

Control de un robot móvil en entornos domésticos

Efraín Grisales Ramírez

Universidad Nacional de Colombia
Sede Manizales
Facultad de Ingeniería y Arquitectura
Departamento de Ingeniería Eléctrica, Electrónica y Computación
Manizales, Colombia
2013

Control de un robot móvil en entornos domésticos

Efraín Grisales Ramírez

Tesis presentada como requisito parcial para optar al título de:
Magister en Ingeniería - Automatización Industrial

Director:

Ph.D. Gustavo Adolfo Osorio Londoño

Línea de Investigación:

Sistemas de Control, Robótica y Automatización

Grupo de Investigación:

Grupo de Percepción y Control Inteligente (PCI)

Universidad Nacional de Colombia

Facultad de Ingeniería y Arquitectura

Departamento de Ingeniería Eléctrica, Electrónica y Computación

Manizales, Colombia

2013

Control of mobile robot in domestic environments

Efraín Grisales Ramírez

A Thesis presented for the degree of:
Master of Engineering - Industrial Automation

Supervisor:
Ph.D. Gustavo Adolfo Osorio Londoño

Research Areas:
Control Systems, Robotics and Automation
Research group:
Perception and Intelligent Control Group (PCI)

Universidad Nacional de Colombia
Faculty of Engineering and Architecture
Department of Electrical, Electronic and Computer Engineering
Manizales, Colombia
2013

A Dios, por haberme brindado fortaleza, salud y sabiduría para concluir esta tesis.

A mi esposa por apoyarme y ser paciente en todo momento.

A mi familia por brindarme la confianza necesaria para ser una gran persona.

“Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber”.

Albert Einstein

Agradecimientos

A mi Director, el Profesor Gustavo Adolfo Osorio Londoño, por sus invaluable contribuciones y su incondicional apoyo durante el desarrollo de esta investigación.

A Colciencias y su programa de Becas Jóvenes Investigadores e Innovadores 2011, por financiar mis estudios de maestría durante 1 año.

A la Universidad Nacional de Colombia, por la formación que recibí en los niveles de pregrado y posgrado.

A los integrantes del Grupo de Trabajo Académico PCI y ABC Dynamics, por su apoyo durante este proceso.

Resumen

El problema de planificación de trayectorias de cobertura completa es de gran importancia dentro de la robótica móvil debido a que es parte fundamental en aplicaciones domésticas e industriales como: aspiradores, barredores, estregadores de piso, limpiadores de fachadas, limpiadores de piscinas, vigilancia, exploración de minas, entre otros. En estas tareas se hace necesario recorrer todo el espacio libre sin colisionar. En la literatura se proponen diferentes técnicas alternativas para la solución del problema, todas ellas obteniendo resultados muy similares. Es por esto que en este documento se usa un método ya reportado y conocido como “ida y vuelta” (de sus siglas en inglés *Back-and-Forth*), con la salvedad que este se modifica para generar una trayectoria suave y así tener una mejor respuesta del controlador implementado. El algoritmo de planificación consiste en hacer un movimiento en zigzag, avanzando la mayor distancia posible y cubriendo todo el espacio libre. El resultado del proceso de planificación es usada como trayectoria de referencia para un robot móvil de guiado diferencial que realiza una tarea de aseo en el área de trabajo.

Los algoritmos de control empleados en la literatura son numerosos, partiendo de técnicas con modelos lineales, no lineales, difusas, algoritmos de control genéticos y redes neuronales. En este caso se optó por una técnica no lineal y discontinua por realimentación de entrada-salida del vector de estados y se aplicó al modelo cinemático para el seguimiento de trayectorias.

En el trabajo de investigación se estudió el comportamiento de un robot móvil de guiado diferencial sobre el plano XY , el cual muestra las simulaciones del modelo cinemático del móvil, el control para el seguimiento de trayectorias y el algoritmo de planificación de cobertura completa con su respectivo seguimiento.

Palabras clave: Algoritmo, cobertura, control, planificación, realimentación, robótica, seguimiento, vehículo.

Abstract

The path planning problem of full coverage is very important in mobile robotics, because these are fundamental in domestic and industrial vacuum cleaners, sweepers, floor cleaners, building structure cleaners, pool cleaners, surveillance, mining exploration, among others. In these tasks is necessary to travel all the free space without colliding. The literature suggests various Alternative techniques to solve the problem, all results obtained so like. That is why this paper uses a method already reported and known as “back and forth”, except that this is modified to generate a smooth trajectory so that, you can have a better driver response implemented. The algorithm consists of an zigzag, progressing as far away possible and covering the entire space. This task is scheduled to be performed on a guide mobile robot that performs a task difference toiletries in a workspace.

The control algorithms used in the literature are numerous techniques starting linear, nonlinear, fuzzy, genetic control algorithms and neural networks. In this if we chose a nonlinear technique and discontinuous feedback input - output and the state vector applied to the kinematic model to track paths.

In the research work we studied the behavior of a mobile robot guidance differential on the XY plane , which shows the kinematic model simulations the cell, the control for trajectory tracking and scheduling algorithm full coverage with its own tracking.

Keywords: Algorithm, control, coverage, feedback, mobile, planning, robotics, tracking.

Contenido

Agradecimientos	IX
Resumen	XI
Contenido	XIII
Lista de Figuras	XV
Lista de Tablas	XVII
Lista de símbolos	XVIII
1. Introducción	1
2. Vehículo de Guiado Diferencial	4
2.1. Configuraciones de robots móviles	4
2.2. Modelado matemático de la cinemática de un robot móvil	5
2.3. Análisis de resultados	8
2.3.1. Movimiento lineal	8
2.3.2. Movimiento circular anti-horario	9
2.3.3. Movimiento circular horario	11
2.3.4. Movimiento de rotación sobre su propio eje	12
2.4. Conclusiones	14
3. Control discontinuo del modelo cinemático	15
3.1. Ley de control	16
3.2. Seguimiento de trayectorias con control discontinuo	17
3.2.1. Linealización entrada-salida por realimentación dinámica	17
3.2.2. Linealización por realimentación estática	19
3.2.3. Control discontinuo del modelo cinemático	20
3.3. Resultados	21
3.3.1. Trayectoria circular	21
3.3.2. Trayectoria en espiral	24
3.4. Conclusiones	27

4. Algoritmos de cobertura completa para tareas de aseo	29
4.1. Sistemas de navegación autónoma	30
4.2. Planeación de trayectorias de cobertura completa	31
4.3. Resultados	33
4.3.1. Escena 1	33
4.3.2. Escena 2	37
4.4. Conclusiones	41
5. Conclusiones y recomendaciones	42
5.1. Conclusiones	42
5.2. Recomendaciones	43
A. Anexo: Modelo cinemático del robot	45
A.1. Método de Euler	45
A.2. Modelo matemático del robot de guiado diferencial	47
B. Anexo: Control discontinuo del modelo cinemático	49
B.1. Programa principal	49
B.2. Función plantaRB	57
B.3. Función derivada	57
B.4. Función seguimiento1	58
C. Anexo: Algoritmo para cobertura completa	59
C.1. Función principal	59
C.2. Función de movimiento horizontal	66
C.3. Función de movimiento curvo	69
C.4. Función generadora de curvas de Bezier	72
Bibliografía	74

Lista de Figuras

2-1. Modelo de guiado diferencial.	6
2-2. Variables en el espacio de estados.	9
2-3. Variables de estado en el tiempo.	9
2-4. Variables en el espacio de estados.	10
2-5. Variables de estado en el tiempo.	10
2-6. Variables en el espacio de estados.	12
2-7. Variables de estado en el tiempo.	12
2-8. Variables en el espacio de estados.	13
2-9. Variables de estado en el tiempo.	13
3-1. Diagrama de bloques del sistema de control discontinuo.	21
3-2. Trayectoria deseada y obtenida.	22
3-3. Error cuadrático medio de posición.	23
3-4. Error cuadrático medio de orientación.	23
3-5. Velocidad angular en la rueda derecha.	23
3-6. Velocidad angular en la rueda izquierda.	23
3-7. Trayectoria deseada y obtenida.	25
3-8. Error cuadrático medio de posición.	25
3-9. Error cuadrático medio de orientación.	25
3-10.Velocidad angular en la rueda derecha.	26
3-11.Velocidad angular en la rueda izquierda.	26
4-1. Diagrama de estados del algoritmo.	32
4-2. Recorrido de limpieza.	33
4-3. Recorrido inicial.	34
4-4. Trayecto planeado, para realizar la tarea de aseo.	34
4-5. Trayectoria deseada y obtenida.	35
4-6. Error cuadrático medio de posición.	36
4-7. Velocidad angular en la rueda derecha.	37
4-8. Velocidad angular en la rueda izquierda.	37
4-9. Recorrido inicial.	37
4-10.Trayecto planeado, para realizar la tarea de aseo.	38
4-11.Trayectoria deseada y obtenida.	39

4-12. Error cuadrático medio de posición.	40
4-13. Velocidad angular en la rueda derecha.	41
4-14. Velocidad angular en la rueda izquierda.	41

Lista de Tablas

3-1. Medidas de desempeño del controlador.	24
3-2. Medidas de desempeño del controlador.	27
4-1. Acciones a realizar en los estados de la Figura 1.	32
4-2. Comportamiento de la trayectoria.	36
4-3. Comportamiento de la trayectoria.	39

Lista de símbolos

Símbolos con letras latinas

Símbolo	Término	Unidad SI
e	Error de posición del móvil	m
e_x	Error de posición en x del móvil	m
e_y	Error de posición en y del móvil	m
\dot{e}_x	Primera derivada del error de posición en x del móvil	
\dot{e}_y	Primera derivada del error de posición en y del móvil	
\ddot{e}_x	Segunda derivada del error de posición en x del móvil	
\ddot{e}_y	Segunda derivada del error de posición en y del móvil	
l	Distancia entre cada rueda y el centro del vehículo	m
r	Radio de las ruedas	m
v	Velocidad lineal del vehículo	$\frac{m}{seg}$
v_d	Velocidad lineal de la rueda derecha	$\frac{m}{seg}$
v_i	Velocidad lineal de la rueda izquierda	$\frac{m}{seg}$
x	Distancia recorrida en la abscisa	m
y	Distancia recorrida en la ordenada	m
\dot{x}	Velocidad en la dimensión de la abscisa	$\frac{m}{seg}$
\dot{y}	Velocidad en la dimensión de la ordenada	$\frac{m}{seg}$

Símbolos con letras griegas

Símbolo	Término	Unidad SI
α_0	Parámetro del controlador	
α_1	Parámetro del controlador	
β_0	Parámetro del controlador	
β_1	Parámetro del controlador	
φ	Orientación del móvil	<i>rad</i>
$\dot{\varphi}$	Derivada orientación ó velocidad angular	$\frac{rad}{seg}$
ω	Velocidad angular	$\frac{rad}{seg}$
ω_d	Velocidad angular rueda derecha	$\frac{rad}{seg}$
ω_i	Velocidad angular rueda izquierda	$\frac{rad}{seg}$
ζ	Velocidad lineal (u_1)	$\frac{m}{seg}$
$\dot{\zeta}$	\tilde{u}_1	

Abreviaturas

Abreviatura	Término
-------------	---------

<i>CCPP</i>	Planificación de Trayectorias de Cobertura Completa
<i>CCPPP</i>	Problema de Planificación de Trayectorias de Cobertura Completa
<i>ABNN</i>	A Bioinspired Neural Network
<i>AED</i>	Algoritmo de Extension Dinámica

1. Introducción

Dentro de la robótica, los vehículos guiados automáticamente o robots móviles, son un tema de investigación interesante por varias razones. Primero, porque debido al grado de inteligencia y todo lo que implica su implementación, se requiere de la aplicación de técnicas de varias disciplinas, tales como; física, matemáticas, mecánica, electrónica, control automático y computación. Segundo, porque los vehículos guiados automáticamente son una aproximación cercana a la creación de sistemas capaces de procesar la información de su entorno y establecer un comportamiento similar al del ser humano. Tercero, porque las aplicaciones de los vehículos guiados automáticamente son innumerables, tales como: exploración minera y marítima, reconocimiento de terreno, inspección y vigilancia, misiones de búsqueda y rescate de personas, limpieza de entornos, asistencia médica, ocio y entretenimiento, investigación y desarrollo, investigación militar, industria metalmecánica, industria química, agricultura, transporte, entre otros. En complemento y con el objetivo de conformar una herramienta más eficaz, las plataformas también son utilizadas en aplicaciones que requieren de la integración con otros tipos de robots, tal es el caso de los vehículos que incorporan un brazo manipulador utilizados como elementos de seguridad en la inspección y manejo de paquetes sospechosos, neutralización de dispositivos explosivos, manipulación de productos químicos y materiales radiactivos, entre otros.

Por esto son de gran importancia los algoritmos de planificación de trayectorias, que son usados en diferentes aplicaciones en donde se hace necesario seguir un trayecto evitando obstáculos, los cuales pueden ser de diferente naturaleza, aumentando o disminuyendo la complejidad del algoritmo. Una tarea de planificación suele tener tres tipos de resultados: rutas, caminos y trayectorias. Una ruta es el conjunto ordenado de configuraciones que deben ser alcanzadas por el robot, mientras que por camino se entiende la discretización de una función continua que interpola las configuraciones definidas en una ruta y finalmente, cuando se habla de trayectoria se está haciendo referencia a un camino que tiene asociado un perfil cinemático; es decir, a cada configuración perteneciente al camino se le asocia una velocidad.

El resultado de la planificación se implementa en sistemas de navegación controlada. Una característica común de todos estos sistemas, es el uso de un controlador, el cual, en tiempo real permite que el sistema siga la trayectoria previamente obtenida. En el control para el seguimiento de trayectorias se hace necesario conocer la posición del vehículo en el plano XY . En [1] se observa el método de navegación inercial para la localización del móvil.

En los últimos años, una gran cantidad de autores han abordado el Problema de Planificación de Trayectorias de Cobertura Completa (CCPPP), con el fin de implementarlo en robots móviles, en los que se hace necesario recorrer todo el espacio libre para tareas de aseo (aspiradora autónoma [2, 3, 4, 5], barredora automática [6], estregador de pisos), vigilancia, seguridad, limpiador de ventanas [7], limpiador de tanques de almacenamiento [8], entre otros. En estos trabajos se encuentran diferentes métodos que principalmente se basan en la aplicación de algoritmos inspirados en redes neuronales [4, 9, 10, 11, 12, 13], dividiendo el espacio en neuronas libres, neuronas ocupadas y neuronas no exploradas; también se basan en la descomposición de celdas o en formas complejas para seccionar el espacio y clasificarlo, en libre u ocupado, como se ve en [14, 15, 16, 17, 18, 19, 20, 21, 22]. Otros autores utilizan métodos heurísticos, implementan el filtrado de Kalman o realizan la cobertura del espacio con ayuda de cámaras, en [2, 3, 15, 23, 24, 25, 26, 27] se observan algunos trabajos con los métodos mencionados.

En estos trabajos el tipo de móvil más utilizado es el de guiado diferencial, debido a que es de mayor facilidad implementar un algoritmo de control en este tipo de configuración que en otras topologías. En estos el control utilizado varía de acuerdo al autor, siendo más utilizadas técnicas no lineales y control difuso. Los algoritmos de control son empleados para el seguimiento de caminos cuando se realiza la planificación de trayectorias. En algunos casos, para la planificación de dichos trayectos es necesaria la reconstrucción del entorno a priori, el cual ayuda a conocer cual es el espacio libre y cual el ocupado; pero en otros casos no surge la necesidad de reconstrucción, debido a que el robot empieza a desplazarse mapeando el entorno para realizar en línea la planificación.

Para la construcción de este proyecto de investigación, se asume que el robot trabaja sobre el plano XY y se implementan algoritmos en Matlab[®] con el fin de simular el comportamiento de cada uno de los bloques estudiados, partiendo de la premisa de conocer el entorno previamente, es decir, se asume que el espacio libre se conoce y se realiza la Planificación de Trayectorias de Cobertura Completa (CCPP), con ayuda del algoritmo *Back-and-Forth Motion* con la modificación pertinente para generar una trayectoria suave, indispensable si se desea que el error en el proceso de seguimiento sea pequeño. Esta planificación se hizo para implementarlo sobre un robot móvil de guiado diferencial, en un ambiente cerrado libre de obstáculos, el cual debe ser cubierto en su totalidad para llevar a cabo la tarea de aseo correspondiente. El control implementado en este móvil es no lineal, discontinuo y por

realimentación de entrada-salida, en el cual se realiza una conmutación entre dos leyes de control, que actúan sobre el vehículo cuando se presentan diferentes condiciones, con el fin de evitar las singularidades presentadas en cada una de ellas.

El documento de tesis es estructurado de la siguiente forma: en el Capítulo 2, se presentan algunas configuraciones de robots móviles, así como el tipo de ruedas que estos pueden tener. También se muestra el modelo matemático de la cinemática del vehículo de guiado diferencial y algunos resultados en los que se analiza el movimiento ante diferentes condiciones iniciales y las variables de control. En el Capítulo 3, se describe el tipo de controlador utilizado, mostrando la respuesta del sistema cuando se desea seguir diferentes trayectorias. En el Capítulo 4, se muestran algunas investigaciones que abarcan el Problema de Planificación de Trayectorias de Cobertura Completa (CCPPP) y se desarrolla el algoritmo de cobertura completa para realizar tareas de aseo en entornos domésticos, mediante el *Back-and-Forth Motion* modificado. Finalmente en el Capítulo 5 se presentan las conclusiones, recomendaciones y trabajos futuros.

2. Vehículo de Guiado Diferencial

La robótica ha jugado un papel importante durante el desarrollo de la humanidad, y su evolución ha ido siempre de la mano con la construcción de dispositivos que realicen tareas que facilitan el trabajo. El concepto de robot comenzó a escucharse en 1920 del escritor checo Capek, en su obra dramática Rossum's Universal Robots, en la que utilizó el término robot a partir de la palabra checa robot, que significa servidumbre o trabajo forzado. Por su parte, Asimov introdujo, por primera vez, el término robótica, en la historia Runaround de su obra "I, Robot", incluyendo en ese mismo trabajo, las tres leyes de la robótica. Fue así, que como consecuencia de los trabajos que hasta entonces se habían realizado, amparados en el éxito de las obras mencionadas, la robótica acaparó la atención, no sólo de aficionados a la ciencia ficción, sino también de científicos e investigadores que empezaron a trabajar en ésta.

En este capítulo se da una pequeña introducción a la robótica móvil, mostrando las diferentes configuraciones de robots móviles con ruedas, luego se desarrolla el modelo matemático del vehículo de guiado diferencial; éste consta de dos ruedas de tracción y una rueda de apoyo que se usa para equilibrar la plataforma móvil. Se obtiene el modelo cinemático a partir de la teoría de la mecánica clásica, en el que se tiene en cuenta la orientación en el plano cartesiano del vehículo.

2.1. Configuraciones de robots móviles

En [28] se muestran diferentes tipos de configuraciones de robots móviles, tales como: marinos, aéreos, de desplazamiento sobre otros fluidos, de locomoción por medio de patas, de locomoción por bandas y vehículos con ruedas.

Aunque existen diferentes tipos de configuraciones cinemáticas de robots móviles (bípedos, hexápodos, acuáticos, entre otros), las cuales dependen principalmente de la aplicación del vehículo. Este documento se enfocará en los robots móviles sobre ruedas y para aplicaciones en ambientes controlados (interiores), en las que se pueden encontrar las siguientes configuraciones: Ackerman, triciclo clásico, tracción diferencial, skid steer, síncrona y tracción omnidireccional, estas se observan en [29].

