

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO ENSENADA



INGENIERÍA EN ELECTRÓNICA

**SINCRONIZACIÓN TEMPORAL ESPACIALMENTE DEPENDIENTE DE
ROBOTS MÓVILES SIMULANDO UN ENTORNO LUCIÉRNAGA**

**TESIS QUE PARA OBTENER EL GRADO DE INGENIERO EN
ELECTRÓNICA PRESENTA**

BRIAN JAVIER AGUILERA CEDILLO

Ensenada, B.C., México. Marzo 2024

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO UNIDAD ENSENADA

SINCRONIZACIÓN TEMPORAL ESPACIALMENTE DEPENDIENTE DE ROBOTS
MÓVILES SIMULANDO UN ENTORNO LUCIÉRNAGA

TESIS

Que para obtener el grado de ingeniero en electrónica presenta

BRIAN JAVIER AGUILERA CEDILLO

Y aprobada por el siguiente comité:

Dr. Adrian Arellano Delgado

Director de comité

Dr. Fausto Abundiz Pérez

Miembro del comité

Dra. Liliana Cardoza Avendaño

Miembro del comité

Dr. Miguel Ángel Murillo Escobar

Miembro del comité

Dr. César Cruz Hernández

Miembro de comité

Ensenada, B.C., México. Marzo 2024

Dedicatoria

A mis padres

María Yolanda Cedillo Rodríguez

y

Juan Manuel Aguilera Anguiano,

A mis hermanas

Cindy Vianey Aguilera Cedillo

y

Michelle Deny Aguilera Cedillo.

“Por su constante apoyo familiar a lo largo de mi vida”.

Agradecimientos

Al Dr. Adrian Arellano Delgado, por la ayuda como director de tesis con su tiempo para aclaramiento de dudas y guía el desarrollo de esta tesis.

A la Dra. Rosa Martha López Gutiérrez, por toda la amabilidad y apoyo ofrecido durante el transcurso de mi carrera de ingeniería en electrónica y ofrecer la oportunidad de realizar esta tesis.

Al Dr. César Cruz Hernández, por el apoyo con la aprobación del tema de tesis y la coordinación para que se me fuera asignado.

A los miembros del comité de sistemas complejos, por la elección, aprobación y conocimiento previo acerca de este tema de tesis para poder haberla realizado.

A la Universidad Autónoma de Baja California (UABC) campus Sauzal, por darme la oportunidad de cursar y terminar la licenciatura de ingeniería en electrónica en la Facultad de Ingeniería, Arquitectura y Diseño (FIAD).

Al Consejo Nacional de Humanidades, Ciencia y Tecnología (CONAHCYT) por el apoyo económico recibido a través del proyecto de Investigación en Ciencia Básica entre instituciones, “Sincronización de Sistemas Complejos y Algunas Aplicaciones”. Ref. 166654 y continuación (A1-S-31628).

RESUMEN de la tesis de **Brian Javier Aguilera Cedillo**, presentada como requisito para obtener el grado de INGENIERO EN ELECTRÓNICA. Ensenada, B.C. México, Marzo 2024.

Sincronización temporal espacialmente dependiente de robots móviles simulando un entorno luciérnaga

Resumen aprobado por:



Dr. Adrian Arellano Delgado

Director de tesis

En esta propuesta de tesis de licenciatura, se plantea realizar una sincronización temporal espacialmente dependiente con robots móviles de tipo tracción diferencial los cuales simularán un entorno de una red de luciérnagas. En particular, se tienen un grupo de robots móviles terrestres en los que se pretende alcanzar la sincronización de diferentes señales oscilantes emitidas por cada robot móvil, esto se realizará en un entorno virtual a través de un simulador. Este tipo de sincronización se propone como una sincronización espacialmente dependiente de la posición de los robots móviles, es decir, la ley de control convencional propuesta para sincronizar las señales oscilantes acopladas a los robots será dependiente de la posición en el espacio de los robots que podrán estar sincronizados (en fase o anti fase) o no, a una trayectoria periódica o generada de manera aleatoria.

Palabras clave: Sincronización, Osciladores, Robot móvil, Linealización por retroalimentación dinámica.

ABSTRACT of the thesis presented by **Brian Javier Aguilera Cedillo**, as a requirement to obtain the ELECTRONIC ENGINEER degree. Ensenada, B.C. México, March 2024.

Spatially-dependent time synchronization of mobile robots simulating a firefly environment

Abstract approved by:



Dr. Adrian Arellano Delgado

Thesis Director

In this bachelor thesis proposal, it is proposed to perform a spatially dependent time synchronization with differential traction type mobile robots which will simulate an environment of a firefly network. In particular, we have a group of terrestrial mobile robots in which we intend to achieve the synchronization of different oscillating signals emitted by each mobile robot, this will be done in a virtual environment through a simulator. This type of synchronization is proposed as a spatially dependent synchronization of the position of the mobile robots, i.e., the conventional control law proposed to synchronize the oscillating signals coupled to the robots will be dependent on the position in space of the robots that may be synchronized (in phase or anti-phase) or not, to a periodic or randomly generated trajectory.

Keywords: Synchronization, Oscillators, Mobile robot, Linearization by dynamic feedback.

Tabla de Contenido

	Página
Dedicatoria	III
Agradecimientos	IV
Resumen	V
Abstract	VI
Lista de figuras	IX, X
Capítulo 1. Introducción	1
1.1. Motivación.	2
1.2. Hipótesis.	2
1.3. Propuesta de solución.	2
1.4. Objetivos generales.	2
1.5. Objetivos específicos.	2
1.6. Organización de la tesis.	3
1.7. Conclusiones del capítulo.	3
Capítulo 2. Preliminares	4
2.1. Sistemas.	4
2.2. Sistema dinámico.	4
2.3. Modelado matemático.	5
2.4. Sistemas lineales y no lineales.	6
2.5. Sistemas complejos.	6, 7
2.6. Teoría de grafos.	7, 8
2.7. Topología de conexión de estrella.	8, 9
2.8. Matriz asociada a un grafo.	9-11
2.9. Matriz de acoplamiento.	11
2.10. Conclusiones del capítulo.	11

Tabla de Contenido (continuación)

	Página
Capítulo 3. Modelo matemático de movimiento del robot móvil tipo tracción diferencial	12
3.1. Modelo de control.	12, 13
3.2. Simulación con destino fijo.	13-15
3.3. Simulación con ruta fija.	15-17
3.4. Conclusiones del capítulo.	17
Capítulo 4. Sistema generador de posiciones aleatorias	18
4.1. Sistema aleatorio con un robot diferencial.	18-20
4.2. Sistema anticollisiones con dos robots diferenciales.	21-23
4.3. Conclusiones del capítulo.	23
Capítulo 5. Sistema oscilatorio para la sincronización espacialmente dependiente	24
5.1. Método de sincronización.	24-26
5.2. Simulación de la sincronización.	26, 27
5.3. Conclusiones del capítulo.	27
Capítulo 6. Resultados finales del experimento	28
6.1. Resultados finales del movimiento en el plano.	28-31
6.2. Resultados finales de sincronización de ondas.	32-36
6.3. Conclusiones del capítulo.	36
Capítulo 7. Conclusiones	37
7.1. Conclusiones.	37
7.2. Trabajo futuro.	37
Literatura citada.	38
Anexos.	39-71

Lista de figuras

Figura	Página
1. Esquema de control de un sistema en lazo abierto.	4
2. Esquema de control de un sistema de lazo cerrado.	4
3. Red de comunicación de dos nodos (modo bidireccional).	7
4. Red de comunicación de dos nodos (modo unidireccional/maestro-esclavo).	8
5. Molécula de metano.	8
6. Topología de conexión de red tipo estrella o de árbol con 5 nodos (modo maestro/esclavo).	9
7. Plano de movimiento de un robot diferencial.	13
8. Trayectoria realizada en el plano por el robot diferencial para viajar a un solo punto (regulación).	14
9. Gráfica de movimiento vectorial realizado por el robot diferencial al viajar a un solo punto.	15
10. Trayectoria realizada en el plano de coordenadas (x,y) por el robot diferencial para hacer la figura de una rosa polar.	16
11. Gráfica de movimiento temporal realizado por el robot diferencial al realizar el seguimiento de la trayectoria de la rosa polar.	17
12. Trayectoria realizada en el plano de coordenadas (x,y) por el robot diferencial teniendo una serie de destinos diferentes.	19
13. Gráfica de movimiento temporal realizado por el robot diferencial al realizar el seguimiento de la trayectoria de los diferentes puntos generados.	20
14. Trayectoria realizada en el plano de coordenadas (x,y) por dos robots diferenciales teniendo rutas aleatorias.	21
15. Gráfica de movimiento temporal realizado por los robots diferenciales.	22

Lista de figuras (continuación)

Figura	Página
16. Gráfica del error (distancia) entre los dos robots diferenciales.	23
17. Dos robots acercándose pero lo suficientemente lejos para no activar el acoplamiento para la sincronización.	25
18. Dos robots acercándose y a la distancia indicada para aplicar el acoplamiento para la sincronización.	25
19. Dos robots alejándose a causa de las anticollisiones pero ya sincronizados.	26
20. Sincronización de dos osciladores con menor intensidad de acoplamiento ($C=0.0001$). . .	26
21. Sincronización de dos osciladores con mayor intensidad de acoplamiento ($C=0.0002$). . .	27
22. Trayectoria realizada por los 5 robots móviles sobre una superficie de 40x40 metros. . .	28
23. Gráficas de movimiento temporal de los 5 robots móviles.	29
24. Gráficas del error (distancia) que tienen los robots esclavos (robots 2, 3, 4 y 5) contra el robot 1.	30
25. Gráfica del error (o distancia) entre todos los robots.	31
26. Comparación de activación del sistema anticollisiones con la distancia entre los robots.	31
27. Gráficas de sincronización de 2 señales de los robots móviles 1-2 y 1-3.	32
28. Gráficas de sincronización de 2 señales de los robots móviles 1-4 y 1-5.	33
29. Gráficas de sincronización de 2 señales de los robots móviles 2-3 y 2-4.	33
30. Gráficas de sincronización de 2 señales de los robots móviles 2-5 y 3-4.	34
31. Gráficas de sincronización de 2 señales de los robots móviles 3-5 y 4-5.	34
32. Gráfica de las señales de activación de todas las sincronizaciones de los osciladores. . . .	35
33. Gráfica de la sincronización de las 5 señales de los robots móviles.	36

Capítulo 1. Introducción

Los inicios del origen de la idea de una máquina robotizada nos lleva tan lejos como hasta los mismos tiempos de la civilización más antigua con registros históricos la cual fue Mesopotamia, con la creación del primer reloj de agua el cual es considerado por muchos como la primera máquina robotizada de la historia.

A lo largo de la historia se verían muchos más diseños y conceptos similares que no contaban con una palabra ligada que los pudieran definir. No es sino hasta mucho tiempo más tarde en el año 1920 del calendario gregoriano cuando se le asimiló el nombre “robot” para poder definir a este tipo de máquinas las cuales pueden realizar diferentes actividades por si solas, este término provino de una obra de teatro por el autor Karel Čapek (1890-1938) llamada “R.U.R (Robots Universales Rossum)”, dicha palabra tiene origen Checo y viene de la palabra “robota” que significa “trabajo forzado”, con lo cual se nos da a entender que un robot es una máquina que nos ayuda a hacer trabajos sin necesidad de que un ser humano los realice.

En la actualidad, un robot es bastante sofisticado si se comparan a los de la antigüedad, puesto que cuentan con un cierto grado de visión debido a la introducción de sensores, capacidad de decisión a través de poder computacional y actuación que es implementado por las partes mecánicas móviles.

Por otra parte, en la naturaleza nos encontramos con una variedad de casos de sincronización que regularmente son utilizadas para mejorar la eficiencia de las acciones o trabajos, desde seres vivos como por ejemplo los pájaros que forman parvadas con el fin de optimizar la aerodinámica de su vuelo donde todos los integrantes se ven beneficiados, también está el caso de las escuelas de peces que viajan juntos para encontrar comida más fácilmente y mantenerse más seguros ante posibles depredadores. Estos tipos de sincronizaciones son de naturaleza social, que es uno de los comportamientos que se quiere reproducir en esta tesis, a saber, la sincronización que hacen los grupos de luciérnagas macho con sus luces al momento de cortejar a una hembra para reproducirse, debido a que las luciérnagas machos imitan el patrón de alumbramiento con el tiempo para encontrar en el intervalo de oscuridad a los destellos menos potentes de las hembras.

Esta tesis se centra en el trabajo alrededor de la sincronización de señales en el dominio del tiempo que emitirán robots móviles imitando el entorno luciérnaga anteriormente mencionado, los robots móviles cuentan con tracción diferencial por lo cual la velocidad de rotación de cada una de sus dos ruedas es independiente a la otra.

1.1. Motivación

La motivación para realizar esta tesis de licenciatura es la indagación en el tema del control de la sincronización de un conjunto de robots mediante sistemas complejos para simular el efecto que hacen los grupos de luciérnagas cuando estos se aproximan unos a otros, puesto que estos al encontrarse cerca imitan la frecuencia en la que destellan. De ahí surgió el interés por replicar dicho comportamiento pero esta vez recreado en un sistema de control con robots móviles los cuales pueden seguir trayectorias periódicas o aleatorias.

1.2. Hipótesis

Es posible alcanzar la sincronización de un modelo simple de un oscilador electrónico espacialmente dependiente de la posición de un robot móvil, es decir, en un entorno luciérnaga, aplicando un control adecuado.

1.3. Propuesta de solución

Los resultados numéricos serán presentados en simulaciones mediante programación, para lo cual, se utilizará el programa de Matlab para simular el proceso de la sincronización de señales ondulatorias a través de la creación de gráficas temporales dinámicas, de fase y de error entre los robots.

1.4. Objetivos generales

Analizar los comportamientos emergentes, en particular la sincronización, de las dinámicas de osciladores acoplados a robots móviles en un entorno luciérnaga, donde la sincronización es espacialmente dependiente del movimiento de los robots móviles.

1.5. Objetivos específicos

- Análisis y simulación del modelo matemático del robot móvil tipo tracción diferencial.
- Analizar y simular el sistema de posiciones aleatorias para el movimiento de los robots móviles del tipo tracción diferencial.
- Analizar y simular el sistema oscilatorio de posiciones aleatorias para la sincronización espacialmente dependiente.
- Analizar y simular la ley de control del sistema de posiciones aleatorias para la sincronización espacialmente dependiente de los sistemas oscilatorios.

1.6. Organización de la tesis

La memoria de esta tesis se encuentra organizada de la siguiente manera:

En el capítulo 2, se presentan los conocimientos necesarios que preceden la realización de esta tesis, para poder tener un conocimiento y entender en cierto grado acerca de los temas con los que se van a trabajar. En el capítulo 3, se habla en detalle acerca del modelo matemático utilizado para controlar el movimiento del robot y se presentara dicho movimiento simulado. En el capítulo 4, se presenta contenido acerca de la generación y seguimiento de rutas aleatorias para el robot de tracción diferencial al igual que su simulación. En el capítulo 5, se muestra el modelo de sincronización de onda sinusoidal entre dos robots diferenciales junto con su simulación. En el capítulo 6, se retoman los temas tratados en los capítulos 3, 4 y 5 a través de una demostración simulada de un ambiente de sincronización tipo luciérnaga con 5 robots diferenciales con trayectorias generadas de forma aleatoria con implementación de un sistema anticollisiones. Por último, en el capítulo 7 se dan algunas conclusiones con respecto a lo tratado al igual que posibles futuras mejoras.

1.7. Conclusiones del capítulo

En este primer capítulo se definieron los márgenes que abarcaran la tesis y los objetivos a los que se quiere llegar, además de explicar el tema que motivo a la realización de ésta, además, se presenta algo de trasfondo con información introductoria acerca del tema escogido y la situación que se quiere recrear con los robots móviles.

Capítulo 2. Preliminares

En este capítulo se presentarán términos y conceptos con los cuales se trabajará en esta tesis, al igual que la correspondiente definición de terminología y conceptos fundamentales.

2.1. Sistema

Un sistema es una combinación de componentes que actúan conjuntamente para alcanzar un objetivo específico. Un componente es una sola unidad funcional de un sistema. El concepto de sistema no se limita en absoluto al ámbito de los fenómenos físicos, sino que puede extenderse a fenómenos dinámicos abstractos, como los que se dan en la economía, el transporte, el crecimiento demográfico y la biología [1].

2.2. Sistema dinámico y estático

Un sistema es denominado dinámico si su salida actual depende de una entrada pasada; si su salida actual depende sólo de la entrada actual, el sistema se conoce como estático. La salida de un sistema estático permanece constante si el valor de entrada no cambia, por lo que la salida sólo cambia cuando cambia la entrada. En un sistema dinámico, la salida cambia con el paso del tiempo, siempre y cuando el sistema no se encuentre en un estado de equilibrio [1].

Un ejemplo de diseño en donde un sistema no varía el valor de su salida con respecto al tiempo (figura 1) y uno en el que sí es variable en el tiempo (figura 2), es a través de la implementación de una retroalimentación al sistema, puesto que este funciona como una señal adicional de entrada que se toma de la salida actual y funciona como un control para obtener una salida específica [2].

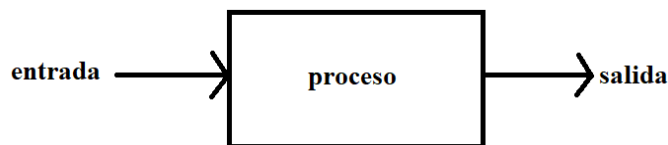


Figura 1: Esquema de control de un sistema en lazo abierto.

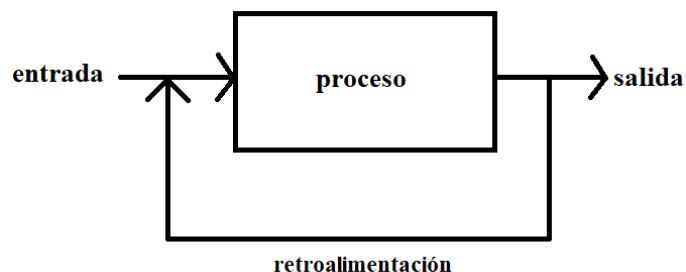


Figura 2: Esquema de control de un sistema de lazo cerrado.

2.3. Modelado matemático

Para poder diseñar un sistema se debe comenzar con una predicción de su rendimiento antes de que el propio sistema pueda diseñarse en detalle o construirse realmente. Esta predicción se basa en una descripción matemática de las características dinámicas del sistema. Esta descripción matemática se denomina modelo matemático. Para muchos sistemas físicos, los modelos matemáticos útiles se describen en términos de ecuaciones diferenciales [1].

El modelado matemático implica la descripción de características importantes del sistema mediante conjuntos de ecuaciones. Aplicando las leyes físicas a un sistema concreto, puede ser posible desarrollar un modelo matemático que describa la dinámica del sistema. Dicho modelo puede incluir parámetros desconocidos, que deben evaluarse mediante pruebas reales. Sin embargo, las leyes físicas que rigen el comportamiento de un sistema no están completamente definidas y la formulación de un modelo matemático puede resultar imposible. En tal caso, puede recurrirse a un proceso de modelización experimental. En este proceso, el sistema se somete a un conjunto de entradas conocidas y con el fin de medir sus salidas [1].

Al intentar construir un modelo matemático, hay que llegar a un compromiso entre la simplicidad del modelo y la precisión de los resultados del análisis. Es importante tener en cuenta que los resultados obtenidos del análisis sólo son válidos en la medida en que el modelo se aproxime a un sistema físico determinado.

Para determinar un modelo razonablemente simplificado, se debe decidir qué variables y relaciones físicas son despreciables y cuáles son cruciales para la precisión del modelo. Para obtener un modelo en forma de ecuaciones diferenciales lineales, deben ignorarse los parámetros distribuidos y las no linealidades que puedan estar presentes en el sistema físico. Si los efectos que estas propiedades ignoradas tienen sobre la respuesta son pequeños, entonces los resultados del análisis de un modelo matemático y los resultados del estudio experimental del sistema físico estarán en buena concordancia.

Normalmente, al resolver un problema nuevo, conviene construir primero un modelo simplificado para obtener una idea general sobre la solución. Después, se puede construir un modelo matemático más detallado y utilizarlo para un análisis más completo.

El modelo analizado es una descripción matemática aproximada del sistema físico; no es el sistema físico en sí. En realidad, ningún modelo matemático puede representar con completa precisión ningún componente o sistema físico.

Siempre hay aproximaciones y suposiciones. Estas aproximaciones y suposiciones limitan el margen de validez del modelo matemático, el grado de aproximación sólo puede determinarse experimentalmente, por lo cual entre más aproximado sea al sistema real más efectivo es el modelo creado.

2.4. Sistemas lineales y no lineales

Para los sistemas lineales, las ecuaciones que constituyen el modelo son lineales. La propiedad más importante de los sistemas lineales es que en ellos se logra aplicar el principio de superposición. El cual establece que la respuesta producida por aplicaciones simultáneas de dos o más entradas diferentes es la suma de dos respuestas individuales. Por consiguiente, en los sistemas lineales, la respuesta a varias entradas puede calcularse tratando una entrada cada vez y sumando después los resultados. Como resultado de la superposición, las soluciones complicadas de las ecuaciones diferenciales lineales pueden derivarse como una suma de soluciones simples. En un sistema dinámico, si la causa y el efecto son proporcionales, el sistema puede considerarse lineal. [3] [4].

Aunque las relaciones físicas suelen representarse mediante ecuaciones lineales, en muchos casos de situaciones reales estas relaciones pueden no ser del todo lineales. Por lo que para que puedan ser tratadas dentro de un sistema lineal son sólo utilizados en rangos de funcionamiento limitados. Por ejemplo, muchos sistemas hidráulicos hacen uso de relaciones no lineales entre sus variables, pero a menudo se representan mediante ecuaciones lineales dentro de rangos de funcionamiento limitados en donde se puedan comportar como tal.

