



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS
AVANZADOS DEL INSTITUTO POLITÉCNICO
NACIONAL

Unidad Zacatenco
Departamento de Ingeniería Eléctrica
Sección de Mecatrónica

Navegación autónoma de un robot móvil en un ambiente
desconocido usando una red de neuronas electrónicas

T E S I S

Que presenta

Ing. Enrique Martínez Sánchez

Para obtener el grado de
Maestro en Ciencias

En la especialidad de
Ingeniería Eléctrica

Directores de la Tesis:

Dr. Alejandro Rodríguez Ángeles

Dr. Jonatán Peña Ramírez

Ciudad de México

Febrero 2022

*Para todo ese ser especial,
que en algún momento depósito su confianza y apoyo incondicional,
parece que se cumple el sueño con peso de tonelada.*

Agradecimientos

Agradezco al CONACyT por el apoyo económico que permitió realizar mis estudios de maestría, mediante la beca con programa 000331 - MAESTRÍA EN CIENCIAS EN INGENIERÍA ELÉCTRICA, y el proyecto de Ciencia Básica A1-S-26123, “Análisis, control y sincronización de sistemas complejos con interconexiones dinámicas y acoplamientos flexibles.”.

Al CINVESTAV, por otorgarme la oportunidad de pertenecer a esta gran institución. A cada uno de los doctores del Departamento de Ingeniería Eléctrica, Sección de Mecatrónica, por sus enseñanzas que han fortalecido diversos aspectos de mi vida, así como a todo el personal académico y administrativo de la Sección.

A mis asesores, los doctores Alejandro Rodríguez Angeles y Jonatán Peña Ramírez, por todas las herramientas y el apoyo que me brindaron a lo largo de todo el desarrollo de este trabajo.

A los doctores Hugo Rodríguez Cortés y Jaime Álvarez Gallegos, así como al M. en C. Ricardo Francisco Núñez Pérez por fungir como sinodales de este trabajo.

Por introducirme en este camino a los doctores Ángel F. Palacios Lazcano y Francisco J. Cortés González.

A mis compañeros y amigos, con los cuales compartí toda esta ardua y maravillosa etapa, gracias por las diversas experiencias y momentos que atesoré: Diego, Ignacio, Saulo, Ray, Ezequiel, Olaf, Jurgen, Michelle, Gabriel, Daniel, Omar, Eduardo, Felipe, Alan, Erick, David, Miguel, Ivan, Brian, Javier, Fernando, Ulises e Irving.

A mis padres María Concepción y Enrique, a mis hermanos M. Elena, Ignacio, Diana y Rubi, así como a mis figuras paternas y maternas que se han involucrado a lo largo de todo mi trayecto Samuel, Raúl, Salvador, Laura, Sandra, Beatriz, Liliana y Concepción, gracias por su cariño.

Por haber sido mis oídos y mi fuente de animo agradezco a Fátima, Frida, Samuel, Viridiana, Gustavo, S. Estafanía y Link.

Finalmente, a mis antiguos colegas que aun mantienen su aprecio por mi: J. Francisco, A. Leonardo, Alberto, Pedro, J. Luis, Ponce, Victor, Edwin, Ohtokani, Joryi, Diego, Selene, Jessica, Celeste y Nathaly.

Gracias a todos por su afecto y palabras de aliento...

Resumen

En este trabajo se propone un control híbrido para la navegación autónoma de un robot móvil en un espacio desconocido con obstáculos estáticos y dinámicos. Este control híbrido está conformado por la conmutación entre un control que permite el seguimiento de trayectoria y un control de evasión basado en neuronas tipo Hindmarsh-Rose (HR), las cuales están interconectadas mediante un acoplamiento difuso. La conmutación se da en función de la lectura de los sensores integrados al robot y se opera a nivel de velocidad de rueda. Para lograr la evasión de obstáculos, se utiliza una fuerza de acoplamiento pequeña, lo cual permite aprovechar la desincronización de los potenciales de membrana ante entradas de corriente externas diferentes para cada neurona, de esta manera se obtiene una mayor sensibilidad en la estrategia de detección. Por otra parte, se hace uso de un acoplamiento dinámico para establecer una estrategia de evasión frontal. Para mostrar el desempeño del control se presentan resultados de simulación.

Abstract

In this work, a hybrid control strategy is proposed for autonomous navigation of a mobile robot in an unknown space with static and dynamic obstacles. This hybrid control is composed by the commutation between a collision avoidance control based on Hindmarsh-Rose (HR) neurons, which are interconnected by a diffusive coupling, and a posture tracking control. The commutation is determined by the distance sensors that are incorporated on the frontal hemisphere of the mobile robot. The control is implemented at wheel speed level. A small coupling force is used to take advantage of the desynchronization on the membrane potentials when different inputs (external currents) are applied to each neuron. In this way, a greater sensitivity to small differences is obtained and a strategy of lateral avoidance is generated. On the other hand, a dynamic coupling is used to establish a strategy for frontal avoidance. Additionally, the performance of the proposed controller is numerically illustrated by means of computer simulations.

Índice general

1. Introducción	3
1.1. Generalidades	4
1.1.1. Robots móviles	4
1.1.2. Sensores y dispositivos empleados en robots móviles	7
1.1.3. Neurona y potencial de membrana	10
1.2. Estado del arte	12
1.3. Planteamiento del problema	16
1.4. Hipótesis	16
1.5. Objetivos	16
1.5.1. Objetivo general	16
1.5.2. Objetivos particulares	17
1.6. Productos generados	17
1.7. Organización de la tesis	18
2. Marco teórico	19
2.1. Modelos neuronales	19
2.1.1. Modelo Hodgkin-Huxley	19
2.1.2. FitzHugh-Nagumo	21
2.1.3. Modelo Morris-Lecar	23
2.1.4. Modelo Hindmarsh-Rose	24
2.1.5. Modelo Izhikevich	27
2.1.6. Discusión de los modelos neuronales	29
2.2. Acoplamiento de neuronas	29
2.2.1. Acoplamiento unidireccional	29
2.2.2. Acoplamiento difuso	33
2.2.3. Acoplamiento dinámico	35
2.2.3.1. Estado de relajación completa	37
2.2.4. Discusión de los acoplamientos	39

ÍNDICE GENERAL

2.3. Sincronización de neuronas	39
2.3.1. Definición 1. Sincronización completa.	39
2.3.2. Definición 2 Sincronización práctica.	40
2.4. Modelo cinemático y seguimiento de trayectoria	40
2.4.1. Modelo cinemático	40
2.4.2. Control de seguimiento de trayectoria	42
3. Diseño de la estrategia de navegación autónoma	45
3.1. Estrategia de evasión	45
3.1.1. Estrategia de evasión lateral	46
3.1.2. Estrategia de evasión frontal	48
3.1.3. Función G	50
3.2. Control híbrido	51
4. Resultados de simulación numérica	53
4.1. Implementación en Matlab-Simulink 2020b	53
4.1.1. Subsistema detector-contador de espigas	56
4.2. Co-simulación Gazebo-Simulink	57
4.2.1. Bloques ROS para la co-simulación	57
4.2.2. Diagrama a bloques para la co-simulación	58
4.3. Simulación 1	60
4.3.1. Resultados de la simulación 1	61
4.4. Simulación 2	64
4.4.1. Resultados de la simulación 2	65
4.5. Simulación 3	70
4.5.1. Resultados de la simulación 3	71
4.6. Simulación 4	76
4.6.1. Resultados de la simulación 4	78
4.7. Simulación 5	81
4.7.1. Resultados de la simulación 5	82
5. Conclusiones y trabajo futuro	87

Índice de figuras

1.1.	Tipos de ruedas empleadas en robots móviles con ruedas. . . .	5
1.2.	Tipos de robots de acuerdo con la clasificación de los grados de libertad δ_m y δ_s	6
1.3.	Ejemplo de sensores instalados en un robot móvil.	8
1.4.	Estructura básica de una neurona.	10
1.5.	Comunicación de tres neuronas que intercambian potenciales de membrana.	11
1.6.	Diagrama de un potencial de acción, o spike, generado por una neurona.	11
1.7.	Actividad o modo spiking en el potencial de membrana.	12
1.8.	Actividades tipo burst en el potencial de membrana.	13
2.1.	Potencial de membrana con actividad tipo spiking, Hodgkin y Huxley.	21
2.2.	Circuito electrónico Jin-Ichi Nagumo.	22
2.3.	Potencial de membrana con actividad tipo spiking, FitzHugh y Nagumo.	22
2.4.	Potencial de membrana con actividad tipo spiking, modelo Morris y Lecar.	24
2.5.	Actividades en el potencial de membrana del modelo HR, con la misma sintonización de parámetros y diferentes corrientes externas.	26
2.6.	Actividades en el potencial de membrana del modelo Izhikevich.	28
2.7.	Acoplamiento unidireccional entre dos neuronas HR.	30
2.8.	Potenciales de membrana de dos neuronas HR (2.7)-(2.8) interconectadas con acoplamiento unidireccional.	31
2.9.	Acoplamientos unidireccionales que muestran la sincronización del sistema completo.	32
2.10.	Acoplamiento difuso entre dos neuronas HR.	33

ÍNDICE DE FIGURAS

2.11. Potenciales de membrana de dos neuronas HR (2.10)-(2.11) interconectadas con acoplamiento difuso (2.12).	34
2.12. Potenciales de membrana de dos neuronas HR interconectadas con acoplamiento dinámico.	36
2.13. Potencial de membrana de dos neuronas HR acopladas mediante acoplamiento dinámico, misma entrada I	38
2.14. Robot móvil tipo 2.0 y orientación de los sensores.	41
3.1. Gráfico de la función G , para un valor de $x_{fmin} = 0.5[m]$	50
4.1. Diagrama a bloques para la simulación de un robot móvil programado con el control híbrido.	54
4.2. Diagrama a bloques del subsistema que contiene la estrategia de seguimiento de trayectoria y el control híbrido.	54
4.3. Mapas utilizados en las simulaciones numéricas.	55
4.4. Diagrama a bloques del subsistema detector-contador de espigas.	56
4.5. Bloques ROS que ofrece la paquetería Robotics System Toolbox.	58
4.6. Diagrama a bloques para la simulación de un robot móvil programado con el control híbrido y la co-simulación Gazebo-Simulink.	59
4.7. Subsistema para el control de seguimiento de trayectoria y control híbrido.	59
4.8. Parámetros y configuración del bloque Subscribe.	60
4.9. Trayectorias de los robots móviles.	62
4.10. Velocidades de rueda $\omega_{l_{speed}}$ y $\omega_{r_{speed}}$. Robot 1.	62
4.11. Zoom en la evolución de los potenciales de membrana x_1, x_2 , que actúan en el robot 1. Los potenciales se desincronizan cuando se detecta un obstáculo	63
4.12. Evolución de los potenciales de membrana x_3, x_4 y el potencial de membrana x_{4A} , que actúan en el robot 1.	63
4.13. Trayectorias de los robots móviles.	66
4.14. Velocidades de rueda $\omega_{l_{speed}}$ y $\omega_{r_{speed}}$. Robot 1.	66
4.15. Errores de posición. Robot 1.	67
4.16. Zoom en la evolución de los potenciales de membrana x_1, x_2 , que actúan en el robot 1.	68
4.17. Evolución de los potenciales de membrana x_3, x_4 y el potencial de membrana x_{4A} , que actúan en el robot 1	69
4.18. Trayectorias de los robots móviles.	72
4.19. Velocidades de rueda $\omega_{l_{speed}}$ y $\omega_{r_{speed}}$. Robot 1.	72
4.20. Errores de posición. Robot 1.	73

4.21. Zoom en la evolución de los potenciales de membrana x_1, x_2 , que actúan en el robot 1.	74
4.22. Evolución de los potenciales de membrana x_3, x_4 y el potencial de membrana x_{4A} , que actúan en el robot 1.	75
4.23. Zoom en la evolución de los potenciales de membrana x_1, x_2 , que actúan en el robot 2.	75
4.24. Evolución de los potenciales de membrana x_3, x_4 y el potencial de membrana x_{4A} , que actúan en el robot 2.	76
4.25. Trayectoria del robot móvil.	78
4.26. Trayectoria del robot móvil.	79
4.27. Velocidades de rueda $\omega_{l_{speed}}$ y $\omega_{r_{speed}}$	79
4.28. Zoom en la evolución de los potenciales de membrana x_1, x_2 , que actúan en el robot móvil.	80
4.29. Evolución de los potenciales de membrana x_3, x_4 y el potencial de membrana x_{4A} , que actúan en el robot móvil.	80
4.30. Trayectoria del robot móvil.	82
4.31. Velocidades de rueda $\omega_{l_{speed}}$ y $\omega_{r_{speed}}$	83
4.32. Error de posición.	84
4.33. Zoom en la evolución de los potenciales de membrana x_1, x_2 , que actúan en el robot móvil.	85
4.34. Evolución de los potenciales de membrana x_3, x_4 y el potencial de membrana x_{4A} , que actúan en el robot móvil.	85

Capítulo 1

Introducción

La robótica ha alcanzado grandes avances en diferentes campos de aplicación industrial y servicios, esto se debe al avance que han tenido los dispositivos eléctricos y electrónicos, la obtención de nuevos materiales, así como el desarrollo en procesamiento de datos [1]. Los brazos robóticos, o manipuladores, constituyen un gran impacto en el campo industrial, donde se desempeñan tareas principalmente de transporte, montaje y ensamble de materiales y componentes, en las cuales, el robot manipulador puede moverse de forma repetitiva, con gran velocidad y precisión con el fin de acelerar y aumentar los resultados de la tarea, que en su mayoría se igualan o son superiores a los obtenidos con mano de obra humana. Sin embargo, un robot manipulador fijo tiene la desventaja de la falta de movilidad, es decir, tiene un espacio de trabajo limitado. Es por ello que un robot móvil resalta su importancia en la industria, ya que puede complementar este aspecto y dar mayor grado de libertad al desplazarse por toda la planta de fabricación [2]. En general, hoy en día se tienen diversas aplicaciones para los robots móviles en los campos industrial y de servicios, donde en cada una de ellas el robot móvil tiene cierto grado de autonomía y comparte o no el entorno y colaboración con seres humanos. Algunos ejemplos de las aplicaciones que puede desempeñar un robot móvil son: transporte de material, limpieza, vigilancia, localización de objetos, exploración no tripulada, automatización de procesos, asistencia médica, misiones militares, agricultura, entretenimiento, entre otras. El amplio catálogo de aplicaciones, así como el desempeño y eficiencia que puede proporcionar la robótica móvil, propician que sus áreas de estudio sigan extendiéndose día con día [3].

1.1. Generalidades

En esta sección, se presentan algunas generalidades que permiten abordar y conectar con las siguientes secciones. En particular, se presentan algunos conceptos básicos sobre robots móviles y sus tipos, los tipos de sensores que suelen emplearse en este tipo de robots, así como la estructura básica de una neurona y una de sus características más relevantes, el potencial de membrana. Por otra parte, se menciona el estado del arte y se da a conocer el planteamiento del problema, la hipótesis y los objetivos de este proyecto.

1.1.1. Robots móviles

Un robot móvil se define como un sistema electromecánico capaz de desplazarse sin estar sujeto físicamente a un solo punto [4].

El desarrollo de robots móviles corresponde a la necesidad de extender el campo de aplicación de la robótica. En particular, se pretende incrementar cada vez más la autonomía de los robots, con la meta final de reducir la intervención humana. El nivel de autonomía se define de acuerdo con el grado de participación de un operador a cargo del robot móvil para que este logre su tarea [5]. Dichos niveles son:

Teleoperación: El operador tiene el control completo del móvil, el cual no toma iniciativa alguna, excepto quizás, el detenerse cuando la comunicación falla.

Modo seguro: El operador dirige el movimiento, pero el robot toma iniciativas para protegerse a sí mismo, por ejemplo, realizar paros de emergencia para evitar colisiones o no moverse si se excede su límite de carga.

Control compartido: El robot elige su propia ruta y responde de manera autónoma al entorno, lo que le permite realizar maniobras de evasión por sí mismo; el operador únicamente debe introducir comandos de manera intermitente.

Navegación dirigida: El operador programa una ruta o una posición objetivo, la cual debe ser alcanzada por el robot utilizando sus propios medios en cuanto a evasión de obstáculos se refiere.

Completamente autónomo: El robot realiza la planificación, selección de rutas y evasión de obstáculos, además es capaz de reconocer el entorno sin mayor intervención que una instrucción de alto nivel.

Por otra parte, los robots móviles pueden clasificarse de diferentes formas, una de ellas es de acuerdo con su entorno de operación: vehículos aéreos,

acuáticos, subacuáticos y terrestres. Los robots móviles terrestres pueden ser clasificados por el tipo de locomoción utilizada, en general, los tres medios de movimiento son: por ruedas, por patas y por orugas [1, 2, 3, 4]. La locomoción por patas y orugas ha sido ampliamente estudiada, pero el mayor desarrollo está en los robots móviles con ruedas, esto es debido a las ventajas que presentan las ruedas con respecto a las patas y a las orugas. Un robot móvil con ruedas es un vehículo capaz de realizar un movimiento autónomo, que está equipado para su movimiento con sensores y actuadores que son accionados por una computadora [6]. Dentro de sus atributos más relevantes, destacan su eficiencia energética en superficies lisas y firmes, no causan desgaste significativo en la superficie donde se mueven y requieren de un número menor de partes que usualmente son menos complejas en comparación con los robots de patas y de orugas, lo que permite que su construcción sea más sencilla.

Un aspecto fundamental en los robots móviles con ruedas son los tipos de ruedas que se emplean, de entre ellas, las más convencionales son: rueda fija, rueda direccional (Steering), rueda castor y rueda sueca (Swedish), ver Figura 1.1.

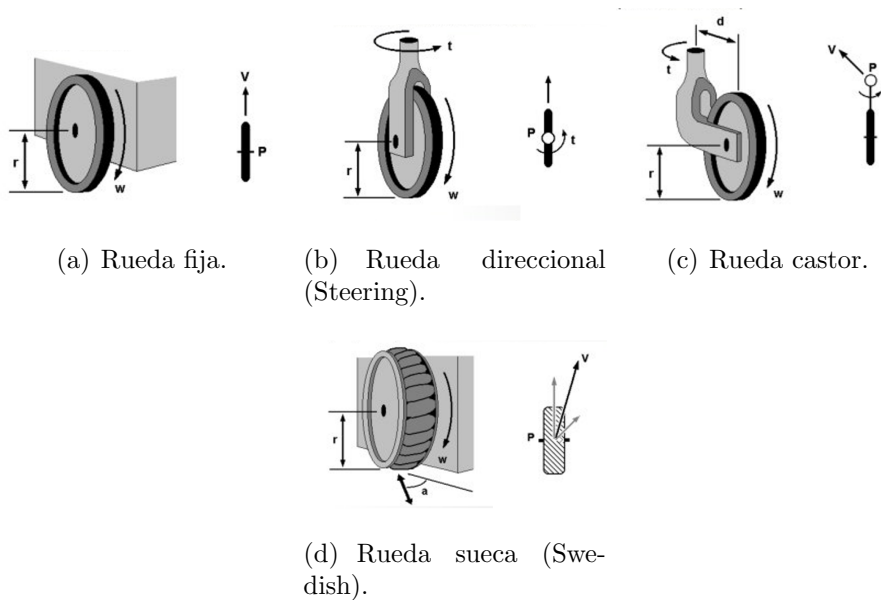


Figura 1.1: Tipos de ruedas empleadas en robots móviles con ruedas.

De acuerdo con la configuración de las ruedas y las restricciones a la movilidad de los robots, éstos pueden ser clasificados de acuerdo con su número de

CAPÍTULO 1. INTRODUCCIÓN

grados de libertad. El grado de movilidad δ_m y el grado de direccionalidad δ_s definen el tipo de estructura que se designará mediante una denominación de la forma “Robot tipo (δ_m, δ_s) ”. En [6] se ha determinado que las combinaciones posibles de estos grados de libertad (δ_m, δ_s) son cinco, las cuales se muestran en la Figura 1.2, y se describen a continuación:

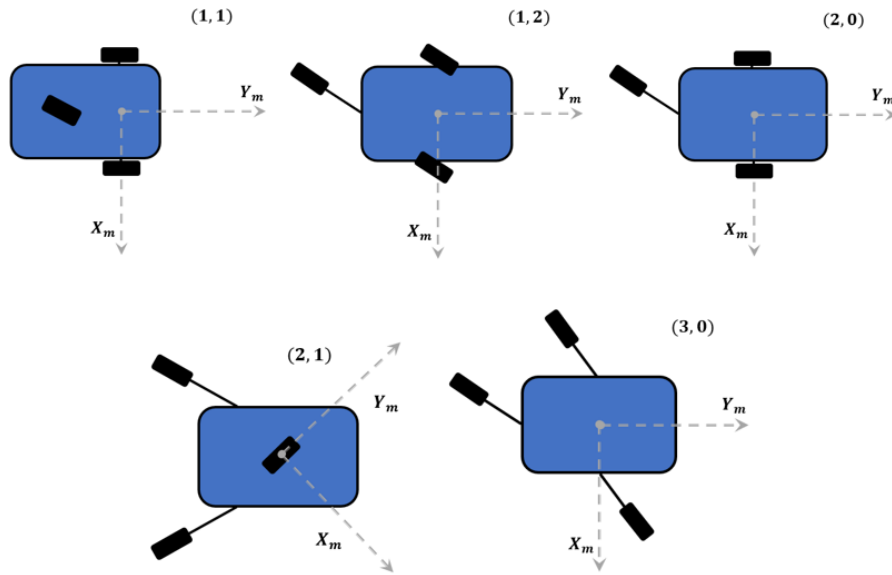


Figura 1.2: Tipos de robots de acuerdo con la clasificación de los grados de libertad δ_m y δ_s .

Robots tipo (1,1): una o más ruedas fijas en un eje común, y una o más ruedas de dirección, las cuales deben ser coordinadas para poder dar orientación a vehículos. Este tipo de vehículos no pueden desplazarse con movimientos laterales.

Robots tipo (1,2): Presentan dos ruedas de dirección independiente y no tienen ruedas fijas. Su velocidad se restringe a pertenecer a trayectorias parametrizadas simultáneamente por ambos ángulos de orientación.

Robots tipo (2,0): dos o más ruedas motrices fijas alineadas en un eje común, conlleva a que este tipo de vehículos no puedan desplazarse con movimientos laterales. Para poder girar, una de las ruedas gira a menor velocidad que la otra.

Robots tipo (2,1): dos ruedas motrices no fijas y una de dirección. La velocidad es restringida a pertenecer a trayectorias sin traslación lateral.

