación total), a continuación se muestra el procedimiento, iniciando con la definición del POM.

$$(POM) \quad \min_{u \in U} [f_1(u), f_2(u)], \quad (2-54)$$

donde,

$$f_1(u) = \frac{1}{2} \int_{t_0}^{t_1} u^2(t) dt$$

$$f_2(u) = \int_{t_0}^{t_1} |\dot{u}(t)| dt$$

Ahora se encuentra el conjunto de Pareto para dos parámetros de optimización. Primero se emplea una nueva variable denominada $\alpha \in [0, 1]$, que hace posible optimizar un solo parámetro (esfuerzo de control o variación total) cuando se asigne a α alguno de los valores extremos del conjunto indicado. En la Ec. (2-55), se observa el problema de optimización multiobjetivo.

$$\min_{u \in U} \quad \alpha f_1(u) + (1 - \alpha) f_2(u) \quad (2-55)$$

En [67] se muestra el desarrollo matemático para la obtención de las expresiones de control (2-56) del POM para el DI con parámetros de esfuerzo de control y variación total.

$$u(t) = \begin{cases} u_1, & \text{si } 0 \leq t < t_1 \\ u_1 + \bar{\lambda}_1(t - t_1), & \text{si } t_1 \leq t < t_2 \\ u_3, & \text{si } t_2 \leq t < 1 \end{cases}, \quad (2-56)$$

donde, $\bar{\lambda}_1$ se encuentra con (2-57), t_1 se despeja de (2-58) conociendo las condiciones iniciales y finales del problema, t_2 se halla con la expresión $t_2 = 1 - t_1$, u_1 y u_3 se muestran en las Ec. (2-59) y (2-60) respectivamente.

$$\bar{\lambda}_1 = \pm \frac{2\alpha}{t_1^2} \quad (2-57)$$

$$4t_1^3 - 3 \left(2 \pm \frac{x_2(1) + x_2(0) + 2(x_1(0) - x_1(1))}{\alpha} \right) t_1^2 + 1 = 0 \quad (2-58)$$

$$u_1 = x_2(1) - x_2(0) - \frac{\bar{\lambda}_1}{2}(1 - 2t_1) \quad (2-59)$$

$$u_3 = 2(x_2(1) - x_2(0)) - \bar{u}_1 \quad (2-60)$$

Para encontrar las señales de control que solucionan el POM descrito en (2-55) del ejemplo estudiado en el POS, se toma $t \in [0, 1]$, la condición inicial $X_0 = [-10, -4]^T$ y la condición final $X_1 = [0, 0]^T$. En la Figura 2-8a se muestra el conjunto de Pareto para el POM de (2-55). En esta gráfica se puede observar que el esfuerzo de control y la variación total dependen del valor seleccionado de α , siendo $\alpha = 1 * 10^{-6}$ el POS para minimizar el esfuerzo de control y $\alpha = 1 * 10^6$ el POS para minimizar la variación total de la señal. En la Figura 2-8a se detallan las señales de control para algunos valores de α .

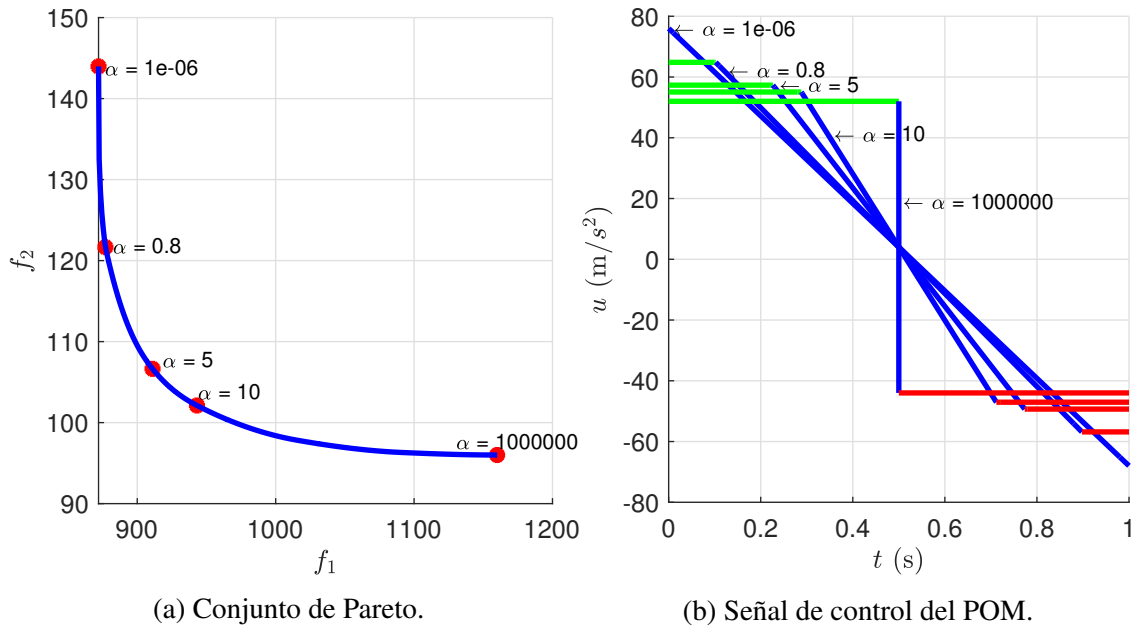


Figura 2-8.: Conjunto de Pareto y señal de control para el POM.

Si se soluciona el POM para el DI con $\alpha = 0.8$ se obtiene la posición 2-9a, velocidad 2-9b y señal de control 2-9c de la partícula.

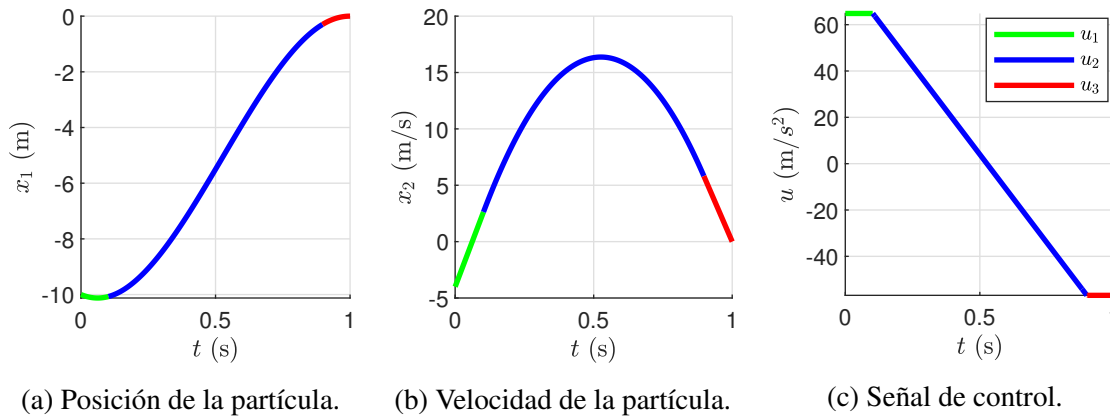


Figura 2-9.: Señales en el tiempo del POM para $\alpha = 0.8$.

2.4. Conclusión

Como se puede apreciar, el POS y POM en tiempo continuo pueden ser solucionados de forma analítica en algunos casos, pero se hace un poco extenso su desarrollo matemático, debido a que es necesario aplicar el cálculo de variaciones y el mínimo de Pontryagin. Estos problemas se pueden solucionar de manera numérica empleando técnicas de sintonización como son el LQG y el LQR. En el capítulo siguiente se solucionan algunos de estos problemas de optimización de forma numérica, por medio de la aplicación del AMC, el cual convierte el problema continuo en uno de optimización combinatoria en grafos. Se realiza una comparación entre el caso continuo y discreto para algunos problemas, con el fin de observar la pertinencia del método implementado.

Para encontrar trayectorias con un mejor desempeño en varios parámetros es necesario emplear el POM para combinar diferentes objetivos de optimización y tener en cuenta las posibles situaciones que se pueden encontrar a la hora de controlar sistemas dinámicos. El problema de energía cinética en este capítulo se dejan planteado para ser solucionado de forma numérica en el siguiente capítulo.

3. Optimización combinatoria empleando el AMC para el DI en 1D

En el capítulo anterior se realizó el análisis en tiempo continuo a diferentes problemas de optimización sobre el movimiento del DI en 1D. Ahora se pretende mostrar por medio de la aplicación del AMC como éstos se pueden transformar en problemas de optimización combinatoria. Primero se debe construir el espacio de estados de celda simulando la evolución del modelo dinámico del DI ante diferentes señales de control constantes en un tiempo dado. Luego se determinan las celdas existentes en la discretización y las conexiones entre ellas, guardando los costos de tiempo, esfuerzo de control y energía cinética que implican llevar la partícula de una celda a otra. Estos valores de conexión son almacenados en diferentes grafos, los cuales contienen las celdas conectadas, así como el peso generado. Los grafos son empleados para realizar la búsqueda del camino más corto que une a dos nodos en el espacio de estados de celda y por tanto, se tendrá una trayectoria subóptima que une el origen con el destino.

3.1. Mapeo de Celdas

En un intento por encontrar caminos más eficientes y prácticos para determinar el comportamiento global de sistemas fuertemente no lineales, el AMC fue propuesto en la década de los 80s. La idea básica detrás de este método es no considerar el espacio de estados como continuo sino como una colección de un gran número de celdas de estado, donde cada celda es tomada como una entidad de estado. El enfoque inicial del AMC era el análisis global de la dinámica de sistemas y en la actualidad se aplica para la búsqueda de soluciones en problemas de optimización [90], diseño de controladores no lineales, entre otras.

Actualmente el mapeo de celdas desarrollado en la década de los 80 por C. S. Hsu es de interés para la comunidad científica, debido al desarrollo de sistemas computacionales de alto desempeño y la proliferación de los datos a gran escala. En [91] se muestra una descripción del método y algunas aplicaciones del mapeo de celdas dentro de la ingeniería. Hasta ahora 2 tipos de mapeo de celda han sido investigados, el simple y el generalizado.

En el AMC se consideran vecinos de una celda a todos los posibles destinos de la evolución de un sistema. Para considerarse vecino, los destinos deben estar dentro del espacio de estados. Los estados que se encuentran fuera del espacio de trabajo definido inicialmente no se tienen en cuenta para la construcción del grafo. Cada sistema dinámico tiene sus propias limitaciones, y si se considera el desplazamiento de una partícula como caso de estudio, las restricciones serán de posición, velocidad y aceleración.

El algoritmo empleado es el mapeo de celdas simple (MCS) y se implementa para discretizar el espacio de estados del problema de optimización del DI, que pueden ser representados por medio de un modelo dinámico lineal. Para emplear el AMC en la construcción de rutas de referencia para el movimiento del DI, se realiza lo siguiente:

1. Se genera un mapa con cada una de las variables de posición y velocidad lineal del DI, tomando en cuenta las dimensiones del entorno y los límites de velocidad de la partícula.
2. Se define un conjunto de valores constantes que serán aplicados al modelo dinámico del DI. Estos elementos son las señales de control (la aceleración de la partícula) y son constantes para que en cada celda su variación total sea nula.
3. Se determina el tiempo de aplicación de las señales de control.
4. Se guarda en un grafo la condición inicial y final de la evolución de cada iteración, así como el costo empleado en la simulación.
5. Se hace de forma iterativa los pasos 2, 3 y 4 para cada una de las celdas del espacio de estados.

La característica del mapeo de celdas simple (MCS) [92] es que cada celda tiene una sola celda imagen; mientras que en el mapeo generalizado [93],[94] una celda tiene muchas casillas imagen con una distribución de probabilidad dada entre ellas. Ambos métodos son utilizados para determinar el comportamiento global de un sistema no lineal. Cada mapeo de celdas tiene su propia estructura matemática y es considerada de interés por sí misma. Para los casos de estudio de este documento se implementa el MCS, puesto que existe una única celda imagen para el sistema dinámico ante una entrada específica con condiciones iniciales dadas.

El problema del mapeo de celdas se puede definir como se expresa en la Ec. (3-1).

$$x_{k+1} = C(x_k, \mu), \quad (3-1)$$

donde x_{k+1} es la imagen de x_k aplicando el mapeo de celdas simple C con un vector de parámetros μ . El mapeo de celdas puede depender del paso k , pero en este trabajo se considera un MCS estacionario donde C no depende explícitamente de k como se hace en [54].

El proceso de evolución del sistema dado en la Ec. (3-1) se realiza para cada una de las celdas del espacio de estados, aplicando las diferentes señales de control. Al momento de terminar la evolución, se guarda en un grafo la información del nodo x_k y x_{k+1} , así como el tiempo empleado y la señal de control aplicada.

El AMC permite discretizar el espacio de estados, para solucionar diferentes problemas de optimización por medio de la búsqueda del camino más corto en un grafo. Entre los problemas de optimización que se pueden solucionar, están: el de mínimo tiempo de navegación en el movimiento de una partícula [58], mínima distancia, menor esfuerzo de control, menor energía, entre otros. También se puede emplear el diseño de controladores realimentados para sistemas lineales [56]. En este trabajo se propone implementar el AMC para solucionar diferentes problemas de optimización combinatoria que son equivalentes a los de tiempo continuo.

3.2. AMC para el DI

En esta sección se realiza la evolución del modelo matemático del DI empleando la Ec. (3-1) y se muestran los diferentes vecinos de una celda según las condiciones iniciales de la partícula.

3.2.1. Vecinos de una celda

En la implementación del AMC para el doble integrador se tienen las variables de estado x_1 y x_2 que representan la posición y velocidad de la partícula respectivamente. Con los límites de velocidad y aceleración del problema definidos, se puede construir el espacio de estados de celda teniendo diferentes valores de control constantes aplicados durante un tiempo específico. Se toman las señales de control constantes para tener variación total cero en cada celda. A continuación, se muestra el análisis de los vecinos de una celda inicial $x_1[0] = 0$ m teniendo 17 señales de control diferentes $u \in \{-0.4, -0.35, \dots, -0.05, 0, 0.05, \dots, 0.35, 0.4\}$ m/s² aplicadas durante 1 s.

Para encontrar todas las celdas vecinas se realiza la evolución del modelo dinámico del DI con las señales de control dadas. Se obtiene la Figura 3-1a con $x_2[0] = 0$ m/s; en ésta se observan varios puntos intermedios entre celdas, pero al momento de construir el grafo solo se consideran los valores iniciales y finales de la simulación. En la Figura 3-1b se muestra la evolución con $x_2[0] = 0.2$ m/s y en la Figura 3-1c con $x_2[0] = -0.2$ m/s.

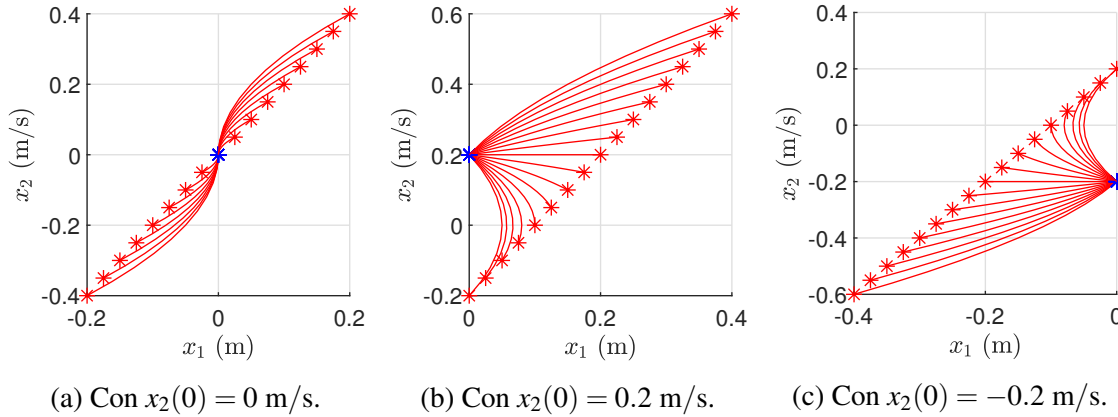


Figura 3-1.: Celdas vecinas con diferentes condiciones iniciales de velocidad. El asterisco azul representa la celda inicial y los rojos las celdas finales o vecinas.

Como se puede apreciar en la Figura 3-1, al aumentar los elementos del conjunto de señales de control se incrementan los vecinos de una celda. Según la forma en la que fue construida la estructura de datos, la resolución de la celda de velocidad depende del delta de la señal de control y el tiempo de aplicación de la misma. Mientras que la resolución de la celda de posición depende del delta encontrado para la velocidad y el tiempo.

3.2.2. Generación del grafo a partir de los vecinos de celda

Se desea construir una estructura de datos que contenga la información de todos los posibles vecinos celda del espacio de estados, así como el costo generado para ir a cada una de estas. En esta tesis se construye una estructura de datos abstracta llamada grafo para cada uno de los parámetros a optimizar. Para generar estas estructuras es necesario realizar la simulación con todas las condiciones iniciales posibles, dadas por la discretización adoptada. Esto conlleva a un gasto computacional que crece exponencialmente según las combinaciones de las aceleraciones (señales de control), velocidades, posiciones y tiempo de activación de la señal de control. En este trabajo no se agrega la variable de tiempo en la combinatoria porque se toman siempre fijas.

Luego se evalúa en que celda terminó la evolución del sistema; si ésta se encuentra fuera del espacio de estados, no se procede a almacenar la conexión ni el costo para llegar a ella. A continuación, se muestra un ejemplo en el que se toman la señales de control $u = \{-0.1, 0, 0.1\}$ m/s² y el espacio de estados definido por: $x_1 \in \{-0.2, -0.1, 0, 0.1, 0.2\}$ m y $x_2 \in \{-0.2, -0.1, 0, 0.1, 0.2\}$ m/s. Todas las señales de control se aplican durante 1 s. Se toma esta discretización para poder ilustrar de la mejor manera al lector.

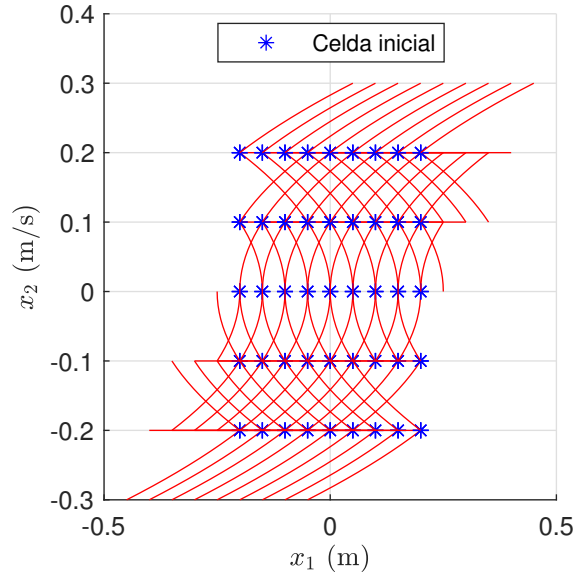


Figura 3-2.: Conexiones entre celdas para el movimiento del DI en 1D.

Luego de simular el sistema se analizan las celdas alcanzadas con cada una de las entradas de control para las diferentes condiciones iniciales. Por medio de esta evolución se obtienen las posiciones y velocidades de la partícula. En la estructura de datos cada asterisco azul simboliza la celda origen y destino de la evolución del sistema. Las aristas representan el costo empleado en cada simulación desde una celda inicial a una final. En el Algoritmo 1 se muestra como se generan grafos de tiempo, esfuerzo de control y energía cinética de la partícula, n_{cellsp} y n_{cellsv} son la cantidad de particiones en la dimensión de posición y velocidad respectivamente. G_t , G_u y G_{E_k} son los grafos de tiempo, esfuerzo de control y energía cinética respectivamente.

Algoritmo 1: AMC para el DI. *Los grafos se calculan con $t_s = 1$ s

Result: G_t, G_u, G_{E_k}

$x_{k+1} = C(x_k, u) \rightarrow \text{AMC};$

$x_1[0], x_2[0], t_0;$

while $i \leq n_{cellsp}$ **do**

while $j \leq n_{cellsv}$ **do**

Evaluar las restricciones cinemáticas y dinámicas;

 Calcular $t_s^*, \frac{1}{2}u^2, \frac{1}{2}mx_2^2;$

$G_t(i, i+1) \leftarrow t_s^*;$

$G_u(i, i+1) \leftarrow \frac{1}{2}u^2;$

$G_{E_k}(i, i+1) \leftarrow \frac{1}{2}mx_2^2;$

$j = j + 1;$

end

$i = i + 1;$

end

Se puede resolver el POS y POM en este trabajo, debido a que se generaron diferentes estructuras. Con estos grafos es posible emplear un método para encontrar el camino más corto entre dos nodos, seleccionando una función objetivo a minimizar que contenga los parámetros a optimizar. Con esta función se procede a implementar un algoritmo de búsqueda del camino más corto en un grafo para dar solución al problema de optimización combinatoria.

3.2.3. Algoritmo de búsqueda del camino más corto

Ahora se desea realizar la búsqueda del camino más corto en un grafo, para lo cual es necesario entender el concepto de optimización discreta, la cual usa valores finitos de un conjunto de datos y es empleada en las matemáticas y la informática. La optimización discreta se divide en dos ramas: programación de enteros, donde el conjunto discreto contiene números enteros; y optimización combinatoria, donde el conjunto discreto contiene objetos o estructuras combinatorias como: secuencias, rutas, entre otras. La optimización combinatoria busca minimizar o maximizar una función objetivo sobre un conjunto finito. Debido a que el conjunto de valores es finito, implica que las variables son discretas. Se pueden emplear dos tipos de algoritmos para resolver estos problemas de optimización: los exactos, que encuentran la solución óptima; y los heurísticos, que encuentran una buena solución sin garantizar que sea la óptima. En los problemas de optimización combinatoria se utilizan grafos, matroides y otras estructuras discretas para la representación de los datos.

Las estructuras empleadas para representar la información dentro del AMC son los grafos denominados G . Estos grafos contienen vértices o nodos y aristas. Las aristas unen un par de nodos y contienen la información de distancia o distancia geodésica por medio de un valor que determina el costo para desplazarse entre esos dos nodos. Se dice que dos nodos son vecinos si contienen una arista que los una.

La teoría de grafos se remonta al año 1736 en donde Leonhard Euler encontró la solución del problema para recorrer la ciudad de Königsberg pasando sólo una vez por cada uno de los siete puentes existentes en la localidad y regresando de nuevo al punto de inicio. Esta teoría fue estudiada por primera vez como teoría matemática por Julius Petersen en 1891 con la publicación [95], que fue un aporte significativo para la teoría de grafos moderna. Esta teoría ha sido de importancia desde el año 1960 hasta la fecha, en donde se observan diferentes publicaciones que se fundamentan en la matemática de grafos. Actualmente los grafos son aplicados en redes sociales, aplicaciones semánticas (Alexa, Google Home o Siri), para el cálculo de rutas óptimas sobre mapas (Google Maps, Waze), entre otras aplicaciones. Algunos trabajos destacados que desarrollan su investigación fundamentada en grafos son: [96], donde se observa un grafo de conocimiento para la optimización de trayectorias combinado con el algoritmo de colonia de hormigas para la solución del problema del vendedor viajero (TSP). [97], donde se muestra un algoritmo para encontrar el camino óptimo entre dos nodos de un grafo.

El problema de encontrar el camino más corto entre dos nodos de un grafo se denomina optimización combinatoria. Algunos métodos empleados para encontrar la ruta más corta en un grafo son: Algoritmo de Dijkstra [98], Algoritmo de búsqueda A^* [99], Algoritmo Floyd-Warshall, Algoritmo Bellman-Ford, Algoritmo de Johnson, Algoritmo de Viterbi, entre otros.

El algoritmo de Dijkstra fue desarrollado por el físico y científico de la computación Edsger Dijkstra en 1959 y se implementa para encontrar el camino más corto entre dos vértices contenidos en un grafo. Este método se puede implementar cuando los pesos (aristas) son no negativos, debido a que encuentra la ruta con menor costo sumando los valores de cada arista. El método de Dijkstra es un algoritmo voraz o greedy, el cual utiliza una estrategia de búsqueda con ayuda de una heurística que determina la solución óptima en cada paso con la esperanza de alcanzar la ruta óptima global para disminuir el costo computacional. Si en el análisis posterior a una etapa surge una solución mejor, el óptimo encontrado en esa etapa es modificado. El orden de complejidad computacional del método sin optimizarse es de $O(|V|^2 + |A|)$, donde A es el número de aristas y V el de vértices.

Debido a que el problema que se presenta es el de un grafo no dirigido con pesos positivos se emplea el Algoritmo de Dijkstra, por su facilidad de implementación y su amplia aplicación en búsqueda sobre grafos. El método permite determinar la trayectoria de la partícula que minimiza la función de costo en el grafo. En la siguiente sección se muestran algunos resultados con diferentes funciones objetivo.

3.3. Solución al problema de optimización combinatoria para el DI por medio del AMC

Para solucionar los problemas de mínimo esfuerzo de control, mínimo tiempo de navegación y minimización del gasto energético con el AMC se emplea la siguiente discretización: se toman señales de control constantes tal que $u \in \{-0.4, -0.35, \dots, -0.05, 0, 0.05, \dots, 0.35, 0.4\}$ m/s². El espacio de estados está definido por: $x_1 \in [-1, 1]$ m y $x_2 \in [-0.8, 0.8]$ m/s, con un incremento $\Delta x_1 = 0.025$ m y $\Delta x_2 = 0.05$ m/s respectivamente. Todas las señales de control se aplican durante 1 s.

Para la discretización planteada anteriormente se realiza la evolución del sistema cambiando las condiciones iniciales y se construyen los grafos de conexiones entre celdas con los diferentes parámetros a optimizar: G_u (esfuerzo de control), G_t (tiempo) y G_{E_k} (energía cinética).

3.3.1. Generación de trayectoria con menor esfuerzo de control

Para generar la trayectoria con menor esfuerzo de control se implementa el algoritmo de Dijkstra para encontrar el desplazamiento de la partícula que minimiza la función de costo de la Ec. (3-2). El problema de optimización se plantea en la Ec. (3-3).

$$F_u = \frac{1}{2} \sum_{i=0}^{n-2} u_i^2 \quad (3-2)$$

$$\left\{ \begin{array}{l} \text{mín} \quad \frac{1}{2} \sum_{i=0}^{n-2} u_i^2, \\ \text{sujeto a:} \quad q_{i+1} = C(q_i, u_i), \\ \quad \quad \quad q_0 = [x_1[0], x_2[0]]^T, \quad q_{n-1} = [x_1[n-1], x_2[n-1]]^T, \\ \quad \quad \quad u_i = \{-\bar{u}, 0, \bar{u}\} \end{array} \right. \quad (3-3)$$

donde q es el vector de estados conformado por la posición x_1 y velocidad x_2 ; $x_1[0]$ es la posición origen del DI con una velocidad inicial $x_2[0]$ y $x_1[n-1]$ es la posición destino con velocidad final $x_2[n-1]$. El DI tardaría $t = (n-1)t_s$ s en completar el movimiento.

Para el ejemplo se analiza el espacio de estados en \mathbb{R}^2 (posición y velocidad). Si se toma como condición inicial $x_1[0] = -1$ m, $x_2[0] = 0$ m/s y final $x_1[n-1] = 1$ m, $x_2[n-1] = 0$ m/s, se obtiene la trayectoria de referencia de la Figura 3-3a y el perfil de velocidad mostrado en la Figura 3-3b que implica el menor esfuerzo de control posible. En estas figuras se muestra la comparación entre lo obtenido por el AMC y el problema continuo (explicado en el capítulo 2). En la Figura 3-3c se muestra la señal de control aplicada a la partícula que soluciona el problema de optimización con mínimo esfuerzo de control.

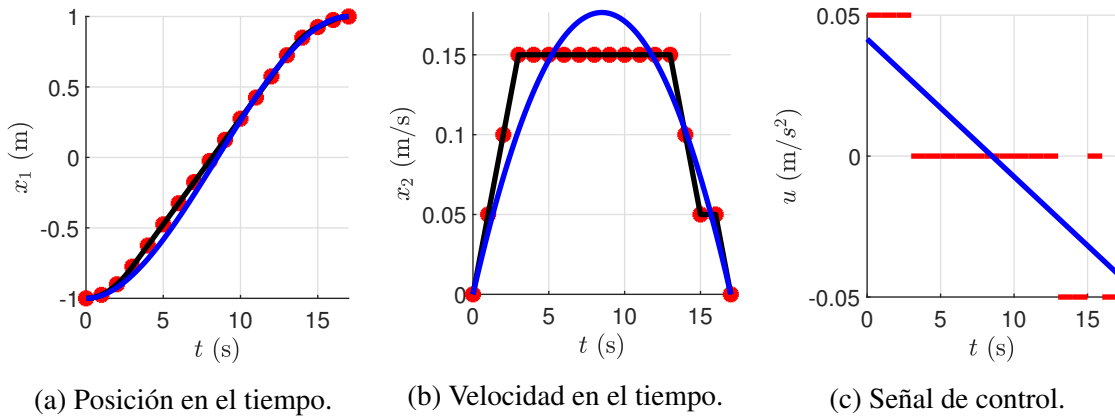


Figura 3-3.: Posición, velocidad y aceleración para minimizar esfuerzo de control. Los puntos rojos muestra la solución del AMC, la línea negra es una interpolación de los valores del AMC y la línea azul es la solución continua.

En la Figura 3-4 se muestran los errores de posición y velocidad en el tiempo entre el POS continuo y la solución del AMC con la discretización definida en la sección 3.3. En esta gráfica se muestra que el error de posición máximo es de $|0.105|$ m, que al comparar con la discretización adoptada es de un poco más de 4 celdas en el espacio de estados. Mientras que el de velocidad es de $|0.05|$ m/s, diferencia que equivale a una celda. Esta precisión puede modificada disminuyendo el tamaño de las celdas y/o aplicando señales de control durante un tiempo menor al empleado. También se observa que hay unas oscilaciones alrededor de cero, así como un error nulo en el inicio y final de las gráficas, esto último, debido a que la partícula está en la posición y velocidad deseada.

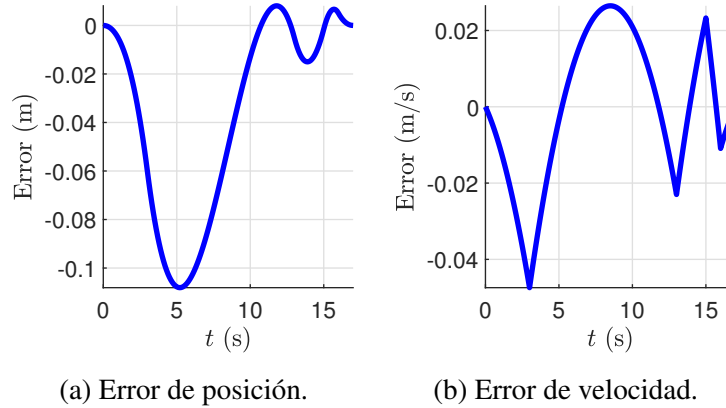


Figura 3-4.: Error del AMC en la minimización del esfuerzo de control.

Ahora se soluciona el mismo problema de optimización con el AMC con la siguiente discretización: se toman señales de control constantes tal que $u \in \{-0.4, -0.3, -0.2, -0.1, 0, 0.1, \dots, 0.3, 0.4\}$ m/s^2 . El espacio de estados está definido por: $x_1 \in [-1, 1]$ m y $x_2 \in [-0.8, 0.8]$ m/s, con un incremento $\Delta x_1 = 0.05$ m y $\Delta x_2 = 0.1$ m/s respectivamente. Todas las señales de control se aplican durante 1 s. El tiempo de

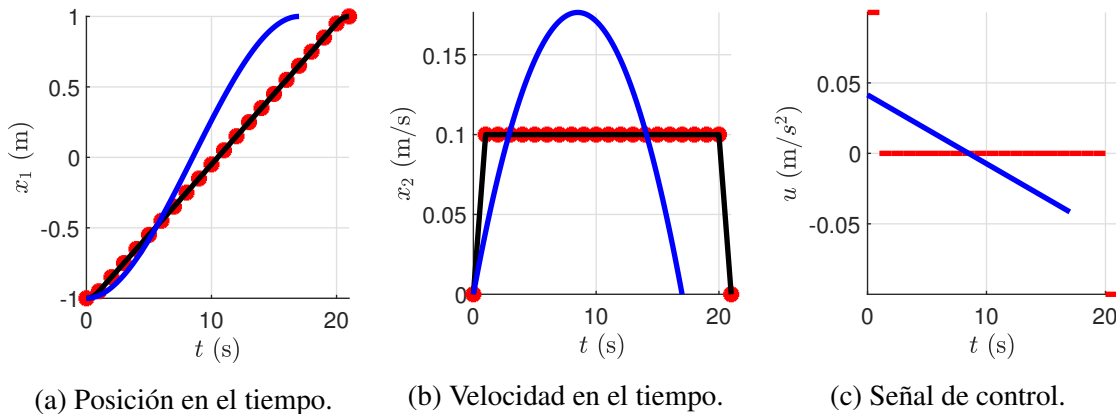


Figura 3-5.: Variables de estado para minimizar esfuerzo de control. Los puntos rojos muestra la solución del AMC con la segunda discretización, la línea negra es una interpolación de los valores del AMC y la línea azul es la solución continua.

Al realizar la medición del esfuerzo de control en las señales resultantes de los problemas continuo y discreto, se tiene que la solución continua presenta un valor de 0.0049, la obtenida por el AMC para la primera discretización es de 0.0075 y en la segunda discretización es de 0.02. Como es de esperar, la que presenta menor esfuerzo de control es la encontrada por el problema continuo. Sin embargo, las obtenidas con el AMC son las menores para cada discretización. Cabe destacar que la solución discreta se acerca a la continua a medida que se disminuye el tamaño de celda, por lo que se puede considerar que la solución numérica es una buena aproximación a la continua.

Costo computacional: Si se analiza el costo computacional en los dos casos de discretización del AMC, se tiene que: en el primero, el tiempo empleado en la construcción del grafo es de 321.51 s para un espacio de estados de 2673 celdas; y en el segundo, el tiempo empleado es de 44.82 s para un espacio de estados de 697. Como se puede apreciar al disminuir de 17 señales de control con un delta de 0.05 a 9 con un incremento de 0.1, se disminuyen la cantidad de celdas en el espacio de estados 3.84 veces y el tiempo de ejecución 7.17 veces.

3.3.2. Generación de trayectoria con menor tiempo de navegación

Para la misma discretización se analiza el problema de mínimo tiempo de navegación con el algoritmo de Dijkstra para encontrar el camino más corto que minimiza la función de costo de la Ec. (3-4). El problema de optimización se plantea en la Ec. (3-5).

$$F_u = \frac{1}{2} \sum_{i=0}^{n-2} t_s \tag{3-4}$$

$$\left\{ \begin{array}{l} \text{mín} \quad \frac{1}{2} \sum_{i=0}^{n-2} t_s, \\ \text{sujeto a:} \quad q_{i+1} = C(q_i, u_i), \\ \quad \quad \quad q_0 = [x_1[0], x_2[0]]^T, \quad q_{n-1} = [x_1[n-1], x_2[n-1]]^T, \\ \quad \quad \quad u_i = \{-\bar{u}, 0, \bar{u}\} \end{array} \right. \tag{3-5}$$

En la Figura 3-6a se observa la posición de la partícula y el perfil de velocidad mostrado en la Figura 3-6b que implica el menor tiempo de navegación posible. En la Figura 3-6c se muestra la señal de control aplicada a la partícula que soluciona el problema de optimización.

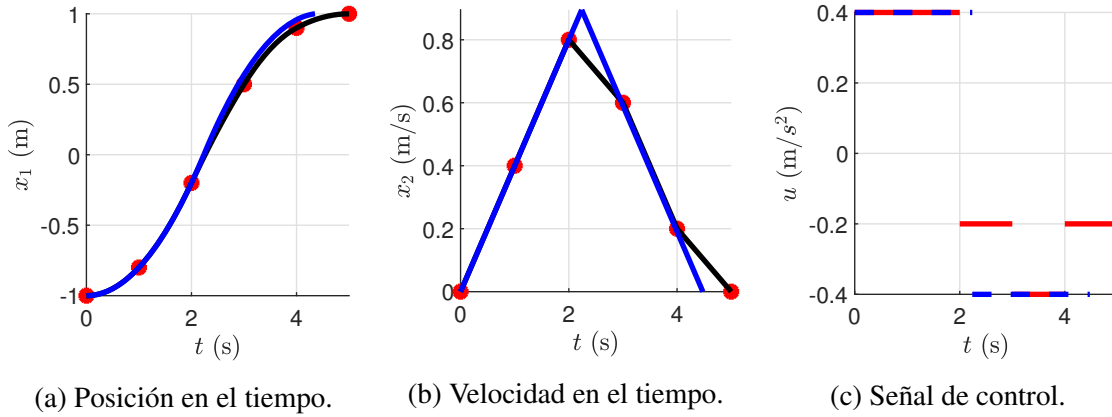


Figura 3-6.: Posición, velocidad y aceleración para minimizar el tiempo de navegación. Los puntos rojos muestra la solución del AMC, la línea negra es una interpolación de los valores del AMC y la línea azul es la solución continua.

Al realizar la medición de los tiempos de navegación en las dos soluciones, se tiene que la continua presenta un valor de 4.476 s para dos señales de control $\{-0.4, 0.4\}$ y la generada por el AMC es de 5 s. La solución del problema continuo es la de menor tiempo. Sin embargo, la obtenida con el AMC es la más cercana a la continua dentro de la discretización de tiempo adoptada. Si se requiere disminuir el valor en la solución discreta es necesario aplicar las señales de control durante un tiempo menor, debido a que en este caso cada una está activa 1 s y por esto el tiempo de navegación en el AMC es un valor entero.

3.3.3. Generación de trayectoria para el desplazamiento con menor energía cinética

Se analiza el problema de mínima energía cinética con el algoritmo de Dijkstra para encontrar el camino más corto que minimiza la función de costo de la Ec. (3-6). El problema de optimización se plantea en la Ec. (3-7).

$$F_u = \frac{1}{2}m \sum_{i=0}^{n-2} x_2^2 \quad (3-6)$$

$$\left\{ \begin{array}{l} \text{mín} \quad \frac{1}{2}m \sum_{i=0}^{n-2} x_2^2, \\ \text{sujeto a:} \quad q_{i+1} = C(q_i, u_i), \\ \quad \quad \quad q_0 = [x_1[0], x_2[0]]^T, \quad q_{n-1} = [x_1[n-1], x_2[n-1]]^T, \\ \quad \quad \quad u_i = \{-\bar{u}, 0, \bar{u}\} \end{array} \right. \quad (3-7)$$

En la Figura 3-7a se observa la posición de la partícula y el perfil de velocidad mostrado en la Figura 3-7b que implica la menor energía cinética posible. En la Figura 3-7c se muestra la señal de control aplicada a la partícula que soluciona el problema de optimización.