En el caso de los sistemas no lineales, la característica más importante es que el principio de superposición no es aplicable. En general, los procedimientos para hallar soluciones a los problemas que plantean estos sistemas son extremadamente complicados. Frecuentemente es necesario linealizar un sistema no lineal cerca de las condiciones de funcionamiento. Una vez que un sistema no lineal se aproxima mediante un modelo matemático lineal, pueden utilizarse una serie de técnicas lineales con fines de análisis y diseño, y consecuentemente en un sistema de control.

2.5. Sistemas complejos

Los sistemas complejos son redes formadas por una serie de componentes que interactúan entre sí, normalmente de forma no lineal. Los sistemas complejos pueden surgir y evolucionar a través de auto organización, de modo que no son ni completamente regulares ni completamente aleatorios, lo que permite el desarrollo de comportamientos emergentes a escalas macroscópicas [5].

Estas propiedades de comportamientos pueden encontrarse en muchos sistemas del mundo real, por mencionar algunas están: los sistemas fisiológicos de un organismo, los cerebros y otros sistemas neuronales, los mercados bursátiles, las redes sociales y el clima. Este último teniendo el famosísimo efecto mariposa en donde explica que hasta el cambio más pequeño en una variable como lo es el aleteo de una mariposa puede influir al clima en gran escala, siendo esta la causa posible de la creación de un ciclón en el otro lado del mundo.

Para entender mejor qué son los sistemas complejos, puede ser útil saber qué no son. Un ejemplo de sistemas que no son complejos es un conjunto de componentes independientes, como el lanzamiento aleatorio de una moneda la cual es un evento totalmente aleatorio, ya que puede caer en cualquiera de sus dos lados haciéndolo completamente impredecible, y por lo tanto no tiene algún grado de control como el que tendría un sistema complejo.

2.6. Teoría de grafos

El comienzo de la teoría de grafos se considera que fue en el primer caso en donde se hizo uso de esta herramienta para resolver un problema matemático en donde Leonhard Paul Euler (1707-1783) en 1736 hizo la publicación de la solución de Euler al problema de los puentes de Königsberg. Los posteriores redescubrimientos de la teoría de grafos por Gustav Robert Kirchhoff (1824-1887) y Arthur Cayley (1821-1895) también tendrían sus raíces en el mundo físico.

Las investigaciones de Kirchhoff sobre las redes eléctricas lo llevaron a desarrollar los conceptos y teoremas básicos sobre las conexiones de tipo árbol en los grafos. Mientras que Cayley estudió los árboles resultantes de la enumeración de isómeros químicos orgánicos. Otro enfoque tipo rompecabezas de los grafos semejantes al problema que resolvió Euler, fue propuesto por William Rowan Hamilton (1805-1865). Después de éste, cobró relevancia la célebre conjetura de los cuatro colores, que ha sido notoria desde entonces. No es casualidad que la teoría de grafos haya sido imaginada varias veces por varios físicos y matemáticos, ya que puede considerarse como un área de las matemáticas aplicadas en donde su uso puede ser empleado en cualquier problema o situación que contenga dos o más elementos que se comunican entre sí.

Un grafo se define de manera matemática como un conjunto de valores finitos $G = (V, E)$, donde, V son el número de nodos, los cuales son la cantidad de entidades que van a interactuar entre sí en el grafo, y E que son la cantidad de conexiones que tienen los nodos o entidades entre ellas mostrando que nodos están directamente conectados [6],[7]. Estos son usados a menudo para analizar un sistema el cual posee múltiples partes que se encuentran interconectadas e intercomunicadas, entre más partes tenga un sistema más complicado se hace su análisis. A continuación se presenta de manera gráfica los dos tipos de comunicación que se puede tener en un grafo, un tipo es donde ambas partes se comunican entre sí mientras que en el otro tipo solo una parte se comunica con la otra (figura 3 y figura 4).

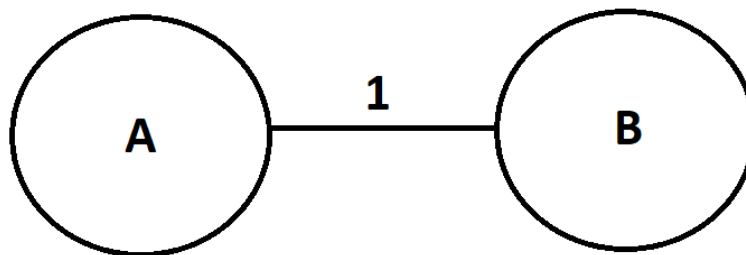


Figura 3: Red de comunicación de dos nodos (modo bidireccional).

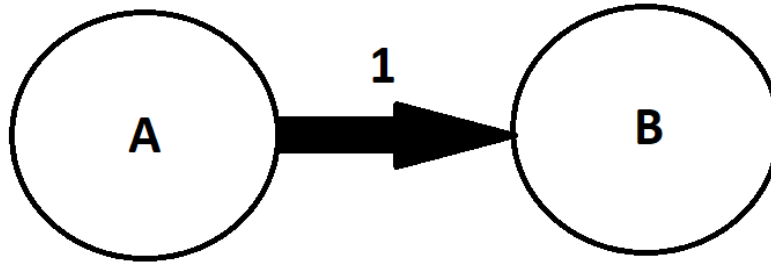


Figura 4: Red de comunicación de dos nodos (modo unidireccional/maestro-esclavo).

2.7. Topología de conexión de estrella

Este tipo de conexión de grafo da su origen en 1847 cuando Kirchhoff ocupaba una manera para poder resolver un sistema de ecuaciones lineales simultáneas que daban corriente en cada rama y alrededor de cada circuito de la red eléctrica. Por lo cual Kirchhoff desarrolló esta manera de conectar los nodos en forma ramificada.

Tiempo después en el año de 1857, Cayley descubrió la importancia de los grafos de tipo árbol al considerar los cambios de variables en el cálculo diferencial. Más tarde, también le dio uso para enumerar hidrocarburos como el compuesto CH_4 el cual está compuesto de una unión de 4 átomos de hidrogeno y uno de carbono y tiene topología tipo estrella (figura 5).

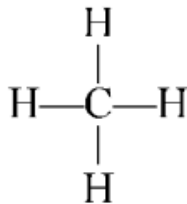


Figura 5: Molécula de metano.

Tomando en cuenta lo anterior y dada la naturaleza del problema a resolver, se usará esta topología de árbol, al ser un grafo no cíclico es ideal para el sistema de control que se utilizará para la sincronización, a través de un sistema maestro-esclavo donde un solo robot funciona como maestro de los demás (figura 6) [8].

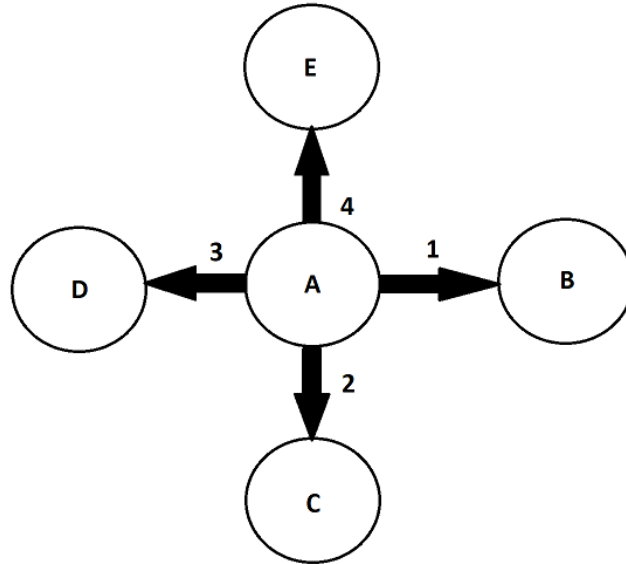


Figura 6: Topología de conexión de red tipo estrella o de árbol con 5 nodos (modo maestro/esclavo).

2.8. Matriz asociada a un grafo

Puesto que se puede analizar la estructura de un grafo en relación de las conexiones que tiene cada elemento en el sistema, es mejor analizar y trabajar con estas conexiones a través de su representación con una matriz.

Dado que un grafo se determina completamente especificando su estructura de adyacencia o su estructura de incidencia, estas especificaciones proporcionan formas mucho más eficientes de representar un grafo grande o complicado que una representación pictórica. Como las computadoras son más hábiles manipulando números que reconociendo imágenes, la práctica más utilizada es introducir a una computadora un grafo a través de su forma matricial. Algunas de las matrices utilizadas son:

La matriz de adyacencia $M(G)$, la cual es una matriz cuadrada de dimensión $N \times N$ (donde N es el número de nodos en la red) y sus elementos m_{ij} (donde m son los valores dados en cada espacio de la matriz donde i es el número de renglón y j es el número de columna), están dados por [6],[9]:

$$m_{ij} = \begin{cases} 1 & \text{si hay conexión entre los nodos } i \text{ y } j, \\ 0 & \text{para los demás casos.} \end{cases} \quad (1)$$

Para el sistema que se escogió (figura 6) la matriz de adyacencia es:

$$M = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (2)$$

La matriz de grado $D(G)$, es una matriz diagonal de dimensión $N \times N$ y sus elementos d_{ij} (donde d son los valores dados en cada espacio de la matriz donde i es el número de renglón y j es el número de columna), sus elementos se encuentran dados por [6]:

$$d_{ij} = \begin{cases} \sum_{k=1}^N m_{ik} & \text{si } i = j, \\ 0 & \text{para los demas casos.} \end{cases} \quad (3)$$

Para el sistema que se escogió (figura 6) la matriz de grado es:

$$D = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

La matriz laplaciana $L(G)$, es una matriz de dimensión $N \times N$ que está dada por la diferencia entre la matriz de grado $D(G)$ y la matriz de adyacencia $M(G)$, y sus elementos están dados por [6]:

$$d_{ij} = \begin{cases} d_{ij} & \text{si } i = j, \\ -1 & \text{si hay conexión entre } i \text{ y } j, \\ 0 & \text{para los demas casos.} \end{cases} \quad (5)$$

Para el sistema que se escogió (figura 6) la matriz laplaciana es:

$$L = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (6)$$

La matriz específica que se usa para poder llevar a cabo la sincronización es con la matriz de acoplamiento $A(G)$ se define como la matriz laplaciana negativa:

$$A(G) = -L(G) \quad (7)$$

Esta última es utilizada en las técnicas de control para analizar las redes y saber que elementos tienen comunicación con otras partes del sistema y así facilitar el trabajo de sincronizar redes con robots móviles.

2.9. Matriz de acoplamiento

La forma en la que se puede analizar un sistema de red con topología de conexión de tipo estrella es con una matriz de acoplamiento $A(G)$, la cual, se usa para identificar cuales nodos del sistema se encuentran conectados con otros.

Tomando de base el sistema formado en el ejemplo de la figura 6, se construyó la matriz de acoplamiento A , mostrada en la ecuación (8), la cual por medio de la ubicación de los valores numéricos nos dice que nodos y cuántos de estos están comunicados con otros, como el sistema consiste de 5 nodos o partes la matriz será de dimensiones 5x5.

En esta matriz las columnas y los renglones representan su respectivo nodo de modo que el nodo a_{11} hace referencia al valor de la primera columna y renglón, y como este nodo es el maestro, y está conectado a los otros 4 nodos del sistema tiene el valor de -4. Lo mismo sucede con los demás nodos, como estos se encuentran conectados solo a un nodo el cual es el maestro, su valor respectivo es -1 para las casillas a_{22} , a_{33} , a_{44} y a_{55} . Por la naturaleza de la matriz de árbol tipo estrella los valores ubicados en a_{12} , a_{13} , a_{14} , a_{15} , a_{21} , a_{31} , a_{41} y a_{51} , son todos el mismo debido a que existe solo una conexión en cada uno de los nodos 2, 3, 4 y 5 hacia el nodo 1.

$$A = \begin{bmatrix} -4 & 1 & 1 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & -1 \end{bmatrix}. \quad (8)$$

2.10. Conclusiones del capítulo

En este segundo capítulo de la tesis se habló acerca de los temas que están fuertemente relacionados con la creación del sistema de acoplamiento que se utilizará en esta tesis, al igual de explicar ciertos conceptos matemáticos y los métodos para trabajar y analizar los diferentes componentes en un sistema como lo es la matriz de acoplamiento.

Capítulo 3. Modelo matemático de movimiento del robot móvil tipo tracción diferencial

En este capítulo se presenta el modelo cinemático respecto al movimiento del robot de tracción diferencial junto con las pruebas de simulación del respectivo modelo con una trayectoria predefinida a seguir.

3.1. Modelo de control

El modelo a utilizar está basado en las fórmulas de control cinemático para un robot móvil de una sola llanta, la cual es la siguiente [10]:

$$\begin{aligned}\dot{x} &= v \cos \theta, \\ \dot{y} &= v \operatorname{sen} \theta, \\ \dot{\theta} &= \omega.\end{aligned}\tag{6}$$

Donde v y ω son las entradas del sistema, la cuales son respectivamente las velocidades lineales y angulares del robot, con este modelo se puede adaptar a un robot de tracción diferencial controlando estas velocidades de la siguiente forma [10]:

$$\begin{aligned}v &= \frac{vL + vR}{2}, \\ \omega &= \frac{vR - vL}{l}.\end{aligned}\tag{7}$$

Con esto se controla el movimiento del robot diferencial modificando las velocidades lineales y angulares a través de la separación de las llantas y la velocidad de giro de ambas ruedas por separado.

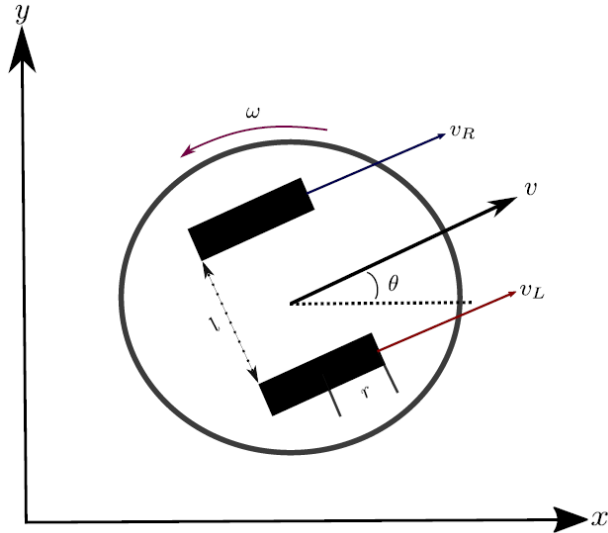


Figura 7: Plano de movimiento de un robot diferencial.

Con referencia a la figura 7, en la cual se representa al robot de tracción diferencial en un plano de con coordenadas x y y , en donde, la variable l representa la distancia entre ambas llantas, la variable r representa el radio de ambas llantas, v_R y v_L son la velocidad de giro de la rueda derecha y la rueda izquierda respectivamente, ω representa la velocidad de giro del robot y v es la velocidad vectorial del robot junto a una dirección, representada por el ángulo θ .

3.2. Simulación con destino fijo

Como la práctica de este robot se ve limitada a un trabajo en simulación en vez de una implementación en el mundo real, se tomaron ciertas libertades a la hora de la realización del modelo, puesto que este fue diseñado en base a:

- Se consideró una fricción nula que pudiera afectar al movimiento de las llantas del robot, haciendo que las llantas tuvieran un perfecto agarre sobre la superficie en la que se encontraba.
- La posición y la orientación angular inicial del robot son consideradas como un valor inicial por lo que se puede colocar en cualquier lugar del plano cartesiano.
- Como los resultados obtenidos son en entornos simulados, es posible asignar cualquier valor tanto para la velocidad como la aceleración del robot móvil, por lo que en algunos casos ocupa menos tiempo para realizar sus movimientos.

Para probar el modelo matemático del robot diferencial, primero se indujo movimiento desde un punto inicial determinado con coordenadas $(x_0, y_0) = (0,0)$ hasta un punto final $(20,30)$, en la superficie plana de la simulación. Para eso se asignaron los siguientes valores en la simulación de las coordenadas de destino:

$$\begin{aligned}x &= 20, \\y &= 30.\end{aligned}\tag{8}$$

En la gráfica de la figura 8 se muestra como el robot móvil toma la ruta más cercana entre ambas coordenadas la cual es una línea recta, por lo que se crea una pendiente ascendente como ruta.

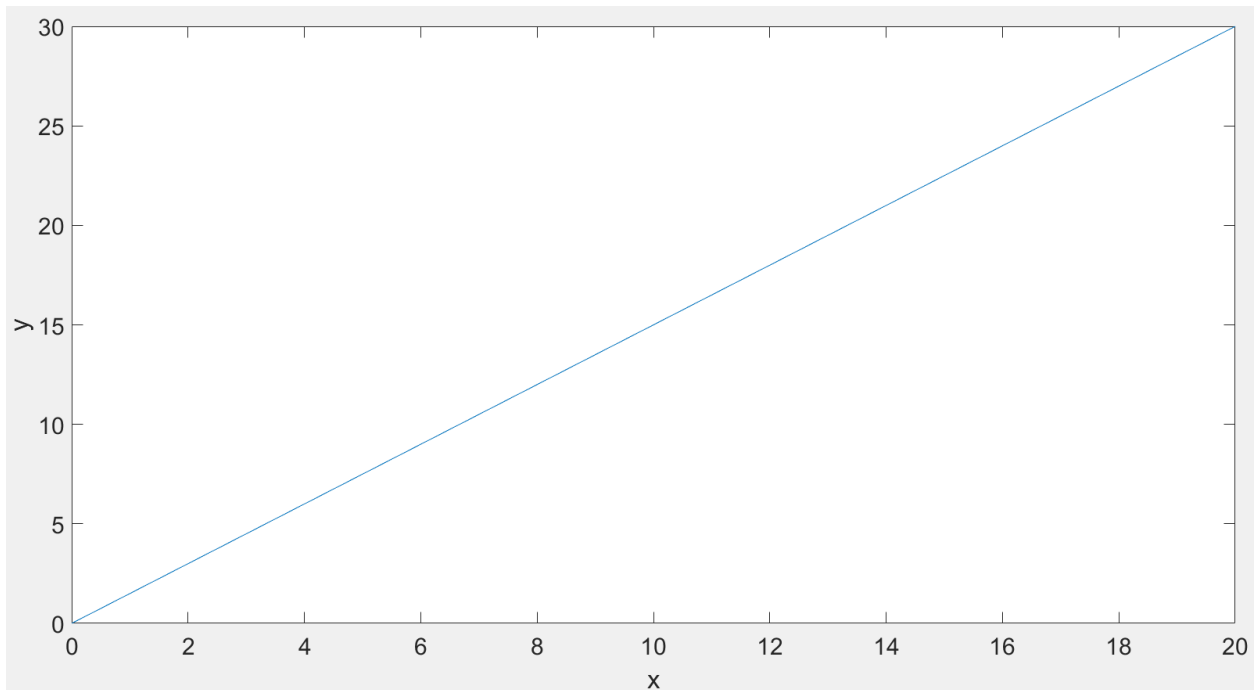


Figura 8: Trayectoria realizada en el plano por el robot diferencial para viajar a un solo punto (regulación).

La gráfica de la figura 9 presenta como es que el robot viaja de las coordenadas en las que inicia hasta el punto final, estas crecen de forma logarítmica debido a que en un inicio, el robot acelera rápidamente y comienza a desacelerar conforme se acerca a su objetivo, por lo que su movimiento es más lento al estar cerca del punto de llegada que al inicio.

Como se colocó el robot móvil con 0 grados como valor de θ , esto hizo que no se necesitara un movimiento rotacional tan pronunciado, por lo que en la gráfica se ve como solo hace un pequeño ajuste hacia la izquierda y después no hay cambio alguno debido a que la ruta que siguió para llegar al punto indicado era una línea recta.

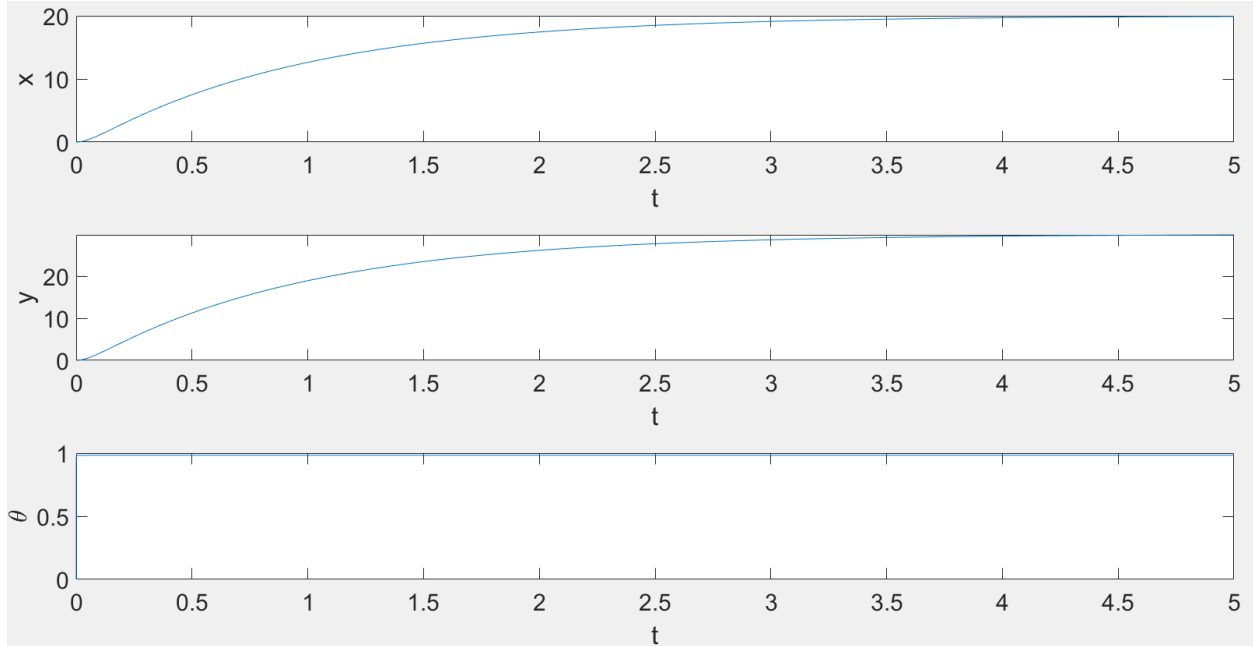


Figura 9: Gráfica de movimiento vectorial realizado por el robot diferencial al viajar a un solo punto.