Dependiendo de la configuración cinemática que lo conforme, los robots móviles utilizan cuatro tipos de ruedas para su locomoción, estas son: convencionales, tipo castor, ruedas de bolas y omnidireccionales. En [29], se observan los diferentes tipos de ruedas de robots móviles. La configuración de tracción diferencial utiliza ruedas convencionales como ruedas motrices y una o dos ruedas tipo castor, de bola u omnidireccionales para proveer de estabilidad al vehículo. Los robots con dichas características más conocidos en la literatura son: Shakey [30], Newt [31], y Hilare [32].

Las diferentes configuraciones de robots móviles se pueden modelar en función de las coordenadas globales (x,y) , la orientación φ respecto al eje horizontal y las velocidades lineales $v=(v_x,v_y)$ y angulares $\omega=(\omega_x,\omega_y)$ del vehículo. Existen distintas formas de locomoción para robots móviles, esto es de acuerdo a la naturaleza del diseño, condiciones del problema y la topografía del terreno donde se realiza la tarea.

2.2. Modelado matemático de la cinemática de un robot móvil

El análisis cinemático corresponde al de vehículos con ruedas, asumiendo las siguientes condiciones:

- El robot se mueve sobre una superficie plana, libre de irregularidades de nivel y textura.
- Los ejes de desplazamiento son perpendiculares a la superficie de trabajo.
- Se despreciará el deslizamiento del robot durante el movimiento.
- La estructura del robot es totalmente rígida, no se consideran en el diseño partes flexibles.
- Los intervalos de tiempo de desplazamiento describirán trayectorias en arcos de circunferencia.
- El sistema de control asumirá el posicionamiento y orientación del vehículo en su totalidad.

En la Figura 2-1, se ve el modelo del vehículo de guiado diferencial, en el cual las variables de control son las velocidades del vehículo (velocidad angular ω y velocidad lineal v). En la Ec. (2-1) se exponen las variables de estado en función de las velocidades lineales y angulares de cada rueda. La posición y orientación del vehículo puede describirse por las coordenadas (x,y) , que denotan la posición del punto medio del eje que une las dos ruedas traseras, con respecto al sistema de referencia fijo XY , y el ángulo (φ) que forma el eje de simetría del vehículo con respecto al eje X positivo [33, 34].

Donde,

$v(t)$ velocidad lineal del punto medio del vehículo.

$\omega(t)$ velocidad angular del punto medio del vehículo.

$\omega_d(t)$ velocidad angular de la rueda derecha.

$\omega_i(t)$ velocidad angular de la rueda izquierda.

r es el radio de las ruedas.

l distancia de separación entre el eje medio del vehículo y la rueda de tracción.

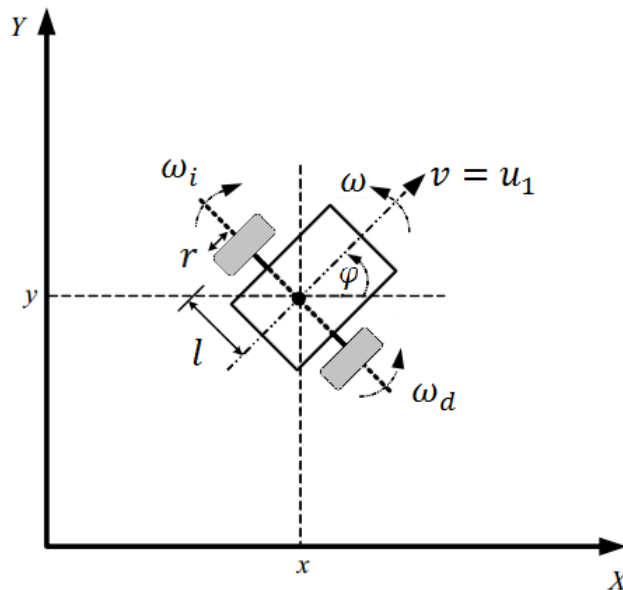


Figura 2-1.: Modelo de guiado diferencial.

$$\begin{aligned} v &= \frac{v_d + v_i}{2} = \frac{(\omega_d + \omega_i)r}{2} \\ \omega &= \frac{v_d - v_i}{2l} = \frac{(\omega_d - \omega_i)r}{2l} \end{aligned} \quad (2-1)$$

El modelo matemático de la cinemática de un robot móvil de guiado diferencial, puede expresarse en términos de las variables de control (v, ω) como se observa en la Ec. (2-2) y luego en velocidades de control (ω_i, ω_d) , como se ve en la Ec. (2-3), en donde r es el radio de las ruedas, l es la mitad de la distancia entre las dos ruedas, \dot{x} y \dot{y} son las velocidades lineales en cada eje y $\dot{\varphi}$ es la velocidad angular del móvil. En [28] se observan las expresiones en función de las variables de control.

$$\begin{aligned} \dot{x} &= v \cos \varphi \\ \dot{y} &= v \sin \varphi \\ \dot{\varphi} &= \omega \end{aligned} \quad (2-2)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \frac{r \cos \varphi}{2} \\ \frac{r \sin \varphi}{2} \\ \frac{-r}{2l} \end{bmatrix} \omega_i + \begin{bmatrix} \frac{r \cos \varphi}{2} \\ \frac{r \sin \varphi}{2} \\ \frac{r}{2l} \end{bmatrix} \omega_d = \begin{bmatrix} \frac{r \cos \varphi}{2} & \frac{r \cos \varphi}{2} \\ \frac{r \sin \varphi}{2} & \frac{r \sin \varphi}{2} \\ \frac{-r}{2l} & \frac{r}{2l} \end{bmatrix} \begin{bmatrix} \omega_i \\ \omega_d \end{bmatrix} \quad (2-3)$$

De la Ec. (2-3), donde se observan las velocidades angulares de la rueda derecha (ω_d) e izquierda (ω_i), se deduce el modelo matemático de la cinemática del sistema de guiado diferencial, el cual se muestra en la Ec. (2-4).

$$\begin{aligned} \dot{x} &= \frac{r}{2} \cos \varphi (\omega_d + \omega_i) \\ \dot{y} &= \frac{r}{2} \sin \varphi (\omega_d + \omega_i) \\ \dot{\varphi} &= \frac{r}{2l} (\omega_d - \omega_i) \end{aligned} \quad (2-4)$$

2.3. Análisis de resultados

Se implementa el algoritmo mostrado en el Anexo A, sobre el software Matlab[®], el cual contiene el modelo matemático mostrado en la Ec. (2-4), correspondiente a la cinemática de un robot móvil de guiado diferencial. Esto con el fin de simular el comportamiento de este vehículo ante diferentes valores de las señales de control (ω_d, ω_i) en lazo abierto.

Realizando la simulación para obtener la posición y orientación seguida por un móvil de guiado diferencial, ante diferentes señales de entrada ω_d y ω_i , y considerando el radio de las ruedas $r = 0,05 \text{ m}$ y la mitad de la separación entre las ruedas $l = 0,15 \text{ m}$, con una posición inicial en el origen de coordenadas y durante un tiempo estimado.

2.3.1. Movimiento lineal

Para que el modelo de guiado diferencial mostrado en la Figura 2-1 describa una trayectoria lineal sobre el plano XY , con una dirección fija, la cual es dada por la condición inicial de la orientación del móvil, que en este caso es $\pi/6$, se debe tener la misma magnitud de velocidad angular en ambas ruedas, es decir, $\omega_d = \omega_i$. En este caso se asignarán los siguientes valores $\omega_d = \omega_i = 1 \text{ rad/seg}$ y al reemplazarlos en el modelo, da como resultado el visto en la Ec. (2-5).

$$\begin{aligned} \dot{x} &= 0,05 \cos \varphi \\ \dot{y} &= 0,05 \sin \varphi \\ \dot{\varphi} &= 0 \end{aligned} \tag{2-5}$$

En la Figura 2-2, se muestra el resultado de la simulación (aplicando el método de Euler) del sistema con los parámetros anteriormente mencionados durante 30 *seg*, en este se observa el gráfico de las variables de estado x , y y φ . Esta presenta un movimiento lineal, manteniendo la orientación constante, debido a que la velocidad angular es igual a cero lo que no cambia el sentido del movimiento. La orientación se mantiene en $\pi/6 \text{ rad}$ condición inicial del móvil, generando un movimiento lineal con orientación constante, lo cual se debe a que las velocidades angulares aplicadas a ambas ruedas son iguales.

En la Figura 2-3, se observa el comportamiento de las variables de estado en el tiempo, las cuales son: la orientación φ , el desplazamiento en x y el desplazamiento en el eje y , en esta se ve como se mantiene constante la orientación ($\pi/6 \text{ rad}$) y las variables x y y crecen de manera lineal.

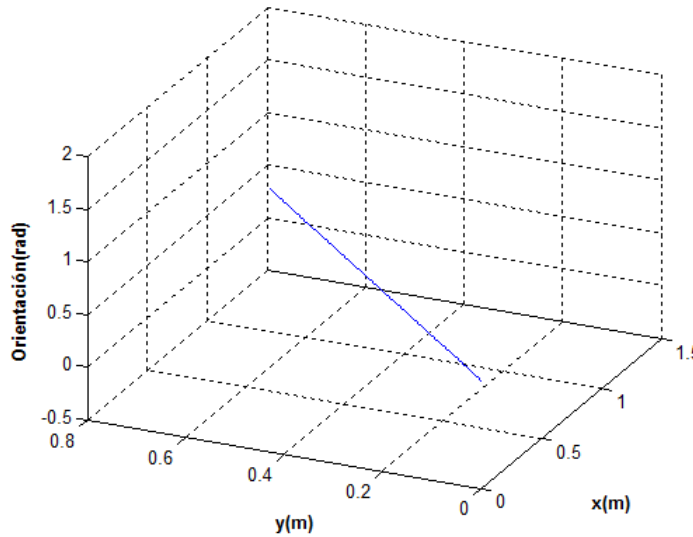


Figura 2-2.: Variables en el espacio de estados.

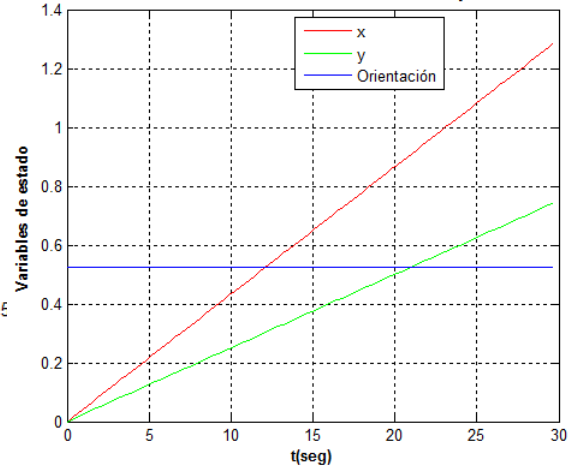


Figura 2-3.: Variables de estado en el tiempo.

2.3.2. Movimiento circular anti-horario

Para que el modelo de guiado diferencial mostrado en la Figura 2-1 describa una trayectoria circular sobre el plano XY , con una dirección inicial, la cual es dada por la condición inicial de la orientación del móvil, que en este caso es $\pi/4$, se debe tener en las ruedas diferentes magnitudes pero igual sentido, es decir, ω_d debe ser mayor que ω_i . En este caso se asignarán los siguientes valores $\omega_d = 1,2 \text{ rad/seg}$ y $\omega_i = 1 \text{ rad/seg}$, dando como resultado el sistema de la Ec. (2-6). Con el fin de ver la circunferencia se aumenta el tiempo de simulación a 380 *seg*.

$$\dot{x} = 0,055 \cos \varphi$$

$$\dot{y} = 0,055 \sin \varphi$$

$$\dot{\varphi} = 0,033$$

(2-6)

En la Figura 2-4, se muestra el resultado de la simulación (aplicando el método de Euler) del sistema con los parámetros anteriormente mencionados, en este se observa el gráfico de las variables de estado x , y y φ . Esta presenta un movimiento circular, incrementando la orientación, debido a que la velocidad angular es constante, lo que genera un incremento en el sentido del movimiento. La orientación empieza en $\pi/4 \text{ rad}$ condición inicial del móvil y aumenta, con el fin de generar un movimiento circular en sentido anti-horario con velocidad angular constante, lo cual se debe a que la velocidad angular aplicada en la rueda derecha es mayor a la aplicada en la izquierda (si esto fuera al contrario, es decir, la velocidad angular en la rueda izquierda mayor a la de la derecha, el movimiento sería circular en sentido horario).

En la Figura 2-5, se observa el comportamiento de las variables de estado en el tiempo, las cuales son: la orientación φ , el desplazamiento en x y el desplazamiento en el eje y , en esta se ve como la orientación del móvil se incrementa iniciando en el valor ($\pi/4 \text{ rad}$) y las variables x y y tienen un comportamiento circular. En esta se alcanza a ver el comportamiento de las variables de estado en el tiempo, en donde se aprecia que x está adelantada a y $\pi/2 \text{ rad}$.

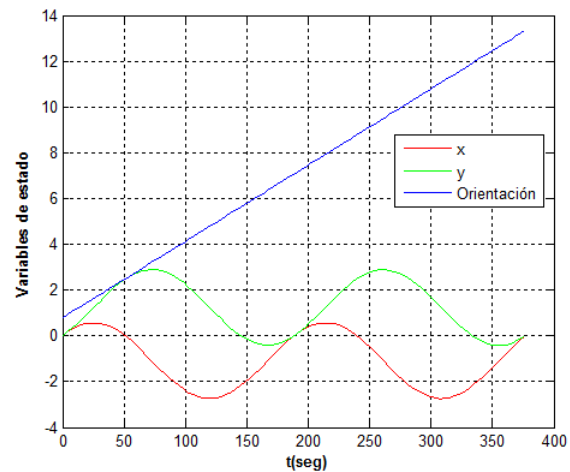
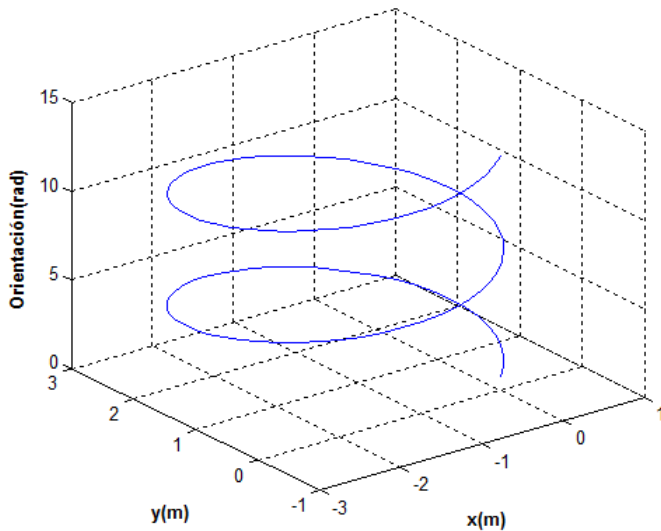


Figura 2-4.: Variables en el espacio de estados.

Figura 2-5.: Variables de estado en el tiempo.

2.3.3. Movimiento circular horario

Para que el modelo de guiado diferencial mostrado en la Figura 2-1 describa una trayectoria circular en sentido horario sobre el plano XY , con una dirección inicial, la cual es dada por la condición inicial de la orientación del móvil, que en este caso es $\pi/4$, se debe tener en las ruedas diferentes magnitudes pero igual sentido, es decir, ω_d debe ser menor que ω_i . En este caso se asignaran los siguientes valores $\omega_d = 1 \text{ rad/seg}$ y $\omega_i = 1,2 \text{ rad/seg}$, dando como resultado el sistema de la Ec. (2-7)

$$\begin{aligned} \dot{x} &= 0,055 \cos \varphi \\ \dot{y} &= 0,055 \sin \varphi \\ \dot{\varphi} &= -0,033 \end{aligned} \tag{2-7}$$

En la Figura 2-6, se muestra el resultado de la simulación (aplicando el método de Euler) del sistema con los parámetros anteriormente mencionados, en esta se observa el gráfico de las variables de estado x , y y φ . La Figura presenta un movimiento circular, disminuyendo la orientación, debido a que la velocidad angular es constante con valor negativo lo que genera un decremento en el sentido del movimiento. La orientación empieza en $\pi/4 \text{ rad}$ condición inicial del móvil y disminuye, con el fin de generar un movimiento circular en sentido horario, lo cual se debe a que la velocidad angular aplicada en la rueda derecha es menor a la aplicada en la izquierda.

En la Figura 2-7, se observa el comportamiento de las variables de estado en el tiempo, las cuales son: la orientación, el desplazamiento en x y el desplazamiento en el eje y , en esta se ve como la orientación del móvil decremента iniciando en el valor $(\pi/4 \text{ rad})$ y las variables x y y tienen un movimiento circular. En esta se alcanza a ver el comportamiento de las variables de estado en el tiempo, en donde se aprecia que y está adelantada a x $\pi/2 \text{ rad}$.

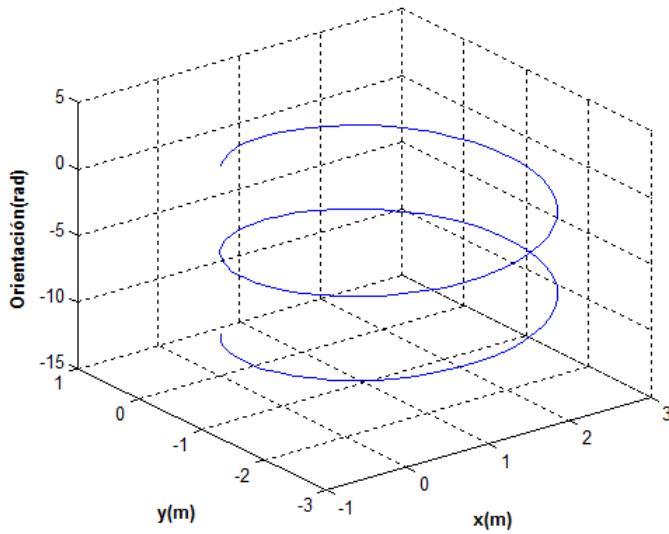


Figura 2-6.: Variables en el espacio de estados.

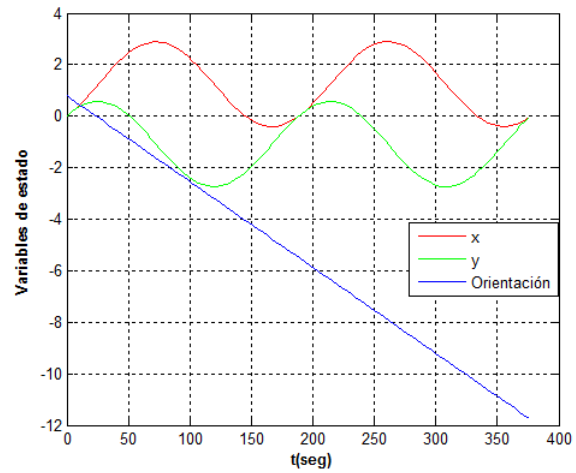


Figura 2-7.: Variables de estado en el tiempo.

2.3.4. Movimiento de rotación sobre su propio eje

Para que el modelo de guiado diferencial mostrado en la Figura 2-1 realice una trayectoria de rotación sobre su propio eje (No hay desplazamiento lineal en el plano XY , sólo hay rotación sobre su propio eje), es necesario que la velocidad angular en la rueda derecha sea igual en magnitud pero diferente en sentido, es decir, $\omega_d = -\omega_i$. La orientación inicial dada es $\pi/4$ y empieza a incrementarse debido a que $\omega_d > \omega_i$. En este caso se asignan los siguientes valores $\omega_d = 1 \text{ rad/seg}$ y $\omega_i = -1 \text{ rad/seg}$, dando como resultado que el modelo matemático del sistema queda dado por la Ec. (2-8).

$$\dot{x} = 0$$

$$\dot{y} = 0$$

$$\dot{\varphi} = 0,33$$

(2-8)

En la Figura 2-8, se muestra el resultado de la simulación (aplicando el método de Euler) del sistema con los parámetros anteriormente mencionados, en este se observa el gráfico de las variables de estado x , y y φ . En la Figura se ve una línea, la cual simboliza que en el plano XY no se presenta movimiento alguno, pero la orientación aumenta linealmente, debido a que la velocidad angular es constante con valor positivo (lo que genera un incremento en la orientación del móvil). La orientación empieza en $\pi/4 \text{ rad}$ (condición inicial del móvil) y aumenta linealmente, con el fin de generar un movimiento de rotación sobre su propio eje (con velocidad angular constante).