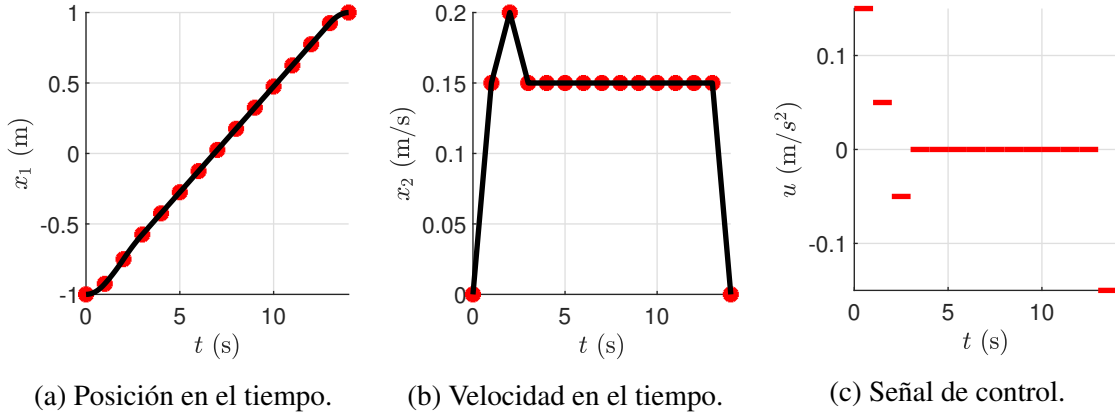


Figura 3-7.: Posición, velocidad y aceleración para minimizar el consumo energético. Los puntos rojos muestra la solución del AMC y la línea negra es una interpolación de estos.

3.3.4. Generación de trayectoria para solucionar el POM con parámetros de tiempo y esfuerzo de control

Se analiza el POM, empleando el algoritmo de Dijkstra para encontrar el camino más corto que minimiza la función de costo de la Ec. (3-10). Se puede encontrar un conjunto de Pareto para determinar los valores adecuados que optimizan los dos objetivos, pero en este ejemplo se toma la función objetivo indicada. El problema de optimización se plantea en la Ec. (3-11).

$$F_{POM} = \sum_{i=0}^{n-2} [0.3t_s + \frac{0.7}{2}u_i^2] \tag{3-8}$$

$$\left\{ \begin{array}{l} \text{mín} \quad \sum_{i=0}^{n-2} [0.3t_s + \frac{0.7}{2}u_i^2], \\ \text{sujeto a:} \quad q_{i+1} = C(q_i, u_i), \\ \quad \quad \quad q_0 = [x_1[0], x_2[0]]^T, \quad q_{n-1} = [x_1[n-1], x_2[n-1]]^T, \\ \quad \quad \quad u_i = \{-\bar{u}, 0, \bar{u}\} \end{array} \right. \tag{3-9}$$

En la Figura 3-8a y el perfil de velocidad mostrado en la Figura 3-8b que implica el menor esfuerzo de control posible. En la Figura 3-8c se muestra la señal de control dada para minimizar la función de costo indicada.

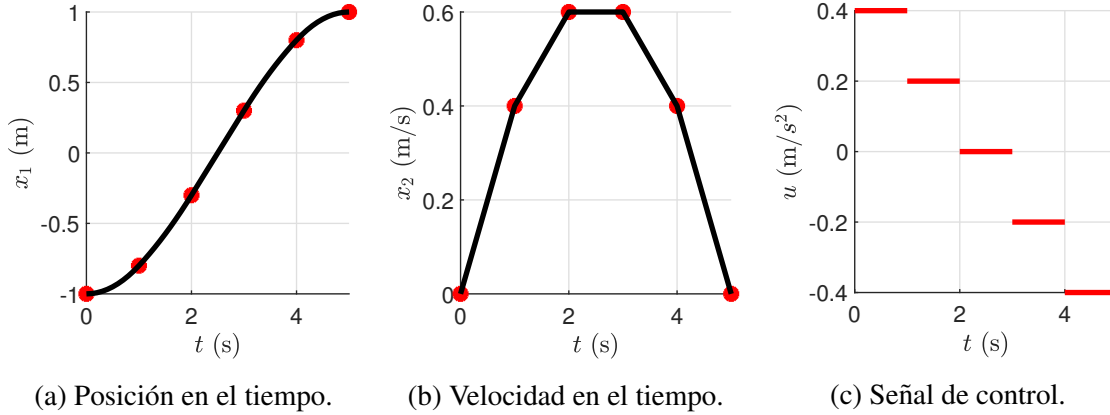


Figura 3-8.: Posición, velocidad y aceleración para minimizar tiempo y esfuerzo de control. Los puntos rojos muestra la solución del AMC y la línea negra es una interpolación de estos.

3.3.5. Generación de trayectoria para solucionar el POM con parámetros de tiempo, energía cinética y esfuerzo de control

Se analiza el POM, empleando el algoritmo de Dijkstra para encontrar el camino más corto que minimiza la función de costo de la Ec. (3-10). Se puede encontrar un conjunto de Pareto para determinar los valores adecuados que optimizan los tres objetivos, pero en este ejemplo se toma la función objetivo indicada. El problema de optimización se plantea en la Ec. (3-11).

$$F_{POM} = \sum_{i=0}^{n-2} \left[0.01t_s + \frac{0.2}{2}mx_2^2 + \frac{0.79}{2}u_i^2 \right] \quad (3-10)$$

$$\left\{ \begin{array}{l} \text{mín} \quad \sum_{i=0}^{n-2} \left[0.01t_s + \frac{0.2}{2}mx_2^2 + \frac{0.79}{2}u_i^2 \right], \\ \text{sujeto a:} \quad q_{i+1} = C(q_i, u_i), \\ \quad \quad \quad q_0 = [x_1[0], x_2[0]]^T, \quad q_{n-1} = [x_1[n-1], x_2[n-1]]^T, \\ \quad \quad \quad u_i = \{-\bar{u}, 0, \bar{u}\} \end{array} \right. \quad (3-11)$$

En la Figura 3-9a y el perfil de velocidad mostrado en la Figura 3-9b que implica la minimización de la función de costo planteada. En la Figura 3-9c se muestra la señal de control necesaria para solucionar el problema de optimización.

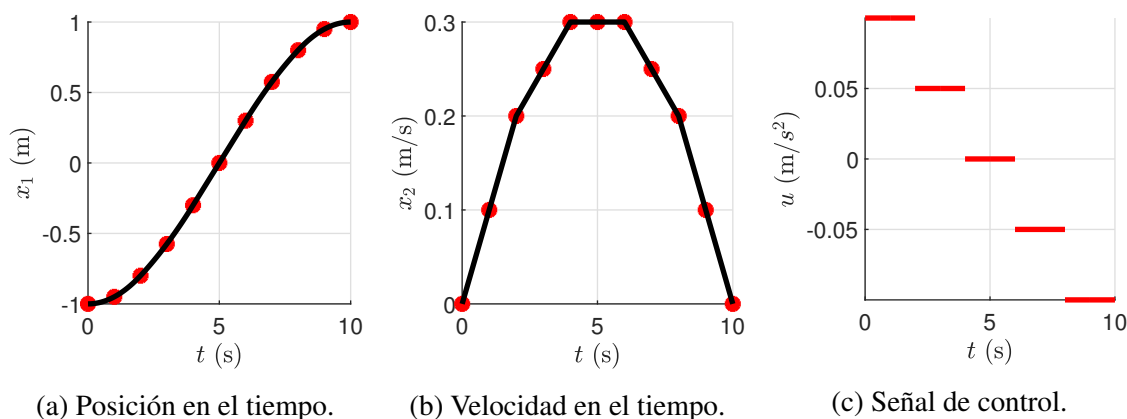


Figura 3-9.: Posición, velocidad y aceleración para minimizar tiempo, energía y esfuerzo de control. Los puntos rojos muestra la solución del AMC y la línea negra es una interpolación de los valores del AMC.

3.4. Análisis de las diferentes trayectorias generadas por el AMC

En la Tabla 3-1 se muestra la comparación de los datos de esfuerzo de control, variación total de la señal de control y tiempo de navegación calculados para cada una de las trayectorias generadas por el AMC en los casos de optimización estudiados. Como se puede observar, la trayectoria que tiene menor esfuerzo de control es la 2, la cual es construida para optimizar solo este parámetro. Las rutas 2 y 5 son las que tienen menor variación total de la señal de control. Los caminos 1 y 4 son los que tienen un menor tiempo de navegación.

Tabla 3-1.: Comparación de trayectorias del movimiento de una partícula en 1D.

| No. | Parámetro de optimización | Esfuerzo de control [u^2] | VT [u] | t [s] |
|-----|--|-------------------------------|------------|-------|
| 1 | Tiempo | 0.46 | 0.9 | 5 |
| 2 | Esfuerzo de control | 0.015 | 0.2 | 18 |
| 3 | Energía cinética | 0.065 | 0.45 | 15 |
| 4 | Tiempo y esfuerzo de control | 0.4 | 0.8 | 5 |
| 5 | Tiempo, energía cinética y esfuerzo de control | 0.05 | 0.2 | 10 |

Si se considera la combinación de los tres criterios de desempeño (esfuerzo de control, variación total y tiempo) se tiene que la trayectoria 5 combina de una manera eficiente los 3, puesto que tiene el segundo menor esfuerzo de control, la menor variación total de la señal de control y el segundo menor tiempo de navegación. Como se pudo apreciar, se puede encontrar una solución que combine varios parámetros de optimización por medio del AMC en la generación de referencias de las variables de estado.

3.5. Conclusión

El mapeo de celdas es un método que divide un problema global en varios locales, facilitando la solución de problemas de optimización, debido a que se analiza el comportamiento en pequeños intervalos. Luego se agrupan los resultados parciales para encontrar la solución general.

El AMC es un método que puede ser implementado para solucionar problemas de optimización de forma numérica, en el que ya no se requiere la solución global (con alta complejidad matemática en muchos casos), sino que se encuentran una serie de mínimos locales que se aproximan al problema continuo. El AMC logra una aproximación aceptable, que depende de la discretización adoptada. En la siguiente parte del trabajo de tesis se hace una extensión al problema en 2D, en el que se introduce el concepto de planificación de trayectorias al móvil más simple (DI).

Los problemas de optimización en tiempo continuo se pueden solucionar de manera discreta empleando optimización combinatoria en la búsqueda del camino más corto en un grafo por medio del algoritmo de Dijkstra. La solución es subóptima y se acerca a la solución analítica del problema continuo.

Parte II.

Planificación de trayectorias del DI en 2D

4. Planificación de trayectorias

Una parte indispensable en los robots móviles autónomos es la planificación de trayectorias, la cual permite a los sistemas móviles tomar la decisión de cómo moverse para completar una tarea. El problema de planificación de trayectorias se desarrolla en la mayoría de los casos para encontrar caminos entre dos puntos en \mathbb{R}^n , comúnmente se trabaja en el plano o en el espacio. Hay un gran número de enfoques a la hora de construir un algoritmo de planificación de trayectorias. Pero en general, los métodos empleados en la literatura buscan rutas que minimicen distancia, evadiendo obstáculos y generando trayectos suaves en posición y velocidad.

4.1. Planificadores de trayectorias

En la literatura se conocen a los algoritmos que generan rutas de referencia de robots como planificadores de trayectorias. Éstos brindan a los sistemas robóticos autonomía en la generación de la ruta a seguir. La navegación se define como la metodología que permite guiar el curso de un robot a través de un entorno con obstáculos. Su objetivo principal es hacer que los robots se muevan de manera independiente en su entorno de trabajo, en el cual deben ser capaces de orientarse y elegir la ruta adecuada que les permita desplazarse mientras completan la tarea. La navegación autónoma establece tres problemas de estudio:

- Localización (¿dónde estoy?), consiste en conocer en todo momento las coordenadas del robot respecto a cierto sistema de referencia.
- Planificación de tareas (¿dónde quiero ir?), se encarga de decidir acciones y su orden de realización para completar una misión.
- Planificación de movimientos (¿cómo puedo llegar ahí?), se encarga de generar automáticamente las acciones necesarias para que el robot se desplace entre posiciones cualesquiera del entorno, sin contratiempos, tales como, colisiones con obstáculos.

Para realizar la planificación de trayectorias se emplean diferentes formas de representación del entorno, algunos ejemplos son: mapa de ruta (roadmap) [100], descomposición en celdas [101], grafos o nodos. Esto depende del tipo de algoritmo que se va a implementar. Los planificadores se pueden clasificar en global o local, en el primero toda la información del entorno es conocida antes de iniciar, mientras que en el segundo se desconoce parte de la información del espacio de trabajo. En la Figura 4-1 se muestra una clasificación general de los algoritmos de planificación de trayectorias. En [102, 103] se muestran algunos métodos de planificación de trayectorias empleados en la literatura.

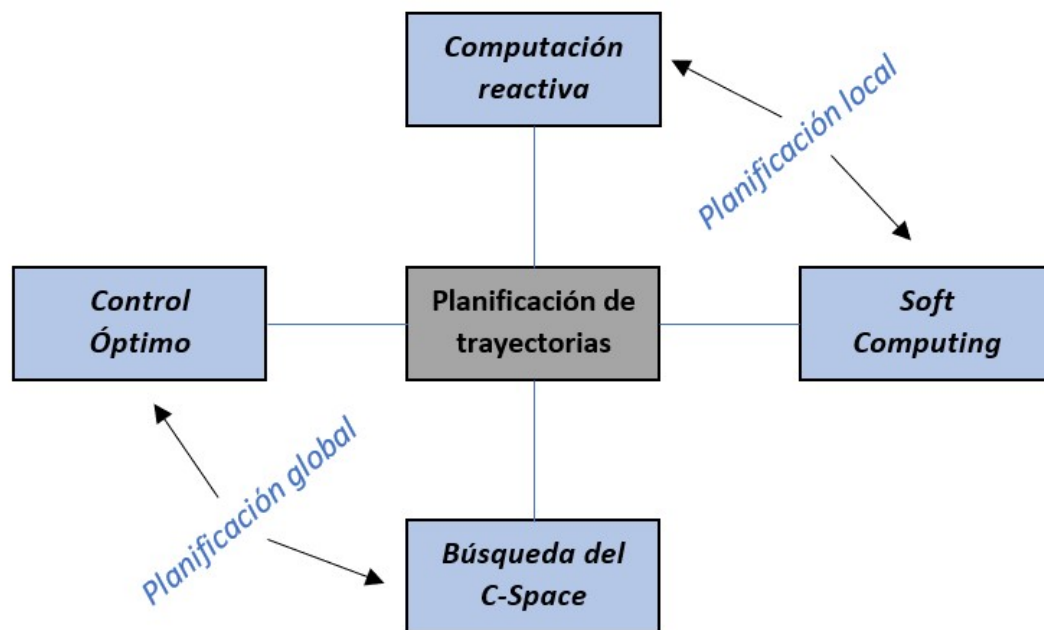


Figura 4-1.: Clasificación general de métodos de planificación de trayectorias.

4.1.1. Computación reactiva

Los algoritmos de planificación de trayectorias de esta clase se caracterizan por dividir el entorno en regiones ocupadas o libres con baja resolución y complejidad computacional. Es un método de planificación local, empleado ampliamente en entornos dinámicos, debido a la capacidad de evasión de obstáculos de manera rápida. Los robots que implementan este tipo de algoritmos deben ser dotados de sensores que le indiquen la presencia de obstáculos, con el fin de planificar de nuevo la trayectoria. Estos algoritmos se dividen en maniobra reactiva y métodos de optimización local. El primero realiza una maniobra evasiva ante la presencia de obstáculos y el segundo modifica la ruta planeada cuando detecta un espacio ocupado.

4.1.2. Soft Computing

Estos algoritmos se emplean en la planificación local y se diseñan para encontrar soluciones cercanas a la óptima con un rango de tolerancia especificado. Entre más cercana sea a la solución óptima, más costo computacional implica. En [104] se detalla el estudio de algunos algoritmos heurísticos basados en redes neuronales, lógica difusa, algoritmos inspirados en la naturaleza, entre otros.

Mirjalili y Dong en [105] proponen una clasificación para este tipo de algoritmos, esta es: métodos evolutivos, de control difuso y de aprendizaje automático. El primero utiliza técnicas inspiradas en la biología y la naturaleza, los últimos dos entran dentro de la categoría de inteligencia artificial, empleando reglas difusas y redes neuronales respectivamente, para implementar estrategias de control.

4.1.3. Búsqueda del C-Space

Estos algoritmos son empleados en la planificación global de trayectorias y consideran todo el espacio alcanzable por el robot. Es por esto que la mayoría de trabajos de esta categoría, se refieren al espacio de trabajo como el C-Space. Para implementar estos métodos se discretiza el espacio y se realiza la búsqueda visitando los nodos que lo comprenden. El camino resultante es la unión de cada una de las muestras encontradas y en algunos casos es necesario realizar un post-procesamiento para suavizar la trayectoria resultante.

Esta categoría se subdivide en dos grupos de algoritmos según la discretización empleada del C-Space. Los métodos de búsqueda en grafos y los basados en el muestreo. Estos últimos pueden seguir realizando la búsqueda aún después de encontrar una solución, con el fin de encontrar mejores rutas.

4.1.4. Control Óptimo

Estos algoritmos se emplean en la planificación de trayectorias global y se basan en la creación de una función de control que llevan al robot desde un estado inicial en el C-Space hasta el destino. Como su nombre lo indica, la planificación de trayectorias se aborda desde un enfoque de control óptimo [106] y se diferencian de los métodos de Soft Computing en que no hay parámetros configurables. En éstos existen dos subcategorías, en la primera se resuelven ecuaciones diferenciales parciales basándose en el principio de programación dinámica y en la segunda se emplea la optimización numérica teniendo en cuenta las restricciones cinemáticas y dinámicas del robot.

4.2. Algoritmos de planificación de trayectorias

En esta sección se mencionan algunos algoritmos empleados en la planificación de trayectorias de robots, empezando por los métodos clásicos, los de búsqueda en grafos, algoritmos reactivos, métodos bioinspirados, algoritmos heurísticos, entre otros.

4.2.1. Grafo de visibilidad

Para la implementación del grafo de visibilidad se debe representar el entorno por medio de figuras geométricas, donde cada vértice visible de los obstáculos se guardan en el grafo junto con las aristas que son las conexiones entre nodos. El algoritmo encuentra el camino óptimo en distancia entre vértices garantizando la evasión de obstáculos. En [107, 108] se observan algunas implementaciones del grafo de visibilidad. El método puede ser empleado en problemas de baja complejidad porque al ser un algoritmo de grafos la explosión combinatoria para encontrar un camino óptimo es grande. La complejidad computacional del algoritmo es de $O(N^2)$, donde N es la cantidad de nodos del grafo.

4.2.2. Diagrama de Voronoi

Este método también se puede encontrar en la literatura con el nombre de polígonos de Thiessen. Este algoritmo es uno de los métodos de interpolación más simples y se basan en la distancia euclidiana. Es un método de planificación global que encuentra trayectos no suaves que evaden obstáculos. La forma de representación del entorno es por medio de la construcción de figuras geométricas que encierran los obstáculos. El cálculo de este algoritmo es de forma rápida haciéndolo eficiente con respecto a tiempo y resultados. En [109, 110] se muestran algunos artículos que implementan diagramas de Voronoi.

4.2.3. Algoritmos de búsqueda del camino más corto

Una forma de planificación de trayectorias es emplear algoritmos de búsqueda del camino más corto en un grafo, en los que se puede encontrar:

Algoritmos de Dijkstra

Es un algoritmo de planificación de trayectorias global y se basa en la búsqueda del camino más corto en un grafo. Para implementarse es necesario clasificar el entorno en nodos y aristas. Éstos método encuentra rutas óptimas normalmente en distancia y no suaves. Es uno de los algoritmos más implementados para la planificación de trayectorias. Normalmente las aristas contienen información de distancia entre nodos, por lo que soluciona el problema de optimización combinatoria para ese parámetro. En [111] se muestra la implementación del algoritmo de Dijkstra.

Algoritmos de Floyd-Warshall

Es un método para encontrar el camino más corto en grafos dirigidos y fue desarrollado en 1959 por Bernard Roy. Este método tiene la característica de encontrar los caminos entre todos los pares de vértices del grafo con una sola ejecución. Con esta información se garantiza que al comparar todas las rutas se encuentra la solución óptima en el grafo. Es un algoritmo booleano que construye una matriz llamada de clausura transitiva en la que se muestran todas las uniones entre vértices, en el caso de no haber conexión entre algún par de nodos se asigna el valor de ∞ en la posición de la matriz. La complejidad del algoritmo es $O(V^3)$.

Algoritmos de Bellman-Ford

Es un método empleado para encontrar el camino más corto en un grafo dirigido y puede trabajar con valores negativos en las aristas. Fue desarrollado en 1958 por Lester Randolph Ford Jr. y Richard E. Bellamn. La complejidad computacional del algoritmo de Bellman-Ford es mayor al del Dijkstra, pero permite el empleo de costos negativos en el grafo. El algoritmo devuelve un valor booleano si encuentra un peso negativo.

4.2.4. Algoritmo A^* (A star)

Este algoritmo de planificación es empleado para encontrar el camino más corto en un grafo conociendo todo el entorno, por lo que es un método global. Éste se clasifica en los métodos de búsqueda en grafos de tipo heurístico o informado. Fue desarrollado en 1968 por Peter E. Hart, et al. Este método usa una función de evaluación heurística que etiqueta los nodos del grafo para determinar la probabilidad de que los vértices hagan parte del camino óptimo. La función de evaluación está compuesta por dos elementos, uno que le indica la distancia del nodo actual al vértice a etiquetar y el segundo le brinda información de distancia entre el vértice a etiquetar y el nodo destino. Si el problema de búsqueda del camino más corto tiene solución, el algoritmo A^* encuentra una ruta empleando menor gasto computacional que el de Dijkstra. La complejidad computacional del algoritmo A^* está relacionada con la calidad de la función de evaluación implementada, si hay una pésima elección de la heurística la complejidad será exponencial, en caso contrario la complejidad será lineal. Para implementar este algoritmo es necesario realizar la representación del espacio por medio de celdas o nodos, clasificándolas como ocupadas o no ocupadas. Este método encuentra la ruta con menor costo, normalmente en distancia y construye caminos no suaves. En [112] se muestra la aplicación del algoritmo A^* .

4.2.5. Algoritmo D* Lite (D star Lite)

Este algoritmo es una modificación del planificador de trayectorias A^* , con la ventaja de encontrar el camino de referencia en menor tiempo y posibilitando su aplicación de forma local. El algoritmo consisten en encontrar el camino más corto en un grafo, por lo que el espacio de trabajo debe representarse por medio de celdas o nodos. Este método encuentra el camino más corto normalmente en distancia construyendo trayectos no suaves. En [113] se muestra una aplicación del planificador D* Lite.

4.2.6. Campos potenciales artificiales (CPA)

Este algoritmo se clasifica dentro de la computación reactiva y es utilizado para generar rutas suaves en posición. En el método de campos potenciales artificiales, el robot se concibe como una partícula inmersa dentro de un campo de potencial cuyas variaciones locales reflejan la estructura del entorno. los obstáculos son modelados por cargas repulsivas y la meta como una carga atractiva. El movimiento del robot se establece de modo iterativo, calculando en cada paso la fuerza generada por el campo de potencial y utilizando la dirección de ésta para modificar el comportamiento del robot. Este algoritmo es utilizado ampliamente en la literatura y en [114–122] se muestra algunos artículos que implementan esta técnica para la planificación de trayectorias de robots.

Los algoritmos basados en campos potenciales constan al menos de dos componentes: uno de atracción hacia el objeto y otro de repulsión desde los obstáculos. La planificación de movimiento consiste en la adecuada combinación de ambos, de manera que conduzcan al robot hacia el objetivo sin colisiones. Esta operación denominada fusión de comandos, genera las referencias de control a los actuadores del robot en forma de velocidades $(\dot{p}, \dot{\phi})$ lineal y angular, respectivamente. En [123] se observa la implementación del planificador por campos potenciales artificiales para la navegación del robot Flamebot y de éste se extraen las ecuaciones para el desarrollo del algoritmo empleado en el capítulo.

Potencial de atracción

La fuerza de atracción genera el movimiento hacia el objetivo. Las entradas son las coordenadas actuales (x, y) y la posición final $(x(1), y(1))$ de la ruta. En la Ec. (4-1), se observa la fuerza de atracción de la posición actual hacia el objetivo, donde la constante K_{atr} es un parámetro de configuración, d_{obj} es la distancia entre la coordenada actual y el objetivo en cada instante. La dirección hacia la meta, $\delta_{atr}(t)$, es un ángulo con relación al sistema de referencia F_W , se puede calcular mediante la Ec. (4-2).

$$\vec{F}_{atr}(t) = K_{atr} \left(\frac{x(1) - x}{d_{obj}} \right) \hat{i} + K_{atr} \left(\frac{y(1) - y}{d_{obj}} \right) \hat{j} \quad (4-1)$$

$$\delta_{atr}(t) = \arctan \left(\frac{y(1) - y}{x(1) - x} \right), (0 \leq \delta_{atr} \leq 2\pi) \quad (4-2)$$

Potencial de repulsión

La fuerza de repulsión evita que el robot colisione con los obstáculos. Sus entradas son la estimación de la posición actual (x, y) y la posición del obstáculo (x_{obs}, y_{obs}) . En la Ec. (4-3), se muestra la fuerza de repulsión del obstáculo hacia la posición actual del robot, la constante K_{rep} es un parámetro de configuración, d_{obs} es la distancia del obstáculo hacia el robot en cada instante o iteración. La dirección de la fuerza, $\delta_{rep}(t)$, es un ángulo con relación al sistema de referencia F_W , se puede calcular mediante la Ec. (4-4).

$$\vec{F}_{rep}(t) = K_{rep} \frac{\cos(\delta_{rep})}{d_{obs}} \hat{i} + K_{rep} \frac{\sin(\delta_{rep})}{d_{obs}} \hat{j} \quad (4-3)$$

$$\delta_{rep}(t) = \arctan \left(\frac{y - y_{obs}}{x - x_{obs}} \right), (0 \leq \delta_{rep} \leq 2\pi) \quad (4-4)$$

4.2.7. Algoritmo de Bug

Como su nombre lo indica, este algoritmo simula el comportamiento de un bicho en un entorno desconocido. Se puede implementar en entornos donde no se tiene información del espacio libre y ocupado. Este genera una ruta entre el punto inicial y el objetivo, la cual es modificada si se encuentra un obstáculo. Al encontrar un bloqueo en el camino inicial, el algoritmo genera un trayecto alternativo rodeando el objeto y cuando tiene el campo de visión libre para seguir la ruta inicial, éste la sigue. En [124–126] se muestran algunos artículos que implementan el algoritmo de bug.

4.2.8. Algoritmos bioinspirados

Los métodos biológicos más utilizados en la literatura como planificadores de trayectorias de robots pueden emplear los siguientes algoritmos: redes neuronales, algoritmos genéticos, algoritmo de colonia de hormigas y algoritmo de filtro de partículas. La forma de representación del entorno varía según la técnica empleada, en algunos casos se hace por medio de celdas. Estos algoritmos encuentran el camino más corto y construyen trayectos que pueden ser suaves según la aplicación. Son planificadores locales porque no requieren conocer toda la información del entorno, en [127] se muestra una aplicación del algoritmo genético para la planificación de robots en entornos estáticos.

4.2.9. Árboles aleatorios de búsqueda rápida

Es un algoritmo aleatorio que está diseñada para emplearse en una amplia variedad de problemas de planificación de trayectorias. Fue introducido en 1998 por Steven M. LaValle [128] como una herramienta eficiente para la búsqueda de rutas en problemas con restricciones cinemáticas y dinámicas. Es un método que encuentra la solución al problema de planificación entregando una ruta factible de seguir sin tener en cuenta algún parámetro de optimización con un bajo costo computacional. En inglés se denomina Rapidly-exploring random tree (RRT).

4.3. Planteamiento del problema de planificación de trayectorias

El problema de planificación de trayectorias en robots se describe en la Ec. (4-5).

$$\dot{x} = f(x, u(t)), \quad x(t_0) = x_0, \quad x(t_f) = x_f, \quad (4-5)$$

donde, $u(t) \in U$ es la acción necesaria para llevar el móvil desde el estado x_0 al x_1 , por medio de la *ecuación de transición de estados* f .

Se define U como el *espacio de acción* para cada estado x , que representa el conjunto de acciones que pueden ser aplicadas. Como parte del problema de planificación se define el conjunto de estados X , así como X_G , este último es el conjunto de *estados objetivo* que satisface $X_G \subset X$.

4.4. Implementación del planificador por CPA

Para implementar el algoritmo de planificación por campos potenciales artificiales, se realiza la suma algebraica entre las componentes de atracción y repulsión en cada punto de la ruta. En la Figura 4-2a se observa una escena sin obstáculos para ir desde una condición inicial en posición $(x(0), y(0))$ hasta una final $(x(1), y(1))$, en ésta se genera una fuerza de atracción desde el origen hasta el objetivo. En la Figura 4-2b se muestra una escena con obstáculos y se emplea el algoritmo para encontrar una trayectoria con las mismas condiciones iniciales y finales que el caso anterior.

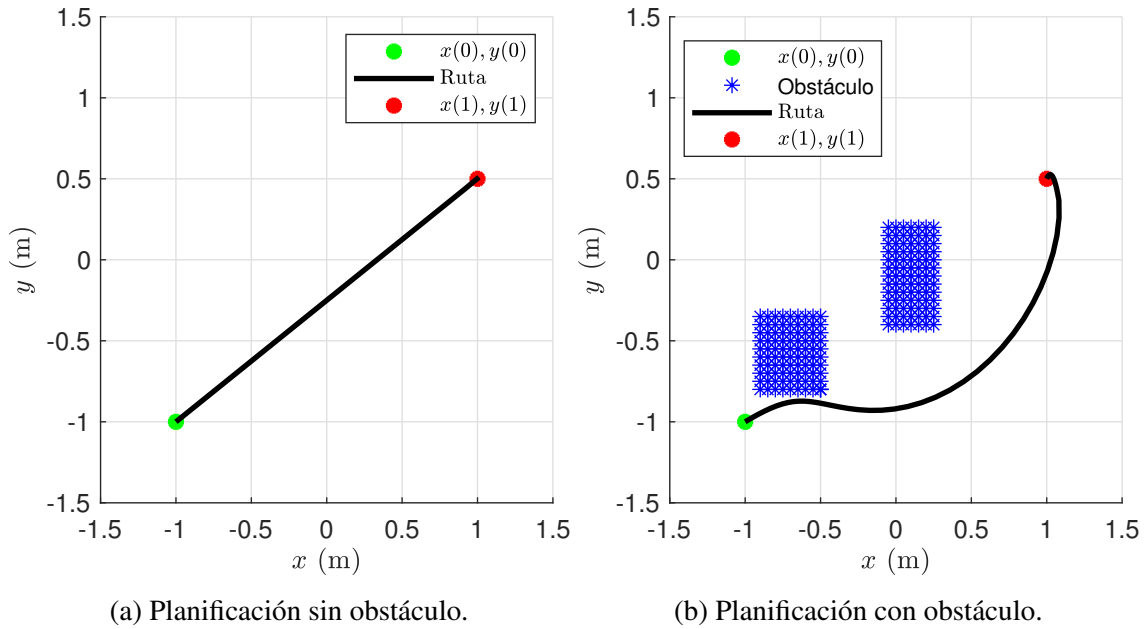


Figura 4-2.: Planificación de trayectorias con el algoritmo de campos potenciales artificiales.

4.5. Implementación del planificador A^*

En la Figura 4-3 se observa la implementación del planificador de trayectorias A^* para ir desde una condición inicial en posición $(x(0), y(0))$ hasta una final $(x(1), y(1))$ evadiendo obstáculos. Este algoritmo encuentra una buena solución con respecto a la ruta de mínima distancia.

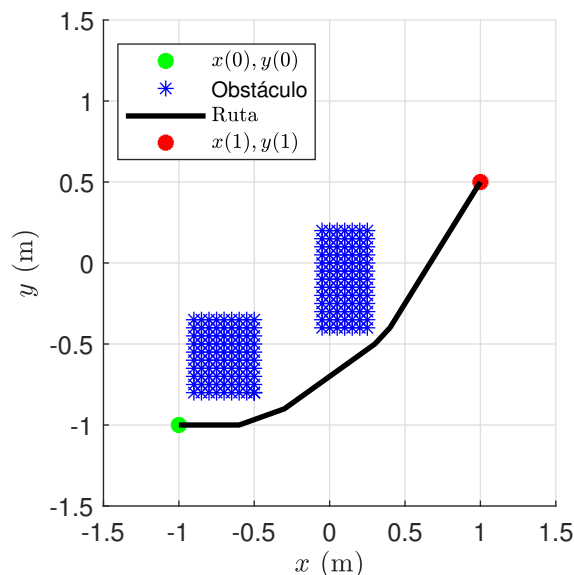


Figura 4-3.: Planificación de trayectorias con el algoritmo A^* .

4.6. Implementación del planificador RRT*

Este algoritmo es una variación del RRT, en donde se garantiza una convergencia en el óptimo a medida que se aumenta la distancia máxima entre conexiones del árbol. En la Figura 4-4 se observa el camino generado por el método RRT* para una distancia máxima de 0.3 m, 0.5 m y 0.8 m, siendo esta última la longitud en la que se presenta la ruta mínima en distancia. Si se sigue aumentando la distancia máxima entre conexiones se presenta el mismo resultado. El experimento se realiza para las mismas condiciones iniciales y finales.

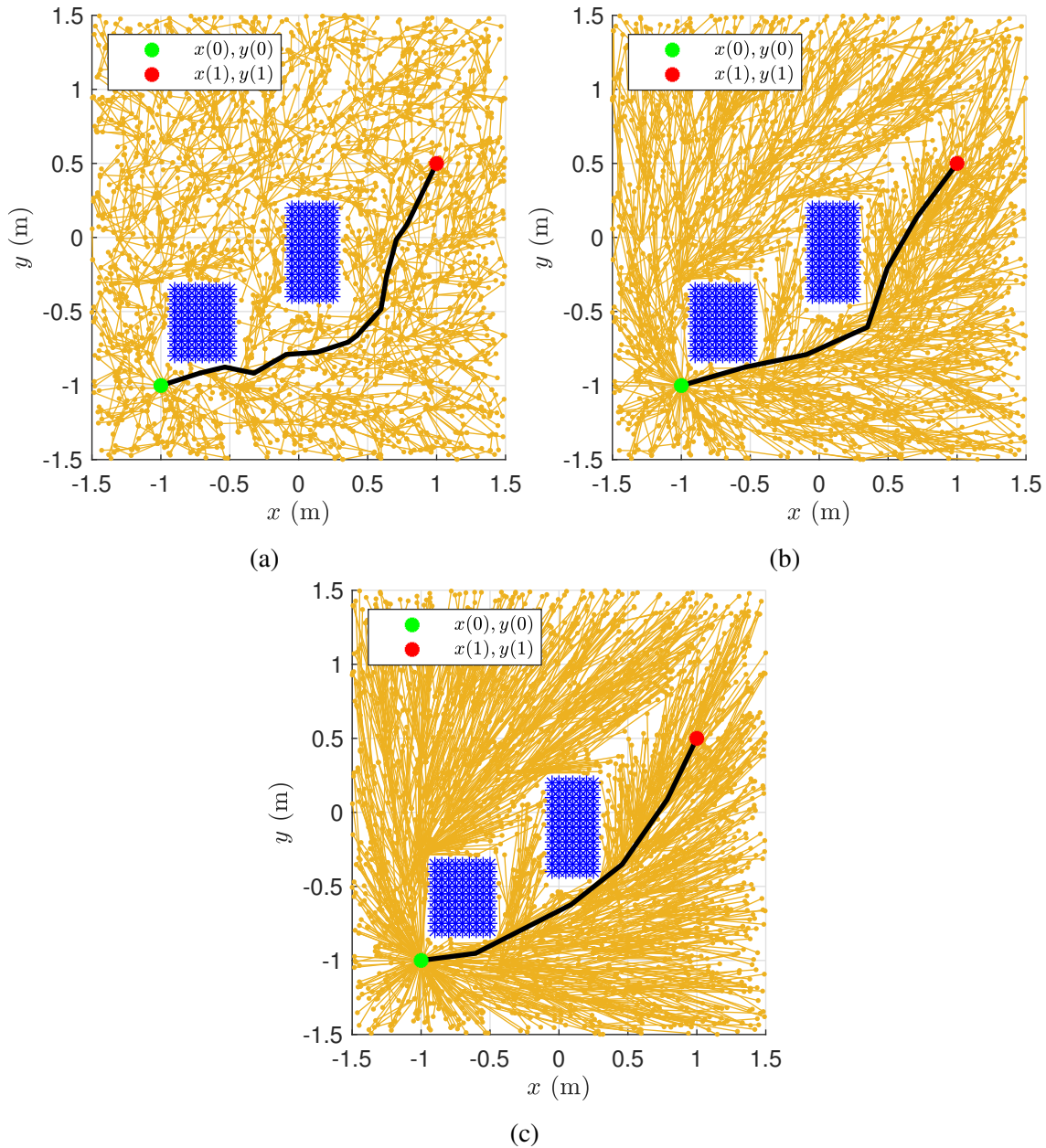


Figura 4-4.: Algoritmo RRT*. (a) distancia de 0.3 m, (b) distancia de 0.5 m, (c) distancia de 0.8 m

4.7. AMC para el DI en 2D

El sistema dinámico que describe el comportamiento de una partícula en dos dimensiones de posición y que tiene dos dimensiones de velocidad se muestra en la Ec. (4-6).

$$\begin{cases} \ddot{x} = u_x \\ \ddot{y} = u_y \end{cases}, \quad (4-6)$$

donde x y y corresponden a las direcciones de movimiento sobre el plano. Las condiciones iniciales de posición y velocidad serán $[x_0, y_0]$ y $[\dot{x}_0, \dot{y}_0]$, y las condiciones finales $[x_1, y_1]$ y (\dot{x}_1, \dot{y}_1) . Para la implementación del AMC sobre el DI, se utiliza el modelo dinámico que determina el comportamiento de la posición y velocidad de la partícula para el movimiento en el plano. Los límites de posición y velocidad son definidos en el espacio de estados de celda. Para mostrar los vecinos de cada celda, se plantea un ejemplo con 9 señales de control diferentes en cada dimensión, por lo que se tiene 81 combinaciones, que logran llevar a la partícula desde una celda inicial de posición y velocidad a 81 vecinos. Cada celda tendrá 81 vecinos de posición y 81 de velocidad, haciendo que una celda tenga 6.561 vecinos en el espacio de estados. A continuación se ilustran los vecinos de una celda, tomando diferentes condiciones iniciales, así como la trayectoria que sigue la partícula.