3.3. Simulación con ruta fija

Una vez realizado la simulación del robot móvil hacia un punto en el plano en específico donde se comprobó que se alcanzaba la regulación de manera correcta, lo siguiente fue implementar el seguimiento de una trayectoria deseada o un patrón de ruta especificada a través de las ecuaciones paramétricas de una rosa polar, en la que se especifica la creación de 8 pétalos. Las ecuaciones paramétricas usadas para poder recrear esta figura son las siguientes:

$$\begin{aligned}
 x(t) &= 5 * \left(\cos \left(2 * \pi * \left(\frac{1}{T1} \right) * t \right) \right) * \left(\text{sen} \left(4 * 2 * \pi * \left(\frac{1}{T1} \right) * t \right) \right), \\
 y(t) &= 5 * \left(\text{sen} \left(2 * \pi * \left(\frac{1}{T2} \right) * t \right) \right) * \left(\text{sen} \left(4 * 2 * \pi * \left(\frac{1}{T2} \right) * t \right) \right).
 \end{aligned}
 \tag{9}$$

En donde se eligió el valor de 5 como ganancia al principio de ambas fórmulas, el cual actúa como un amplificador de amplitud ya que determina el espacio en el plano que abarcará la figura de la rosa polar, las variables $T1$ y $T2$ son las encargadas de definir la velocidad con la que giraran las ruedas al momento de seguir la ruta trazada por las ecuaciones paramétricas, esta se encuentra inversa y multiplicando a $2 * \pi$ debido a que la fórmula para la velocidad angular de rotación es $2 * \pi * (1/T)$, y por último el valor que define cuantos pétalos tendrá la rosa polar es el número 4, este determinará cuantos pétalos se tiene en cada lado del eje de simetría ubicado a la mitad de la figura. Como se colocó el número 4 se tendrán en total 8 pétalos, si se cambia este 4 a un 2 se tendría un total de 4 pétalos, etc.

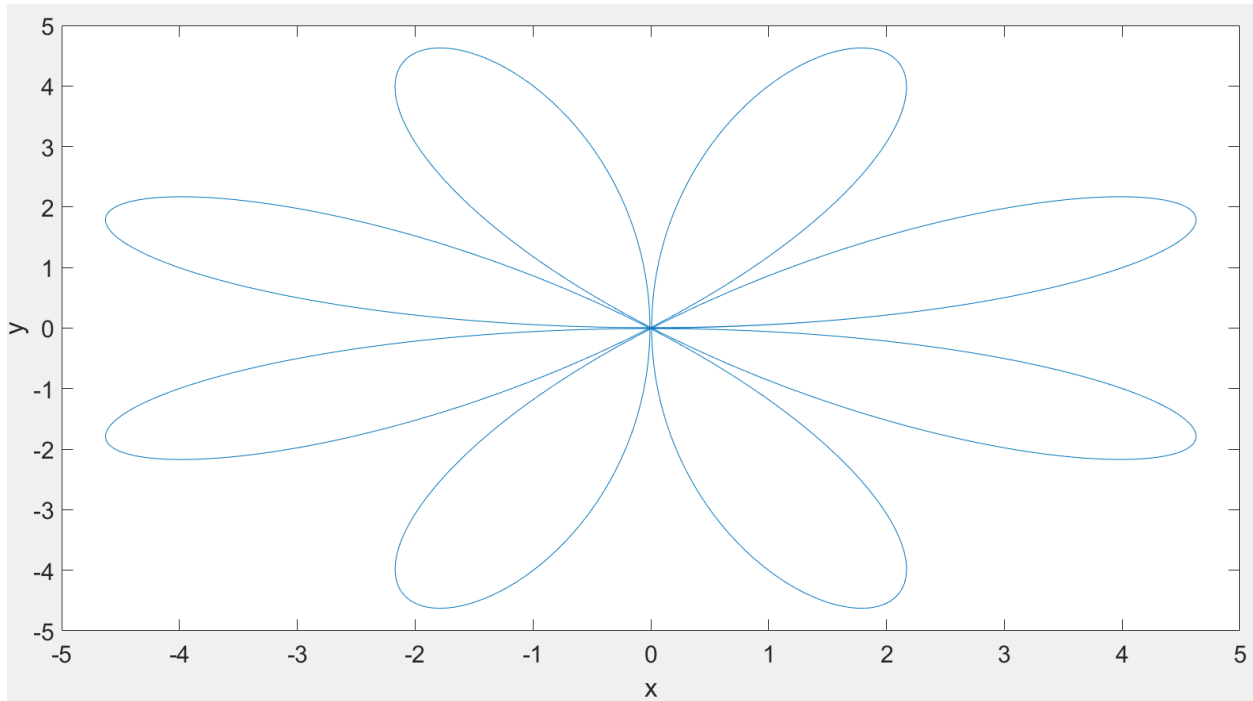


Figura 10: Trayectoria realizada en el plano de coordenadas (x,y) por el robot diferencial para hacer la figura de una rosa polar.

En comparación con el resultado presentado en la figura 8, en donde solo hacíamos viajar al robot en línea recta, en este caso se indujo un cambio de dirección en el robot, con el fin de poder realizar las curvas de los pétalos, dado que solo se tienen giros hacia la izquierda en cada pétalo, esto provoca que en la gráfica de θ de la figura 11, se tenga un aumento en los grados con sentido positivo, si existieran giros hacia la derecha el valor de θ se vería restado.

Los valores en los ejes x y y presentan una fluctuación entre 5 y -5 debido a que estos fueron los márgenes que se especificaron para el movimiento del robot móvil, la trayectoria generada usa como límite máximo el valor de 5 unidades hacia cada uno de los 4 ejes del plano cartesiano, por la complejión de la figura 10, se tiene simetría verticalmente y horizontalmente.

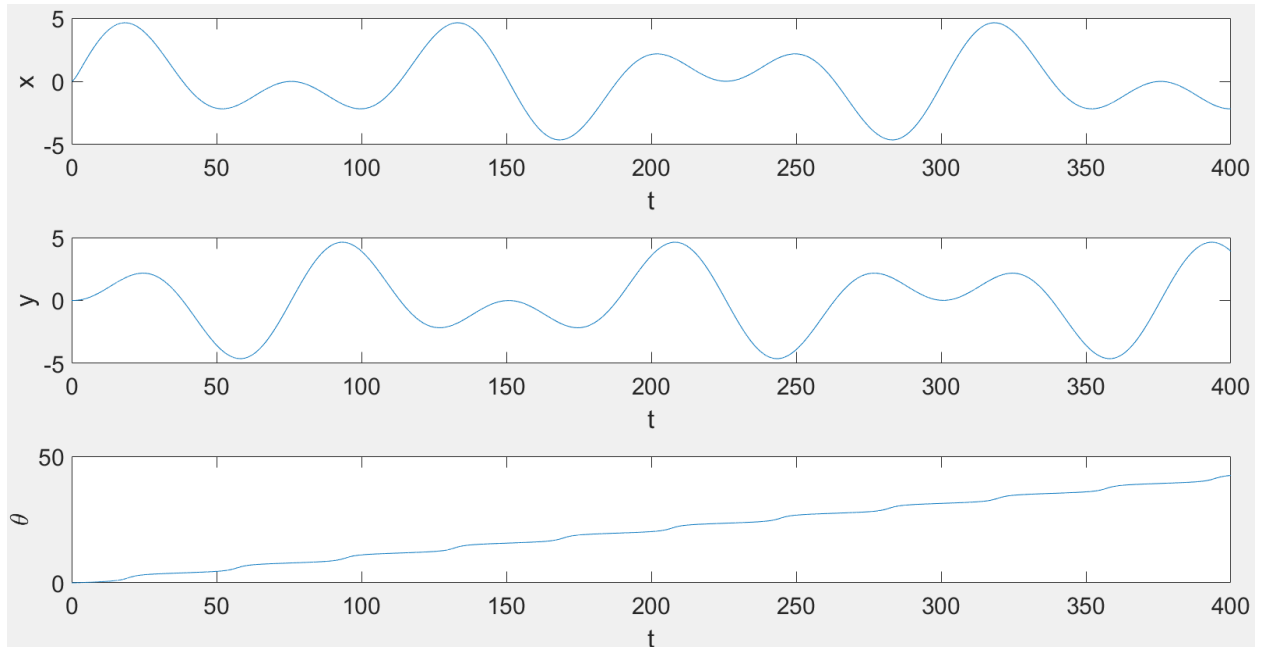


Figura 11: Gráfica de movimiento temporal realizado por el robot diferencial al realizar el seguimiento de la trayectoria de la rosa polar.

La gráfica de movimiento en las coordenadas x y y de la figura 11 nos presenta un patrón diferente respecto a la figura 8 debido a que la ruta tomada es una trayectoria deseada, la cual cambia constantemente en contraste con el caso de regulación que presenta una línea recta al punto deseado, curiosamente por la manera en la que se presentan los resultados de la simulación numérica de la figura 11 se puede ver un patrón con el movimiento de las gráficas de las coordenadas de movimiento, a simple vista, se puede ver como al ser una ruta de una figura simétrica los movimientos hechos en x y y son los mismos tan solo con un desfase en su onda, este tipo de rutas basadas con ecuaciones paramétricas son útiles cuando se tiene un plan de ruta específico planeado para un robot móvil, este puede aplicarse en trabajos de vigilancia o también para algún espectáculo de entretenimiento en donde se quiera trazar alguna figura en particular.

3.4. Conclusiones del capítulo

En este tercer capítulo de la tesis se presentó el modelo de control utilizado para gobernar el movimiento y dirección del robot con tracción diferencial, asimismo el robot fue puesto a prueba a través de dos simulaciones en Matlab en donde primero se usó el sistema simulando una ruta en línea recta, donde el robot solo giro al inicio para posicionarse antes de avanzar, y en la segunda simulación se probó con una ruta con curvas construidas con las ecuaciones de estado de una rosa polar donde se pudieron ver los cambios de dirección durante el movimiento del robot.

Capítulo 4. Sistema generador de posiciones aleatorias

En este capítulo se explicará el método utilizado para crear el sistema generador de posiciones aleatorias para la ruta de los robots junto con un sistema que evite que estos choquen al igual que sus respectivas simulaciones de prueba.

4.1. Sistema aleatorio con un robot diferencial

En esta simulación se colocó el punto de inicio del robot móvil en el origen y se generó una serie de puntos aleatorios a seguir dentro de un margen de 5 unidades espaciales (metros) en todas direcciones.

El método de la generación de rutas aleatorias que se usó en la simulación está ligada al tiempo de la misma, debido a que la cantidad de puntos aleatorios que se genera depende del tiempo que el robot tarde en moverse de la ubicación actual a la siguiente posición, por lo que la cantidad de puntos de ubicaciones nuevas dependen del plazo de tiempo de la simulación. Las ecuaciones que hicieron posible la generación aleatoria de puntos de ruta fueron las siguientes:

$$\begin{aligned}f1(n) &= a + (b - a) * rand(1,1), \\f2(n) &= a + (b - a) * rand(1,1).\end{aligned}\tag{10}$$

En donde $f1$ y $f2$ se encargan de generar ubicaciones en las coordenadas x y y respectivamente para generar los puntos de ruta aleatorios, las variables a y b se definen conforme al espacio requerido en el que se requiere generar el punto aleatorio, en este caso se requieren 5 unidades dentro del plano, por lo que los valores de a y b se eligieron de 5 y -5, respectivamente. Este sistema funciona gracias a un caso de condición la cual genera un nuevo punto de ruta aleatorio cada cierta unidad de tiempo, y deja de generar puntos cuando se acaba el tiempo de simulación, por lo cual la cantidad de rutas generadas está ligado al intervalo de tiempo.

La ruta generada por esta simulación se puede ver en la gráfica de la figura 12, en donde se puede notar todos los diferentes destinos que se generaron a lo largo de la ruta de movimiento, la programación se realizó con el propósito de que el robot móvil avance de forma recta hacia su destino, se puede ver que cuando el sistema genera un nuevo destino provoca que el robot cambie de dirección de forma abrupta.

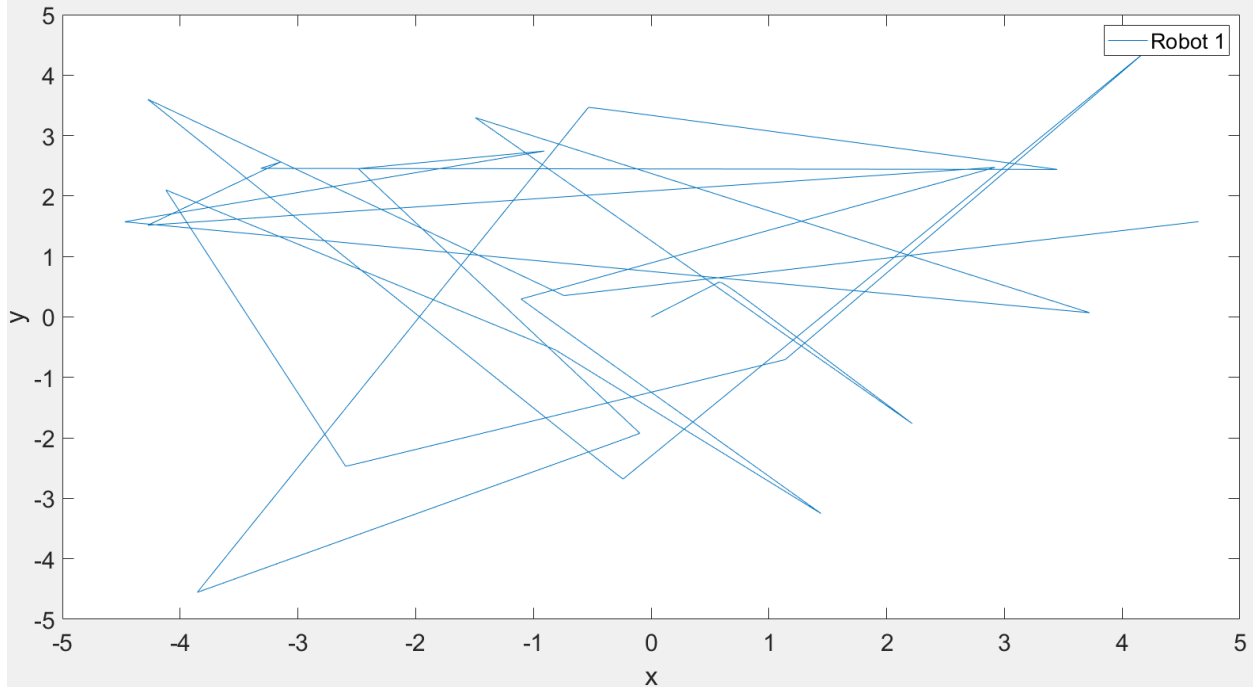


Figura 12: Trayectoria realizada en el plano de coordenadas (x,y) por el robot diferencial teniendo una serie de destinos diferentes.

Como los movimientos que realiza el robot diferencial para viajar entre cada punto es en línea recta, se presenta una misma situación en la primera simulación (figura 8), en donde antes de empezar a moverse de forma lineal el robot rota de manera angular de tal forma que quede directamente alineado a las coordenadas del próximo destino, por lo que se ve en la gráfica (figura 13) θ cambia de valor de forma abrupta solo cuando el robot está a punto de trasladarse de un punto a otro.

Asimismo, al igual que con la primera simulación de movimiento, en este caso se tiene que la ruta de simulación da como resultado trayectorias en forma de línea recta, las gráficas de las coordenadas x y y son exponenciales debido a que la manera en que el robot tipo tracción diferencial acelera es por medio de una retroalimentación de entradas basadas en la distancia restante para llegar al punto de destino, por lo que la velocidad disminuye al estar cercano a las coordenadas x y y marcadas, en ese sentido, a lo largo de la gráfica (figura 13) se ven “aletas de tiburón” tanto con valores positivos como negativos dependiendo en que parte del plano se encuentre el robot móvil.

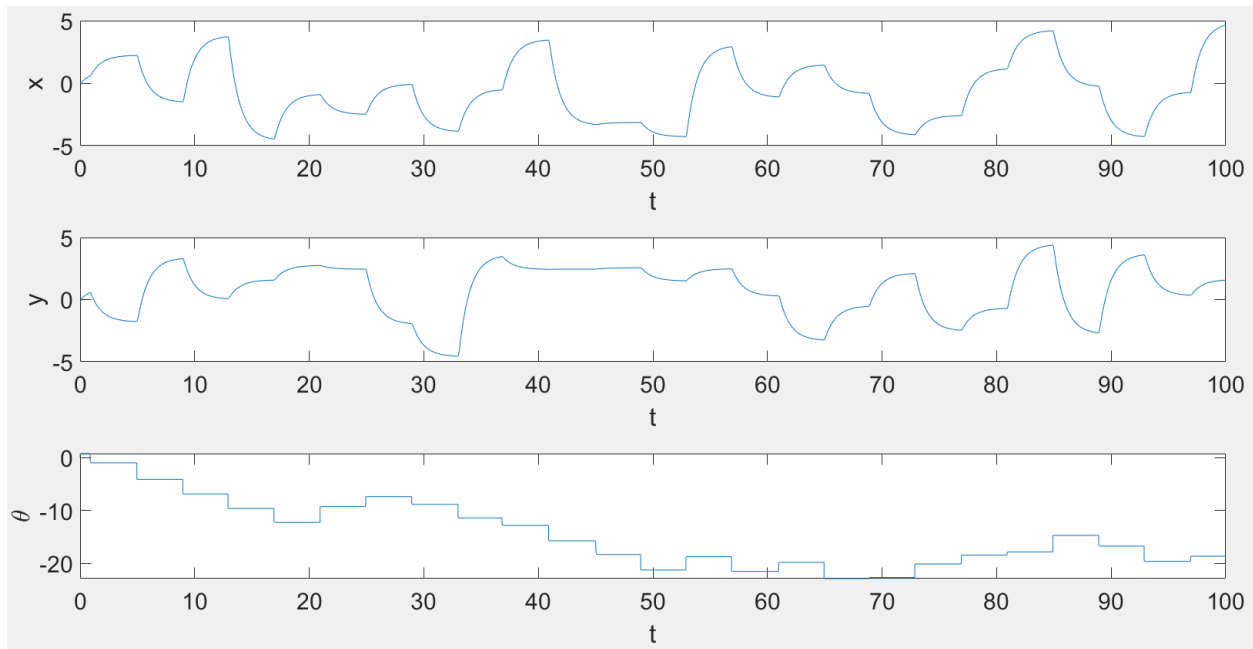


Figura 13: Gráfica de movimiento temporal realizado por el robot diferencial al realizar el seguimiento de la trayectoria de los diferentes puntos generados.

Este tipo de rutas son más adecuadas cuando se requiere programar una ruta impredecible dentro de un rango de área definido, si bien, esto también se puede lograr con ecuaciones paramétricas la ventaja es que al convertirlo en una generación aleatoria se ahorra procesamiento y deja a la computadora decidir los vectores a tomar, acercándose más a los movimiento erráticos que puede tener un insecto volador, como lo son las luciérnagas, en los cuales se basa la representación de esta simulación.

Sin embargo, al tener una secuencia de rutas aleatorias surge la problemática de que al tener una situación con dos o más robots dentro del mismo rango de movimiento se tiene el riesgo de que puedan chocar debido a la imprevisibilidad de sus rutas, lo cual aumenta exponencialmente entre más robots móviles se tengan, por lo cual se recurre a la implementación de un sistema anti colisiones, el cual se hará cargo de evadir a los demás robots móviles si es que estos se encuentran a determinada distancia entre ellos.

Por lo tanto, se creó la siguiente simulación en donde, se tienen una situación en la cual dos robots móviles se encuentra haciendo rutas generadas aleatoriamente con la misma velocidad dentro del mismo rango de espacio, donde se eligieron 10 unidades hacia las 4 direcciones.

4.2. Sistema anticollisiones con dos robots diferenciales

El sistema anticollisiones se creó a base de una condición en donde si el robot 2 se encuentra a cierta distancia (la cual es determinada dentro del mismo código) del robot 1 este lo evadirá cambiando su dirección a través de la generación aleatoria de un nuevo punto de ruta alejado de la trayectoria de este hasta que logre estar a la mínima distancia de acercamiento permitida. La fórmula utilizada para el sistema anticollisiones es la siguiente:

$$x = \sqrt{((f1(a) - f2(a))^2 + (f1(b) - f2(b))^2)} \leq y. \quad (11)$$

Teniendo en cuenta que x es una variable booleana que es falsa siempre y cuando la distancia vectorial de los dos robots sea menor al valor de distancia, lo que se obtiene restando la posición actual del robot 2 a la del robot 1 y elevándolo al cuadrado para no tener valores negativos, donde posteriormente se suman los resultados para después obtener la raíz cuadrada. Si esta condición se cumple x será 0 y el sistema anticollisiones no entrara en acción, pero si el resultado es mayor que y , x será 1 y entonces se procederá a generar una nueva ruta para evitar la colisión.