Robots tipo (3,0): tres ruedas motrices omnidireccionales brindan movilidad completa en el plano sin tener que reorientarse [6].

Un robot móvil autónomo puede utilizarse en tareas de localización, percepción, planificación y control entre otras. La problemática de la localización se aborda con la técnica de odometría mediante sensores y/o sistema de posicionamiento global (GPS) que le proporcionan al robot aquellos datos que corresponden a su posición y orientación en intervalos suficientemente cortos. De forma similar, se cubre la percepción gracias a la variedad de sensores que ofrecen datos del entorno. La planificación puede dividirse en generar o planificar una ruta-trayectoria y en evitar obstáculos no esperados. Para ello, existen diversos métodos que se basan en la búsqueda de caminos libres de obstáculos que minimizan la distancia recorrida en un entorno modelado mediante polígonos o descomposición del espacio en celdas [3]. La planificación de la trayectoria también, puede realizarse de forma dinámica, considerando la posición actual del vehículo y los puntos intermedios de paso definidos en la planificación de la ruta. En esta estrategia la trayectoria se corrige debido a acontecimientos no considerados. Cabe mencionar que, la definición de la trayectoria deberá contener las características cinemáticas del vehículo [2].

Una vez realizada la planificación de ruta, se plantea el problema de seguimiento de trayectoria, que para vehículos con ruedas consiste en determinar el ángulo de dirección teniendo en cuenta la posición y orientación actual del vehículo con respecto a la trayectoria que debe seguir [2].

1.1.2. Sensores y dispositivos empleados en robots móviles

Los sensores se pueden clasificar como propioceptivos o exteroceptivos, y como pasivos o activos [3].

Los sensores propioceptivos miden valores internos del sistema (robot); por ejemplo, la velocidad del motor, la carga de las ruedas, los ángulos de las articulaciones del brazo del robot y el voltaje de la batería.

Los sensores exteroceptivos adquieren información del entorno del robot; por ejemplo, mediciones de distancia, intensidad de la luz, amplitud del sonido. Por lo tanto, las mediciones de los sensores exteroceptivos son interpretadas por el robot para extraer características significativas del entorno.

Los sensores pasivos miden la energía ambiental que entra en el sensor. Ejemplos de sensores pasivos son las sondas de temperatura, los micrófonos y las cámaras con dispositivo de carga acoplada (charge-coupled device CCD).

Los sensores activos emiten energía en el entorno y luego miden la reacción ambiental. Debido a que los sensores activos pueden gestionar interacciones más controladas con el entorno, a menudo logran un rendimiento superior.

CAPÍTULO 1. INTRODUCCIÓN

Sin embargo, un sensor activo puede sufrir interferencias entre su señal y las que están fuera de su control (como las señales emitidas por otros robots cercanos o por sensores similares), las cuales pueden influir en las mediciones resultantes. Ejemplos de sensores activos son los encoders de las ruedas, los sensores ultrasónicos y los sensores de escaneo láser.

Como se mencionó anteriormente, una de las tareas más importantes para un robot móvil autónomo de cualquier tipo, es determinar su localización y conocimiento de su entorno. Para ello, los robots móviles se equipan con diversos sensores y dispositivos ver Figura 1.3, de los cuales se toman mediciones.

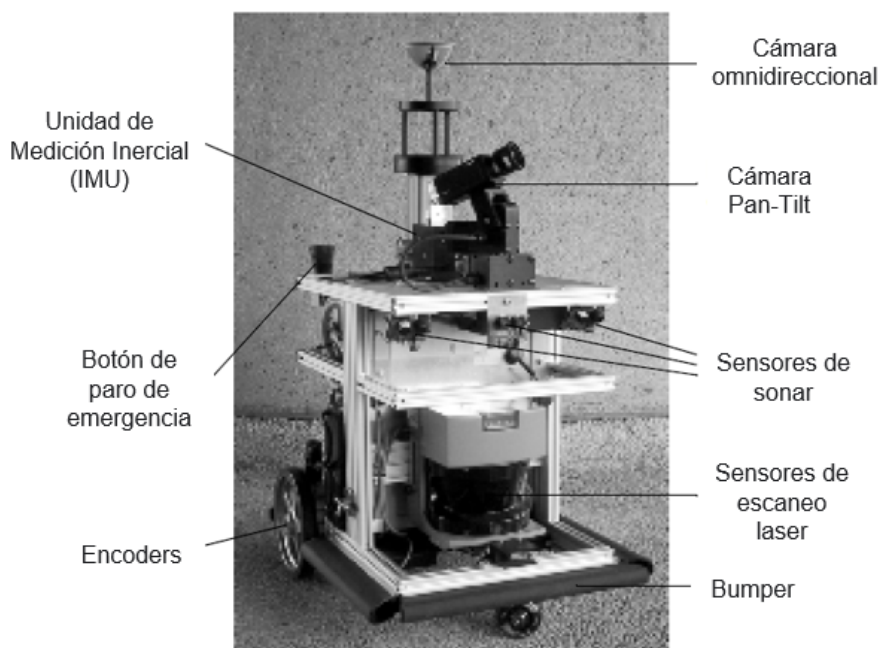


Figura 1.3: Ejemplo de sensores instalados en un robot móvil.

Para lograr la técnica de odometría y cumplir con la tarea de localización se suele recurrir a dispositivos como codificadores rotatorios, giróscopos y acelerómetros.

Codificador óptico: También conocido como encoder óptico, es un dispositivo que convierte el movimiento rotacional en una señal eléctrica (código digital). En robótica móvil se utiliza para determinar la posición y velocidad

de las ruedas. [2, 3].

Gir6scopo: es un dispositivo cuya funci3n es medir la velocidad angular y la orientaci3n del robot [3].

Aceler3metro: Este dispositivo es utilizado para medir los cambios de velocidad con respecto al tiempo, tambi3n puede medir la inclinaci3n y la vibraci3n de las m3quinas [7]

Cabe destacar que cuando se tienen aplicaciones en exteriores donde las distancias recorridas por el veh6culo aut3nomo son considerables, se emplean sistemas de posicionamiento global (GPS) mediante sat3lites [2].

Para la percepci3n del entorno se emplean diferentes tipos de sensores que en su mayor6a son exteroceptivos, los m3s utilizados son: c3maras de video, sensores de escaner l3ser, sensor infrarrojo y ultrasonido.

Las c3maras de video tienen la ventaja de su precio y no tener desgaste mec3nico para la captaci3n de la imagen. Sus desventajas son los requerimientos computacionales, la sensibilidad a las condiciones de iluminaci3n, y los problemas de calibraci3n y fiabilidad [2].

La percepci3n activa mediante sensores l3ser es un m3todo alternativo que ha cobrado relevancia en robots m3viles. Con ellos se hace un barrido del espacio en un rango espec6fico que permite obtener im3genes a distancia y reflectancia a las superficies intersectadas con el haz [2].

Los sensores de ultrasonido son econ3micos y simples para la navegaci3n, los cuales se basan en la determinaci3n del tiempo de vuelo de un pulso de sonido (entre 30[kHz] y 1[MHz]). Sin embargo, la influencia de las condiciones ambientales del entorno puede ser significativa e influir en la medida. De ser as6, debe corregirse con una calibraci3n adecuada empleando m6ltiples frecuencias y t3cnicas de filtrado, adem3s del tratamiento computacional. Otro aspecto importante, es la mala resoluci3n lateral, por lo que se deben emplear t3cnicas de enfoque mediante lentes ac6sticos o transmisores curvos [2].

La selecci3n de los sensores depende mucho de la aplicaci3n que se va desarrollar con el robot m3vil, aunque usualmente se buscan las caracter6sticas de precision, rango, inmunidad a la variaci3n originadas por las condiciones del entorno y por el propio veh6culo, as6 como su tama1o, consumo de energ6a, seguridad de funcionamiento y desgaste con el paso del tiempo [2, 3, 4, 8].

1.1.3. Neurona y potencial de membrana

Una neurona es una célula especializada en la recepción de estímulos y en la conducción del impulso nervioso. Morfológicamente, una neurona típica está compuesta por el soma, el axón y las dendritas, ver Figura 1.4 [9]. El **soma** es el cuerpo celular de la neurona que contiene el núcleo y los

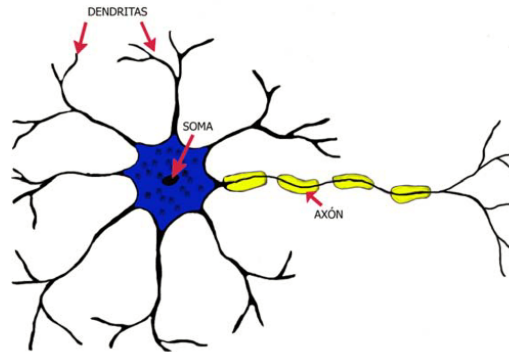


Figura 1.4: Estructura básica de una neurona.

orgánulos principales. El **axón** es una larga prolongación del soma neuronal que actúa como principal conductor donde se propaga la información en forma de señales eléctricas. Las **dendritas** son ramificaciones que proceden del soma, y actúan como los principales receptores de señales nerviosas (estímulos externos) procedentes de los receptores sensoriales o de otras neuronas [9, 10, 11].

Cada célula nerviosa está conectada a otras células nerviosas a través de sus axones y dendritas, esta conexión es denominada como sinapsis, y su comunicación es por la generación de señales eléctricas denominadas impulsos nerviosos, potenciales de acción, o potenciales de membrana, los cuales producen la liberación de los neurotransmisores, véase Figura 1.5. La célula posee una membrana citoplasmática que la separa del medio extracelular, en cada lado de esta membrana celular existe una carga eléctrica diferente, por lo que se produce una diferencia de potencial entre el interior y el exterior.

De esta manera, se define al *potencial de membrana* como aquel potencial generado entre los lados de la membrana celular que separa dos soluciones de diferente concentración iónica. La evolución temporal de este potencial de membrana es lo que caracteriza el comportamiento de una neurona [9, 10, 12].

Se dice que una neurona se encuentra en reposo o relajación cuando no recibe ningún estímulo, esto quiere decir que el interior de la célula respecto al exterior presenta una carga negativa. Este potencial puede variar en deter-

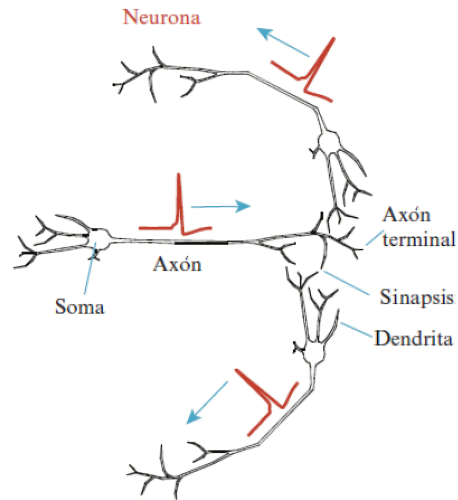


Figura 1.5: Comunicación de tres neuronas que intercambian potenciales de membrana.

minadas circunstancias, produciendo una despolarización si aumenta, o una hiperpolarización en el caso contrario, tal y como se muestra en la Figura 1.6.

Cuando el potencial de membrana se incrementa de manera muy rápida en

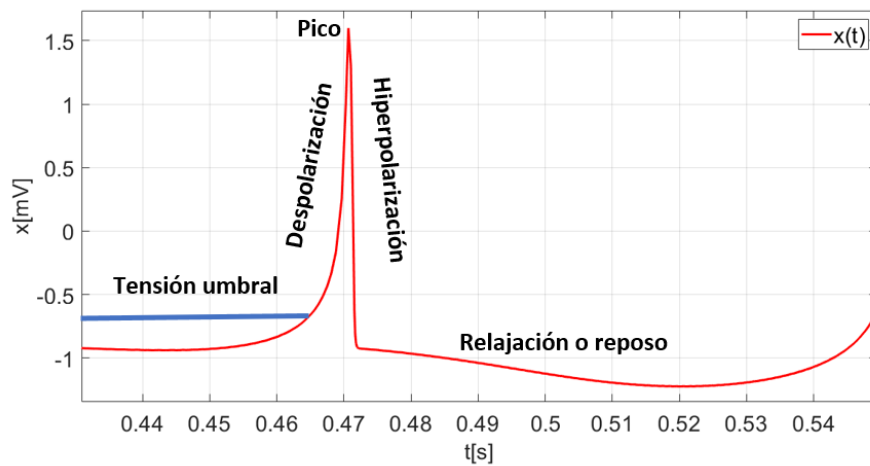


Figura 1.6: Diagrama de un potencial de acción, o spike, generado por una neurona.

un intervalo de tiempo corto, y supera un determinado umbral de despolarización,

zación debido a la carga interna, se produce un *pico* o *spike*. La alternación entre picos y estados de relajación en una neurona se conoce como actividad o modo *spiking* [9, 10, 11, 12], en otras palabras se genera una señal de tren de picos en el potencial de membrana, ver Figura 1.7. A su vez, la agrupación

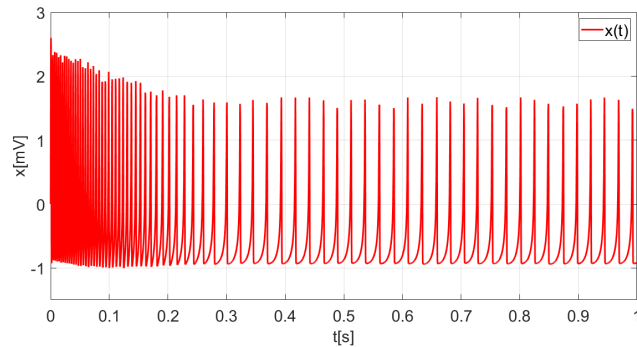
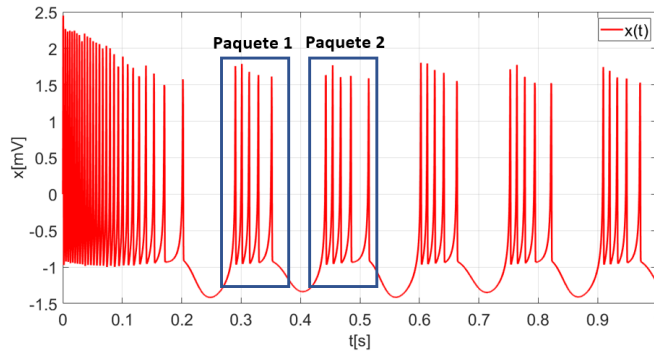


Figura 1.7: Actividad o modo spiking en el potencial de membrana.

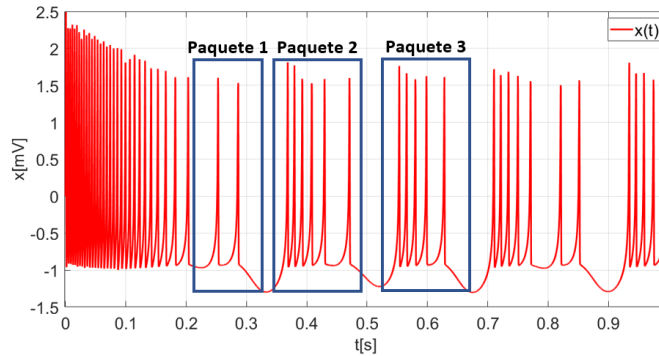
en paquetes de dos o más picos en un corto intervalo de tiempo, se conoce como *ráfagas* o *burst*. La actividad o modo en el potencial de membrana donde se presentan varios *burst* consecutivos con el mismo número de picos en cada paquete se conoce como *bursting*, ver Figura 1.8a. También, existe un modo conocido como *chaotic bursting* o *bursting caótico*, este es una variación de la actividad tipo *bursting*, la diferencia entre estos es que, hay un número diferente de picos en cada paquete consecutivo del *bursting* caótico, ver Figura 1.8b [9, 11, 13].

1.2. Estado del arte

Existen diversas aplicaciones que pueden desempeñar los robots móviles autónomos: el transporte de material, limpieza, vigilancia, localización de objetos, exploración no tripulada, entre otros [1, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22]. La navegación autónoma de los robots móviles puede realizarse por medio de la planificación previa de una ruta, esto suele realizarse en ambientes totalmente conocidos y que no cambian con el paso del tiempo, en estas condiciones se puede probar la existencia de un camino libre de colisiones y posiblemente óptimo hacia la meta (en términos de distancia, tiempo y energía empleada) [16, 17, 18]. Desafortunadamente, el caso anterior es poco común en la mayoría de las aplicaciones donde se considera un entorno real, en las que el ambiente es desconocido. Este hecho requiere que un robot móvil sea capaz de identificar y evadir obstáculos (estáticos y dinámicos), realizar



(a) Actividad o modo bursting.



(b) Actividad o modo bursting caótico.

Figura 1.8: Actividades tipo burst en el potencial de membrana.

autolocalización, tomar decisiones, etc. [1, 14, 15, 22]. Dentro de la amplia gama de controles que abordan la problemática de la navegación autónoma de un robot móvil, se hallan controles con enfoques del tipo híbrido, que son conformados por controladores para la navegación reactiva y controladores para un seguimiento de trayectoria [16, 17, 18, 19, 20]. Ejemplo de ello, son los controladores de sistemas de interferencia difusa que requieren de un entorno parcialmente conocido y se conmutan de la forma más suave posible, para elaborar una fase de pre-procesamiento dependiente a una tarea específica y así poder construir una ruta de referencia óptima en tiempo [16]. Por otra parte, está la combinación entre controles PI, y controladores de lógica difusa con algoritmos de retroceso que actúan de forma simultánea, y se encargan de establecer una fusión de datos obtenidos de sensores, construcción de mapas y tareas de planificación de rutas [17]. Para imitar comportamientos más naturales, se han desarrollado algunos controladores bio-inspirados los cuales poseen un radio de confort y un lazo externo que contiene fuerzas de evasión

[1], además, están aquellos controles que imitan el manejo de automóviles impulsados por una respuesta humana basados en lógica difusa, acciones reflexivas y toma de decisiones de alto nivel [21]. También, se hallan controles que poseen una intercomunicación inalámbrica entre una computadora móvil (que se incorpora al robot y se encarga de recopilar datos del entorno mediante sensores laser), y una computadora estacionaria que se encarga de elaborar un algoritmo de control para la evasión de objetos dinámicos y la búsqueda de una ruta adaptativa [20].

Los diagramas de Voronoi son espacios geométricos o diagramas conformados por segmentos de líneas rectas y de parábolas que rodean a cada obstáculo, creando así una región para cada uno. En el área de la robótica el concepto fundamental de la construcción de los diagramas de Voronoi es ampliar la distancia existente entre el robot y los obstáculos al máximo. De esta manera, la red resultante es un espacio geométrico de las configuraciones que se encuentran equidistantes en los obstáculos más próximos del entorno y la manera de evitar las colisiones con este método es diseñar una trayectoria que siga el robot sobre las líneas generadas en la red de Voronoi, lo que lograría que el dispositivo se mueva siempre en medio entre dos obstáculos de la zona [23].

La exploración rápida de árboles aleatorios o Rapidly Exploring Random Trees (RRT) es un método que se implementa en la robótica debido a que garantiza el descubrimiento completo y la exploración del mapa en el que se requiere que el dispositivo se desplace independientemente de la complejidad del entorno. Se basa en la generación de una estructura arbórea a través de la construcción aleatoria de ramas que rellenan un espacio determinado y que crecen a partir de un punto de origen. Esta particularidad convierte al RRT en un método de generación de rutas de tipo aproximado, ya que genera un muestreo aleatorio del espacio y después una subrutina de detección de colisiones que decide si las trayectorias configuradas son viables o si implican un posible choque, de esta manera presenta una solución simple que no depende de ningún parámetro inicial y favorece la exploración de toda la región para la óptima búsqueda de obstáculos evitando la colisión con estos [23, 24].

Los métodos o estrategias de exploración rápida de árboles aleatorios y diagramas de Voronoi son herramientas efectivas para la planeación de rutas en la navegación autónoma de un robot móvil, sin embargo, tienen la desventaja de no poder contemplar obstáculos dinámicos, es por ello que se suelen utilizar dentro de algoritmos de aprendizaje profundo (deep learning) que ayudan a resolver los problemas de evasión, distinguiendo los obstáculos

estáticos y dinámicos por separado y proponiendo reglas para cada caso [25].

El esquema de control basado en campos potenciales se caracteriza por ser un método que expresa la matemática de generación de trayectorias de forma simple y su tiempo de cálculo es bajo, de tal manera que este algoritmo funciona en tiempo real y tiene la capacidad de ser utilizado dinámicamente debido al sencillo tratamiento de obstáculos fijos y móviles. El estado del robot y la dinámica artificial puede producir una velocidad virtual y aceleración que se utilizan como una referencia instantánea para controlar la posición del robot. En la técnica de campos potenciales para la generación de trayectorias se considera que el movimiento del dispositivo en un campo de fuerzas, genera fuerzas de repulsión con los obstáculos y fuerzas de atracción con el punto de llegada (haciendo analogía con campos potenciales eléctricos, considerando al robot móvil como una carga eléctrica, a los obstáculos como cargas eléctricas del mismo signo que el robot y a la meta o punto de llegada como una carga de signo contrario a la del robot). De este modo, lo que provoca el movimiento del robot móvil desde su punto de partida hasta su meta es la fuerza resultante que se genera a partir de la suma vectorial entre la fuerza repulsiva (suma total de las fuerzas de repulsión ocasionadas por los obstáculos cerca del robot) y la fuerza atractiva (asociada a la meta y al robot móvil), es decir el método sigue una técnica reactiva de navegación, ya que el movimiento del robot se encuentra guiado por el gradiente del campo potencial artificial generado y el proceso de generación de la trayectoria consiste en hallar un camino que mantenga al dispositivo móvil lo más alejado posible de los obstáculos que se encuentren en la zona [23, 26].

El esquema de control basado en campos potenciales es una gran herramienta para la navegación autónoma, sin embargo, tiene algunas desventajas como el uso de un único punto de atracción, ya que puede llevar a situaciones de trampa o atascamiento en las que el método no es capaz de producir la fuerza resultante necesaria para evitar grandes obstáculos. Es por ello que se han presentado enfoques donde realizan mejoras al esquema original de campos potenciales, como el de Campos de Potencial Adaptativo (APF) que se basa en campos de potencial artificiales mejorados y un algoritmo genético que utiliza múltiples puntos auxiliares de atracción para permitir que el robot evite obstáculos grandes o poco espaciados [27]. Existe la estrategia de Campo Potencial Artificial Discreto (DAPF) que ofrece la construcción de un campo de potencial discreto para encontrar una trayectoria más suave y corta, libre de colisiones para entornos estáticos y dinámicos la cual se ejecuta en tiempo casi real [28]. De igual forma se hace uso en controles híbridos como el enfoque conformado por campos de potencial artificiales y la lógica

difusa donde se explotan las ventajas de ambas metodologías [29].