4.7.1. Vecinos de una celda con $\dot{x}_0 = 0$ y $\dot{y}_0 = 0$

Para encontrar las celdas vecinas en este caso, se toma como condición inicial $[x_0, y_0] = [0, 0]$ m, $[\dot{x}_0, \dot{y}_0] = [0, 0]$ m/s y se analiza la evolución del modelo dinámico durante 1 s, con señales de control en cada dimensión entre $[u_{min}, u_{max}] = [-0.2, 0.2]$ m/s² y un $\Delta u = 0.05$ m/s². En la Figura 4-5a se muestran los 81 vecinos de posición y en la Figura 4-5b se muestran los 81 de velocidad. Para la combinación de señales de control en cero la partícula no incrementa su velocidad manteniendo la misma velocidad inicial. Es importante resaltar que cada celda de la Figura 4-5a tiene 81 posibles velocidades, formado las 6.561 celdas vecinas en el espacio de estados.

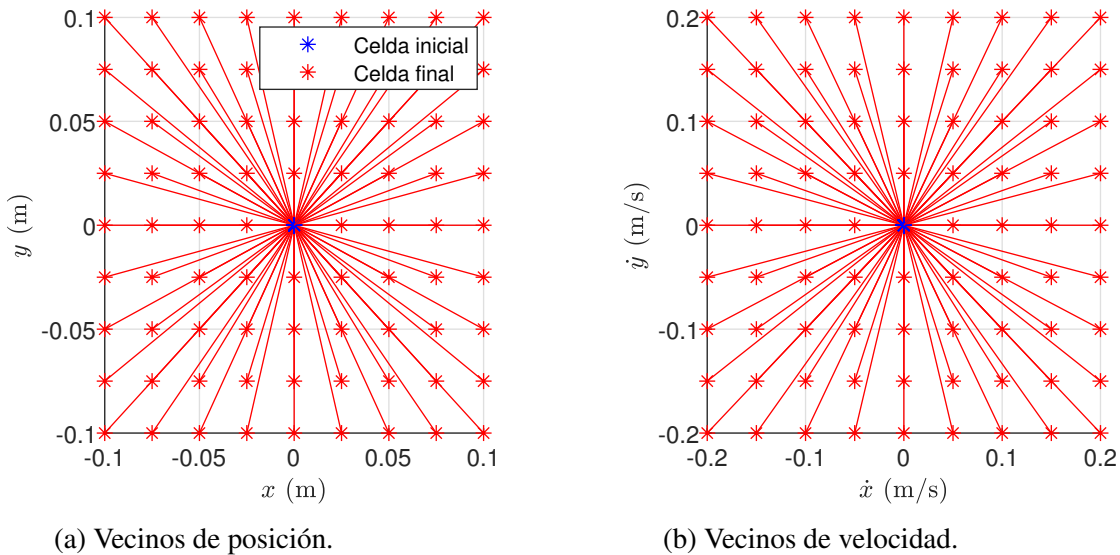


Figura 4-5.: Análisis de celdas vecinas en posición y velocidad para $\dot{x}_0 = 0$ y $\dot{y}_0 = 0$.

4.7.2. Vecinos de una celda con $\dot{x}_0 > 0$ y $\dot{y}_0 > 0$

Para encontrar las celdas vecinas en este caso, se toma como condición inicial $[x_0, y_0] = [0, 0]$ m, $[\dot{x}_0, \dot{y}_0] = (0.1, 0.1)$ m/s y se analiza la evolución del modelo dinámico durante 1 s, con señales de control entre $[u_{min}, u_{max}] = [-0.2, 0.2]$ m/s² que tienen un $\Delta u = 0.05$ m/s². En la Figura 4-6a se muestran los vecinos de posición. En la Figura 4-6b se muestran los de velocidad.

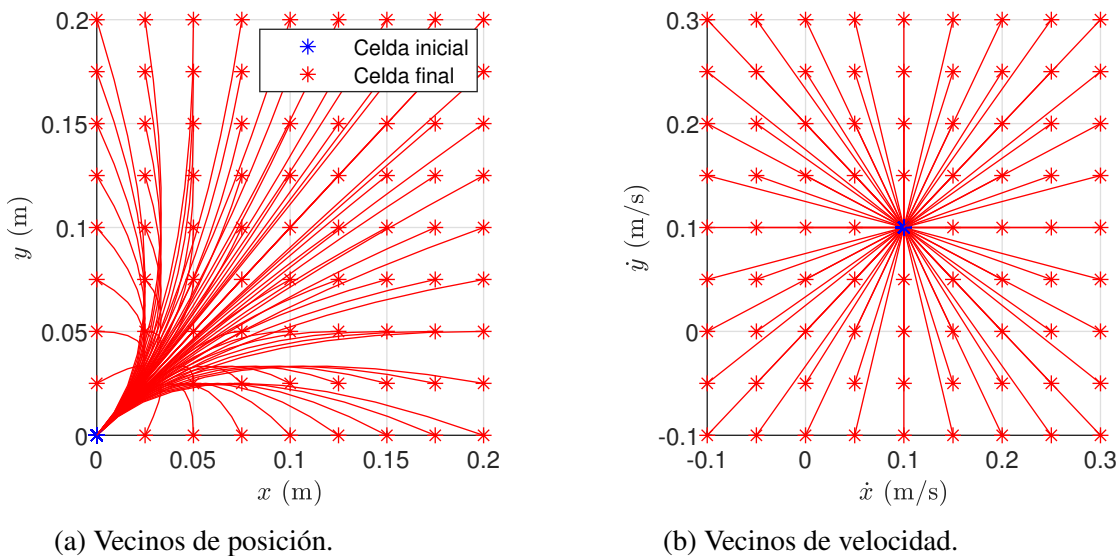


Figura 4-6.: Análisis de celdas vecinas en posición y velocidad para $\dot{x}_0 > 0$ y $\dot{y}_0 > 0$.

4.7.3. Vecinos de una celda con $\dot{x}_0 < 0$ y $\dot{y}_0 < 0$

Para encontrar las celdas vecinas en este caso, se toma como condición inicial $[x_0, y_0] = [0, 0]$ m, $[\dot{x}_0, \dot{y}_0] = [-0.1, -0.1]$ m/s y se analiza la evolución del modelo dinámico durante 1 s, con señales de control entre $[u_{min}, u_{max}] = [-0.2, 0.2]$ m/s² que tienen un $\Delta u = 0.05$ m/s². En la Figura 4-7a se muestran los vecinos de posición. En la Figura 4-7b se muestran los de velocidad.

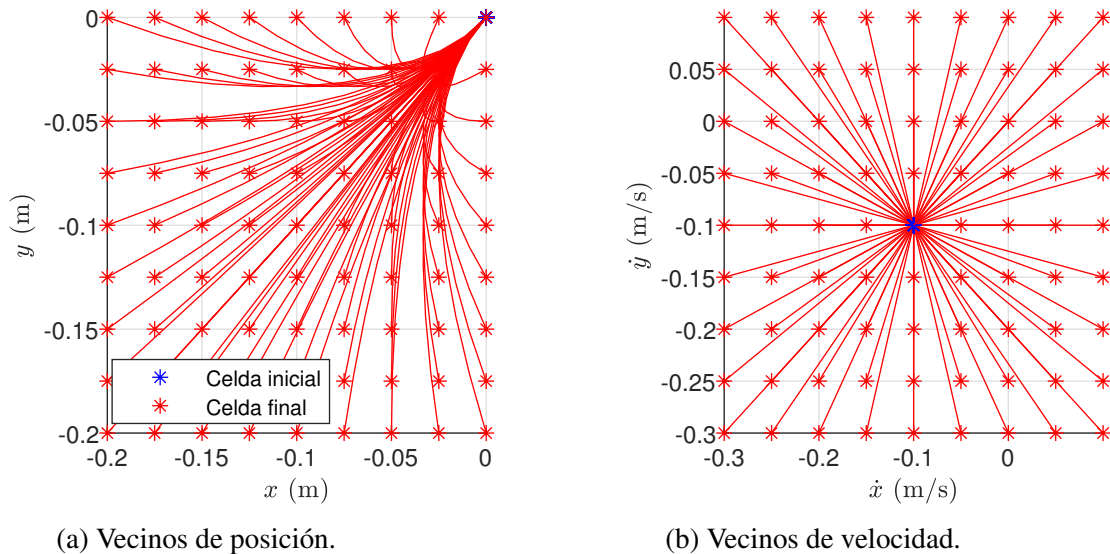


Figura 4-7.: Análisis de celdas vecinas en posición y velocidad para $\dot{x}_0 < 0$ y $\dot{y}_0 < 0$.

4.7.4. Vecinos de una celda con $\dot{x}_0 > 0$ y $\dot{y}_0 < 0$

Para encontrar las celdas vecinas en este caso, se toma como condición inicial $[x_0, y_0] = [0, 0]$ m, $[\dot{x}_0, \dot{y}_0] = [0.1, -0.1]$ m/s y se analiza la evolución del modelo dinámico durante 1 s, con señales de control entre $[u_{min}, u_{max}] = [-0.2, 0.2]$ m/s² que tienen un $\Delta u = 0.05$ m/s². En la Figura 4-8a se muestran los vecinos de posición. En la Figura 4-8b se muestran los de velocidad.

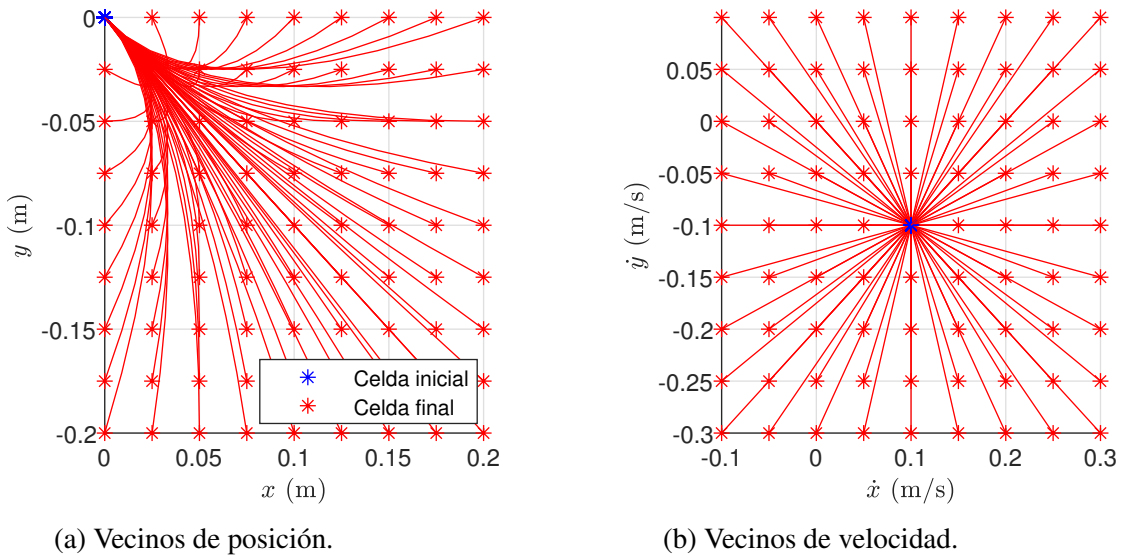


Figura 4-8.: Análisis de celdas vecinas en posición y velocidad para $\dot{x}_0 > 0$ y $\dot{y}_0 < 0$.

En la Figura 4-9 se muestran las señales de control empleadas para la construcción del grafo, éstas son aplicadas durante 1 s y deben estar limitadas por los parámetros de aceleración de la partícula.

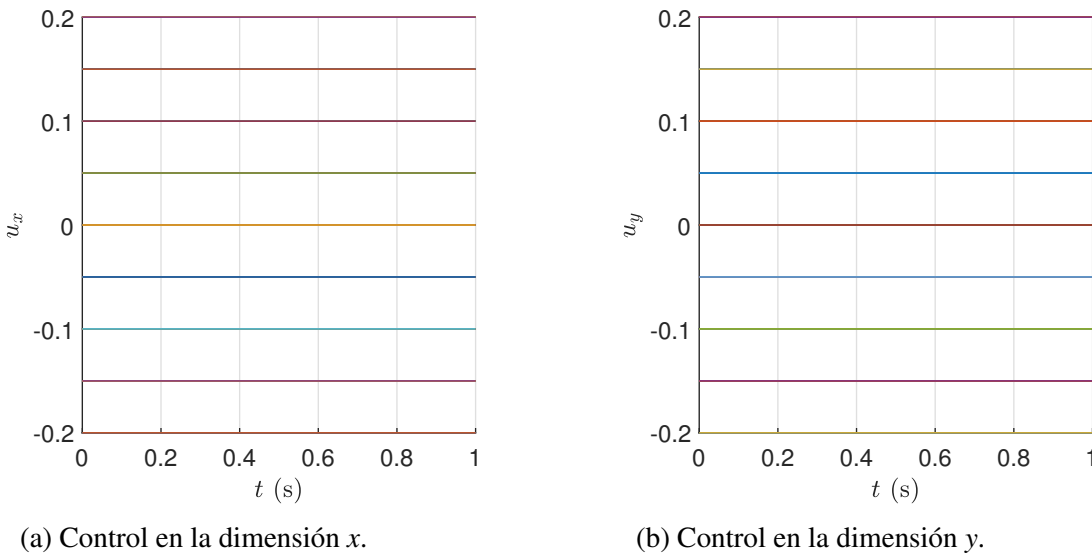


Figura 4-9.: Señales de control.

4.7.5. Construcción del grafo

En los ejemplos anteriores no se tomó en cuenta otra posición inicial, debido a que la geometría era igual en cada celda en comparación con la central $[x, y] = [0, 0]$ m. Ahora, para la construcción del grafo empleado en el planificador de trayectorias del DI se procede a realizar la simulación en un espacio de estados definido por $x = [-1.5, 1.5]$ m y $y = [-1.5, 1.5]$ m, con un $\Delta x = \Delta y = 0.05$ m; $\dot{x} = [-0.5, 0.5]$ m/s y $\dot{y} = [-0.5, 0.5]$ m/s, con un $\Delta \dot{x} = \Delta \dot{y} = 0.1$ m/s y $u_x = [-0.1, 0.1]$ m/s² y $u_y = [-0.1, 0.1]$ m/s², con un $\Delta u = 0.1$ m/s² aplicadas durante un tiempo de 1 s. La discretización fue tomada de esta forma para cada uno de los casos de planificación de trayectorias por medio del AMC. A continuación se emplea el AMC para la generación de ruta con mínima distancia.

4.7.6. Trayectoria con mínima distancia (AMC_d)

Ahora se realiza la implementación del AMC para obtener una trayectoria de referencia que minimice la distancia recorrida desde una posición inicial $[x_0, y_0] = [-1, -1]$ m hasta una final $[x_1, y_1] = [1, 0.5]$ m, con una velocidad inicial y final de $(\dot{x}, \dot{y}) = [0, 0]$ m/s. Se construye la estructura de datos que se muestra en la Figura 4-10a, en esta se indica la cantidad de conexiones entre celdas, representadas por la cantidad de elementos no nulos (nz). La función de costo empleada para la construcción del grafo (G_d) se muestra en la Ec. (4-7) y el problema de optimización se plantea en la Ec. (4-8).

$$F_d = d(p_0, p_{n-1}) = \sum_{i=0}^{n-2} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (4-7)$$

$$\left\{ \begin{array}{l} \text{mín} \quad \sum_{i=0}^{n-2} d(p_i, p_{i+1}), \\ \text{sujeto a:} \quad q_{i+1} = C(q_i, U_i), \quad q_0 = [p_0, \dot{p}_0]^T, \quad q_{n-1} = [p_{n-1}, \dot{p}_{n-1}]^T, \\ \quad \quad \quad U_i = \{u_{x_i}, u_{y_i}\}, \quad u_x, u_y = \{-\bar{u}, 0, \bar{u}\} \end{array} \right. \quad (4-8)$$

donde q es el vector de estados conformado por la posición (p) y velocidad (\dot{p}), p_i es un nodo de la ruta con coordenadas (x_i, y_i) , $d(p_i, p_{i+1})$ es la distancia entre dos nodos adyacentes, p_0 es la posición origen del robot con una velocidad inicial \dot{p}_0 y p_{n-1} son las coordenadas del destino con velocidad final \dot{p}_{n-1} . No obstante las restricciones en las señales de control, el sistema tiene una solución en el espacio de trabajo e infinitas en el espacio de estados; que corresponden a variaciones de velocidad en los nodos intermedios. El robot tardaría $t = (n - 1)t_s$ s en completar cada una de las posibles rutas, y se puede obtener un camino en particular al formular un problema de optimización multiobjetivo (POM). En la Figura 4-10b se muestra la ruta planificada en el plano.

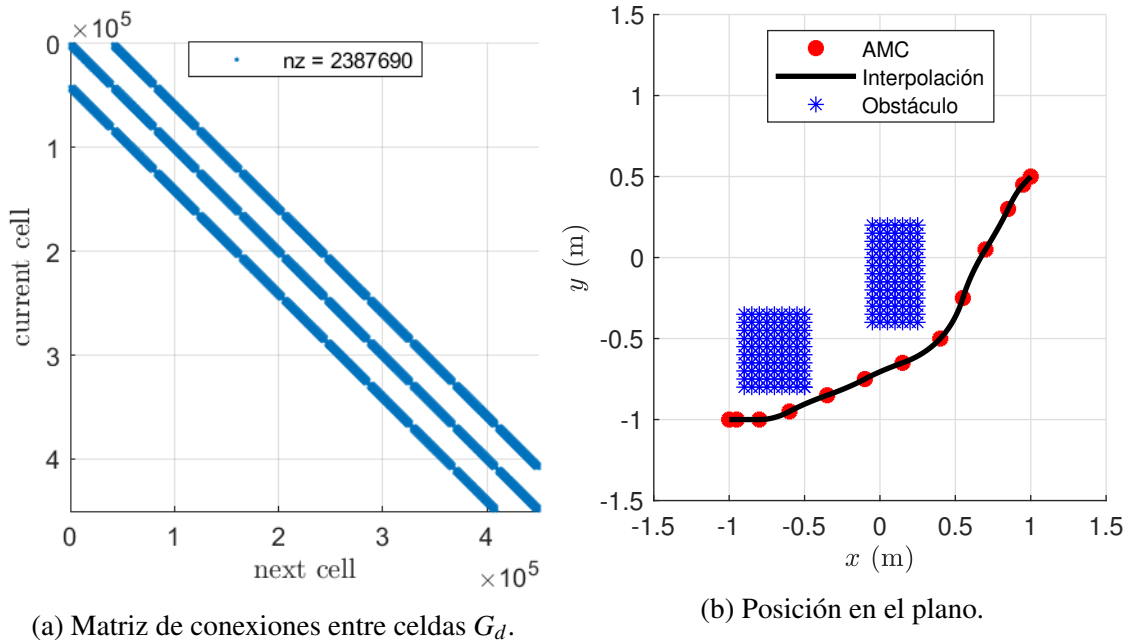


Figura 4-10.: Conexiones entre celda y posición en plano para optimización de distancia.

En la Figura 4-11 se muestran las señales de referencia de posición y en 4-12 las velocidad en el tiempo de la partícula en cada una de las dos dimensiones dadas por el AMC que deben ser seguidas para minimizar la distancia recorrida evadiendo los obstáculos.

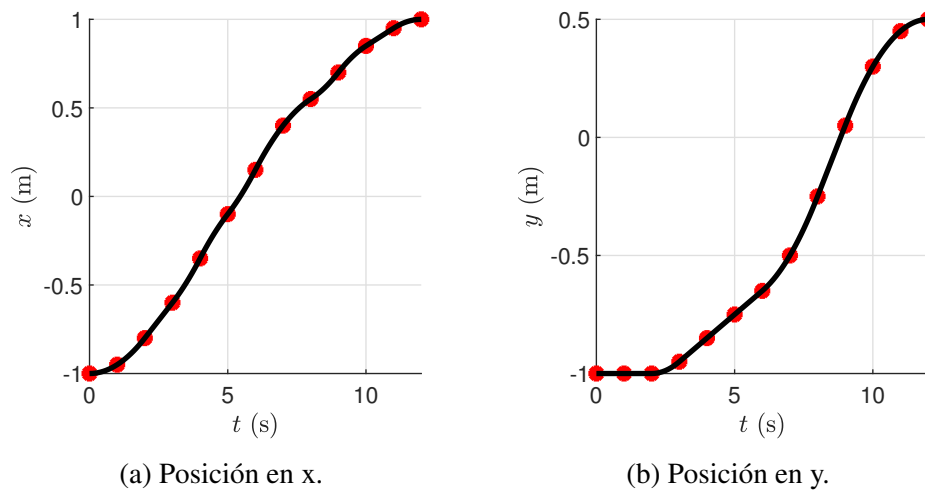


Figura 4-11.: Posición en el tiempo para optimización de distancia.

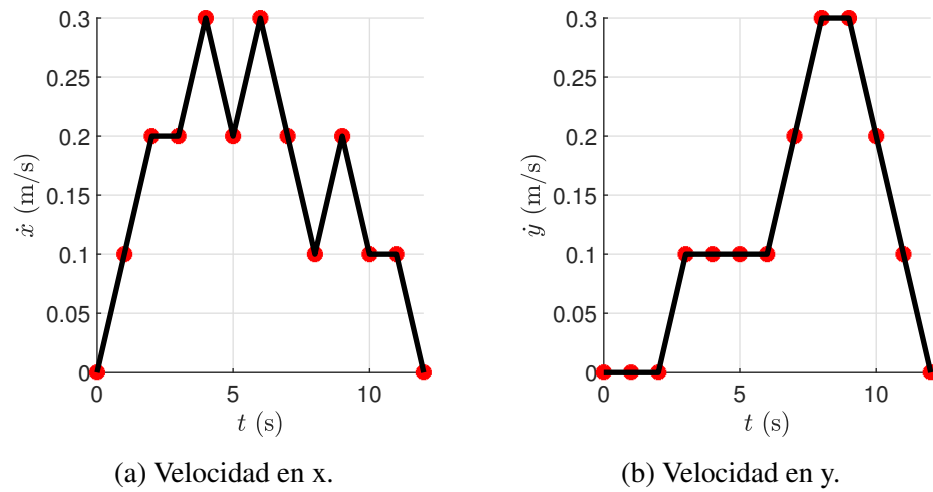


Figura 4-12.: Posición y velocidad en el tiempo para optimización de distancia.

En la Figura 4-13 se muestran las señales de control empleadas en cada dimensión para que la partícula realice el movimiento que minimice la distancia entre las celdas inicial y final.

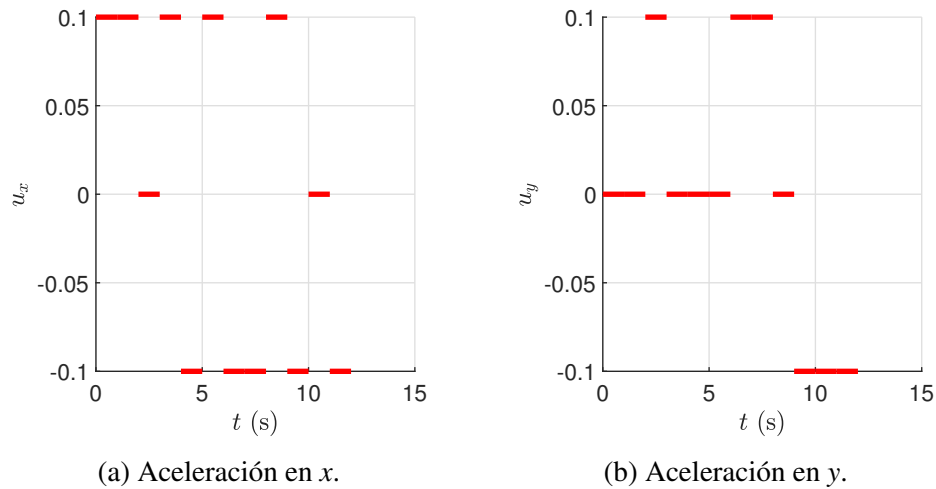


Figura 4-13.: Señales de control para optimización de distancia.

4.8. Análisis de las rutas planificadas

En esta sección se comparan los resultados de los diferentes planificadores de trayectorias implementados en este capítulo y el AMC propuesto para solucionar el problema de mínima distancia. En la Figura 4-14 se observan las trayectorias generadas por los planificadores A^* , CPA, RRT^* y AMC_d . En esta se puede apreciar que la ruta con mayor distancia es la del algoritmo CPA y la menor es la generada por el algoritmo RRT^* .

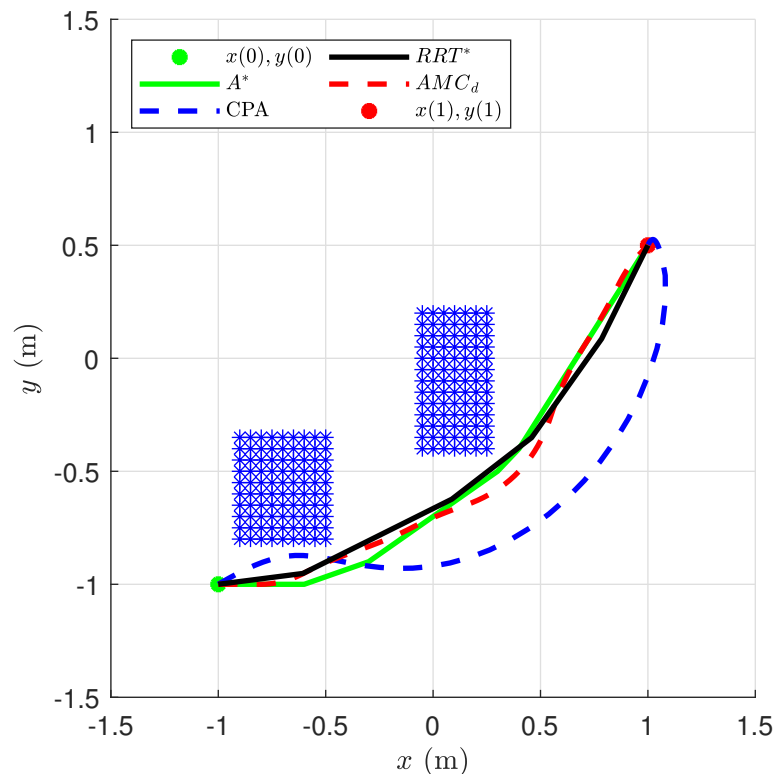


Figura 4-14.: Rutas generadas.

En la Tabla 4-1 se observa la comparación en distancia de las rutas generadas en la Figura 4-14, así como otros dos casos vistos en el algoritmo RRT^* . En esta tabla se detalla que el algoritmo de planificación por CPA entrega una trayectoria suave pero con la mayor distancia, mientras que el algoritmo A^* genera un camino más corto cercano a la de menor longitud. El algoritmo RRT^* entrega la ruta con menor distancia cuando se aumenta la longitud máxima entre las conexiones del árbol a 0.8 m. Otra característica importante de estos algoritmos, es que solo entregan referencias de posición, haciendo imposible estimar el gasto energético, esfuerzo de control, tiempo de navegación, entre otros. Por otro lado el AMC_d genera una distancia cercana a la mínima longitud y además proporciona los perfiles de velocidad que debe seguir el sistema.

Tabla 4-1.: Comparación en distancia de las rutas generadas.

| Ruta | Planificador | Distancia [m] |
|------|------------------|---------------|
| 1 | CPA | 3.0634 |
| 2 | A* | 2.6604 |
| 3 | RRT* con 0.3m | 2.8300 |
| 4 | RRT* con 0.5m | 2.7100 |
| 5 | RRT* con 0.8m | 2.6400 |
| 6 | AMC _d | 2.6847 |

4.9. Conclusión

En este capítulo se mostraron algunos de los algoritmos de planificación de trayectorias existentes en la literatura y se implementaron algunos de ellos, con el fin de comparar los resultados con el AMC_d propuesto. Los algoritmos implementados tienen la característica de entregar caminos de referencia solo en posición, sin tener en cuenta los parámetros de velocidad y aceleración. El método de Campos Potenciales Artificiales construye rutas suaves, mientras que el A* genera un camino más corto (la exactitud depende de la complejidad de la función heurística empleada) y el método de RRT* converge en la mínima distancia cuando se incrementa la longitud máxima entre las conexiones del árbol. En el siguiente capítulo se muestra el desarrollo del AMC para la planificación de trayectorias del DI, solucionando los problemas de optimización discreta diferentes a la distancia, así como el POM.

5. Planificación de trayectorias óptimas para el DI en 2D empleando el AMC

En este capítulo se muestra la aplicación del AMC en la generación de rutas de referencia sobre el plano XY para el movimiento del DI. Este análisis se realiza partiendo del desarrollo del capítulo 3, donde se menciona que el mapeo de celdas puede ser empleado para la solución numérica de problemas de optimización, tales como: tiempo de navegación de una partícula, esfuerzo de control y gasto energético. Ahora se expande el concepto al DI que se desplaza en el plano como se mostró al final del Capítulo 4.

5.1. AMC para la planificación de trayectorias del DI en 2D

El DI es representado por medio del sistema dinámico dado en la Ec. (4-6), que describe el comportamiento de una partícula en dos dimensiones de posición. A continuación se analizan los siguientes problemas de optimización:

5.1.1. Generación de trayectoria para minimizar el esfuerzo de control (AMC_e)

Ahora se desea obtener una trayectoria de referencia que presente el menor esfuerzo de control para llevar al robot desde una celda inicial con coordenadas $[x_0, y_0] = [-1, -1]$ m hasta una final $[x_1, y_1] = [1, 0.5]$ m, manteniendo las velocidades inicial y final en cero.

Si se implementa el algoritmo de Dijkstra para encontrar el camino con menor esfuerzo de control se obtiene la mostrada en la Figura 5-1b. Como se puede apreciar, la trayectoria generada por el AMC evita los obstáculos contenidos en el espacio de trabajo y une las celdas inicial y final, minimizando el esfuerzo de control.

Se construye el grafo (G_u) que se muestra en la Figura 5-1a, con la función de costo de la Ec. (5-1) y el problema de optimización se plantea en la Ec. (5-2).

$$F_u = \frac{1}{2} \sum_{i=0}^{n-2} u_i^2 \quad (5-1)$$

$$\left\{ \begin{array}{l} \text{mín} \quad \frac{1}{2} \sum_{i=0}^{n-2} u_i^2, \\ \text{sujeto a:} \quad q_{i+1} = C(q_i, U_i), \quad q_0 = [p_0, \dot{p}_0]^T, \quad q_{n-1} = [p_{n-1}, \dot{p}_{n-1}]^T, \\ \quad \quad \quad U_i = \{u_{xi}, u_{yi}\}, \quad u_x, u_y = \{-\bar{u}, 0, \bar{u}\} \end{array} \right. \quad (5-2)$$

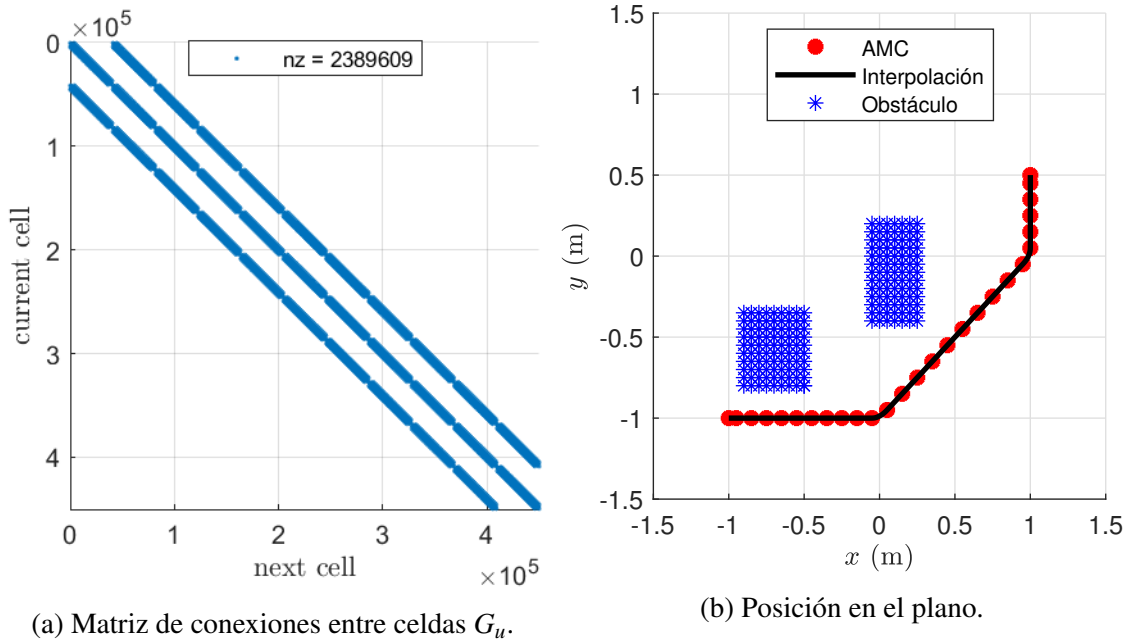


Figura 5-1.: Conexiones entre celda y posición en plano para optimización de esfuerzo de control.

En la Figura 5-2 se muestran las señales de referencia de posición y en 5-3 las de velocidad en el tiempo de la partícula en cada una de las dos dimensiones dadas por el AMC que deben ser seguidas para minimizar el esfuerzo de control evadiendo los obstáculos.

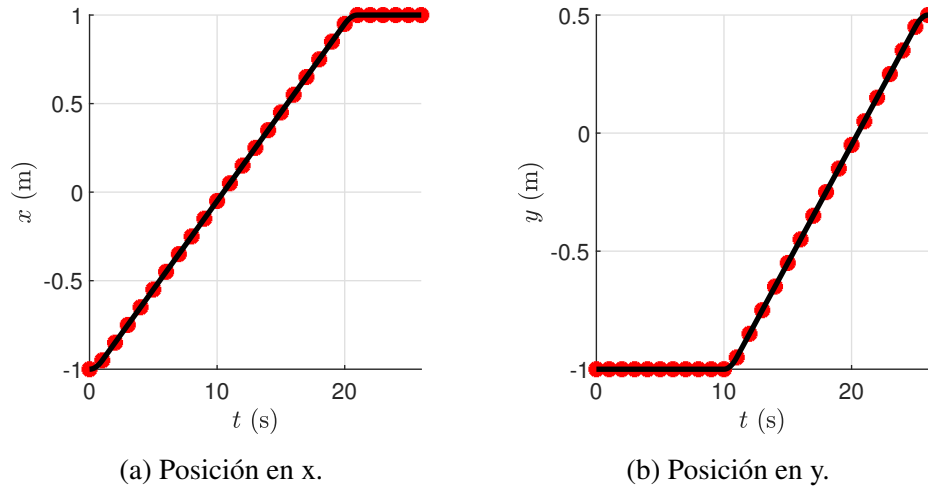


Figura 5-2.: Posición en el tiempo para optimización de esfuerzo de control.

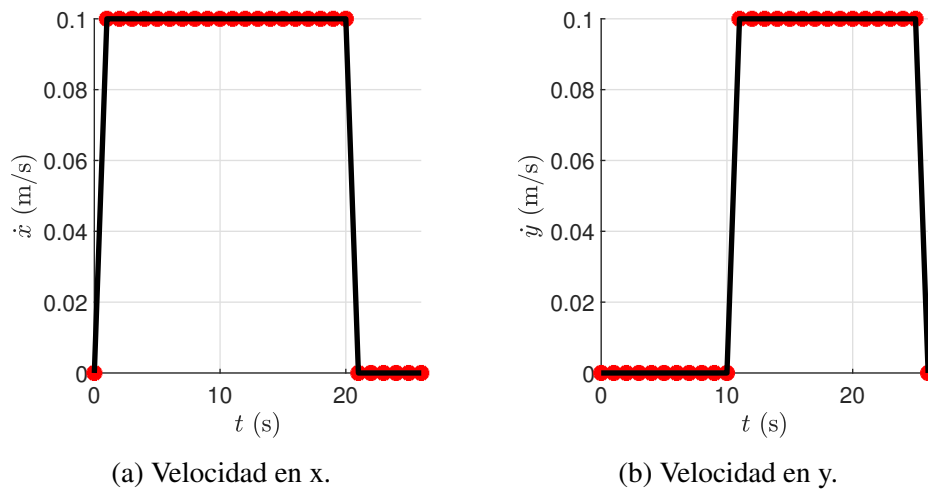


Figura 5-3.: Posición y velocidad en el tiempo para optimización de esfuerzo de control.

En la Figura 5-4 se muestran las señales de control aplicadas en cada dimensión.

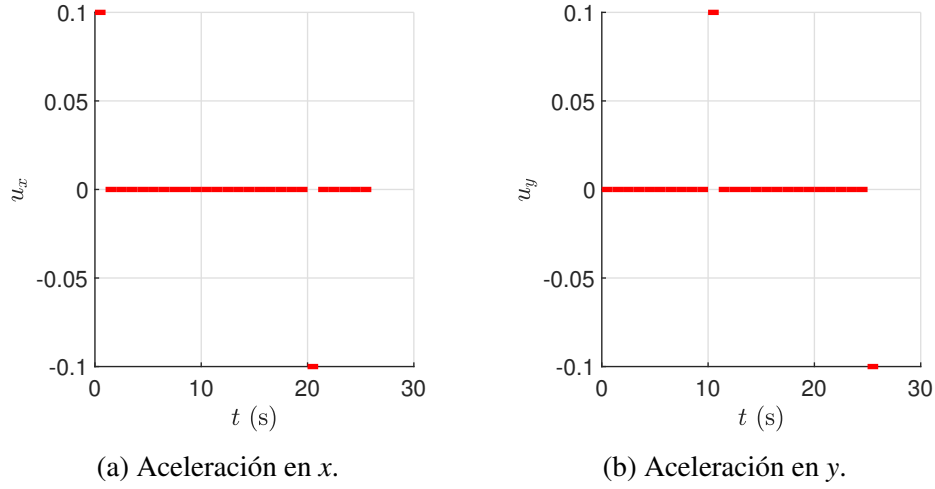


Figura 5-4.: Señales de control para optimización de esfuerzo de control.