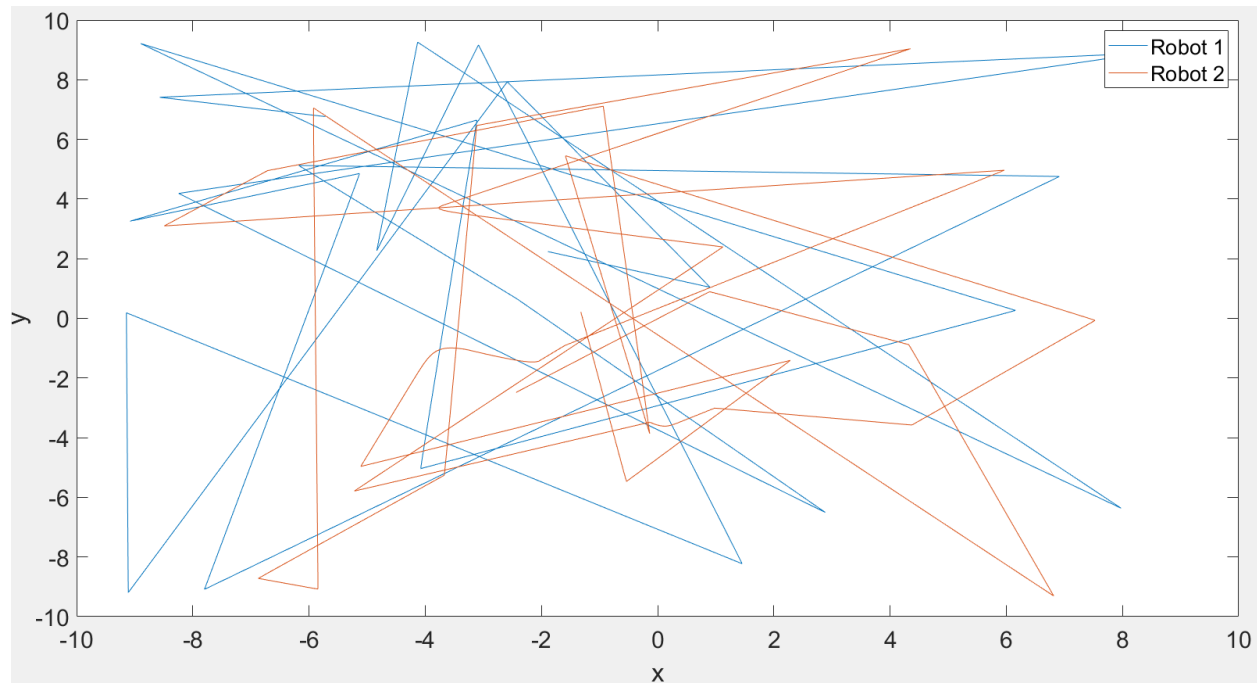


Figura 14: Trayectoria realizada en el plano de coordenadas (x,y) por dos robots diferenciales teniendo rutas aleatorias.

Al igual que con las simulaciones anteriores, se graficaron los movimientos que tuvieron ambos robots dentro del plano con respecto a las coordenadas x y y , lo que se destaca de esta gráfica comparativa de ambos robots móviles (figura 14) es que se ve claramente cómo es que el sistema anticollisiones entra en acción en la gráfica de θ (figura 15), en donde hay picos repentinos causados por la activación del sistema anticollisiones ya que para que el robot 2 se aleje del robot 1 este tuvo que cambiar la dirección de su trayectoria para que esta ya no interfiriera con la del robot 1. Este fenómeno se ve reflejado en un cambio de dirección abrupto por parte del robot 2, adoptando una dirección que lo alejará de la ruta del robot 1.

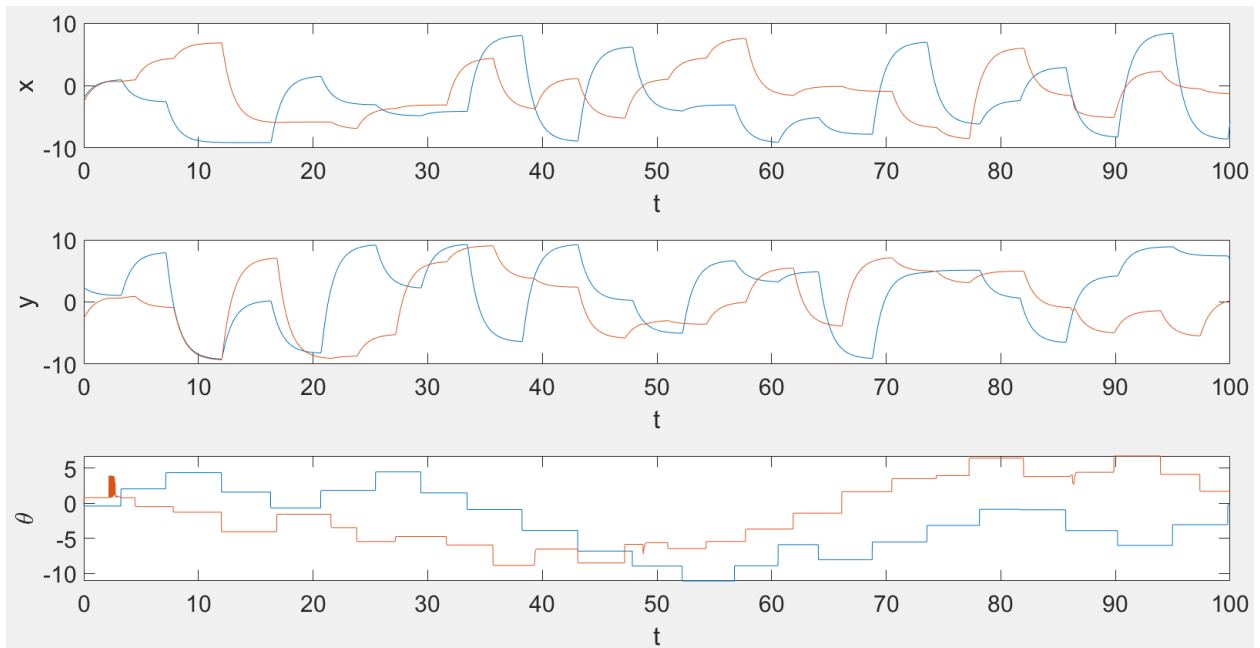


Figura 15: Gráfica de movimiento temporal realizado por los robots diferenciales.

En la gráfica de la figura 16 se muestra el error que hay entre ambas posiciones de los robots móviles en cada instancia de tiempo, donde se puede apreciar cómo es que al momento de que estos se encuentren a una distancia de error cercano a 0.1 metros se activa el sistema de anticollisiones evitando el acercamiento con el fin de evitar la colisión, por lo que cada vez que estos se acercan a la distancia especificada, se puede ver cómo los robots móviles se repelen.

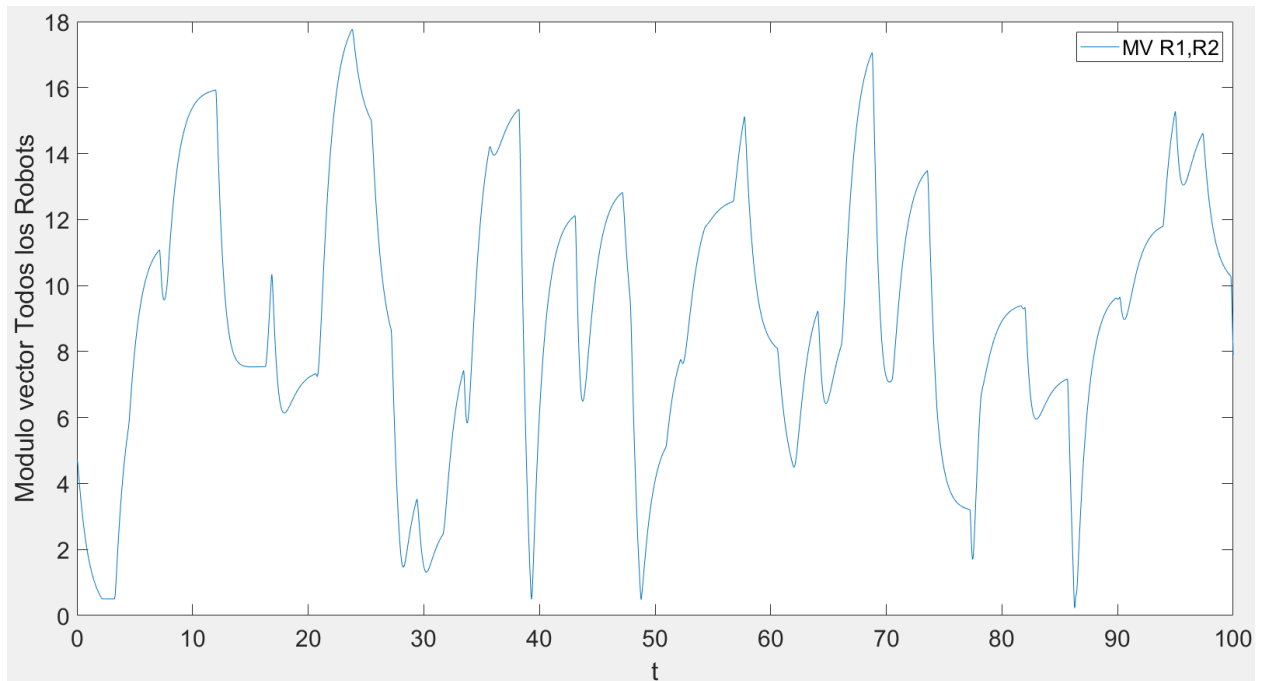


Figura 16: Gráfica del error (distancia) entre los dos robots diferenciales.

4.3. Conclusiones del capítulo

En este cuarto capítulo de la tesis se presentó e implementó el sistema generador de rutas aleatorias para el robot móvil en donde se cambió el tipo de ruta con respecto al capítulo anterior. Las rutas que recorría el robot ya estaban predefinidas y se generaron dentro del código antes de iniciar la simulación. Además, en este capítulo se realizó una segunda simulación considerando dos robots diferenciales en donde se empleó un sistema anticollisiones para evitar que las rutas generadas de forma aleatoria hicieran que los robots colisionaran.

Capítulo 5. Sistema oscilatorio para la sincronización espacialmente dependiente

En este capítulo se presenta y se analiza el modo de sincronización de los osciladores acoplados a los robots diferenciales. Además, se presentan simulaciones de las implementaciones realizadas.

5.1. Método de sincronización

Para poder imitar la sincronización que ocurre con las luces que emiten las luciérnagas se creó el siguiente modelo de onda sinusoidal con la que se ven comúnmente asociadas:

$$\text{Luz de luciérnaga} = A \text{ Sen}(x). \quad (12)$$

En este caso no es de principal interés la magnitud de las oscilaciones sino la frecuencia en la que se encienden y apagan, el parámetro de A de la fórmula (ecuación 12), la cual es la amplitud, se le puede asignar un valor de manera arbitraria, en este caso se le dio el valor de 1. El valor el cual se va a cambiar para que se diferencien las señales oscilatorias una de otra y sea posible ver el efecto de sincronización es la variable de frecuencia x .

Debido a que la manera en la que una luciérnaga emite sus luces es gradualmente, tanto cuando ganan como cuando pierde intensidad, es fácil asociar este comportamiento a una onda sinusoidal la cual va alternando constantemente al pasar del tiempo entre estos dos estados.

El método en que se logra la sincronización (ecuación 13) es similar a la del sistema anticollisiones (ecuación 11) por no decir que funciona bajo el mismo principio, ya que el acoplamiento entre las luciérnagas se activa junto con la activación del sistema anticollisiones haciendo que al medir la distancia que existe entre dos robots que interactúan entre sí, y si esta distancia es menor o igual a un límite preestablecido es cuando el sistema determina el acoplamiento para la sincronización de ambas señales de cada oscilador. Al igual que con la fórmula anticollisiones, esta también usa los valores de cada iteración del movimiento vectorial de ambos robots involucrados en la comparación, por lo que $f1$ es la posición del robot en una instancia de tiempo que se usará como referencia para calcular la distancia con $f2$ o robot 2, y será la distancia mínima antes de aplicar el acoplamiento entre dos osciladores tomando en cuenta a x como la variable indicadora encargada de decidir si los osciladores se acoplan o no.

$$x = \sqrt{((f1(a) - f2(a))^2 + (f1(b) - f2(b))^2)} \leq y. \quad (13)$$

A continuación, en las figuras 17, 18 y 19, se muestra de manera visual el proceso de la activación del acoplamiento para lograr la sincronización de las ondas oscilatorias de dos robots que entran en interacción siendo uno el maestro (robot 1) el cual tiene la oscilación de muestra (oscilador A) a ser replicado por el oscilador (oscilador b) del robot esclavo (robot 2) al momento en que la distancia (d) llega a un valor determinado. Esta representación visual es un ejemplo de activación de la sincronización cuando los dos robots se acercan a la distancia suficiente para que se active el sistema anticollisiones y a la vez inicie la sincronización.

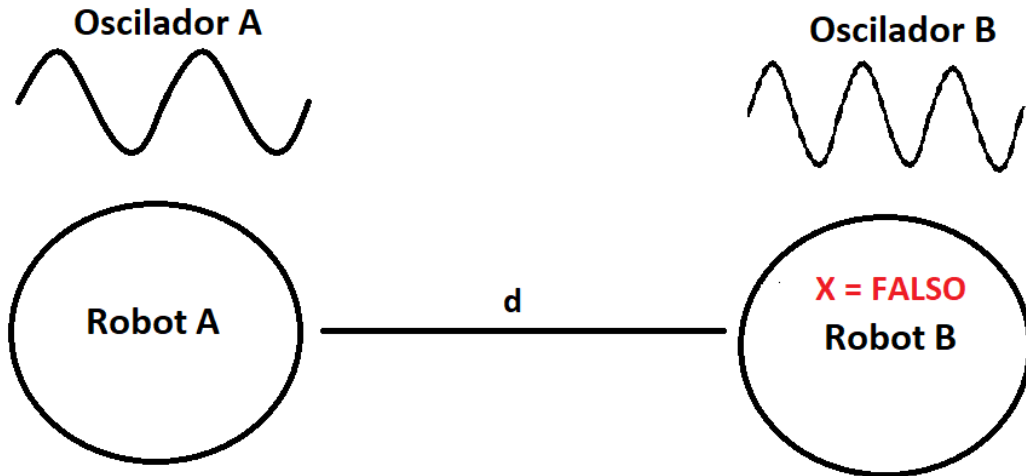


Figura 17: Dos robots acercándose pero lo suficientemente lejos para no activar el acoplamiento para la sincronización.

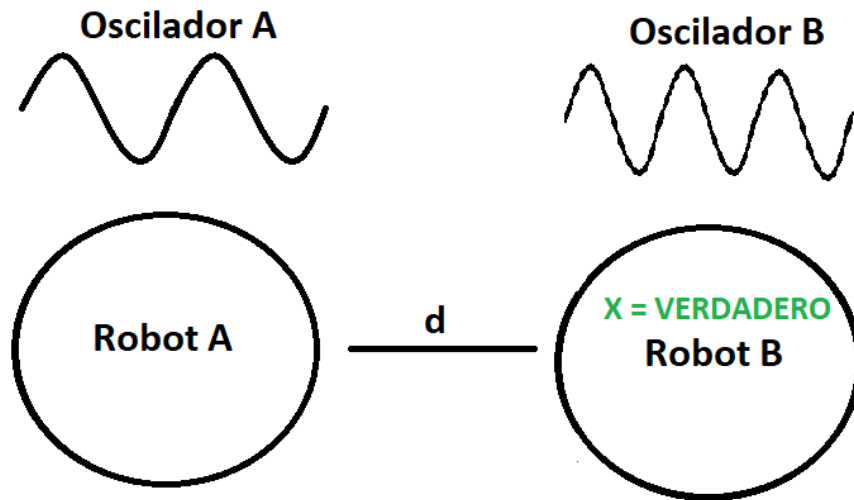


Figura 18: Dos robots acercándose y a la distancia indicada para aplicar el acoplamiento para la sincronización.

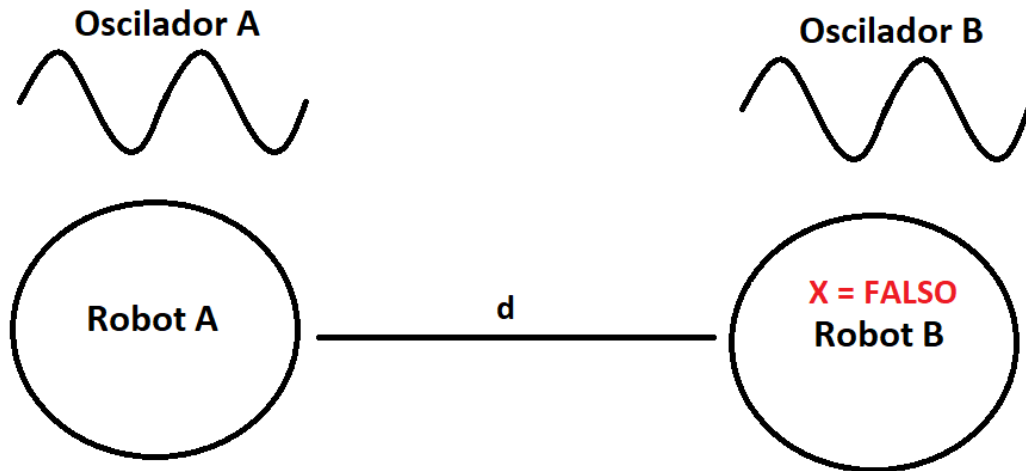


Figura 19: Dos robots alejándose a causa del anticollisiones pero ya sincronizados.

5.2. Simulación de la sincronización

Dado que el modelo de sincronización implementado (figuras 20 y 21) en la simulación funciona, se deduce que es debido a que en este modelo se usan las variables de la velocidad angular de ambas señales para generarlas en sus futuras iteraciones y a la vez las resta debido que entre más iteraciones ocurran menos será el error hasta alcanzar el estado de sincronización completa. La intensidad con la que se sincronizan puede ser cambiada a través de una constante de acoplamiento C , entre menor sea el valor de la constante de acoplamiento más se demora en alcanzar la sincronización de los osciladores.

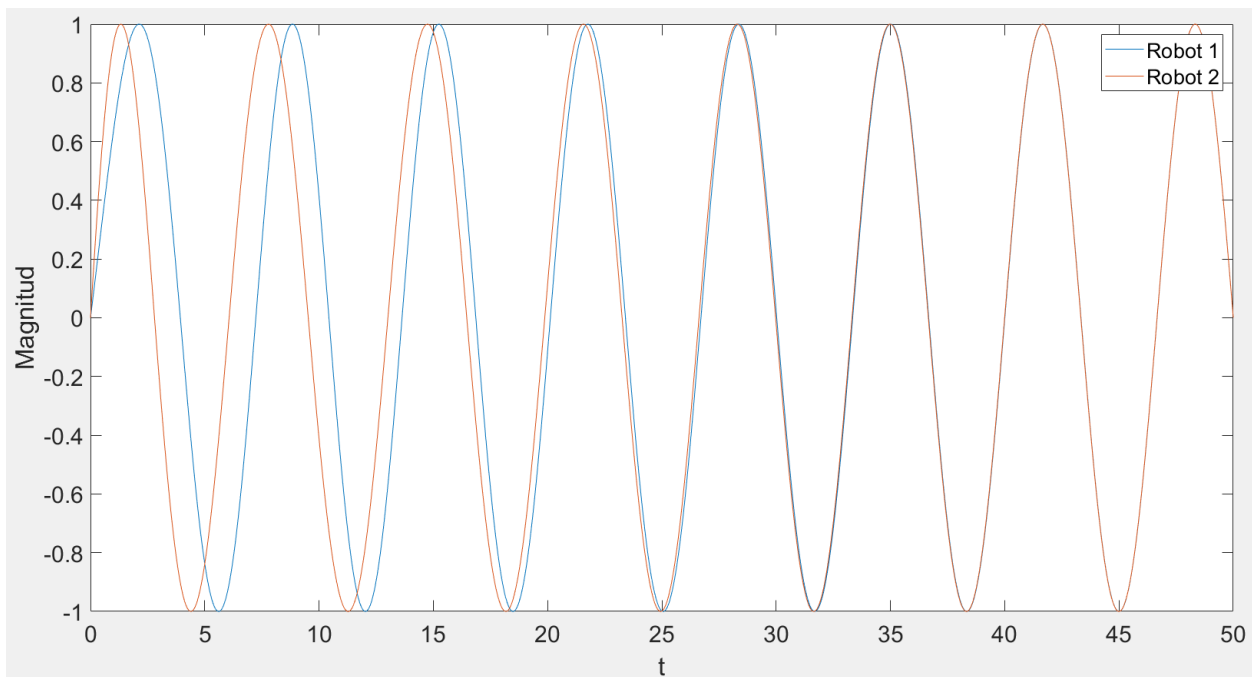


Figura 20: Sincronización de dos osciladores con menor intensidad de acoplamiento ($C=0.0001$).