En la Figura 2-9, se observa el comportamiento de las variables de estado en el tiempo, las cuales son: la orientación, el desplazamiento en x y el desplazamiento en el eje y , en esta se ve como la orientación del móvil incrementa su valor iniciando en ($\pi/4 \text{ rad}$) y las variables x y y presentan un valor nulo, lo que significa que no hay desplazamiento en el plano XY .

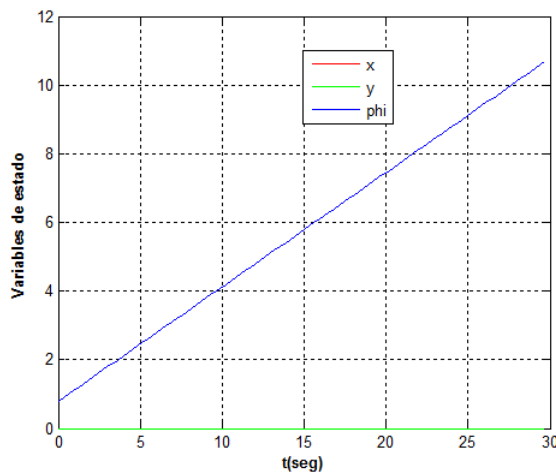
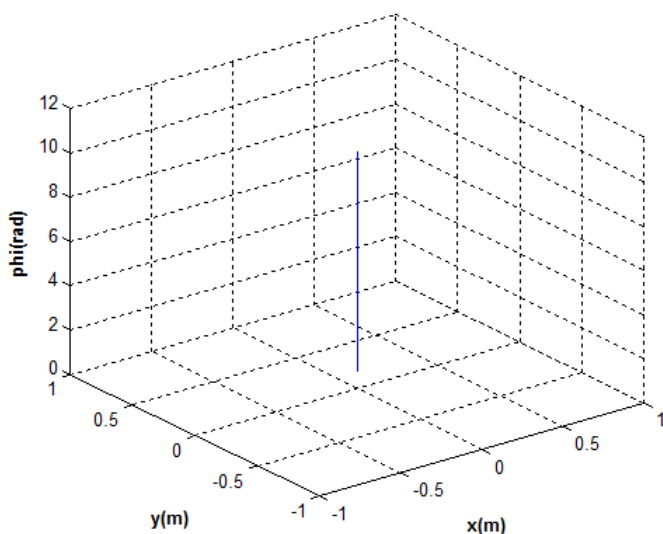


Figura 2-8.: Variables en el espacio de estados.

Figura 2-9.: Variables de estado en el tiempo.

2.4. Conclusiones

Los modelos matemáticos de la cinemática de robots móviles emulan el comportamiento de estos en el plano, lo cual es de gran importancia, debido a que se puede predecir el comportamiento de los vehículos ante condiciones iniciales específicas y valores de las variables de control dados. Esto es necesario cuando se está trabajando sobre estrategias de control para el seguimiento de trayectorias, puesto que se desea ver el comportamiento del móvil y realizar la comparación con una trayectoria planificada y así conocer el comportamiento del sistema para proceder a implementarlo.

En este caso se estudia el modelo cinemático de robots móviles con guiado diferencial, los cuales constan de dos ruedas de tracción y una libre, utilizada de apoyo; la dirección del movimiento es dada por la diferencia de las magnitudes de las velocidades angulares en las ruedas derecha e izquierda del móvil.

3. Control discontinuo del modelo cinemático

Al aplicar una ley de control al robot móvil, el objetivo es realizar el seguimiento exacto o asintótico de una trayectoria preestablecida sobre el plano XY. Este seguimiento lo debe realizar con un punto guía fijo, definido y localizado en la plataforma del vehículo. En otras palabras, hay que diseñar un control que permita llevar a las variables de estado (x, y, φ) del vehículo al valor deseado (x_d, y_d, φ_d) , bajo la condición de que el vehículo se encuentre inicialmente sobre la trayectoria o en un punto de referencia cercano a ella.

Para lograr el seguimiento de la trayectoria, se propone una estrategia de control donde se conmuta entre dos esquemas de control. El primer esquema consiste en la linealización entrada-salida por realimentación estática del vector de estados [35], en este las variables de salida son (x, φ) . El segundo esquema consiste en la linealización entrada-salida por realimentación dinámica del vector de estados, en este las variables de salida son (x, y) del punto guía $P_m(x_m, y_m)$.

Una variante del primer esquema de control propuesto, consiste en seleccionar otro par de salidas, que bien puede ser otro punto guía que se localice en la parte frontal de la plataforma del vehículo denotado como $P_f(x_f, y_f)$. Con esta nueva ley de control se realiza el seguimiento de la trayectoria en todo instante de tiempo. Por consiguiente, para lograr la convergencia de las tres variables de estado del vehículo se diseña un controlador de estructura variable, el cual tiene semejanza al trabajado en [29, 35, 36, 37, 38]. Este controlador cambia entre un control de linealización estático y linealización dinámico. Por tanto, la ley de control implementada en el robot móvil es discontinua y no lineal, ésta se encarga de que el robot siga la trayectoria de referencia, para esto se debe controlar la velocidad angular de las ruedas y así se garantiza el seguimiento de una trayectoria.

3.1. Ley de control

Se parte del modelo mostrado en la Ec. (3-1), donde $u_1 = v$, $u_2 = \omega$, con el cual se puede construir cualquier trayectoria sobre el plano. Debido a que las variables de control en el vehículo de guiado diferencial son ω_d y ω_i , el modelo matemático debe estar en función de estas variables, para esto se reemplaza la Ec. (3-2) en la Ec. (3-1) obteniendo el sistema mostrado en la Ec. (3-3).

$$\begin{aligned} \dot{x} &= u_1 \cos \varphi \\ \dot{y} &= u_1 \sin \varphi \\ \dot{\varphi} &= u_2 \end{aligned} \tag{3-1}$$

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{2l} & -\frac{r}{2l} \end{bmatrix} \begin{bmatrix} \omega_d \\ \omega_i \end{bmatrix} \tag{3-2}$$

$$\begin{aligned} \dot{x} &= \frac{(\omega_d + \omega_i)r}{2} \cos(\varphi) \\ \dot{y} &= \frac{(\omega_d + \omega_i)r}{2} \sin(\varphi) \end{aligned} \tag{3-3}$$

$$\dot{\varphi} = \frac{(\omega_d - \omega_i)r}{2l}$$

En la Ec. (3-3) se expresa el modelo matemático de un robot móvil de guiado diferencial visto en la Figura 2-1, en el cual se desea implementar una ley de control para el seguimiento de trayectorias. En este sistema las variables de control son: ω_i y ω_d , velocidad angular de la rueda izquierda y la rueda derecha respectivamente, las cuales deben ser modificadas por el controlador para seguir el camino deseado.

En los robots móviles orientables sin ruedas las variables de control son: v (velocidad lineal) y ω (velocidad angular), con las cuales se puede construir casi cualquier trayectoria en el plano. Si éstas variables son las de control en el modelo, se necesita realizar una transformación para tenerlas en función de las velocidades angulares de cada rueda y así aplicarlas al sistema de guiado diferencial.

En la Ec. (3-2), se define la velocidad lineal y angular en función de las variables de control ω_i y ω_d , esto con el fin de aplicar el algoritmo de control al modelo de guiado diferencial. Si se despejan las variables de control ω_i y ω_d de la Ec. (3-2), estas quedan en función de las velocidades lineal y angular, generando el modelo de la Ec. (3-4).

$$\begin{bmatrix} \omega_d \\ \omega_i \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{l}{r} \\ \frac{1}{r} & -\frac{l}{r} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3-4)$$

3.2. Seguimiento de trayectorias con control discontinuo

El seguimiento de trayectorias es indispensable en la robótica, debido a que de ello depende la ejecución de una tarea específica, para esto se implementan algoritmos de control. En este caso la estrategia de control es no lineal discontinua, en la cual se implementan dos linealizaciones entrada-salida diferentes, las cuales son conmutadas para lograr el control en todo instante de tiempo y en diversas condiciones del móvil en el plano, para esto se realiza lo siguiente:

3.2.1. Linealización entrada-salida por realimentación dinámica

Se desarrolla una linealización de la respuesta entrada-salida del sistema, en la cual se consideran las variables x y y como variables de estudio, y que determinan la posición del centro del móvil; las variables de salida u_1 y u_2 , se definen como las variables de control. Estas deben ser corregidas para modificar la trayectoria del vehículo.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos \varphi & 0 \\ \sin \varphi & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = A(\varphi) \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3-5)$$

En la Ec. (3-5), se puede observar que la matriz A es singular para todo valor, para solucionar dicho problema como primera medida se derivan nuevamente las variables x y y , se define una nueva variable de estado ζ y una nueva variable de control \tilde{u}_1 , las cuales están relacionadas en Ec. (3-1), de la siguiente manera: $\dot{\zeta} = \tilde{u}_1$, $u_1 = \zeta$. Se sabe que en la Ec. (3-1), $\dot{\varphi} = u_2$, por ende la Ec. (3-6), queda definida como:

$$\begin{aligned} \ddot{x} &= -\zeta u_2 \sin \varphi + \tilde{u}_1 \cos \varphi \\ \ddot{y} &= \zeta u_2 \cos \varphi + \tilde{u}_1 \sin \varphi \end{aligned} \quad (3-6)$$

Ahora si se organiza la Ec. (3-6) en forma matricial, se obtiene como resultado el modelo visto en la Ec. (3-7), en donde se observa que la matriz A es singular cuando $\zeta = 0$, es decir, cuando la velocidad lineal del vehículo es igual a cero ($u_1 = v = 0$). Esto ocurre cuando el móvil no presenta desplazamiento en x y y (cuando se encuentra en reposo o el movimiento que esta realizando es de rotación sobre su propio eje).

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \cos \varphi & -\zeta \sin \varphi \\ \sin \varphi & \zeta \cos \varphi \end{bmatrix} \begin{bmatrix} \tilde{u}_1 \\ u_2 \end{bmatrix} = A_e(\varphi, \zeta) \begin{bmatrix} \tilde{u}_1 \\ u_2 \end{bmatrix} \quad (3-7)$$

Se define la realimentación dinámica:

$$\begin{bmatrix} \tilde{u}_1 \\ u_2 \end{bmatrix} = A_e^{-1}(\varphi, \zeta) \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

donde las nuevas variables de control están definidas por la Ec. (3-8).

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \ddot{x} - \alpha_1(\dot{x} - \dot{x}_d) - \alpha_0(x - x_d) \\ \ddot{y} - \beta_1(\dot{y} - \dot{y}_d) - \beta_0(y - y_d) \end{bmatrix} \quad (3-8)$$

Si se define las variables de error $e_x(t)$ y $e_y(t)$, mediante:

$$e_x = x - x_d$$

$$e_y = y - y_d$$

Es posible demostrar que su dinámica esta gobernada por las ecuaciones diferenciales mostradas en la Ec. (3-9).

$$\begin{aligned} \ddot{e}_x + \alpha_1 \dot{e}_x + \alpha_0 e_x &= 0 \\ \ddot{e}_y + \beta_1 \dot{e}_y + \beta_0 e_y &= 0 \end{aligned} \quad (3-9)$$

Bajo la elección apropiada de los parámetros α_i y β_i con $i \in 0, 1$, es posible demostrar que $e_x, e_y \rightarrow 0$ cuando $t \rightarrow \infty$ siempre que $\alpha_0, \alpha_1 > 0$. La matriz A vuelve a ser singular en un modelo ideal sin restricciones en la velocidad y posición del móvil cuando $\zeta = 0$. Para evitar esta inestabilidad, se propone una nueva ley de control para una nueva variable de salida.

3.2.2. Linealización por realimentación estática

Con el propósito de evitar el problema de singularidad en el caso de la linealización por realimentación dinámica, en esta sección se propone una ley de control con respecto a un nuevo conjunto de variables de salida. Específicamente x, φ como variables de salida, las cuales están relacionadas con las variables de control como se muestra en la Ec. (3-10).

$$\begin{bmatrix} \dot{x} \\ \dot{\varphi} \end{bmatrix} = \tilde{A}(\varphi) \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \cos \varphi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3-10)$$

Las variables de control u_1, u_2 , están definidas por la Ec. (3-11).

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \tilde{A}^{-1}(\varphi) \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{\cos \varphi} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (3-11)$$

Finalmente, eligiendo las nuevas variables de control v_1 y v_2 como:

$$\begin{bmatrix} \dot{x} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \dot{x}_d - \alpha e_x \\ \dot{\varphi}_d - \beta e_\varphi \end{bmatrix} \quad (3-12)$$

Es fácil mostrar que el error de seguimiento está gobernado por el sistema de ecuaciones diferenciales dado en la Ec. (3-13).

$$\begin{aligned} \dot{e}_x + \alpha e_x &= 0 \\ \dot{e}_\varphi + \beta e_\varphi &= 0 \end{aligned} \quad (3-13)$$

Bajo la elección apropiada de los parámetros α_i y β_i , es posible demostrar que $e_x, e_\varphi \rightarrow 0$ cuando $t \rightarrow \infty$, siempre que $\alpha_0, \alpha_1 > 0$ lo que implica a la vez que $x \rightarrow x_d$ y $\varphi \rightarrow \varphi_d$ cuando $t \rightarrow \infty$. Combinando la Ec. (3-11), y la Ec. (3-12), se concluye que los perfiles de velocidad que los actuadores del robot deben seguir están dados por:

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{\cos \varphi} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_d - \alpha(x - x_d) \\ \dot{\varphi}_d - \beta(\varphi - \varphi_d) \end{bmatrix}$$

Es importante notar que la ley de control anterior no está definida globalmente. En efecto, la variable u_1 crece sin ninguna cota cuando $\varphi \rightarrow \frac{k\pi}{2}$, $k = \pm 1, \pm 2, \dots, n$.

3.2.3. Control discontinuo del modelo cinemático

Como se menciona anteriormente, la ley de control por retroalimentación dinámica no está definida cuando $u_1 = \zeta = 0$. Esta condición ocurre cuando el robot se encuentra inicialmente en reposo, o bien cuando la trayectoria preestablecida obliga al robot a detenerse momentáneamente. Para evitar este problema se propone una estructura variable, la cual admite descripciones matemáticas distintas durante períodos de operación diferentes. Para esto se realiza una conmutación, que se encarga de cambiar la ley de control por realimentación estática (esta se utiliza siempre y cuando la magnitud de la velocidad lineal del robot u_1 sea inferior a un cierto umbral $\epsilon > 0$, es decir, cuando $|\zeta| \leq \epsilon$) por la ley de control por realimentación dinámica y permanece en este estado mientras la velocidad lineal sea diferente de cero.

Es importante notar que la ley de control estática no está definida para $\varphi = \frac{k\pi}{2}$, $k = \pm 1, \pm 2, \dots, n$. Por lo anterior, se asume que las trayectorias del robot no intersectan la subvariedad

$$S = \left\{ (x, y, \varphi, \zeta) \mid \zeta = 0, \varphi = \frac{k\pi}{2} \right\}$$

donde ambas leyes de control no están definidas.

El esquema de control se ilustra en la Figura **3-1**, en ésta se identifica un algoritmo de estructura variable y se encarga del control del modelo cinemático; éste conmuta entre la linealización por realimentación estática y la linealización por realimentación dinámica a fin de evitar la singularidad del último, la cual se presenta cuando el móvil se detiene o rota sobre su propio eje ($v = 0$).

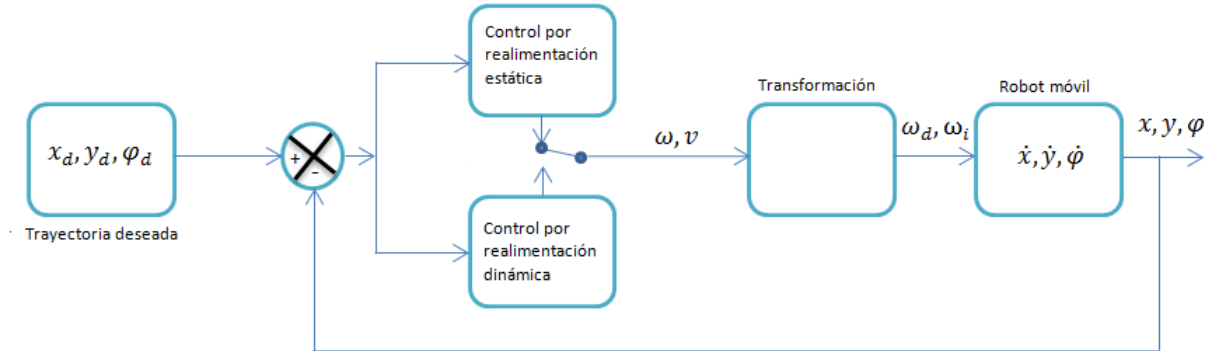


Figura 3-1.: Diagrama de bloques del sistema de control discontinuo.

3.3. Resultados

Usando el algoritmo mostrado en el Anexo B, se realiza la simulación de dos trayectorias en las que se emplea el control conmutado descrito en este capítulo.

Para lograr la trayectoria descrita por el vehículo, es necesario que las variables x_d y y_d (entrada del algoritmo) sean vectores obtenidos por el muestreo de las coordenadas del móvil en la trayectoria deseada. El algoritmo se diseñó para que las velocidades angulares de las ruedas se saturaran a una magnitud de $\pm 20 \text{ rad/seg}$.

3.3.1. Trayectoria circular

Las características del robot móvil son: $r = 0,05 \text{ m}$, $l = 0,15 \text{ m}$, estos son el radio de las ruedas y la mitad de la distancia entre las dos ruedas del vehículo; el trayecto circular está parametrizado por la ecuación (3-14) y los parámetros del controlador para obtener una trayectoria con un error pequeño son: $\alpha_1 = 1,4$, $\alpha_0 = 1,2$, $\beta_1 = 1,4$, $\beta_0 = 1,2$, los cuales fueron seleccionados empíricamente.

$$\begin{aligned} x_d &= 0,3 \cos(t) \\ y_d &= 0,3 \sin(t) \end{aligned} \tag{3-14}$$

Esta es una trayectoria circular de 30 *cm* de radio, la cual debe ser seguida por el vehículo de guiado diferencial. En la Figura 3-2, se observa el comportamiento del móvil en el tiempo, cuando este actúa bajo la acción de control implementada, la trayectoria deseada se representa por medio de la línea roja y la azul indica la posición del móvil en un período de la trayectoria, el móvil inicia en el punto $(x \ y \ \varphi)=(0,305 \ 0,01 \ \frac{\pi}{2})$.

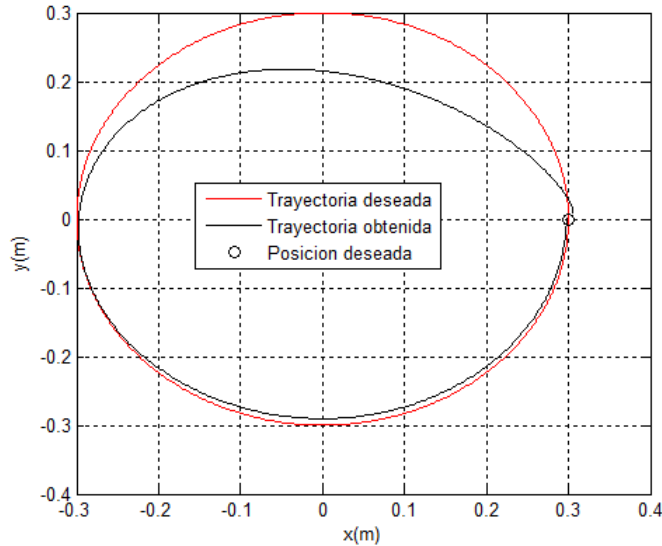


Figura 3-2.: Trayectoria deseada y obtenida.