5.1.2. Generación de trayectoria con menor tiempo de navegación (AMC_t)

Tomando las mismas condiciones iniciales y finales de los problemas anteriores, se realiza la implementación del AMC para de obtener una trayectoria de referencia que minimice el tiempo de navegación de un robot móvil entre dos celdas del espacio de estados.

Como se busca minimizar el tiempo total de navegación dentro de la discretización adoptada, la función de costo empleada para la construcción del grafo (G_t) se muestra en la Ec. (5-3) y el problema de optimización se plantea en la Ec. (5-4).

$$F_t = \sum_{i=0}^{n-2} t_s \quad (5-3)$$

$$\left\{ \begin{array}{l} \text{mín} \quad \sum_{i=0}^{n-2} t_s, \\ \text{sujeto a:} \quad q_{i+1} = C(q_i, U_i), \quad q_0 = [p_0, \dot{p}_0]^T, \quad q_{n-1} = [p_{n-1}, \dot{p}_{n-1}]^T, \\ \quad \quad \quad U_i = \{u_{xi}, u_{yi}\}, \quad u_x, u_y = \{-\bar{u}, 0, \bar{u}\} \end{array} \right. \quad (5-4)$$

En la Figura 5-5b se muestra la ruta planificada para la minimización del tiempo de navegación del robot.

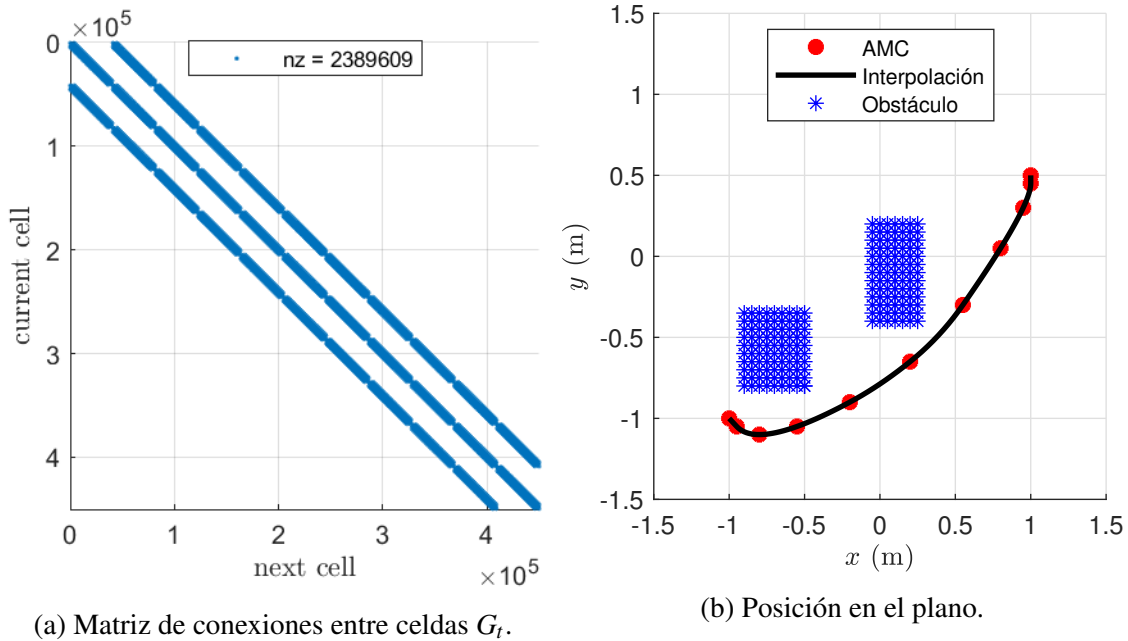


Figura 5-5.: Conexiones entre celda y posición en plano para menor tiempo de navegación.

En la Figura 5-6 se muestran las señales de referencia de posición y en 5-7 las de velocidad en el tiempo del DI en cada una de las dos dimensiones dadas por el AMC. Estas deben ser seguidas para minimizar el tiempo de navegación evadiendo los obstáculos.

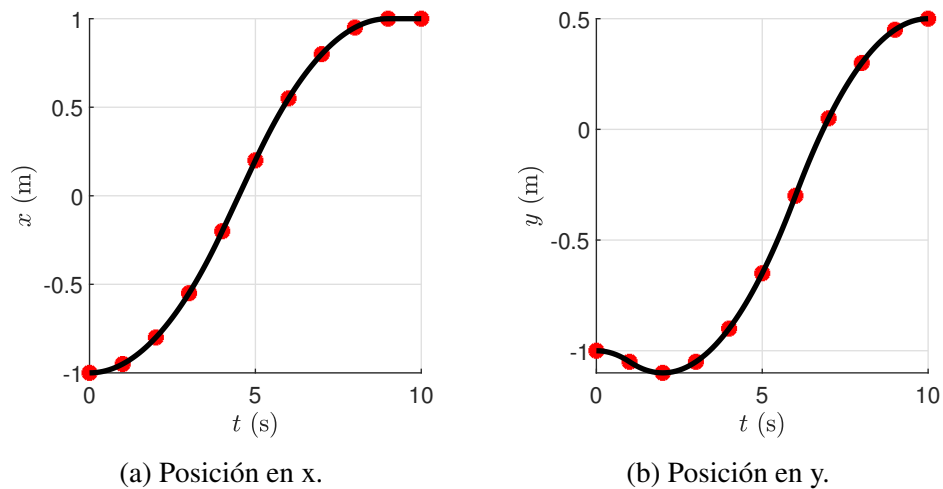


Figura 5-6.: Posición en el tiempo para menor tiempo de navegación.

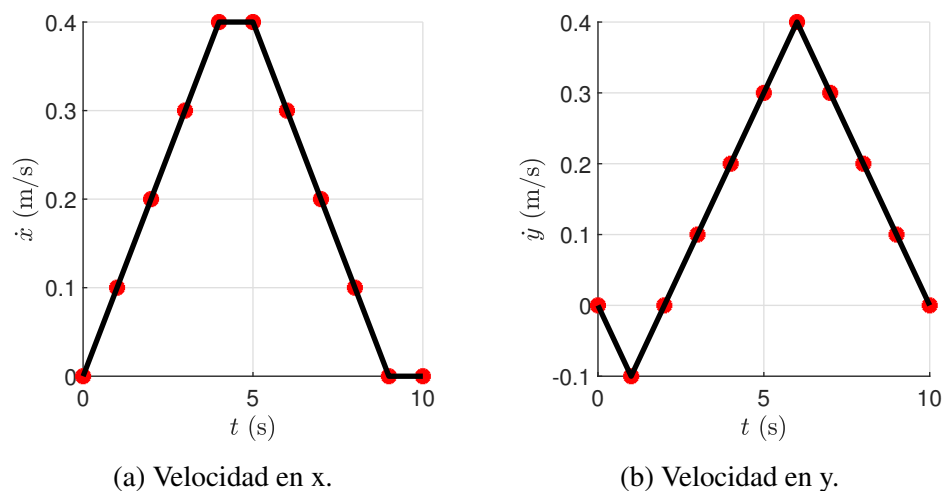


Figura 5-7.: Velocidad en el tiempo para menor tiempo de navegación.

En la Figura 5-8 se muestran las señales de control aplicadas en cada dimensión.

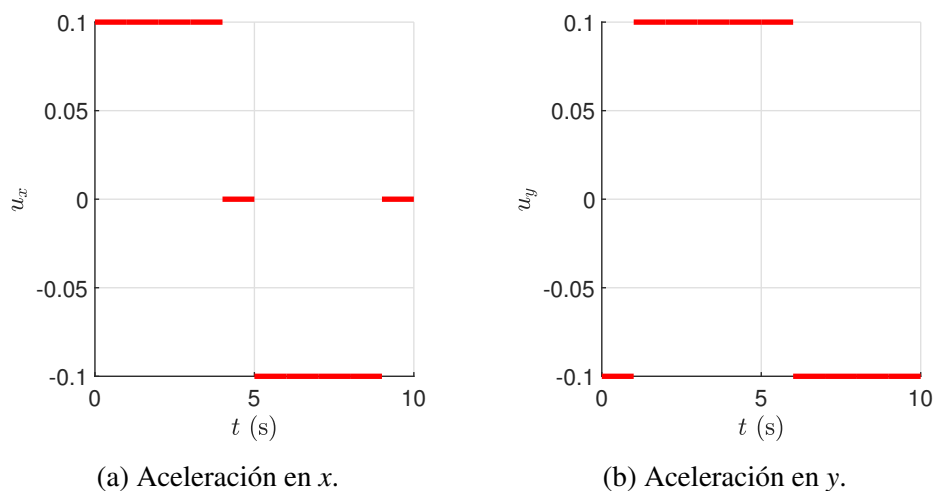


Figura 5-8.: Señales de control para menor tiempo de navegación.

5.1.3. Generación de trayectoria con POM de distancia y tiempo (AMC_{dt})

Ahora se busca encontrar una trayectoria que minimice una función de costo que contenga dos parámetros a optimizar, este caso se denomina problema de optimización multi-objetivo (POM). Para solucionar el POM se toman las mismas condiciones iniciales y finales de los ejemplos anteriores. La función objetivo mostrada en la Ec. (5-5) es determinada arbitrariamente, en esta función se observan parámetros de distancia y tiempo de navegación. Si se emplean las funciones de costo (4-7) y (5-3), se puede construir el problema de optimización como se muestra en la Ec. (5-6).

$$F_{POM} = 0.7F_d + 0.3F_t \quad (5-5)$$

$$\left\{ \begin{array}{l} \text{mín} \quad \sum_{i=0}^{n-2} [0.7 d(p_i, p_{i+1}) + 0.3 t_s], \\ \text{sujeto a: } q_{i+1} = C(q_i, U_i), \quad q_0 = [p_0, \dot{p}_0]^T, \quad q_{n-1} = [p_{n-1}, \dot{p}_{n-1}]^T, \\ \quad \quad U_i = \{u_{xi}, u_{yi}\}, \quad u_x, u_y = \{-\bar{u}, 0, \bar{u}\} \end{array} \right. \quad (5-6)$$

En la Figura 5-9 se muestra la ruta planificada para la minimización de la función de costo (5-5). Como se puede apreciar la ruta de referencia generada por el AMC evita los obstáculos contenidos en el espacio de trabajo y une el punto inicial y final.

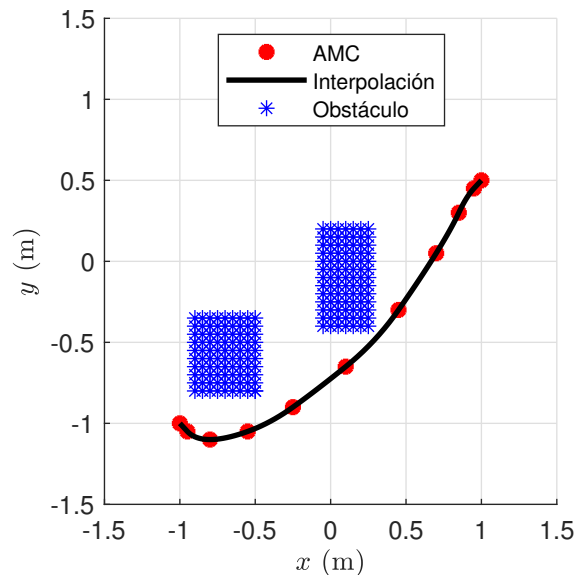


Figura 5-9.: Posición en el plano para optimización de distancia y tiempo.

En la Figura 5-10 se muestran las señales de referencia de posición y en la Figura 5-11 velocidad en el tiempo de la partícula en cada una de las dos dimensiones dadas por el AMC que deben ser seguidas para minimizar la distancia recorrida y el tiempo de navegación evadiendo los obstáculos.

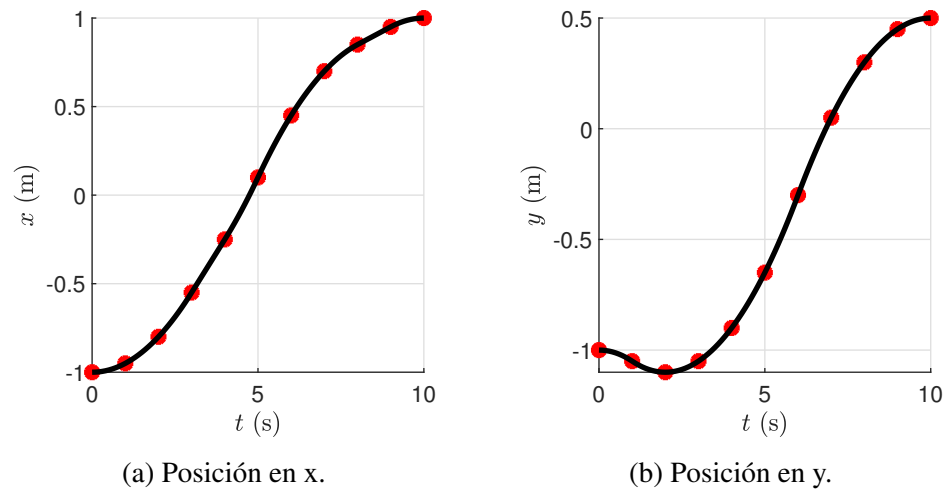


Figura 5-10.: Posición en el tiempo para optimización de distancia y tiempo.

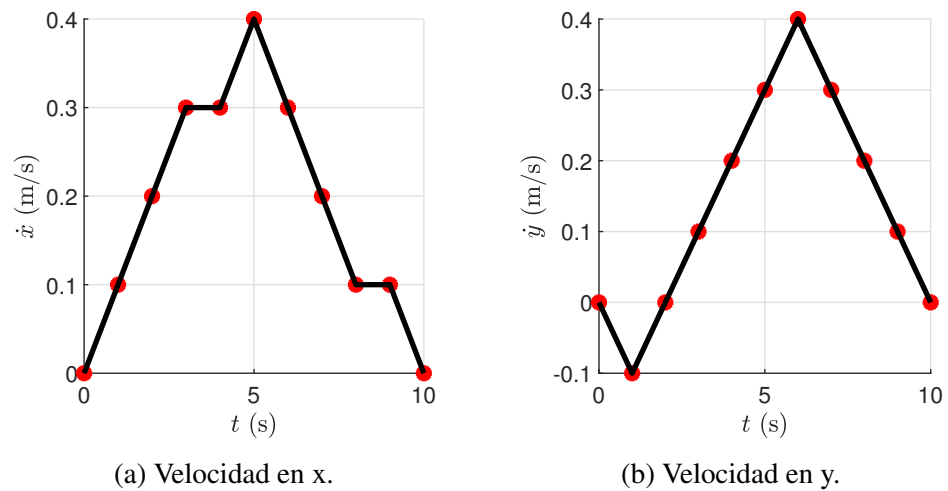


Figura 5-11.: Velocidad en el tiempo para optimización de distancia y tiempo.

En la Figura 5-12 se muestran las señales de control de cada dimensión.

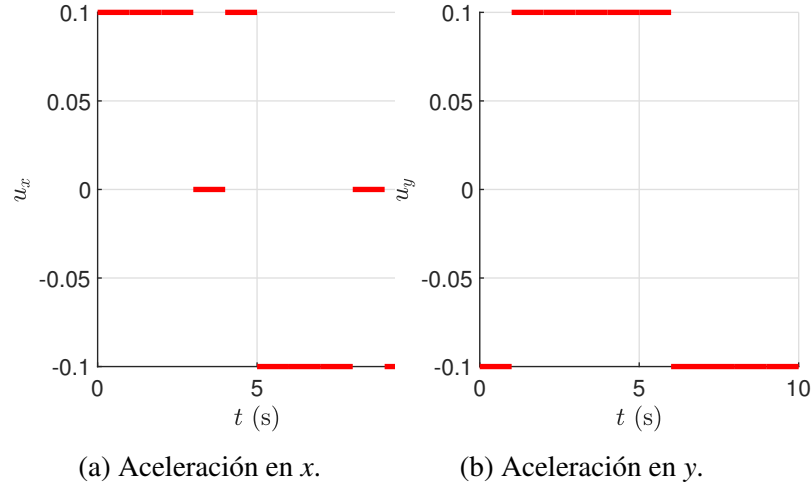


Figura 5-12.: Señales de control para optimización de distancia y tiempo.

5.1.4. Generación de trayectoria con POM de distancia y esfuerzo de control (AMC_{de})

Ahora se busca encontrar una trayectoria que minimice una función de costo que contenga dos parámetros a optimizar, este caso se denomina problema de optimización multi-objetivo (POM). Para solucionar el POM se toman las mismas condiciones iniciales y finales de los ejemplos anteriores. La función objetivo mostrada en la Ec. (5-7) es determinada arbitrariamente, en ésta se observan parámetros de distancia y esfuerzo de control. Si se emplean las funciones de costo (4-7) y (5-1), se puede construir el problema de optimización como se muestra en la Ec. (5-8).

$$F_{POM} = 0.7F_d + 0.3F_u \quad (5-7)$$

$$\left\{ \begin{array}{l} \text{mín} \quad \sum_{i=0}^{n-2} \left[0.7 d(p_i, p_{i+1}) + \frac{0.3}{2} u_i^2 \right], \\ \text{sujeto a:} \quad q_{i+1} = C(q_i, U_i), \quad q_0 = [p_0, \dot{p}_0]^T, \quad q_{n-1} = [p_{n-1}, \dot{p}_{n-1}]^T, \\ \quad \quad \quad U_i = \{u_{xi}, u_{yi}\}, \quad u_x, u_y = \{-\bar{u}, 0, \bar{u}\} \end{array} \right. \quad (5-8)$$

En la Figura 5-13 se muestra la ruta planificada para la minimización de la función de costo (5-7). Como se puede apreciar la ruta de referencia generada por el AMC evita los obstáculos contenidos en el espacio de trabajo y une el punto inicial y final.

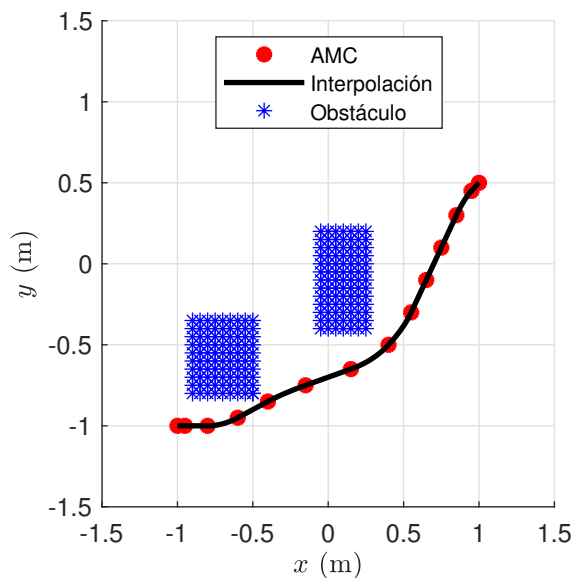


Figura 5-13.: Posición en el plano para optimización de distancia y esfuerzo de control.

En la Figura 5-14 se muestran las señales de referencia de posición y en la Figura 5-15 las de velocidad en el tiempo de la partícula en cada una de las dos dimensiones dadas por el AMC que deben ser seguidas para minimizar la distancia recorrida y el esfuerzo de control evadiendo obstáculos.

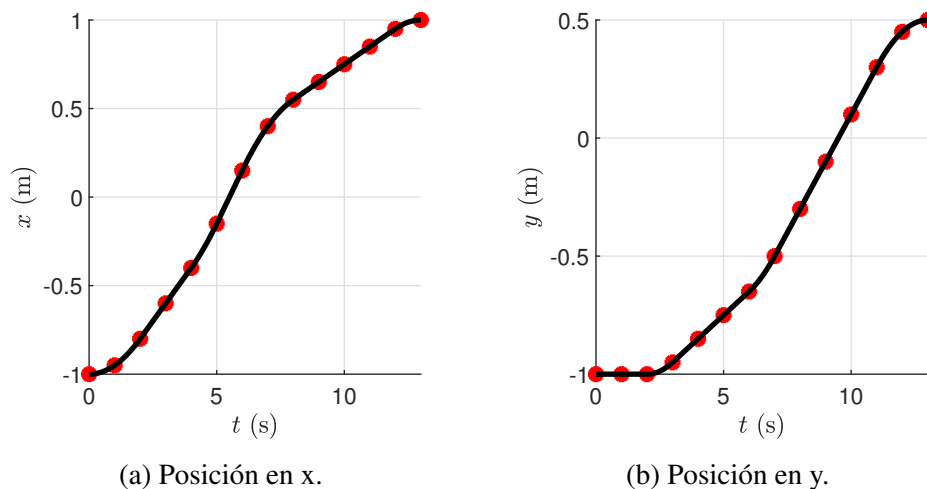


Figura 5-14.: Posición en el tiempo para optimización de distancia y esfuerzo de control.

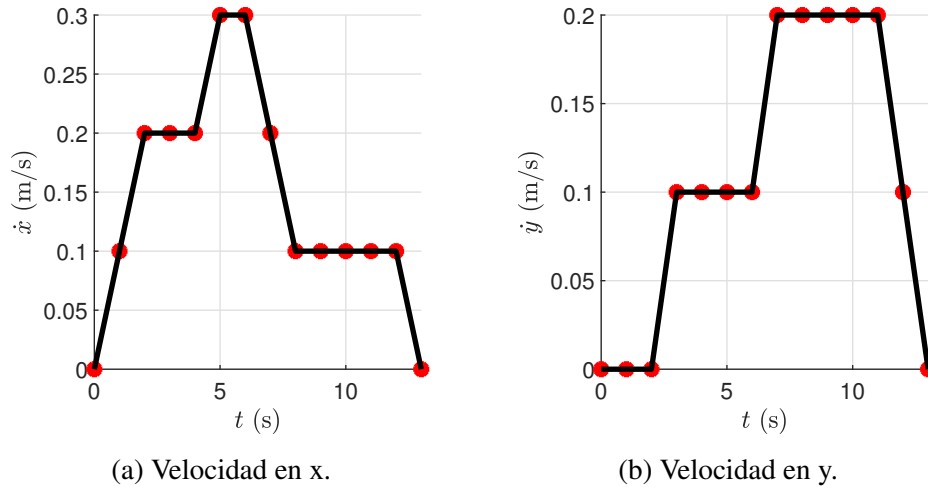


Figura 5-15.: Velocidad en el tiempo para optimización de distancia y esfuerzo de control.

En la Figura 5-16 se muestran las señales de control de cada dimensión.

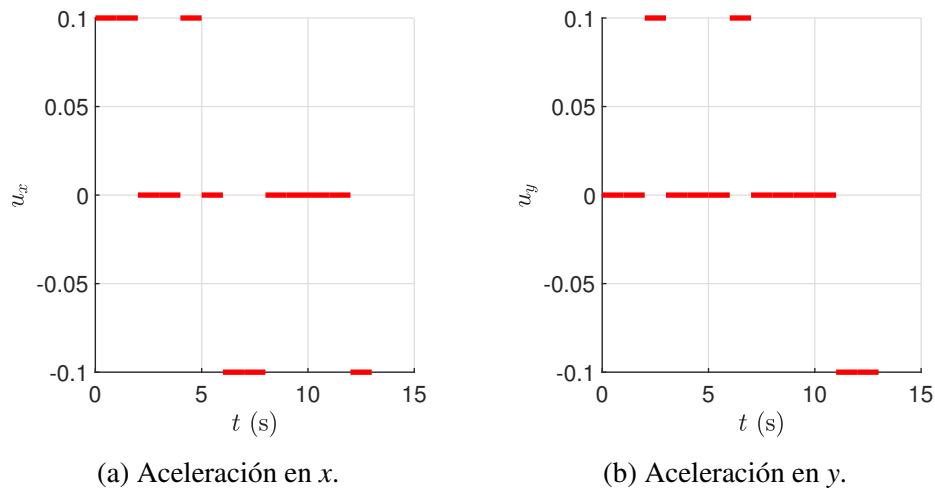


Figura 5-16.: Señales de control para optimización de distancia y esfuerzo de control.

5.1.5. Generación de trayectoria con POM de tiempo, esfuerzo de control y distancia (AMC_{POM})

Ahora se busca encontrar una trayectoria que minimice una función de costo que contenga varios parámetros a optimizar. Para solucionar el POM se toman las mismas condiciones iniciales y finales de los ejemplos anteriores. La función objetivo mostrada en la Ec. (5-9) es determinada arbitrariamente, en esta función se observan parámetros de distancia, esfuerzo de control y tiempo de navegación. Si se emplean las funciones de costo (4-7), (5-1) y (5-3), se puede construir el problema de optimización como se muestra en la Ec. (5-10).

$$F_{POM} = 0.78F_d + 0.2F_u + 0.02F_t \quad (5-9)$$

$$\left\{ \begin{array}{l} \text{mín} \quad \sum_{i=0}^{n-2} [0.78 d(p_i, p_{i+1}) + \frac{0.2}{2} u_i^2 + 0.02 t_s], \\ \text{sujeto a: } q_{i+1} = C(q_i, U_i), \quad q_0 = [p_0, \dot{p}_0]^T, \quad q_{n-1} = [p_{n-1}, \dot{p}_{n-1}]^T, \\ U_i = \{u_{x_i}, u_{y_i}\}, \quad u_x, u_y = \{-\bar{u}, 0, \bar{u}\} \end{array} \right. \quad (5-10)$$

En la Figura 5-17 se muestra la ruta planificada para la minimización de la función de costo (5-9). Como se puede apreciar la ruta de referencia generada por el AMC evita los obstáculos contenidos en el espacio de trabajo y une el punto inicial y final.

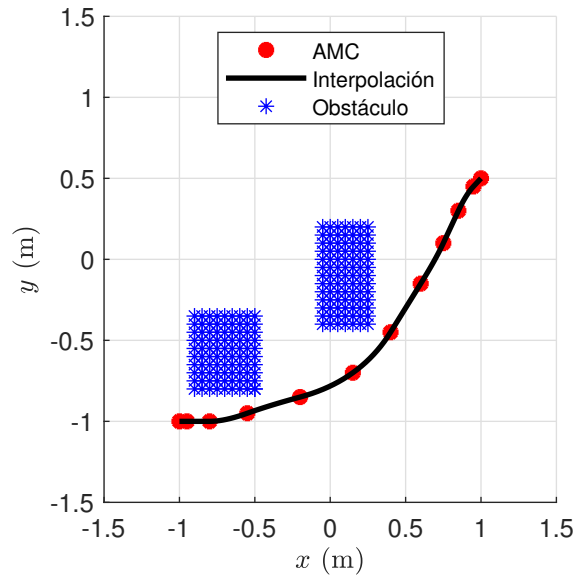


Figura 5-17.: Posición en el plano para optimización de tiempo, esfuerzo de control y distancia.

En la Figura 5-18 se muestran las señales de referencia de posición y en la Figura 5-19 las de velocidad en el tiempo de la partícula en cada una de las dos dimensiones dadas por el AMC que deben ser seguidas para minimizar el tiempo de navegación, el esfuerzo de control y la distancia recorrida evadiendo obstáculos.

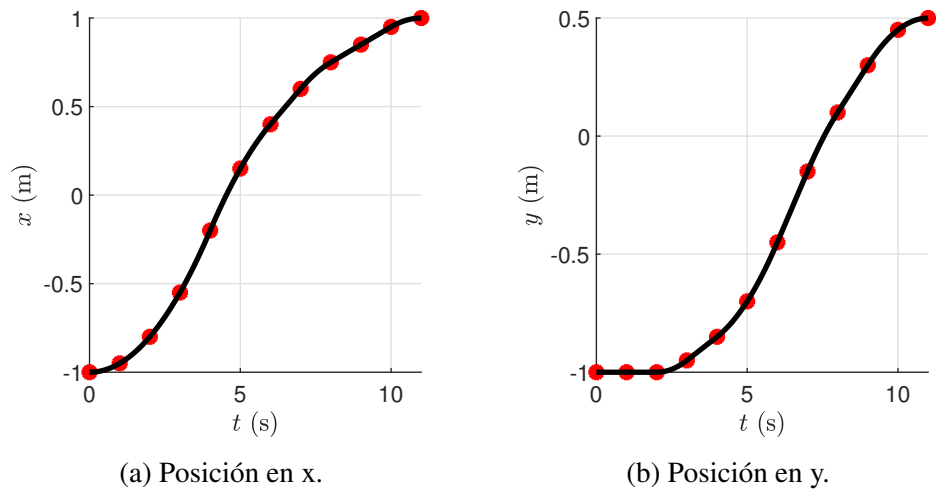


Figura 5-18.: Posición en el tiempo para optimización de tiempo, esfuerzo de control y distancia.

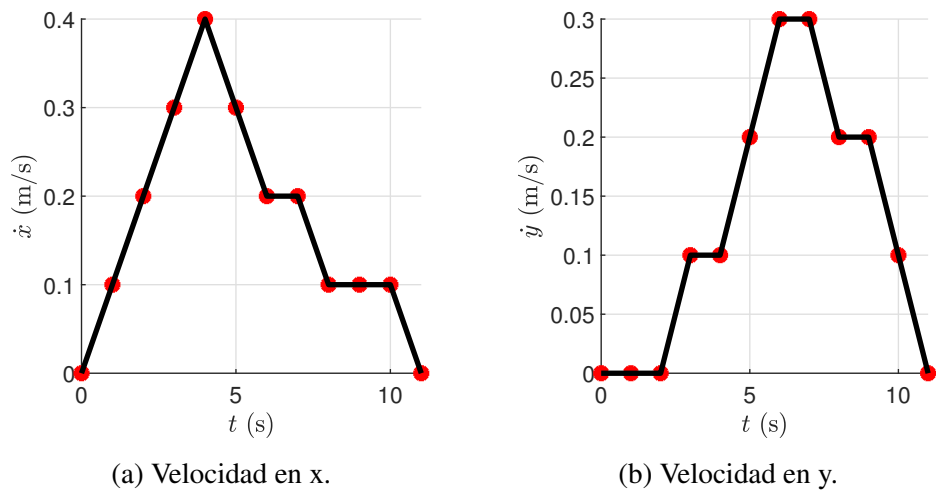


Figura 5-19.: Velocidad en el tiempo para optimización de tiempo, esfuerzo de control y distancia.

En la Figura 5-20 se muestran las señales de control de cada dimensión para el POM.

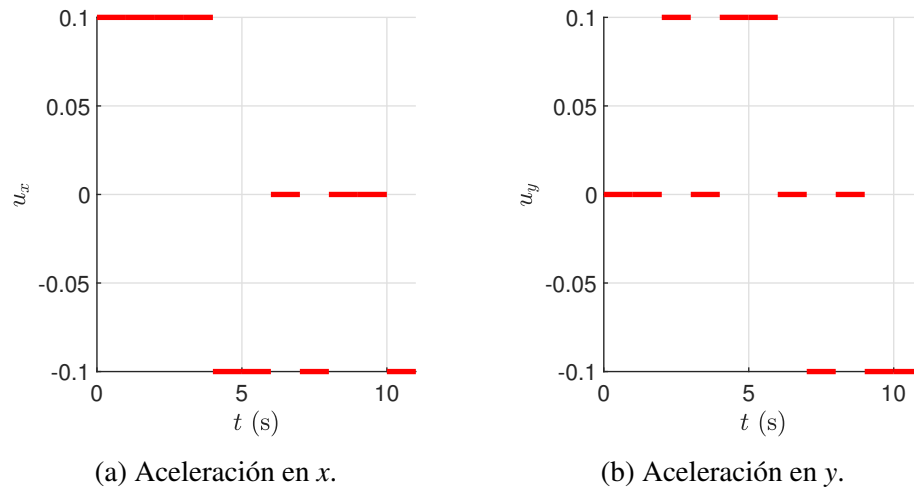


Figura 5-20.: Señales de control para optimización de tiempo, esfuerzo de control y distancia.

5.1.6. Análisis de las diferentes trayectorias generadas por el AMC

Para analizar el desempeño del método planteado por medio del AMC, se realiza una comparación entre cada uno de los resultados, los cuales se presentan en la Figura 5-21.

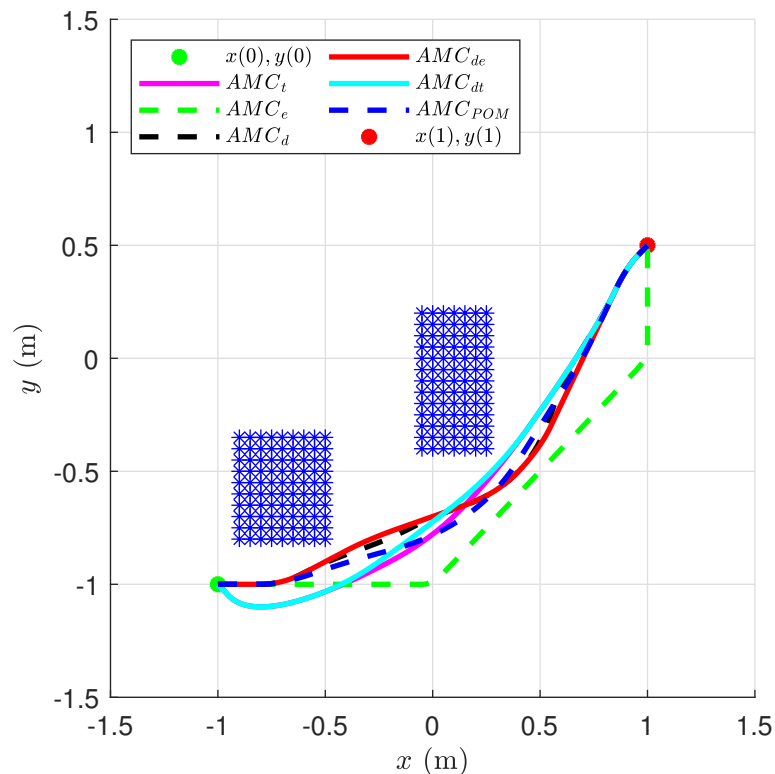


Figura 5-21.: Comparación de planificadores con AMC

En la Tabla 5-1 se muestra la comparación de los parámetros de esfuerzo de control, variación total de la señal de control, tiempo de navegación y distancia calculados para cada una de las trayectorias de la partícula generadas por el AMC en cada uno de los casos de optimización estudiados. Como se puede observar la trayectoria 3 tiene el menor esfuerzo de control, la mayor distancia, el mayor tiempo de navegación que es por lo menos el doble de las demás y para este caso posee la menor variación total de la señal. Los caminos 1 y 4 son los que tienen un menor tiempo de navegación, el esfuerzo de control es mayor en comparación con las demás.

Si se considera la combinación de los cuatro criterios de desempeño (esfuerzo de control, variación total, tiempo y distancia) se tiene que las trayectorias 1, 4, 5 y 6 combinan de una manera eficiente todos, puesto que tienen un esfuerzo de control y variación total pequeños, el tiempo de navegación y la distancia recorrida son similares. Con este análisis se muestra que se pueden encontrar soluciones que combinen varios parámetros de optimización por medio del AMC para generar referencias de las variables de estado que hagan que el DI tenga un desempeño adecuado.

Tabla 5-1.: Comparación de trayectorias del movimiento de una partícula en 2D.

| Ruta | Parámetro de optimización | Esfuerzo (u^2) | VT (u) | t (s) | Distancia (m) |
|------|---|--------------------|------------|-------|---------------|
| 1 | Tiempo | 0.18 | 1 | 10 | 2.7370 |
| 2 | Distancia | 0.16 | 1.9 | 12 | 2.6847 |
| 3 | Esfuerzo de control | 0.04 | 0.6 | 26 | 2.9023 |
| 4 | Distancia y tiempo | 0.18 | 1 | 10 | 2.7258 |
| 5 | Distancia y esfuerzo de control | 0.1 | 1.1 | 13 | 2.6851 |
| 6 | Tiempo, esfuerzo de control y distancia | 0.14 | 1.3 | 11 | 2.6862 |

5.2. Complejidad computacional del AMC

A lo largo de este documento se ha hablado de la complejidad computacional del AMC, la cual aumenta considerablemente por varios factores, y el tiempo de ejecución depende del hardware en el que se implemente y el lenguaje empleado. A continuación se detallan los tiempos de generación de las diferentes estructuras creadas sobre los distintos lenguajes y entornos de ejecución, así como los problemas encontrados durante este trabajo de investigación.

La primera implementación del AMC se realizó en el software Matlab, sobre una laptop con procesador core i5-2430M a 2.4 GHz y memoria RAM de 4 GB a 1333 MHz.. Ésta tuvo un buen desempeño para el desplazamiento del DI en 1D, pero al momento de simular el movimiento sobre el plano (2D), se aumentó el tiempo de ejecución a 4 [h]. En esta última, se generaron matrices dispersas con 923.521 celdas, que se guardaron en un archivo de 47 MB. Después de analizar el tiempo de ejecución, surgió la necesidad de realizar el código en un lenguaje que optimizara los tiempos, por lo cual se empleó el ANSI C. Para llevar a cabo la implementación, fue necesario utilizar la librería de matrices dispersas denominada *csparse*, con una ligera modificación en la impresión de archivos. El algoritmo en C minimizó el tiempo de ejecución hasta los 442 s, pero aumentó la cantidad de archivos (1 por grafo) y tamaño de los mismos hasta los 270 MB.

La segunda implementación del AMC es una versión optimizada y con una nueva forma de crear la estructura de datos. Se programó sobre el lenguaje C y en una laptop con las siguientes características: procesador core i7-6700HQ a 2.6 GHz y memoria RAM de 16 GB a 2133 MHz, para un espacio de estados con 450.241 celdas. Cada grafo se guardó en archivos independientes y cada uno tardó un tiempo de 205 s en generarse. Este algoritmo fue necesario ejecutarlo varias veces para guardar cada uno de los grafos, debido a que al tratar de compilarlo todo, se desbordaba la memoria del sistema. El tamaño final de cada archivo es de 80,68 MB. Para este algoritmo se utilizaron 3 señales de control diferentes por dimensión (9 combinaciones de señales de control en las dos dimensiones) aplicadas durante 1 s. Al disminuir el tiempo de la señal de control, se disminuye la discretización del espacio de posición y velocidad, aumentando la cantidad de celdas.