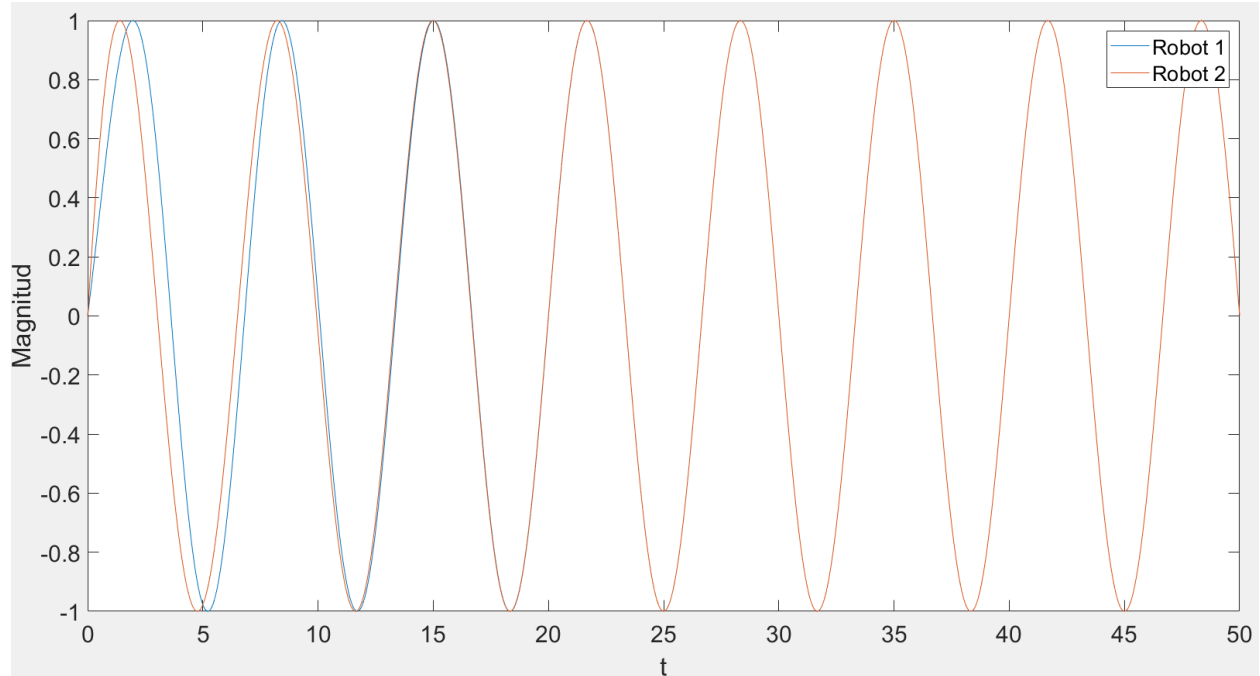


Figura 21: Sincronización de dos osciladores con mayor intensidad de acoplamiento ($C=0.0002$).

Ya que el modelo de acoplamiento usado se basa en la sincronización de tipo maestro esclavo, debido a que la onda sinusoidal del robot 2 es la que se sincroniza a la frecuencia del robot 1, esto hace alusión al entorno luciérnaga comentado al principio en donde como las luciérnagas imitan el patrón de emisión de luz de la luciérnaga que está teniendo más éxito atrayendo la atención de posibles parejas, por lo que naturalmente las demás lo imitarán.

5.3. Conclusiones del capítulo

En este quinto capítulo de la tesis se presenta el sistema oscilatorio que se usará en cada uno de los cinco robots diferenciales, y como es que funciona el sistema de sincronización de los osciladores al momento de que un robot móvil se encuentre a cierta distancia de otro, provocando la sincronización de sus osciladores.

Capítulo 6. Resultados finales del experimento

Con todo lo anteriormente explicado en los capítulos anteriores, ya se tiene una comprensión global para alcanzar el objetivo general de la tesis, por lo que a continuación se presenta la simulación final, en la cual se plasma el objetivo general de esta tesis, el cual, es la recreación de la sincronización de señales con robots diferenciales en un entorno luciérnaga.

6.1. Resultados finales del movimiento en el plano

Esta simulación final se hizo tomando en cuenta 5 robots móviles que realizaran el movimiento en un espacio de 40x40 unidades de medición (metros). Como los casos anteriores, estos robots inician en posiciones ligeramente diferentes con el fin de que la primera ruta generada de manera aleatoria no se vea influenciada por el sistema anticollisiones.

La siguiente figura (figura 22) refleja los resultados de las rutas que se pueden ver a simple vista. Los robots móviles están marcados cada uno con un color diferente para poder apreciar cada una de sus rutas. Como se pudo ver en la gráfica de movimiento en el plano de la anterior simulación (figura 14) presentando el sistema anticollisiones, y sabiendo que la generación aleatoria de rutas tiene la particularidad de que son solo rutas en línea recta, se puede ver las anormales rutas con trazados con curvas circulares, los cuales son provocados cuando los robots móviles se encuentran peligrosamente cerca de otros robots móviles, activándolo y por consiguiente generando una ruta alterna que evada la colisión.

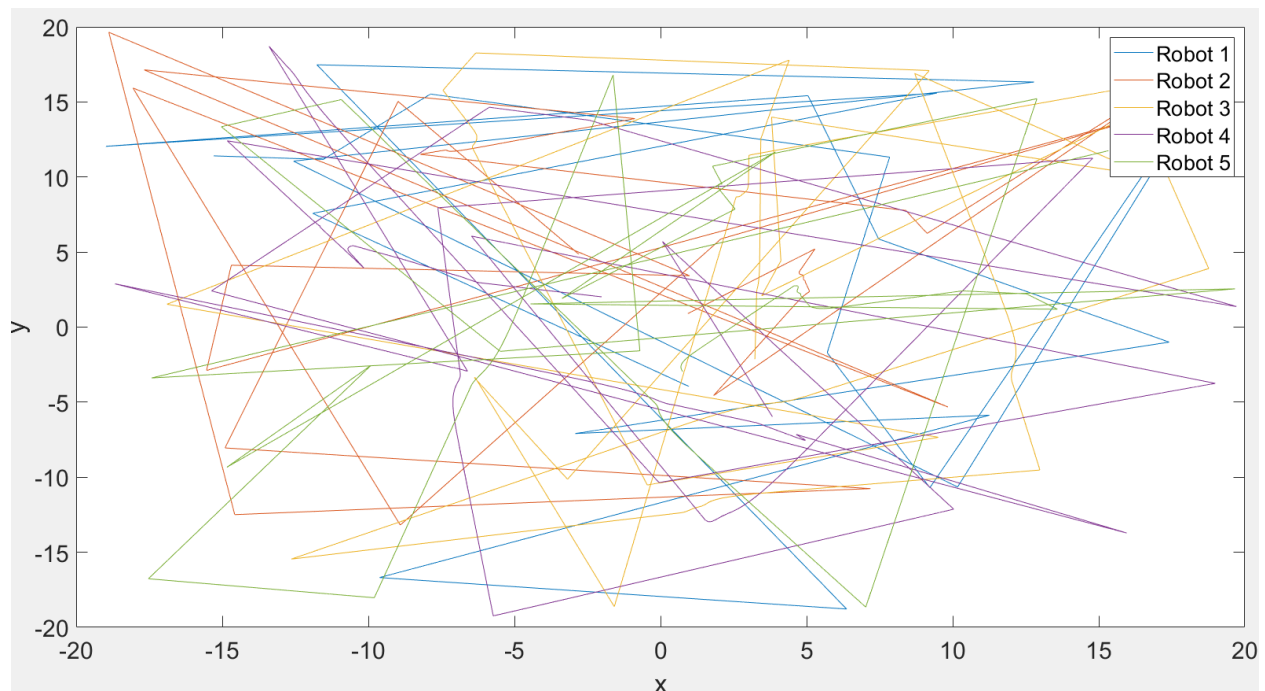


Figura 22: Trayectoria realizada por los 5 robots móviles sobre una superficie de 40x40 metros.

En la figura 23, se ven los movimientos que hizo cada robot móvil en el lapso total de 100 unidades de tiempo (segundos). En contraste con la figura 22 donde se realiza un mapeo de la ruta entera que hizo cada robot en un espacio de 40x40 unidades, en esta figura (figura 23) se puede ver el movimiento que efectuó cada uno de los robots tanto en el eje x como en el eje y de las coordenadas de la superficie.

Como se puede apreciar por la gráfica (figura 23), cada línea tiene un color diferente para apreciar a los distintos robots, donde al ver y analizar la gráfica a base de lo que ya sabemos acerca de la aceleración hecha por los robots al viajar de un punto a otro, se puede ver como ninguna aleta de tiburón o línea se encima con otra en el mismo valor de x en el plazo de tiempo, este mismo comportamiento se puede ver con el grafico de las coordenadas y .

Con respecto a la gráfica de la orientación, θ la cual nos muestra el giro de dirección de los robots en donde se puede lograr apreciar el momento en que el sistema evita que los robots colisionen en acción. Estos se ven en forma de pequeñas zonas en la gráfica en donde se pueden ver porciones un poco más gruesas debido al cambio de dirección al re ajustar un nuevo destino en donde no colisione con el robot actual. Pero como estas gráficas solo sirven para mostrar el movimiento que realizaron los robots móviles a lo largo de la simulación, es posible que no se perciba el efecto de la anticolisión, por lo cual las siguientes gráficas (figuras 24 y 25) muestran esos datos.

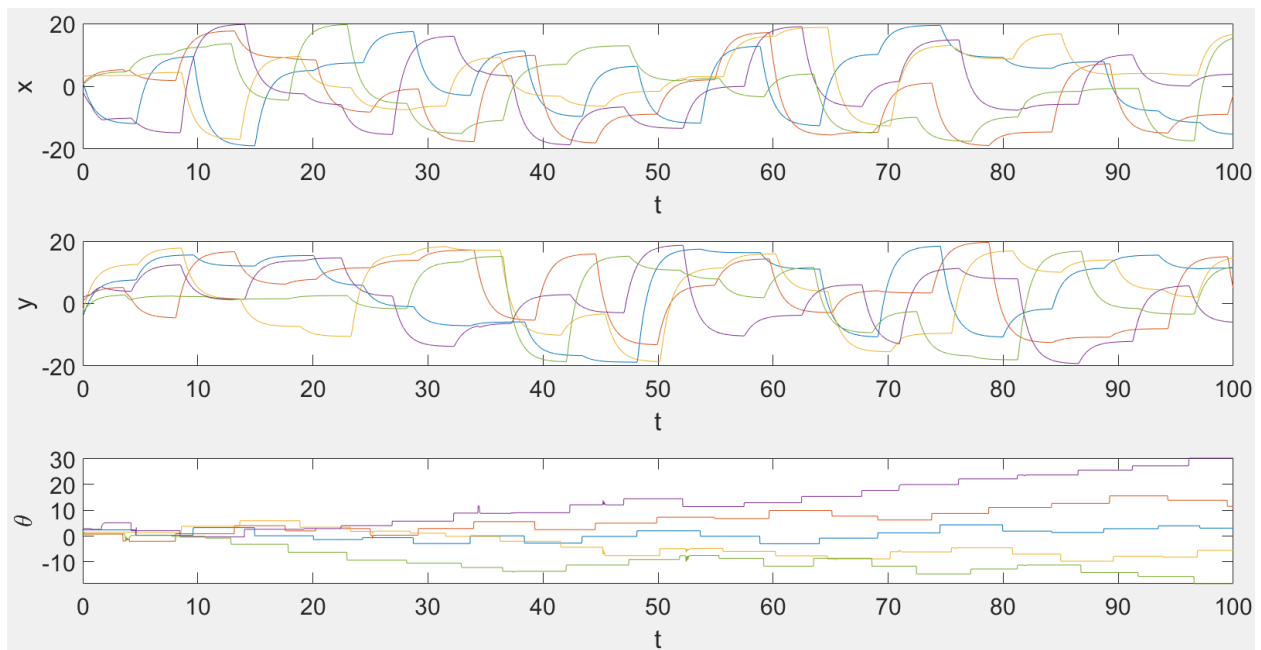


Figura 23: Gráficas de movimiento temporal de los 5 robots móviles.

En las gráficas de la figura 24 se muestra el error que tienen todas las relaciones posibles de los 4 robots móviles en relación al robot 1 en cuando a la distancia en la que se encuentran. Aquí se tiene el valor de la resta respecto a las distancias de sus posiciones en x y y que tienen respecto al robot 1. Se puede comparar con los valores obtenidos de las gráficas anteriores (figuras 22 y 23) en donde se muestran los cambios de posición de todos los robots móviles al transcurso de la simulación, con lo cual se comprueba la activación del sistema preventivo de colisiones cuando el error en x y el error en y es cercano a 0 al mismo tiempo.

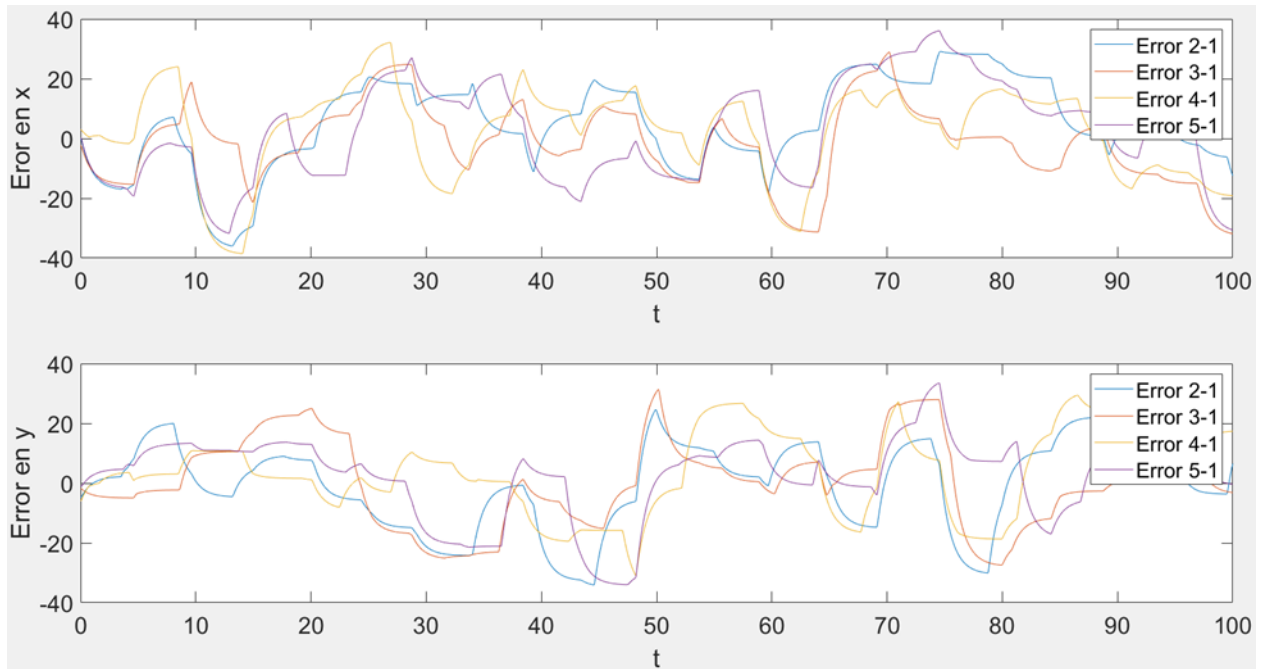


Figura 24: Gráficas del error (distancia) que tienen los robots esclavos (robots 2, 3, 4 y 5) contra el robot 1.

La gráfica de la figura 25 muestra una representación visual más sencilla de analizar con respecto a las anteriormente presentadas (ver figura 24). En cambio, de la figura 25 podemos destacar que se muestra de lleno el modulo vector el cual facilita la visualización de la distancia entre los robots de la red, es decir, con el modulo vector se puede apreciar como el valor de los errores nunca llegan a 0, ya que antes de que lleguen al valor 0.1 se activa el anti colisiones, haciendo que los robots implicados se alejen entre ellos para no chocar provocando que aumente la distancia entre los robots.

Además, en la gráfica de la figura 25 se pueden apreciar los 10 valores diferentes de errores que se pueden tener con todas las combinaciones únicas entre los 5 robots móviles.

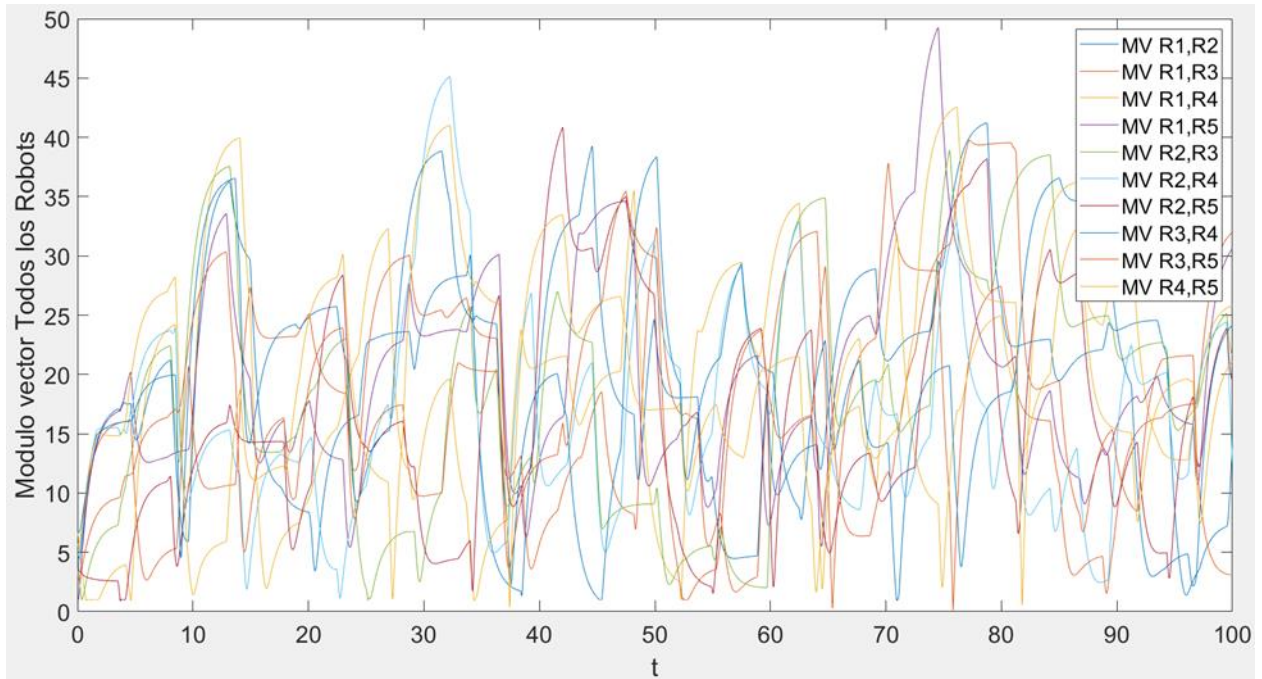


Figura 25: Gráfica del error (o distancia) entre todos los robots.

La figura 26 presenta los pulsos de activación del sistema anti-colisiones, en donde, cada vez que una línea se levanta de 0 a 1 indica la activación de este. En la parte inferior de la figura 26 se muestra una comparación con los escalones de activación de arriba de ella con el fin de apreciar cómo es que los errores cambiaron en determinados lugares del tiempo de la simulación.

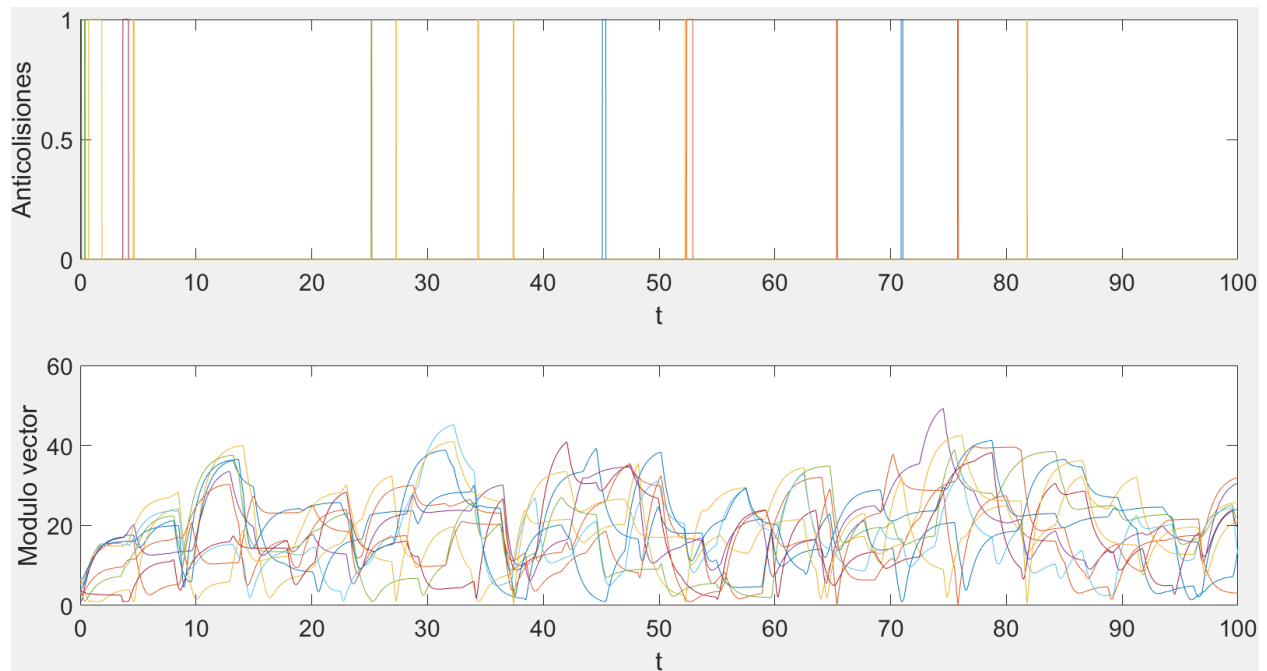


Figura 26: Comparación de activación del sistema anticolisiones con la distancia entre los robots.

6.2. Resultados finales de sincronización de ondas

Una vez realizado el análisis del impacto del sistema de anti colisiones con respecto a las rutas de los robots móviles obtenidos en los resultados de las simulaciones numéricas, ahora es turno de analizar la sincronización de las señales sinusoidales que cada uno de ellos tiene. Como fue explicado con anterioridad, la sincronización de las señales de cada robot móvil pretende imitar el parpadeo de la luz en una luciérnaga, la manera en la que se activa la sincronización de los osciladores de los robots móviles es cuando estos se encuentren a menos de 0.5 metros del mismo modo que el sistema anticollisiones es activado, solo que en comparación con el sistema anticollisiones la sincronización solo ocurre una vez si se trata de un mismo par de robots móviles.