Como se puede apreciar en la Figura 3-2, el control responde de una manera óptima minimizando el error de posición, haciendo que el móvil siga la trayectoria de referencia. En la Figura 3-3 se observa el error cuadrático medio de posición del móvil, el cual es grande al inicio, debido a que éste empieza en un punto fuera del camino planeado y con una dirección inicial diferente a la referencia. Después de un tiempo el móvil alcanza la trayectoria deseada disminuyendo el error, debido al acercamiento del vehículo al camino trazado. En la Figura 3-4, se aprecia el error cuadrático medio de orientación del móvil, el cual es grande al inicio y va disminuyendo a medida que el vehículo se acerca la trayectoria de referencia.

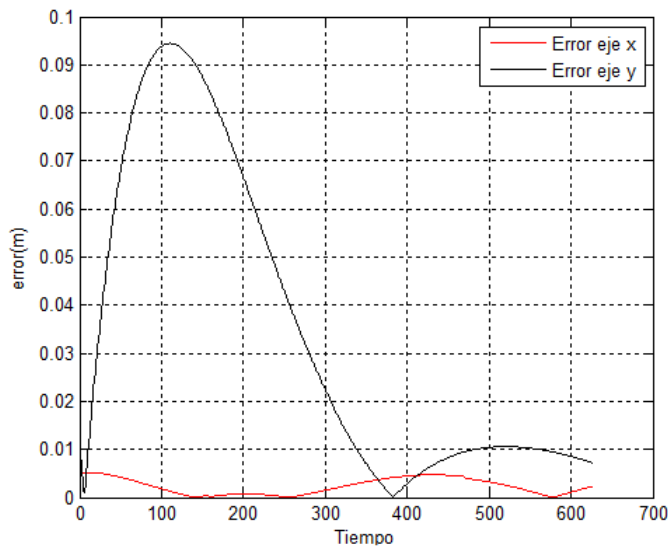


Figura 3-3.: Error cuadrático medio de posición.

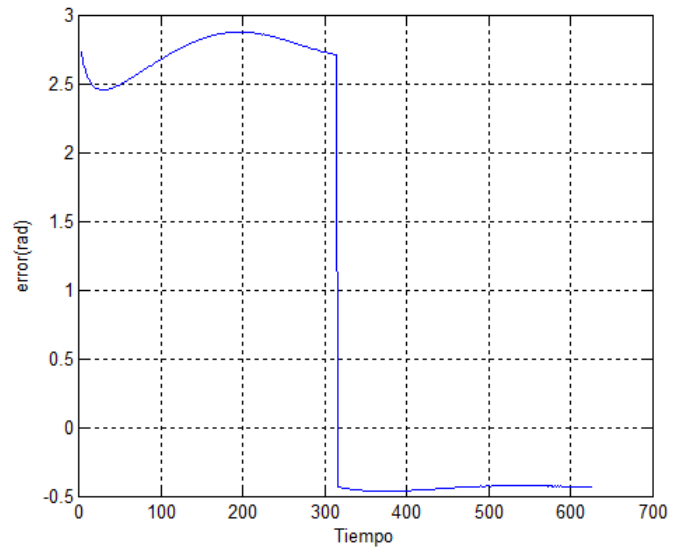


Figura 3-4.: Error cuadrático medio de orientación.

En la Figura 3-5, se observa la velocidad angular de la rueda derecha en el tiempo y la Figura 3-6 muestra la velocidad angular de la rueda izquierda en el tiempo; éstas gráficas exponen el comportamiento de los perfiles de velocidad del móvil. En éstas se ve como cambian las velocidades para que el móvil alcance la trayectoria deseada, luego que la trayectoria descrita alcanza la referencia se ve un comportamiento casi constante de las velocidades angulares.

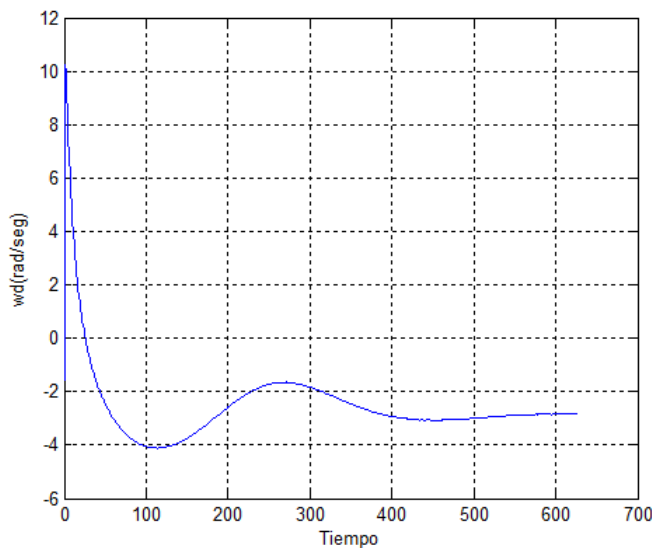


Figura 3-5.: Velocidad angular en la rueda derecha.

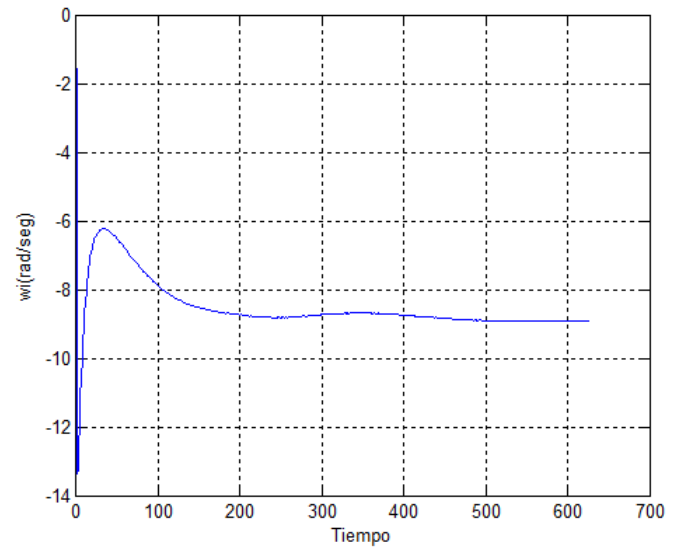


Figura 3-6.: Velocidad angular en la rueda izquierda.

En la Tabla **3-1**, se observan las medidas de desempeño del controlador, las cuales son: error máximo, error mínimo, la media del error y la desviación estándar de éste. Como se puede apreciar, hay mayor error en el eje y , lo cual se puede ver en la Figura **3-2**. Estas medidas de desempeño se realizaron sobre el error cuadrático medio del controlador [39]. Al comparar la media del error con la tabla de índices de desempeño mostrada en [29], se tiene que el controlador responde adecuadamente haciendo que el móvil siga la trayectoria deseada.

Tabla 3-1.: Medidas de desempeño del controlador.

Criterio de medida	Máximo	Mínimo	Promedio	Desviación
Error de posición en x (cm)	0.5179	0.000321	0.23	0.1677
Error de posición en y (cm)	9.45	0.0051	3.46	3.25

3.3.2. Trayectoria en espiral

Las características del móvil son: $r = 0,05 \text{ m}$, $l = 0,15 \text{ m}$, estos son el radio de las ruedas y la mitad de la distancia entre las dos ruedas del vehículo; el camino en espiral esta parametrizado por la ecuación (3-15) y los parámetros del controlador para obtener un error de posición pequeño son: $\alpha_1 = 1,4$, $\alpha_0 = 1,2$, $\beta_1 = 1,4$, $\beta_0 = 1,2$, los cuales fueron seleccionados empíricamente.

$$\begin{aligned} x_d &= 0,1t \cos(t) \\ y_d &= 0,1t \sin(t) \end{aligned} \tag{3-15}$$

En la Figura **3-7**, se observa el comportamiento del móvil en el tiempo, cuando este actúa bajo la acción de control implementada, la trayectoria deseada se representa por medio de la línea roja y la azul indica la posición del móvil en todo instante de tiempo. El móvil inicia en el punto $(x \ y \ \varphi) = (-0,1 \ 0 \ \pi)$ y sigue la trayectoria de referencia con un error pequeño hasta que la velocidad angular de las ruedas se satura, a partir de este punto la ruta seguida por el móvil se aleja de la deseada.

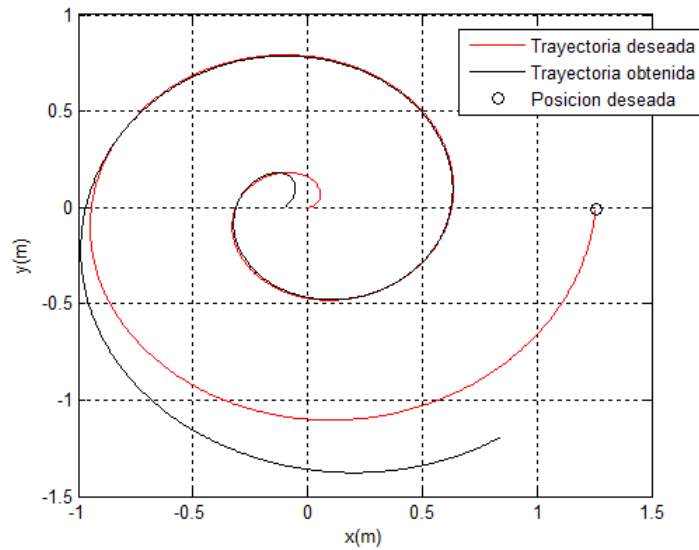


Figura 3-7.: Trayectoria deseada y obtenida.

En la Figura 3-8 se observa el error cuadrático medio de posición del móvil, el cual es grande al inicio, debido a que éste empieza en un punto fuera del camino planeado y con una dirección inicial diferente a la referencia. Después de un tiempo el móvil alcanza la trayectoria deseada disminuyendo el error, debido al acercamiento del vehículo al camino trazado. En la Figura 3-9, se aprecia el error cuadrático medio de orientación del móvil, el cual es grande al inicio y va disminuyendo a medida que el vehículo se acerca la trayectoria de referencia.

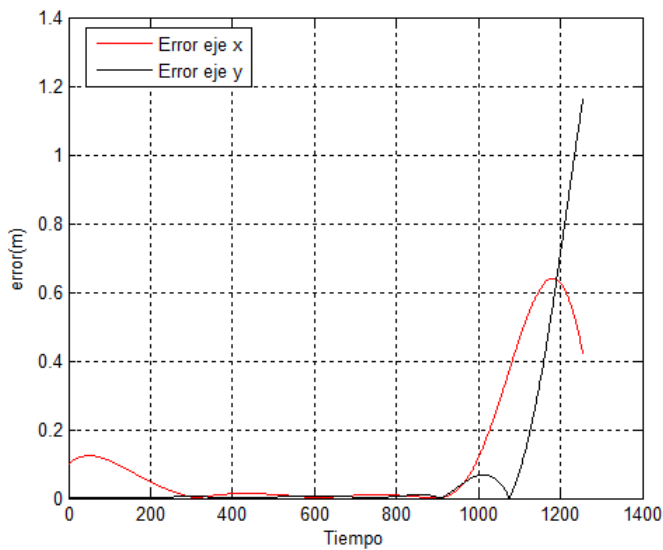


Figura 3-8.: Error cuadrático medio de posición.

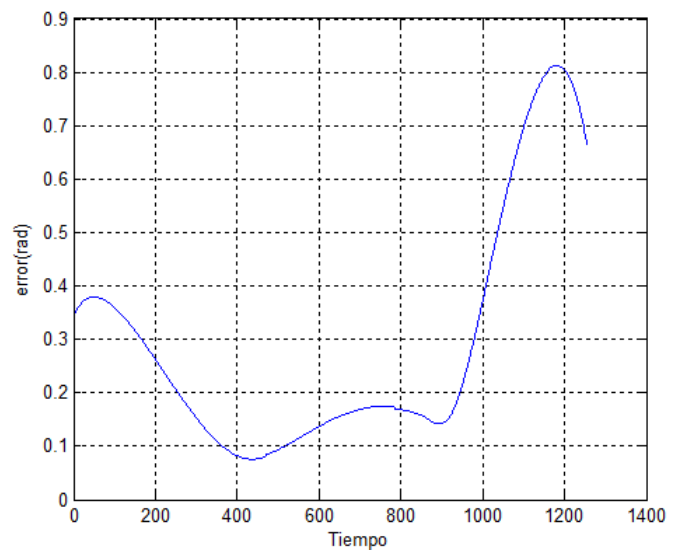


Figura 3-9.: Error cuadrático medio de orientación.

En las Figuras **3-10** y **3-11**, se observan los perfiles de velocidad angular de las ruedas derecha e izquierda respectivamente con respecto al tiempo. En éstas se ve como las velocidades cambian rápidamente para que el móvil alcance la trayectoria deseada, luego que la trayectoria descrita alcanza la referencia se ve un comportamiento casi lineal de la velocidades. Se debe tener en cuenta que el móvil no debe alcanzar ni sobrepasar la velocidad angular máxima de los motores ($\pm 20 \text{ rad/seg}$) para que el controlador sea capaz de responder y hacer que el éste siga el camino deseado. En la Figura **3-11** se observa la saturación en velocidad de la rueda izquierda, haciendo que el sistema no siga al trayecto planeado y que el error empiece a aumentar.

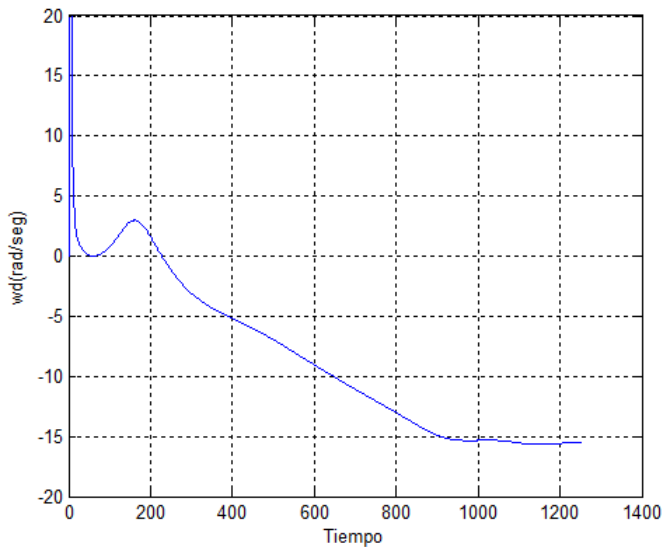


Figura 3-10.: Velocidad angular en la rueda derecha.

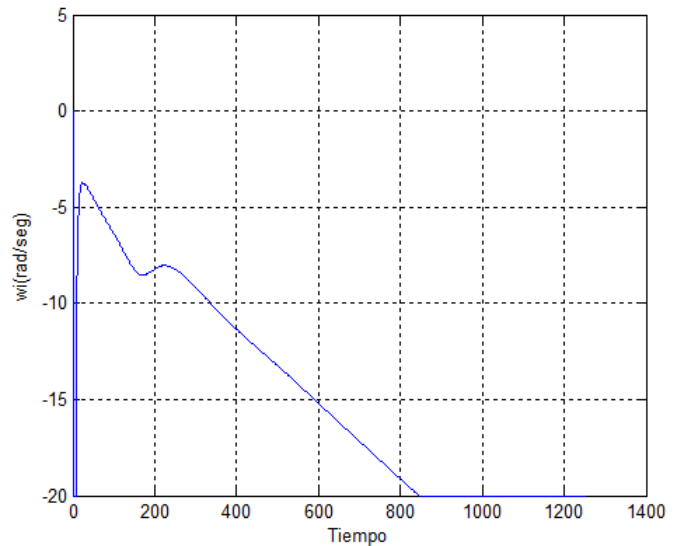


Figura 3-11.: Velocidad angular en la rueda izquierda.

En la Tabla **3-2**, se observan las medidas de desempeño del controlador, las cuales son: error máximo, error mínimo, la media del error y la desviación estándar de éste. Como se puede apreciar, hay mayor error en el eje x , lo cual se puede ver en la Figura **3-7**. Estas medidas de desempeño se realizaron sobre el error cuadrático medio del controlador [39]. Al comparar la media del error con la tabla de índices de desempeño mostrada en [29], se tiene que el controlador responde adecuadamente haciendo que el móvil siga la trayectoria deseada.

Tabla 3-2.: Medidas de desempeño del controlador.

Criterio de medida	Máximo	Mínimo	Promedio	Desviación
Error de posición en x (cm)	12.35	0.0012	2.85	3.84
Error de posición en y (cm)	2.35	0.00099	0.4344	0.31

3.4. Conclusiones

Para el control de robots móviles se implementan diferentes métodos, en los cuales se emplean conceptos de la teoría de control. Estos métodos desarrollan técnicas para sistemas lineales y no lineales, las cuales se basan en redes neuronales, lógica difusa, sistemas de aprendizaje autónomo e inteligencia artificial.

El control en robots móviles se realiza para el seguimiento de caminos explícitos, previamente definidos y para la realización de maniobras reaccionando según la información suministrada por los sensores de percepción del entorno, en los que se pueden destacar sensores de proximidad y sistemas de visión.

En este capítulo se empleó la metodología de la linealización entrada-salida por realimentación estática y por realimentación dinámica del vector de estados, para diseñar el controlador de un vehículo guiado automáticamente, con el fin de llevar a cabo la tarea de seguimiento de trayectoria.

La primera ley de control se basa en la linealización entrada-salida por realimentación estática del vector de estados. Con esta ley se logra la convergencia de las variables (x, φ) del modelo cinemático del vehículo. La dinámica remanente asociada a esta ley de control está dada por la variable y . Las singularidades de esta ley se presentan en $\varphi = \pm \frac{k\pi}{s}$, con $k = 1, 2, 3, \dots$

La segunda ley de control se basa en la linealización entrada-salida por realimentación dinámica del vector de estados. En esta ley se aplica el algoritmo de extensión dinámica, AED para lograr la convergencia de las variables de estado (x,y) y evadir las singularidades inherentes al modelo cinemático del vehículo. En esta ley de control la dinámica interna es trivial.

Se emplea un esquema de control basado en la linealización de entrada-salida, el cual trabaja con el modelo cinemático y la trayectoria de referencia del vehículo, generando los perfiles de velocidad necesarios en cada rueda del robot. El control realiza la corrección de la ruta descrita por el móvil, teniendo un error pequeño siempre y cuando se seleccionen los parámetros adecuados del controlador. Para lograr una mayor aproximación a la realidad se utilizan perfiles de velocidad no mayor a 20 rad/seg ni menor a -20 rad/seg , lo cual puede generar una saturación que afecta la acción de control y aumenta el error de posición.

4. Algoritmos de cobertura completa para tareas de aseo

Los algoritmos de planificación de trayectorias son ampliamente utilizados en la robótica móvil en diferentes aplicaciones, que se hace necesario seguir un trayecto evitando obstáculos. En la robótica móvil se pueden presentar dos casos al momento de la planificación, los cuales son: en el primero se tiene información apriori del entorno o área de trabajo, con esto se puede proyectar una ruta a seguir, en donde no se debe colisionar con los obstáculos; en el segundo no se tiene información del medio (entorno desconocido), por consiguiente se debe planear una trayectoria adicional para el reconocimiento del área o se empieza a recorrer el área y se va reconstruyendo el ambiente a medida que se recorre.