Actualmente, una de las propuestas más atractivas de control para la navegación autónoma de robots móviles es el uso de un “cerebro artificial”, el cual está formado por redes de neuronas. Los candidatos para estos cerebros artificiales son las redes de neuronas tipo spiking, interconectadas entre sí, y cuya respuesta en los potenciales de membrana de cada una de ellas es del tipo spiking, imitando de esa forma la arquitectura de un cerebro biológico [14, 15].

1.3. Planteamiento del problema

El problema que se aborda en este trabajo de tesis está dividido en dos partes. Por un lado, se busca desarrollar una estrategia de evasión de obstáculos que permita a un robot diferencial navegar de manera autónoma en un ambiente desconocido. Por otra parte, se pretende desarrollar un control híbrido que, en la ausencia de obstáculos, garantice que el robot móvil siga una trayectoria deseada y que, en la presencia de obstáculos, sea capaz de realizar acciones evasivas de manera autónoma y retomar la trayectoria deseada.

En la literatura, existen algunos trabajos en los que se aborda este problema. Sin embargo, la aportación de este trabajo es que la estrategia de evasión de obstáculos estará basada en un arreglo de modelos neuronales acoplados y sincronizados.

1.4. Hipótesis

Es posible desarrollar una estrategia de evasión de obstáculos para un robot diferencial utilizando un arreglo de neuronas Hindmarsh-Rose acopladas y excitadas por una corriente externa, la cual puede ser generada a partir de las mediciones obtenidas de un sensor tipo Lidar.

1.5. Objetivos

1.5.1. Objetivo general

Diseñar e implementar un control basado en neuronas tipo Hindmarsh-Rose (HR) que permita la navegación autónoma de un robot móvil tipo (2,0) en un ambiente desconocido.

1.5.2. Objetivos particulares

Los objetivos particulares se enlistan a continuación.

- Realizar un estudio del modelo neuronal tipo Hindmarsh-Rose y diferentes tipos de acoplamientos.
- Establecer una estrategia de evasión lateral mediante un acoplamiento difuso entre las neuronas HR.
- Establecer una estrategia de evasión frontal a través de un acoplamiento dinámico entre las neuronas HR.
- Implementar un control de seguimiento de trayectoria.
- Implementar un control híbrido formado por las estrategias de seguimiento de trayectoria y evasión.

1.6. Productos generados

Se escribió el artículo titulado: “Navegación autónoma de un robot móvil en un ambiente desconocido usando una red de neuronas tipo Hindmarsh-Rose”, el cual se presentó de forma virtual en el XXIII Congreso Mexicano de Robótica COMROB 2021, celebrado de manera virtual en Tijuana, México, el día 29 de octubre del 2021. Obteniendo el reconocimiento del premio Rafael Kelly como el mejor artículo presentado.

Se escribió el artículo titulado: “Autonomous navigation of a mobile robot using a network of Hindmarsh-Rose (HR) neurons”, el cual se presentó de forma virtual, en la 18^a Conferencia Internacional en Ingeniería Eléctrica, Informática y Control Automático 2021 (2021 18TH International Conference on Electrical Engineering, Computing Science and Automatic Control CCE), celebrada de manera virtual en la Ciudad de México, México el día 12 de noviembre del 2021.

En la sección de anexos se proporciona una copia de estos artículos. En ambos se muestra el desarrollo del controlador híbrido y se ilustra su desempeño con ayuda de la simulación 3 tratada en el capítulo 4 de este trabajo. La estructura de los capítulos 2 y 3 de este trabajo de tesis se basan en estos artículos.

1.7. Organización de la tesis

El presente trabajo, consta de cinco capítulos. El Capítulo 2 expone el marco teórico y la teoría necesaria para alcanzar los objetivos del trabajo. En el Capítulo 3 se muestra el desarrollo del proyecto, en el cual se establece la estrategia de evasión y el control híbrido. El Capítulo 4 muestra los resultados de simulación numérica, así como una discusión de los mismos. Finalmente, en el Capítulo 5, se presentan las conclusiones y trabajo a futuro.

Capítulo 2

Marco teórico

Dentro del marco teórico se exponen las herramientas a utilizar en este trabajo, tales como, diferentes modelos neuronales que pueden ser candidatos para esta aplicación, algunos de los acoplamientos que se suelen utilizar para interconectar neuronas, y se proporciona una definición formal de sincronización de sistemas. Por otra parte, se muestra el modelo cinemático de un robot móvil tipo (2,0) y se describe la estrategia del seguimiento de trayectoria empleada en este proyecto.

2.1. Modelos neuronales

En la literatura, se han reportado varias propuestas para modelar diferentes comportamientos de una neurona, es decir, el catálogo de modelos matemáticos para una neurona es bastante amplio. Debido a ello, en este apartado solo se presentan los modelos neuronales clásicos que se encargan de generar potenciales de membrana del tipo spiking y bursting, ya que estos se relacionan con el contexto del proyecto.

2.1.1. Modelo Hodgkin-Huxley

Con base en los trabajos obtenidos por McCulloch (Neurofisiólogo) y Walter Pitts (matemático), además del desarrollo de las técnicas de fijación o control de voltaje que mantuvo Kenneth S. Cole (físico que estudió las propiedades de las membranas), en 1952 los neurofisiólogos Alan Lloyd Hodgkin y Andrew Fielding Huxley diseñaron un modelo matemático en el que se basan algunos de los modelos realistas actuales, este modelo neuronal está descrito por las de Ecs. (2.1) [9, 10, 12, 13].

$$\begin{aligned}
 C\dot{v} &= -\bar{g}_{Na}m^3h(v - E_{Na}) - \bar{g}_kn^4(v - E_k) - \bar{g}_L(v - E_L) + I_{ext}(t), \\
 \dot{m} &= \alpha_m(v)(1 - m) - \beta_m(v)m, \\
 \dot{n} &= \alpha_n(v)(1 - n) - \beta_n(v)n, \\
 \dot{h} &= \alpha_h(v)(1 - h) - \beta_h(v)h,
 \end{aligned} \tag{2.1}$$

con parámetros

$$\begin{aligned}
 E_{Na} &= 115[mV], & E_k &= -12[mV], & E_L &= 10.6[mV], \\
 \bar{g}_{Na} &= 120[ms/cm^2], & \bar{g}_k &= 36[ms/cm^2], & \bar{g}_L &= 0.3[ms/cm^2], \\
 \alpha_n(v) &= \frac{0.1 - 0.01v}{e^{1-0.1v} - 1}, & \alpha_m(v) &= \frac{2.5 - 0.01v}{e^{2.5-0.1v} - 1}, & \alpha_h(v) &= 0.07e^{\left(-\frac{v}{20}\right)}, \\
 \beta_n(v) &= 0.125e^{\left(-\frac{v}{80}\right)}, & \beta_m(v) &= 4e^{\left(-\frac{v}{18}\right)}, & \beta_h(v) &= \frac{1}{e^{3-0.1v} + 1},
 \end{aligned}$$

donde v es el potencial de membrana, C es la capacitancia de la membrana por unidad de área. En general n , m , y h son las variables de conductancia y toman valores entre 0 y 1, n se refiere a las moléculas de potasio, así como, m y h se refieren a las moléculas de sodio. \bar{g}_{Na} , \bar{g}_k y \bar{g}_L representan las conductancias de sodio, potasio y de todos aquellos residuos no pertenecientes a los iones anteriores, respectivamente. La velocidad del cambio de estado de cierre a apertura, o viceversa, de las puertas se representa mediante las α_i y β_i , $i = n, m, h$, respectivamente. I_{ext} representa la corriente externa aplicada a la neurona. Los valores de los parámetros que fueron seleccionados por Hodgkin y Huxley, les permitieron reproducir en simulaciones numéricas, los valores del potencial de membrana medidos en el axón de un calamar gigante, que corresponden a un comportamiento tipo spiking, ver Figura 2.1.

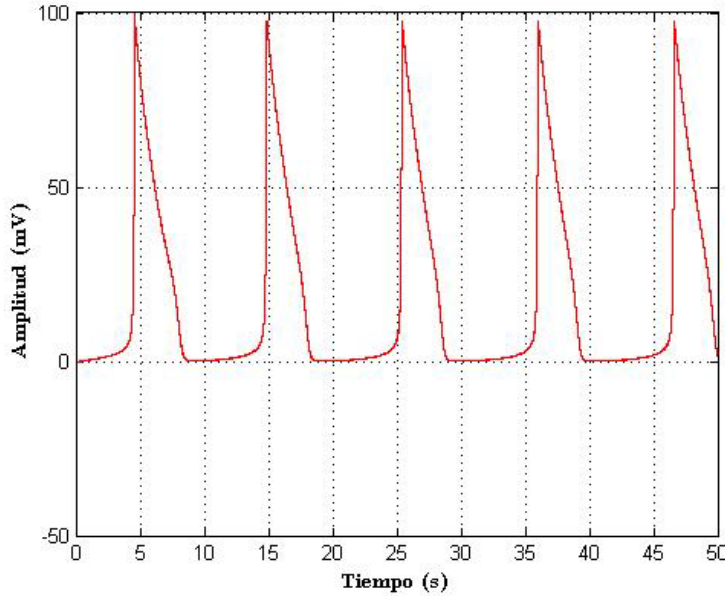


Figura 2.1: Potencial de membrana con actividad tipo spiking, Hodgkin y Huxley.

2.1.2. FitzHugh-Nagumo

En 1961 Richard FitzHugh propuso una simplificación de 2 dimensiones al modelo de Hodgkin-Huxley, llamándolo modelo de Bonhoeffer-van der Pol. Dicho modelo fue implementado mediante un circuito electrónico diseñado por el ingeniero japonés Jin-Ichi Nagumo (Figura 2.2), pasando a llamarse el modelo FitzHugh-Nagumo representado por el conjunto de Ecs. (2.2) [9, 11, 13]

$$\begin{aligned}\dot{v} &= v - \frac{v^3}{3} - w + I_{ext}, \\ \dot{w} &= a(bv + d - cw),\end{aligned}\tag{2.2}$$

donde v representa el potencial de membrana, el cual está en función del tiempo y w es una variable de recuperación lenta esencial para la repolarización (o hiperpolarización). I_{ext} representa la corriente externa de excitación a la neurona. Los parámetros a, b, c, d son parámetros adimensionales. La constante a suele ser muy pequeña ($a \ll 1$), haciendo que la relajación de v sea muy rápida. Para lograr resultados similares a los del modelo Hodgkin-Huxley, se tomaron los siguientes valores para los parámetros: $a = 0.08, b = 1, c = 0.8, d = 0.7$.

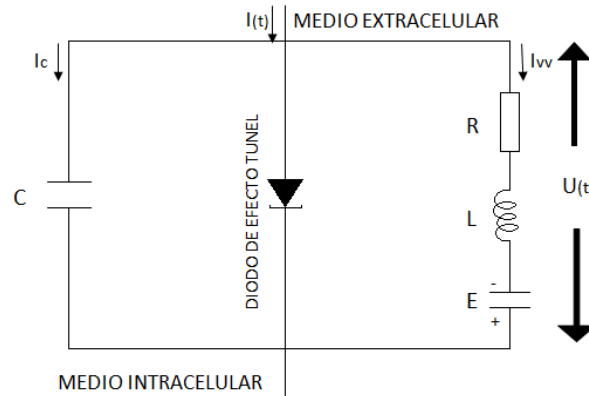


Figura 2.2: Circuito electrónico Jin-Ichi Nagumo.

Como resultado, se obtuvo un modelo simplificado compuesto por 2 ecuaciones (una lineal y otra no lineal), del modelo propuesto por Hodgkin-Huxley que se componía de 4 ecuaciones, en el que se eliminan las variables m y n , ver Ecs. (2.1), que modelan la apertura de los canales iónicos del sodio y potasio. En la Figura 2.3 se puede apreciar el potencial obtenido en simulaciones numéricas del modelo FitzHugh-Nagumo.

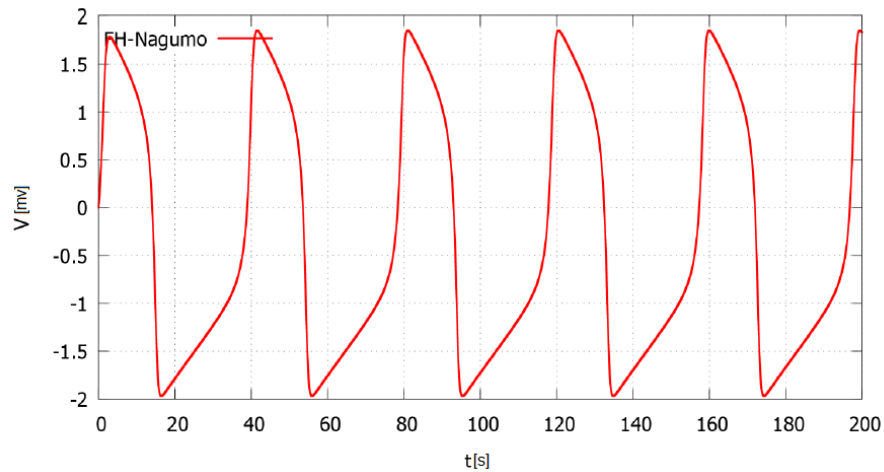


Figura 2.3: Potencial de membrana con actividad tipo spiking, FitzHugh y Nagumo.

2.1.3. Modelo Morris-Lecar

Cathy Morris y Harold Lecar, en el año 1981, desarrollaron un modelo matemático de una neurona biológica el cual también se considera una reducción del modelo de Hodgkin-Huxley. Este sistema de ecuaciones describe de forma simplificada la relación entre la activación de los canales iónicos y el potencial de membrana, el potencial depende de la actividad de los canales iónicos, y la actividad de los canales de iones depende de la tensión. El modelo fue desarrollado originalmente para describir la fibra muscular gigante del percebe, pero se ha usado para describir otros sistemas, tales como neuronas del ganglio estomatogástrico de la langosta y las neuronas sensoriales espinales de mamíferos. El modelo neuronal se presenta en el conjunto de Ecs. (2.3) [9, 13, 30]

$$\begin{aligned}
 C\dot{V} &= I - g_L V_L - g_{Ca} M(V - V_{Ca}) - g_K N(V - V_K), \\
 \dot{M} &= \lambda_M(V)(M_\infty(V) - M), \\
 \dot{N} &= \lambda_N(V)(N_\infty(V) - N),
 \end{aligned} \tag{2.3}$$

$$\begin{aligned}
 M_\infty(V) &= \frac{1}{2} \left(1 + \tanh \left(\frac{V - V_1}{V_2} \right) \right), \\
 N_\infty(V) &= \frac{1}{2} \left(1 + \tanh \left(\frac{V - V_3}{V_4} \right) \right), \\
 \lambda_M(V) &= \bar{\lambda}_M \cosh \left(\frac{V - V_1}{2V_2} \right), \\
 \lambda_N(V) &= \bar{\lambda}_N \cosh \left(\frac{V - V_3}{2V_4} \right),
 \end{aligned}$$

donde:

V = Potencial de membrana [mv].

C = Capacitancia de la membrana [F/cm^2].

I = Es la corriente aplicada con unidades [$\mu A/cm^2$].

g_L, g_{Ca}, g_K = Conductancia máxima o instantánea para los caminos de fuga, Ca^{++} y K^+ , respectivamente [$mmho/cm^2$].

V_L, V_{Ca}, V_K = Potencial de equilibrio correspondiente a las conductancias de fuga, Ca^{++} y K^+ respectivamente [mv].

M = Porción de canales de Ca^{++} abiertos.

CAPÍTULO 2. MARCO TEÓRICO

N = Porción de canales de K^+ abiertos.

$M_\infty(V), N_\infty(V)$ = Porción de canales abiertos de Ca^{++} y K^+ , en estado estacionario.

$\lambda_M(V), \lambda_N(V)$ = Tasa constante para abrir los canales de Ca^{++} y K^+ [s^{-1}].

$\bar{\lambda}_M, \bar{\lambda}_N$ = Máxima tasa constante para abrir los canales de Ca^{++} y K^+ [s^{-1}].

V_1 = Potencial al que $M_\infty = 0.5$ [mv].

V_2 = Recíproco de la pendiente de la dependencia de voltaje de M_∞ [mv].

V_3 = Potencial al que $N_\infty = 0.5$ [mv].

V_4 = Recíproco de la pendiente de la dependencia de voltaje de N_∞ [mv].

La característica más interesante en el comportamiento de la fibra muscular del percebe es la oscilación del voltaje, que ocurre solo cuando los sistemas de Ca^{++} y K^+ están operando al mismo tiempo. En este sistema reducido, solo se utilizan la primera y la tercera ecuación del sistema original. El resultado del potencial de membrana obtenido a través de simulaciones numéricas puede apreciarse en la Figura 2.4.

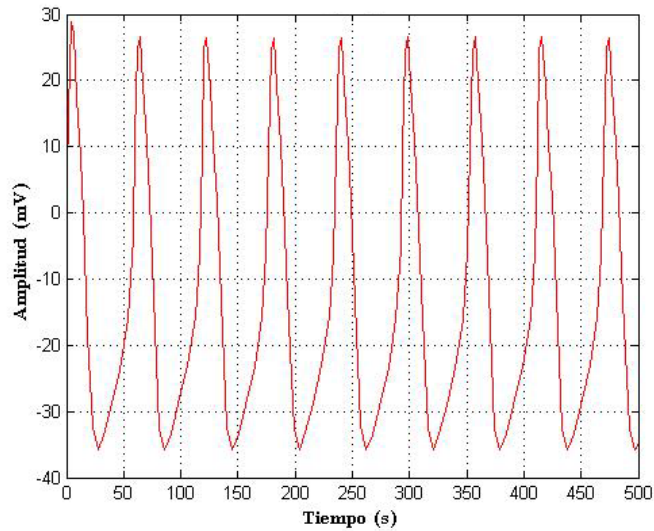


Figura 2.4: Potencial de membrana con actividad tipo spiking, modelo Morris y Lecar.

2.1.4. Modelo Hindmarsh-Rose

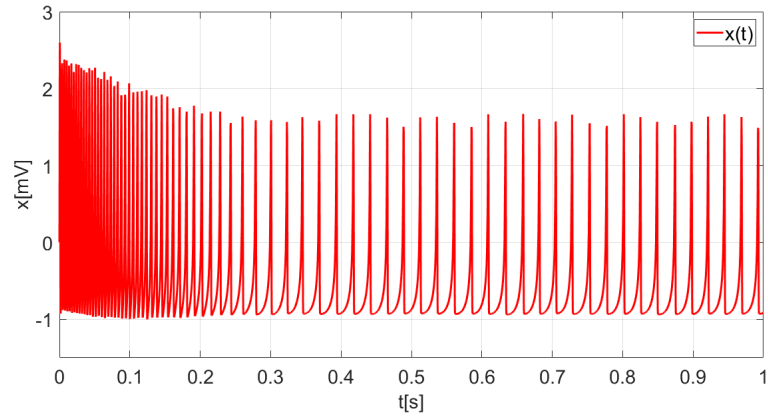
En 1982, L. Hindmarsh y R.M. Rose modificaron el modelo FitzHugh Nagumo, reemplazando la función lineal por una cuadrática dotando al modelo de la capacidad de describir de forma más realista los potenciales de

membrana en intervalos de tiempo cortos, el cual se considero como su primer modelo. Sin embargo, en 1984 presentaron un modelo mejorado, en el que se incorpora una tercera ecuación, dando lugar a un sistema de ecuaciones diferenciales no lineales tridimensional, el cual es capaz de simular casi por completo el comportamiento neuronal generado por el modelo Hodgkin-Huxley, pero con una estructura más sencilla. permitiendo de esta manera reproducir varios comportamientos del tipo spiking, bursting, y comportamiento bursting caótico [9, 11, 12, 13, 14, 15, 30, 31]. El modelo esta descrito por:

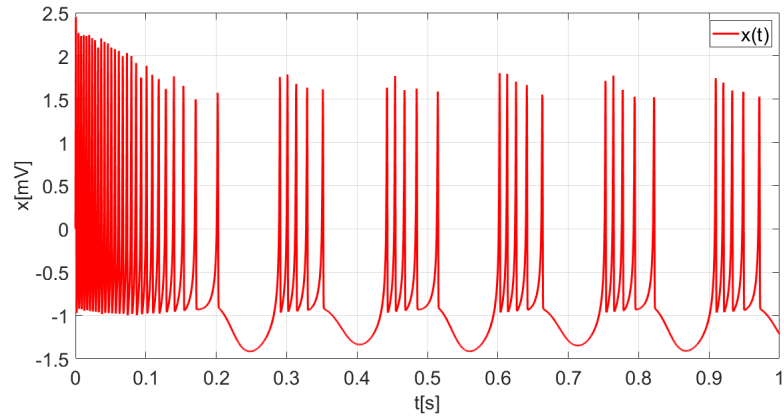
$$\begin{aligned}\dot{x} &= y + bx^2 - ax^3 - z + I, \\ \dot{y} &= c - dx^2 - y, \\ \dot{z} &= r[s(x - x_0) - z],\end{aligned}\tag{2.4}$$

donde $x \in R$ es el potencial de membrana, $y \in R$ es la corriente de recuperación que modela el transporte de iones de potasio y sodio a través de canales de iones rápidos, $z \in R$ es la corriente de adaptación que modela el transporte de otros iones a través de canales lentos. Los parámetros $a, b, c, d, s, r, x_0 \in R$ son constantes. $a \in R$ permite conmutar entre comportamientos de bursting y spiking, además determina la frecuencia del spiking. Por otra parte $r \in R$ gobierna el comportamiento bursting, $s \in R$ gobierna la adaptación, si este posee valores pequeños (alrededor de 1) provoca un comportamiento spiking rápido. $x_0 \in R$ es el potencial de reposo del sistema que sirve como potencial de referencia, y $b, d, c \in R^+$ son parámetros positivos. Finalmente $I \in R$ es la corriente externa (entrada del modelo). Al tener una correcta sintonización o configuración de los parámetros, se puede variar el valor de la corriente externa, de tal manera que, entre mayor sea la corriente, mayor será la frecuencia de oscilación en el potencial de la membrana. De igual forma, la variación de la corriente externa produce diferentes tipos de actividad o modos como los ya mencionados spiking, bursting, y bursting caótico.