El AMC se ejecuta una vez para cada grafo del espacio de trabajo y se guardan en diferentes archivos de texto. Luego, para poner los obstáculos existentes en el entorno de trabajo se realiza una rutina adicional en Matlab donde se lee un archivo con la codificación de celda libre (1) o celda ocupada (0). El algoritmo analiza las coordenadas en las que se localizan los obstáculos, las convierte en celdas según la codificación de la estructura trabajada y reemplaza el costo por cero en todas las celdas donde se localiza el obstáculo. El costo computacional de esta rutina depende de la cantidad de celdas del espacio de trabajo y la cantidad de nodos ocupados por un obstáculo, oscilando entre 5 y 45 minutos para las pruebas realizadas.

5.3. Conclusión

El AMC puede ser empleado para la solución de problemas de optimización simples o multi-objetivo de forma numérica en la planificación de trayectorias del DI. A medida que crecen las dimensiones, o la discretización de las variables de estado disminuye, el costo computacional se incrementa considerablemente. Los ejemplos mostrados en esta sección fueron escogidos después de realizar un análisis de pertinencia entre el costo computacional y la discretización del problema.

Debido a la forma en la que se genera la estructura de datos, el tiempo de ejecución depende del lenguaje de programación y el sistema empleado. Como trabajo futuro, se pretende encontrar una forma optimizada en la construcción del algoritmo, aprovechando la geometría del problema, que es similar en todo el espacio de estados.

Parte III.

**Planificación de trayectorias de Robots
móviles**

6. Modelo dinámico de robots móviles de guiado diferencial

Este capítulo se muestra el modelo dinámico de un robot móvil de guiado diferencial que describe el comportamiento del sistema teniendo en cuenta los parámetros cinemáticos, dinámicos e inerciales, así como las limitaciones de movimiento asociadas. Ésto con el fin de observar como la linealización por realimentación aplicada al modelo genera un sistema equivalente al DI agregando las ecuaciones que gobiernan la dinámica de las ruedas, que serán empleadas en el siguiente capítulo para el cálculo de energía consumida por el robot en el seguimiento de la ruta planificada con el AMC. En la Figura 6-1 se muestra el vehículo empleado para encontrar el modelo dinámico descrito en [129] y que se reproduce en este capítulo.

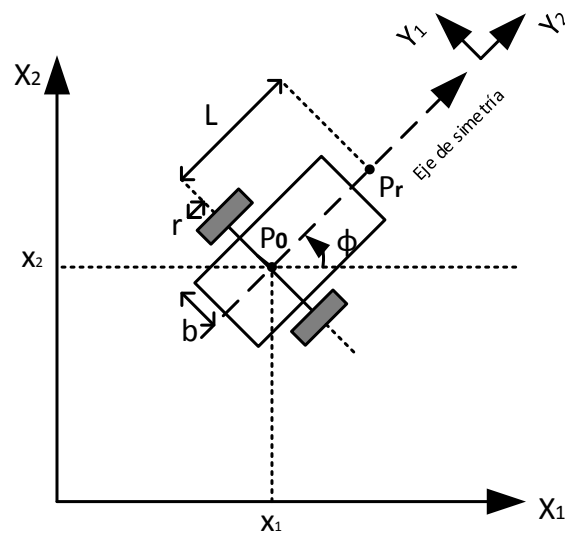


Figura 6-1.: Robot móvil de guiado diferencial.

A continuación se detallan cada una de las variables y parámetros empleados en el desarrollo matemático.

| Símbolo | Término |
|------------------|--|
| b | Distancia entre cada rueda y el eje de simetría |
| r | Radio de las llantas |
| c | Constante igual a $\frac{r}{2b}$ |
| d | Distancia entre el punto P_0 y el centro de masa del robot |
| I_c | Momento de inercia del robot sin ruedas ni motor |
| I_m | Momento de inercia de cada rueda con motor cerca al diámetro de la rueda |
| I_ω | Momento de inercia de cada rueda con motor |
| m_c | Masa del robot sin ruedas ni motor |
| m_ω | Masa de cada rueda con motor |
| x_1 | Variable de estado 1 (Posición del robot en el eje x) |
| x_2 | Variable de estado 2 (Posición del robot en el eje y) |
| \dot{x}_1 | Derivada de variable estado 1 (Velocidad del robot en el eje x) |
| \dot{x}_2 | Derivada de variable estado 2 (Velocidad del robot en el eje y) |
| θ_1 | Posición angular de la rueda 1 |
| θ_2 | Posición angular de la rueda 2 |
| $\dot{\theta}_1$ | Velocidad angular de la rueda 1 |
| $\dot{\theta}_2$ | Velocidad angular de la rueda 2 |
| ϕ | Posición angular del robot |
| $\dot{\phi}$ | Velocidad angular del robot |
| P_0 | Punto medio del eje imaginario de las ruedas |
| P_r | Punto de control hacia adelante del robot |
| L | Distancia entre P_0 y P_r |

Se define la primera restricción de movimiento del vehículo de la Figura 6-1, que indica que el móvil no puede desplazarse de forma lateral.

$$\dot{x}_2 \cos \phi - \dot{x}_1 \sin \phi = 0 \quad (6-1)$$

La segunda restricción de movimiento se obtiene al asumir que cada rueda gira sin deslizamiento.

$$\dot{x}_1 \cos \phi + \dot{x}_2 \sin \phi + b\dot{\phi} = r\dot{\theta}_1 \quad (6-2)$$

$$\dot{x}_1 \cos \phi + \dot{x}_2 \sin \phi - b\dot{\phi} = r\dot{\theta}_2 \quad (6-3)$$

Utilizando técnicas de geometría diferencial, resulta que, dos de las tres restricciones son no holónomas y la tercera es holónoma. Para obtener la restricción holónoma como se muestra en [130] se resta la Ec. (6-2) de la Ec. (6-3), dando como resultado la Ec. (6-4).

$$2b\dot{\phi} = r(\dot{\theta}_1 - \dot{\theta}_2) \quad (6-4)$$

Integrando la Ec. (6-4), se obtiene la Ec. (6-5), la cual es una restricción holónoma.

$$\phi = c(\theta_1 - \theta_2) \quad (6-5)$$

La Ec. (6-6) es también no holónoma y se obtiene sumando la Ec. (6-2) y la Ec. (6-3).

$$\dot{x}_1 \cos \phi + \dot{x}_2 \sin \phi = cb(\dot{\theta}_1 + \dot{\theta}_2) \quad (6-6)$$

Al escribir las ecuaciones de restricción (6-1) y (6-6) en forma matricial:

$$A(q)\dot{q} = 0, \quad (6-7)$$

donde,

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \theta_1 \\ \theta_2 \end{bmatrix} \quad (6-8)$$

$$A(q) = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{bmatrix} = \begin{bmatrix} -\sin \phi & \cos \phi & 0 & 0 \\ -\cos \phi & -\sin \phi & cb & cb \end{bmatrix} \quad (6-9)$$

La energía cinética total de la base móvil de dos ruedas es:

$$K = \frac{1}{2}m(\dot{x}_1^2 + \dot{x}_2^2) + m_c c d (\dot{\theta}_1 - \dot{\theta}_2)(\dot{x}_2 \cos \phi - \dot{x}_1 \sin \phi) + \frac{1}{2}I_\omega(\dot{\theta}_1^2 + \dot{\theta}_2^2) + \frac{1}{2}I_c^2(\dot{\theta}_1 - \dot{\theta}_2)^2, \quad (6-10)$$

donde,

$$m = m_c + 2m_\omega$$

$$I = I_c + 2m_\omega b^2 + 2I_m$$

La ecuación de Lagrange de movimiento para sistemas robóticos no holónomos es gobernada por:

$$\frac{d}{dt} \left(\frac{\partial K}{\partial \dot{q}_i} \right) - \frac{\partial K}{\partial q_i} = \tau - a_{1i}\lambda_1 - a_{2i}\lambda_2 \quad (6-11)$$

$$i = 1, \dots, 4$$

donde q_i es la coordenada generalizada en Ec. (6-8), τ_i es la fuerza generalizada, a_{ij} es el coeficiente de ecuación de restricción Ec. (6-9) y λ_1, λ_2 son los multiplicadores de Lagrange. Sustituyendo la energía cinética total Ec. (6-10) en la ecuación Ec. (6-11), se obtiene:

$$m\ddot{x}_1 - m_c d(\ddot{\phi} \sin \phi + \dot{\phi}^2 \cos \phi) = \lambda_1 \sin \phi + \lambda_2 \cos \phi \quad (6-12)$$

$$m\ddot{x}_2 + m_c d(\ddot{\phi} \cos \phi - \dot{\phi}^2 \sin \phi) = -\lambda_1 \cos \phi + \lambda_2 \sin \phi \quad (6-13)$$

$$m_c c d(\ddot{x}_2 \cos \phi - \ddot{x}_1 \sin \phi) + (I_c^2 + I_\omega)\ddot{\theta}_1 - I_c^2 \ddot{\theta}_2 = \tau_1 - c b \lambda_2 \quad (6-14)$$

$$-m_c c d(\ddot{x}_2 \cos \phi - \ddot{x}_1 \sin \phi) - I_c^2 \ddot{\theta}_1 + (I_c^2 + I_\omega)\ddot{\theta}_2 = \tau_2 - c b \lambda_2 \quad (6-15)$$

donde τ_1 y τ_2 son los torques de las dos ruedas. Estas ecuaciones pueden ser escritas en forma matricial:

$$M(q)\ddot{q} + V(q, \dot{q}) = E(q)\tau - A^T(q)\lambda \quad (6-16)$$

donde $A(q)$ es definida en la Ec. (6-9) y las otras variables se definen como:

$$M(q) = \begin{bmatrix} m & 0 & -m_c c d \sin \phi & m_c c d \sin \phi \\ 0 & m & m_c c d \cos \phi & -m_c c d \cos \phi \\ -m_c c d \sin \phi & m_c c d \cos \phi & I_c^2 + I_\omega & -I_c^2 \\ m_c c d \sin \phi & -m_c c d \cos \phi & -I_c^2 & I_c^2 + I_\omega \end{bmatrix}$$

$$V(q, \dot{q}) = \begin{bmatrix} -m_c d \dot{\phi}^2 \cos \phi \\ -m_c d \dot{\phi}^2 \sin \phi \\ 0 \\ 0 \end{bmatrix}, \quad E(q) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}, \quad \lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}$$

6.1. Realización en el espacio de estados

Se establece una realización en el espacio de estados de la ecuación de movimiento Ec. (6-16) y las restricciones Ec. (6-7). Dejando $S(q)$ una matriz 4×2

$$S(q) = \begin{bmatrix} s_1(q) & s_2(q) \end{bmatrix} = \begin{bmatrix} cb \cos \phi & cb \cos \phi \\ cb \sin \phi & cb \sin \phi \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (6-17)$$

Cuyas columnas están en el espacio de nulo de $A(q)$ en la ecuación de restricción Ec. (6-7). Entonces $A(q)S(q) = 0$. De la ecuación de restricción Ec. (6-7), la velocidad \dot{q} debe estar en el espacio nulo de $A(q)$. Entonces $\dot{q} \in \text{span}\{S1(q), S2(q)\}$, y existe un vector suave $\eta = [\eta_1 \ \eta_2]^T$ tal que:

$$\dot{q} = S(q)\eta \quad (6-18)$$

$$\ddot{q} = S(q)\dot{\eta} + \dot{S}(q)\eta \quad (6-19)$$

donde, $\eta = \dot{\theta}$, $\dot{\theta} = \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 \end{bmatrix}^T$. Ahora se multiplica a ambos lados de la Ec. (6-16) por $S^T(q)$ y notando que $S^T(q)A^T(q) = 0$ y $S^T(q)E(q) = I_{2 \times 2}$, dando como resultado:

$$S^T(q)M(q)\ddot{q} + S^T(q)V(q, \dot{q}) = S^T(q)E(q)\tau - S^T(q)A^T(q)\lambda$$

Simplificando se obtiene la Ec.

$$S^T(q)M(q)\ddot{q} + S^T(q)V(q, \dot{q}) = \tau \quad (6-20)$$

Sustituyendo la Ec. (6-19) en la Ec. (6-20), se obtiene:

$$S^T(q)M(q)(S(q)\dot{\eta} + \dot{S}(q)\eta) + S^T(q)V(q, \dot{q}) = \tau \quad (6-21)$$

Ahora haciendo el cambio al espacio de estados siguiente:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \theta_1 \\ \theta_2 \\ \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} q \\ \eta \end{bmatrix} \quad (6-22)$$

donde, $\eta = \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 \end{bmatrix}^T$

Representando el sistema de la Ec. (6-21) en el espacio de estados dado en la Ec. (6-22), se obtiene:

$$\dot{x} = f(x) + g(x)\tau \quad (6-23)$$

donde, τ es la entrada del sistema en lazo abierto.

$$f(x) = \begin{bmatrix} S\eta \\ -(S^TMS)^{-1}(S^TMS\dot{\eta} + S^TV) \end{bmatrix}, \quad g(x) = \begin{bmatrix} 0 \\ (S^TMS)^{-1} \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

Aplicando la realimentación de estados para tener facilidad en el control y con esto una nueva variable de entrada μ .

$$\tau = \alpha^1(x) + \beta^1(x)\mu$$

$$\tau = (S^TMS\dot{\eta} + S^TV) + (S^TMS)S^TE\mu \quad (6-24)$$

Al reemplazar τ , el sistema queda como se muestra en la Ec. (6-25).

$$\dot{x} = f^1(x) + g^1(x)\mu \quad (6-25)$$

donde,

$$f^1(x) = \begin{bmatrix} S\eta \\ 0 \end{bmatrix}, \quad g^1(x) = \begin{bmatrix} 0 \\ I_{2 \times 2} \end{bmatrix}, \quad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$$

6.2. Control Look-Ahead

Debido a que el punto central P_0 del móvil no puede ser controlado usando realimentación estática como se muestra en la Ec. (6-28), se debe utilizar un punto hacia adelante P_r , con una distancia $L \neq 0$ entre P_0 y P_r .

Se toma un nuevo sistema coordenado, cuya salida es la posición del punto P_r

$$y = h(x) = \begin{bmatrix} x_1 + L \cos \phi \\ x_2 + L \sin \phi \end{bmatrix} \quad (6-26)$$

Se verifica que el sistema sea linealizabile con esta ecuación de salida, derivando hasta encontrar la entrada.

$$\dot{y} = \frac{\partial h(x)}{\partial x} \dot{x} = \frac{\partial h(x)}{\partial x} (f^1(x) + g^1(x)\mu)$$

$$\dot{y} = \begin{bmatrix} cb \cos \phi - cL \sin \phi & cb \cos \phi + cL \sin \phi \\ cb \sin \phi + cL \cos \phi & cb \sin \phi - cL \cos \phi \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \Phi(x)\eta$$

En \dot{y} no aparece la entrada μ , por tanto se deriva una vez más.

$$\ddot{y} = \Phi(x)\dot{\eta} + \dot{\Phi}(x)\eta = \Phi(x)\mu + \dot{\Phi}(x)\eta$$

$$\dot{\eta} = \begin{bmatrix} \ddot{\theta}_1 & \ddot{\theta}_2 \end{bmatrix}^T$$

La entrada aparece en la segunda derivada de la salida. La matriz de acople en este caso es $\Phi(x)$ y su determinante es:

$$\det(\Phi(x)) = -2c^2bL \quad (6-27)$$

$$\det(\Phi(x)) \neq 0 \rightarrow L \neq 0 \quad (6-28)$$

Por tanto si $L = 0$, es decir $P_r = P_0$ la matriz es singular y no se puede realizar la realimentación de entrada-salida. Para el caso que $L \neq 0$ la matriz es no singular y se puede realizar el control. La señal de realimentación para lograr la linealización por realimentación de entrada-salida es:

$$\mu = \Phi^{-1}(x)(u - \dot{\Phi}(x)\eta) \quad (6-29)$$

Si se reemplaza la ley de realimentación no lineal dada por la Ec. (6-29) en el sistema, se obtienen la Ec. (6-30) (6-31). Como se puede apreciar, estas ecuaciones representan el modelo matemático del DI.

$$\ddot{y}_1 = u_1 \quad (6-30)$$

$$\ddot{y}_2 = u_2 \quad (6-31)$$

La señal de control aplicada en cada dimensión tiene la forma mostrada:

$$u_i = \ddot{y}_{d_i} - k_1(y_i - y_{d_i}) - k_2(\dot{y}_i - \dot{y}_{d_i}) \quad (6-32)$$

donde, $y_i(t)$ es la posición en el tiempo en cada dimensión, $y_{d_i}(t)$ es la posición deseada, $\dot{y}_i(t)$ es la velocidad, $\dot{y}_{d_i}(t)$ es la velocidad deseada. El error en cada coordenada se define a continuación:

$$e_i = y_i - y_{d_i}$$

Para realizar el diseño del control $k_1, k_2 > 0$, tal que el seguimiento del error sea exponencialmente estable $e(t) \rightarrow 0$ si $t \rightarrow \infty$.

$$\ddot{e}_i + k_2\dot{e}_i + k_1e_i = 0$$

6.3. Dinámica interna

Se construye un difeomorfismo para representar el sistema en forma normal. Como el grado relativo de cada salida es 2, se pueden construir cuatro componentes del difeomorfismo $h_1(x)$, $L_f h_1(x)$, $h_2(x)$ y $L_f h_2(x)$. Como la variable de estado es de 6 dimensiones se necesita encontrar también θ_1 y θ_2 . Por tanto, la transformación del difeomorfismo es:

$$z = T(x) = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{bmatrix} = \begin{bmatrix} h_1(x) \\ L_f h_1(x) \\ h_2(x) \\ L_f h_2(x) \\ \theta_1 \\ \theta_2 \end{bmatrix} \quad (6-33)$$

Para verificar que $T(x)$ es un difeomorfismo, se calcula el Jacobiano.

$$\frac{\partial T}{\partial x} = \begin{bmatrix} 1 & 0 & -cL\sin\phi & cL\sin\phi & 0 & 0 \\ 0 & 0 & pos_1 & pos_2 & cb\cos\phi - cL\sin\phi & cb\cos\phi + cL\sin\phi \\ 0 & 1 & cL\cos\phi & -cL\cos\phi & 0 & 0 \\ 0 & 0 & pos_3 & pos_4 & cb\sin\phi + cL\cos\phi & cb\sin\phi - cL\cos\phi \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

donde,

$$\begin{aligned} pos_1 &= -c^2b\dot{\theta}_1\sin\phi - c^2L\dot{\theta}_1\cos\phi - c^2b\dot{\theta}_2\sin\phi + c^2L\dot{\theta}_2\cos\phi, \\ pos_2 &= c^2b\dot{\theta}_1\cos\phi - c^2L\dot{\theta}_1\sin\phi + c^2b\dot{\theta}_2\cos\phi + c^2L\dot{\theta}_2\cos\phi, \\ pos_3 &= c^2b\dot{\theta}_1\sin\phi + c^2L\dot{\theta}_1\cos\phi + c^2b\dot{\theta}_2\sin\phi - c^2L\dot{\theta}_2\cos\phi, \\ pos_4 &= -c^2b\dot{\theta}_1\cos\phi + c^2L\dot{\theta}_1\sin\phi - c^2b\dot{\theta}_2\cos\phi - c^2L\dot{\theta}_2\cos\phi, \end{aligned}$$

al analizar la matriz $\frac{\partial T}{\partial x}$ se determina que es de rango completo, por lo que $T(x)$ es una transformación válida en el espacio de estados. La transformación inversa $x = T^{-1}(z)$ se muestra a continuación:

$$x_1 = z_1 - L\cos(cz_5 - cz_6)$$

$$x_2 = z_3 - L\cos(cz_5 - cz_6)$$

$$\theta_1 = z_5$$

$$\theta_2 = z_6$$

$$\begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \Phi^{-1} \begin{bmatrix} z_2 \\ z_4 \end{bmatrix}$$

Se dividen las ecuaciones en dos partes, la parte lineal y la no lineal que tiene la dinámica interna del sistema.

$$\begin{aligned} z^1 &= \begin{bmatrix} z_1 & z_2 & z_3 & z_4 \end{bmatrix}^T \\ z^2 &= \begin{bmatrix} z_5 & z_6 \end{bmatrix}^T \end{aligned}$$

Después de aplicar la realimentación de la Ec. (6-29), el sistema del robot móvil es representado por la forma normal descrita en las Ec. (6-34), (6-35) y (6-36).

$$\dot{z}^1 = Az^1 + Bu \quad (6-34)$$

$$\dot{z}^2 = w(z^1, z^2) \quad (6-35)$$

$$y = Cz^1 \quad (6-36)$$

donde,

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$w(z^1, z^2) = \Phi^{-1}(z) \begin{bmatrix} z_2 \\ z_4 \end{bmatrix} = -\frac{1}{2c^2bL} \begin{bmatrix} cb \sin \phi - cL \cos \phi & -cb \cos \phi - cL \sin \phi \\ -cb \sin \phi - cL \cos \phi & cb \cos \phi - cL \sin \phi \end{bmatrix} \begin{bmatrix} z_2 \\ z_4 \end{bmatrix}$$

La Ec. (6-34) representa el comportamiento del punto P_r del robot, en cuanto a posición y velocidad en las dos dimensiones. La Ec. (6-35) contiene información de la dinámica interna del vehículo y no es observable. Ahora se define la dinámica cero del sistema de control, la cual es el comportamiento del sistema cuando las salidas son cero, lo que implica que $z^1 = 0$, en la Ec. (6-37) se observa la dinámica cero:

$$\dot{z}^2 = w(0, z^2) = 0 \quad (6-37)$$

Entonces, $z^2 = 0$ permanece constante mientras las salidas son cero. La dinámica cero es estable, pero no asintóticamente estable. En otras palabras, si el punto P_r permanece quieto igual el robot. Ahora se analiza la dinámica interna cuando el punto P_r está en movimiento, específicamente cuando se mueve hacia adelante y hacia atrás.

Movimiento hacia atrás

Para esto se asume que el robot inicia de frente hacia la dirección en X positiva y el punto de referencia es controlado al moverse en la dirección negativa de X . La velocidad del punto de referencia es:

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} z_2 \\ z_4 \end{bmatrix} = \begin{bmatrix} -\varepsilon(t) \\ 0 \end{bmatrix}$$

donde $\varepsilon(t) > 0$. Sustituyendo esto en la Ec. (6-35), se obtiene:

$$\begin{bmatrix} \dot{z}_5 \\ \dot{z}_6 \end{bmatrix} = \frac{\varepsilon(t)}{2c^2bL} \begin{bmatrix} cb \sin \phi - cL \cos \phi \\ -cb \sin \phi - cL \cos \phi \end{bmatrix}$$

Una solución de la dinámica interna es:

$$z_5^* = -\frac{1}{r}t + c_1 \quad (6-38)$$

$$z_6^* = -\frac{1}{r}t + c_1 \quad (6-39)$$

donde c_1 es una constante, por lo que las ruedas rotan con la misma velocidad, moviéndose en línea recta en la dirección negativa de X . Ahora para estudiar la estabilidad del movimiento descrito por las Ec. (6-38) y (6-39). La estabilidad del movimiento interno en z^2 puede ser formulado como la estabilidad del punto de equilibrio en ζ .

$$\zeta_1 = z_5 - z_5^*$$

$$\zeta_2 = z_6 - z_6^*$$

Se puede expresar la dinámica interna en terminos de $\zeta = \begin{bmatrix} \zeta_1 & \zeta_2 \end{bmatrix}^T$.

$$\dot{\zeta} = \begin{bmatrix} \dot{\zeta}_1 \\ \dot{\zeta}_2 \end{bmatrix} = \frac{\varepsilon(t)}{2c^2bL} \begin{bmatrix} cb \sin(c\zeta_1 - c\zeta_2) - cL \cos(c\zeta_1 - c\zeta_2) \\ -cb \sin(c\zeta_1 - c\zeta_2) - cL \cos(c\zeta_1 - c\zeta_2) \end{bmatrix} + \begin{bmatrix} 1/r \\ 1/r \end{bmatrix}$$

Este sistema tiene un subespacio de equilibrio caracterizado por

$$E_\zeta = \{\zeta | \zeta_1 = \zeta_2\}$$

Debido a que no se puede concluir sobre esta aproximación, se utiliza el método de Liapunov para establecer la condición de estabilidad. Considere la función candidata para función de Liapunov:

$$V(\zeta) = 1 - \cos(c\zeta_1 - c\zeta_2)$$

En una vecindad de E_ζ , $V(\zeta) = 0$ si $\zeta \in E_\zeta$, y $V(\zeta) > 0$ si $\zeta \notin E_\zeta$. Entonces $V(\zeta)$ es positiva definida con respecto a E_ζ .

$$\dot{V}(\zeta) = \frac{\partial V}{\partial \zeta} \dot{\zeta} = \frac{\varepsilon(t)}{L} \sin^2(c\zeta_1 - c\zeta_2)$$

Desde $\varepsilon(t) > 0$, $\dot{V}(\zeta)$ es también positiva definida con respecto a E_ζ . Por lo tanto el subespacio de equilibrios E_ζ no es estable.

Movimiento hacia adelante

Ahora, si el robot se mueve en la dirección positiva de X , la velocidad del punto de referencia es:

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} z_2 \\ z_4 \end{bmatrix} = \begin{bmatrix} \varepsilon(t) \\ 0 \end{bmatrix}$$

donde $\varepsilon(t) > 0$. Usando la misma función de Liapunov, se puede mostrar similarmente que:

$$\dot{V}(\zeta) = -\frac{\varepsilon(t)}{L} \sin^2(c\zeta_1 - c\zeta_2),$$

a lo largo del movimiento interno hacia adelante el sistema es estable, por ser $\dot{V}(\zeta)$ negativa definida.

Como se puede apreciar, el robot móvil de guiado diferencial puede ser linealizado de manera analítica y se encuentra que el sistema controlado se asemeja a la dinámica del DI en el punto de control P_r y su dinámica interna muestra el comportamiento de las ruedas y el móvil en el tiempo.

6.4. Simulaciones de la evolución del modelo dinámico linealizado

Tomando el modelo dinámico dado por las Ec. (6-34), (6-35), se analiza el movimiento del sistema en los puntos P_r y P_0 desde una condición inicial dada y aplicando señales de control constantes. Éstas son entregadas al sistema como se muestra en la Ec. (6-34) para observar la evolución del móvil y las ruedas. Con esto, se observa el comportamiento de las variables de estado y se procede a implementarlo en el AMC para robots móviles en el siguiente capítulo. Se debe tener presente que en un lazo de control, los valores de u_i dependen de la Ec. (6-32), por lo que en ese caso se deben entregar posiciones y velocidades de referencia para que el vehículo realice el seguimiento de la ruta. Los parámetros empleados para el algoritmo son: $r = 0.1$ m, $b = 0.165$ m, $L = 0.05$ m, los cuales pertenecen al robot *Nomad Super Scout*.

Continuando con la planificación de trayectorias por medio del AMC, se toman señales de control constantes, para observar el comportamiento del robot y posteriormente comparar con los resultados con el AMC e implementar la dinámica interna de las ruedas para estimar el consumo energético, esfuerzo de control y tiempo de navegación. A continuación se analizan diferentes casos en los que se cambia la condición inicial y la señal de control aplicada. Para todos los casos se utilizan señales de control constantes aplicadas durante 2 s.

6.4.1. Movimiento sobre el eje x con condiciones iniciales nulas

En este caso se realiza el movimiento sobre el eje x en un tiempo dado de 2 s, desde una condición inicial de $(x_0, y_0) = (0, 0)$ m, $(\dot{x}_0, \dot{y}_0) = (0, 0)$ m/s, $\phi = 0^\circ$ y con señales de control $(u_x, u_y) = (0.4, 0)$. En la Figura 6-2 se muestran las señales de control constante aplicadas durante 2 s. Debido a que la señal de control en la dimensión y es cero, el móvil sólo se desplaza en la dimensión x .

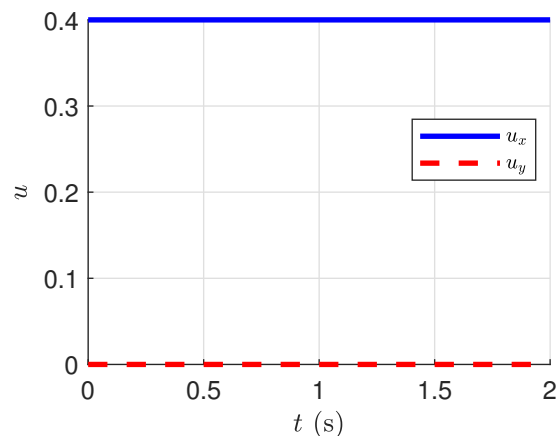


Figura 6-2.: Señales de control en el tiempo para movimiento sobre el eje x .

En la Figura 6-3 se muestra el movimiento de P_0 y P_r del robot. Se observa que P_0 empieza desde $(x, y) = (-0.05, 0)$ m, debido a que el punto de control es P_r con coordenadas iniciales $(x, y) = (0, 0)$ m y la distancia entre los dos valores es de $L = 0.05$ m. El valor final de posición en el plano también depende del punto de análisis, donde $P_r = (0.8, 0)$ m y $P_0 = (0.75, 0)$ m. En la dimensión de la ordenada no hay desplazamiento, por lo que la posición en la coordenada y siempre es cero.

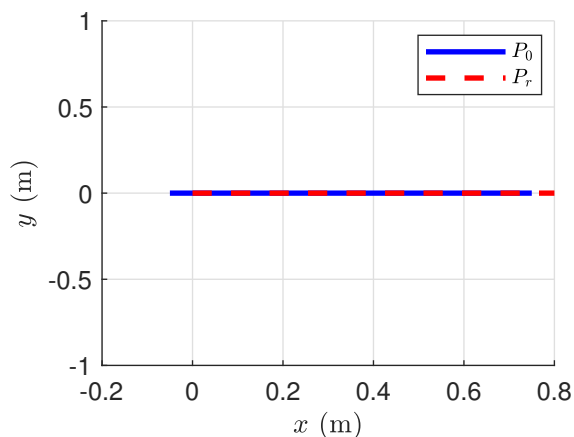


Figura 6-3.: Posición en el plano del robot para movimiento sobre el eje x .

En la Figura 6-4 se muestran las posiciones de los puntos P_0 y P_r en x y y , mientras que en la Figura 6-5 las velocidades lineales con respecto al tiempo. En la dimensión de la ordenada no hay desplazamiento, por lo que la velocidad en la coordenada y siempre es cero.

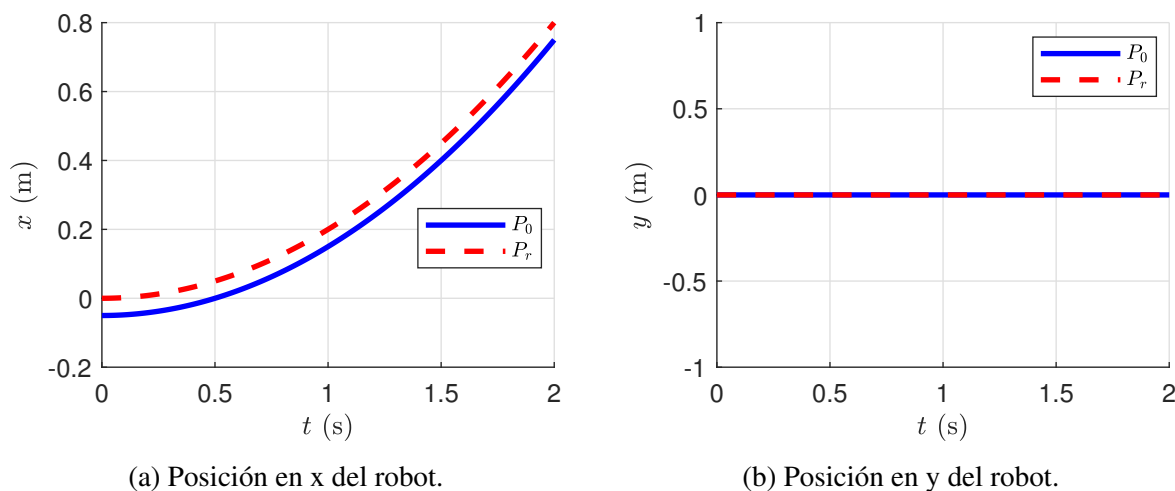
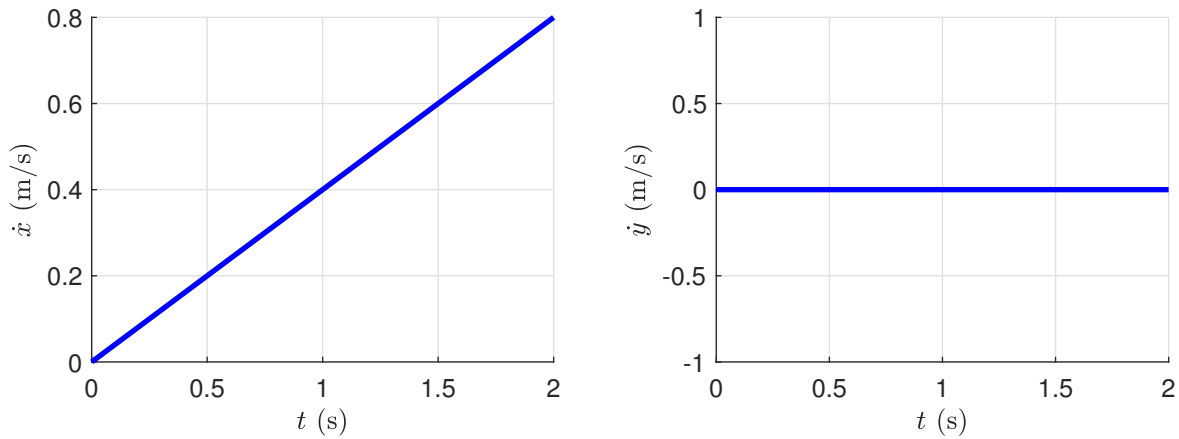


Figura 6-4.: Posición del robot en el tiempo para movimiento sobre el eje x .

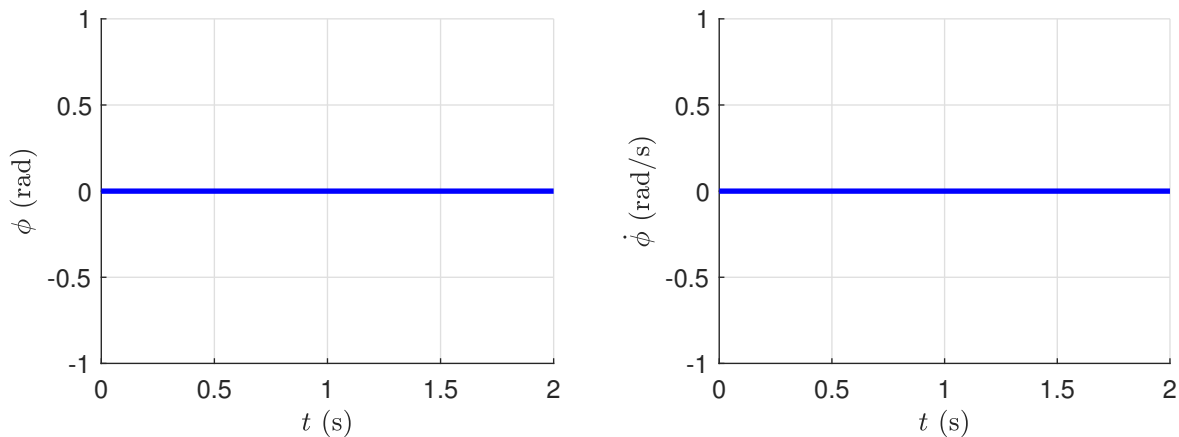


(a) Velocidad en x del robot.

(b) Velocidad en y del robot.

Figura 6-5.: Velocidad del robot en el tiempo para movimiento sobre el eje x.

En la Figura 6-6 se muestra la posición y velocidad angular del robot con respecto al tiempo. Debido a que se aplica una señal de control solo en la dimensión x y que la condición inicial del ángulo de la plataforma es 0° , la orientación del vehículo no cambia en el tiempo.

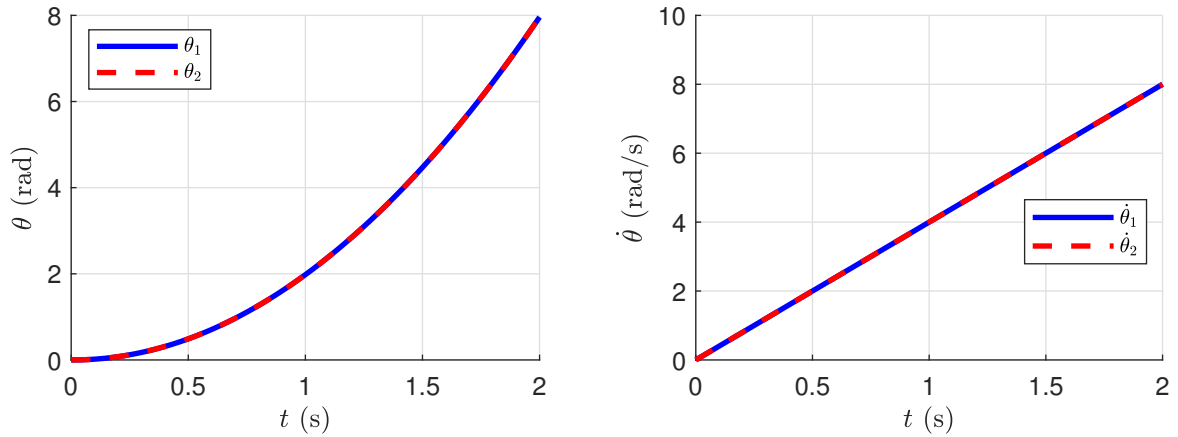


(a) Posición angular del robot.

(b) Velocidad angular del robot.

Figura 6-6.: Posición y velocidad angular del móvil para movimiento sobre el eje x.