En algunas aplicaciones es de vital importancia cubrir todo el espacio libre, lo cual se realiza por medio del algoritmo de Planificación de Trayectorias de Cobertura Completa (CCPP); éste es utilizado en aplicaciones de robots de aseo, estudiadas en [2, 3, 5, 6, 11, 13, 16, 19, 25, 27, 40, 41, 42, 43, 44], vehículos autónomos submarinos [45], robots para búsqueda de minas, cortadores de césped autónomos [46], equipos de cosecha de cultivos [47, 48], limpiadores fachadas y ventanas [7, 49], robots de limpieza de tanques de almacenamiento de líquidos [8], limpiador de piscinas autónomo [50], robots de limpieza de residuos [51].

Algunas de las técnicas utilizadas para solucionar el problema de cobertura completa (CCPPP), se mencionan en [52] y estas se podrían resumir como sigue:

- Descomponer formas complejas en polígonos convexos, luego se definen los obstáculos como huecos en el área de trabajo, posteriormente se encuentra el circuito Hamiltoniano en el gráfico generado mediante la conexión de los centros de los polígonos, este método se estudia en [14]. En este se utilizan 2 estrategias, la primera tiene como objetivo minimizar el número de vueltas necesarias para completar el trayecto, mientras que en la segunda el objetivo es minimizar la máxima longitud del camino.
- Emplear un algoritmo aplicando “*A Bioinspired Neural Network*”, el cual se estudia a fondo en [4, 5, 9, 10, 12, 13, 43, 53, 54]
- Utilizar sistemas de posicionamiento, tales como encoder, cámaras, compás electrónico, GPS; esto con el fin de recorrer el entorno inexplorado, mientras se realizan tareas de aseo, seguridad, entre otras. En [2, 3, 11, 25] se estudia este caso.

- Dividir el espacio en Boustrophedon cell, el cual es una generalización del método de descomposición trapezoidal, luego realizar el algoritmo *back-and-forth Motion* e implementar el algoritmo A^* , para búsqueda del punto más cercano, con el fin de dar marcha a atrás e ir a otra región. Éste método se explica en [17, 23, 55].
- Realizar una descomposición del espacio en celdas, para implementar un método heurístico asignando una función de costo a cada celda, el costo de la celda es bajo si se ha realizado el movimiento cubriéndola en más de una ocasión, con esto se garantiza que el robot cubra las celdas no exploradas [15, 21, 24, 56, 26].
- Descomponer el espacio en celdas, para luego realizar el algoritmo de cobertura, el cual puede ser genético, aleatorio, entre otros [16, 19, 20, 22, 57, 58].
- Implementar varios robots para cubrir todo el espacio, en [18, 27, 41, 59, 60, 61]

Para realizar la tarea de cobertura, se hace necesario que el algoritmo sea eficiente, encontrando el camino más corto o el trayecto con menor consumo de energía, en [44, 62], se muestra un estudio para encontrar la trayectoria más eficiente.

En [42], se muestra el comportamiento de los diferentes robots domésticos de aseo en el mercado, los cuales son: Roomba (iRobot), CleanMate (Infinuvo), IClebo (Yujin Robot), Koolvac (Koolatron), Trilobite (Electrolux), Hauzen VC-RE72V (Samsung).

El *back-and-forth Motion* consiste en seguir una secuencia de pasos para realizar una trayectoria en zigzag, en el cual se realizan movimiento verticales y horizontales para recorrer todo el espacio libre.

El proyecto de investigación implementa una modificación de este algoritmo, debido a que se presentan problemas de discontinuidad en el trayecto cuando se cambia de un movimiento vertical a uno horizontal o viceversa y para este caso se desea que la ruta sea lo mas suavizada posible y así realizar el control para el seguimiento de trayectorias en todo instante de tiempo.

4.1. Sistemas de navegación autónoma

En la robótica móvil, la navegación se define como la metodología o arte 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 [1].
- 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.

Una de las aplicaciones más prometedoras de la robótica móvil es la limpieza del suelo, debido a que se trata de una tarea repetitiva que requiere sólo conocer el entorno o mapearlo. Actualmente existen varios robots que realizan esta función, aunque la mayoría se basan en un algoritmo de exploración aleatorio.

El problema de los algoritmos de aseo consiste en recorrer todo el espacio libre sin colisionar con los obstáculos, algunas estrategias son: *Back-and-Forth Method*, el cual se trata en [11, 19, 55] y consiste en realizar el recorrido mostrado en la Figura 4-2, en [16] se menciona el *Back-and-Forth Method* y se compara con el Boundary Sweep Method, los cuales son utilizados para solucionar el CCPPP (Complete Coverage Path Planning Problem).

4.2. Planeación de trayectorias de cobertura completa

Para realizar esta tarea, se parte de la premisa de conocer el área de trabajo, por lo que ya se tiene las dimensiones y distribución de la escena. Conociendo esto, es necesario realizar un algoritmo, el cual consta de tres partes: la primera ubicación, en donde se realiza la localización e inicialización; la segunda parte es la encargada de posicionar al móvil en la esquina más cercana para iniciar la tarea de aseo en el entorno cerrado y la tercera es la encargada de realizar el proceso de limpieza, el cual es un movimiento repetitivo con el algoritmo de *Back-and-Forth Motion* modificado. En la Figura 4-1, se muestra el diagrama de estados del algoritmo, el cual explica secuencialmente cual es la estrategia a realizar y en la Figura 4-2, se observa la forma de la trayectoria que debe llevar a cabo el robot para realizar la tarea de aseo, ésta corresponde a la acción ejecutada en los estados 6 y 7.

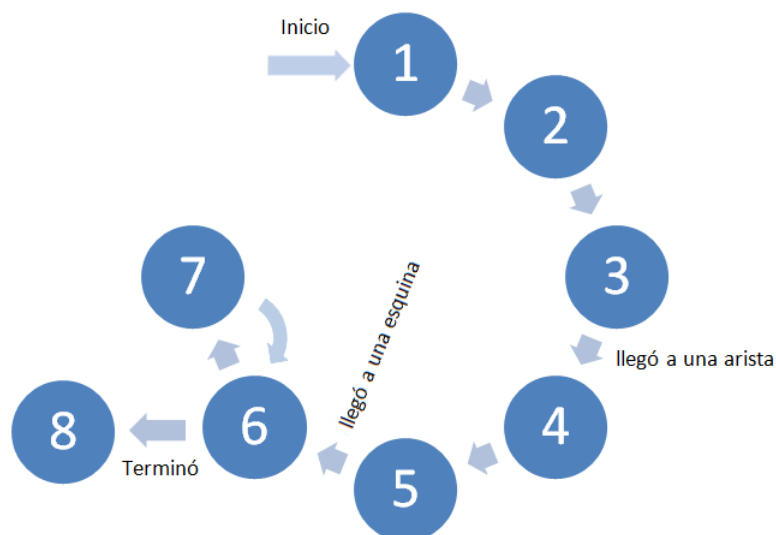


Figura 4-1.: Diagrama de estados del algoritmo.

En la Tabla 4-1, se muestran las acciones de cada estado, estas son las tareas que se deben realizar para obtener la limpieza del espacio mencionado.

Tabla 4-1.: Acciones a realizar en los estados de la Figura 1.

Estado	Tarea
1	Ubicación e inicialización
2	Encontrar la arista más cercana paralela a la ordenada
3	Desplazarse hasta la arista más cercana paralela a la ordenada
4	Encontrar la arista más cercana paralela a la abscisa
5	Desplazarse hasta la arista más cercana paralela a la abscisa
6	Realizar Movimiento x (horizontal) de la tarea de aseo
7	Realizar Movimiento y (vertical) de la tarea de aseo
8	Fin de la tarea de aseo

Los estados 6 y 7 se deben repetir hasta que el robot llega hasta el final del recorrido, este final es dado por las dimensiones del área de trabajo.

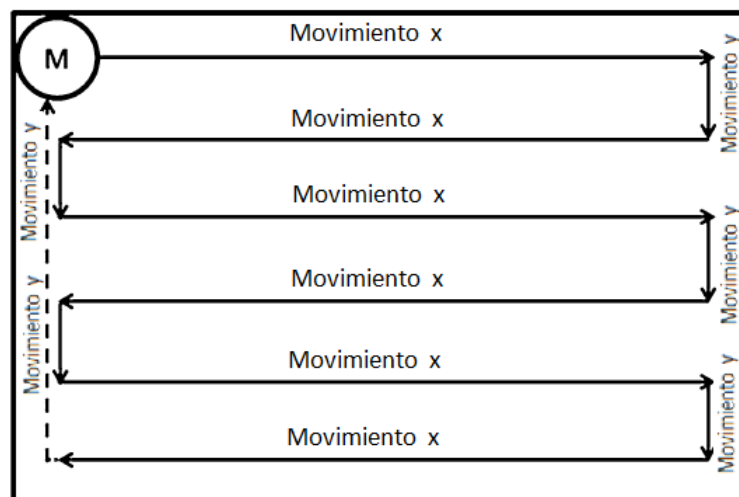


Figura 4-2.: Recorrido de limpieza.

4.3. Resultados

Para realizar la tarea de aseo, se procede a implementar el diagrama de estados mostrado en la Figura 4-1, por tanto se desarrolla un algoritmo en Matlab[®] mostrado en el Anexo C, donde se implementa un planificador de trayectorias para realizar el *Back-and-Forth Motion* modificado, el cual se diseña de tal forma que cubra todo el espacio teniendo en cuenta la geometría y tamaño del robot. El algoritmo planifica un trayecto y con ayuda de la cinemática del móvil se compara y se corrige la posición implementando un control discontinuo del modelo cinemático; el área a limpiar es rectangular de 2 metros de ancho y 2,3 metros de alto, sin obstáculos; teniendo en cuenta esto se realizan las simulaciones. Los valores de los parámetros del controlador para obtener un error de posición bajo son: $\alpha_1 = 1,4$, $\alpha_0 = 1,2$, $\beta_1 = 1,4$, $\beta_0 = 1,2$, estos fueron seleccionados empíricamente.

4.3.1. Escena 1

La simulación se realiza para los parámetros del móvil de guiado diferencial $r = 0,05 \text{ m}$, $l = 0,15 \text{ m}$, los cuales son el radio de las ruedas y la separación entre el centro del vehículo y cada rueda; Tomando las siguientes condiciones iniciales $(x_i \ y_i \ \varphi) = (-0,5 \ 0,8 \ 0)$, se observa en la Figura 4-3 el recorrido inicial del robot, en esta se muestra como el móvil se ubica en la esquina más cercana para poder iniciar la tarea de aseo, dicho movimiento fue realizado por medio del algoritmo planteado en el Anexo C.

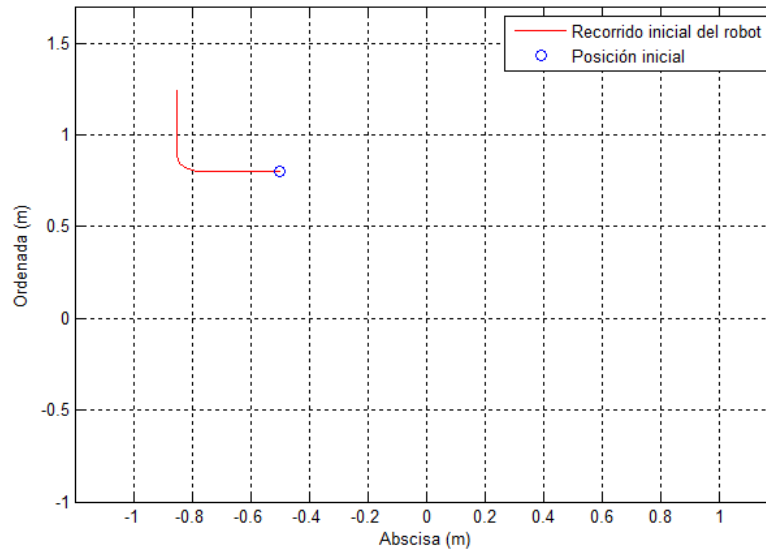


Figura 4-3.: Recorrido inicial.

Después de realizar la simulación, en la Figura 4-4 se muestra la trayectoria planificada del algoritmo de aseo, en la cual se observa el como el móvil se desplaza siguiendo el *Back-and-Forth Motion* modificando el movimiento vertical, haciendo una trayectoria suave, esto es debido a que el controlador utiliza la derivada de la trayectoria y si en esta se presenta discontinuidades como se ve en la Figura 4-2, el control no responde de forma adecuada, dejando un error de posición grande. Por tal motivo se suaviza la trayectoria por medio de curvas de Bezier, generando una semicircunferencia.

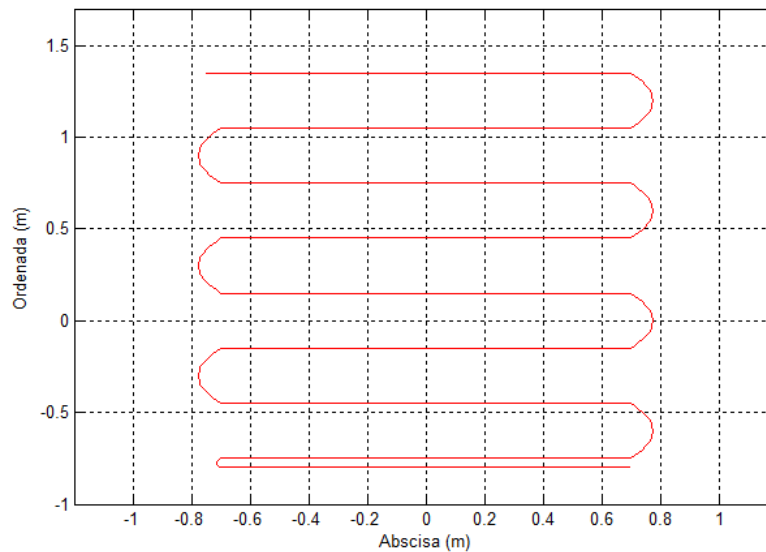


Figura 4-4.: Trayecto planeado, para realizar la tarea de aseo.

Teniendo la ruta planeada y simulando la cinemática del robot se compara punto a punto, tomando muestras con un intervalo de tiempo pequeño. El sistema será realimentado para poder comparar el recorrido del móvil con respecto a la trayectoria de referencia. Se puede observar el comportamiento real dado por la cinemática del móvil, si se grafican los puntos muestreados. La ley de control utilizada es discontinua, no lineal y por realimentación de entrada-salida; en la Figura 4-5, se observa la trayectoria de referencia y la seguida por el móvil.

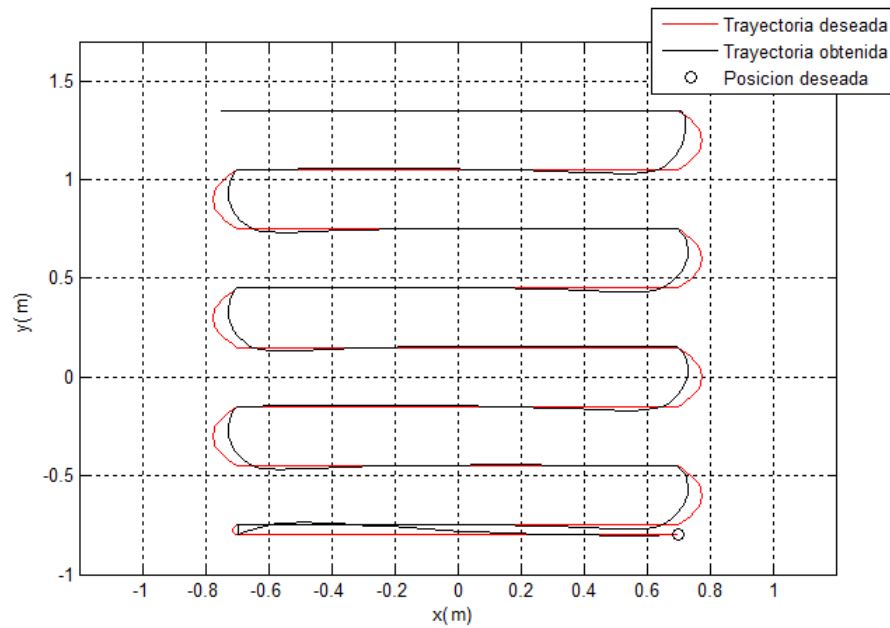


Figura 4-5.: Trayectoria deseada y obtenida.

Como se puede apreciar en la Figura 4-5, la ley de control responde de una manera óptima minimizando el error de posición, lo cual se logra haciendo que el movimiento seguido por el móvil sea lo más cercano posible al de la referencia. En esta se presenta error cuando el robot realiza el movimiento vertical, este es generado por el cambio de la trayectoria y es posible mejorarlo con la optimización de los parámetros del controlador, pero en este caso se toman los valores establecidos al inicio, para los cuales el error de posición es resumido en la Tabla 4-2, en donde se observa que el error es bajo y por tanto el espacio sin asear es muy pequeño.

Tabla 4-2.: Comportamiento de la trayectoria.

Criterio de medida	Máximo	Mínimo	Promedio	Desviación
Error de posición en x (cm)	13.6	0.0978	5.37	3.81
Error de posición en y (cm)	6.38	0	1.39	1.35

En la Figura 4-6, se observa el comportamiento del error cuadrático medio de la posición del móvil en el tiempo, se nota como el error en x es mayor que el de y , debido al error generado en el movimiento curvo en el cual las señales se alejan en la dimensión horizontal.

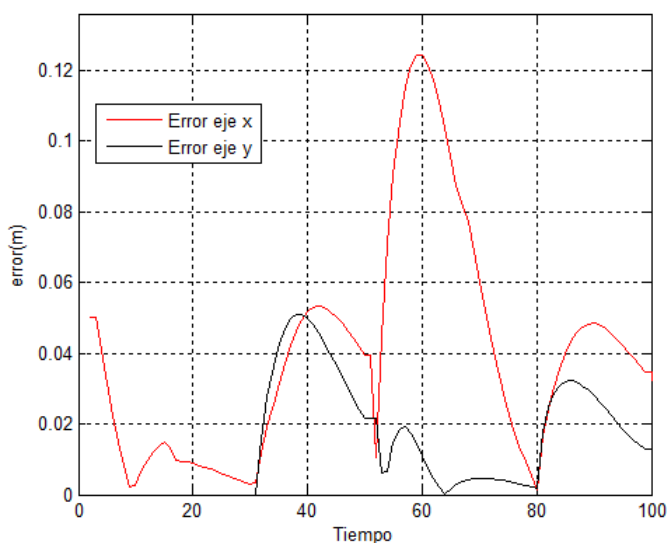


Figura 4-6.: Error cuadrático medio de posición.

En la Figura 4-7, se observa la velocidad angular de la rueda derecha en el tiempo y la Figura 4-8, muestra la velocidad angular de la rueda izquierda en el tiempo; éstas gráficas exponen el comportamiento de las velocidades angulares de las ruedas del móvil y se observa como cambian para que el móvil alcance la trayectoria deseada. Debido a que el robot es de guiado diferencial la dirección del movimiento esta condicionada por las velocidades angulares de las ruedas.

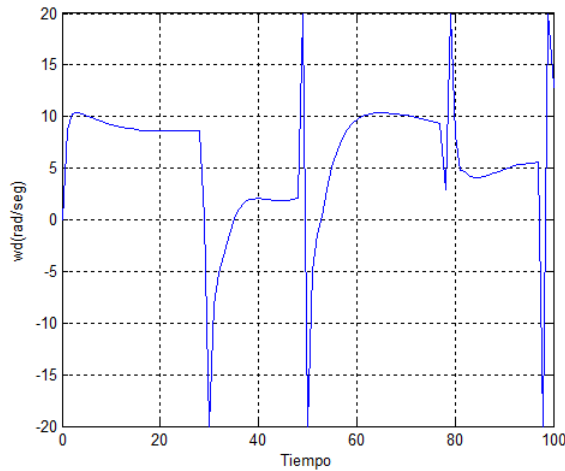


Figura 4-7.: Velocidad angular en la rueda derecha.