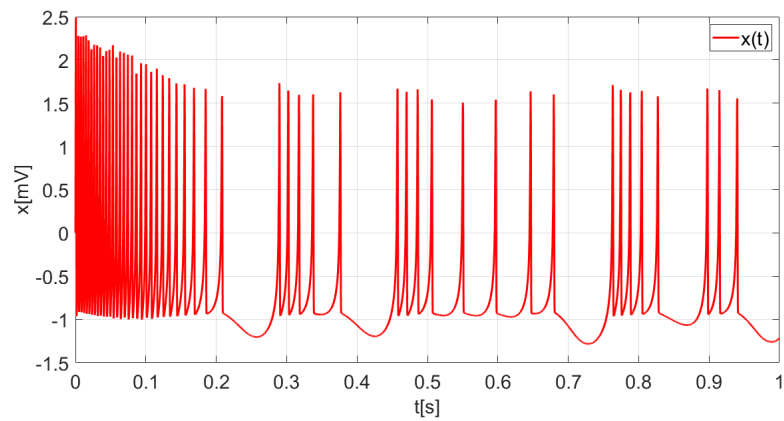
La Figura 2.5 muestra el potencial de membrana para una sintonización de valores típicos de los parámetros, $a = 1, b = 3, c = 1, d = 5, s = 4, x_0 = -1.6, r = 0.005$, con tres diferentes valores de la corriente externa I , lo cual reproduce los típicos modos spiking, bursting, y bursting caótico.



(a) Actividad o modo spiking, $I = 3.8$.



(b) Actividad o modo bursting, $I = 3$.



(c) Actividad o modo bursting caótico, $I = 3.281$.

Figura 2.5: Actividades en el potencial de membrana del modelo HR, con la misma sintonización de parámetros y diferentes corrientes externas.

2.1.5. Modelo Izhikevich

En 2003, Eugene Izhikevich propuso un modelo de neurona pulsante simplificado, que basa su comportamiento en un sistema de 2 ecuaciones diferenciales ordinarias y presenta actividades del tipo spiking y bursting. El modelo está dado por el siguiente conjunto de ecuaciones [9, 11, 13].

$$\begin{aligned}\dot{v} &= \alpha v^2 + \beta v + \gamma - u + I, \\ \dot{u} &= a(bv - u),\end{aligned}\tag{2.5}$$

con los parámetros fijados en los siguientes valores,

$$\begin{aligned}\alpha &= 0.04[mv^{-1}ms^{-1}], \\ \beta &= 5[ms^{-1}], \\ \gamma &= 140[mvms^{-1}],\end{aligned}$$

y condiciones iniciales v_0 y u_0 .

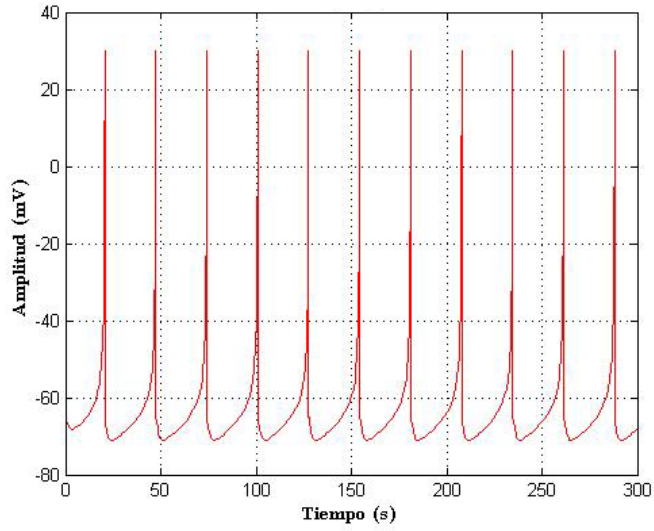
La variable v representa el potencial de membrana de la neurona y u representa una variable de ajuste para la repolarización de la membrana, ambas tienen dimensión de voltaje. I representa la corriente de entrada, a , c y d son parámetros dimensionales, y b un parámetro adimensional. Cuando el potencial alcanza su punto máximo (30mv), tanto el potencial de membrana como la variable de recuperación son reiniciados de acuerdo con lo siguiente

$$\text{si } v \geq 30mv, \text{ entonces } \begin{cases} v = c, \\ u = u + d. \end{cases}\tag{2.6}$$

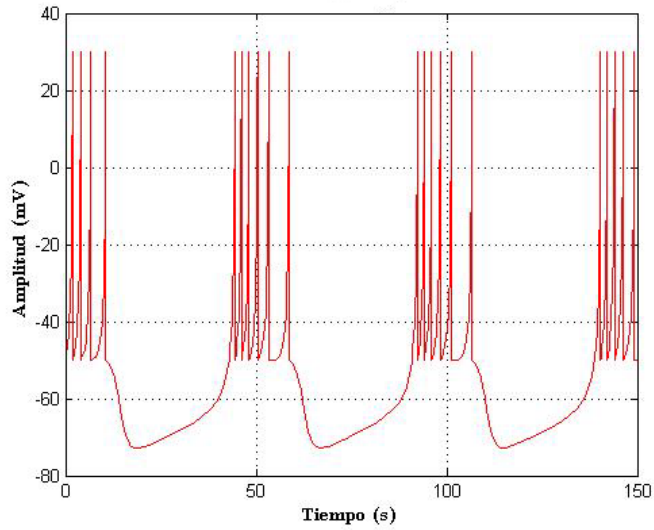
El potencial en reposo (o potencial de umbral) de este modelo oscila entre los $-70[mv]$ y $-60[mv]$ dependiendo del valor de b . El parámetro a representa la escala de tiempo de la variable u . Cuanto más bajo sea el valor, más lenta será la recuperación. Un valor típico es $a = 0.02[ms^{-1}]$. El parámetro b representa la sensibilidad de la variable u a las fluctuaciones del potencial de membrana v por debajo del umbral. Cuanto mayores sean u y v las posibles oscilaciones surgirán con más fuerza. Un valor típico es $b = 0.2$. El parámetro c representa el valor de reajuste después del reset para la variable del potencial de membrana v . Un valor típico es $c = -65[mv]$. El parámetro d representa el valor de reajuste pos-pico para la variable de recuperación u . Un valor típico es $d = 2[mv]$. Los valores anteriores de los parámetros se consideran típicos ya que al fijarlos de esa manera y al variar la entrada I se pueden obtener comportamientos en el potencial de membrana cercanos a

CAPÍTULO 2. MARCO TEÓRICO

los de una neurona natural, los cuales son del tipo spiking y bursting, como se muestra en la Figura 2.6.



(a) Actividad o modo spiking.



(b) Actividad o modo bursting.

Figura 2.6: Actividades en el potencial de membrana del modelo Izhikevich.

2.1.6. Discusión de los modelos neuronales

Después de consultar en la literatura sobre modelos neuronales y observar los resultados que se han obtenido en otros trabajos, se propone utilizar el modelo neuronal Hindmarsh-Rose (HR), puesto que produce una respuesta en el potencial de membrana adecuada para la idea que se tiene como control de evasión, debido a que, es de los más completos produciendo diferentes comportamientos dinámicos tales como spiking y bursting (los cuales se pueden asociar con la estrategia de evasión). Se considera un modelo neuronal rápido (es decir, produce una respuesta del potencial de membrana en menos de 1[s]), su aplicación en otros trabajos ha logrado un desempeño satisfactorio [9, 10, 12, 13, 14, 15], además, su implementación en electrónica analógica es relativamente simple.

2.2. Acoplamiento de neuronas

Al presentarse diferentes funciones cognitivas y procesos neurológicos, es común hallar una interconexión entre un grupo de neuronas (sinapsis) donde sus respuestas están sincronizadas. Para que dos o más neuronas se interconecten debe haber un canal, llamado acoplamiento, del cual se sabe que existen dos tipos: acoplamiento por sinapsis químicas o eléctricas.

Existen varios esquemas de acoplamiento para modelar las sinapsis químicas y eléctricas, entre ellos, los esquemas de acoplamiento unidireccional [32], acoplamiento difuso [14, 15], y acoplamiento dinámico [13], por mencionar algunos. Debido a que este trabajo va orientado con la aplicación del modelo neuronal Hindmarsh-Rose (HR), se presentan estos acoplamientos utilizando este tipo de neuronas HR.

2.2.1. Acoplamiento unidireccional

El método surgió de la idea de sincronizar dos sistemas caóticos idénticos con señales conductoras comunes y fue introducido por Pecora y Carroll en 1990 y se basa en vincular una trayectoria del sistema maestro con los mismos valores del esclavo. Los dos sistemas en configuración maestro-esclavo pueden sincronizarse vinculando la trayectoria x del sistema maestro a las ecuaciones y y z del sistema esclavo [32].

Con la suposición de dos neuronas HR idénticas (con la misma sintonización de parámetros), e interconectadas a través de un acoplamiento unidireccional como se muestra en la Figura 2.7, se establece que la dinámica de

todo el sistema de estados se rige por las Ecs.(2.7) y (2.8).

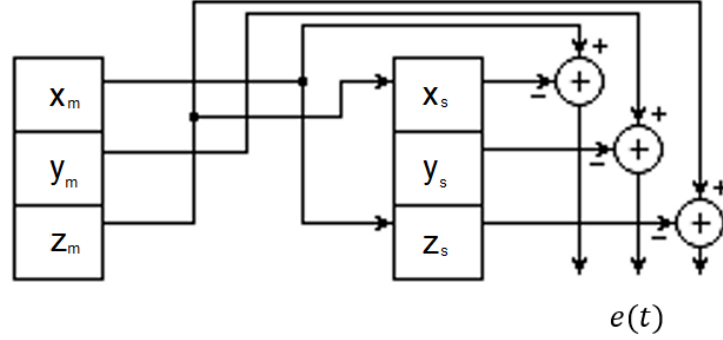


Figura 2.7: Acoplamiento unidireccional entre dos neuronas HR.

$$\begin{aligned}\dot{x}_m &= y_m + bx_m^2 - ax_m^3 - z_m + I_m, \\ \dot{y}_m &= c - dx_m^2 - y_m, \\ \dot{z}_m &= r[s(x_m - x_0) - z_m],\end{aligned}\tag{2.7}$$

$$\begin{aligned}\dot{x}_s &= y_s + bx_s^2 - ax_s^3 - z_m + I_s, \\ \dot{y}_s &= c - dx_s^2 - y_s, \\ \dot{z}_s &= r[s(x_m - x_0) - z_s],\end{aligned}\tag{2.8}$$

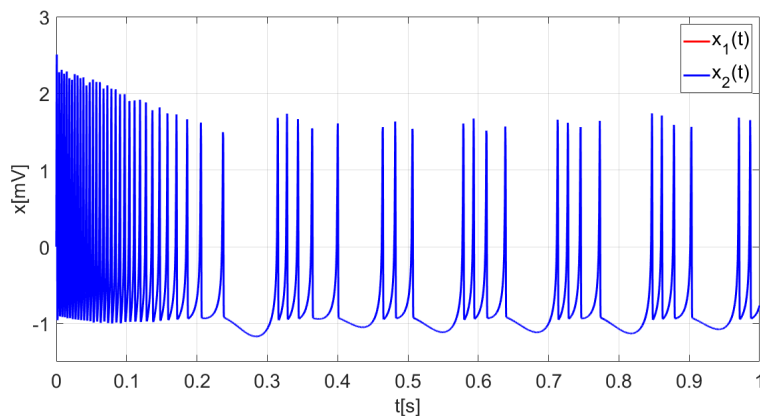
Los sistemas (2.7) (sistema-neurona maestro) y (2.8) (sistema-neurona esclavo), están vinculados mediante el mismo valor de $x_m(t)$ en $z_m(t)$ y $z_s(t)$, y a su vez están vinculados por el mismo valor de $z_m(t)$ en los subsistemas $x_m(t)$ y $x_s(t)$. El error de sincronización está dado por las diferencias: $X = x_m - x_s$, $Y = y_m - y_s$, $Z = z_m - z_s$, debido a ello, la dinámica del error se expresa como: $\dot{X} = \dot{x}_m - \dot{x}_s$, $\dot{Y} = \dot{y}_m - \dot{y}_s$, $\dot{Z} = \dot{z}_m - \dot{z}_s$

$$\begin{aligned}\dot{X} &= Y + b(x_m^2 - x_s^2) - a(x_m^3 - x_s^3), \\ \dot{Y} &= c - d(x_m^2 - x_s^2) - Y, \\ \dot{Z} &= -rZ,\end{aligned}\tag{2.9}$$

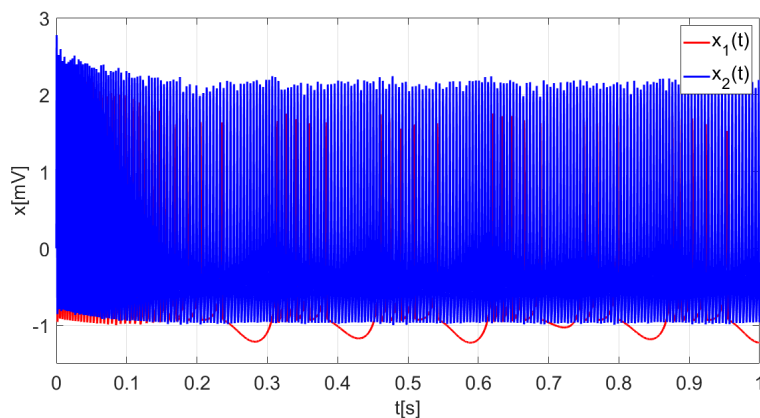
En la Figura 2.8a se puede apreciar una simulación numérica de la sincronización utilizando el acoplamiento unidireccional ante una misma entrada de corriente externa $I_m = I_s = I = 3.28$ para el par de neuronas HR. Por otra parte, se muestra en la Figura 2.8b la misma simulación

2.2. ACOPLAMIENTO DE NEURONAS

numérica pero con entradas de corriente externa diferentes para cada neurona, es decir, $I_m = 3.28$ e $I_s = 5$. Se utilizan los siguientes parámetros: $a = 1, b = 3, c = 1, d = 5, s = 4, x_0 = -1.6, r = 0.005$.



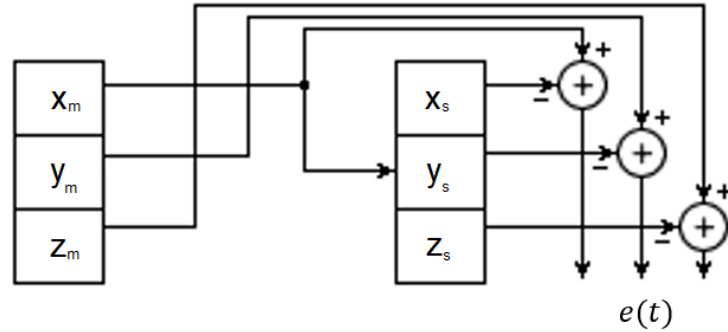
(a) Sincronización de dos neuronas HR (2.7)-(2.8) mediante acoplamiento unidireccional y entradas $I_m = I_s = I = 3.28$.



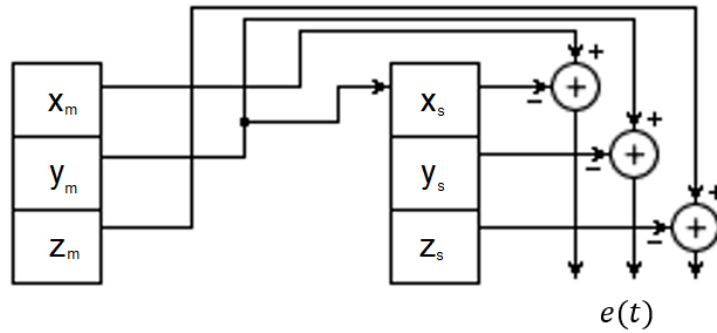
(b) Pérdida de sincronización de dos neuronas HR (2.7)-(2.8) mediante acoplamiento unidireccional y entradas $I_m = 3.28, I_s = 5$.

Figura 2.8: Potenciales de membrana de dos neuronas HR (2.7)-(2.8) interconectadas con acoplamiento unidireccional.

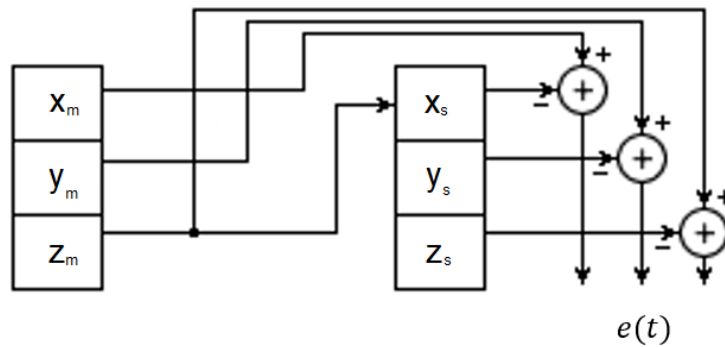
Cabe mencionar que existen otros acoplamientos unidireccionales posibles, que muestran una sincronización instantánea del sistema completo con un solo vínculo, como se muestra en la Figura 2.9.



(a) Acoplamiento unidireccional con vínculo mediante el mismo valor de $x_m(t)$ en $y_m(t)$ y $y_s(t)$.



(b) Acoplamiento unidireccional con vínculo mediante el mismo valor de $y_m(t)$ en $x_m(t)$ y $x_s(t)$.



(c) Acoplamiento unidireccional con vínculo mediante el mismo valor de $z_m(t)$ en $x_m(t)$ y $x_s(t)$.

Figura 2.9: Acoplamientos unidireccionales que muestran la sincronización del sistema completo.

2.2.2. Acoplamiento difuso

Un par de neuronas HR idénticas e interconectadas mediante un acoplamiento difuso, véase la Figura 2.10, está descrito por las siguientes ecuaciones [13, 14, 15]

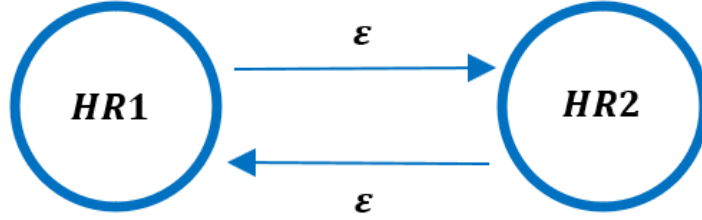


Figura 2.10: Acoplamiento difuso entre dos neuronas HR.

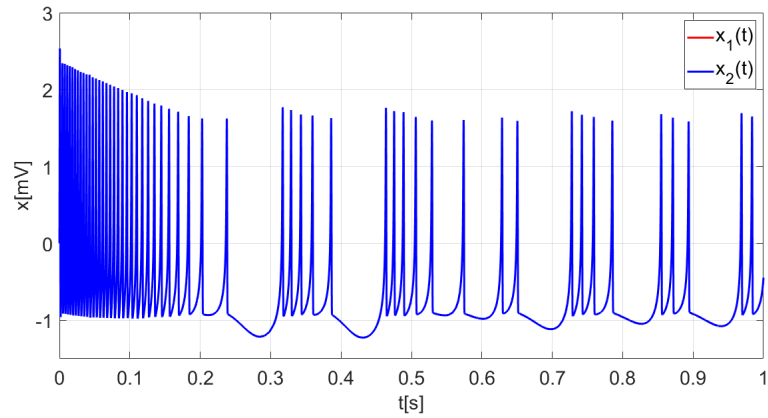
$$HR1 : \begin{cases} \dot{x}_1 = y_1 + bx_1^2 - ax_1^3 - z_1 + I_1 + v_1, \\ \dot{y}_1 = c - dx_1^2 - y_1, \\ \dot{z}_1 = r[s(x_1 - x_0) - z_1], \end{cases} \quad (2.10)$$

$$HR2 : \begin{cases} \dot{x}_2 = y_2 + bx_2^2 - ax_2^3 - z_2 + I_2 + v_2, \\ \dot{y}_2 = c - dx_2^2 - y_2, \\ \dot{z}_2 = r[s(x_2 - x_0) - z_2], \end{cases} \quad (2.11)$$

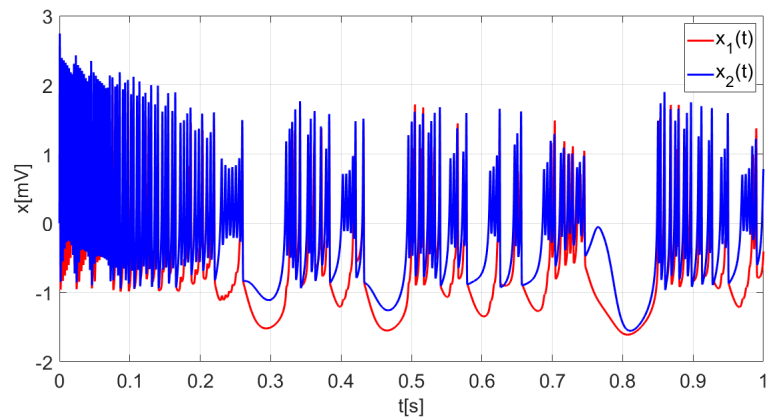
$$\begin{cases} v_1 = \varepsilon(x_2 - x_1), \\ v_2 = \varepsilon(x_1 - x_2). \end{cases} \quad (2.12)$$

La sincronización de los sistemas-neuronas HR1 (Ecs.(2.10)) y HR2 (Ecs.(2.11)) depende ampliamente de la fuerza de acoplamiento $\varepsilon > 0$. Con una fuerza de acoplamiento lo suficientemente grande, las neuronas HR1 y HR2 pueden sincronizarse aun cuando tienen entradas de corriente externa I ligeramente diferentes, es decir, $I_1 \approx I_2$.

En la Figura 2.11a se muestra el resultado de una simulación numérica del par de neuronas HR (2.10)-(2.11) con un acoplamiento difuso (2.12), cuya fuerza de acoplamiento es $\varepsilon = 0.5$, y sus entradas de corriente son iguales, es decir, $I_1 = I_2 = I = 3.28$. Por otra parte, en la Figura 2.11b se presenta la misma simulación numérica pero con entradas de corriente diferentes ($I_1 = 3.28, I_2 = 5$). Además, se utiliza una sintonización que corresponde con los valores: $a = 1, b = 3, c = 1, d = 5, s = 4, x_0 = -1.6, r = 0.005$.



(a) Sincronización de dos neuronas HR (2.10)-(2.11) mediante acoplamiento difuso (2.12) con $\varepsilon = 0.5$ y entradas $I_1 = I_2 = I = 3.28$.



(b) Pérdida de sincronización de dos neuronas HR (2.10)-(2.11) con acoplamiento difuso (2.12) con $\varepsilon = 0.5$ y entradas $I_1 = 3.28, I_2 = 5$.