En la Figura 6-7 se muestran las posiciones y velocidades angulares de las ruedas del robot con respecto al tiempo. Como el robot no cambia su orientación en el intervalo de análisis, las velocidades angulares de las ruedas son iguales durante todo el trayecto donde se aplican las señales de control. Para conocer el comportamiento de las ruedas se utiliza la Ec. (6-35) con la información de las velocidades lineales previamente calculadas por la Ec. (6-34).



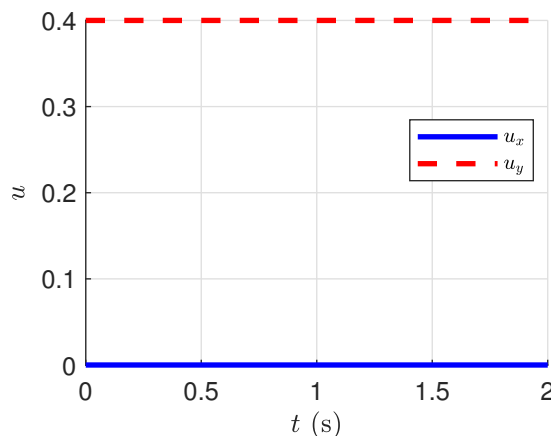
(a) Posición angular de las ruedas.

(b) Velocidad angular de las ruedas.

Figura 6-7.: Posición y velocidad angular de las ruedas para movimiento sobre el eje x.

6.4.2. Movimiento sobre el eje y con condiciones iniciales nulas

En este caso se realiza el movimiento sobre el eje y en un tiempo dado de 2s, desde una condición inicial de $(x_0, y_0) = (0, 0)$ m, $(\dot{x}_0, \dot{y}_0) = (0, 0)$ m/s, $\phi = 0^\circ$ y con señales de control $(u_x, u_y) = (0, 0.4)$. En la Figura 6-8 se muestran las señales de control constante aplicadas durante 2 s. Debido a que la señal de control en la dimensión x es cero, el punto de control P_r del móvil sólo se desplaza en la dimensión y.

**Figura 6-8.:** Señales de control en el tiempo para movimiento sobre el eje y.

En la Figura 6-9 se muestra el movimiento de P_0 y P_r del robot. Se observa que P_0 empieza desde $(x, y) = (-0.05, 0)$ m, debido a que el punto de control es P_r con coordenadas iniciales $(x, y) = (0, 0)$ m. El valor final de posición en el plano también depende del punto de análisis, donde $P_r = (0, 0.8)$ m y $P_0 = (0, 0.75)$ m. En la dimensión de la abscisa no hay desplazamiento en el punto P_r , por lo que la posición en la coordenada x siempre es cero. Para el punto P_0 si hay desplazamiento en la dimensión x mientras éste se ubica correctamente atrás de P_r en el movimiento lineal.

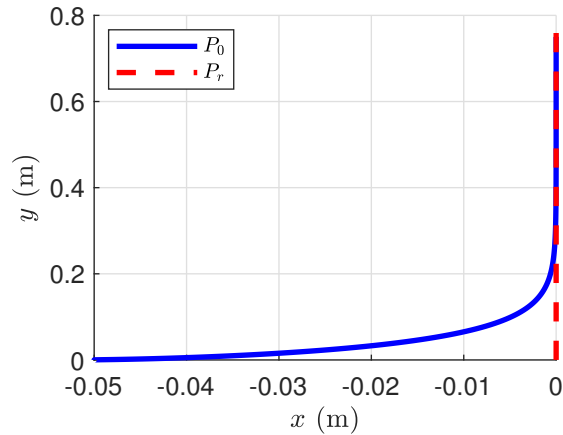


Figura 6-9.: Posición en el plano del robot para movimiento sobre el eje y.

En la Figura 6-10 se muestran las posiciones de los puntos P_0 y P_r en x y y y en la Figura 6-11 las velocidades lineales con respecto al tiempo. Se muestran las gráficas de las velocidades lineales del punto de control P_r . En la dimensión de la abscisa no hay desplazamiento del punto P_r , por lo que la velocidad en la coordenada x siempre es cero.

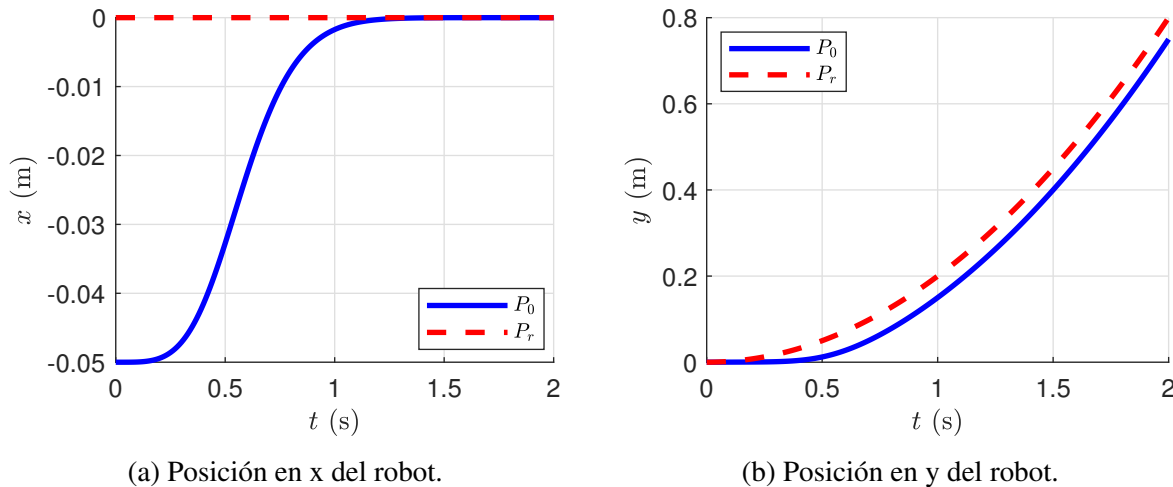
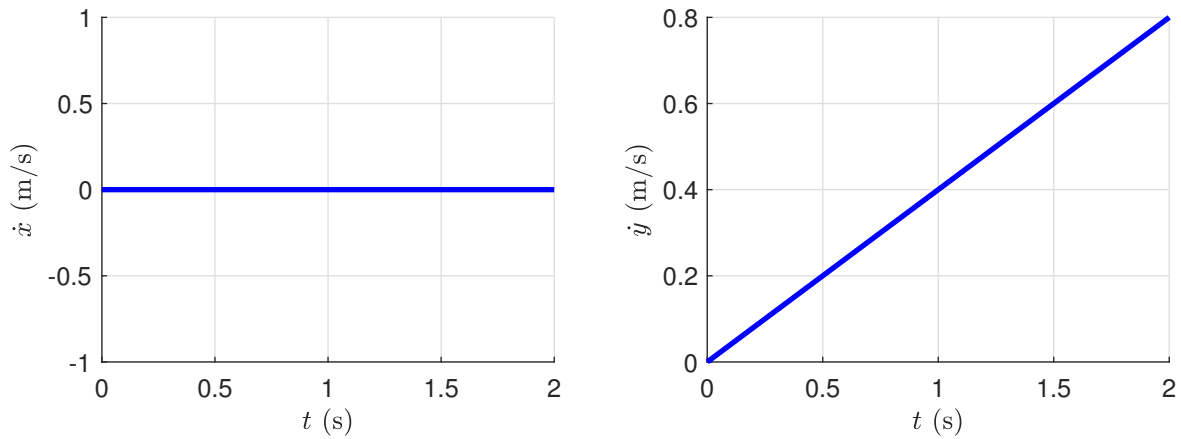


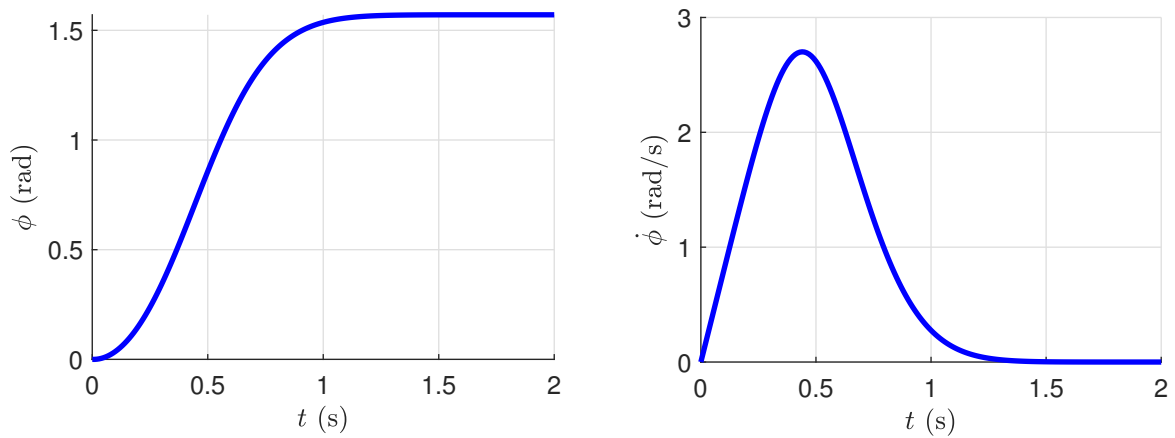
Figura 6-10.: Posición del robot en el tiempo para movimiento sobre el eje y.

En la Figura 6-12 se muestra la posición y velocidad angular del robot en el tiempo. Debido a que hay un desplazamiento del punto P_0 el ángulo de la plataforma varía desde 0° (condición inicial) hasta un valor de $\phi = 90^\circ$, haciendo que el móvil quede de frente al movimiento dado.



(a) Velocidad en x del robot.

(b) Velocidad en y del robot.

Figura 6-11.: Velocidad del robot en el tiempo para movimiento sobre el eje y.

(a) Posición angular del robot.

(b) Velocidad angular del robot.

Figura 6-12.: Posición y velocidad angular del móvil para movimiento sobre el eje y.

En la Figura 6-13 se muestran las posiciones y velocidades angulares de las ruedas del robot con respecto al tiempo. Como el robot debe cambiar su orientación para ubicarse de frente al movimiento, las velocidades angulares de las ruedas son diferentes en magnitud y sentido para hacer girar la plataforma hasta $\phi = 90^\circ$, en ese momento las velocidades angulares serán iguales. Para conocer el comportamiento de las ruedas se utiliza la Ec. (6-35) con la información de las velocidades lineales previamente calculadas por la Ec. (6-34).

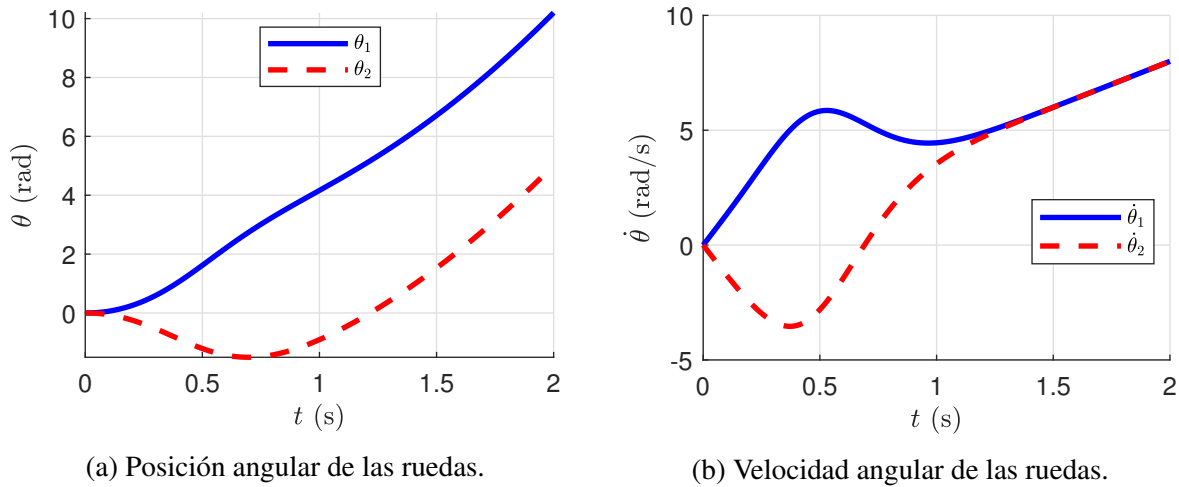


Figura 6-13.: Posición y velocidad angular de las ruedas para movimiento sobre el eje y.

6.4.3. Movimiento en dos dimensiones con $\phi = 0^\circ$

En este caso se realiza el movimiento en dos dimensiones en un tiempo dado de 2s, desde una condición inicial de $(x_0, y_0) = (0, 0)$ m, $(\dot{x}_0, \dot{y}_0) = (0, 0)$ m/s, $\phi = 0^\circ$ y con señales de control $(u_x, u_y) = (0.4, 0.4)$. En la Figura 6-14 se muestran las señales de control constante aplicadas durante 2 s. Debido a que las señales de control son diferentes de cero, el móvil se desplaza en las dos dimensiones. Como la señales de control son iguales en magnitud y las dos son positivas, el movimiento tiende a ser a un ángulo de 45° del eje x.

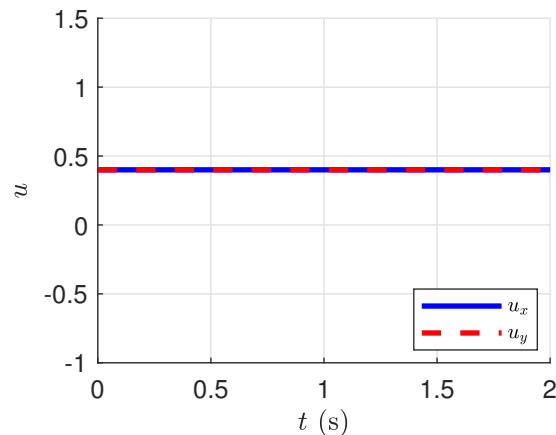


Figura 6-14.: Señales de control en el tiempo para movimiento en el plano con $\phi = 0^\circ$.

En la Figura 6-15 se muestra el movimiento de P_0 y P_r del robot. Se observa que P_0 empieza desde $(x, y) = (-0.05, 0)$ m, debido a que el punto de control es P_r con coordenadas iniciales $(x, y) = (0, 0)$ m. El valor final de posición en el plano también depende del punto de análisis. Al inicio hay un movimiento que indica una rotación del móvil, con el fin de ubicarse de frente al movimiento.

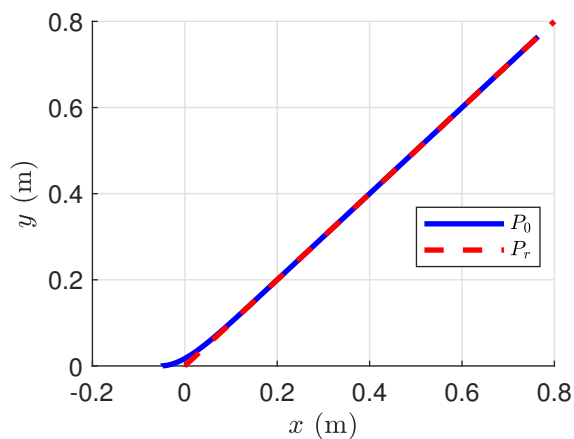
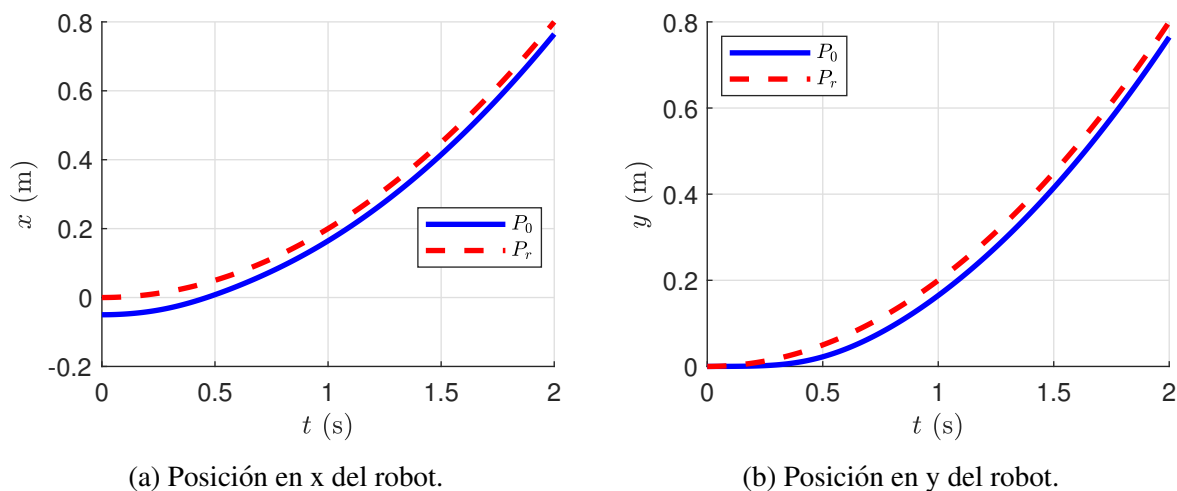


Figura 6-15.: Posición del robot para movimiento en el plano con $\phi = 0^\circ$.

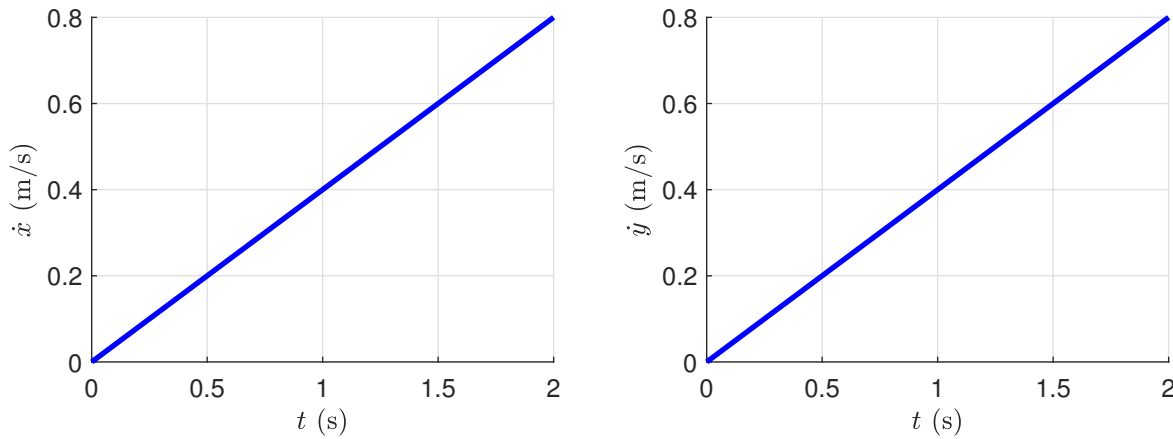
En la Figura 6-16 se muestran las posiciones de los puntos P_0 y P_r en x y y y en la Figura 6-17 las velocidades lineales con respecto al tiempo. La velocidad mostrada es la del punto P_r en cada uno de los ejes. Se observa que hay una rotación de la plataforma hasta que se llega al ángulo máximo dado por la Figura 6-18, en ese momento ya no hay rotación del móvil.



(a) Posición en x del robot.

(b) Posición en y del robot.

Figura 6-16.: Posición del robot en el tiempo para movimiento en el plano con $\phi = 0^\circ$.

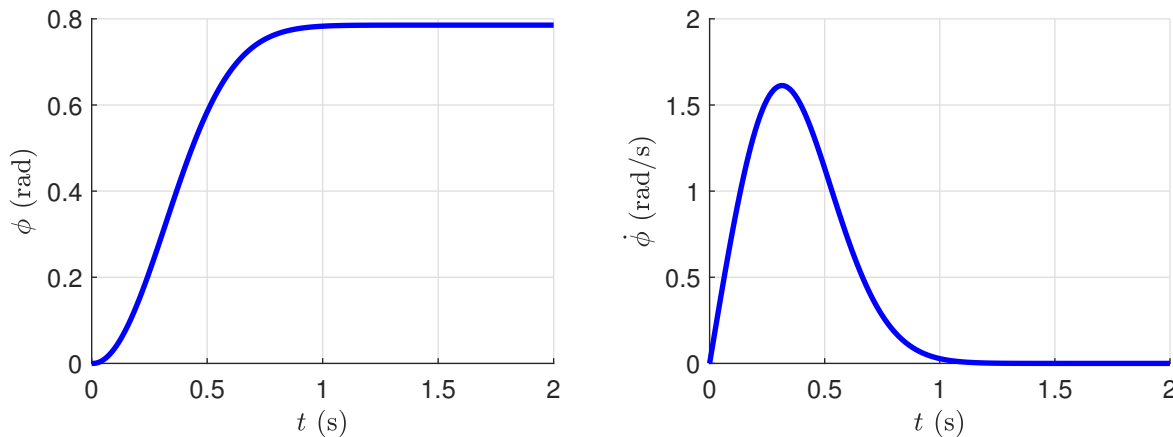


(a) Velocidad en x del robot.

(b) Velocidad en y del robot.

Figura 6-17.: Velocidad del robot en el tiempo para movimiento en el plano con $\phi = 0^\circ$.

En la Figura 6-18 se muestra la posición y velocidad angular del robot con respecto al tiempo. La orientación del móvil empieza a aumentar desde la condición inicial $\phi = 0^\circ$ hasta $\phi = 45^\circ$, debido a que se aplican señales de control iguales en magnitud y positivas las dos.



(a) Posición angular del robot.

(b) Velocidad angular del robot.

Figura 6-18.: Posición y velocidad angular del móvil para movimiento en el plano con $\phi = 0^\circ$.

En la Figura 6-19 se muestran las posiciones y velocidades angulares de las ruedas del robot con respecto al tiempo. Como el robot cambia su orientación en el tiempo para ubicarse de frente al movimiento, las velocidades angulares de las ruedas son diferentes en magnitud y sentido para hacer girar la plataforma hasta $\phi = 45^\circ$ hasta que llega a esta orientación, en ese momento las velocidades angulares serán iguales.

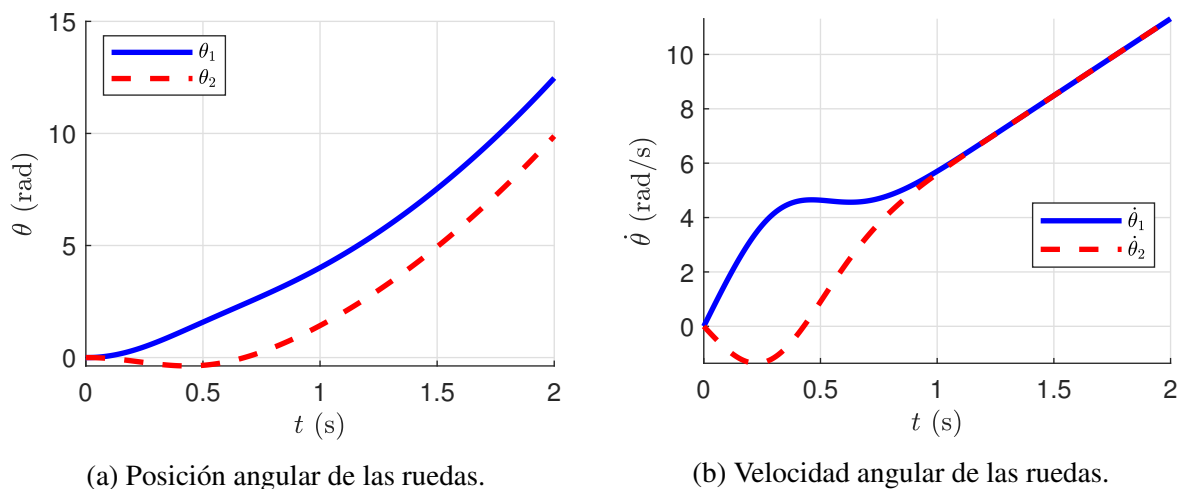


Figura 6-19.: Posición y velocidad angular de las ruedas para movimiento en el plano con $\phi = 0^\circ$.

6.4.4. Movimiento en dos dimensiones con $\phi = 135^\circ$

En este caso se realiza el movimiento en dos dimensiones en un tiempo dado de 2s, desde una condición inicial de $(x_0, y_0) = (0, 0)$ m, $(\dot{x}_0, \dot{y}_0) = (0, 0)$ m/s, $\phi = 135^\circ$ y con señales de control $(u_x, u_y) = (0.4, 0.4)$. En la Figura 6-20 se muestran las señales de control constante aplicadas durante 2 s. Debido a que las señales de control son diferentes de cero, el móvil se desplaza en las dos dimensiones. Como la señales de control son iguales en magnitud y las dos son positivas, el movimiento tiende a ser a un ángulo de 45° del eje x .

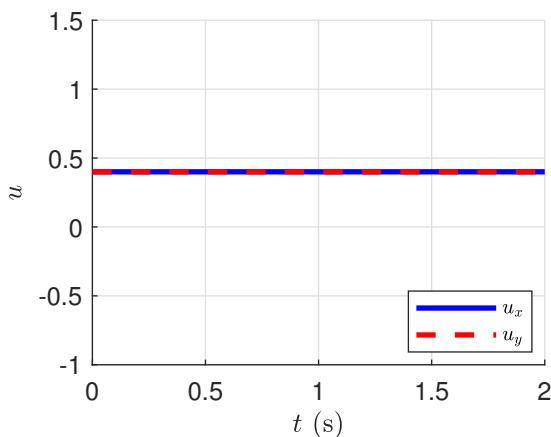


Figura 6-20.: Señales de control en el tiempo para movimiento en el plano con $\phi = 135^\circ$.

En la Figura 6-21 se muestra el movimiento de P_0 y P_r del robot. Se observa que P_0 empieza desde $(x, y) = (0.05, -0.05)$ m, debido a que el punto de control es P_r con coordenadas iniciales $(x, y) = (0, 0)$ m. Al inicio hay un movimiento que indica una rotación del móvil, con el fin de ubicarse de frente al movimiento, puesto que la plataforma inicia con un ángulo de $\phi = 135^\circ$, ésta rota hasta ubicarse como se mencionó.

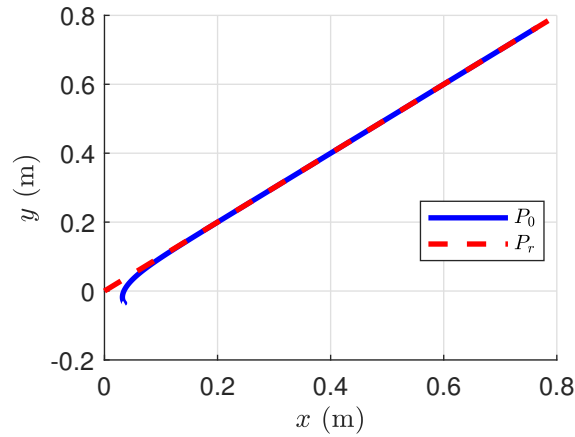


Figura 6-21.: Posición del robot para movimiento en el plano con $\phi = 135^\circ$.

En la Figura 6-22 se muestran las posiciones de los puntos P_0 y P_r en x y y y en la Figura 6-23 las velocidades lineales con respecto al tiempo. La velocidad mostrada es la del punto P_r en cada uno de los ejes. Se observa que hay una rotación de la plataforma hasta que se llega al ángulo máximo dado por la Figura 6-24, en ese momento ya no hay rotación del móvil.

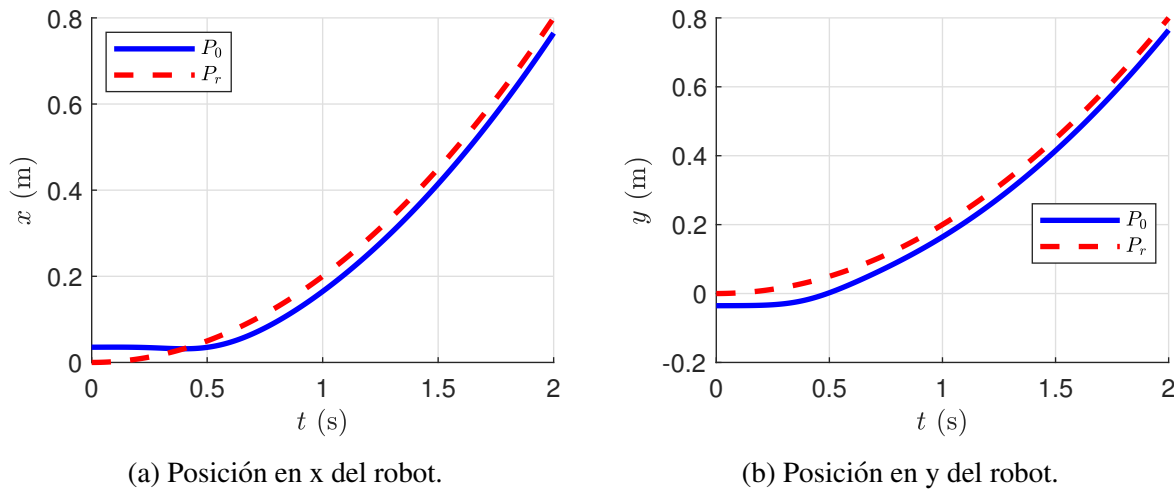
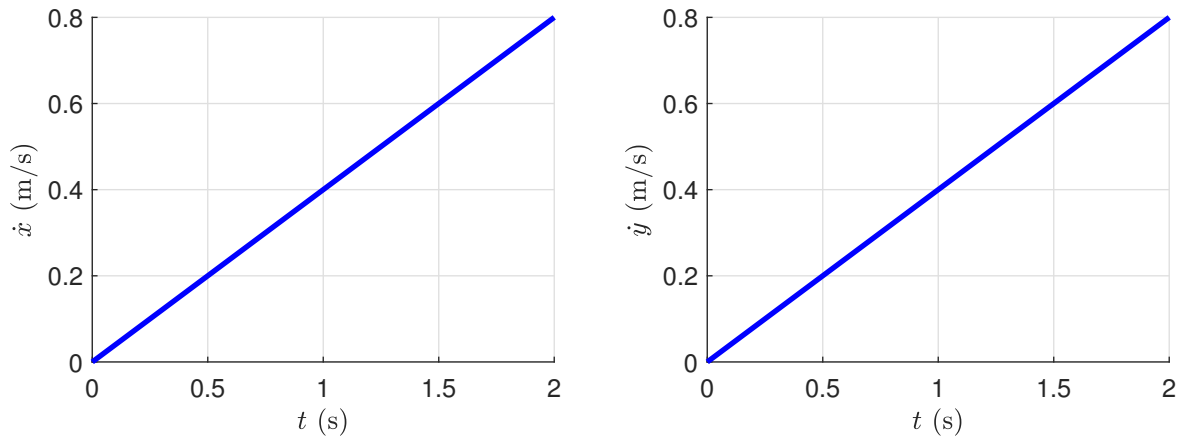


Figura 6-22.: Posición del robot en el tiempo para movimiento en el plano con $\phi = 135^\circ$.

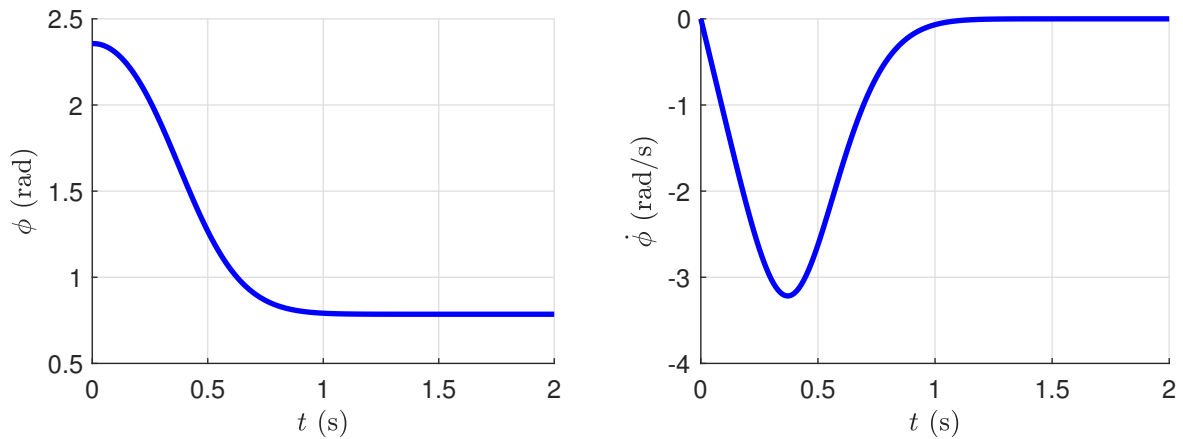


(a) Velocidad en x del robot.

(b) Velocidad en y del robot.

Figura 6-23.: Velocidad del robot en el tiempo para movimiento en el plano con $\phi = 135^\circ$.

En la Figura 6-24 se muestra la posición y velocidad angular del robot con respecto al tiempo. La orientación del móvil empieza a disminuir desde la condición inicial $\phi = 135^\circ$ hasta $\phi = 45^\circ$, debido a que se aplican señales de control iguales en magnitud y positivas las dos.



(a) Posición angular del robot.

(b) Velocidad angular del robot.

Figura 6-24.: Posición y velocidad angular del móvil para movimiento en el plano con $\phi = 135^\circ$.

En la Figura 6-25 se muestran las posiciones y velocidades angulares de las ruedas del robot con respecto al tiempo. Como el robot cambia su orientación en el tiempo para ubicarse de frente al movimiento, las velocidades angulares de las ruedas son diferentes en magnitud y sentido para hacer girar la plataforma hasta $\phi = 45^\circ$ hasta que llega a esta orientación, en ese momento las velocidades angulares serán iguales.

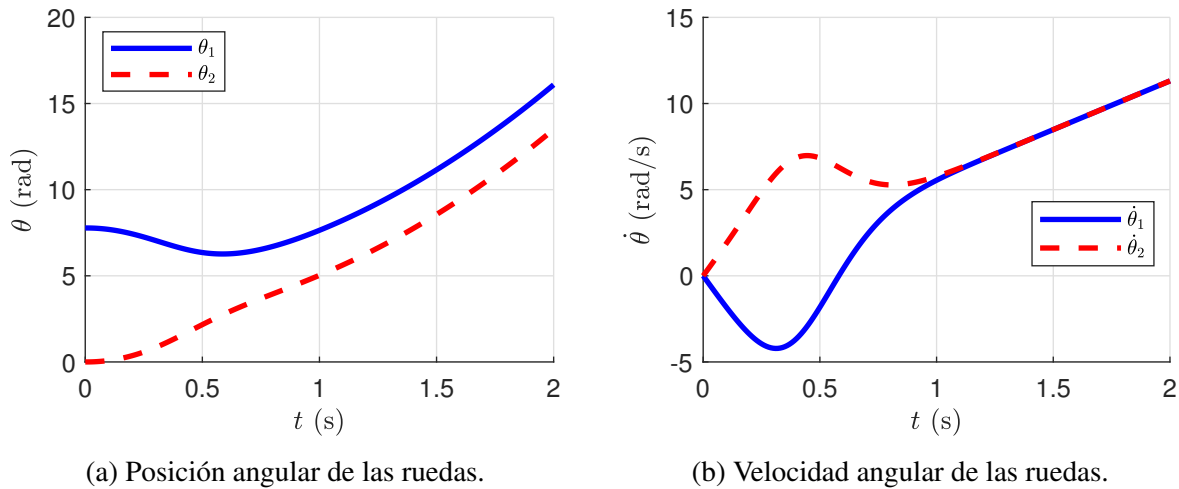


Figura 6-25.: Posición y velocidad angular de las ruedas para movimiento con $\phi = 135^\circ$.

6.5. Conclusión

El modelo dinámico descrito en el capítulo incluye las restricciones cinemáticas y dinámicas del móvil, por tanto la evolución del sistema genera trayectorias y curvas de velocidad que son físicamente realizables por el móvil de guiado diferencial. En la dinámica también se incluyen parámetros físicos del robot, los cuales pueden ser utilizados en el cálculo de la energía consumida en el seguimiento de una trayectoria previamente establecida.

Como se pudo observar, al realizar la linealización por realimentación se obtiene el modelo matemático linealizado para el movimiento del punto P_r del robot, el cual es mostrado en la Ec. (6-34). Este modelo es análogo al movimiento del DI en el plano y permite determinar el comportamiento cuando se desplaza en este espacio. Por tanto, de ahora en adelante para efectos de simplificación si se habla del movimiento del robot será del punto P_r del mismo, el cual puede realizar cualquier desplazamiento sin restricciones como se muestra en la acción de control y las simulaciones.

Para encontrar las velocidades angulares del móvil es necesario implementar el modelo no lineal dado por la Ec. (6-35), tomando las variables de velocidad y posición del robot. Las velocidades angulares serán tenidas en cuenta para el cálculo de la energía consumida por el móvil y para determinar las posiciones y velocidades que deben tomar las ruedas en el seguimiento de la ruta de referencia por parte del robot.

7. Planificación de trayectorias de robots móviles empleando el AMC

En robótica los métodos de planificación de trayectorias son empleados para dotar de un grado de inteligencia a los sistemas. Los planificadores de trayectorias son parte importante dentro de la robótica, puesto que son la base fundamental en el movimiento de robots autónomos. Los algoritmos de planificación generan trayectorias de referencia que luego son seguidas por los sistemas de control que tienen implementados cada plataforma.

En este capítulo se muestra la aplicación del AMC con el fin de generar rutas de referencia sobre el plano para el movimiento de robots móviles de guiado diferencial que optimicen de forma individual y conjunta los parámetros de tiempo, distancia, esfuerzo de control y energía. También se presenta el caso particular de un robot omnidireccional de tres ruedas de la compañía FESTO denominado Robotino[®] que sigue la ruta planificada por el AMC para optimizar los parámetros de distancia, esfuerzo de control y tiempo.

Por la forma del modelo dinámico linealizado mostrado en la Ec. (6-34) se puede deducir que el movimiento del punto P_r del robot para el control de seguimiento de trayectorias es análogo al movimiento de una partícula en 2 dimensiones. Por tanto, de ahora en adelante para efectos de simplificación si se habla del movimiento del robot será del punto P_r del mismo, el cual puede realizar cualquier desplazamiento sin restricciones como se muestra en la acción de control y las simulaciones.