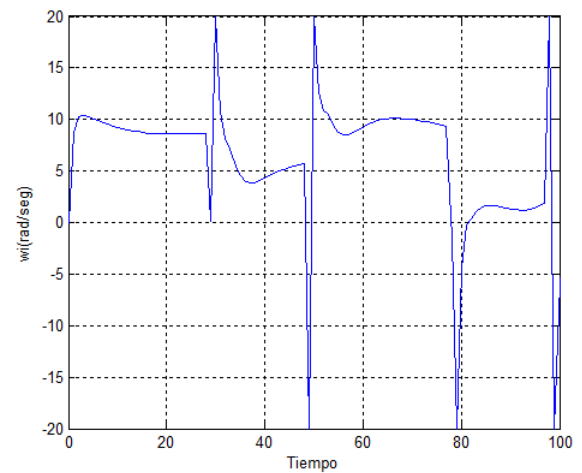


Figura 4-8.: Velocidad angular en la rueda izquierda.

4.3.2. Escena 2

La simulación se ejecuta para los parámetros del móvil de guiado diferencial $r = 0,05 \text{ m}$, $l = 0,15 \text{ m}$, los cuales son el radio de las ruedas y la separación entre el centro del vehículo y cada rueda. Tomando las condiciones iniciales siguientes del móvil $(x_i \ y_i \ \varphi) = (0,6 \ 0,8 \ 0)$, se observa en la Figura 4-9 el recorrido inicial del robot, en este se ve como el móvil se ubica en la esquina más cercana, con el fin de iniciar la tarea de aseo. Este movimiento es generado por el algoritmo planteado en el Anexo C.

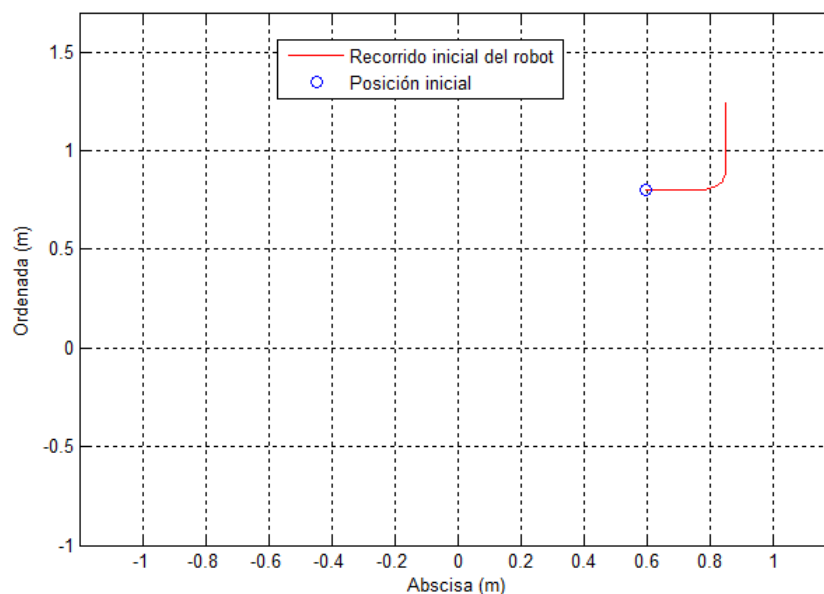


Figura 4-9.: Recorrido inicial.

Después de realizar la simulación, en la Figura 4-10 se muestra la trayectoria planificada del algoritmo de aseo, en la cual se observa el como el móvil se desplaza siguiendo el *Back-and-Forth Motion* modificando el movimiento vertical, haciendo una trayectoria suave, esto se hace porque el controlador utiliza la derivada de la trayectoria y si en esta se presenta discontinuidades como se ve en la Figura 4-2, el control no responde de forma adecuada, dejando un error de posición grande. Por tal motivo se suaviza la trayectoria por medio de curvas de Bezier, generando una semicircunferencia.

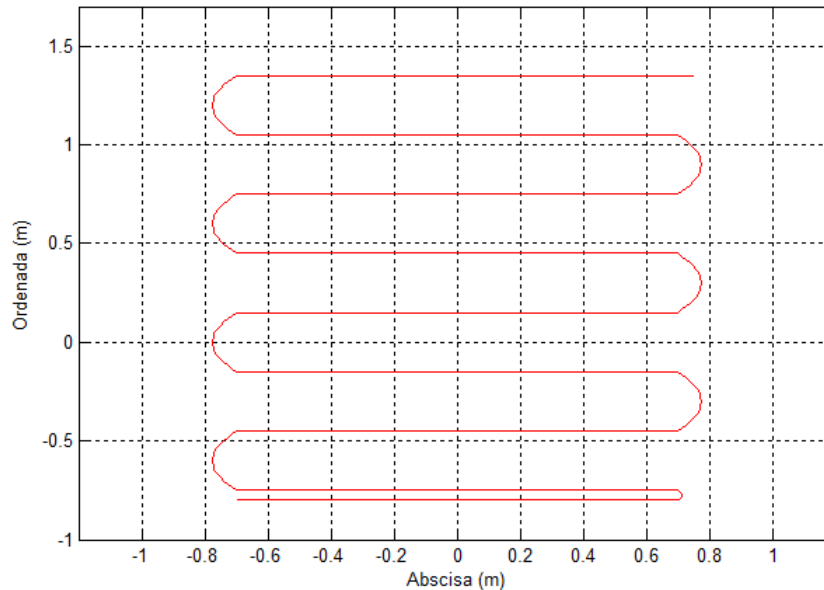


Figura 4-10.: Trayecto planeado, para realizar la tarea de aseo.

Teniendo la ruta planeada y simulando la cinemática del robot se compara punto a punto, tomando muestras con un intervalo de tiempo pequeño. El sistema será realimentado para poder comparar el recorrido del móvil con respecto a la trayectoria de referencia. Se puede observar el comportamiento real dado por la cinemática del móvil, si se grafican los puntos muestreados. La ley de control utilizada es discontinua, no lineal y por realimentación de entrada-salida; en la Figura 4-11, se observa la trayectoria de referencia y la seguida por el móvil.

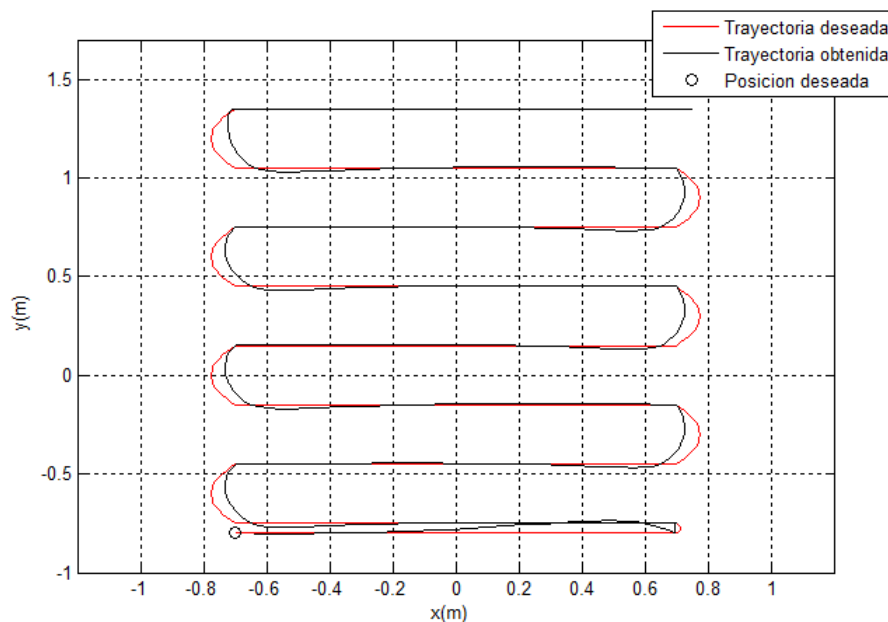


Figura 4-11.: Trayectoria deseada y obtenida.

Como se puede apreciar en la Figura 4-11, la ley de control responde de una manera óptima, haciendo que el movimiento seguido por el móvil sea lo más cercano posible al de la referencia. En esta se presenta error cuando el robot realiza el movimiento vertical, este es generado por el cambio de la trayectoria y es posible mejorarlo con la optimización de los parámetros del controlador, pero en este caso se toman los valores establecidos al inicio, para los cuales el error de posición es resumido en la Tabla 4-3, en donde se observa que el error es bajo y por tanto el espacio sin asear es muy pequeño.

Tabla 4-3.: Comportamiento de la trayectoria.

Criterio de medida	Máximo	Mínimo	Promedio	Desviación
Error de posición en x (cm)	13.6	0.0978	5.37	3.81
Error de posición en y (cm)	6.38	0	1.39	1.35

En la Figura 4-12, se observa el comportamiento del error cuadrático medio de posición en el tiempo, se nota como el error en x es mayor que el de y , debido a que en el movimiento curvo las señales se alejan en la dimensión horizontal.

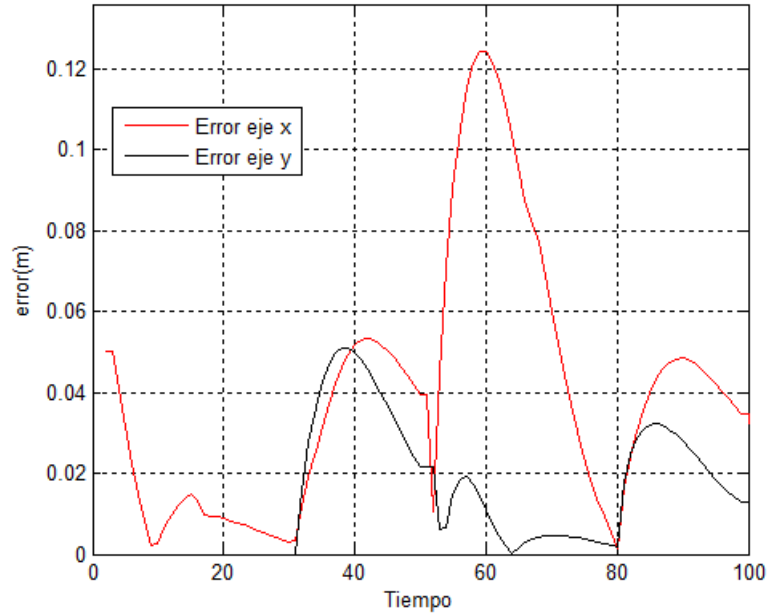


Figura 4-12.: Error cuadrático medio de posición.

En la Figura 4-13, se observa la velocidad angular de la rueda derecha en el tiempo y la Figura 4-14, muestra la velocidad angular de la rueda izquierda en el tiempo; éstas gráficas exponen el comportamiento de las velocidades angulares de las ruedas del móvil y se observa como cambian para que el móvil alcance la trayectoria deseada, debido a que el robot es de guiado diferencial la dirección del movimiento esta condicionada por las velocidades angulares de las ruedas.

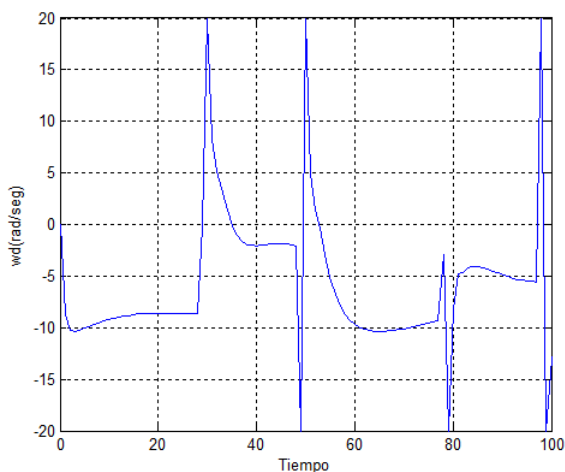


Figura 4-13.: Velocidad angular en la rueda derecha.

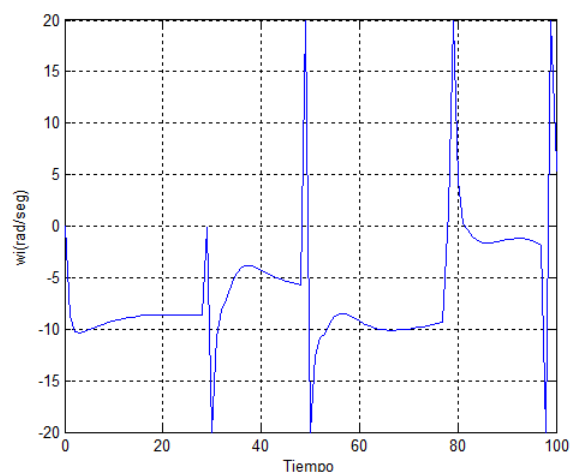


Figura 4-14.: Velocidad angular en la rueda izquierda.

4.4. Conclusiones

De acuerdo a los estudios, el problema de planeación de trayectorias de cobertura completa (CCPPP) es de gran importancia dentro de la robótica móvil, debido a que hay un gran número de aplicaciones que requieren implementar este tipo de algoritmos. En estas aplicaciones las tareas de aseo juegan un papel importante, puesto que es una tarea repetitiva, en la que se requiere cubrir todo el espacio libre para hacer limpieza del entorno.

Algunos de los métodos utilizados son aplicados a entornos conocidos (como se desarrolla en este trabajo), pero también pueden ser aplicados en entornos desconocidos, en donde se debe utilizar un método combinado para reconstruir el entorno y cubrir todo el área de trabajo, este puede ser por: la aplicación de redes neuronales artificiales, descomposición en celdas triangulares, descomposición Boustrophedon o descomposición trapezoidal, subdivisión en regiones, algoritmo de recorrido aleatorio con función de costo.

En este trabajo se desarrollan algoritmos para realizar la tarea de aseo con un robot móvil de guiado diferencial en un entorno doméstico cerrado sin obstáculos, por medio del *Back and Forth Motion* suavizado, para que el controlador responda de manera óptima minimizando el error de posición se deben seleccionar los parámetros correctos del controlador, los cuales en este caso fueron encontrados empíricamente por lo que se deberá implementar un método adaptativo para hallar los parámetros adecuados automáticamente en cada ruta.

5. Conclusiones y recomendaciones

5.1. Conclusiones

En este trabajo se simula un sistema móvil de guiado diferencial, el cual tiene dos ruedas de tracción y una de apoyo para la generación de trayectorias, en el cual el modelo cinemático es no lineal y tiene como variables de control las velocidades angulares de las ruedas derecha e izquierda, es decir, al controlar estas velocidades, se puede generar casi cualquier trayectoria sobre el plano.

Se presenta una estrategia de control para el sistema, la cual es no lineal por realimentación de entrada-salida y se debe realizar la conmutación entre dos leyes para evitar las singularidades de cada una de estas y así controlar el sistema en todo instante de tiempo. La estrategia de control propuesta es similar a la mostrada en [36, 29, 38], con la excepción de no tener lazo de control interno para controlar la tensión de los motores.

La planificación de trayectorias se realiza por medio del algoritmo *Back-and-Forth Motion* con la modificación pertinente para generar un trayecto sin discontinuidades y así mejorar el desempeño del controlador. Antes esto no había sido implementado, por lo que se requerían técnicas de control complejas para poder seguir la ruta planeada a la perfección.

Con este trabajo se entrega un conjunto de rutinas para trabajar de manera modular el problema de cobertura completa de espacios libres, con el fin de implementar una tarea de aseo con robots móviles. Estas tienen la ventaja de ser modulares, permitiendo al usuario cambiar el tipo de configuración del robot móvil y la estrategia de control para facilitar la implementación de diferentes tipos de configuraciones en la cobertura total del espacio libre, lo cual antes era un poco complejo, debido a que no se tenía la practicidad de distribuir el problema en etapas.

El controlador permite determinar los valores de velocidad angular de las ruedas del móvil para seguir una trayectoria generada por el planificador. Se utiliza el controlador propuesto en [29, 38, 36], el cual esta compuesto por una estrategia de control del modelo cinemático del móvil, este desarrolla una linealización de la respuesta entrada-salida del sistema. La respuesta del controlador no presenta discontinuidades en la trayectoria, debido a que se implemento el método de curvas de Bezier para tener un camino suavizado y conseguir un rendimiento adecuado del controlador. En los resultados de simulación se mostró un buen comportamiento en el seguimiento de trayectorias, en el cual se alcanza una posición próxima a la deseada, con un error bajo.

Debido a la complejidad existente en el problema de planificación de trayectorias de cobertura completa, este trabajo busca facilitar su comprensión, por tanto se fracciona el problema en partes y con esto se trabajan los algoritmos, dando como resultado un problema modular el cual es de fácil comprensión para el usuario. Luego de entender cada división se procede a la implementación en conjunto de los algoritmos con el fin de lograr la tarea de cobertura completa para el aseo de una región.

5.2. Recomendaciones

Es necesario implementar un método para la elección de los parámetros del controlador, los cuales se seleccionaron de manera empírica y por ende no se tiene certeza de haber seleccionado los que generan una mejor respuesta. Por ejemplo, se podría implementar un método adaptativo para la selección adecuada de los parámetros α y β , el cual calcule los valores con los cuales el controlador responda de manera que minimice el error de posición.

Se recomienda implementar el vehículo de guiado diferencial para tareas de aseo, para esto es necesario utilizar sistemas de localización, los cuales pueden ser: odométricos, con visión artificial, inerciales, GPS, entre otros. Debido a que ya se trabajó en sistemas de navegación inercial, logrando una ponencia en el IEEE LARC - LARS - CCAC & IASCW 2011 con el artículo llamado “Uso de sistemas inerciales en el control de robots móviles con guiado diferencial”, en éste se obtuvo las señales de los sensores inerciales, acelerómetros y giróscopo; estas señales se procesaron para determinar el desplazamiento del móvil en el plano.

La localización es parte fundamental para el control y seguimiento de trayectorias, debido a que se necesita conocer la posición del vehículo en todo instante para comparar ésta con la trayectoria deseada y proceder a corregirla por medio del controlador, el cual entrega los perfiles de velocidad necesarios en las ruedas para que el móvil siga el camino planificado. Por tanto se propone implementar el método estudiado en [1] que es similar al trabajado en el artículo “Uso de sistemas inerciales en el control de robots móviles con guiado diferencial”.

Como el trabajo realizado partió de una escena ya estipulada, es deseable realizar la reconstrucción de esta por medio de técnicas de visión y/o con ayuda de sensores de proximidad. Estas se deben implementar cuando se tiene un entorno desconocido en el cual se ubica un robot móvil, el cual debe recorrer el espacio libre en su totalidad. Con ayuda de estas técnicas se pueden obtener escenas más completas listas para la planificación de trayectorias de cobertura completa.