Figura 2.11: Potenciales de membrana de dos neuronas HR (2.10)-(2.11) interconectadas con acoplamiento difuso (2.12).

Cabe mencionar que si las señales de corriente externa son diferentes ($I_1 \neq I_2$), entonces la sincronización se pierde. Por otra parte, en este trabajo, la fuerza de acoplamiento ε se escoge de tal manera que la sincronización se pierde ante cambios mínimos de las corrientes I_1 e I_2 .

2.2.3. Acoplamiento dinámico

Considere dos neuronas HR idénticas e interconectadas mediante un acoplamiento dinámico, descritas por [14, 15]

$$HR1 : \begin{cases} \dot{x}_1 &= y_1 + bx_1^2 - ax_1^3 - z_1 + I_1 + h_2, \\ \dot{y}_1 &= c - dx_1^2 - y_1, \\ \dot{z}_1 &= r[s(x_1 - x_0) - z_1], \end{cases} \quad (2.13)$$

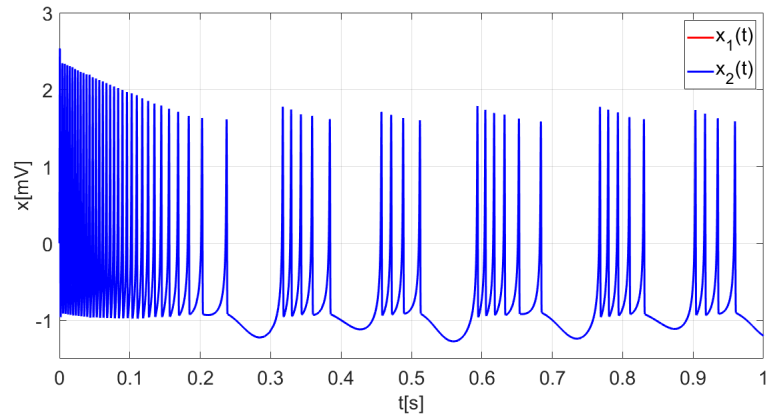
$$HR2 : \begin{cases} \dot{x}_2 &= y_2 + bx_2^2 - ax_2^3 - z_2 + I_2 - Gh_2, \\ \dot{y}_2 &= c - dx_2^2 - y_2, \\ \dot{z}_2 &= r[s(x_2 - x_0) - z_2], \end{cases} \quad (2.14)$$

$$\begin{cases} \dot{h}_1 &= -\alpha h_1 + h_2, \\ \dot{h}_2 &= -\gamma_1 h_1 - \gamma_2 h_2 - k(x_1 - x_2), \end{cases} \quad (2.15)$$

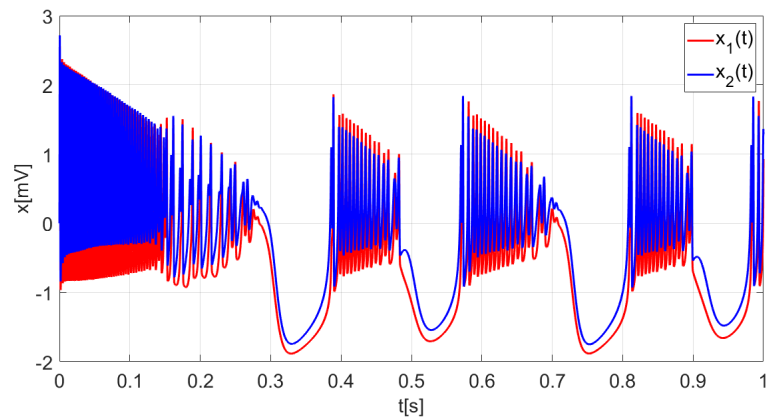
con fuerza de acoplamiento k y parámetros $\alpha, \gamma_1, \gamma_2$.

La sincronización de los sistemas-neuronas HR1 (Ecs.(2.13)) y HR2 (Ecs.(2.14)) posee una fuerza de acoplamiento k y parámetros $\alpha, \gamma_1, \gamma_2$. En comparación con los acoplamientos anteriores, este goza de una ventaja general que consiste en una mejor sincronización entre las neuronas HR1 y HR2 aun cuando tienen entradas de corriente externa I diferentes, es decir, $I_1 \neq I_2$.

En la Figura 2.12a se muestra el resultado de una simulación numérica acerca del par de neuronas HR con un acoplamiento dinámico, cuya fuerza de acoplamiento es $k = 10$, así como sus parámetros tienen un valor $\alpha = \gamma_1 = 1, \gamma_2 = 10$, y entradas de corriente iguales $I_1 = I_2 = I = 3.28$. Por otra parte, en la Figura 2.12b se presenta la misma simulación numérica pero con entradas de corriente diferentes ($I_1 = 3.28, I_2 = 5$). Además, se utiliza una sintonización con los valores: $a = 1, b = 3, c = 1, d = 5, s = 4, x_0 = -1.6, r = 0.005, G = 1$.



(a) Sincronización de dos neuronas HR (2.13)-(2.14) mediante acoplamiento dinámico (2.15) con $k = 10$ y entradas $I_1 = I_2 = I = 3.28$.



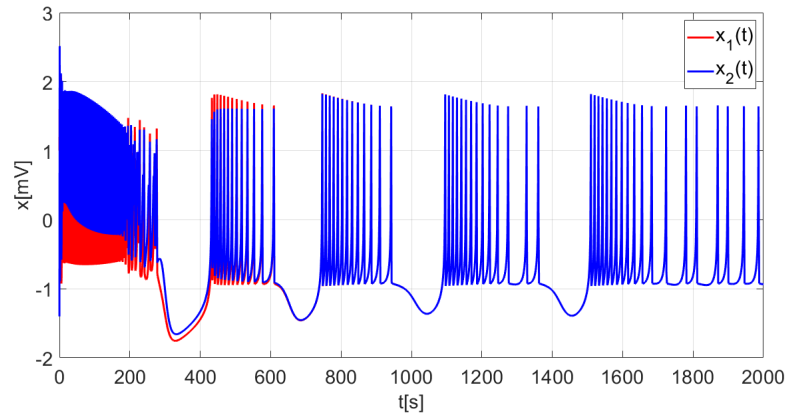
(b) Sincronización práctica de dos neuronas HR (2.13)-(2.14) mediante acoplamiento dinámico (2.15) con $k = 10$ y entradas $I_1 = 3.28, I_2 = 5$.

Figura 2.12: Potenciales de membrana de dos neuronas HR interconectadas con acoplamiento dinámico.

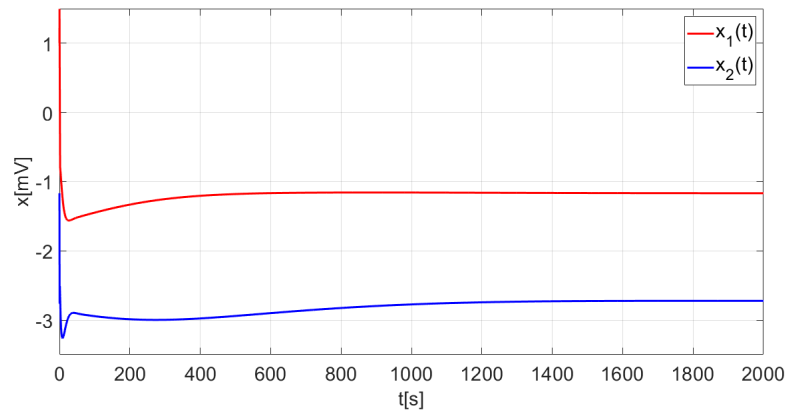
2.2.3.1. Estado de relajación completa

Además de una mejor sincronización entre neuronas, el acoplamiento dinámico puede producir un modo o actividad de relajación completa entre ambas neuronas. Para generar este modo, se debe contar con una condición inicial y una sintonización de los parámetros adecuadas, además, las entradas para el par de neuronas deben ser las mismas ($I_1 = I_2 = I$). Una vez cumplido este requisito, la variación del parámetro G en el sistema de Ecs. (2.14), permite pasar de un modo de oscilación (o spiking) a un modo de relajación completa para ambas neuronas, y viceversa. En [13], se menciona que con valores de $G \leq -10$, los potenciales de membrana de las neuronas operan en un estado de relajación completa, y para valores de $G > -10$, los potenciales de membrana de las neuronas operan en un estado de spiking. A continuación se presenta una simulación numérica que ilustra lo mencionado anteriormente. Para ello se consideran los siguientes parámetros: $a = 1, b = 3, c = 1, d = 5, s = 4, x_0 = -1.6, r = -0.0021, I_1 = I_2 = 3.281., \alpha = \gamma_1 = 1, k = \gamma_2 = 10$. Las condiciones iniciales son: $x_1(0) = 1, y_1(0) = 0.3, z_1(0) = 2., x_2(0) = -1.4, y_2(0) = -0.13, z_2(0) = -0.12., h_1(0) = 0, h_2(0) = 0$.

En la Figura 2.13a se puede apreciar la sincronización de las neuronas en un modo oscilatorio (bursting) que puede reproducir el acoplamiento dinámico (con un valor $G = 1$), así como en la Figura 2.13b se muestran los potenciales de membrana de las neuronas en un modo de relajación completa (con un valor $G = -10$).



(a) Potencial de membrana en modo oscilatorio de dos neuronas HR acopladas mediante acoplamiento dinámico, con $G = 1$.



(b) Potencial de membrana en modo relajación completa de dos neuronas HR acopladas mediante acoplamiento dinámico, con $G = -10$.

Figura 2.13: Potencial de membrana de dos neuronas HR acopladas mediante acoplamiento dinámico, misma entrada I .

2.2.4. Discusión de los acoplamientos

Al consultar los diferentes esquemas que proporcionan un acoplamiento entre neuronas, se opta por utilizar el acoplamiento difuso para interconectar un par de neuronas HR, ya que este proporciona una mejor respuesta que el acoplamiento unidireccional. También, se utilizará una fuerza de acoplamiento ε relativamente pequeña, ver (2.12), para garantizar que los sistemas acoplados pierden la sincronía ante mínimas variaciones en las corrientes de entrada de las neuronas HR. Por otro lado, se implementará un acoplamiento dinámico para interconectar otro par de neuronas HR y aprovechar el modo de relajación y de oscilación que este puede producir.

2.3. Sincronización de neuronas

Un problema de investigación que ha tomado un gran interés desde la perspectiva de dinámica y control es la sincronización de sistemas, es decir, cuando varios sistemas acoplados se comportan de la misma manera. Considere el sistema

$$\dot{x} = f(x, z), \quad (2.16)$$

$$\dot{y} = f(y, z), \quad (2.17)$$

$$\dot{z} = g(x, y, z), \quad (2.18)$$

donde $x, y \in R^n, z \in R^m$, son variables de estado y las funciones $f : R^n \times R^m \rightarrow R^n$ y $g : R^n \times R^n \times R^m \rightarrow R^m$ son localmente Lipschitz. Cabe notar que los subsistemas (2.16) y (2.17) describen sistemas idénticos que interactúan por medio de (2.18).

Además, nótese que (2.16) y (2.17) se pueden considerar como dos neuronas HR idénticas, y (2.18) es un acoplamiento dinámico para un par de neuronas (esta misma idea también se puede aplicar para un acoplamiento difuso).

2.3.1. Definición 1. Sincronización completa.

Se dice que el sistema (2.16), (2.17), (2.18) está completamente sincronizado de manera asintótica si las soluciones x, y y z de (2.16), (2.17), (2.18) con condiciones iniciales $x(0), y(0)$ y $z(0)$ satisfacen [13]

$$\lim_{t \rightarrow \infty} e(t) := x(t) - y(t) = 0, \quad (2.19)$$

y

$$\lim_{t \rightarrow \infty} z(t) = 0. \quad (2.20)$$

Sin embargo, en la práctica los sistemas no pueden cumplir (2.19) y (2.20), es decir, es imposible tener sistemas físicos idénticos. Por tal motivo, se introduce el concepto de sincronización práctica.

2.3.2. Definición 2 Sincronización práctica.

Se dice que los sistemas (2.16), (2.17), (2.18) están prácticamente sincronizados de manera asintótica si las soluciones $x(t)$, $y(t)$ y $z(t)$ de (2.16), (2.17), (2.18) con condiciones iniciales $x(0)$, $y(0)$ y $z(0)$ satisfacen [13]

$$\lim_{t \rightarrow \infty} |e(t)| := |x(t) - y(t)| \leq \epsilon_1 \quad (2.21)$$

y

$$\lim_{t \rightarrow \infty} z(t) \leq \epsilon_2 \quad (2.22)$$

con $\epsilon_1 > 0$ y $\epsilon_2 > 0$.

El estudio de este fenómeno requiere determinar la estabilidad de la solución síncrona. Debido a esto, se han buscado herramientas que permitan conocer la estabilidad de dicho estado o las condiciones bajo las cuales el estado síncrono es estable. En [13, 14, 15] se realiza un análisis de estabilidad de la sincronización para el sistema (2.16), (2.17), (2.18), en el cual se recurre a una función maestra de estabilidad y al cálculo del máximo exponente transverso de Lyapunov a partir de una ecuación denominada ecuación variacional.

2.4. Modelo cinemático y seguimiento de trayectoria

2.4.1. Modelo cinemático

Para poder diseñar un control para un robot móvil, primero se debe conocer el modelo matemático por el cual se rige su movimiento. Es por ello que, a continuación se presenta un robot móvil tipo 2.0, véase la Figura 2.14, cuyo modelo cinemático está dado por

2.4. MODELO CINEMÁTICO Y SEGUIMIENTO DE TRAYECTORIA

$$\begin{aligned}\dot{x} &= V \cos(\theta), \\ \dot{y} &= V \sin(\theta), \\ \dot{\theta} &= W,\end{aligned}\tag{2.23}$$

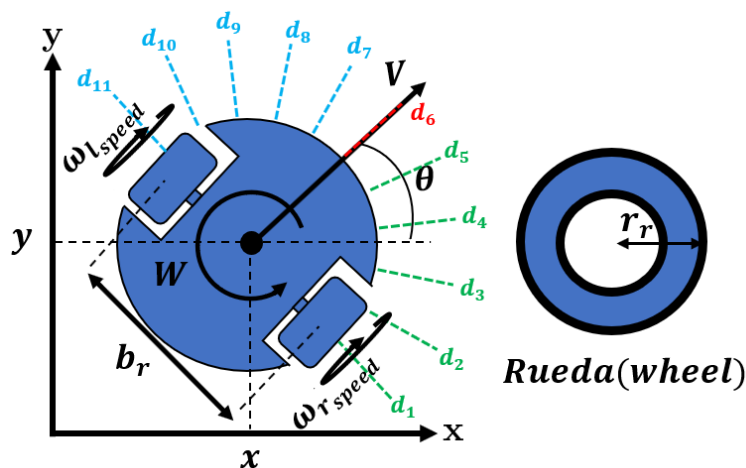


Figura 2.14: Robot móvil tipo 2.0 y orientación de los sensores.

donde (x, y) es el punto que representa el centro y posición del robot, el ángulo θ es su orientación con respecto al eje horizontal, mientras que V y W son las velocidades traslacional y rotacional, respectivamente. $\omega_{l\text{speed}}$ y $\omega_{r\text{speed}}$ corresponden a las velocidades angulares de las ruedas izquierda y derecha, respectivamente, r_r es el radio de las ruedas, y b_r la distancia de separación que hay entre ellas. Por último, las líneas punteadas de colores representan las posiciones radiales de los sensores que están espaciados una distancia equidistante de $\frac{\pi}{10}$.

2.4.2. Control de seguimiento de trayectoria

Para poder establecer el seguimiento de trayectoria se opta por la estrategia de un robot virtual de referencia el cual genera la trayectoria deseada, de esta forma el robot móvil se encargará de seguir al robot virtual de referencia. De acuerdo con [6], se debe determinar V y W , véase (2.23) que actúan como entradas de control para el robot móvil, esto se realiza por medio del siguiente procedimiento.

1) Se establecen las ecuaciones paramétricas de la posición deseada x_d y y_d , así como sus primeras y segundas derivadas \dot{x}_d, \dot{y}_d y \ddot{x}_d, \ddot{y}_d , respectivamente.

2) Se calculan las velocidades V_r, W_r , y la posición angular θ_r del robot virtual

$$\begin{aligned} V_r &= \sqrt{\dot{x}_d^2 + \dot{y}_d^2}, \\ W_r &= \frac{\dot{x}_d \ddot{y}_d - \dot{y}_d \ddot{x}_d}{V_r^2}, \\ \theta_r &= \tan^{-1} \left(\frac{\dot{y}_d}{\dot{x}_d} \right). \end{aligned} \quad (2.24)$$

3) Se formula el error de seguimiento

$$e = \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_d - x \\ y_d - y \\ \theta_r - \theta \end{pmatrix}. \quad (2.25)$$

4) Se implementa el control [6]

$$\begin{aligned} u_1 &= -k_1 e_1, \\ u_2 &= -k_2 \tanh(10V_r) e_2 - k_3 e_3. \end{aligned} \quad (2.26)$$

5) Se sintonizan las ganancias k_1, k_2, k_3 del control

6) Se determina el valor de V y W

$$\begin{aligned} V &= -u_1 + V_r \cos(e_3), \\ W &= W_r - u_2. \end{aligned} \quad (2.27)$$

7) Finalmente, se obtienen las velocidades angulares de cada rueda del robot

$$\omega_{rT} = \frac{1}{r_r} \left(V + \frac{b_r W}{2} \right), \quad (2.28)$$

$$\omega_{lT} = \frac{1}{r_r} \left(V - \frac{b_r W}{2} \right), \quad (2.29)$$

2.4. MODELO CINEMÁTICO Y SEGUIMIENTO DE TRAYECTORIA

donde ω_{r_T} y ω_{l_T} son las velocidades angulares de la rueda derecha e izquierda, respectivamente, r_r es el radio de las ruedas del robot, b_r es la separación entre las ruedas. Note que las Ecs. (2.28) y (2.29) permiten operar a nivel de velocidad de rueda.

Esta estrategia de seguimiento de trayectoria se elige porque proporciona un buen desempeño y el procedimiento es relativamente sencillo de implementar, además, la única modificación fue utilizar una función $\tanh()$ en lugar de la función $\text{sign}()$ en el control, esto es para tener un controlador continuo y evitar el chattering, ver Ec. (2.26).

Capítulo 3

Diseño de la estrategia de navegación autónoma

En este capítulo se presenta el diseño de una estrategia de evasión de obstáculos, la cual permite a un robot móvil navegar de manera autónoma en un espacio desconocido. Además, se introduce el diseño de un control híbrido, el cual está basado en la estrategia de evasión antes mencionada y el control de seguimiento de trayectoria discutido en el Capítulo 2.

3.1. Estrategia de evasión

La idea principal de la estrategia de evasión de obstáculos consiste en hacer variar a las corrientes externas (I_1 e I_2) de un primer par de neuronas tipo Hindmarsh-Rose (HR1 y HR2) en función de las lecturas proporcionadas por los sensores laterales tipo Lidar que están equipados en el robot móvil, siendo las lecturas del grupo de sensores derecho asociadas con la variación de la corriente externa I_1 , y las lecturas del grupo de sensores izquierdo con la variación de la corriente externa I_2 . Esta variación en las corrientes externas de cada modelo, permite aumentar o disminuir la frecuencia de picos producidos en los comportamientos de espiguelo de los potenciales de membrana (x_1 y x_2) de cada neurona, los cuales son contados mediante una ventana de tiempo y estos conteos de picos son asociados a las velocidades de rueda del robot móvil, siendo el conteo de picos en x_1 asociado con la velocidad de rueda izquierda y el conteo en x_2 asociado con la velocidad de rueda derecha. De este modo se controlan las velocidades de rueda en función de los obstáculos que vaya detectando el robot móvil.

CAPÍTULO 3. DISEÑO DE LA ESTRATEGIA DE NAVEGACIÓN AUTÓNOMA

Por otra parte, se interconecta otro par de neuronas tipo Hindmarsh-Rose (HR3 y HR4) mediante un acoplamiento dinámico y se utiliza la lectura del sensor frontal para producir modos de relajación completa y spiking en el potencial de membrana de la neurona HR4. Dependiendo de la actividad o modo del potencial de membrana de HR4, se hace cambiar el sentido de giro de la rueda derecha. Con ello, se puede establecer una estrategia de evasión frontal, ya que si el robot móvil detecta un obstáculo con el sensor frontal a una distancia de umbral, se cambia el sentido de giro de la rueda derecha para efectuar un giro de evasión. Esta estrategia se detalla en la respectiva sección.

3.1.1. Estrategia de evasión lateral

El procedimiento de la estrategia de evasión lateral está inspirado en [15] y se muestra a continuación.

1) Se establecen los modelos neuronales [13, 14, 15]

$$HRi : \begin{cases} \dot{x}_i &= y_i + bx_i^2 - ax_i^3 - z_i + I_i + v_i, \\ \dot{y}_i &= c - dx_i^2 - y_i, \\ \dot{z}_i &= r[s(x_i - x_0) - z_i], \end{cases} \quad (3.1)$$

donde $i = 1, 2$, $x_i \in R$ es el potencial de membrana, $y_i \in R$ es la corriente de recuperación, $z_i \in R$ es la corriente de adaptación, a, b, c, d, s, r, x_0 son parámetros constantes, e $I_i \in R$ son las corrientes externas (entradas del modelo), al tener una correcta configuración de los parámetros, se puede variar el valor de la corriente externa, entre mayor sea, mayor frecuencia o número de espigues habrá en el potencial de membrana y viceversa.

Las neuronas HR1 y HR2, interactúan a través de un acoplamiento difuso dado por [14, 15]

$$\begin{cases} v_1 = \varepsilon(x_2 - x_1), \\ v_2 = \varepsilon(x_1 - x_2), \end{cases} \quad (3.2)$$

con fuerza de acoplamiento $\varepsilon > 0$.

Este acoplamiento difuso permite establecer una sincronización entre los potenciales de membrana x_1 y x_2 cuando sus corrientes externas son las mismas ($I_1 = I_2$), sin embargo, ante diferentes entradas de corriente externa ($I_1 \neq I_2$), los potenciales de membrana no están sincronizados y presentan un comportamiento de espigues de diferente frecuencia.

2) Se distribuye equidistantemente la posición radial de 11 sensores tipo lidar, véase la Figura 2.14.