Para la visualización de los resultados, se optó por mostrar en una gráfica cada estado de sincronizaciones que se pueden obtener con las combinaciones únicas entre los 5 robots, es decir, se obtuvieron 10 gráficas (figuras 27, 28, 29, 30 y 31), en donde se muestran dos osciladores con el número correspondiente a su respectivo robot móvil. Como cada oscilador se puede sincronizar con cualquier otro, se puede ver en distintas comparaciones de oscilaciones como alguna de las dos frecuencias correspondientes a los osciladores analizados parece estar cambiando de valor, aunque al final se alcanza la sincronización entre los osciladores.

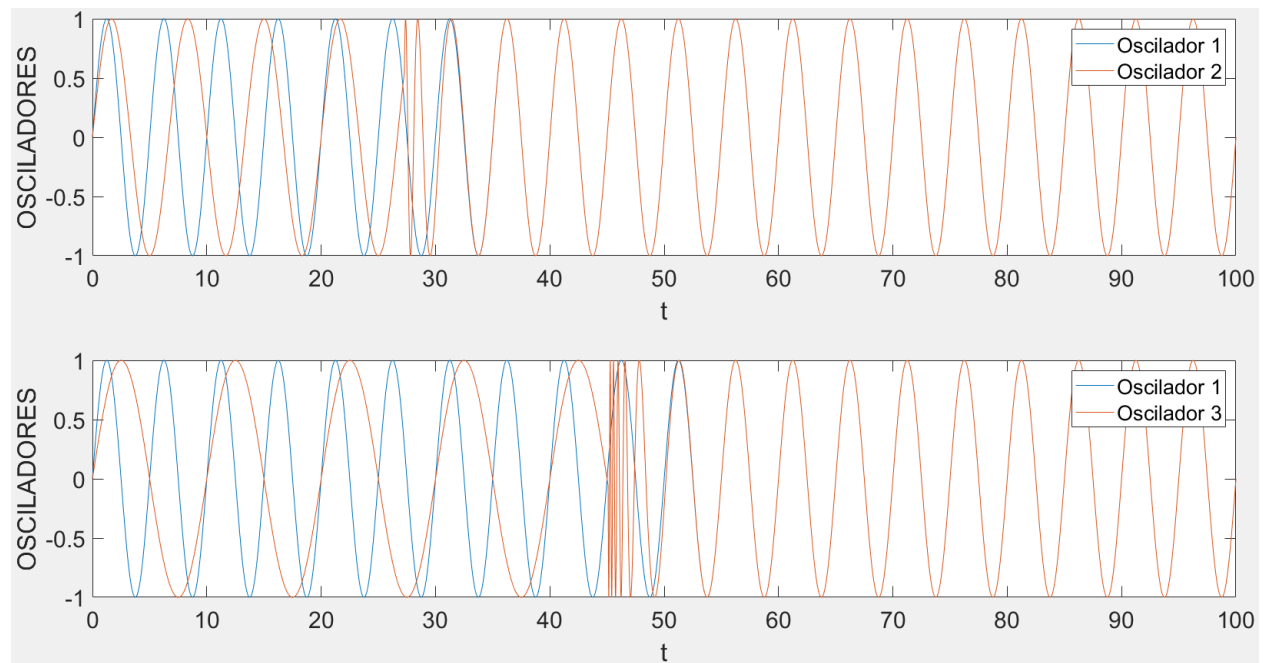


Figura 27: Gráficas de sincronización de 2 señales de los robots móviles 1-2 y 1-3.

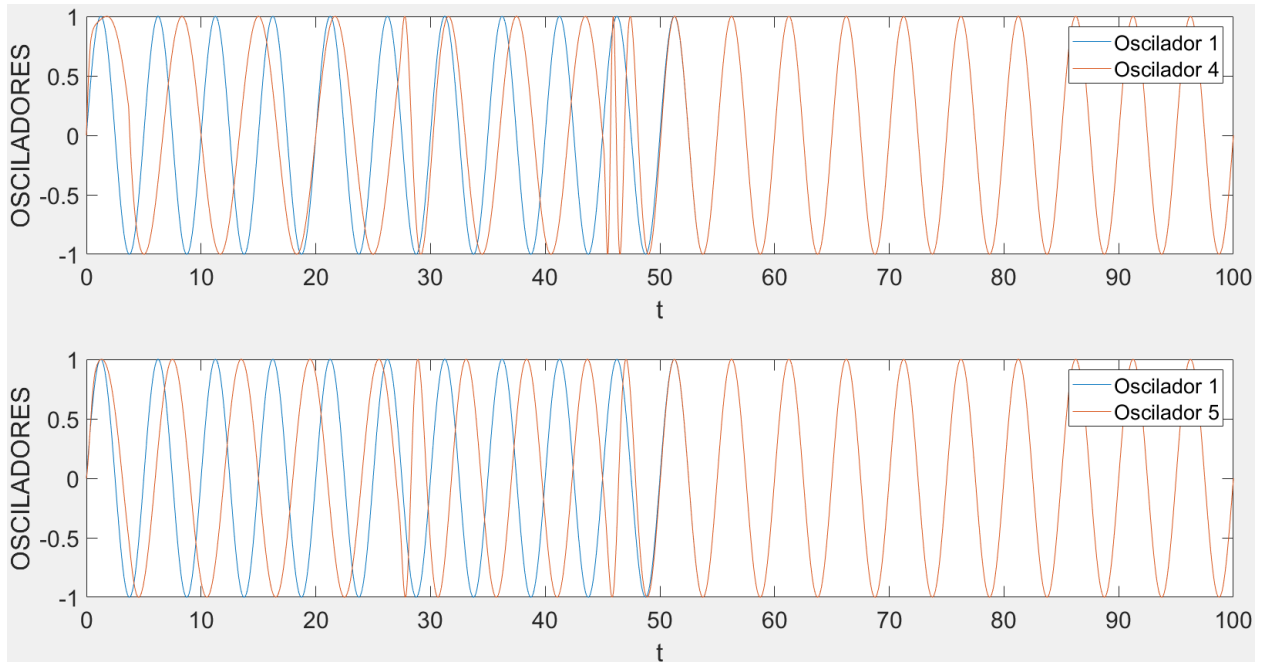


Figura 28: Gráficas de sincronización de 2 señales de los robots móviles 1-4 y 1-5.

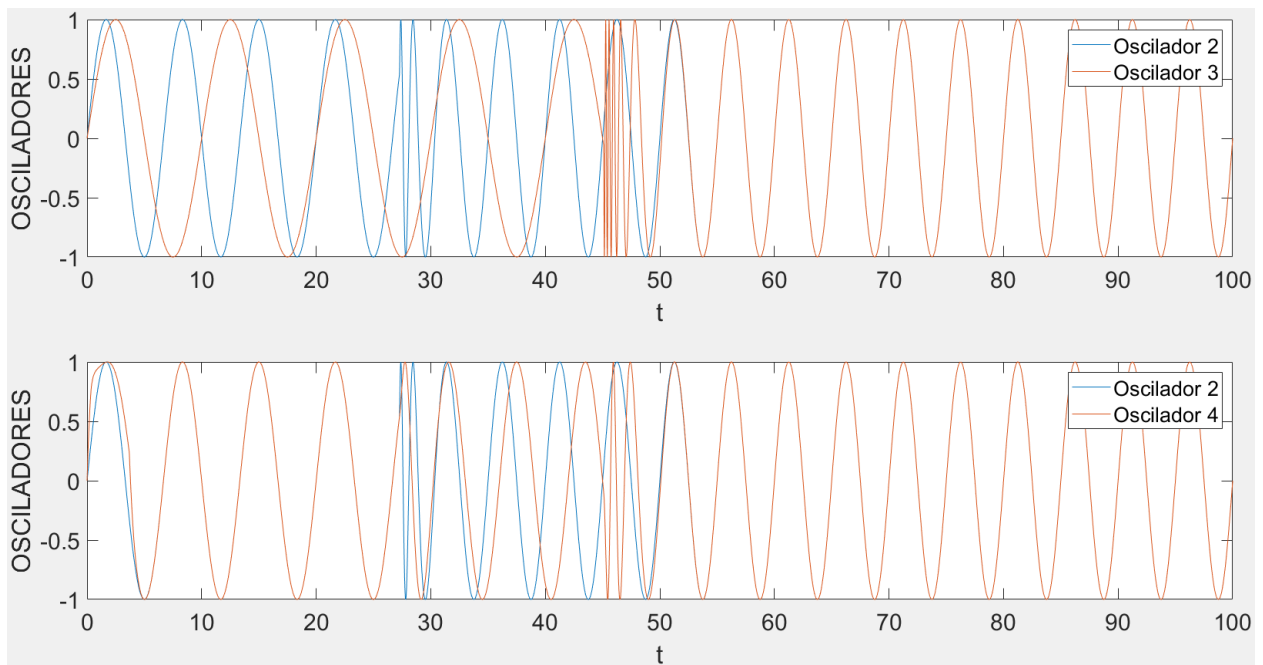


Figura 29: Gráficas de sincronización de 2 señales de los robots móviles 2-3 y 2-4.

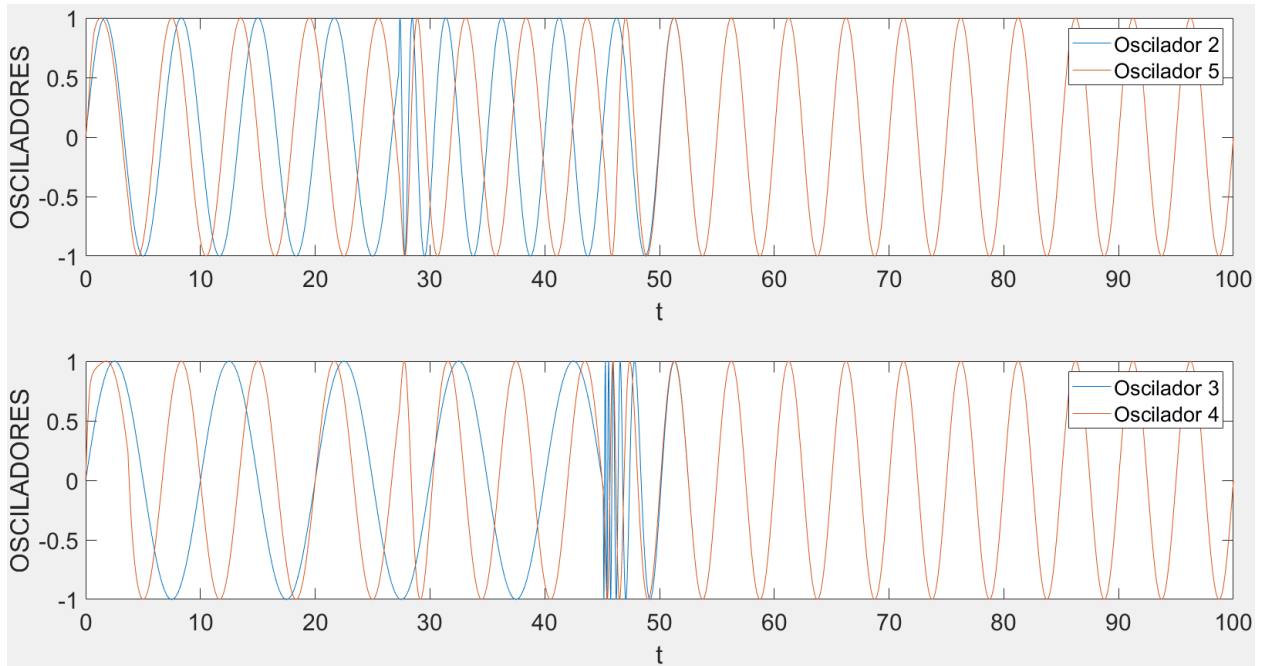


Figura 30: Gráficas de sincronización de 2 señales de los robots móviles 2-5 y 3-4.

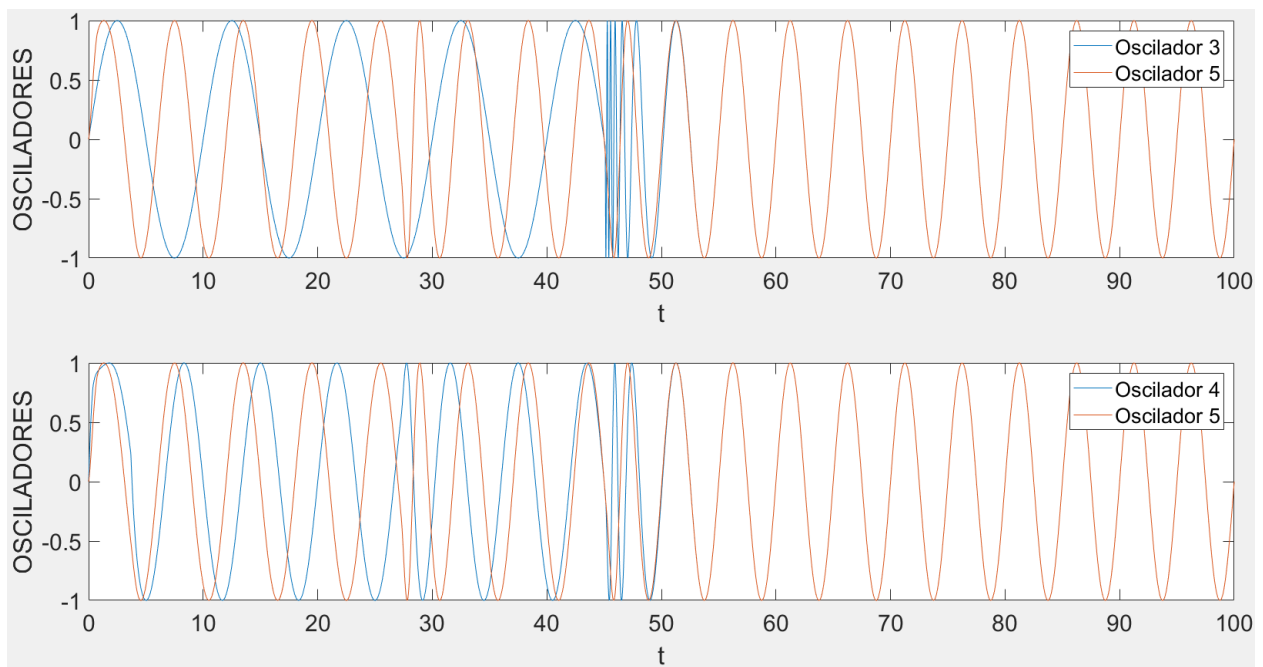


Figura 31: Gráficas de sincronización de 2 señales de los robots móviles 3-5 y 4-5.

En la gráfica de la figura 32, se muestra las activaciones del sistema de sincronización de los osciladores de los robots, cada posible sincronización es presentada con un color diferente, al ser variables que solo tienen dos estados (encendido/apagado) al no detectar otro robot lo suficientemente cerca para activar el anticollisiones el valor será 0 y no se iniciará la sincronización de las señales. En cambio, si los robots alcanzan una distancia menor al umbral establecido, la variable tomará el valor de 1 y se activará la sincronización. Se cuentan con 10 posibles casos de sincronización únicas con los 5 robots.

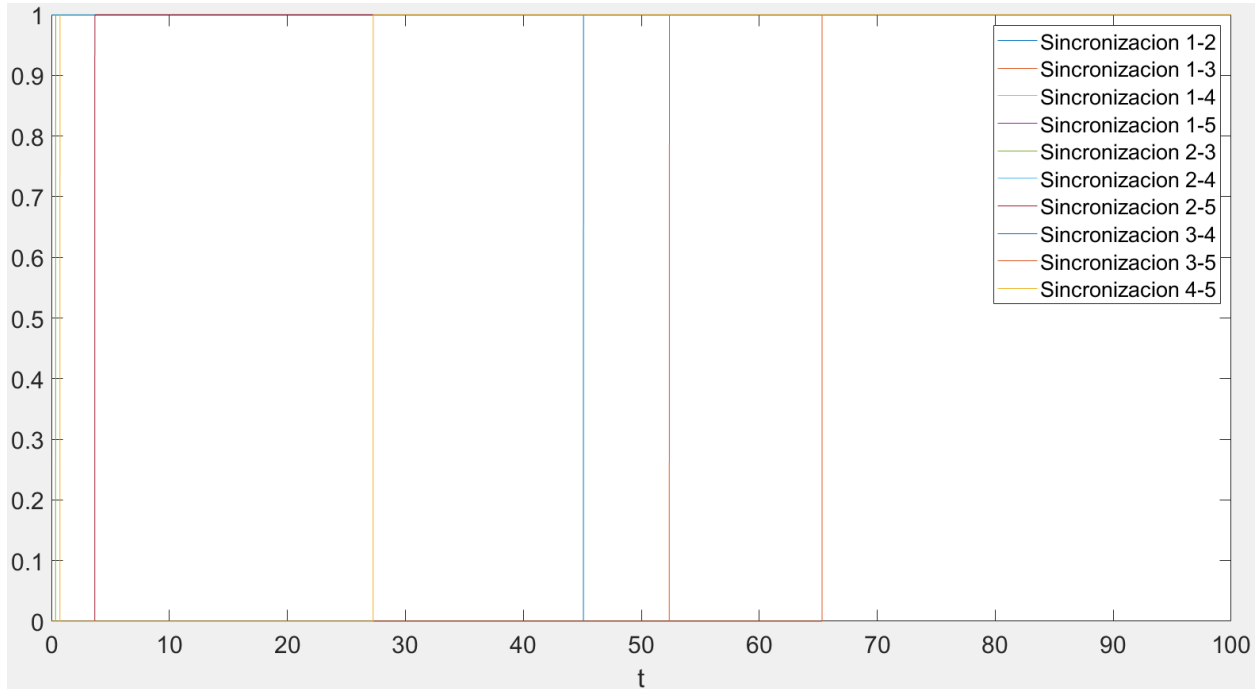


Figura 32: Gráfica de la señales de activación de todas las sincronizaciones de los osciladores.

Después de visualizar las señales de los 5 osciladores en pares para averiguar que osciladores se sincronizan con cuales, en la gráfica de la figura 33 podemos apreciar una comparación de los 5 osciladores durante todo el tiempo de la simulación. Analizando la información de los resultados obtenidos en estas últimas simulaciones numéricas se puede ver cómo es que las señales de los osciladores acoplados a los robots móviles se sincronizan a la frecuencia del oscilador 5, por lo cual se puede concluir que en esta simulación las luciérnagas 1, 2, 3 y 4 (robots esclavos) una vez juntas lo suficientemente cerca de la luciérnaga 5 (robot maestro), se sincronizaron a él.

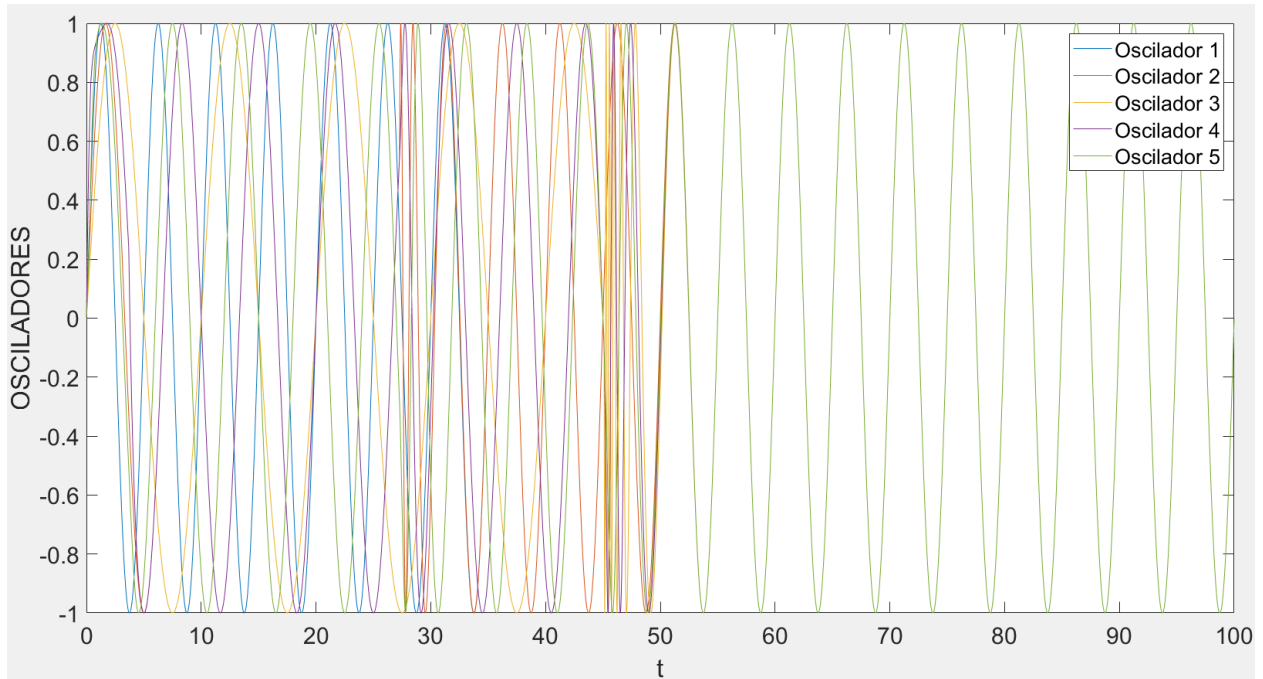


Figura 33: Gráfica de la sincronización de las 5 señales de los robots móviles.