A. Anexo: Modelo cinemático del robot

A.1. Método de Euler

En el siguiente código se desarrolla el método de Euler, con el fin de simular el comportamiento del robot móvil. Este algoritmo emplea el modelo del vehículo de guiado diferencial mostrado en la función del sistema de ecuaciones.

```
%% CINEMÁTICA DEL SISTEMA CON GUIADO DIFERENCIAL
clc
clear all
close all

% % Tiempo de simulación
% inicial = input('Ingrese el tiempo inicial de simulación: ');
% final = input('Ingrese el tiempo final de simulación: ');
%
% % Condiciones iniciales del móvil
% conx = input('Ingrese la condición inicial de x: ');
% cony = input('Ingrese la condición inicial de y: ');
% conphi = input('Ingrese la condición inicial de phi: ');
%
% % Número de iteraciones del método
% n = input('Ingrese la cantidad de iteraciones: ');

inicial = 0;
final = 30;
conx = 0;
cony = 0;
conphi = pi/4;
n = 100;

h=(final-inicial)/n;           % Paso de integración del método
t(1)=inicial;                 % Tiempo inicial de graficación
```

```
y(1,:)=[conx; cony; conphi];    % Vector de condiciones iniciales

% Método de Euler
for i=1:n-1
    t(i+1)=t(i)+h;
end

for k=1:n-1
    y(k+1,:) = y(k,:) + h*feval(@robotica_1,t(k),y(k,:));
end

% Diagrama de fase de todas las variables de estado
figure(1)
plot3(y(:,1),y(:,2),y(:,3),'b')
xlabel('x(m)', 'FontSize',10, 'FontWeight', 'bold')
ylabel('y(m)', 'FontSize',10, 'FontWeight', 'bold')
zlabel('phi(rad)', 'FontSize',10, 'FontWeight', 'bold')
title('Diagrama de fase en 3 dimensiones', 'FontSize',13, 'FontWeight', 'bold')
grid on

% Diagrama de fase de x y y
figure(2)
plot(y(:,1),y(:,2),'b')
xlabel('x(m)', 'FontSize',10, 'FontWeight', 'bold')
ylabel('y(m)', 'FontSize',10, 'FontWeight', 'bold')
title('Diagrama de fase de x y y', 'FontSize',13, 'FontWeight', 'bold')
grid on

% Diagrama de fase de x y phi
figure(3)
plot(y(:,1),y(:,3),'b')
xlabel('x(m)', 'FontSize',10, 'FontWeight', 'bold')
ylabel('phi(rad)', 'FontSize',10, 'FontWeight', 'bold')
title('Diagrama de fase de x y phi', 'FontSize',13, 'FontWeight', 'bold')
grid on

% Diagrama de fase de y y phi
figure(4)
plot(y(:,2),y(:,3),'b')
```

```

xlabel('y(m)', 'FontSize',10, 'FontWeight', 'bold')
ylabel('phi(rad)', 'FontSize',10, 'FontWeight', 'bold')
title('Diagrama de fase de y y phi', 'FontSize',13, 'FontWeight', 'bold')
grid on

% Comportamiento de las variables de estado (x,y,phi) en el tiempo
figure(5)
plot(t,y(:,1),'r')
hold on;
plot(t,y(:,2),'g')
hold on;
plot(t,y(:,3),'b')
xlabel('t(seg)', 'FontSize',10, 'FontWeight', 'bold')
ylabel('Variables de estado', 'FontSize',10, 'FontWeight', 'bold')
legend('x','y','phi', 'FontSize',10, 'FontWeight', 'bold')
title('Variables de estado en el tiempo', 'FontSize',13, 'FontWeight', 'bold')
grid on

```

A.2. Modelo matemático del robot de guiado diferencial

El modelo del sistema de guiado diferencial, se observa a continuación:

```

%% SISTEMAS DE ECUACIONES DEL GUIADO DIFERENCIAL
%y(1) es equivalente a la variable de estado dx
%y(2) es equivalente a la variable de estado dy
%y(3) es equivalente a la variable de estado dphi
%x es la distancia recorrida en la abscisa del plano
%y es la distancia recorrida en la ordenada del plano
%phi es la orientación del robot en el plano, ángulo formado entre un eje
%paralelo al Y y el eje del móvil

```

```

% -----
function dy = robotica_1(t,y)
r=0.05;
L=0.15;
wd=1;
wi=-1;
global r L wd wi

```

```

R = [ (r/2)*cos(y(3))*(wd+wi)

```

```
(r/2)*sin(y(3))*(wd+wi)
(r/(2*L))*(wd-wi)];
```

```
dy = R ;
```

```
% -----
```

B. Anexo: Control discontinuo del modelo cinemático

B.1. Programa principal

El programa principal, es el que contiene el algoritmo de control y llama las funciones de plantaRB, derivada, seguimiento1.

```
%% Control no lineal discontinuo de un robot móvil
% Controlador externo de realimentación de entrada-salida

% Las variables de entrada al sistema son: La trayectoria deseada
% (xd,yd), los parámetros del controlador (alpha1, alpha0, beta1, beta0) y
% el tiempo de muestreo (t)

% alpha1=input('Ingrese el valor de alpha1: ');
% alpha0=input('Ingrese el valor de alpha1: ');
% beta1=input('Ingrese el valor de alpha1: ');
% beta0=input('Ingrese el valor de alpha1: ');
% ti=input('Ingrese el tiempo inicial del muestreo: ');
% tf=input('Ingrese el tiempo final del muestreo: ');
% h=input('Ingrese el paso para el muestreo: ');
% t=ti:h:tf;

clc, close all, clear all;

h=0.01;

% Muestreo para trayectoria espiral
% t=0:h:3*pi;

% Muestreo para trayectoria circular
t=0:h:2*pi;
```

```

% Parámetros del controlador

alpha1=1.4;
alpha0=1.2;
beta1=1.4;
beta0=1.2;

% Trayectoria Circular
xd=0.3*cos(t);
yd=0.3*sin(t);

% Trayectoria Espiral
% xd=0.1*t.*cos(t);
% yd=0.1*t.*sin(t);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Llamada a la función deriada %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[xd1,xd2,t1,t2]=derivada(xd,t);      % xd1,yd1 derivada primera de xd, yd
[yd1,yd2,t1,t2]=derivada(yd,t);      % xd2,yd2 derivada segunda de xd, yd

% Orientación deseada del móvil

for i=1:length(t1)
    zd(i)=atan(yd1(i)/xd1(i));
end

% Derivadas del ángulo de orientación

[zd1,zd2,t1,t2]=derivada(zd,t);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ALGORITMO DE CONTROL %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% r = Radio de las ruedas del móvil
% L = La mitad de la separación entre las dos ruedas
% x,y = Coordenadas del punto de referencia del móvil

```

```

% z = Ángulo de orientación del móvil
% wd,wi = Velocidades angulares de la rueda derecha e izquierda deseada
% alpha, beta = Parámetros del controlador discontinuo por linealización
% U1 y U2 variables de control, velocidad lineal y angular
% x1,y1 = Primera derivada de x y y respectivamente

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Condiciones iniciales %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r=0.05;                %Radio de las ruedas del robot
L=0.15;                %Separación entre cada rueda y el centro de masa

% Condiciones iniciales para trayectoria espiral
% x(1)=-0.1;
% y(1)=0;
% z(1)=0+pi;

% Condiciones iniciales para trayectoria circular
x(1)=0.305;
y(1)=0.01;
z(1)=pi/2+pi;

wd(1)=zd(1);          % Velocidad angular derecha inicial
wi(1)=zd(1);          % Velocidad angular izquierda inicial

U1(1)=(wd(1)+wi(1))*(r/2);    % Variable de control (velocidad lineal)
U2(1)=(wd(1)-wi(1))*(r/(2*L)); % Variable de control (Velocidad angular)
x1(1)=xd1(1);
y1(1)=yd1(1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CONTROLADOR %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:length(xd2)

    % Derivadas de los puntos cartesianos de la ubicación del móvil
    T1=[t(i) t(i+1)];
    t6(1)=T1(1,1);
    t6(2)=T1(1,2);

```

```

dy=seguimiento1([t6(1) t6(2)],[x(i) y(i) z(i)],wd(i),wi(i),r,L);
x1(i+1)=dy(1);
y1(i+1)=dy(2);

%control
%options=odeset;
[T2,Y2]=ode23(@seguimiento1,[t6(1) t6(2)],[x(i) y(i) z(i)],odeset,wd(i),wi(i),r,L);
x(i+1)=Y2(11,1);
y(i+1)=Y2(11,2);
z(i+1)=Y2(11,3);

% Variables de control
v1(i)=xd2(i)-alpha1*(x1(i)-xd1(i))-alpha0*(x(i)-xd(i));
v2(i)=yd2(i)-beta1*(y1(i)-yd1(i))-beta0*(y(i)-yd(i));

% Se halla U1
% PlantaRB contiene las ecuaciones diferenciales del robot móvil
[T,Y]=ode23(@plantaRB,[t(i) t(i+1)]],[U1(i)],odeset,v1(i),v2(i),z(i));
U1(i+1)=Y(11,1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Switcheo de control para U1=0 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if U1(i+1)==0;

    %% Linealización por realimentación estática

    %Control por linealización a las variables x y z (Phi)
    % Variables de control
    v1(i)=xd1(i)-alpha0*(x(i)-xd(i));
    v2(i)=zd1(i)-beta0*(zd(i)-z(i));

    U1(i+1)=(1/cos(z(i)))*v1(i);
    U2(i+1)= v2(i);

    wd(i+1)=(U1(i+1)+U2(i+1))/r;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Saturación de las señales de velocidad angular

```



```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Saturación de las señales de velocidad angular
if (wi(i+1))>20
    wi(i+1)=20;
else if wi(i+1)<(-20)
    wi(i+1)=-20;
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Tiempos de graficación
tm(i)=i;
tn(i)=i-1;
end

% Error cuadrático medio
ex(i)=[((xd(i)-x(i))^2)^(1/2)]*100/[abs((max(xd)-min(xd)))];
ey(i)=[((yd(i)-y(i))^2)^(1/2)]*100/[abs((max(yd)-min(yd)))];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Error en x y y
error_x(i)=(xd(i)-x(i));
error_y(i)=(yd(i)-y(i));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end

% Vectores de graficación
tn(i+1)=length(tn)+1;

tm1=tm(1:length(tm));
tm2=tm(2:length(tm));

```

```
ex=ex(2:length(ex));
ey=ey(2:length(ey));
```

```
error_x=error_x(2:length(error_x));
error_y=error_y(2:length(error_y));
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Media del error, desviación estándar y varianza
```

```
max_error_x = max(abs(error_x))
```

```
max_error_y = max(abs(error_y))
```

```
min_error_x = min(abs(error_x))
```

```
min_error_y = min(abs(error_y))
```

```
m_error_x = mean(error_x)
```

```
m_error_y = mean(error_y)
```

```
d_error_x = std(error_x)
```

```
d_error_y = std(error_y)
```

```
% v_error_x = var(error_x)
```

```
% v_error_y = var(error_y)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Graficación %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
figure
```

```
plot(xd,yd,'r',x,y,'k',xd(length(xd)),yd(length(yd)),'ko');grid on;
```

```
title('Trayectoria deseada y obtenida');
```

```
legend('Tray. deseada','Tray. obtenida','Posición deseada','location','Best')
```

```
xlabel('x(m)');
```

```
ylabel('y(m)');
```

```
figure
```

```
plot(tm2,ex,'r',tm2,ey,'k');grid on;
```

```
title('Error cuadrático medio en x y y en el tiempo');
```

```

legend('Error eje x','Error eje y','location','Best');
xlabel('Tiempo');
ylabel('error(m)');

```

```

figure
plot(tm2,error_x,'r',tm2,error_y,'k');grid on;
title('Error en x y y en el tiempo');
legend('Error eje x','Error eje y','location','Best');
xlabel('Tiempo');
ylabel('error(m)');

```

```

figure
plot(tn,wd);grid on;title('Velocidad angular rueda derecha deseada');
xlabel('Tiempo');
ylabel('wd(rad/seg)');

```

```

figure
plot(tn,wi);grid on;title('Velocidad angular rueda izquierda deseada');
xlabel('Tiempo');
ylabel('wi(rad/seg)');

```

B.2. Función plantaRB

Es la función encargada de simular el comportamiento del robot, con el fin de obtener la velocidad lineal adecuada del sistema.

```

function dU1=plantaRB(t,U1,v1,v2,teta)

%Ecuación diferencial de dU1
% Se utiliza cuando U1 es diferente de cero (realimentación dinámica)

dU1(1)=cos(teta)*v1+sin(teta)*v2;

```

B.3. Función derivada

Es la función encargada de encontrar las primeras y segundas derivadas de las señales en el programa principal.

```

% Función que determina la derivada del vector A
%de longitud dos y tiempo de derivación t

```

```

function[B,C,t1,t2]=derivada(A,t)
h=(max(t)-min(t))/length(t); % Intervalo de integración

B=diff(A)/h; % Primera derivada
C=diff(B)/h; % Segunda derivada
C=(round(C*1000))/1000; % aproximaciones
B=(round(B*1000))/1000;

% Tiempo obtenido para la primera derivada
t1=linspace(t(1),t(length(t)),length(B));
% Tiempo obtenido para la segunda derivada
t2=linspace(t(1),t(length(t)),length(C));

```

B.4. Función seguimiento1

Es la función encargada de simular el comportamiento del robot móvil en el programa principal.

```

function dy=seguimiento1(t,y,wd,wi,r,L)

% Ecuación diferencial que representa la cinemática de un robot móvil,
% siendo y(1),y(2) la posición de referencia del móvil, y(3) el ángulo de
% orientación del móvil

dy=zeros(3,1);
dy(1)=((wd+wi)*r*cos(y(3)))/2;
dy(2)=((wd+wi)*r*sin(y(3)))/2;
dy(3)=((wd-wi)*r)/(2*L);

```

C. Anexo: Algoritmo para cobertura completa

C.1. Función principal

A continuación se muestra el programa principal, el cual contiene el controlados y llama a las funciones encargadas de planificar el movimiento horizontal y el movimiento curvo.

```
clear all, clc, close all;
syms a b c d
%Definición de condiciones iniciales
xi=-0.5;
yi=0.8;

%Definición del espacio de trabajo
lsx=1;
lix=-1;
lsy=1.5;
liy=-0.8;

tx=sqrt((lsx-lix)^2);
ty=sqrt((lsy-liy)^2);

lc=min(tx,ty);
ll=max(tx,ty);

d11=sqrt((xi-lix)^2);
d12=sqrt((xi-lsx)^2);
d21=sqrt((yi-liy)^2);
d22=sqrt((yi-lsy)^2);

%Condición para ir hacia el lado más corto en la dimensión 1
if(d11<=d12)
    %Desplazarse hacia la izquierda
```

```

vertical=a;
i=1;
x1(1)=xi;
while(x1(i)>=((lix+0.15)+0.1))
    x1(i+1)=x1(i)-0.05;
    i=i+1;
end
y1=yi*ones(1,length(x1));
else
%Desplazarse hacia la derecha
vertical=b;
i=1;
x1(1)=xi;
while(x1(i)<=((lsx-0.15)-0.1))
    x1(i+1)=x1(i)+0.05;
    i=i+1;
end
y1=yi*ones(1,length(x1));
end

%Condición para ir hacia el lado más corto en la dimensión 2
if(d21<=d22)
    %Desplazarse hacia abajo
    horizontal=d;
    i=1;
    y2(1)=y1(1,length(y1))-0.1;
    while(y2(i)>=((liy+0.15)+0.1))
        y2(i+1)=y2(i)-0.05;
        i=i+1;
    end
    if(vertical==a)
        x2=(x1(1,length(x1))-0.1)*ones(1,length(y2));
    else if(vertical==b)
        x2=(x1(1,length(x1))+0.1)*ones(1,length(y2));
    end
end

else
%Desplazarse hacia arriba
horizontal=c;

```



```
i=1;
y2(1)=y1(1,length(y1))+0.1;
while(y2(i)<=((lsy-0.15)-0.1))
    y2(i+1)=y2(i)+0.05;
    i=i+1;
end
if(vertical==a)
    x2=(x1(1,length(x1))-0.1)*ones(1,length(y2));
else if(vertical==b)
    x2=(x1(1,length(x1))+0.1)*ones(1,length(y2));
end
end
end

% Puntos para la curva de Bezier 1
%-----
%-----

punto1 = [x1(1,length(x1)) y1(1,length(y1))];

if(vertical==a)
    punto2 = [x1(1,length(x1))-0.1 y1(1,length(y1))];
else if(vertical==b)
    punto2 = [x1(1,length(x1))+0.1 y1(1,length(y1))];
end
end

punto3 = [x2(1,1) y2(1,1)];
puntos = [punto1; punto2; punto3];

p_bez = bezier1(3,puntos);

p_bez = p_bez';

xts=cat(2,x1,p_bez(1,:),x2);
yts=cat(2,y1,p_bez(2,:),y2);

figure
plot(xts,yts,'r',xi,yi,'bo')
hold on
```

```

grid on
axis([lix-0.2*abs(lix) lxs+0.2*abs(lix) liy-0.2*abs(liy) lsy+0.2*abs(liy)]);
legend('Recorrido inicial del robot','Posición inicial');
title('Trayectoria inicial para posicionamiento con curvas de Bezier');
xlabel('Abscisa (m)');
ylabel('Ordenada (m)');

%-----
%-----
%Variable que indica que llego al límite en y,
%cuando vale cero no ha llegado al límite
x=0;

%% Movimiento x
m=1;
j=1;
% j=mod(m,2);
[xa,ya]=mov_x(j,vertical,horizontal,xts,yts,lxs,lix,lsy,liy,m);

figure
plot(xa,ya,'r')
hold on
grid on
axis([lix-0.2*abs(lix) lxs+0.2*abs(lix) liy-0.2*abs(liy) lsy+0.2*abs(liy)]);
title('Recorrido 1 de aseo del robot');
xlabel('Abscisa (m)');
ylabel('Ordenada (m)');

punto1 = [xts(length(xts)) yts(length(yts))];

if(horizontal==c)
    punto2 = [xts(length(xts)) yts(length(xts))+0.1];
else
    punto2(2) = [xts(length(xts)) yts(length(xts))-0.1];
end

punto3 = [xa(1,1) ya(1,1)];
puntos = [punto1; punto2; punto3];

p_bez1 = bezier1(3,puntos);

```

```
p_bez1 = p_bez1';

xts1=cat(2,xts,p_bez1(1,:),xa);
yts1=cat(2,yts,p_bez1(2,:),ya);

figure
plot(xts1,yts1,'r',xi,yi,'bo')
hold on
grid on
axis([lix-0.2*abs(lix) lxs+0.2*abs(lix) liy-0.2*abs(lix) lsy+0.2*abs(lix)]);
legend('Recorrido inicial del robot','Posición inicial');
title('Trayectoria inicial para posicionamiento con curvas de Bezier');
xlabel('Abscisa (m)');
ylabel('Ordenada (m)');

%% Movimiento curvo
x=0;
[p_bez2,x]=mov_c(horizontal,xa,ya,lxs,lix,lsy,liy,x);

xta=cat(2,xa,p_bez2(1,:));
yta=cat(2,ya,p_bez2(2,:));

figure
plot(xta,yta,'r')
hold on
grid on
axis([lix-0.2*abs(lix) lxs+0.2*abs(lix) liy-0.2*abs(lix) lsy+0.2*abs(lix)]);
title('Recorrido de aseo del robot');
xlabel('Abscisa (m)');
ylabel('Ordenada (m)');

j=2;
while(x==0)
    %% Movimiento z
    m=m+1;

    [xa2,ya2]=mov_x(j,vertical,horizontal,xta,yta,lxs,lix,lsy,liy,m);
    xta=cat(2,xta,xa2);
```

```
yta=cat(2,yta,ya2);
clear xa2 ya2

%% Movimiento y

[p_bez3,x]=mov_c(horizontal,xta,yta,lsx,lix,lsy,liy,x);
xta=cat(2,xta,p_bez3(1,:));
yta=cat(2,yta,p_bez3(2,:));
clear p_bez3

end

figure
plot(xta,yta,'r')
hold on
grid on
axis([lix-0.2*abs(lix) lsx+0.2*abs(lix) liy-0.2*abs(lix) lsy+0.2*abs(lix)]);
title('Recorrido de aseo del robot2');
xlabel('Abscisa (m)');
ylabel('Ordenada (m)');

%% Movimiento z
m=m+1;

[xa2,ya2]=mov_x(j,vertical,horizontal,xta,yta,lsx,lix,lsy,liy,m);
xta=cat(2,xta,xa2);
yta=cat(2,yta,ya2);
clear xa2 ya2

figure
plot(xta,yta,'r')
hold on
grid on
axis([lix-0.2*abs(lix) lsx+0.2*abs(lix) liy-0.2*abs(lix) lsy+0.2*abs(lix)]);
title('Recorrido de aseo del robot final');
xlabel('Abscisa (m)');
ylabel('Ordenada (m)');
```



```
figure
plot(tn,wi);grid on;title('Velocidad angular rueda izquierda deseada');
xlabel('Tiempo');
ylabel('wi(rad/seg)');
```