3) Se obtienen las distancias mínimas captadas por, todos los sensores (d), por el grupo de sensores del lado izquierdo (d_{izq}), por el grupo del lado derecho (d_{der}) y la lectura del sensor frontal (d_{fron}).

$$\begin{aligned}
 d_{der} &= \min(d_1, d_2, d_3, d_4, d_5), \\
 d_{izq} &= \min(d_7, d_8, d_9, d_{10}, d_{11}), \\
 d_{fron} &= d_6, \\
 d &= \min(d_{izq}, d_{der}, d_{fron}).
 \end{aligned} \tag{3.3}$$

4) Se establecen las expresiones para las corrientes externas I_1, I_2 (entradas de HR1-HR2) en función de las distancias de los grupos de sensores derecho e izquierdo [14, 15].

$$I_1 = \begin{cases} a_1 I_{min} & d_{der} \leq x_{min} \\ b_1 I_{min} \left(\frac{I_{max}}{I_{min}} \right) \frac{d_{der} - x_{min}}{x_{max} - x_{min}} & x_{min} < d_{der} < x_{max} \\ c_1 I_{max} & d_{der} \geq x_{max} \end{cases} \tag{3.4}$$

$$I_2 = \begin{cases} a_1 I_{min} & d_{izq} \leq x_{min} \\ b_1 I_{min} \left(\frac{I_{max}}{I_{min}} \right) \frac{d_{izq} - x_{min}}{x_{max} - x_{min}} & x_{min} < d_{izq} < x_{max} \\ c_1 I_{max} & d_{izq} \geq x_{max} \end{cases} \tag{3.5}$$

donde I_{min} e I_{max} son las corrientes externas mínima y máxima, respectivamente, x_{min} y x_{max} son las distancias mínima y máxima, respectivamente, que puede leer el sensor. Los parámetros a_1, b_1, c_1 , se utilizan para un mejor ajuste en los valores que toman las corrientes externas y con ello tener una mejor precisión en los giros de evasión.

De esta manera las entradas de corriente externa son (exponencialmente) proporcionales a las distancias medidas por los grupos de sensores.

5) Se realiza un conteo de espigas en x_1 y x_2 durante una ventana de tiempo (en este trabajo se consideró una ventana de 100.0[ms]) y se obtienen las velocidades de cada rueda por medio de la siguiente expresión.

$$\omega_{lV} = n_1 p; \quad \omega_{rV} = n_2 p \tag{3.6}$$

CAPÍTULO 3. DISEÑO DE LA ESTRATEGIA DE NAVEGACIÓN AUTÓNOMA

donde ω_{l_v} y ω_{r_v} son las velocidades de las ruedas izquierda y derecha, respectivamente, n_j el número de espigas contadas en el potencial de membrana x_j , $j = 1, 2$ dentro de una ventana de tiempo y p una ganancia de sintonización para regular la velocidad del robot móvil.

Entonces, de forma general, la idea de la estrategia lateral consiste en aprovechar el conteo de picos que se presentan en las respuestas de los potenciales de membrana x_1, x_2 , que varían su frecuencia con respecto a la desincronización o variación de las corrientes externas, en función de las distancias mínimas hacia los obstáculos que captan los grupos de sensores Lidar. Y con ello, asociar este conteo de picos con las velocidades de rueda del robot móvil.

En la Ec. (3.6) se ve que se asocia el conteo de picos del potencial de membrana x_1 con la velocidad de rueda izquierda ω_{l_v} y a su vez el conteo de picos en el potencial de membrana x_2 se asocia con la velocidad de rueda derecha ω_{r_v} . La Ec. (3.4) indica que, si el robot móvil detecta algún obstáculo cerca del lado derecho, la corriente externa I_1 que corresponde a la neurona HR1 disminuye su valor, por lo que, la frecuencia de picos en el potencial de membrana es menor y ocasiona que la velocidad de rueda izquierda disminuya y se efectúe un giro de evasión (de forma análoga, esta idea también aplica para la Ec. (3.5) y los obstáculos del lado izquierdo).

Por último, se debe aclarar que en los modelos neuronales se utiliza un escalamiento en tiempo $t^* = w_c t$ en las neuronas HR1 y HR2, donde t es el tiempo, t^* es el tiempo escalado y w_c es el factor de escalamiento, esto se realiza con el fin de tener una señal del potencial de membrana adecuada para la estrategia de evasión.

3.1.2. Estrategia de evasión frontal

La estrategia de evasión lateral solo permite realizar evasiones cuando se detectan obstáculos con los sensores laterales que posee el robot móvil, por tal motivo se propone una estrategia de evasión frontal basada en la interconexión de dos neuronas tipo Hindmarsh-Rose (HR3 y HR4) y que consiste en modificar las Ecs. (3.7), en donde se añade una variable que conmuta entre los valores unitario positivo y negativo, para alterar el sentido de giro de la rueda derecha. La variable que conmuta entre los valores unitario positivo y negativo se relaciona con el comportamiento o modo del potencial de membrana de la neurona HR4 que esta en función del sensor frontal. El siguiente procedimiento describe dicha estrategia de evasión frontal.

1) Se establecen los modelos neuronales [13, 14, 15]

$$HRi : \begin{cases} \dot{x}_i &= y_i + bx_i^2 - ax_i^3 - z_i + I_i + v_i, \\ \dot{y}_i &= c - dx_i^2 - y_i, \\ \dot{z}_i &= r_d[s(x_i - x_0) - z_i], \end{cases} \quad (3.7)$$

$$\begin{aligned} v_3 &= h_2 \\ v_4 &= -Gh_2 \end{aligned}$$

donde $i = 3, 4$, $x_i \in R$ es el potencial de membrana, $y_i \in R$ es la corriente de recuperación, $z_i \in R$ es la corriente de adaptación, a, b, c, d, s, r_d, x_0 son parámetros constantes e $I_3 = I_4 = I$ es la corriente externa. Finalmente G es una expresión propuesta, del tipo exponencial que está en función del sensor frontal.

$$G = -11e^{-3(d_{fron} - x_{fmin})} + 1. \quad (3.8)$$

Con la expresión G se obtienen valores que le permiten a x_3 y x_4 cambiar de un comportamiento del tipo espiguelo a un comportamiento de relajación completa y viceversa. x_{fmin} es una distancia de umbral que cuando es alcanzada por la lectura del sensor frontal induce el cambio de un comportamiento a otro.

2) Se interconectan las neuronas HR3 y HR4 mediante el acoplamiento dinámico

$$\begin{cases} \dot{h}_1 &= -\alpha h_1 + h_2, \\ \dot{h}_2 &= -\gamma_1 h_1 - \gamma_2 h_2 - k(x_3 - x_4), \end{cases} \quad (3.9)$$

con fuerza de acoplamiento k y parámetros $\alpha, \gamma_1, \gamma_2$.

3) Se define la variable x_{4A} como

$$x_{4A} = \begin{cases} 1 & \text{si } x_4 + 2.5 > 0, \\ 0 & \text{si } x_4 + 2.5 = 0, \\ -1 & \text{si } x_4 + 2.5 < 0. \end{cases} \quad (3.10)$$

4) Se añade la señal x_{4A} en las Ecs. (3.6) para obtener las siguientes expresiones

$$\omega_{l_s} = n_1 p; \quad \omega_{r_s} = x_{4A} n_2 p, \quad (3.11)$$

donde ω_{l_s} y ω_{r_s} son las velocidades de las ruedas izquierda y derecha, respectivamente, que representan la estrategia de evasión completa.

CAPÍTULO 3. DISEÑO DE LA ESTRATEGIA DE NAVEGACIÓN AUTÓNOMA

Recordando que el potencial de membrana x_4 puede entrar en un comportamiento tipo espiguelo, o en un modo de relajación completa, esto dependiendo del valor que tome G (ver Ec. (3.8)), se establece que la señal x_{4A} toma un valor unitario positivo ($x_{4A} = 1.0$) cuando x_4 entra en modo espiguelo, sin embargo, x_{4A} toma un valor unitario negativo ($x_{4A} = -1.0$) cuando x_4 entra en el modo de relajación completa. De acuerdo con una adecuada sintonización en los parámetros de HR3 y HR4, y la propuesta en (3.8), x_4 entra en modo de relajación completa cuando $d_{fron} \leq x_{fmin}$, si esto no se satisface entonces, x_4 tendrá un modo espiguelo. De esta forma, al detectarse un obstáculo frontal a una distancia de x_{fmin} , se realiza un giro de evasión.

3.1.3. Función G

La Figura 3.1 muestra el gráfico de la función G para un valor de $x_{fmin} = 0.5[m]$ y un intervalo de lectura del sensor frontal $0[m] \leq d_{fron} \leq 10[m]$. En ella se puede ver como su comportamiento es del tipo exponencial invertido, de tal forma que cuando se tengan valores altos de d_{fron} , se corresponden valores de $G = 1$, es decir,

$$\lim_{d_{fron} \rightarrow \infty} G(d_{fron}) = 1.$$

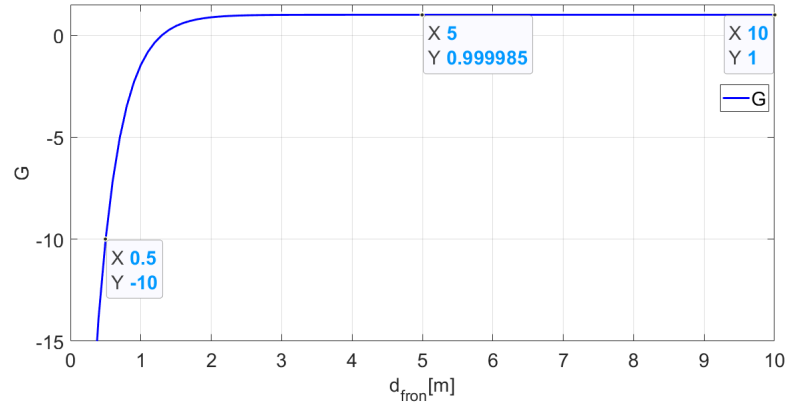


Figura 3.1: Gráfico de la función G , para un valor de $x_{fmin} = 0.5[m]$.

La función $G(d_{fron})$ también, fue diseñada tal que, G tome el valor de $G = -10$ cuando $d_{fron} = x_{fmin}$. En la Figura 3.1 se muestra como se cumple lo anterior considerando $x_{fmin} = 0.5[m]$.

Entonces con la propuesta de la función G se asegura que el comportamiento de x_4 entre en modo de relajación completa cuando el sensor frontal tome un valor igual o menor a la distancia de umbral ($d_{fron} \leq x_{fmin}$), y en caso contrario, entre en modo espiguelo cuando se aleje de este valor ($d_{fron} > x_{fmin}$).

3.2. Control híbrido

En este trabajo se presenta un control híbrido que permite seguir una trayectoria deseada y a su vez realizar la evasión de obstáculos. Dicho control surge de la combinación de la estrategia de seguimiento de trayectoria y la estrategia de evasión, es decir, de las Ecs. (2.28), (2.29) y (3.11). La condición que se establece para operar la conmutación de estos dos controles es la siguiente:

$$\left. \begin{array}{l} \omega_{l_{speed}} = \omega_{l_S} \\ \omega_{r_{speed}} = \omega_{r_S} \end{array} \right\} d \leq d_c \quad (3.12)$$

$$\left. \begin{array}{l} \omega_{l_{speed}} = \omega_{l_T} \\ \omega_{r_{speed}} = \omega_{r_T} \end{array} \right\} d > d_c \quad (3.13)$$

$\omega_{l_{speed}}$ y $\omega_{r_{speed}}$ son las velocidades de la rueda izquierda y derecha, respectivamente, que se entregan al robot, y d_c es un valor umbral que se obtiene de forma heurística (recordar que d es la distancia mínima captada por todos los sensores, ver (3.3)).

La prueba de estabilidad del control híbrido consiste en el análisis por separado de cada estrategia, para el caso del control del seguimiento de trayectoria, dicha prueba se puede consultar en [22]. Por otro lado, para el caso de los modelos neuronales, la prueba de estabilidad consiste en garantizar la sincronización entre las neuronas [13, 14, 15].

Capítulo 4

Resultados de simulación numérica

En este capítulo se describe la implementación del control híbrido realizada en el software de Matlab-Simulink 2020b, esto debido a que la plataforma ofrece herramientas para la simulación de robots móviles como la Robotics System Toolbox y Mobile Robotics Simulation Toolbox, además, permite realizar una co-simulación con el entorno de Gazebo. Por otro lado, se muestran algunos casos de estudio en los cuales se pueden apreciar los resultados de simulación numérica, que ilustran el desempeño del control.

4.1. Implementación en Matlab-Simulink 2020b

Para la simulación del robot móvil se elaboró un diagrama a bloques en Matlab-Simulink 2020b, ver Figura 4.1, de acuerdo con las herramientas proporcionadas por ambas paqueterías (toolbox). Este diagrama está conformado por: **1)** un bloque de función que contiene el cálculo de las corrientes externas para la alimentación del modelo de las neuronas HR. **2)** un bloque de subsistema que contiene los modelos de las neuronas HR1, HR2, HR3, HR4, así como su acoplamiento respectivo. **3)** un subsistema que contiene los detectores-contadores de espigas. **4)** una etapa de acondicionamiento para las señales de los potenciales de membrana x_3, x_4 . **5)** un subsistema que abarca el control de la trayectoria deseada, un bloque de función que se encarga de la conmutación de controles (control híbrido), un bloque de función que establece las velocidades traslacional y rotacional máximas de saturación del robot, y el modelo cinemático del robot móvil (bloque Differential

CAPÍTULO 4. RESULTADOS DE SIMULACIÓN NUMÉRICA

Drive Simulation), ver Figura 4.2. **6)** Un bloque de sensores Lidar (Multi robot Lidar) y el bloque que permite visualizar las trayectorias o posiciones de los robots móviles de forma animada (Robot Visualizer y Multi Robot Enviroment). Además del diagrama a bloques, se crearon algunos mapas con obstáculos puntuales y áreas rectangulares que simulan habitaciones y pasillos, ver Figuras 4.3a-4.3c. Para lograrlo, se desarrolló un programa donde se utilizan las funciones `binaryOccupancyMap` y `setOccupancy`.

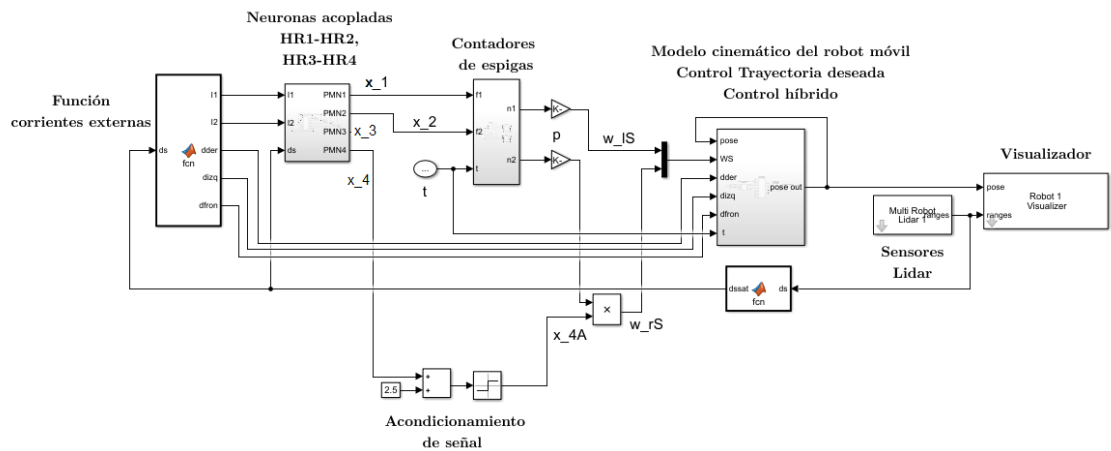


Figura 4.1: Diagrama a bloques para la simulación de un robot móvil programado con el control híbrido.

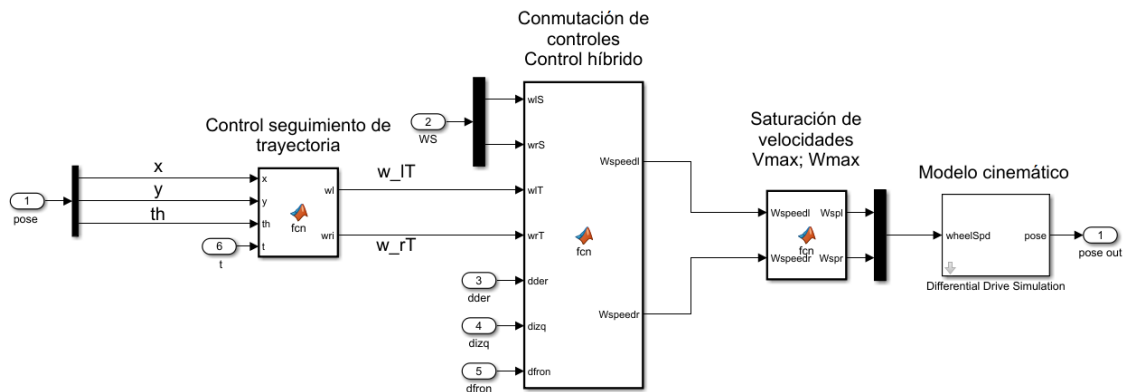
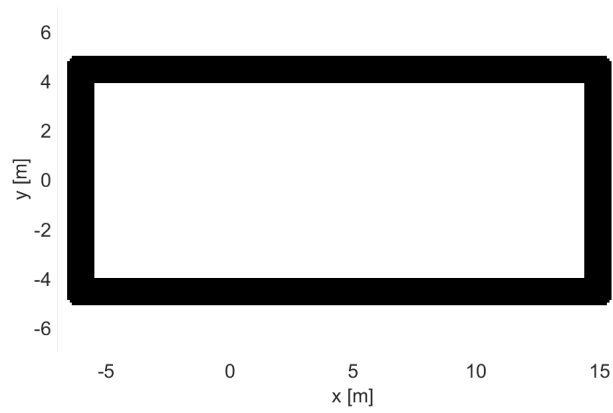
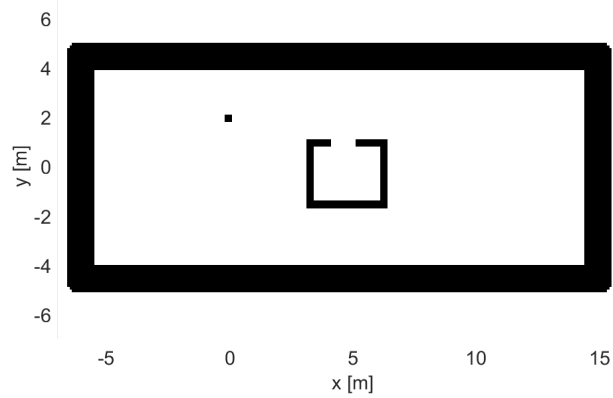


Figura 4.2: Diagrama a bloques del subsistema que contiene la estrategia de seguimiento de trayectoria y el control híbrido.

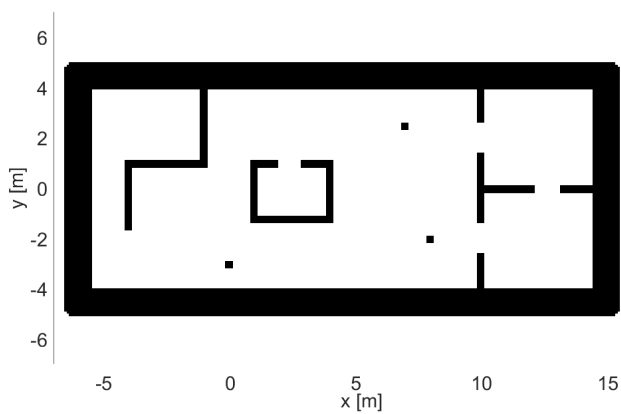
4.1. IMPLEMENTACIÓN EN MATLAB-SIMULINK 2020B



(a) Mapa 1, sin obstáculos.



(b) Mapa 2, con obstáculo puntual y área rectangular.



(c) Mapa 3, con obstáculos puntuales y áreas rectangulares.

Figura 4.3: Mapas utilizados en las simulaciones numéricas.

4.1.1. Subsistema detector-contador de espigas

La necesidad de un contador de espigas se debe a la estrategia de evasión basada en el modelo neuronal HR, ya que, para poder ejecutarla se requiere contar el número de espigas que emite el potencial de membrana, durante una ventana de tiempo. Al no haber un bloque específico para esta tarea, en este apartado se explica brevemente su construcción.

El contador de espigas realizado en la plataforma Matlab-Simulink 2020b es un subsistema que está conformado por cuatro etapas principales, de las cuales tres son para acondicionamiento de señal. La primera de ellas es una conversión de la señal de entrada a una señal cuadrada (del tipo pulso unitario o cruce por cero), esto se realiza para facilitar la detección del flanco de subida y tener un mejor conteo, por otro lado, esta etapa permite omitir la parte transitoria del modelo HR (correspondiente al período de tiempo de 0[s] a 250[ms]). La segunda etapa consiste en la transformación de la señal cuadrada a espigas o impulsos unitarios que corresponden a los flancos de subida, esta señal se obtiene mediante el uso del bloque Edge Detector. En la tercera etapa se define una señal cuadrada del tipo pulso unitario que actúa como la ventana de tiempo, cuyo tiempo en alto es de 100[ms] (tiempo donde se cuenta la señal de espiguelo), y su tiempo bajo corresponde a 10[ms] (tiempo donde se entrega la respuesta del conteo). La cuarta y última etapa es una función que se encarga de contar el número de espigas que se encuentran en la ventana de 100[ms], para ello se utilizan tres bloques de memoria para dar condición inicial al conteo, dar almacenamiento de la cuenta, y hacer uso como banderas para determinar cuándo empieza y cuándo acaba cada ciclo de conteo y de entrega de resultado. El diagrama a bloques se puede apreciar en la Figura 4.4.

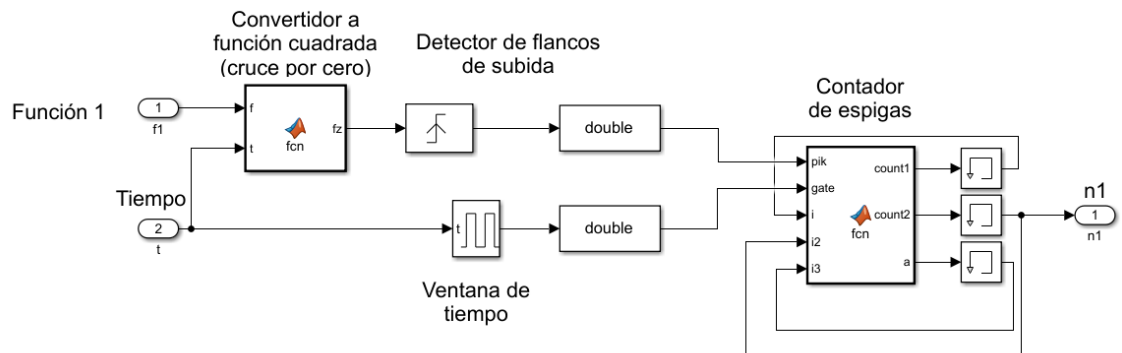


Figura 4.4: Diagrama a bloques del subsistema detector-contador de espigas.