6.3. Conclusiones del capítulo

En este sexto capítulo de la tesis se presentaron los resultados de la simulación final de los cinco robots móviles diferenciales en donde cada uno contaba con una ruta generada aleatoriamente y un oscilador acoplado a cada robot a diferente frecuencia, en los resultados obtenidos de las simulaciones se pudo apreciar la activación de los osciladores al igual que se pudo ver cuales osciladores se sincronizaron primero con gráficas comparativas.

Capítulo 7. Conclusiones

Conclusiones

Con la realización de este trabajo de tesis se pudo apreciar que las luciérnagas muestran un particular tipo de comportamientos a la hora de buscar pareja, y como estos se pueden reproducir a un conjunto de robots móviles con osciladores los cuales imitaban la función de la luz intermitente de las luciérnagas. Si bien, la señal utilizada para sincronizarse fue una sinusoidal sencilla, que es el tipo de señal continua que mejor imita los patrones de los destellos de una luciérnaga, esto no quiere decir que solo este tipo de señal puede ser empleada para el caso de sincronización de señales. Los osciladores empleados también pueden contener señales más complicadas como lo sería una señal caótica.

Además de la sincronización, otro aspecto importante que se logra con este tipo de comportamientos es que la sincronización depende de la distancia en que se encuentren las luciérnagas unas de otras, debido a que sus destellos solo se sincronizaban si estas están lo suficientemente cerca entre ellas imitando la frecuencia de los destellos de la luciérnaga macho que mejor estuviera acortejando con las féminas y en eso se basó la creación del sistema de sincronización junto con el sistema para evitar las colisiones entre los robots móviles puesto que a diferencia de las luciérnagas los robots no contaban con algún tipo de percepción integrada.

Posible mejoras y trabajo futuro

Cómo trabajo a futuro, se puede realizar una implementación física del sistema, en donde se pueda ver de manera práctica y no simulada el comportamiento de sincronización de los robots móviles imitando a las luciérnagas. También se puede ver la futura implementación de 10 robots móviles simultáneos en vez de 5, por cuestiones demostrativas en la simulación solo se hizo de 5 robots simultáneos.

Literatura citada

- [1] Ogata K. (2014). *System Dynamics*. (4^{ta} ed.). Pearson. 1-4.
- [2] Bossomaier T. & Green D. (2007). *Complex Systems*. Cambridge University Press. 251-253.
- [3] Khalil H. (1996). *Nonlinear systems*. (2^{da} ed.). Prentice Hall. 3 y 4.
- [4] Khalil H. (2002). *Nonlinear systems*. (3^{ra} ed.). Prentice Hall. 3 y 4.
- [5] Sayama H. (2015). *Introduction to the Modeling and Analysis of Complex Systems*. Open SUNY Textbooks. 3 y 4.
- [6] Dharwadker A. & Pirzada S. (2011). *Graph Theory*. Institute of mathematics. 1-3, 263, 288, 323.
- [7] Wah. C. (2007). *Synchronization in complex networks of nonlinear dynamical systems*. World Scientific Publishing. 5 y 6.
- [8] A. Arellano-Delgado, C. Cruz-Hernández, R. M. López Gutiérrez, C. Posadas-Castillo, "Outer Synchronization of Simple Firefly Discrete Models in Coupled Networks", *Mathematical Problems in Engineering*, vol. 2015, Article ID 895379, 14 pages, 2015. <https://doi.org/10.1155/2015/895379>.
- [9] López Parra A. (2017). *Formación en grupos de robots móviles*. Centro de Investigación Científica y de Educación Superior de Ensenada.
- [10] Suster P. & Jadlovska A. (2010). *Neural Tracking Trajectory of the Mobile Robot Khepera II in Internal Model Control Structure*. International Conference Process Control. 2 y 3.

Anexos

Código 1. Código de programación en Matlab para viajar a un punto fijo.

```
function continuodiferencial()
clear all;
clc;
close all;
tspan=[0 5];
Con0=[0 0 0 0.01]
options = odeset('RelTol',1e-6);
[t,Y]=ode45(@f,tspan,[Con0],[options]);

figure
subplot(3,1,1)
plot(t,Y(:,1))
xlabel('t','FontSize', 18)
ylabel('x','FontSize', 18)
set(gca, 'fontsize', 18)
subplot(3,1,2)
plot(t,Y(:,2))
xlabel('t','FontSize', 18)
ylabel('y','FontSize', 18)
set(gca, 'fontsize', 18)
subplot(3,1,3)
plot(t,Y(:,3))
xlabel('t','FontSize', 18)
ylabel('\theta','FontSize', 18)
set(gca, 'fontsize', 18)
figure
plot(Y(:,1),Y(:,2))
xlabel('x','FontSize', 18)
ylabel('y','FontSize', 18)
set(gca, 'fontsize', 18)
end
```

```

function [ecua]=f(t,x)
r=20.5/1000;
L=88.41/1000;
c11=20;
c21=20;
c12=20;
c22=20;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%punto
    xd=20;
    yd=30;
    xdp=0;
    ydp=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    xx1=x(4)*cos(x(3));
    yy1=x(4)*sin(x(3));
    u1=c11*(xd-x(1)) + c12*(xdp-xx1);
    u2=c21*(yd-x(2)) + c22*(ydp-yy1);
    W=((u2*cos(x(3)))-u1*sin(x(3)))/x(4);
    Wl = (x(4)/r) - ((L*W)/(2*r));
    Wr = (x(4)/r) + ((L*W)/(2*r));
    vL = r*Wl;
    vR = r*Wr;
    v=(vR+vL)/2;
    w=(vR-vL)/L;
ecua=[
    v*cos(x(3));
    v*sin(x(3));
    w;
    u1*cos(x(3))+u2*sin(x(3));
    ];
end

```

Código 2. Código de programación en Matlab para la ruta de rosa polar de ocho pétalos.

```
function continuodiferencial()
clear all;
clc;
close all;
tspan=[0 400];
Con0=[0 0 0 0.01]
options = odeset('RelTol',1e-6);
[t,Y]=ode45(@f,tspan,[Con0],[options]);
figure
subplot(3,1,1)
plot(t,Y(:,1))
xlabel('t','FontSize', 18)
ylabel('x','FontSize', 18)
set(gca, 'fontsize', 18)
subplot(3,1,2)
plot(t,Y(:,2))
xlabel('t','FontSize', 18)
ylabel('y','FontSize', 18)
set(gca, 'fontsize', 18)
subplot(3,1,3)
plot(t,Y(:,3))
xlabel('t','FontSize', 18)
ylabel('\theta','FontSize', 18)
set(gca, 'fontsize', 18)
figure
plot(Y(:,1),Y(:,2))
xlabel('x','FontSize', 18)
ylabel('y','FontSize', 18)
set(gca, 'fontsize', 18)
end
function [ecua]=f(t,x)
r=20/1000;
```

```

L=88.41/1000;
c11=20;
c21=20;
c12=20;
c22=20;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
T1=300;
T2=300;
%ocho petalos
    xd=5*(cos(2*pi*(1/T1)*t))*(sin(4*2*pi*(1/T1)*t));
    yd=5*(sin(2*pi*(1/T2)*t))*(sin(4*2*pi*(1/T2)*t));
    xdp=-(2*pi*(1/T1))*((sin(2*pi*(1/T1)*t)*sin(4*2*pi*(1/T1)*t))-
(4*cos(2*pi*(1/T1)*t)*cos(4*2*pi*(1/T1)*t)));
ydp=(2*pi*(1/T2))*((cos(2*pi*(1/T2)*t))*sin(4*(2*pi*(1/T2)*t)))+(4*(sin(2*pi
*(1/T2)*t))*cos(4*(2*pi*(1/T2)*t)));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    xx1=x(4)*cos(x(3));
    yy1=x(4)*sin(x(3));
    u1=c11*(xd-x(1)) + c12*(xdp-xx1);
    u2=c21*(yd-x(2)) + c22*(ydp-yy1);
    W=(u2*cos(x(3))-u1*sin(x(3)))/x(4);
    Wl = (x(4)/r) - ((L*W)/(2*r));
    Wr = (x(4)/r) + ((L*W)/(2*r));
    vL = r*Wl;
    vR = r*Wr;
    v=(vR+vL)/2;
    w=(vR-vL)/L;
ecua=[
    v*cos(x(3));
    v*sin(x(3));
    w;
    u1*cos(x(3))+u2*sin(x(3));
];
end

```

Código 3. Código de programación en Matlab para la ruta con destinos generados aleatoriamente.

```
function continuodiferencial()
clear all;
clc;
close all;
global n
global xx
global yy
global zz
global f1
global f2
n=1;
tspan=[0 100];
Con0=[0 0 0 0.01]
options = odeset('RelTol',1e-6);
[t,Y]=ode45(@f,tspan,[Con0],[options]);
figure
subplot(3,1,1)
plot(t,Y(:,1))
xlabel('t','FontSize', 18)
ylabel('x','FontSize', 18)
set(gca, 'fontsize', 18)
subplot(3,1,2)
plot(t,Y(:,2))
xlabel('t','FontSize', 18)
ylabel('y','FontSize', 18)
set(gca, 'fontsize', 18)
subplot(3,1,3)
plot(t,Y(:,3))
xlabel('t','FontSize', 18)
ylabel('\theta','FontSize', 18)
set(gca, 'fontsize', 18)
```

```

figure
plot(Y(:,1),Y(:,2))
xlabel('x','FontSize', 18)
ylabel('y','FontSize', 18)
legend('Robot 1')
set(gca, 'fontsize', 18)
end
function [ecua]=f(t,x)
global n
global xx
global yy
global zz
global f1
global f2
r=20.5/1000;
L=88.41/1000;
c11=20;
c21=20;
c12=20;
c22=20;
f1(1)=1;
f2(1)=1;
a=5;
b=-5;
if n<t
n=n+4
f1(n) = a + (b-a).*rand(1,1);
f2(n) = a + (b-a).*rand(1,1);
else
end
xd=f1(n);
yd=f2(n);
xdp=0;

```

```

        ydp=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%robot 1
        xx1=x(4)*cos(x(3));
        yy1=x(4)*sin(x(3));
        u1=c11*(xd-x(1)) + c12*(xdp-xx1);
        u2=c21*(yd-x(2)) + c22*(ydp-yy1);
        W=((u2*cos(x(3)))-u1*sin(x(3)))/x(4);
        Wl = (x(4)/r) - ((L*W)/(2*r));
        Wr = (x(4)/r) + ((L*W)/(2*r));
        vL = r*Wl;
        vR = r*Wr;
        v=(vR+vL)/2;
        w=(vR-vL)/L;

        ecua=[
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%robot 1
            v*cos(x(3));           %x(1)
            v*sin(x(3));           %x(2)
            w;                       %x(3)
            u1*cos(x(3))+u2*sin(x(3)); %x(4)
        ];
end

```

Código 4. Código de programación en Matlab para la ruta con destinos generados aleatoriamente con dos robots.

```
function continuodiferencial()
clear all;
clc;
close all;
global n
global n2
global xx
global yy
global zz
global f1
global f2
global f3
global f4
n=1;
n2=1;
tspan=[0 100];
N=2;%numero de robots
c=3; %numeros aleatorios entre c y d
d=-3;%numeros aleatorios entre c y d
%%%%%%condiciones iniciales aleatorias entre c y d
x0=c + (d-c).*rand(1,N);
y0=c + (d-c).*rand(1,N);
Con0=[x0(1) y0(1) 0 0.01 x0(2) y0(2) 0 0.01]
[t,Y]=ode45(@f,tspan,[Con0]);
figure
subplot(3,1,1)
plot(t,Y(:,1),t,Y(:,5))
xlabel('t','FontSize', 18)
ylabel('x','FontSize', 18)
set(gca, 'fontsize', 18)
subplot(3,1,2)
plot(t,Y(:,2),t,Y(:,6))
```

```

xlabel('t','FontSize', 18)
ylabel('y','FontSize', 18)
set(gca, 'fontsize', 18)
    subplot(3,1,3)
plot(t,Y(:,3),t,Y(:,7))
xlabel('t','FontSize', 18)
ylabel('\theta','FontSize', 18)
set(gca, 'fontsize', 18)
figure
plot(Y(:,1),Y(:,2),Y(:,5),Y(:,6))
xlabel('x','FontSize', 18)
ylabel('y','FontSize', 18)
legend('Robot 1','Robot 2')
set(gca, 'fontsize', 18)
    figure
plot(t,sqrt((Y(:,1)-Y(:,5)).^2 + (Y(:,2)-Y(:,6)).^2))
xlabel('t','FontSize', 18)
ylabel('Modulo vector Todos los Robots','FontSize', 18)
legend('MV R1,R2')
set(gca, 'fontsize', 18)
end
function [ecua]=f(t,x)
global n
global n2
global xx
global yy
global zz
global f1
global f2
global f3
global f4
r=20.5/1000;
L=88.41/1000;

```

```

c11=20;
c21=20;
c12=20;
c22=20;
f1(1)=1;
f2(1)=1;
a=10;
b=-10;
if abs(f1(n)-x(1))<.1 && abs(f2(n)-x(2))<.1
    n=n+1
    f1(n) = a + (b-a).*rand(1,1);
    f2(n) = a + (b-a).*rand(1,1);
else
end
    xd=f1(n);
    yd=f2(n);
    xdp=0;
    ydp=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f3(1)=1;
f4(1)=1;
if abs(f3(n2)-x(5))<.1 && abs(f4(n2)-x(6))<.1
    n2=n2+1
    f3(n2) = a + (b-a).*rand(1,1);
    f4(n2) = a + (b-a).*rand(1,1);
else
end
    xd2=f3(n2);
    yd2=f4(n2);
    xdp2=0;
    ydp2=0;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%robot 1

    xx1=x(4)*cos(x(3));
    yy1=x(4)*sin(x(3));
    u1=c11*(xd-x(1)) + c12*(xdp-xx1);
    u2=c21*(yd-x(2)) + c22*(ydp-yy1);
    W=((u2*cos(x(3)))-u1*sin(x(3)))/x(4);
    Wl = (x(4)/r) - ((L*W)/(2*r));
    Wr = (x(4)/r) + ((L*W)/(2*r));
    vL = r*Wl;
    vR = r*Wr;
    v=(vR+vL)/2;
    w=(vR-vL)/L;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%modulos vectores
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%distancia entre robot 1 y 2
    if sqrt((x(1)-x(5))^2 + (x(2)-x(6))^2) <=.5
        delta_r12=10;
    else
        delta_r12=0;
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%robot 2

    xx2=x(8)*cos(x(7));
    yy2=x(8)*sin(x(7));
    u3=c11*(xd2-x(5) -delta_r12*sign(x(1)-x(5))) + c12*(xdp2-xx2);
    u4=c21*(yd2-x(6) -delta_r12*sign(x(2)-x(6))) + c22*(ydp2-yy2);
    W2=((u4*cos(x(7)))-u3*sin(x(7)))/x(8);
    Wl2 = (x(8)/r) - ((L*W2)/(2*r));
    Wr2 = (x(8)/r) + ((L*W2)/(2*r));
    vL2 = r*Wl2;
    vR2 = r*Wr2;
    v2=(vR2+vL2)/2;
    w2=(vR2-vL2)/L;

ecua=[

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%robot 1
    v*cos(x(3));           %x(1)
    v*sin(x(3));          %x(2)
    w;                     %x(3)
    u1*cos(x(3))+u2*sin(x(3)); %x(4)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%robot 2
    v2*cos(x(7));         %x(5)
    v2*sin(x(7));        %x(6)
    w2;                   %x(7)
    u3*cos(x(7))+u4*sin(x(7)); %x(8)
    ];
end

```

Código 5. Código de programación en Matlab para el sistema que genera ondas oscilatorias.

```
clc
clear
close all
fa(1)=.1;
fb(1)=.2;
wa(1)=2*pi*fa(1);
wb(1)=2*pi*fb(1);
OSC1(1)=0;
OSC2(1)=0;
tn(1)=0;
c=0.0001;
for i=1:50000
    OSC1(i+1)=sin(wa(i)*tn(i)); %oscilador asociado al robot 1
    OSC2(i+1)=sin(wb(i)*tn(i)); %oscilador asociado al robot 2
    wa(i+1)=wa(i)+c*(-wa(i)+wb(i));
    wb(i+1)=wb(i)+c*(-wb(i)+wa(i));
    tn(i+1)=tn(i)+.001;
end
plot(tn,OSC1,tn,OSC2)
xlabel('t','FontSize', 18)
ylabel('Magnitud','FontSize', 18)
legend('Robot 1', 'Robot 2')
set(gca, 'fontsize', 18)
```

Código 6. Programa para la sincronización espacialmente dependiente de los osciladores para 5 robots móviles.

```
function continuodiferencial()
clear all;
clc;
close all;
global n
global n2
global n3
global n4
global n5
global xx
global yy
global zz
global f1
global f2
global f3
global f4
global f5
global f6
global f7
global f8
global f9
global f10
n=1;
n2=1;
n3=1;
n4=1;
n5=1;
tspan=[0:.001:100];
N=5;%numero de robots
c=5; %numeros aleatorios entre c y d
d=-5;%numeros aleatorios entre c y d
%%%%%%%%%condicioens iniciales aleatorias entre c y d
```

```

x0=c + (d-c).*rand(1,N);
y0=c + (d-c).*rand(1,N);
Con0=[x0(1) y0(1) 0 0.01 x0(2) y0(2) 0 0.01 x0(3) y0(3) 0 0.01 x0(4) y0(4)
0 0.01 x0(5) y0(5) 0 0.01]
[t,Y]=ode45(@f,tspan,[Con0]);
figure
subplot(3,1,1)
plot(t,Y(:,1),t,Y(:,5),t,Y(:,9),t,Y(:,13),t,Y(:,17))
xlabel('t','FontSize', 18)
ylabel('x','FontSize', 18)
set(gca, 'fontsize', 18)
subplot(3,1,2)
plot(t,Y(:,2),t,Y(:,6),t,Y(:,10),t,Y(:,14),t,Y(:,18))
xlabel('t','FontSize', 18)
ylabel('y','FontSize', 18)
set(gca, 'fontsize', 18)
subplot(3,1,3)
plot(t,Y(:,3),t,Y(:,7),t,Y(:,11),t,Y(:,15),t,Y(:,19))
xlabel('t','FontSize', 18)
ylabel('\theta','FontSize', 18)
set(gca, 'fontsize', 18)
figure
plot(Y(:,1),Y(:,2),Y(:,5),Y(:,6),Y(:,9),Y(:,10),Y(:,13),Y(:,14),Y(:,17),Y(:,18))
xlabel('x','FontSize', 18)
ylabel('y','FontSize', 18)
legend('Robot 1','Robot 2','Robot 3','Robot 4','Robot 5')
set(gca, 'fontsize', 18)
figure
subplot(2,1,1)
plot(t,Y(:,1)-Y(:,5),t,Y(:,1)-Y(:,9),t,Y(:,1)-Y(:,13),t,Y(:,1)-Y(:,17))
xlabel('t','FontSize', 18)
ylabel('Error en x','FontSize', 18)
legend('Error 2-1','Error 3-1','Error 4-1','Error 5-1')

```

```

set(gca, 'fontsize', 18)
subplot(2,1,2)
plot(t,Y(:,2)-Y(:,6),t,Y(:,2)-Y(:,10),t,Y(:,2)-Y(:,14),t,Y(:,2)-Y(:,18))
xlabel('t','FontSize', 18)
ylabel('Error en y','FontSize', 18)
legend('Error 2-1','Error 3-1','Error 4-1','Error 5-1')
set(gca, 'fontsize', 18)

figure

plot(t,sqrt((Y(:,1)-Y(:,5)).^2 + (Y(:,2)-Y(:,6)).^2),t,sqrt((Y(:,1)-
Y(:,9)).^2 + (Y(:,2)-Y(:,10)).^2),t,sqrt((Y(:,1)-Y(:,13)).^2 + (Y(:,2)-
Y(:,14)).^2),t,sqrt((Y(:,1)-Y(:,17)).^2 + (Y(:,2)-Y(:,18)).^2),t,sqrt((Y(:,5)-
Y(:,9)).^2 + (Y(:,6)-Y(:,10)).^2),t,sqrt((Y(:,5)-Y(:,13)).^2 + (Y(:,6)-
Y(:,14)).^2),t,sqrt((Y(:,5)-Y(:,17)).^2 + (Y(:,6)-Y(:,18)).^2),t,sqrt((Y(:,9)-
Y(:,13)).^2 + (Y(:,10)-Y(:,14)).^2),t,sqrt((Y(:,9)-Y(:,17)).^2 + (Y(:,10)-
Y(:,18)).^2),t,sqrt((Y(:,13)-Y(:,17)).^2 + (Y(:,14)-Y(:,18)).^2))

xlabel('t','FontSize', 18)

ylabel('Modulo vector Todos los Robots','FontSize', 18)

legend('MV R1,R2', 'MV R1,R3', 'MV R1,R4', 'MV R1,R5','MV R2,R3', 'MV
R2,R4', 'MV R2,R5', 'MV R3,R4', 'MV R3,R5', 'MV R4,R5')

set(gca, 'fontsize', 18)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%sincronizacion de la luciernaga

Variablesincro12=0;

for i=1:length(t)
    if (sqrt((Y(i,1)-Y(i,5)).^2 + (Y(i,2)-Y(i,6)).^2)) <1.1
        Variablesincro12(i)=1;
    else
        Variablesincro12(i)=0;
    end
end

Variablesincro13=0;

for i=1:length(t)
    if (sqrt((Y(i,1)-Y(i,9)).^2 + (Y(i,2)-Y(i,10)).^2)) <1.1
        Variablesincro13(i)=1;
    else
        Variablesincro13(i)=0;
    end
end