C.2. Función de movimiento horizontal

En esta sección se muestra la función encargada de generar el movimiento horizontal del *Back-and-Forth motion*.

```
function [xa,ya]=mov_x(j,vertical,horizontal,xt,yt,lsx,lix,lsy,liy,m)
% Función que realiza el movimiento horizontal del robot
```

```
syms a b c d
```

```
if(vertical==a)
%Se utiliza para ver si es movimiento impar o par
%y así determinar la dirección de éste
    if((mod(m,2))==1)
        %Empezó en esquina A ó esquina C y debe ir hacia la derecha
        %Movimiento impar
        i=1;
        if(j==1)
            xa(1)=xt(length(xt))+0.1;
        else
            xa(1)=xt(length(xt));
        end
        while(xa(i)<=((lsx-0.15)-0.15))
            xa(i+1)=xa(i)+0.05;
            i=i+1;
        end
        if(horizontal==c)
            if(j==1)
                ya=(yt(length(yt))+0.1)*ones(1,length(xa));
            else
                ya=(yt(length(yt)))*ones(1,length(xa));
            end
        else if(horizontal==d)
```

```
        if(j==1)
            ya=(yt(length(yt))-0.1)*ones(1,length(xa));
        else
            ya=(yt(length(yt)))*ones(1,length(xa));
        end
    end
end

else
    %Movimiento par
    i=1;
    if(j==1)
        xa(1)=xt(length(xt))-0.1;
    else
        xa(1)=xt(length(xt));
    end
    while(xa(i)>=((lix+0.15)+0.15))
        xa(i+1)=xa(i)-0.05;
        i=i+1;
    end
    if(horizontal==c)
        if(j==1)
            ya=(yt(length(yt))+0.1)*ones(1,length(xa));
        else
            ya=(yt(length(yt)))*ones(1,length(xa));
        end
    else if(horizontal==d)
        if(j==1)
            ya=(yt(length(yt))-0.1)*ones(1,length(xa));
        else
            ya=(yt(length(yt)))*ones(1,length(xa));
        end
    end
end

end

else
    if((mod(m,2))==1)
        %Movimiento impar
        %Empezó en esquina B ó esquina D y debe ir hacia la izquierda
```

```

i=1;
if(j==1)
    xa(1)=xt(length(xt))-0.1;
else
    xa(1)=xt(length(xt));
end
while(xa(i)>=((lix+0.15)+0.15))
    xa(i+1)=xa(i)-0.05;
    i=i+1;
end
if(horizontal==c)
    if(j==1)
        ya=(yt(length(yt))+0.1)*ones(1,length(xa));
    else
        ya=(yt(length(yt)))*ones(1,length(xa));
    end
else if(horizontal==d)
    if(j==1)
        ya=(yt(length(yt))-0.1)*ones(1,length(xa));
    else
        ya=(yt(length(yt)))*ones(1,length(xa));
    end
end
end
end
else
    %Movimiento par
    i=1;
    if(j==1)
        xa(1)=xt(length(xt))+0.1;
    else
        xa(1)=xt(length(xt));
    end
    while(xa(i)<=((lsx-0.15)-0.15))
        xa(i+1)=xa(i)+0.05;
        i=i+1;
    end
    if(horizontal==c)
        if(j==1)
            ya=(yt(length(yt))+0.1)*ones(1,length(xa));
        else

```



```
p_bez2 = p_bez2';

else
    %Esta cerca de la vertical b y es movimiento curvo 2
    punto1 = [xt(length(xt)) yt(length(yt))];

    punto2 = [xt(length(xt))+dr yt(length(yt))-dr];

    punto3 = [xt(length(xt)) yt(length(yt))-2*dr];
    puntos = [punto1; punto2; punto3];

    p_bez2 = bezier1(3,puntos);

    p_bez2 = p_bez2';

end
else
    if(de1<de2)
        %Esta cerca de la vertical a y es movimiento curvo 1
        punto1 = [xt(length(xt)) yt(length(yt))];

        punto2 = [xt(length(xt))-0.15 yt(length(yt))-0.15];

        punto3 = [xt(length(xt)) yt(length(yt))-0.3];
        puntos = [punto1; punto2; punto3];

        p_bez2 = bezier1(3,puntos);

        p_bez2 = p_bez2';

    else
        %Esta cerca de la vertical b y es movimiento curvo 2
        punto1 = [xt(length(xt)) yt(length(yt))];

        punto2 = [xt(length(xt))+0.15 yt(length(yt))-0.15];

        punto3 = [xt(length(xt)) yt(length(yt))-0.3];
        puntos = [punto1; punto2; punto3];
```

```
        p_bez2 = bezier1(3,puntos);

        p_bez2 = p_bez2';

    end

end

else

%Empezó en esquina C ó esquina D y debe subir
if(sqrt((yt(length(yt))-lsy)^2)<0.3)
    x=2;          %Llegó a la parte superior
    dr=sqrt((yt(length(yt))-lsy)^2)/2;
    if(de1<de2)
        %Esta cerca de la vertical a y es movimiento curvo 1
        punto1 = [xt(length(xt)) yt(length(yt))];

        punto2 = [xt(length(xt))-dr yt(length(yt))+dr];

        punto3 = [xt(length(xt)) yt(length(yt))+2*dr];
        puntos = [punto1; punto2; punto3];

        p_bez2 = bezier1(3,puntos);

        p_bez2 = p_bez2';

    else
        %Esta cerca de la vertical b y es movimiento curvo 2
        punto1 = [xt(length(xt)) yt(length(yt))];

        punto2 = [xt(length(xt))+dr yt(length(yt))+dr];

        punto3 = [xt(length(xt)) yt(length(yt))+2*dr];
        puntos = [punto1; punto2; punto3];

        p_bez2 = bezier1(3,puntos);

        p_bez2 = p_bez2';

    end

end

else
    if(de1<de2)
```

```

    %Esta cerca de la vertical a y es movimiento curvo 1
    punto1 = [xt(length(xt)) yt(length(yt))];

    punto2 = [xt(length(xt))-0.15 yt(length(yt))+0.15];

    punto3 = [xt(length(xt)) yt(length(yt))+0.3];
    puntos = [punto1; punto2; punto3];

    p_bez2 = bezier1(3,puntos);

    p_bez2 = p_bez2';

else
    %Esta cerca de la vertical b y es movimiento curvo 2
    punto1 = [xt(length(xt)) yt(length(yt))];

    punto2 = [xt(length(xt))+0.15 yt(length(yt))+0.15];

    punto3 = [xt(length(xt)) yt(length(yt))+0.3];
    puntos = [punto1; punto2; punto3];

    p_bez2 = bezier1(3,puntos);

    p_bez2 = p_bez2';

end
end
end

```

C.4. Función generadora de curvas de Bezier

En esta sección se muestra la función encargada de generar las curvas de Bezier a partir de 3 puntos.

```
function p_bez = bezier1(n,puntos)
```

```
nplot = 20;
```

```
t = linspace(0,1,nplot);
```

```
for i = 1:n
    p(i,:) = puntos(i,:);
hold off;
[X,Y,p_bez] = CASTELJAU(0,1,p,t);
end
```

Bibliografía

- [1] Jerome Gaysse. A low cost absolute 2d position calculation system. In *SICE-ICASE International Joint Conference 2006*, 2006.
- [2] Zhangjun Song, Huifen Liu, Jianwei Zhang, Liwei Wang, and Ying Hu. A localization algorithm for low-cost cleaning robots based on kalman filter. In *8th World Congress on Intelligent Control and Automation*, pages 1450 – 1455, 2010.
- [3] Tsutomu Takeshita, Tetsuo Tomizawa, and Akihisa Ohya. A house cleaning robot system –path indication and position estimation using ceiling camera–. Technical report, Intelligent Robot Laboratory, Univ. of Tsukuba, Tsukuba, Japan.
- [4] Simon X. Yang and Chaomin Luo. A neural network approach to complete coverage path planning. *IEEE TRANSACTIONS ON SYSTEMS*, 34:718 – 725, 2004.
- [5] Chaomin Luo and Simon X. Yang. A real-time cooperative sweeping strategy for multiple cleaning robots. In *2002 IEEE International Symposium on Intelligent Control*, pages 660 – 665, 2002.
- [6] E. Prassler, D. Schwammkrug, B. Rohrmoser, and G. Schmidl. A robotic road sweeper. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 3, pages 2364 – 2369, 2000.
- [7] R.D Schraft, U Bräuning, T Orlowski, and M Hornemann. Automated cleaning of windows on standard facades. *Automation in Construction*, 9(5–6):489 – 501, 2000.
- [8] Anil Mital, Mangesh Kulkarni, Ronald Huston, and Sam Anand. Robot cleaning of underground liquid storage tanks: Feasibility and design considerations. *Robotics and Autonomous Systems*, 20(1):49 – 60, 1997.
- [9] Chaomin Luo and Simon X. Yang. A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments. *IEEE Transactions on Neuronal Networks*, 19:1279 – 1298, 2008.
- [10] Xuena Qiu, Jiatao Song, Xuejun Zhang, and Shirong Liu. A complete coverage path planning method for mobile robot in uncertain environments. In *Proc. 6th World Congress on Intelligent Control and Automation*, pages 8892 – 8896, 2006.

-
- [11] Wang Meiting, Tan Shili, Ding Junjian, and Yan Liwen. Complete coverage path planning of wall-cleaning robot using visual sensor. In *The Eighth International Conference on Electronic Measurement and Instruments*, pages 159 – 164, 2007.
- [12] Chaomin Luo, Simon X. Yang, and Max Q.-H. Meng. Neurodynamics based complete coverage navigation with real-time map building in unknown environments. In *International Conference on Intelligent Robots and Systems*, 2006.
- [13] Chaomin Luo, Simon X. Yang, and Xiaobu Yuan. Real-time area-covering operations with obstacle avoidance for cleaning robots. In *IEEE International Conference on Intelligent Robots and Systems*, 2002.
- [14] Pedro Rocha and A.Miguel Gomes. A decomposition approach for the complete coverage path planning problem. Technical report, INESCPorto, Faculdade de Engenharia, Universidade do Porto, 2008.
- [15] Yutian Mao, Lihua Dou, Jie Chen, Hao Fang, Haiqiang Zhang, and Hu Cao. Combined complete coverage path planning for autonomous mobile robot in indoor environment. In *7th Asian Control Conference*, pages 1468 – 1473, 2009.
- [16] Joon Seop Oh, Yoon Ho Choi, Jin Bae Park, and Yuan F. Zheng. Complete coverage navigation of cleaning robots using triangular-cell-based map. *IEEE Transactions on Industrial Electronics*, 51:718 – 726, 2004.
- [17] Zhang Chibin, Wang Xingsong, and Du Yong. Complete coverage path planning based on ant colony algorithm. In *International conference on Mechatronics and Machine Vision in Practice*, pages 357 – 361, 2008.
- [18] Adiyabaatar Janchiv, Dugarjav Batsaikhan, Gook hwan Kim, and Soon-Geul Lee. Complete coverage path planning for multi-robots based on. In *11th International Conference on Control, Automation and Systems*, pages 824 – 827, 2011.
- [19] Marten Waanders. Coverage path planning for mobile cleaning robots. In *15th Twente Student Conference on IT*, 2011.
- [20] Paulo A. Jimenez, Bijan Shirinzadeh, Ann Nicholson, and Gursel Alici. Optimal area covering using genetic algorithms. In *IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics*, pages 1 – 5, 2007.
- [21] Raphael Mannadiar and Ioannis Rekleitis. Optimal coverage of a known arbitrary environment. In *2010 IEEE International Conference on Robotics and Automation*, 2010.

-
- [22] Jung Won Kang, Si Jong Kim, Myung Jin Chung, Hyun Myung, Jun Ho Park, and Seok Won Bang. Path planning for complete and efficient coverage operation of mobile robots. In *IEEE International Conference on Mechatronics and Automation*, 2007.
- [23] Abdallah Ntawumenyikizaba, Hoang Huu Viet, and TaeChoong Chung. An online complete coverage algorithm for cleaning robots based on boustrophedon motions and a* search. Technical report, Dept. of Computer Engineering, Kyung Hee University, South Korea, 2012.
- [24] K.S. Senthilkumar and K. K. Bharadwaj. Spanning tree based terrain coverage by multi robots in unknown environments. Technical report, School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, 2008.
- [25] Jieru Chi, Guowei Yang, and Jie Yang. The on-line coverage algorithm and localization technique of the intelligent cleaning robot. Technical report, Automation & Engineering College, Qingdao University, China, 2007.
- [26] G. Zhang, S. Ferrari, and M. Qian. An information roadmap method for robotic sensor path planning. *Intelligent and Robotic Systems*, 56:69 – 98, 2009.
- [27] Yu Liu, Xiaoyong Lin, and Shiqiang Zhu. Combined coverage path planning for autonomous cleaning robots in unstructured environments. In *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, pages 8271 – 8276, 2008.
- [28] Anibal Ollero Baturone. *Robótica Manipuladores y Robots Móviles*. 2001.
- [29] José Felipe Camarena García. Análisis cinemático, dinámico y control en tiempo real de un vehículo guiado automáticamente. Master’s thesis, Centro Nacional de Investigación y Desarrollo Tecnológico, Departamento de Mecatrónica, Cuernavaca, Morelos, México, 2009.
- [30] N. J. Nilsson. Shakey the robot. Technical report, Artificial Intelligence Center, Computer Science and Technology Division, SRI International, Menlo Park, CA, 1984.
- [31] R. Hollis. Newt: a mobile, cognitive robot. *Byte*, 2:30–45, 1997.
- [32] G. Giralt, R. Sobek, and R. Chatila. A multi-level planning and navigation system for a mobile robot; a first approach to hilare. In Japan Tokyo, editor, *IJCAI*, pages 335–338, 1979.
- [33] C. Canudas de Wit, B. Siciliano, and G. Bastin. Theory of robot control. *Springer-Verlag, London*, 1996.

-
- [34] G. Campion, G. Bastin, and B. D’Andrea-Novel. Structural properties and classification of kinematics and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation, Sacramento, California*, 1991.
- [35] Hebertt Sira-Ramírez, Richard Marquez, Franklin Rivas-Echevarría, and Orestes Llanes-Santiago. Control de sistemas no lineales. In *Control Automático*, 2011.
- [36] Mauricio Fernando Jaramillo Morales. Planificador de trayectorias para un robot móvil orientado a la reconstrucción en 3d de entornos desconocidos. Master’s thesis, Universidad Nacional de Colombia Sede Manizales, 2010.
- [37] R. Silva Ortigoza, M. A. Molina Vilchis, V. M. Hernández Guzmán, G. Silva Ortigoza, M. Marciano Melchor, and E. A. Portilla Flores. Modelado y control de un robot móvil tipo newt en la tarea de seguimiento de trayectira. *Revista Electrónica de Estudios Telemáticos*, 7:129 – 145, 2008.
- [38] Eduardo Aranda Bricaire, Tomás Salgado Jiménez, and Martin Velasco Villa. Control no lineal discontinuo de un robot móvil. *Computación y Sistemas*, Edición Especial:42 – 49, 2002.
- [39] Katsuhiko Ogata. *Ingeniería de Control Moderna*. 1998.
- [40] Y. Takahashi and O. Tsubouchi. Tension control of wire suspended mechanism and application to bathroom cleaning robot. In *SICE 2000. Proceedings of the 39th SICE Annual Conference. International Session Papers*, pages 143 – 147, 2000.
- [41] Jeong H. Lee, Jeong S. Choi, Beom H. Lee, and Kong W. Lee. Complete coverage path planning for cleaning task using multiplerobots. *IEEE International Conference on Systems*, pages 3618 – 3622, 2009.
- [42] T. Palleja, M. Tresanchez, M. Teixido, and J. Palacin. Modeling floor-cleaning coverage performances of some domestic mobile robots in a reduced scenario. *Robotics and Autonomous Systems*, 58:37 – 45, 2010.
- [43] Chaomin Luo, Yang S.X., and Stacey D.A. Real-time path planning with deadlock avoidance of multiple cleaning robots. *Robotics and Automation, 2003. Proceedings. ICRA ’03. IEEE International Conference on*, 3:4080 – 4085, 2003.
- [44] Eunil Park, Ki Joon Kim, and Angel P. del Pobil. Energy efficient complete coverage path planning for vacuum cleaning robots. *Future Information Technology, Application, and Service Lecture Notes in Electrical Engineering*, 164:23 – 31, 2012.
- [45] Cheng Fang and S. Anstee. Coverage path planning for harbour seabed surveys using an autonomous underwater vehicle. In *OCEANS 2010 IEEE - Sydney*, pages 1–8, 2010.

-
- [46] Miri Weiss-Cohen, Igal Sirotin, and Erez Rave. Lawn mowing system for known areas. *IEEE Transactions Computer Society*, pages 539 – 544, 2008.
- [47] Timo Oksanen and Arto Visala. Coverage path planning algorithms for agricultural field machines. *Field Robotics*, 26(8):651 – 668, 2009.
- [48] Guoyu Zuo, Peng Zhang, and Junfei Qiao. Path planning algorithm based on sub-region for agricultural robot. In *Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on*, volume 2, pages 197 – 200, 2010.
- [49] T. Bohme, U. Schmucker, N. Elkmann, and M. Sack. Service robots for facade cleaning. In *Industrial Electronics Society, 1998. IECON '98. Proceedings of the 24th Annual Conference of the IEEE*, volume 2, pages 1204 – 1207, 1998.
- [50] M. Simoncelli, G. Zunino, H.I. Christensen, and K. Lange. Autonomous pool cleaning: Self localization and autonomous navigation for cleaning. *Autonomous Robots*, 9(3):261 – 270, 2000.
- [51] E. Prassler, E. Stroulia, and M. Strobel. Office waste cleanup: an application for service robots. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 3, pages 1863 – 1868, 1997.
- [52] Howie Choset. Coverage for robotics – a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):113–126, 2001.
- [53] Xuena Qiu, Shirong Liu, and Simon X. Yang. A rolling method for complete coverage path planning in uncertain environments. In *Proc. IEEE International Conference on Robotics and Biomimetics*, pages 146 – 151, 2004.
- [54] Guoqing Hu, Zhengwei Hu, and Hongbo Wang. Complete coverage path planning for road cleaning robot. *IEEE Transactions*, pages 643 – 648, 2009.
- [55] Hammad Naeem. Coverage path planning for automated inspection of known environment. Master’s thesis, Department of Computer Architecture and Technology University of Girona, 2011.
- [56] Ping-Min Hsu, Chun-Liang Lin, and Meng-Yao Yang. On the complete coverage path planning for mobile robots. *Intelligent & Robotic Systems*, July 2013.
- [57] E. Garcia and P. Gonzalez de Santos. Mobile-robot navigation with complete coverage of unstructured environments. *Robotics and Autonomous Systems*, 46(4):195 – 204, 2004.

-
- [58] Tae-Kyeong, Lee Sang-Hoon, Baek Se-Young Oh, and Young-Ho Choi. Complete coverage algorithm based on linked smooth spiral paths for mobile robots. In *Proc. International Conference Control, Automation, Robotics and Vision*, pages 609 – 615, 2010.
- [59] Aydin Sipahioglu, Gokhan Kirlik, Osman Parlaktuna, and Ahmet Yazici. Energy constrained multi-robot sensor-based coverage path planning using capacitated arc routing approach. *Robotics and Automation Systems*, 58:529 – 538, 2010.
- [60] Shuang Liu, Dong Sun, and Changan Zhu. Coordinated motion planning for multiple mobile robots along designed paths with formation requirement. *Mechatronics, IEEE/ASME Transactions on*, 16(6):1021 – 1031, 2011.
- [61] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli. The sensor-based random graph method for cooperative robot exploration. *Mechatronics, IEEE/ASME Transactions on*, 14(2):163–175, 2009.
- [62] Zhiyang Yao. Finding efficient robot path for the complete coverage of a known space. In *International Conference on Intelligent Robots and Systems*, 2006.