4.2. Co-simulación Gazebo-Simulink

Para tener un mejor resultado de simulación numérica, es favorable implementar el mismo esquema de evasión en un ambiente más parecido a la realidad como lo ofrece el entorno de Gazebo. Debido que el software de Matlab-Simulink proporciona herramientas para entablar una comunicación con Gazebo, mediante el uso de una máquina virtual, además de que la simulación del esquema de control se realizó primero en esta plataforma, se opta por realizar una co-simulación entre Gazebo y Simulink que permite simular al robot móvil en el ambiente de Gazebo, a partir de la programación y el uso de los tópicos, suscriptores y editores desde Simulink.

Para entender de una mejor forma la co-simulación entre Gazebo y Simulink, en esta sección se muestran y describen brevemente los bloques ROS que permiten elaborar esta comunicación, así como el diagrama a bloques de la simulación.

4.2.1. Bloques ROS para la co-simulación

El uso de estos bloques en Simulink permite realizar una co-simulación con Gazebo, ya sea desde un solo pc que contiene el sistema operativo Ubuntu, o a través de Windows y el uso de una maquina virtual que contenga el sistema operativo Ubuntu. En este caso se prueba y utiliza la segunda opción por practicidad, donde se emplea el software VMware Workstation 16 Player que corresponde a la maquina virtual. También, cabe mencionar que la forma en que se establece la comunicación con la maquina virtual (VMware) es mediante las direcciones IP de la pc y la que se crea con la VMware. Aclarando lo anterior, se procede a mostrar los bloques ROS, ver, Figura 4.5.

El bloque **Publish**, ver Figura 4.5a, hace referencia a un editor (o publisher) el cual se encarga de publicar mensajes a un tópico en específico, en este trabajo se utiliza para mandarle los datos de las velocidades traslacional y rotacional al robot móvil, esto se debe a que Gazebo no permite enviar los datos de velocidades de ruedas directamente. En la Figura 4.5b se puede apreciar el bloque **Subscribe** que representa un suscriptor (o subscriber) encargado de tomar o recibir datos de un tópico en específico, en este trabajo se utiliza para obtener información referente a la odometría del robot móvil y las lecturas de los sensores de escaneo láser. El bloque **Blank Message** puede ser observado en la Figura 4.5c, con el se especifica el tipo de mensaje que se quiere publicar a un tópico en particular, por ejemplo, para publicar en el tópico `cdm_vel`, se deben utilizar mensajes del tipo `geometry_msgs/Twist`. La Figura 4.5d muestra el bloque **Bus Selector**, el cual permite seleccionar

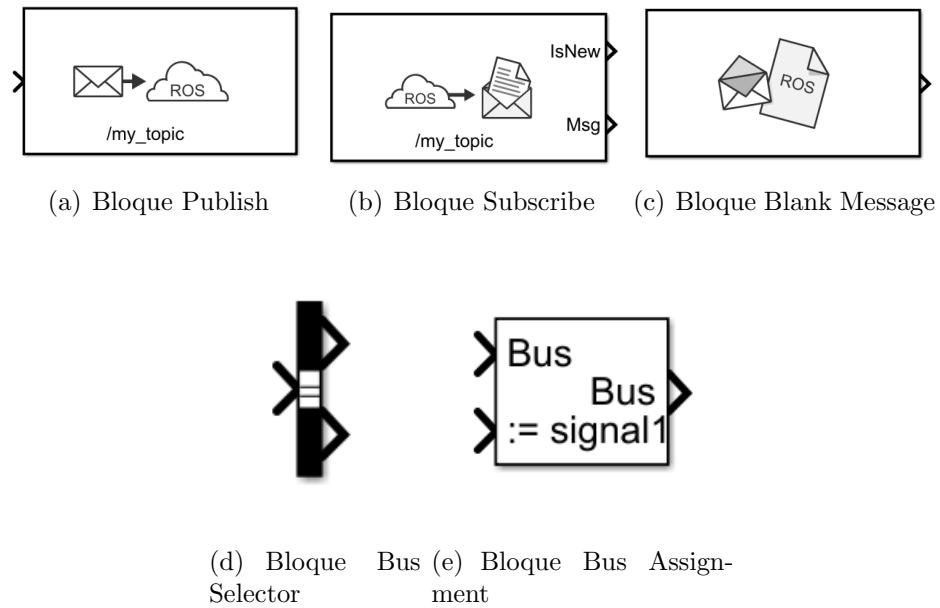


Figura 4.5: Bloques ROS que ofrece la paquetería Robotics System Toolbox.

la información o datos requeridos de un suscriptor, por ejemplo, dentro del tópic `laser_scan` se puede seleccionar el rango de los sensores laser, el ángulo máximo, el ángulo mínimo y la distancia de separación entre cada sensor, entre otros datos. Finalmente, el bloque **Bus Assignment** mostrado en la Figura 4.5e, es utilizado para asignar el tipo de datos o información que se quiera publicar en un tópic, por ejemplo, para el tópic `cdm_vel` se pueden mandar datos para la velocidad traslacional o la velocidad rotacional.

4.2.2. Diagrama a bloques para la co-simulación

En la Figura 4.6 se muestra el diagrama a bloques que contiene la estrategia del control híbrido propuesta, y hace posible establecer la co-simulación entre Gazebo-Simulink.

Este diagrama a bloques es una adaptación del diagrama a bloques mostrado en la Figura 4.1, a este se le agregaron los bloques ROS en las etapas de recepción de datos de los sensores Lidar y los datos de odometría, así como en la etapa de envío de datos de las velocidades traslacional y rotacional. En la Figura 4.7 se muestra el subsistema que corresponde al control de seguimiento de trayectoria y control híbrido.

El subsistema consiste de cuatro etapas, la primera se encarga de recibir los datos de odometría, la segunda es una conversión de cuaterniones a

4.2. CO-SIMULACIÓN GAZEBO-SIMULINK

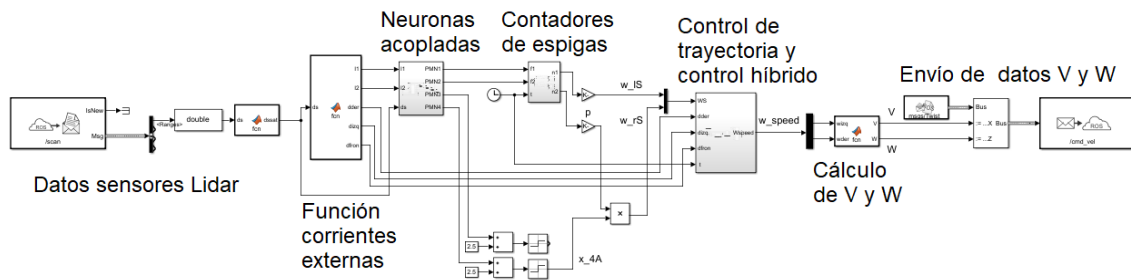


Figura 4.6: Diagrama a bloques para la simulación de un robot móvil programado con el control híbrido y la co-simulación Gazebo-Simulink.

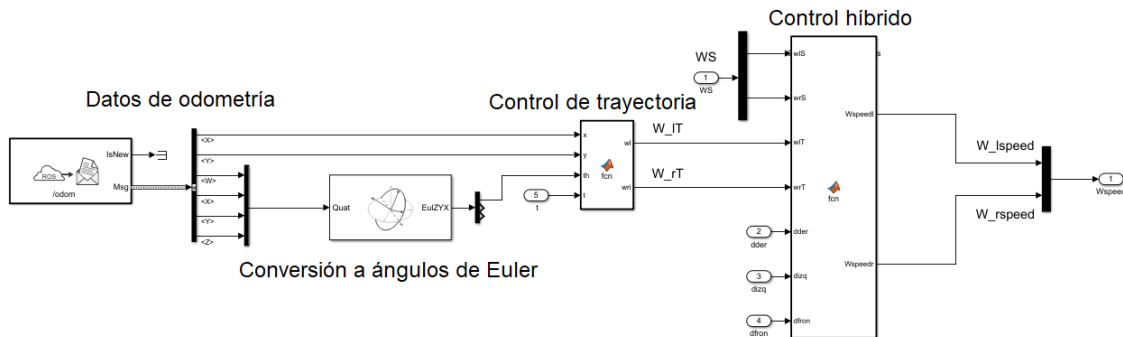


Figura 4.7: Subsistema para el control de seguimiento de trayectoria y control híbrido.

ángulos de Euler, esto con el fin de obtener la posición angular requerida en la tercera etapa, la cual consiste en el cálculo del control de trayectoria deseada. Finalmente, la cuarta etapa se refiere al control híbrido que entrega las velocidades de rueda.

Cabe recordar que ROS proporciona el modelo matemático del robot móvil y no acepta directamente las entradas de velocidades de rueda, es por ello que se hace una conversión de velocidades de rueda a velocidades traslacional y rotacional.

Es importante mencionar que la comunicación con la máquina virtual se establece mediante su puerto y dirección IP generados por Gazebo y estos deben ser ingresados en la configuración de red de algún bloque ROS, por ejemplo, en la configuración del bloque Subscribe, ver Figura 4.8.

Por otro lado, la configuración de los sensores y el tiempo de muestreo del robot móvil (que en este caso es un turtlebot3 burger) se modifica en Gazebo dentro de los archivos del propio robot y del “mundo”. Con respecto

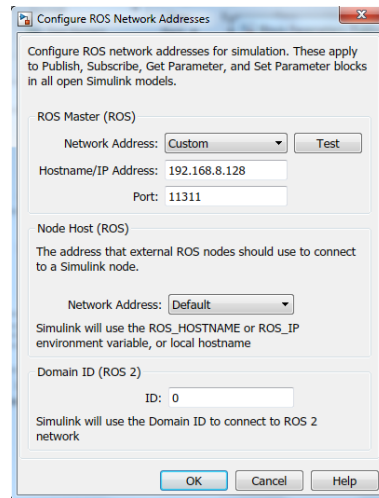


Figura 4.8: Parámetros y configuración del bloque Subscribe.

al tiempo de muestreo, este se escoge igual al de Simulink para que estén operando con la misma unidad de tiempo de simulación.

4.3. Simulación 1

Esta simulación se realizó en el ambiente de Matlab-Simulink 2020b y consiste en colocar en el mapa 3, ver Figura 4.3c, a cuatro robots móviles que están programados únicamente con la estrategia de evasión basada en el modelo neuronal HR. La tarea es que los cuatro robots móviles naveguen de forma autónoma alrededor del mapa, sin colisionar. Con ello se consigue poner a prueba el desempeño del control basado en el modelo neuronal ante obstáculos estáticos y dinámicos, así como áreas rectangulares que simulan cuartos y pasillos.

A continuación, en las Tablas 4.1 y 4.2 se muestra la sintonización de los parámetros utilizada, y las condiciones iniciales de los robots, respectivamente. Cabe aclarar que en este trabajo se emplearon datos del robot móvil Turtlebot3 (r_r radio de las ruedas, b_r separación entre las ruedas, V_{max} velocidad traslacional máxima, y W_{max} velocidad rotacional máxima [33]), ya que se piensa realizar la implementación en un robot móvil de ese tipo. Además, se considera un tiempo de muestreo de $0.05[s]$.

Parámetros del control basado en el modelo neuronal						
a_1	b_1	c_1	I_{min}	I_{max}	x_{min}	x_{max}
0.95	1.35	0.85	3.25	10	0.4[m]	5.0[m]
a	b	c	d	s	x_0	r
1.0	3.0	1.0	5.0	4.0	-1.6	0.005
ϵ	$\gamma_1 = \alpha$	$\gamma_2 = k$	I	x_{fmin}	p	r_r
0.5	1.0	10.0	3.281	0.5[m]	0.7	33.0[mm]
b_r	V_{max}	W_{max}	r_d			
160.0[mm]	0.22[m/s]	2.84[rad/s]	-0021			

Tabla 4.1: Sintonización de parámetros del control basado en el modelo neuronal para los robots. Experimento 1.

Condiciones iniciales			
robot i	$x_i(0)$	$y_i(0)$	$\theta_i(0)$
robot 1	2.5[m]	0.0[m]	$\frac{\pi}{2}$ [rad]
robot 2	13.0[m]	-2.0[m]	π [rad]
robot 3	13.0[m]	-3.0[m]	$\frac{\pi}{2}$ [rad]
robot 4	-5.0[m]	2.0[m]	$-\frac{\pi}{2}$ [rad]

Tabla 4.2: Condiciones iniciales de los robots para el experimento 1.

4.3.1. Resultados de la simulación 1

En la Figura 4.9 se puede observar la evolución de las trayectorias realizadas por los robots, donde las trayectorias color magenta, azul, cian y roja corresponden a los robots 1, 2, 3 y 4, respectivamente. En [34] se proporciona un hipervínculo, el cual dirige hacia un video que muestra de forma animada, la evolución de las trayectorias de los robots móviles que se consideran en la simulación numérica, ya que de esta manera, se aprecian las evasiones realizadas por parte de los robots. La Figura 4.10 muestra el comportamiento de las velocidades de rueda izquierda y derecha, que corresponden al robot 1. En ella se puede apreciar como varía la velocidad de acuerdo con la percepción del robot móvil (detección de obstáculos), siendo una velocidad de rueda más grande que la otra para realizar un giro de evasión, y cuando el robot se desplaza en línea recta, ambas velocidades de rueda poseen el mismo valor. Una forma más clara de notar los resultados aportados por el control basado en el modelo neuronal, es observando los gráficos de la Figura 4.11, que corresponden a un zoom de la evolución de los potenciales de membrana x_1 y x_2 , en estos se nota como en el comienzo se tiene una

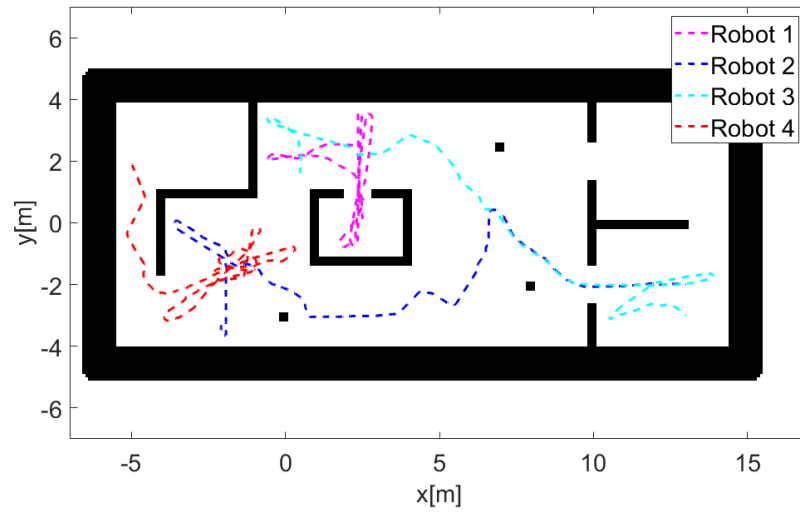


Figura 4.9: Trayectorias de los robots móviles.

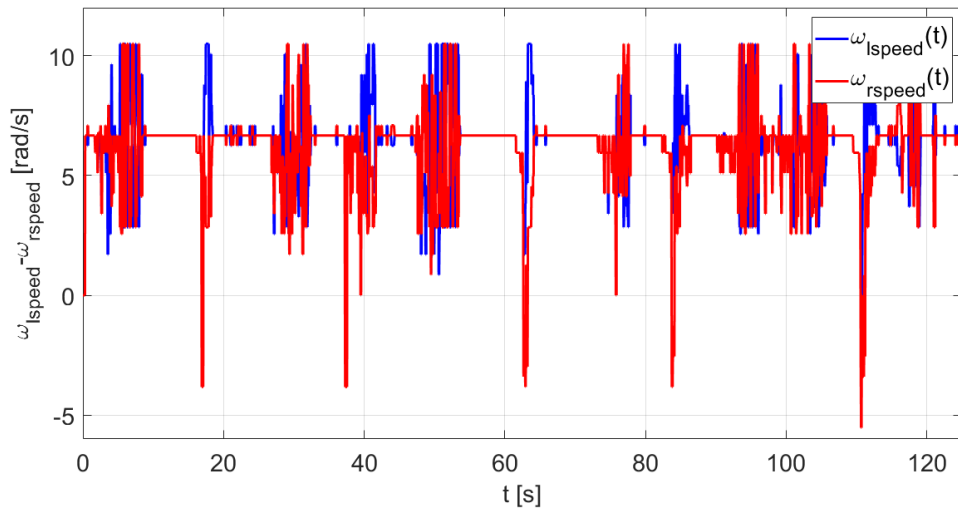


Figura 4.10: Velocidades de rueda $\omega_{l_{speed}}$ y $\omega_{r_{speed}}$. Robot 1.

sincronización en las señales x_1 y x_2 , esto quiere decir que el robot móvil posee la misma velocidad en ambas ruedas y esto le permite desplazarse en línea recta. Por otra parte, al final de las señales se aprecia como hay un mayor número de espigueso en un potencial de membrana que en otro, lo cual significa que una velocidad de rueda es mayor que la otra y se efectúa un giro de evasión. En la Figura 4.12 se muestran los potenciales de membrana

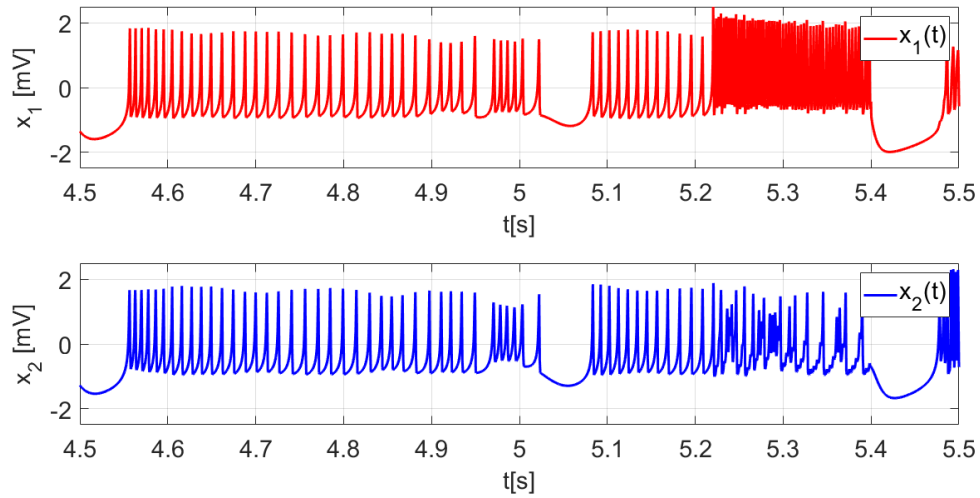


Figura 4.11: Zoom en la evolución de los potenciales de membrana x_1, x_2 , que actúan en el robot 1. Los potenciales se desincronizan cuando se detecta un obstáculo

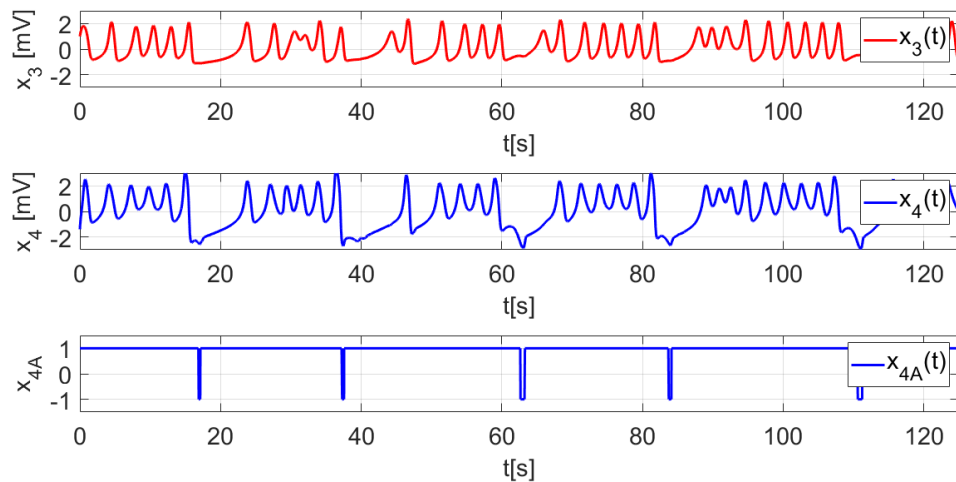


Figura 4.12: Evolución de los potenciales de membrana x_3, x_4 y el potencial de membrana x_{4A} , que actúan en el robot 1.

x_3 y x_4 del robot 1, en ellos se ve como sus comportamientos cambian del tipo espiguelo a un modo de relajación completa durante un breve intervalo de tiempo, esto ocasiona que x_{4A} tome un valor unitario negativo y cambie el sentido de giro de la rueda derecha para efectuar una evasión frontal (nótese

que el robot 1 realizó cinco evasiones frontales, durante todo el experimento).

Como el resultado para el resto de los robots es similar al del robot 1, se omiten sus gráficos correspondientes.

4.4. Simulación 2

Esta simulación se realizó en el ambiente de Matlab-Simulink 2020b y consiste en colocar en cada esquina del área rectangular del mapa 1, ver Figura 4.3a, a cuatro robots móviles programados con el control híbrido. La tarea es que los cuatro robots naveguen de forma autónoma, siguiendo la trayectoria diagonal para llegar a su esquina contraria, sin colisionar. El propósito de esta simulación es ilustrar el desempeño del control híbrido ante obstáculos dinámicos.

Para conseguir la trayectoria en diagonal, en este caso, se recurre a las ecuaciones paramétricas de la recta.

$$(x_d, y_d) = (x_0, y_0) + \omega_s t(a_v, b_v), \quad (4.1)$$

donde (x_d, y_d) es la posición deseada, (x_0, y_0) el punto o condición inicial, (a_v, b_v) es un vector que dirige el sentido y orientación de la recta y ω_s es un parámetro que determina la velocidad con la que se forma la recta. Cabe aclarar que se ocupa una trayectoria deseada diferente por cada robot móvil, ya que cada uno se dirige hacia un sentido y dirección diferente.