```

```

end
    Variablesincro14=0;
for i=1:length(t)
    if (sqrt((Y(i,1)-Y(i,13)).^2 + (Y(i,2)-Y(i,14)).^2)) <1.1
        Variablesincro14(i)=1;
    else
        Variablesincro14(i)=0;
    end
end
    Variablesincro15=0;
for i=1:length(t)
    if (sqrt((Y(i,1)-Y(i,17)).^2 + (Y(i,2)-Y(i,18)).^2)) <1.1
        Variablesincro15(i)=1;
    else
        Variablesincro15(i)=0;
    end
end
    Variablesincro23=0;
for i=1:length(t)
    if (sqrt((Y(i,5)-Y(i,9)).^2 + (Y(i,6)-Y(i,10)).^2)) <1.1
        Variablesincro23(i)=1;
    else
        Variablesincro23(i)=0;
    end
end
    Variablesincro24=0;
for i=1:length(t)
    if (sqrt((Y(i,5)-Y(i,13)).^2 + (Y(i,6)-Y(i,14)).^2)) <1.1
        Variablesincro24(i)=1;
    else
        Variablesincro24(i)=0;
    end
end

```

```

        Variablesincro25=0;
for i=1:length(t)
    if (sqrt((Y(i,5)-Y(i,17)).^2 + (Y(i,6)-Y(i,18)).^2)) <1.1
        Variablesincro25(i)=1;
    else
        Variablesincro25(i)=0;
    end
end

        Variablesincro34=0;
for i=1:length(t)
    if (sqrt((Y(i,9)-Y(i,13)).^2 + (Y(i,10)-Y(i,14)).^2)) <1.1
        Variablesincro34(i)=1;
    else
        Variablesincro34(i)=0;
    end
end

        Variablesincro35=0;
for i=1:length(t)
    if (sqrt((Y(i,9)-Y(i,17)).^2 + (Y(i,10)-Y(i,18)).^2)) <1.1
        Variablesincro35(i)=1;
    else
        Variablesincro35(i)=0;
    end
end

        Variablesincro45=0;
for i=1:length(t)
    if (sqrt((Y(i,13)-Y(i,17)).^2 + (Y(i,14)-Y(i,18)).^2)) <1.1
        Variablesincro45(i)=1;
    else
        Variablesincro45(i)=0;
    end
end
k12=find(Variablesincro12>0);

```

```
posmin12=min(k12);
k13=find(Variablesincro13>0);
posmin13=min(k13);
k14=find(Variablesincro14>0);
posmin14=min(k14);
k15=find(Variablesincro15>0);
posmin15=min(k15);
k23=find(Variablesincro23>0);
posmin23=min(k23);
k24=find(Variablesincro24>0);
posmin24=min(k24);
k25=find(Variablesincro25>0);
posmin25=min(k25);
k34=find(Variablesincro34>0);
posmin34=min(k34);
k35=find(Variablesincro35>0);
posmin35=min(k35);
k45=find(Variablesincro45>0);
posmin45=min(k45);

fa(1)=.1;
fb(1)=.5;
fc(1)=.10;
fd(1)=.15;
fe(1)=.20;

wa(1)=2*pi*fa(1);
wb(1)=2*pi*fb(1);
wc(1)=2*pi*fc(1);
wd(1)=2*pi*fd(1);
we(1)=2*pi*fe(1);

OSC1(1)=0;
OSC2(1)=0;
OSC3(1)=0;
OSC4(1)=0;
```

```

OSC5(1)=0;
tn(1)=0;
    c=0;
    c1=0;
    c2=0;
    c3=0;
    c4=0;
    c5=0;
    c6=0;
    c7=0;
    c8=0;
    c9=0;
for i=1:length(t)
    if i < posmin12
        Variablesincro112(i)=0;
        c=0;
    else
        Variablesincro112(i)=1;
        c=0.001;
    end
    if i < posmin13
        Variablesincro113(i)=0;
        c1=0;
    else
        Variablesincro113(i)=1;
        c1=0.001;
    end
    if i < posmin14
        Variablesincro114(i)=0;
        c2=0;
    else
        Variablesincro114(i)=1;
        c2=0.001;
    end
end

```

```

end
if i < posmin15
    Variablesincro115(i)=0;
    c3=0;
else
    Variablesincro115(i)=1;
    c3=0.001;
end
if i < posmin23
    Variablesincro223(i)=0;
    c4=0;
else
    Variablesincro223(i)=1;
    c4=0.001;
end
if i < posmin24
    Variablesincro224(i)=0;
    c5=0;
else
    Variablesincro224(i)=1;
    c5=0.001;
end

if i < posmin25
    Variablesincro225(i)=0;
    c6=0;
else
    Variablesincro225(i)=1;
    c6=0.001;
end
if i < posmin34
    Variablesincro334(i)=0;
    c7=0;

```

```

else
    Variablesincro334(i)=1;
    c7=0.001;
end
if i < posmin35
    Variablesincro335(i)=0;
    c8=0;
else
    Variablesincro335(i)=1;
    c8=0.001;
end
if i < posmin45
    Variablesincro445(i)=0;
    c9=0;
else
    Variablesincro445(i)=1;
    c9=0.001;
end
end
OSC5(i+1)=sin(wa(i)*tn(i)); %oscilador asociado al robot 1
OSC4(i+1)=sin(wb(i)*tn(i)); %oscilador asociado al robot 2
OSC3(i+1)=sin(wc(i)*tn(i)); %oscilador asociado al robot 3
OSC2(i+1)=sin(wd(i)*tn(i)); %oscilador asociado al robot 4
OSC1(i+1)=sin(we(i)*tn(i)); %oscilador asociado al robot 5
wa(i+1)=wa(i)+c*(-1*wa(i)+wb(i))+c1*(-1*wa(i)+wc(i))+c2*(-
1*wa(i)+wd(i))+c3*(-1*wa(i)+we(i));
wb(i+1)=wb(i)+c4*(-1*wb(i)+wc(i))+c5*(-1*wb(i)+wd(i))+c6*(-
1*wb(i)+we(i));
wc(i+1)=wc(i)+c7*(-1*wc(i)+wd(i))+c8*(-1*wc(i)+we(i));
wd(i+1)=wd(i)+c9*(-1*wd(i)+we(i));
we(i+1)=we(i);
tn(i+1)=tn(i)+.001;
end
figure
subplot(2,1,1)

```

```

plot(t,Variablesincro12,t,Variablesincro13,t,Variablesincro14,t,Variablesincro15,t,Variablesincro23,t,Variablesincro24,t,Variablesincro25,t,Variablesincro34,t,Variablesincro35,t,Variablesincro45)

xlabel('t','FontSize', 18)

ylabel('Anticolisiones','FontSize', 18)

set(gca, 'fontsize', 18)

subplot(2,1,2)

plot(t,sqrt((Y(:,1)-Y(:,5)).^2 + (Y(:,2)-Y(:,6)).^2),t,sqrt((Y(:,1)-Y(:,9)).^2 + (Y(:,2)-Y(:,10)).^2),t,sqrt((Y(:,1)-Y(:,13)).^2 + (Y(:,2)-Y(:,14)).^2),t,sqrt((Y(:,1)-Y(:,17)).^2 + (Y(:,2)-Y(:,18)).^2),t,sqrt((Y(:,5)-Y(:,9)).^2 + (Y(:,6)-Y(:,10)).^2),t,sqrt((Y(:,5)-Y(:,13)).^2 + (Y(:,6)-Y(:,14)).^2),t,sqrt((Y(:,5)-Y(:,17)).^2 + (Y(:,6)-Y(:,18)).^2),t,sqrt((Y(:,9)-Y(:,13)).^2 + (Y(:,10)-Y(:,14)).^2),t,sqrt((Y(:,9)-Y(:,17)).^2 + (Y(:,10)-Y(:,18)).^2),t,sqrt((Y(:,13)-Y(:,17)).^2 + (Y(:,14)-Y(:,18)).^2))

xlabel('t','FontSize', 18)

ylabel('Modulo vector','FontSize', 18)

set(gca, 'fontsize', 18)

figure

Tnn= linspace(0,100,length(OSC1));

subplot(2,1,1)

plot(Tnn,OSC1,Tnn,OSC2)

xlabel('t','FontSize', 18)

ylabel('OSCILADORES','FontSize', 18)

legend('Oscilador 1','Oscilador 2')

set(gca, 'fontsize', 18)

subplot(2,1,2)

plot(Tnn,OSC1,Tnn,OSC3)

xlabel('t','FontSize', 18)

ylabel('OSCILADORES','FontSize', 18)

legend('Oscilador 1','Oscilador 3')

set(gca, 'fontsize', 18)

figure

Tnn= linspace(0,100,length(OSC1));

subplot(2,1,1)

plot(Tnn,OSC1,Tnn,OSC4)

xlabel('t','FontSize', 18)

```

```

ylabel('OSCILADORES','FontSize', 18)
legend('Oscilador 1','Oscilador 4')
set(gca, 'fontsize', 18)
subplot(2,1,2)
plot(Tnn,OSC1,Tnn,OSC5)
xlabel('t','FontSize', 18)
ylabel('OSCILADORES','FontSize', 18)
legend('Oscilador 1','Oscilador 5')
set(gca, 'fontsize', 18)
figure
Tnn= linspace(0,100,length(OSC2));
subplot(2,1,1)
plot(Tnn,OSC2,Tnn,OSC3)
xlabel('t','FontSize', 18)
ylabel('OSCILADORES','FontSize', 18)
legend('Oscilador 2','Oscilador 3')
set(gca, 'fontsize', 18)
subplot(2,1,2)
plot(Tnn,OSC2,Tnn,OSC4)
xlabel('t','FontSize', 18)
ylabel('OSCILADORES','FontSize', 18)
legend('Oscilador 2','Oscilador 4')
set(gca, 'fontsize', 18)
figure
Tnn= linspace(0,100,length(OSC2));
subplot(2,1,1)
plot(Tnn,OSC2,Tnn,OSC5)
xlabel('t','FontSize', 18)
ylabel('OSCILADORES','FontSize', 18)
legend('Oscilador 2','Oscilador 5')
set(gca, 'fontsize', 18)
Tnn= linspace(0,100,length(OSC3));
subplot(2,1,2)

```

```

plot(Tnn,OSC3,Tnn,OSC4)
xlabel('t','FontSize', 18)
ylabel('OSCILADORES','FontSize', 18)
legend('Oscilador 3','Oscilador 4')
set(gca, 'fontsize', 18)

figure

Tnn= linspace(0,100,length(OSC3));
subplot(2,1,1)
plot(Tnn,OSC3,Tnn,OSC5)
xlabel('t','FontSize', 18)
ylabel('OSCILADORES','FontSize', 18)
legend('Oscilador 3','Oscilador 5')
set(gca, 'fontsize', 18)

Tnn= linspace(0,100,length(OSC4));
subplot(2,1,2)
plot(Tnn,OSC4,Tnn,OSC5)
xlabel('t','FontSize', 18)
ylabel('OSCILADORES','FontSize', 18)
legend('Oscilador 4','Oscilador 5')
set(gca, 'fontsize', 18)

figure

plot(t,Variablesincro112,t,Variablesincro113,t,Variablesincro114,t,Variablesincro115,t,Variablesincro223,t,Variablesincro224,t,Variablesincro225,t,Variablesincro334,t,Variablesincro335,t,Variablesincro445)

xlabel('t','FontSize', 18)
ylabel('','FontSize', 18)
set(gca, 'fontsize', 18)

legend('Sincronizacion 1-2','Sincronizacion 1-3','Sincronizacion 1-4','Sincronizacion 1-5','Sincronizacion 2-3','Sincronizacion 2-4','Sincronizacion 2-5','Sincronizacion 3-4','Sincronizacion 3-5','Sincronizacion 4-5')

figure

Tnn= linspace(0,100,length(OSC1));
plot(Tnn,OSC1,Tnn,OSC2,Tnn,OSC3,Tnn,OSC4,Tnn,OSC5)
xlabel('t','FontSize', 18)
ylabel('OSCILADORES','FontSize', 18)

```

```

    legend('Oscilador 1','Oscilador 2','Oscilador 3','Oscilador 4','Oscilador
5')
    set(gca, 'fontsize', 18)
end
function [ecua]=f(t,x)
global n
global n2
global n3
global n4
global n5
global xx
global yy
global zz
global f1
global f2
global f3
global f4
global f5
global f6
global f7
global f8
global f9
global f10
r=20.5/1000;
L=88.41/1000;
c11=20;
c21=20;
c12=20;
c22=20;
a=20;
b=-20;
f1(1) = a + (b-a).*rand(1,1);
f2(1) = a + (b-a).*rand(1,1);
if abs(f1(n)-x(1))<.1 && abs(f2(n)-x(2))<.1

```

```

n=n+1
f1(n) = a + (b-a).*rand(1,1);
f2(n) = a + (b-a).*rand(1,1);
else
end

xd=f1(n);
yd=f2(n);
xdp=0;
ydp=0;

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

a=20;
b=-20;
f3(1) = a + (b-a).*rand(1,1);
f4(1) = a + (b-a).*rand(1,1);
if abs(f3(n2)-x(5))<.1 && abs(f4(n2)-x(6))<.1
n2=n2+1
f3(n2) = a + (b-a).*rand(1,1);
f4(n2) = a + (b-a).*rand(1,1);
else
end

xd2=f3(n2);
yd2=f4(n2);
xdp2=0;
ydp2=0;

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

a=20;
b=-20;
f5(1) = a + (b-a).*rand(1,1);
f6(1) = a + (b-a).*rand(1,1);
if abs(f5(n3)-x(9))<.1 && abs(f6(n3)-x(10))<.1
n3=n3+1
f5(n3) = a + (b-a).*rand(1,1);
f6(n3) = a + (b-a).*rand(1,1);

```

```

else
end

    xd3=f5(n3);
    yd3=f6(n3);
    xdp3=0;
    ydp3=0;

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

a=20;
b=-20;
f7(1)=a + (b-a).*rand(1,1);
f8(1)=a + (b-a).*rand(1,1);
if abs(f7(n4)-x(13))<.1 && abs(f8(n4)-x(14))<.1
    n4=n4+1
    f7(n4) = a + (b-a).*rand(1,1);
    f8(n4) = a + (b-a).*rand(1,1);
else
end

    xd4=f7(n4);
    yd4=f8(n4);
    xdp4=0;
    ydp4=0;

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

a=20;
b=-20;
f9(1)=a + (b-a).*rand(1,1);
f10(1)=a + (b-a).*rand(1,1);
if abs(f9(n5)-x(17))<.1 && abs(f10(n5)-x(18))<.1
    n5=n5+1
    f9(n5) = a + (b-a).*rand(1,1);
    f10(n5) = a + (b-a).*rand(1,1);
else
end

    xd5=f9(n5);

```

```

yd5=f10(n5);
xdp5=0;
ydp5=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%robot 1
xx1=x(4)*cos(x(3));
yy1=x(4)*sin(x(3));
u1=c11*(xd-x(1)) + c12*(xdp-xx1);
u2=c21*(yd-x(2)) + c22*(ydp-yy1);
W=(u2*cos(x(3))-u1*sin(x(3)))/x(4);
Wl = (x(4)/r) - ((L*W)/(2*r));
Wr = (x(4)/r) + ((L*W)/(2*r));
vL = r*Wl;
vR = r*Wr;
v=(vR+vL)/2;
w=(vR-vL)/L;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%modulos vectores
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%distancia entre robot 1 y 2
if sqrt((x(1)-x(5))^2 + (x(2)-x(6))^2) <1
delta_r12=10;
else
delta_r12=0;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%distancia entre robot 1 y 3
if sqrt((x(1)-x(9))^2 + (x(2)-x(10))^2) <1
delta_r13=10;
else
delta_r13=0;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%distancia entre robot 1 y 4
if sqrt((x(1)-x(13))^2 + (x(2)-x(14))^2) <1
delta_r14=10;
else
delta_r14=0;

```

```

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%distancia entre robot 1 y 5
if sqrt((x(1)-x(17))^2 + (x(2)-x(18))^2) <1
delta_r15=10;
else
delta_r15=0;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%distancia entre robot 2 y 3
if sqrt((x(9)-x(5))^2 + (x(10)-x(6))^2) <1
delta_r23=10;
else
delta_r23=0;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%distancia entre robot 2 y 4
if sqrt((x(13)-x(5))^2 + (x(14)-x(6))^2) <1
delta_r24=10;
else
delta_r24=0;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%distancia entre robot 2 y 5
if sqrt((x(17)-x(5))^2 + (x(18)-x(6))^2) <1
delta_r25=10;
else
delta_r25=0;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%distancia entre robot 3 y 4
if sqrt((x(13)-x(9))^2 + (x(14)-x(10))^2) <1
delta_r34=10;
else
delta_r34=0;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%distancia entre robot 3 y 5
if sqrt((x(17)-x(9))^2 + (x(18)-x(10))^2) <1

```

```

delta_r35=10;
else
delta_r35=0;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%distancia entre robot 4 y 5
if sqrt((x(17)-x(13))^2 + (x(18)-x(14))^2) <1
delta_r45=10;
else
delta_r45=0;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%robot 2
xx2=x(8)*cos(x(7));
yy2=x(8)*sin(x(7));
u3=c11*(xd2-x(5) -delta_r12*sign(x(1)-x(5))-delta_r23*sign(x(9)-x(5))-
delta_r24*sign(x(13)-x(5))-delta_r25*sign(x(17)-x(5))) + c12*(xdp2-xx2);
u4=c21*(yd2-x(6) -delta_r12*sign(x(2)-x(6))-delta_r23*sign(x(10)-
x(6))-delta_r24*sign(x(14)-x(6))-delta_r25*sign(x(18)-x(6))) + c22*(ydp2-yy2);
W2=((u4*cos(x(7)))-u3*sin(x(7)))/x(8);
Wl2 = (x(8)/r) - ((L*W2)/(2*r));
Wr2 = (x(8)/r) + ((L*W2)/(2*r));
vL2 = r*Wl2;
vR2 = r*Wr2;
v2=(vR2+vL2)/2;
w2=(vR2-vL2)/L;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%robot 3
xx3=x(12)*cos(x(11));
yy3=x(12)*sin(x(11));
u5=c11*(xd3-x(9) -delta_r13*sign(x(1)-x(9))-delta_r23*sign(x(5)-x(9))-
delta_r34*sign(x(13)-x(9))-delta_r35*sign(x(17)-x(9))) + c12*(xdp3-xx3);
u6=c21*(yd3-x(10) -delta_r13*sign(x(2)-x(10))-delta_r23*sign(x(6)-
x(10))-delta_r34*sign(x(14)-x(10))-delta_r35*sign(x(18)-x(10))) + c22*(ydp3-
yy3);
W3=((u6*cos(x(11)))-u5*sin(x(11)))/x(12);
Wl3 = (x(12)/r) - ((L*W3)/(2*r));
Wr3 = (x(12)/r) + ((L*W3)/(2*r));
vL3 = r*Wl3;

```

```

vR3 = r*Wr3;
v3=(vR3+vL3)/2;
w3=(vR3-vL3)/L;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%robot 4
xx4=x(16)*cos(x(15));
yy4=x(16)*sin(x(15));
u7=c11*(xd4-x(13) -delta_r14*sign(x(1)-x(13))-delta_r24*sign(x(5)-
x(13))-delta_r34*sign(x(9)-x(13))-delta_r45*sign(x(17)-x(13))) + c12*(xdp4-
xx4);
u8=c21*(yd4-x(14) -delta_r14*sign(x(2)-x(14))-delta_r24*sign(x(6)-
x(14))-delta_r34*sign(x(10)-x(14))-delta_r45*sign(x(18)-x(14))) + c22*(ydp4-
yy4);
W4=((u8*cos(x(15)))-u7*sin(x(15)))/x(16);
Wl4 = (x(16)/r) - ((L*W4)/(2*r));
Wr4 = (x(16)/r) + ((L*W4)/(2*r));
vL4 = r*Wl4;
vR4 = r*Wr4;
v4=(vR4+vL4)/2;
w4=(vR4-vL4)/L;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%robot 5
xx5=x(20)*cos(x(19));
yy5=x(20)*sin(x(19));
u9=c11*(xd5-x(17) -delta_r15*sign(x(1)-x(17))-delta_r25*sign(x(5)-
x(17))-delta_r35*sign(x(9)-x(17))-delta_r45*sign(x(13)-x(17))) + c12*(xdp5-
xx5);
u10=c21*(yd5-x(18) -delta_r15*sign(x(2)-x(18))-delta_r25*sign(x(6)-
x(18))-delta_r35*sign(x(10)-x(18))-delta_r45*sign(x(14)-x(18))) + c22*(ydp5-
yy5);
W5=((u10*cos(x(19)))-u9*sin(x(19)))/x(20);
Wl5 = (x(20)/r) - ((L*W5)/(2*r));
Wr5 = (x(20)/r) + ((L*W5)/(2*r));
vL5 = r*Wl5;
vR5 = r*Wr5;
v5=(vR5+vL5)/2;
w5=(vR5-vL5)/L;
ecua=[

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%robot 1
v*cos(x(3)); %x(1)
v*sin(x(3)); %x(2)
w; %x(3)
u1*cos(x(3))+u2*sin(x(3)); %x(4)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%robot 2
v2*cos(x(7)); %x(5)
v2*sin(x(7)); %x(6)
w2; %x(7)
u3*cos(x(7))+u4*sin(x(7)); %x(8)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%robot 3
v3*cos(x(11)); %x(9)
v3*sin(x(11)); %x(10)
w3; %x(11)
u5*cos(x(11))+u6*sin(x(11)); %x(12)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%robot 4
v4*cos(x(15)); %x(13)
v4*sin(x(15)); %x(14)
w4; %x(15)
u7*cos(x(15))+u8*sin(x(15)); %x(16)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%robot 5
v5*cos(x(19)); %x(17)
v5*sin(x(19)); %x(18)
w5; %x(19)
u9*cos(x(19))+u10*sin(x(19)); %x(20)
];

```

end