7.1. AMC para robots móviles

En esta sección se muestran los resultados de la implementación del AMC para el movimiento sobre el plano de un robot móvil de guiado diferencial, utilizando el modelo dinámico descrito en (6-34). Los límites de posición y velocidad del móvil son definidos en la discretización del entorno del capítulo 5, el cual es el mismo y con condiciones iniciales y finales similares. Las simulaciones se realizan para 9 combinaciones diferentes de señales de control que logran llevar al móvil desde una celda inicial hasta una final. Para analizar el comportamiento de las señales de control del móvil se grafican las posiciones y velocidades angulares del robot que deben ser seguidas para hacer el seguimiento de la ruta planificada desde una posición inicial $(x_0, y_0) = (-1, -1)$ m hasta una final $(x_1, y_1) = (1, 0.5)$ m, con una velocidad inicial y final de $(\dot{x}, \dot{y}) = (0, 0)$ m/s.

7.1.1. Generación de trayectoria con optimización de distancia

En el capítulo 5 se implementó el AMC para generar la trayectoria del DI que une dos celdas en el espacio de estados con la menor distancia. En la Figura 7-1 se muestra el mismo ejercicio de optimización para el robot. Si se aplica la Ec. (6-35) se pueden observar en las Figuras 7-2a y 7-2b las posiciones y velocidades angulares que deben seguir las ruedas del robot para que el móvil llegue al objetivo.

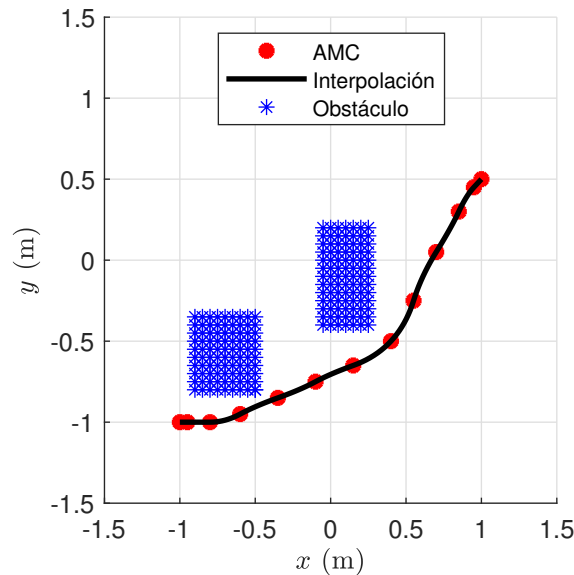


Figura 7-1.: Ruta para menor distancia.

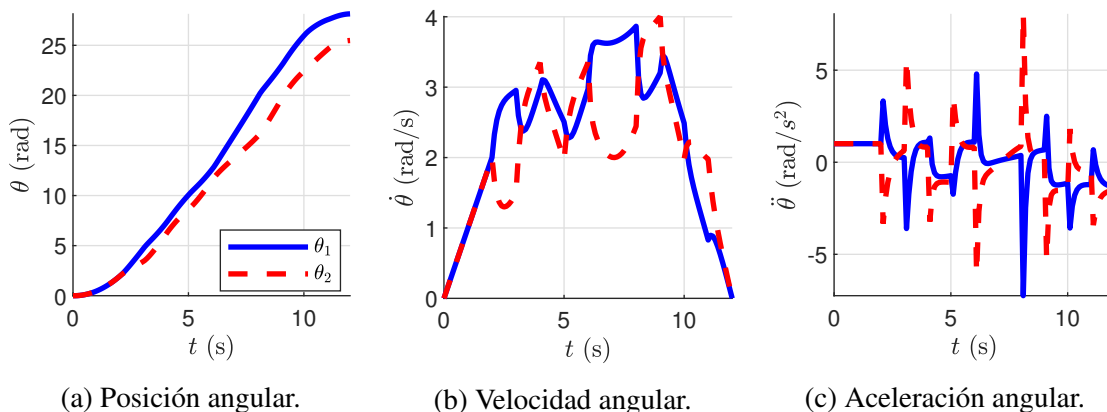
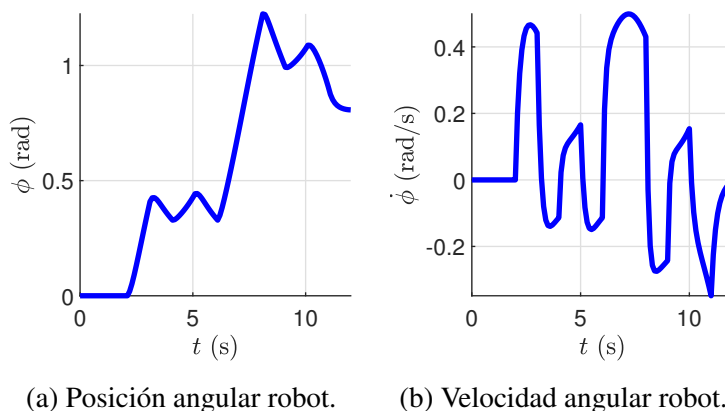


Figura 7-2.: Comportamiento de las ruedas para trayectoria optimizada en distancia.

En la Figura 7-2c se presentan las aceleraciones angulares de las ruedas, en donde el valor máximo es de 8.06 rad/s^2 y el mínimo de -7.25 rad/s^2 , con una duración de aplicación en ambos casos de 0.4 s aproximadamente. Al convertir la aceleración angular máxima en aceleración tangencial para robots con ruedas de 0.1 m de radio, se obtiene un valor de 0.806 m/s^2 . Esta cifra es inferior a la obtenida en un Tesla Modelo S P85D Semi Autónomo, que llega a una aceleración máxima de 10.288 m/s^2 . Por otro lado, la aceleración angular máxima obtenida en la simulación representa un aumento máximo en la velocidad angular del motor de 76.967 RPM cada segundo, una cifra realizable físicamente.

La orientación inicial del móvil es de $\phi = 0 \text{ rad}$ y al variar las velocidades angulares de las ruedas como en la Figura 7-2, la plataforma modifica su orientación en el tiempo como se muestra en la Figura 7-3a. En la Figura 7-3b se observa la velocidad de giro del robot.



(a) Posición angular robot.

(b) Velocidad angular robot.

Figura 7-3.: Comportamiento del robot para trayectoria optimizada en distancia.

7.1.2. Generación de trayectoria con optimización de esfuerzo de control

En la Figura 7-4 se muestra la trayectoria del robot que une dos celdas en el espacio de estados con el menor esfuerzo de control. Se pueden observar las posiciones y velocidades angulares que deben seguir las ruedas del robot para que el móvil llegue al objetivo evadiendo obstáculos en las Figuras 7-5a y 7-5b. En comparación con el caso anterior, los picos de aceleración angular en las ruedas son menores, debido a la forma de la velocidad angular.

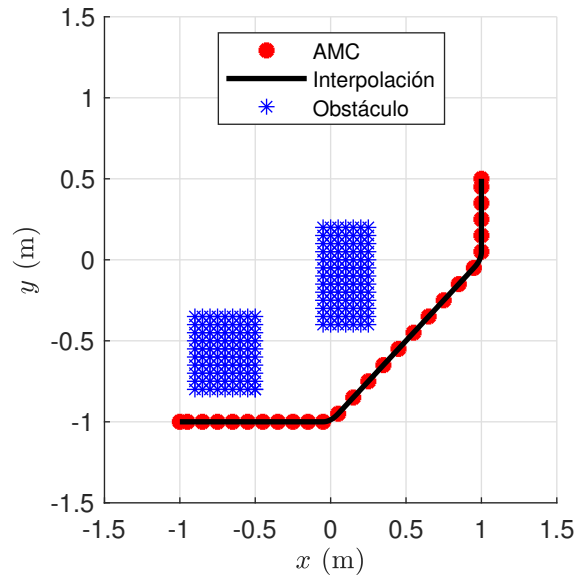
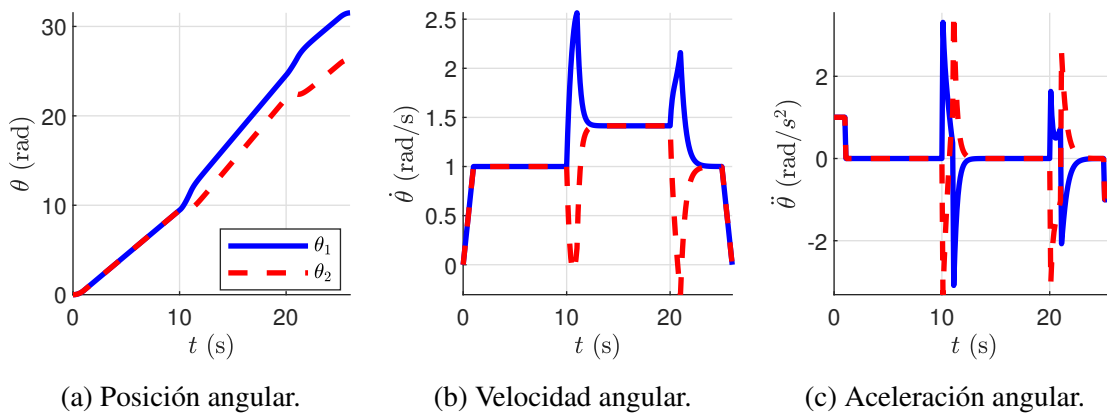


Figura 7-4.: Ruta para menor esfuerzo de control.



(a) Posición angular.

(b) Velocidad angular.

(c) Aceleración angular.

Figura 7-5.: Comportamiento de las ruedas para trayectoria optimizada en esfuerzo de control.

La orientación inicial del móvil es de $\phi = 0$ rad y al variar las velocidades angulares de las ruedas como en la Figura 7-5, la plataforma modifica su orientación en el tiempo como se muestra en la Figura 7-6a. En la Figura 7-6b se observa la velocidad de giro del robot.

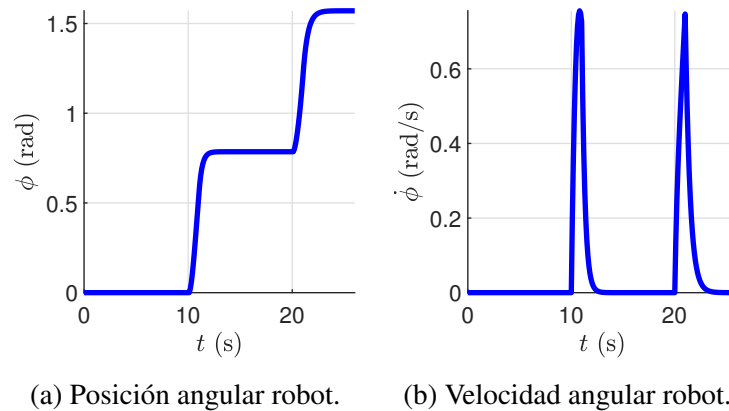


Figura 7-6.: Comportamiento del robot para trayectoria optimizada en esfuerzo de control.

7.1.3. Generación de trayectoria con menor tiempo de navegación

En la Figura 7-7 se muestra la trayectoria generada para que el robot se mueva desde la condición inicial a la final con menor tiempo de navegación. En las Figuras 7-8a y 7-8b se observan las posiciones y velocidades angulares que deben seguir las ruedas del robot para que el móvil llegue al objetivo y en la Figura 7-8c se observa la aceleración angular de las ruedas que para este caso toma como valor máximo 7.69 rad/s^2 y mínimo -6.98 rad/s^2 con duraciones de 0.8 s aproximadamente. En este caso se busca llegar a la máxima velocidad posible para el desplazamiento del robot y al limitar el espacio de estados en la discretización se garantizan valores físicamente realizables.

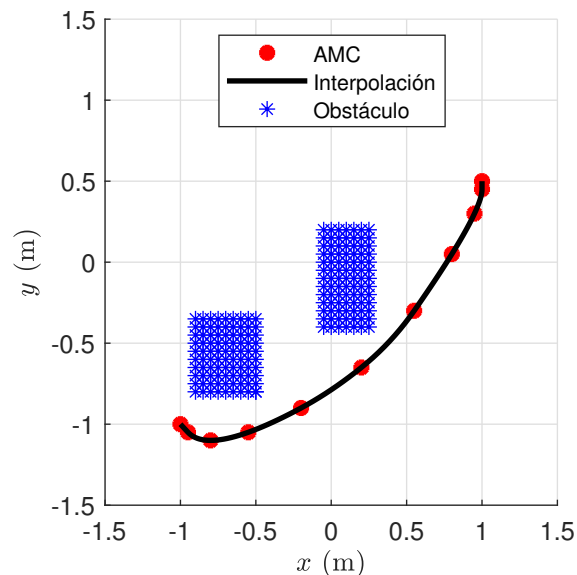


Figura 7-7.: Ruta para menor tiempo de navegación.

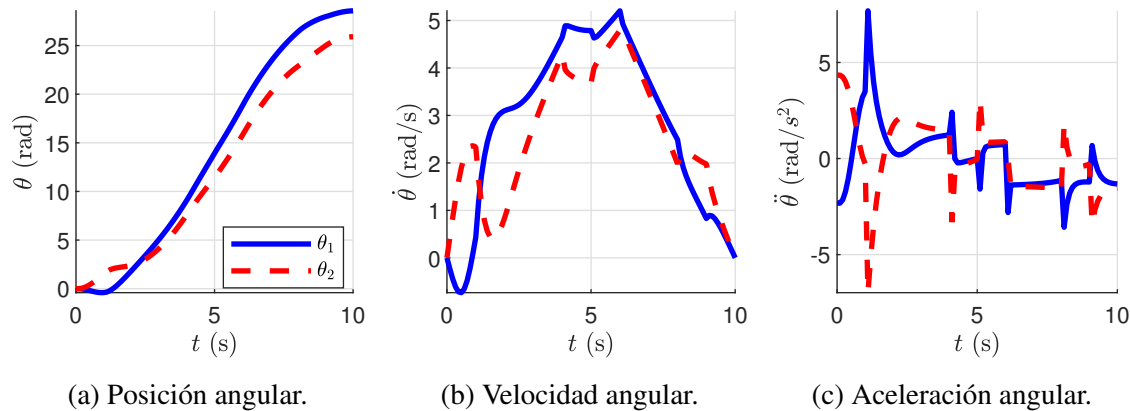


Figura 7-8.: Comportamiento de las ruedas para trayectoria optimizada en tiempo.

La orientación inicial del móvil es de $\phi = 0$ rad y al variar las velocidades angulares de las ruedas como en la Figura 7-8, la plataforma modifica su orientación en el tiempo como se muestra en la Figura 7-9a. En la Figura 7-9b se observa la velocidad de giro del robot.

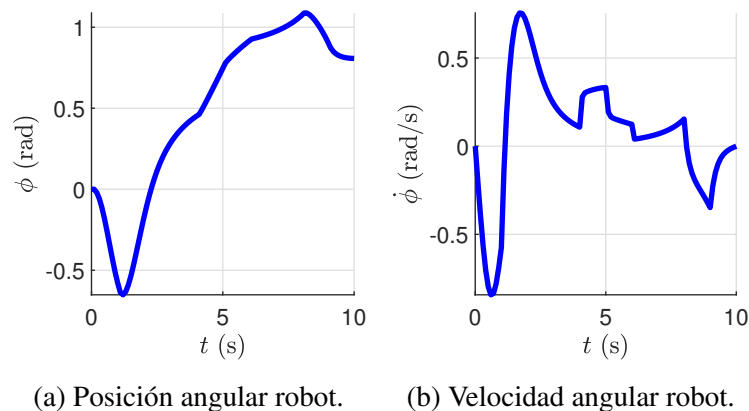


Figura 7-9.: Comportamiento del robot para trayectoria optimizada en tiempo.

7.1.4. Generación de trayectoria con POM de esfuerzo de control, distancia y tiempo

En la Figura 7-10 se muestra la trayectoria planificada para el robot móvil que tiene dos celdas en el espacio de estados para minimizar la función de costo $F = 0.2 * F_u + 0.78 * F_d + 0.02 * F_t$. Si al resultado del sistema lineal se aplica (6-35) se pueden observar en las Figuras 7-11a y 7-11b las posiciones y velocidades angulares que deben seguir las ruedas del robot para que el móvil llegue al objetivo evadiendo obstáculos minimizando distancia.

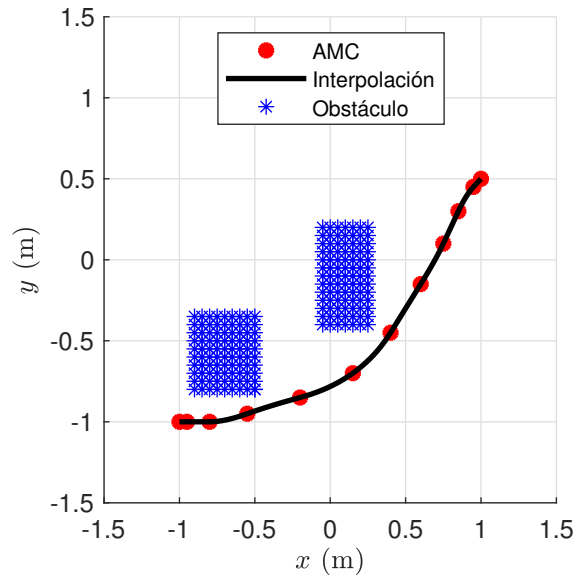


Figura 7-10.: Ruta para POM

En la Figura 7-11c se observan las velocidades angulares de las ruedas, que debido a la combinación de parámetros de optimización genera una señal con varios aumentos y decrementos.

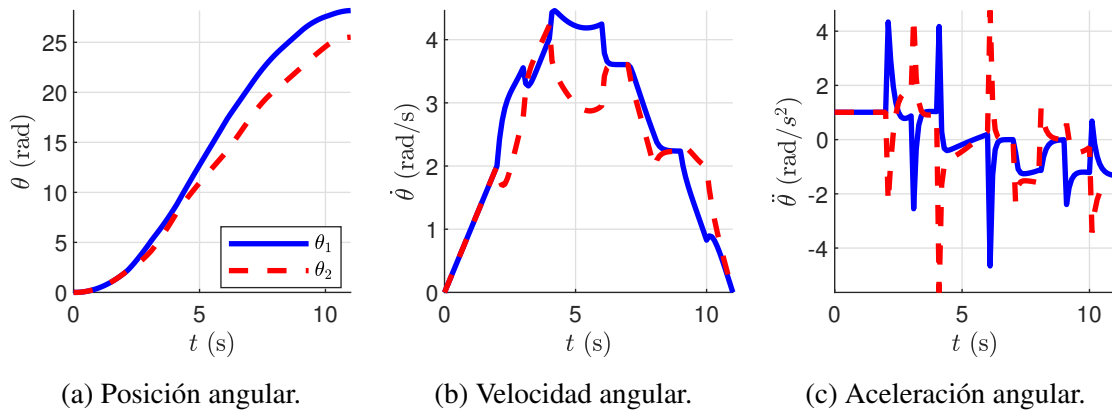


Figura 7-11.: Comportamiento de las ruedas para trayectoria con optimización multiobjetivo.

La orientación inicial del móvil es de $\phi = 0$ rad y al variar las velocidades angulares de las ruedas como en la Figura 7-11, la plataforma modifica su orientación en el tiempo como se muestra en la Figura 7-12a. En la Figura 7-12b se observa la velocidad de giro del robot.

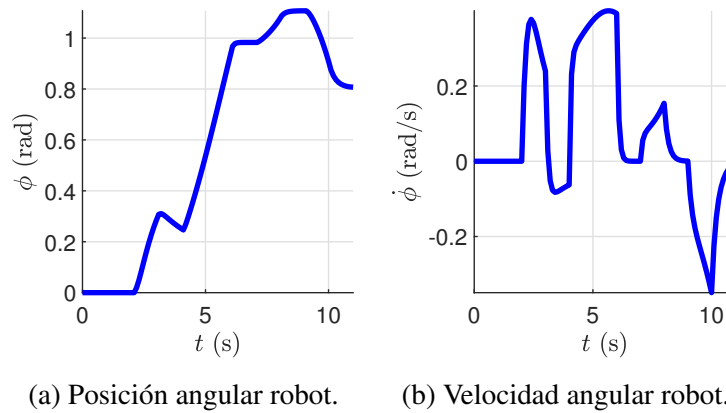


Figura 7-12.: Comportamiento del robot para trayectoria con optimización multiobjetivo.

7.1.5. Análisis de resultados de los planificadores de trayectorias

Ahora, en la Figura 7-13 se presenta la comparación de las rutas generadas por el AMC optimizando distancia, así como algunos POM. También se implementó el planificador RRT* con distancia máxima de 0.8 m. Para todos los casos el movimiento de un robot móvil es desde una celda inicial $[x_0, y_0]$ hasta una final $[x_1, y_1]$.

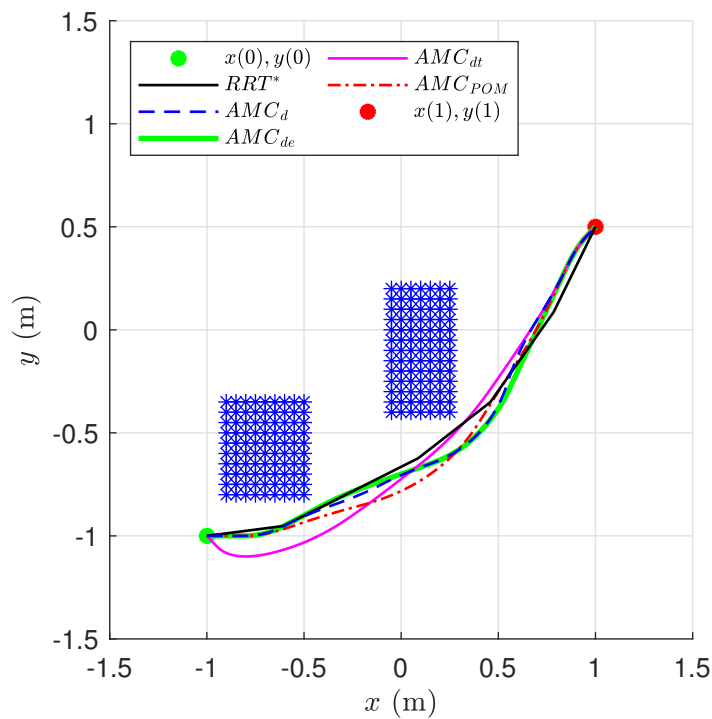


Figura 7-13.: Comparación planificadores de trayectorias.

En la Tabla 7-1 se observa la distancia generada para cada una de las rutas. El planificador que genera la trayectoria con menor distancia es el RRT*. Este método no entrega los perfiles de velocidad y aceleración del móvil, lo cual no permite estimar el tiempo de navegación necesario, esfuerzo de control generado en el seguimiento de la ruta, variación total de la señal de control, entre otros.

Por otro lado, el planificador basado en el AMC si permite hacer estas estimaciones. El AMC_d presenta oscilaciones en las velocidades intermedias, aumentando el tiempo de navegación y variación total con respecto a los otros planificadores por AMC. El AMC_{de} y AMC_{MOP} entregan la ruta con un buen desempeño en los perfiles de velocidad, tiempo de navegación, variación total, distancia y esfuerzo de control.

Tabla 7-1.: Comparación de trayectorias del movimiento de un robot en el plano.

| Ruta | Planificador | Distancia (m) |
|------|---|---------------|
| 1 | AMC _d | 2.6847 |
| 2 | AMC _{de} | 2.6851 |
| 3 | AMC _{dt} | 2.7258 |
| 4 | AMC _{MOP} | 2.6862 |
| 5 | RRT* con distancia máxima de conexión 0.8 m | 2.6400 |

Al comparar porcentualmente los planificadores con el algoritmo RRT* (Referencia), se encuentra que los resultados del AMC presentan una ruta mayor porcentualmente en distancia de: AMC_d = 1.69% , AMC_{de} = 1.71% , AMC_{MOP} = 1.75% y el AMC_{dt} = 3.25%. Cabe aclarar que el AMC tiene en cuenta los perfiles de velocidad y aceleración del móvil, para tener un mejor desempeño en el seguimiento de la ruta, así como la optimización de otros parámetros.

7.2. AMC para generar trayectorias con mínimo gasto energético en robots móviles de guiado diferencial

En robótica móvil el consumo energético juega un papel importante en la autonomía del vehículo, debido a que estas plataformas se desplazan por un terreno sin estar fijos a un lugar haciendo necesario el uso de baterías como sistema de alimentación. Algunos trabajos en la literatura son enfocados en estudiar el consumo energético, en [131–138] se muestran algunos. Por tanto, es necesario optimizar los movimientos del robot, para generar un menor gasto energético sin comprometer la ejecución de las tareas predefinidas. En [139–143] se muestran algunos trabajos para la generación de trayectorias con optimización en energía.

Los consumos de energía dependen en gran medida del tipo de trayectoria seguida por el robot, así como el peso de carga, la topografía del terreno, entre otros. En esta sección se propone la implementación del AMC para el movimiento lineal en una dimensión de posición y en el cual se analizan todas las posibles perfiles de posición y velocidad seguidas por un robot móvil de guiado diferencial que se desplaza hacia adelante, con el fin de construir un grafo que contenga la información del gasto energético para movimientos pequeños dentro del espacio de estados y luego con la implementación del método de búsqueda de caminos cortos se encuentre la trayectoria con menor gasto energético.

El AMC propuesto para la planificación de trayectorias de robots puede modificarse con el fin de calcular el gasto energético por medio del modelo dinámico propuesto en el capítulo 6. Un ejemplo del empleo de este parámetro de optimización fue introducido en el capítulo 3, donde se tomó solo la energía cinética de la partícula. Para esta aplicación se tendrá en cuenta que la energía consumida por el sistema robótico es determinado por diferentes factores mostrados en la Ec. (7-1), la cual es empleada para calcular el consumo energético en diferentes artículos en la literatura [144–146].

$$E_{battery} = E_{dc} + E_k + E_f + E_{elect} \quad (7-1)$$

donde, $E_{battery}$ es la energía de la batería usada, E_{dc} es la consumida por los motores, E_k es la pérdida de energía cinética, E_f es la pérdida debido a la fricción y E_{elect} es la consumida por los elementos electrónicos del sistema.

En [147] se presenta la validación experimental de un modelo para el consumo energético del robot *Nomad Super Scout* cuando se desplaza en línea recta hacia adelante. Este móvil es de guiado diferencial con las características estudiadas en este capítulo 6. De [148] se extraen las expresiones de potencia consumida (P_1 y P_2) de cada una de las ruedas, que se muestran en las Ec. (7-2) y (7-3). Éstas se implementan en el AMC, con el fin de generar trayectorias que puedan optimizar el consumo energético del robot.

$$P_1 = 0.0203\ddot{\theta}_1^2 + 0.148\ddot{\theta}_1\dot{\theta}_1 + 0.0014\ddot{\theta}_1^2 m_p + 0.102\ddot{\theta}_1 + 0.0017\dot{\theta}_1^2 + 0.005\ddot{\theta}_1\dot{\theta}_1 m_p + 0.373\dot{\theta}_1 + 0.0034\ddot{\theta}_1 m_p + 0.129 \quad (7-2)$$

$$P_2 = 0.0203\ddot{\theta}_2^2 + 0.148\ddot{\theta}_2\dot{\theta}_2 + 0.0014\ddot{\theta}_2^2 m_p + 0.0949\ddot{\theta}_2 + 0.0017\dot{\theta}_2^2 + 0.005\ddot{\theta}_2\dot{\theta}_2 m_p + 0.346\dot{\theta}_2 + 0.0032\ddot{\theta}_2 m_p + 0.111 \quad (7-3)$$

7.2.1. Cálculo de energía para el movimiento en una dimensión con AMC

En este apartado se realiza la evolución del modelo dinámico con el fin de calcular la energía consumida por cada rueda. Se analiza el movimiento del punto P_r del robot móvil de guiado diferencial entre $x_0 = 0$ m y $x_1 = 0.2$ m, con $\dot{x}_0 = 0$ m/s y aceleración (señal de control) $u = 0.4$ m/s², se obtienen los siguientes resultados: En la Figura 7-14 se muestra la posición y velocidad del móvil.

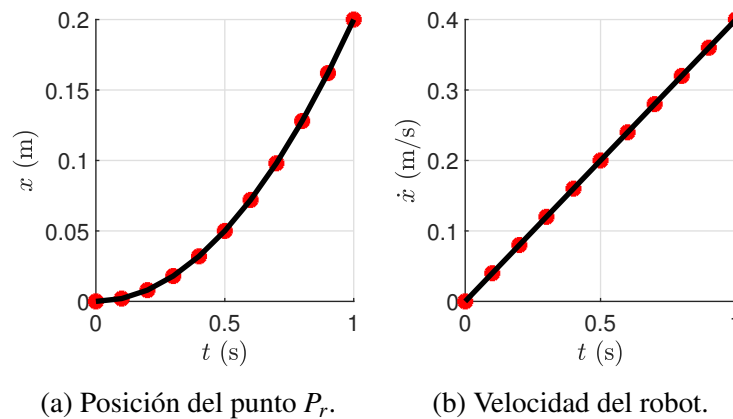


Figura 7-14.: Posición y velocidad.

En la Figura 7-15 se muestra la aceleración del robot móvil de guiado diferencial para el ejemplo descrito anteriormente.

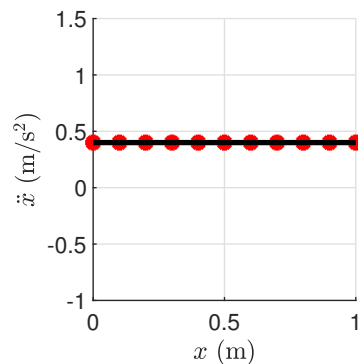


Figura 7-15.: Aceleración del móvil.

En la Figura 7-16 se muestra la posición y velocidad angular de las ruedas del móvil para el movimiento previamente descrito. En la Figura 7-17 se muestra la aceleración angular de las ruedas del móvil para el movimiento descrito.

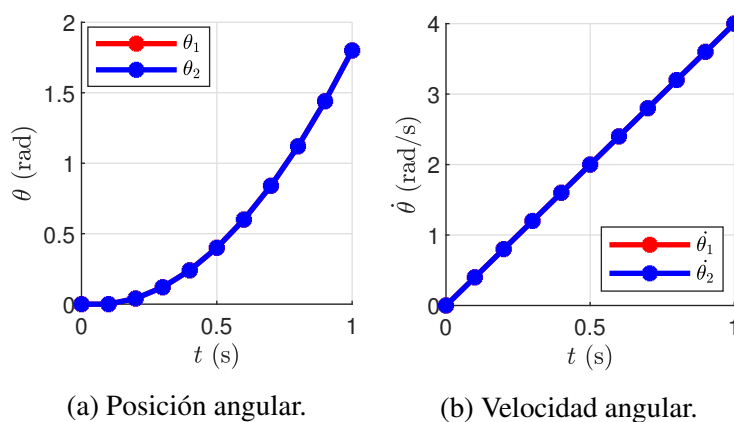


Figura 7-16.: Posición y velocidad angular de las ruedas.

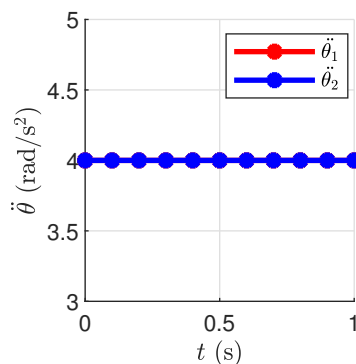


Figura 7-17.: Aceleración angular de las ruedas.

En la Figura 7-18a se muestra la potencia instantánea de las ruedas del móvil para el movimiento descrito. La potencia es calculada por las expresiones de potencia dadas en (7-2) y (7-3). En la Figura 7-18b se observa la energía consumida por las ruedas del móvil para el movimiento descrito.

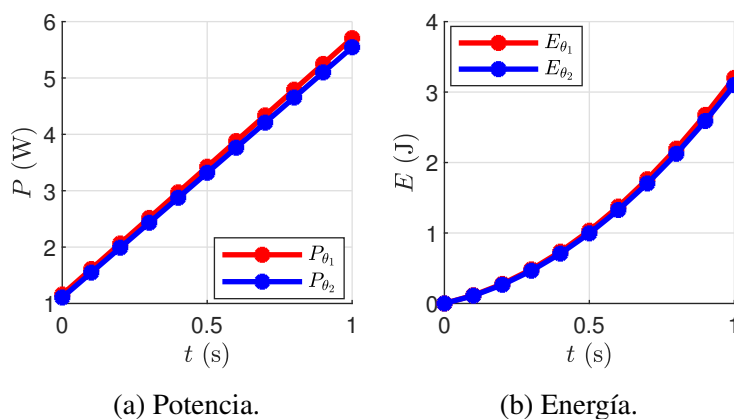


Figura 7-18.: Energía y potencia consumida por cada rueda.

Este análisis se hace para cada vecino de celda en el espacio de estados, generando el valor de energía consumida por cada rueda. Si se suman los valores de energía consumida por cada rueda se obtiene el total consumido por el móvil. Este dato se guarda en el grafo de energía denominado G_E .

7.2.2. Generación de trayectoria para el desplazamiento con menor consumo de energía por medio del AMC

Después de tener los grafos de conexiones se busca la optimización de la función de costo $F = G_E$ con el algoritmo de Dijkstra con el fin de encontrar un camino para desplazarse hacia adelante en línea recta sobre el plano XY con el menor gasto energético. Si se toma como condiciones iniciales $x_0 = -1$ m, $y_0 = 0$ m, $\dot{x}_0 = 0$ m/s, $\dot{y}_0 = 0$ m/s y finales $x_1 = 1$ m, $y_1 = 0$ m, $\dot{x}_1 = 0$ m/s, $\dot{y}_1 = 0$ m/s, se obtiene la trayectoria de referencia en el plano en la Figura 7-19, la de posición en x en el tiempo se muestra en la Figura 7-20a y el perfil de velocidad en la Figura 7-20b. En la Figura 7-20c se muestra la señal de control dada para minimizar el gasto energético. Debido a que no tiene desplazamiento en la dimensión y , no se grafican las señales de posición, velocidad y aceleración en este eje.

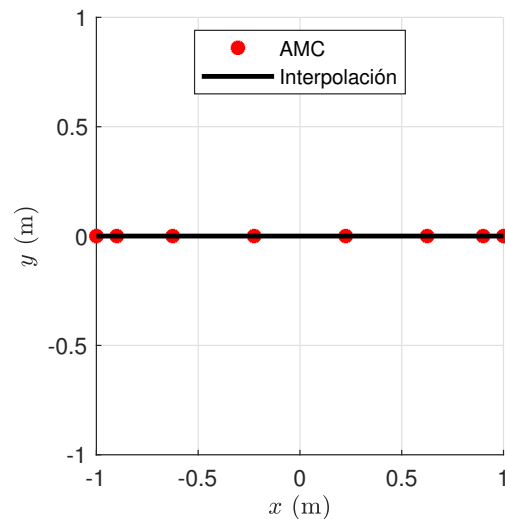


Figura 7-19.: Desplazamiento del robot en el plano.

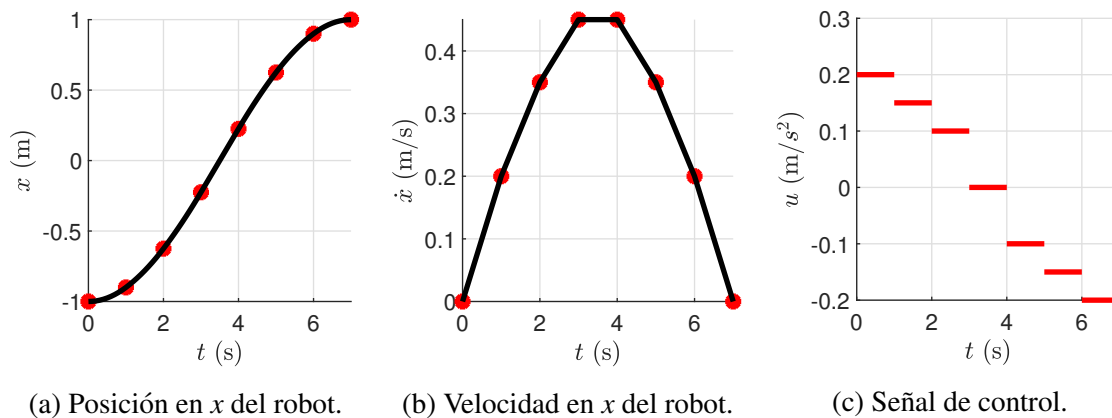


Figura 7-20.: Posición, velocidad y aceleración para minimizar consumo energético.

7.3. Aplicación del AMC para la planificación de trayectorias de robots omnidireccionales con 3 ruedas

En esta sección se muestra el experimento computacional presentado en [149] para validar el comportamiento de un robot móvil omnidireccional de 3 ruedas en el seguimiento de una trayectoria planificada por medio del AMC. Para esto, se utiliza el software Robotino[®] SIM DEMO, el cual describe el comportamiento del robot móvil denominado Robotino[®] de la empresa FESTO. Este simulador muestra el movimiento del robot en un entorno virtual según lo indicado desde el software Matlab y hace exactamente los mismos movimientos que haría el robot real.

Para llevar a cabo la simulación se define el entorno $x = y = [-1.5, 1.5]$ m con $\Delta x = \Delta y = 0.1$ m, las restricciones de velocidad $\dot{x} = \dot{y} = [-0.2, 0.2]$ m/s con $\Delta \dot{x} = \Delta \dot{y} = 0.05$ m/s y las de aceleración $u_x = [-0.1, 0.1]$ con $\Delta u = 0.1$ aplicadas durante 1 s. Se construyen los grafos de tiempo, esfuerzo de control y distancia entre celdas por medio de una rutina en lenguaje C. Ésta genera tres matrices dispersas cuadradas que contienen los pesos de todas las conexiones posibles del espacio de trabajo y las restricciones definidas. Los grafos contienen los costos de cada parámetro a optimizar y son en su mayoría elementos no nulos.

Luego se procede a definir un objeto fijo dentro del espacio de trabajo, con el fin de eliminar las conexiones de todas las celdas ocupadas por el obstáculo y su zona de seguridad. Posteriormente se implementa el algoritmo de Dijkstra para encontrar el camino más corto entre la posición inicial $[x_0, y_0] = [-1, -1]$ m y final $[x_1, y_1] = [1, 1]$ m con velocidad inicial y final $(\dot{x}, \dot{y}) = [0, 0]$ m/s. La ruta encontrada es libre de colisiones y optimiza la función de costo $F = 0.7 * F_d + 0.28 * F_u + 0.02 * F_t$.

El móvil utilizado es el que se observa en la Figure 7-21. Éste es un robot móvil omnidireccional de tres ruedas, cuya distancia entre cada rueda y el centro de la plataforma $L = 0.185$ m, radio de cada rueda $r = 0.04$ m. Las ruedas tienen una separación entre sí de 120° y la primera rueda es la que está a $\phi = 180^\circ$ con respecto al eje positivo de la abscisa del marco de referencia del móvil. Para seguir con la numeración de las ruedas se toma en sentido anti horario; por tanto, la segunda rueda estará a un ángulo $\phi = 300^\circ$ y la tercera a $\phi = 60^\circ$.

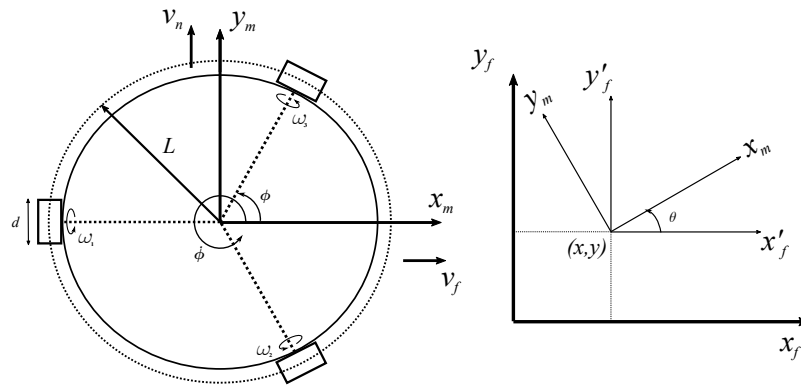


Figura 7-21.: Sistemas coordenados del robot móvil.

En la Figura 7-22 se muestra la trayectoria de referencia obtenida con el AMC, la ruta seguida por el robot móvil en un espacio de trabajo con obstáculo. Para el seguimiento se realizó un control por realimentación de entrada-salida del modelo cinemático. En la Figura 7-23 se muestran las posiciones y velocidades en cada eje dadas por el AMC, la evolución del modelo cinemático en azul punteado y el movimiento seguido por la plataforma omnidireccional en el simulador de color verde.

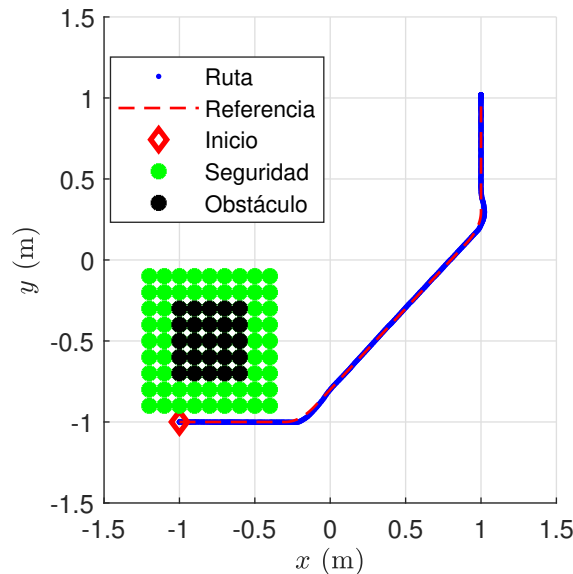


Figura 7-22.: Ruta planificada por el AMC y posición en el plano del Robotino®.

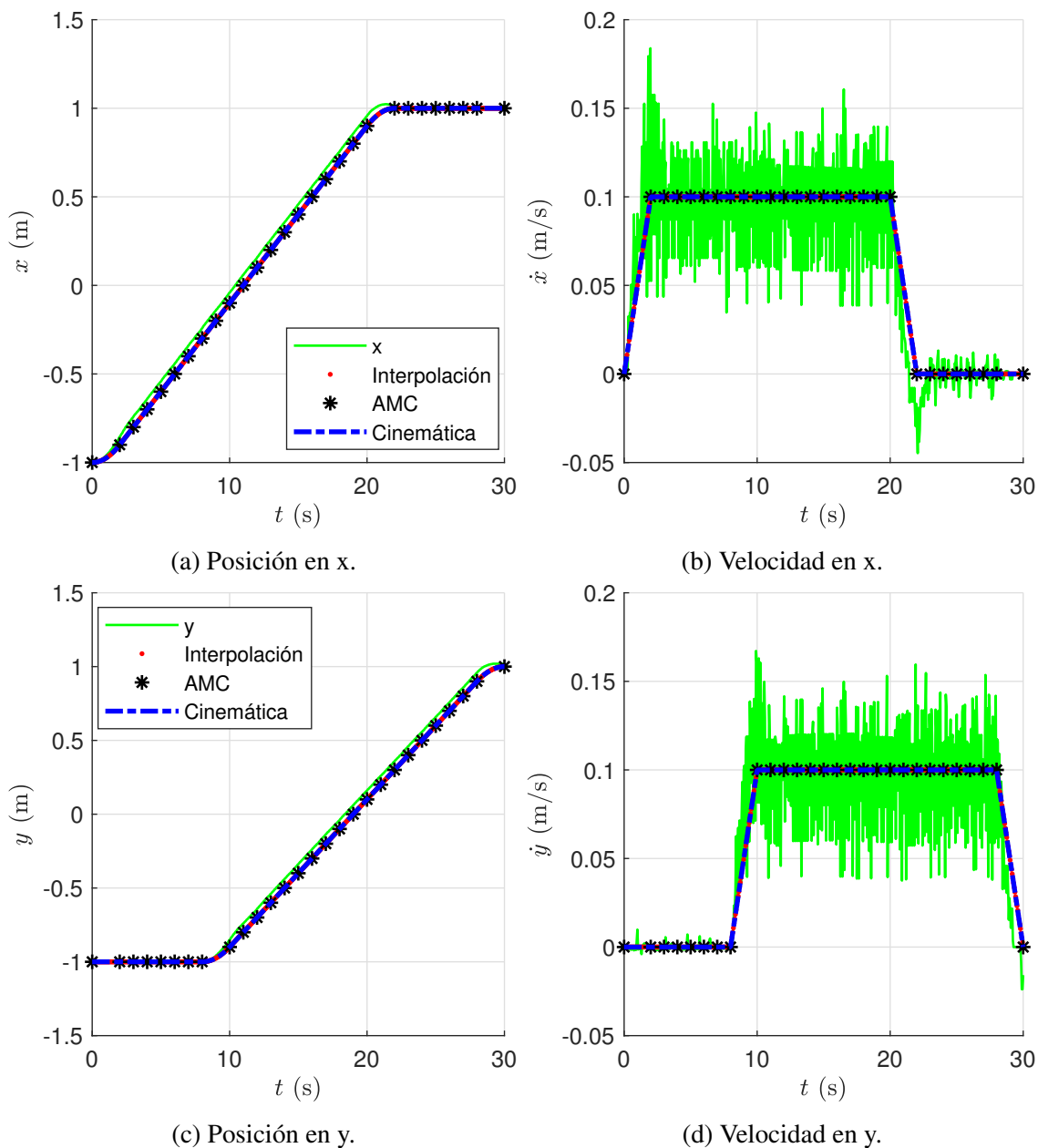


Figura 7-23.: Posiciones y velocidades dadas por el AMC y el Robotino[®].

El Robotino[®] tiene 3 ruedas omnidireccionales que permiten el desplazamiento hacia cualquier dirección sin tener que cambiar la orientación. Este comportamiento es similar al que presenta el punto P_r del modelo dinámico del robot móvil de guiado diferencial. Para determinar las velocidades angulares de cada rueda en el robot es necesario implementar la Ec. (7-4), la cual realiza la transformación de las velocidades lineales y angulares de la plataforma al movimiento necesario de cada una de las ruedas del móvil. En la Figura 7-24 se muestran las velocidades angulares de cada rueda obtenidas en el simulador para el seguimiento de la trayectoria planificada por el AMC.

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 0 & -1 & L \\ \sqrt{3}/2 & 1/2 & L \\ -\sqrt{3}/2 & 1/2 & L \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (7-4)$$

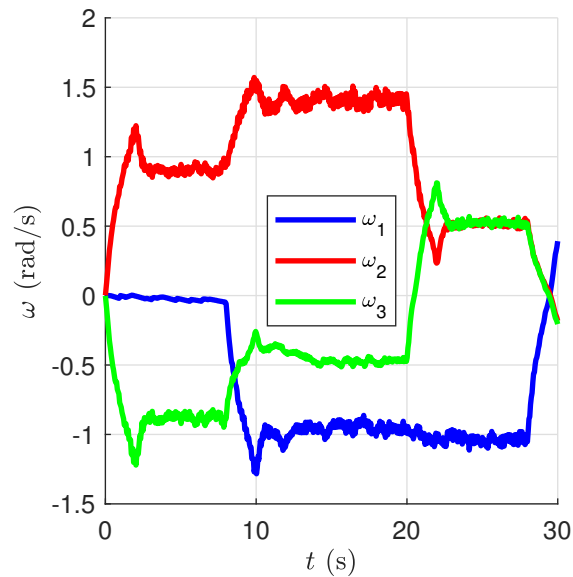


Figura 7-24.: Velocidad angular de cada rueda.

7.4. Conclusión

Como se pudo observar en este capítulo, el AMC implementado puede emplearse para la planificación de trayectorias óptimas de robots móviles. Las condiciones necesarias son: que el sistema pueda ser representado con modelos matemáticos linealizables por medio de algún método analítico o aprendizaje de máquina y que su equivalencia lineal sea análoga al DI para el movimiento en \mathbb{R}^n .

Al AMC puede ser empleado en problemas de optimización multiobjetivo, garantizando una buena aproximación a la solución continua por medio de métodos discretos. El algoritmo minimiza diferentes objetivos en la planificación de trayectorias de robots móviles, tales como: distancia de la ruta, tiempo de navegación, esfuerzo de control, entre otros.

Desde la discretización del problema continuo se definen los límites de velocidad y aceleración lineal del móvil, así como el tiempo de evolución del sistema. Por lo que las rutas encontradas en la solución son suaves en las variables de estado y generan perfiles de velocidad, que dependiendo el radio de las ruedas son físicamente realizables por el robot sin tener problemas de saturación en los actuadores.

Al tener un planificador que utiliza el modelo dinámico del robot, se pueden analizar las velocidades angulares de cada rueda y los parámetros dinámicos del móvil. Por tanto, es posible implementar un modelo del gasto energético donde se pueden introducir diferentes fenómenos físicos y generar trayectorias que minimicen este parámetro.

El AMC se puede extender a diferentes robots móviles, empleando el modelo matemático de cada plataforma. También se puede observar que el número de dimensiones del AMC depende de la aplicación en la que se implemente, para este trabajo se utilizó una partícula que se mueve en una dimensión y robots que se desplazan en el plano, dejando la posibilidad de un trabajo futuro para robots espaciales.

Bibliografía

- [1] A. Ollero B. *Libro: Robótica: Manipuladores y Robots Móviles*. Marcombo, 2005.
- [2] J. F. Camarena. Tesis de maestría: Análisis cinemático, dinámico y control en tiempo real de un vehículo guiado automáticamente, 2009.
- [3] Y. Goto and A. Stentz. Mobile robot navigation: The CMU system. *IEEE Intelligent Systems*, 2(4), 1987.
- [4] S. M. LaValle. Motion planning part 1: The essentials. *IEEE Robotics & Automation Society Magazine*, 18:79–89, 2011.
- [5] R. A. Jarvis. Collision-free trajectory planning using the distance transforms. *Mechanical Engineering Transactions*, 1985.
- [6] S. M. LaValle. Motion planning part 2: Wild frontiers. *IEEE Robotics & Automation Society Magazine*, 18:108–118, 2011.
- [7] A. Srivastava, D. Kartikey, U. Srivastava, V. Srivastava, and S. Rajesh. Non holonomic shortest robot path planning in a dynamic environment using polygonal obstacles. pages 553–558. Proceedings of the IEEE International Conference on Industrial Technology, 2010.
- [8] G. Arechavaleta. Optimización de trayectorias para sistemas sujetos a restricciones no holónomas. *Computación y Sistemas*, 14:365–382, 2011.
- [9] G. D. S. Lee, K. S. Lee, H. G. Park, and M. H. Lee. Optimal path planning with holonomic mobile robot using localization vision sensors. pages 1883–1886. Proceedings of the Control Automation and Systems (ICCAS), 2010.
- [10] N. Ganganath, C. Cheng, and C. K. Tse. An aco-based off-line path planner for nonholonomic mobile robots. pages 1038–1041. Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), 2014.
- [11] S. Gay, S. Dégallier, U. Pattacini, A. Ijspeert, and J. S. Victor. Integration of vision and central pattern generator based locomotion for path planning of a non-holonomic crawling humanoid robot. pages 183–189. Proceedings of the IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems (IROS), 2010.

-
- [12] A. Masoud. Kinodynamic motion planning. *IEEE Robotics and Automation Magazine*, 17:1048–1066, 2010.
- [13] R. V. Cowlagi and P. Tsiotras. Kinematic feasibility guarantees in geometric path planning using history-based transition costs over cell decompositions. pages 5388–5393. Proceedings of the American Control Conference (ACC), 2010.
- [14] O. Castillo, L. Trujillo, and P. Melin. Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots. *Soft Computing*, 11:269–279, 2007.
- [15] J. K. Goyal and K. S. Nagla. A new approach of path planning for mobile robots. pages 863–867. International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2014.
- [16] J. Guo, L. Liu, Q. Liu, and Y. Qu. An improvement of d* algorithm for mobile robot path planning in partial unknown environment. volume 3, pages 10–13. Proceedings of the 2nd International Conference on Intelligent Computation Technology and Automation, 2009.
- [17] A. Elshamli, H. A. Abdullah, and S. Areibi. Genetic algorithm for dynamic path planning. volume 2, pages 677–680. Proceedings of the Canadian Conference on Electrical and Computer Engineering, 2004.
- [18] C. E. Thomaz and M. Vellasco. Mobile robot path planning using genetic. *World Wide Web Internet And Web Information Systems*.
- [19] J. Lu and D. Yang. Path planning based on double-layer genetic algorithm. Proceedings of the 3rd International Conference on Natural Computation (ICNC), 2007.
- [20] F. Lingelbach. Path planning using probabilistic cell decomposition. page 467–472 Vol.1. IEEE International Conference on Robotics and Automation (ICRA), 2004.
- [21] M. Piaggio and R. Zaccaria. Using roadmaps to classify regions of space for autonomous robot navigation. *Robotics and Autonomous Systems*, 25:209–217, 1998.
- [22] S. X. Yang and C. Luo. A neural network approach to complete coverage path planning. *IEEE Transactions on Systems*, 34:718–725, 2004.
- [23] B. Shi, P. Cheng, and N. Cheng. 3d flight path planning based on rrt for rnp requirements. pages 51–56. Proceedings of the IEEE International Conference on Information and Automation, 2012.
- [24] G. F. Luger. *Libro: Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Addison Wesley, 2005.
- [25] S. M. LaValle. *Libro: Planning algorithms*. Cambridge University Press, 2006.

- [26] G. I. R. K. Galgamuwa, L. K. G. Liyanage, M. P. B. Ekanayake, and B. G. L. T. Samaranyake. Simplified controller for three wheeled omni directional mobile robot. pages 314–319. Proceedings of the IEEE 10th International Conference on Industrial and Information Systems (ICIIS), 2016.
- [27] P. F. Islas García. Tesis de pregrado: Sistema de control para el desplazamiento omnidireccional de un robot móvil, 2012.
- [28] D. Cong, C. Liang, Q. Gong, X. Yang, and J. Liu. Path planning and following of omnidirectional mobile robot based on b-spline. pages 4931–4936. Proceedings of the Chinese Control And Decision Conference (CCDC), 2018.
- [29] J. Wang, S. A. Chepinskiy, A. J. Krasnov, B. Zhang, H. Liu, Y. Chen, and D. A. Khvostov. Geometric path following control for an omnidirectional mobile robot. pages 1063–1068. Proceedings of the 21st International Conference on Methods & Models in Automation & Robotics (MMAR), 2016.
- [30] H. Kim and B. K. Kim. Online minimum-energy trajectory planning and control on a straight-line path for three-wheeled omnidirectional mobile robots. *IEEE Transactions on Industrial Electronics*, 61:4771–4779, 9 2014.
- [31] N. M. Abdul Ghani, D. Ju, H. Z. Othman, and M. A. Ahmad. Two wheels mobile robot using optimal regulator control. pages 1066–1070. Proceedings of the 10th International Conference on Intelligent Systems Design and Applications (ISDA), 2010.
- [32] Y. Liu, J. J. Zhu, R. L. Williams, and J. Wu. Omni-directional mobile robot controller based on trajectory linearization. *Robotics and Autonomous Systems*, 56:461–479, 5 2008.
- [33] M. S. Masmoudi, N. Krichen, M. Masmoudi, and N. Derbel. Fuzzy logic controllers design for omnidirectional mobile robot navigation. *Applied Soft Computing*, 49:901–919, 12 2016.
- [34] J. Åkesson, A. Blomdell, and R. Braun. Design and control of yaip - an inverted pendulum on two wheels robot. pages 2178–2183. Proceedings of the IEEE International Conference on Control Applications, 2006.
- [35] J. Li, X. Gao, Q. Huang, and O. Matsumoto. Controller design of a two-wheeled inverted pendulum mobile robot. pages 7–12. Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA), 2008.
- [36] Y. Kim, S. H. Kim, and Y. K. Kwak. Dynamic analysis of a nonholonomic two-wheeled inverted pendulum robot. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 44:25–46, 9 2005.

-
- [37] Y. Kim, S. H. Lee, and D. H. Kim. Dynamic equations of a wheeled inverted pendulum with changing its center of gravity. pages 853–854. Proceedings of the International Conference on Control, Automation and Systems, 2011.
- [38] Y. Watanabe, T. Kanada, and G. Chen. Robust h2 control for two-wheeled inverted pendulum using lego mindstorms. Proceedings of the Australian Control Conference, 2011.
- [39] C. H. Chiu and Y. F. Peng. Design and implement of the self-dynamic controller for two-wheel transporter. pages 480–483. Proceedings of the IEEE International Conference on Fuzzy Systems, 2006.
- [40] P. K. W. Abeygunawardhana and M. Toshiyuki. Stability improvement of two wheel mobile manipulator by real time gain control technique. pages 79–84. Proceedings of the 2nd International Conference on Industrial and Information Systems (ICIIS), 2007.
- [41] C. Acar and T. Murakami. Underactuated two-wheeled mobile manipulator control using nonlinear backstepping method. pages 1680–1685. Proceedings of the Industrial Electronics Conference, 2008.
- [42] P. K. W. Abeygunawardhana, M. Defoort, and T. Murakami. Self-sustaining control of two-wheel mobile manipulator using sliding mode control. pages 792–797. Proceedings of the International Workshop on Advanced Motion Control (AMC), 2010.
- [43] C. Acar and T. Murakami. Motion control of dynamically balanced two-wheeled mobile manipulator through cog manipulation. pages 715–720. Proceedings of the International Workshop on Advanced Motion Control (AMC), 2010.
- [44] S. Ahmad, N. H. Siddique, and M. O. Tokhi. A modular fuzzy control approach for two-wheeled wheelchair. *Intelligent and Robotic Systems: Theory and Applications*, 64:401–426, 12 2011.
- [45] C. Acar and T. Murakami. Multi-task control for dynamically balanced two-wheeled mobile manipulator through task-priority. pages 2195–2200. Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE), 2011.
- [46] S. Ahmad, M. Aminuddin, and M. A. Sadiq M. Shukor. Modular hybrid control for double-link two-wheeled mobile robot. pages 807–813. Proceedings of the International Conference on Computer and Communication Engineering (ICCCCE), 2012.
- [47] R. P. M. Chan, K. A. Stol, and C. R. Halkyard. Review of modelling and control of two-wheeled robots. *Annual Reviews in Control*, 37:89–103, 4 2013.
- [48] C. Li, F. Li, S. Wang, F. Dai, Y. Bai, X. Gao, and L. Kejie. Dynamic adaptive equilibrium control for a self-stabilizing robot. pages 609–614. Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), 2010.

- [49] S. Lee and S. Jung. Novel design and control of a home service mobile robot for korean floor-living life style: Koboker. pages 863–867. Proceedings of the 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), 2011.
- [50] S. Kim and S. J. Kwon. Sdre based nonlinear optimal control of a two-wheeled balancing robot. *Journal of Institute of Control, Robotics and Systems*, 17:1037–1043, 10 2011.
- [51] Z. Kausar, K. Stol, and N. Patel. Performance enhancement of a statically unstable two wheeled mobile robot traversing on an uneven surface. pages 156–162. Proceedings of the IEEE Conference on Robotics, Automation and Mechatronics (RAM), 2010.
- [52] Z. Kausar, K. Stol, and N. Patel. Stability region estimation of statically unstable two wheeled mobile robots. pages 1379–1384. 2011 IEEE International Conference on Robotics and Biomimetics, ROBIO, 2011.
- [53] Z. Kausar, K. Stol, and N. Patel. Nonlinear control design using lyapunov function for two-wheeled mobile robots. pages 123–128. Proceedings of the 19th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), 2012.
- [54] C. S. Hsu. *Libro: Cell-to-Cell mapping: A method of global analysis for nonlinear systems*. Springer, 1987.
- [55] J. Q. Sun, F. R. Xiong, O. Schütze, and C. Hernández. *Libro: Cell mapping methods*. Springer Singapore, 2019.
- [56] C. Hernández, Y. Naranjani, Y. Sardahi, W. Liang, O. Schütze, and J. Q. Sun. Simple cell mapping method for multi-objective optimal feedback control design. *International Journal of Dynamics and Control*, 1:231–238, 9 2013.
- [57] F. Y. Wang and P. J. A. Lever. A cell mapping method for general optimum trajectory planning of multiple robotic arms. *Robotics and Autonomous Systems*, 12:15–27, 3 1994.
- [58] C. S. Hsu. A discrete method of optimal control based upon the cell state space concept. *Journal of Optimization Theory and Applications*, 46:547–569, 1985.
- [59] A. Locatelli. *Libro: Optimal control of a double integrator*. Springer Cham, 2017.
- [60] F. Gordillo-Álvarez. Contribuciones al problema del control óptimo, 1994.
- [61] S. S. Rao. *Libro: Engineering optimization: Theory and practice*. John Wiley & Sons, 2019.
- [62] D. E. Kirk. *Libro: Optimal Control Theory: An Introduction*. 2004.
- [63] R. Szabolcsi. Robust lqg controller design for the small unmanned aerial vehicle. *Review of the Air Force Academy*, pages 31–38, 2018.

- [64] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38:3–20, 1 2002.
- [65] L. C. Evans. *Libro: An introduction to mathematical optimal control theory version 0.2*. 1983.
- [66] R. Loxton, Q. Lin, and K. L. Teo. Minimizing control variation in nonlinear optimal control. *Automatica*, 49:2652–2664, 2013.
- [67] C. Yalçın-Kaya. Optimal control of the double integrator with minimum total variation. *Optimization Theory and Applications*, pages 966–981, 2020.
- [68] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, E. F. Mishchenko, K. N. Trirogoff, and L. W. Neustadt. *Libro: L.S. Pontryagin selected works: The mathematical theory of optimal processes*. Routledge, 2000.
- [69] A. T. Fuller. Relay control systems optimized for various performance criteria. *IFAC Proceedings Volumes*, 1:520–529, 1960.
- [70] D. Liberzon. *Libro: Calculus of variations and optimal control theory*. Princeton University Press, 2011.
- [71] Y. N. Kiselev, M. V. Orlov, and S. M. Orlov. A solution to fuller’s problem using constructions of pontryagin’s maximum principle. *Moscow University Computational Mathematics and Cybernetics*, 42:152–162, 2018.
- [72] M. Caponigro, R. Ghezzi, B. Piccoli, and E. Trélat. Regularization of chattering phenomena via bounded variation controls. *IEEE Transactions on Automatic Control*, 63:2046–2060, 2018.
- [73] A. T. Fuller. Minimization of various performance indices for a system with bounded control. *International Journal of Control*, 41:1–37, 1985.
- [74] M. Dellnitz, S. Ober-Blöbaum, M. Post, O. Schütze, and B. Thiere. A multi-objective approach to the design of low thrust space trajectories using optimal control. *Celestial Mechanics and Dynamical Astronomy*, 105:33–59, 9 2009.
- [75] J. Lagunas-Jiménez, J. R. Lagunas-Jiménez, V. M. Moo-Yam, and B. Ortíz-Moctezuma. Two-degrees-of-freedom robust pid controllers tuning via a multiobjective genetic algorithm. *Computación y Sistemas*, 18:259–273, 6 2014.
- [76] O. Schutze, M. Vasile, O. Junge, M. Dellnitz, and D. Izzo. Designing optimal low-thrust gravity-assist trajectories using space pruning and a multi-objective approach. *Engineering Optimization*, 41:155–181, 2 2009.

- [77] S. Koziel and A. Bekasiewicz. Multi-objective optimization of expensive electromagnetic simulation models. *Applied Soft Computing*, 47:332–342, 2016.
- [78] L. Jourdan, O. Schütze, T. Legrand, E. G. Talbi, and J. L. Wojkiewicz. An analysis of the effect of multiple layers in the multi-objective design of conducting polymer composites. *Materials and Manufacturing Processes*, 24:350–357, 3 2009.
- [79] D. Temple and M. Colette. A goal-programming enhanced collaborative optimization approach to reducing lifecycle costs for naval vessels. *Structural and Multidisciplinary Optimization*, 53:1261–1275, 2016.
- [80] Y. Chen and B. Peng. Multi-objective optimization on multi-layer configuration of cathode electrode for polymer electrolyte fuel cells via computational-intelligence-aided design and engineering framework. *Applied Soft Computing*, 43:357–371, 2016. Acceptance from webpage.
- [81] S. Panda. Multi-objective pid controller tuning for a facts-based damping stabilizer using non-dominated sorting genetic algorithm-ii. *International Journal of Electrical Power and Energy Systems*, 33:1296–1308, 9 2011.
- [82] I. Chiha, N. Liouane, and P. Borne. Tuning pid controller using multiobjective ant colony optimization. *Applied Computational Intelligence and Soft Computing*, 2012:1–7, 2012.
- [83] M. Khoie, K. Salahshoor, E. Nouri, and A. K. Sedigh. Pid controller tuning using multi-objective optimization based on fused genetic-immune algorithm and immune feedback mechanism. volume 6839 LNAI, pages 267–276. Proceedings of the IEEE International Conference on Mechatronics and Automation, 2011.
- [84] Kai Zhang, Minshi Chen, Xin Xu, and Gary G. Yen. Multi-objective evolution strategy for multimodal multi-objective optimization. *Applied Soft Computing*, 101:107004, 2021.
- [85] L. F. T. M. I. Solihin and M. L. Kean. Tuning of pid controller using particle swarm optimization (pso). Proceedings of the International Conference on Advanced Science, Engineering and Information Technology, 2011.
- [86] J. Fliege and B. F. Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51:479–494, 2000.
- [87] P. A. N. Bosman. On gradients and hybrid evolutionary algorithms for real-valued multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 16:51–69, 2 2012.
- [88] X. Zhong, W. Fan, J. Lin, and Z. Zhao. A novel multi-objective compass search. volume 1, pages 24–29, 2010.
- [89] V. Pareto. *Libro: Manual of political economy*. A. M. Kelley, 1971.

- [90] C. Hernández O. Schütze and J. Q. Sun. Capítulo: Global multi-objective optimization by means of cell mapping techniques, 2017.
- [91] J. Q. Sun, F. R. Xiong, O. Schütze, and C. Hernández. *Libro: Cell mapping methods: Algorithmic approaches and applications*. Springer Singapore, 6 2018.
- [92] P. J. Zufiria and R. S. Guttalu. The adjoining cell mapping and its recursive unraveling, part i: Description of adaptive and recursive algorithms. *Nonlinear Dynamics*, 4:207–226, 6 1993.
- [93] C. S. Hsu. A probabilistic theory of nonlinear dynamical systems based on the ceil state space concept. *Journal of Applied Mechanics, Transactions ASME*, 49:895–902, 12 1982.
- [94] C. S. Hsu. Global analysis of dynamical systems using posets and digraphs. *International Journal of Bifurcation and Chaos*, 05:1085–1118, 8 1995.
- [95] J. Petersen. Die theorie der regulären graphs. *Acta Mathematica*, 15:193–220, 12 1891.
- [96] W. Li, L. Xia, and Y. Huang. An adaptive ant colony optimization in knowledge graphs. pages 26–32. Proceedings of the 11th IEEE International Conference on Knowledge Graph (ICKG), 2020.
- [97] A. Agrawal, B. Dixit, V. U. Karve, and B. R. Chandavarkar. Optimizing set of paths connecting multiple source-sink pairs. pages 833–838. Proceedings of the IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, RTEICT, 2017.
- [98] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 12 1959.
- [99] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4:100–107, 1968.
- [100] C. Zhong, S. Liu, B. Zhang, Q. Lu, J. Wang, Q. Wu, and F. Gao. A fast on-line global path planning algorithm based on regionalized roadmap for robot navigation. *IFAC-PapersOnLine*, 50:319–324, 2017.
- [101] J. M. Keil. Decomposing a polygon into simpler components. *SIAM Journal on Computing*, 14:799–817, 1985.
- [102] H. Y. Zhang, W. M. Lin, and A. X. Chen. Path planning for the mobile robot: A review. *Symmetry*, 10, 2018.
- [103] J. R. Sánchez-Ibáñez, C. J. Pérez del Pulgar, and A. García-Cerezo. Path planning for autonomous mobile robots: A review. *Sensors*, 21, 2021.

- [104] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser. Heuristic approaches in robot path planning: A survey. *Robotics and Autonomous Systems*, 86:13–28, 2016.
- [105] S. Mirjalili and J. S. Dong. Introduction to nature-inspired algorithms. In *Nature-Inspired Optimizers*, pages 1–5. Springer, 2020.
- [106] D. Tonon, M. Soledad-Aronna, and D. Kalise. *Libro: Optimal control: Novel directions and applications*, volume 1. Springer, 2017.
- [107] D. Filliat and J. A. Meyer. Map-based navigation in mobile robots: I. a review of localisation strategies. *Journal of Cognitive Systems Research*, 4:243–282, 2003.
- [108] T. Asano, T. Asano, L. Guibas, J. Hershberger, and H. Imai. Visibility-polygon search and euclidean shortest paths. pages 155–164. Proceedings of the Annual Symposium on Foundations of Computer Science, 1985.
- [109] K. Su, T. Phan, C. Yang, and W. Wang. Image-based smooth path planning for wheeled robot. pages 203–207. Proceedings of the 11th IEEE International Conference on Control & Automation (ICCA), 2014.
- [110] O. Takahashi and R. J. Schilling. Motion planning in a plane using generalized voronoi diagrams. *IEEE Transactions on Robotics and Automation*, 5:143–150, 1989.
- [111] J. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40:1528–1538, 1995.
- [112] Z. Dong, Z. Chen, R. Zhou, and R. Zhang. A hybrid approach of virtual force and a* search algorithm for uav path re-planning. pages 1140–1145. Proceedings of the 6th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2011.
- [113] A. Stentz. Optimal and efficient path planning for partially-known environments. pages 3310–3317. Proceedings of the IEEE International Conference on Robotics and Automation, 1994.
- [114] Y. Toda and N. Kubota. Path planning using multi-resolution map for a mobile robot. pages 1276–1281. Proceedings of the SICE Annual Conference, 2011.
- [115] O. Khatib. Capítulo: Real-time obstacle avoidance for manipulators and mobile robots, 1986.
- [116] S. S. Ge and Y. J. Cui. Dynamic motion planning for mobile robots using potential field method. *Autonomous robots*, 13:207–222, 2002.
- [117] P. Vadakkepat, K. C. Tan, and W. Ming-Liang. Evolutionary artificial potential fields and their application in real time robot path planning. volume 1, pages 256–263. Proceedings of the Congress on evolutionary computation, 2000.

- [118] R. Raja, A. Dutta, and K. S. Venkatesh. New potential field method for rough terrain path planning using genetic algorithm for a 6-wheel rover. *Robotics and Autonomous Systems*, 72:295–306, 2015.
- [119] Z. Zhou, J. Wang, Z. Zhu, D. Yang, and J. Wu. Tangent navigated robot path planning strategy using particle swarm optimized artificial potential field. *Optik*, 158:639–651, 2018.
- [120] H. H. Triharminto, O. Wahyunggoro, T. Adji, A. I. Cahyadi, and I. Ardiyanto. A novel of repulsive function on artificial potential field for robot path planning. *International Journal of Electrical and Computer Engineering (IJECE)*, 6:3262–3275, 10 2016.
- [121] D. Kim. Escaping route method for a trap situation in local path planning. *International Journal of Control Automation and Systems*, 7:495–500, 5 2009.
- [122] F. Bayat, S. Najafinia, and M. Aliyari. Mobile robots path planning: Electrostatic potential field approach. *Expert Systems with Applications*, 100:68–78, 6 2018.
- [123] G. Bermudez, L. A. Rojas Castellar, H. Montiel, and M. Ceballos. Aplicación del método de campos de potencial artificial para un robot móvil autónomo. *Tecnura*, 7:86–96, 10 2004.
- [124] N. Buniyamin and W. W. Ngah. A simple local path planning algorithm for autonomous mobile robots. *International Journal of Systems Application Engineering and Development*, 5:151–159, 2011.
- [125] V. J. Lumelsky and A. A. Stepanov. Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, 2:403–430, 11 1987.
- [126] I. Kamon, E. Rivlin, and E. Rimon. New range-sensor based globally convergent navigation algorithm for mobile robots. volume 1, pages 429–435. Proceedings of the IEEE International Conference on Robotics and Automation, 1996.
- [127] I. Al-Taharwa, A. Sheta, and M. Al-Weshah. A mobile robot path planning using genetic algorithm in static environment. *Journal of Computer Science*, 4:341–344, 2008.
- [128] S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning. *The annual research report*, 1998.
- [129] X. Yun and Y. Yamamoto. Internal dynamics of a wheeled mobile robot. pages 1288–1294. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1993.
- [130] Y. Yamamoto and X. Yun. Coordinating locomotion and manipulation of a mobile manipulator recommended citation, 1992.

- [131] Y. Mei, Y. H. Lu, Y. C. Hu, and C. S. G. Lee. Energy-efficient motion planning for mobile robots. volume 2004, pages 4344–4349. Proceedings of the IEEE International Conference on Robotics and Automation, 2004.
- [132] Y. Mei, Y. H. Lu, Y. C. Hu, and C. S. G. Lee. A case study of mobile robot’s energy consumption and conservation techniques. volume 2005, pages 492–497. Proceedings of the International Conference on Advanced Robotics (ICAR), 2005.
- [133] C. H. Kim. Phd thesis: Minimum-energy trajectory planning for differential-driven wheeled mobile robot, 2007.
- [134] O. Chuy, E. G. Collins, W. Yu, and C. Ordonez. Power modeling of a skid steered wheeled robotic ground vehicle. pages 4118–4123. Proceedings of the IEEE International Conference on Robotics and Automation, 2009.
- [135] J. Morales, J. L. Martinez, A. Mandow, A. J. Garcia-Cerezo, and S. Pedraza. Power consumption modeling of skid-steer tracked mobile robots on rigid terrain. *IEEE Transactions on Robotics*, 25:1098–1108, 10 2009.
- [136] R. Parasuraman, K. Kershaw, P. Pagala, and M. Ferre. Model based on-line energy prediction system for semi-autonomous mobile robots. volume 2015-September, pages 411–416. Proceedings of the International Conference on Intelligent Systems, Modelling and Simulation (ISMS), 2015.
- [137] S. Dogru and L. Marques. A physics-based power model for skid-steered wheeled mobile robots. *IEEE Transactions on Robotics*, 34:421–433, 4 2018.
- [138] L. Nicholls and Y. Strengers. Robotic vacuum cleaners save energy? raising cleanliness conventions and energy demand in australian households with smart home technologies. *Energy Research and Social Science*, 50:73–81, 4 2019.
- [139] Y. Mei, Y. H. Lu, C. S. G. Lee, and Y. C. Hu. Energy-efficient mobile robot exploration. volume 2006, pages 505–511. Proceedings of the IEEE International Conference on Robotics and Automation, 2006.
- [140] A. Barili, M. Ceresa, and C. Parisi. Energy-saving motion control for an autonomous mobile robot. volume 2, pages 674–676. Proceedings of the IEEE International Symposium on Industrial Electronics, 1995.
- [141] H. Kim and B. K. Kim. Minimum-energy translational trajectory planning for battery-powered three-wheeled omni-directional mobile robots. pages 1730–1735. Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV), 2008.

-
- [142] H. Kim and B. K. Kim. Minimum-energy trajectory planning and control on a straight line with rotation for three-wheeled omni-directional mobile robots. pages 3119–3124. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012.
- [143] S. Liu and D. Sun. Minimizing energy consumption of wheeled mobile robots via optimal motion planning. *IEEE/ASME Transactions on Mechatronics*, 19:401–411, 4 2014.
- [144] A. Stefek, T. van Pham, V. Krivanek, and K. L. Pham. Energy comparison of controllers used for a differential drive wheeled mobile robot. *IEEE Access*, 8:170915–170927, 6 2020.
- [145] M. Wahab, F. Rios-Gutierrez, and A. El-Shahat. Energy modeling of differential drive robots. pages 1–6. Proceedings of the IEEE Southeastcon, 2015.
- [146] M. F. Jaramillo-Morales and J. B. Gomez-Mendoza. Mixed energy model for a differential guide mobile robot. pages 114–119. 2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR), 2018.
- [147] M. F. Jaramillo-Morales, S. Dogru, L. Marques, and J. B. Gomez-Mendoza. Predictive power estimation for a differential drive mobile robot based on motor and robot dynamic models. pages 301–307. Proceedings of the 3rd IEEE International Conference on Robotic Computing (IRC), 2019.
- [148] M. F. Jaramillo-Morales, S. Dogru, J. B. Gomez-Mendoza, and L. Marques. Energy estimation for differential drive mobile robots on straight and rotational trajectories. *International Journal of Advanced Robotic Systems*, 17, 2020.
- [149] E. Grisales-Ramirez, M. F. Jaramillo-Morales, G. Osorio, and J. B. Gomez-Mendoza. Multi-objective optimal path planning for mobile robots using state space cell mapping. Proceedings of the 4th IEEE Colombian Conference on Automatic Control (CCAC), 2019.