A continuación, en las Tablas 4.3, 4.4 y 4.5 se muestra la sintonización de los parámetros, y las condiciones iniciales de los robots, recordando que en este trabajo se emplearon datos del robot móvil Turtlebot3 [33]. Además, se considera un tiempo de muestreo de $0.05[s]$.

Parámetros del control de trayectoria deseada						
robot i	r_r	b_r	ω_s	a_v	b_v	$k_1 = k_2 = k_3$
robot 1	33.0[mm]	160.0[mm]	0.0076	-19.0[m]	7.0[m]	1.2
robot 2	33.0[mm]	160.0[mm]	0.0076	19.0[m]	-7.0[m]	1.2
robot 3	33.0[mm]	160.0[mm]	0.0076	19.0[m]	7.0[m]	1.2
robot 4	33.0[mm]	160.0[mm]	0.0076	-19.0[m]	-7.0[m]	1.2

Tabla 4.3: Sintonización de parámetros del control de seguimiento de trayectoria para los robots. Simulación 2.

Parámetros del control basado en el modelo neuronal						
a_1	b_1	c_1	I_{min}	I_{max}	x_{min}	x_{max}
0.95	1.35	0.85	3.25	10	0.4[m]	5.0[m]
a	b	c	d	s	x_0	r
1.0	3.0	1.0	5.0	4.0	-1.6	0.005
ϵ	$\gamma_1 = \alpha$	$\gamma_2 = k$	I	x_{fmin}	p	r_d
0.5	1.0	10.0	3.281	0.5[m]	0.7	-0021
V_{max}	W_{max}					
0.22[m/s]	2.84[rad/s]					

Tabla 4.4: Sintonización de parámetros del control basado en el modelo neuronal para los robots. Simulación 2.

Condiciones iniciales			
robot i	$x_i(0)$	$y_i(0)$	$\theta_i(0)$
robot 1	14.0[m]	-3.5[m]	$\frac{3\pi}{4}$ [rad]
robot 2	-5.0[m]	3.5[m]	$-\frac{\pi}{4}$ [rad]
robot 3	-5.0[m]	-3.5[m]	$\frac{\pi}{4}$ [rad]
robot 4	14.0[m]	3.5[m]	$\frac{5\pi}{4}$ [rad]

Tabla 4.5: Condiciones iniciales de los robots para la simulación 2.

La sintonización de los parámetros de la Tabla 4.3 se consiguió a partir de los datos en [33] (para el caso de r_r, b_r), ω_s se calcula sustituyendo las ecuaciones paramétricas de la recta en la expresión (2.24) y fijando el vector (a_v, b_v) y una velocidad de referencia V_r . Los parámetros k_1, k_2, k_3 se obtuvieron de manera heurística.

La sintonización de los parámetros de la Tabla 4.4 se consiguió a partir de los datos en [13, 14, 15] y de forma heurística para $a_1, b_1, c_1, \epsilon, p, x_{fmin}$.

4.4.1. Resultados de la simulación 2

En la Figura 4.13 se puede observar el resultado de la evolución de las trayectorias deseadas de los robots móviles, donde las trayectorias color magenta, azul, roja y cian corresponden a los robots 1, 2, 3 y 4, respectivamente. En [35] se incluye un hipervínculo, el cual dirige hacia un video que muestra de forma animada, la evolución de las trayectorias de los robots móviles que se consideran en la simulación numérica, ya que de esta manera, se aprecian las evasiones realizadas por parte de los robots. Las velocidades de rueda

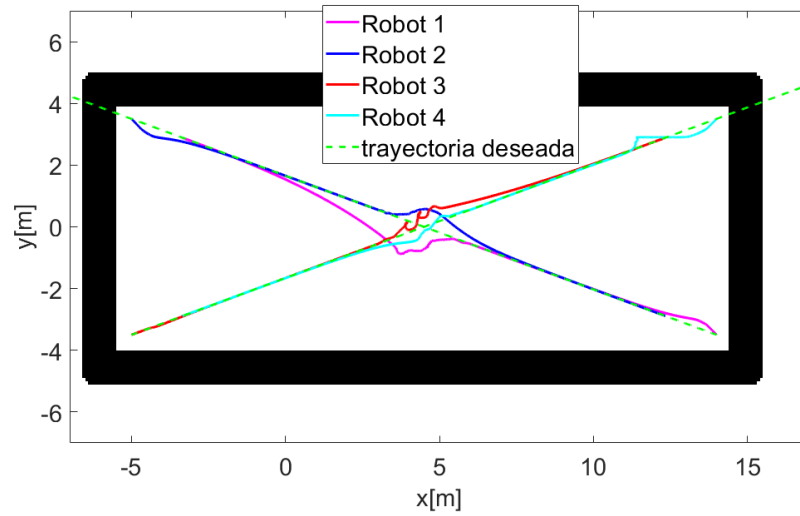


Figura 4.13: Trayectorias de los robots móviles.

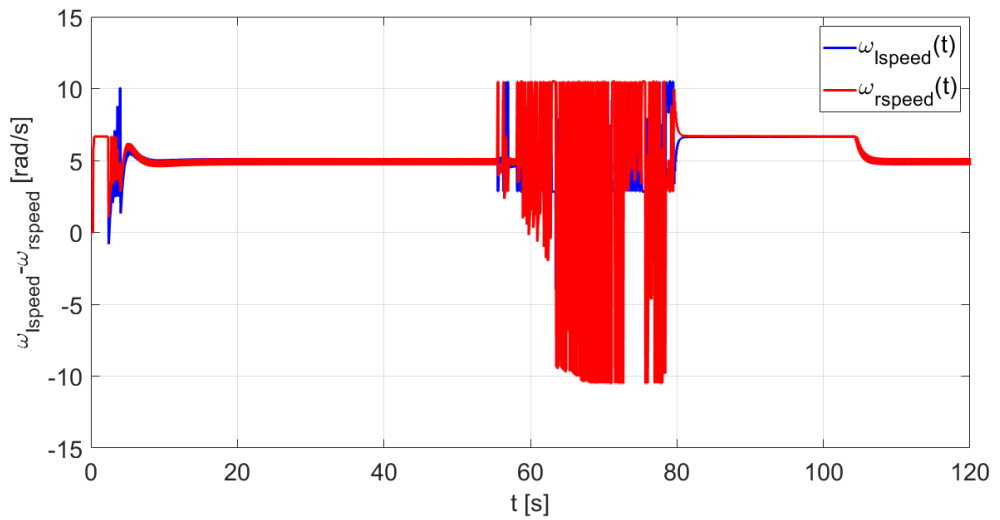


Figura 4.14: Velocidades de rueda $\omega_{l_{speed}}$ y $\omega_{r_{speed}}$. Robot 1.

$\omega_{l_{speed}}$ y $\omega_{r_{speed}}$ se muestran en la Figura 4.14, en la cual es claro ver que al iniciar la simulación no poseen el mismo valor, esto implica que el robot 1 no se desplaza en línea recta, es decir, no tiende a su trayectoria deseada. Esto se debe, porque en su condición inicial, el robot 1 se sitúa en la esquina inferior derecha del área rectangular que compone el mapa, de este modo los sensores del robot móvil detectan las paredes de la esquina, y en este caso el

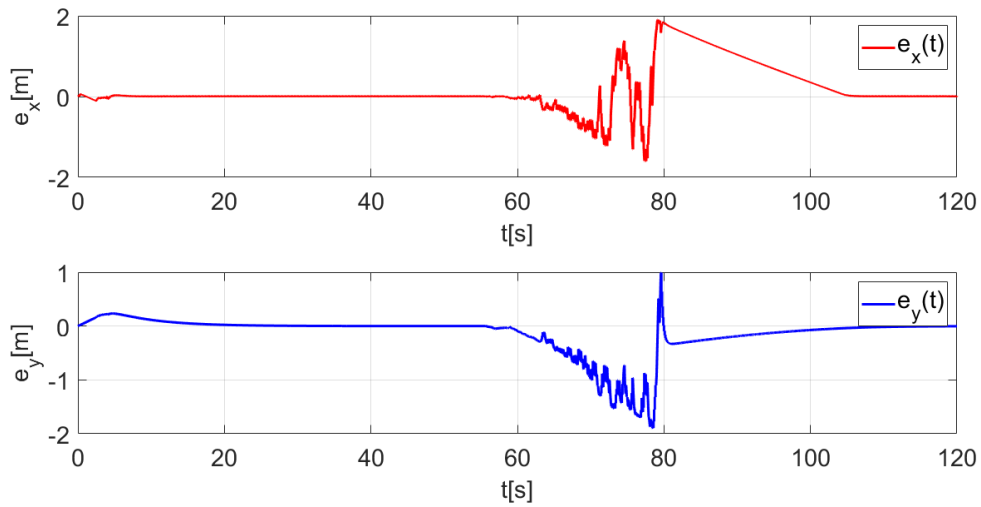
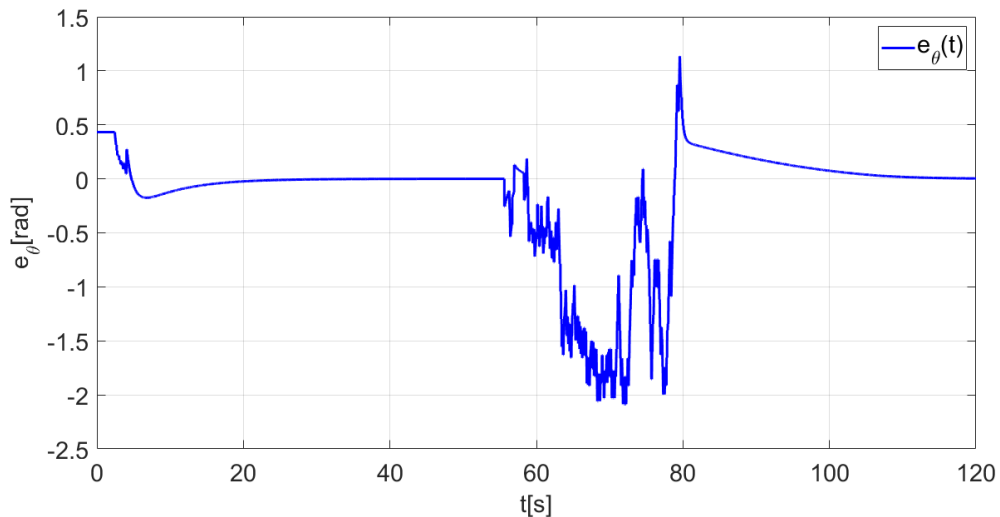
(a) Errores de posición en x e y . Robot 1.(b) Error de posición en θ . Robot 1.

Figura 4.15: Errores de posición. Robot 1.

robot 1 está operando con la estrategia de evasión basada en el modelo neuronal. Sin embargo, mientras el robot 1 se va desplazando, deja de detectar las paredes que componen la esquina, y cambia a la estrategia de seguimiento de trayectoria, de tal forma que, las velocidades de rueda son las mismas y el robot móvil se desplaza en línea recta siguiendo la trayectoria deseada. De igual forma se puede apreciar que en el intervalo de tiempo entre 50[s] y 80[s]

CAPÍTULO 4. RESULTADOS DE SIMULACIÓN NUMÉRICA

las velocidades de rueda vuelven a variar, siendo una mayor que otra, por lo que el robot 1 opera con el control basado en el modelo neuronal y toma acciones de evasión.

Otra forma de ver la respuesta del control híbrido, es observando las Figuras 4.15 a) y b) que representan los errores de posición del robot 1. En estas figuras se nota también como en el comienzo los errores no tienden a cero, pero conforme avanza el tiempo el robot móvil converge a la trayectoria deseada, lo cual implica que los errores de posición tienden a cero. También, en el mismo intervalo de tiempo entre 50[s] y 80[s] se nota como los errores en la Figura 4.15a no tienden a cero, es decir, el robot deja su trayectoria deseada debido a la detección de los obstáculos dinámicos (el resto de los robots), y se modifica la orientación que corresponde a los giros de evasión tal y como se muestra en la Figura 4.15b. Los resultados aportados por el

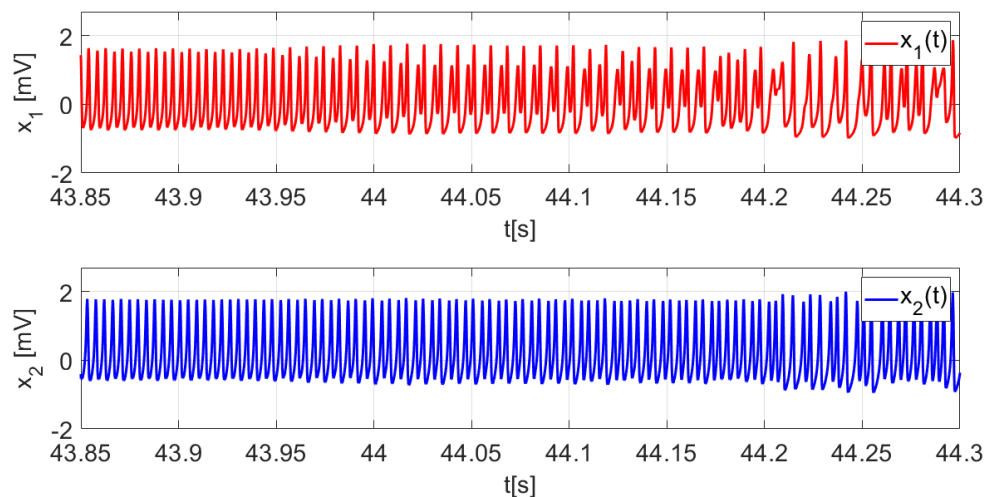


Figura 4.16: Zoom en la evolución de los potenciales de membrana x_1, x_2 , que actúan en el robot 1.

control basado en el modelo neuronal, pueden ser observados en los gráficos de la Figura 4.16, que corresponden a un zoom de la evolución de los potenciales de membrana x_1 y x_2 del robot 1. En estos se nota como en el comienzo de ese intervalo de tiempo, existe una sincronización entre las señales x_1 y x_2 , por tanto, el robot móvil posee la misma velocidad en ambas ruedas y esto le permite desplazarse en línea recta. Por otra parte, al final de las señales se aprecia como se va perdiendo la sincronización entre x_1 y x_2 y hay un mayor número de espiguelo en un potencial de membrana que

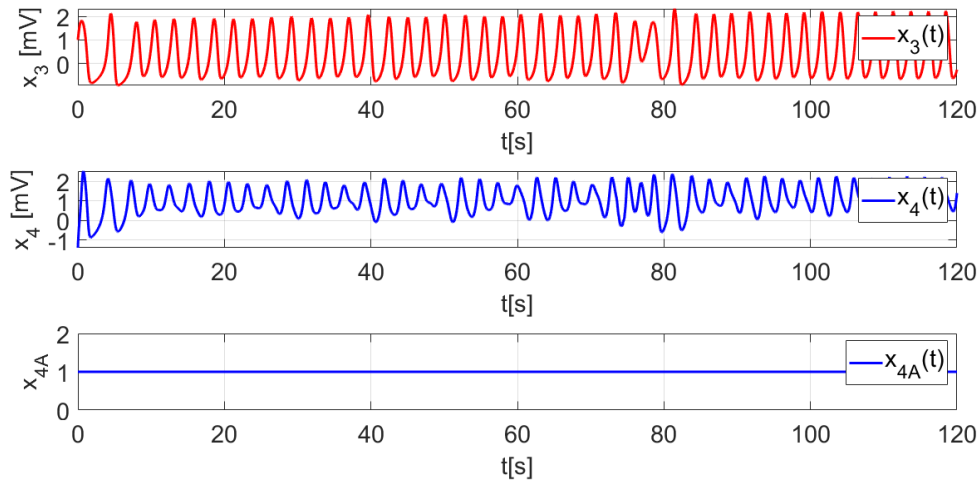


Figura 4.17: Evolución de los potenciales de membrana x_3 , x_4 y el potencial de membrana x_{4A} , que actúan en el robot 1

en otro, lo cual implica que una velocidad de rueda es mayor que la otra y se efectúa un giro de evasión. En la Figura 4.17 se muestran los potenciales de membrana x_3 y x_4 del robot 1, en ellos se ve como sus comportamientos a lo largo de toda la simulación son del tipo espiguelo, esto quiere decir que, el robot 1 en este caso, no realizó ninguna evasión frontal, sino que todas las evasiones realizadas fueron laterales. Este aspecto se puede notar también en la misma Figura 4.17, observando que el comportamiento de x_{4A} a lo largo de toda la simulación es de un valor unitario positivo, lo cual indica que el sentido de giro de la rueda derecha siempre es el mismo que el de la rueda izquierda. El hecho que no haya evasiones frontales se debe a que los robots se van encontrando conforme pasa el tiempo, y en el caso del robot 1, detecta de forma lateral el acercamiento del robot 4 (el robot 4 se acerca al robot 1 desde la esquina superior derecha del área rectangular), lo cual ocasiona que este empiece a tomar acciones de evasión, es decir, comienza a girar. De esta manera cuando el robot 1 se encuentra con el robot 2 (que es el que proviene desde su esquina contraria), lo hace con detecciones laterales también.

Como el resultado para el resto de los robots es similar al del robot 1, se omiten sus gráficos correspondientes.

4.5. Simulación 3

Esta simulación se realizó en el ambiente de Matlab-Simulink 2020b y consiste en colocar a dos robots móviles dentro de un espacio confinado, el robot 1 tiene implementado el control híbrido mientras que el robot 2 únicamente contiene la estrategia de evasión. La tarea es que ambos salgan de un área con una sola salida, ver Figura 4.3b, y posterior a ello naveguen de forma autónoma (sin colisionar) alrededor del resto del mapa, en especial el robot 1 tendrá que desarrollar una trayectoria deseada (en este caso un círculo de radio 2.0[m]) y efectuar una acción de evasión ante un obstáculo estático que se ubica sobre la trayectoria deseada. Por otra parte, el robot 2 navegará de manera autónoma.

Para conseguir la trayectoria circular, se recurre a las ecuaciones paramétricas del círculo.

$$(x_d, y_d) = (r_c \cos(\omega_{sc}t), r_c \sin(\omega_{sc}t)), \quad (4.2)$$

donde (x_d, y_d) es la posición deseada, r_c es el radio del círculo y ω_{sc} es un parámetro que determina la frecuencia con la que se forma el círculo. Los parámetros usados, así como las condiciones iniciales de los robots se muestran en las Tablas 4.6, 4.7 y 4.8. Además, se considera un tiempo de muestreo de 0.05[s].

Parámetros del control de trayectoria deseada para el robot 1				
r_r	b_r	r_c	ω_{sc}	$k_1 = k_2 = k_3$
33.0[mm]	160.0[mm]	2[m]	0.0770	1.2

Tabla 4.6: Sintonización de parámetros para el control de seguimiento de trayectoria del robot 1. Simulación 3.

La sintonización de los parámetros es obtenida de forma similar como se hizo en la simulación 2.

Parámetros del control basado en el modelo neuronal						
a_1	b_1	c_1	I_{min}	I_{max}	x_{min}	x_{max}
0.95	1.35	0.85	3.25	10	0.4[m]	5.0[m]
a	b	c	d	s	x_0	r
1.0	3.0	1.0	5.0	4.0	-1.6	0.005
ϵ	$\gamma_1 = \alpha$	$\gamma_2 = k$	I	x_{fmin}	p	r_d
0.5	1.0	10.0	3.281	0.5[m]	0.7	-0021
V_{max}	W_{max}					
0.22[m/s]	2.84[rad/s]					

Tabla 4.7: Sintonización de parámetros del control basado en el modelo neuronal para los robots. Simulación 3.

Condiciones iniciales			
robot i	$x_i(0)$	$y_i(0)$	$\theta_i(0)$
robot 1	3.7[m]	-1.0[m]	$\frac{\pi}{2}$ [rad]
robot 2	6.0[m]	0.0[m]	π [rad]

Tabla 4.8: Condiciones iniciales de los robots para la simulación 3.

4.5.1. Resultados de la simulación 3

En la Figura 4.18 se puede observar la evolución de las trayectorias del robot 1 (trayectoria color magenta) y el robot 2 (trayectoria color azul). Es importante notar que, una vez que el robot 1 sale del rectángulo central, este trata de converger a la trayectoria deseada (trayectoria color rojo), a la cual llega después de evadir el obstáculo estático que se colocó en medio con el fin de poner a prueba el control híbrido. En [36] se proporciona un hipervínculo, el cual dirige hacia un video que muestra de forma animada, la evolución de las trayectorias de los robots móviles que se consideran en la simulación numérica. De esta manera, se aprecian las evasiones realizadas por parte de ambos robots.

Las señales de las velocidades de rueda $\omega_{l_{speed}}$ y $\omega_{r_{speed}}$ del robot 1 se muestran en la Figura 4.19, donde se puede observar la variación inicial de dichas señales, siendo una mayor que la otra para efectuar giros de evasión. En otras palabras, el robot 1 es operado con la estrategia de evasión basada en el modelo neuronal y no tiende a la trayectoria deseada, sin embargo, después de que el robot 1 sale del área rectangular, es evidente como las velocidades de las ruedas izquierda y derecha son similares y constantes, esto significa que el robot 1 es operado por el control de seguimiento de trayectoria y tiende a

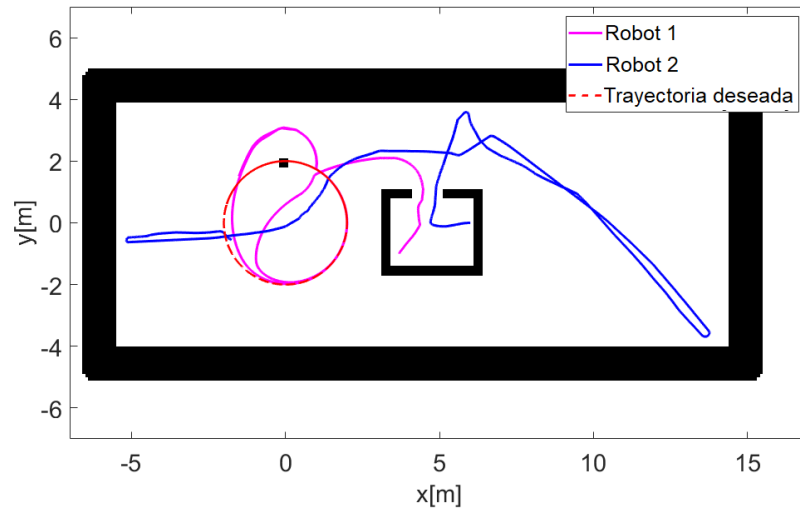


Figura 4.18: Trayectorias de los robots móviles.

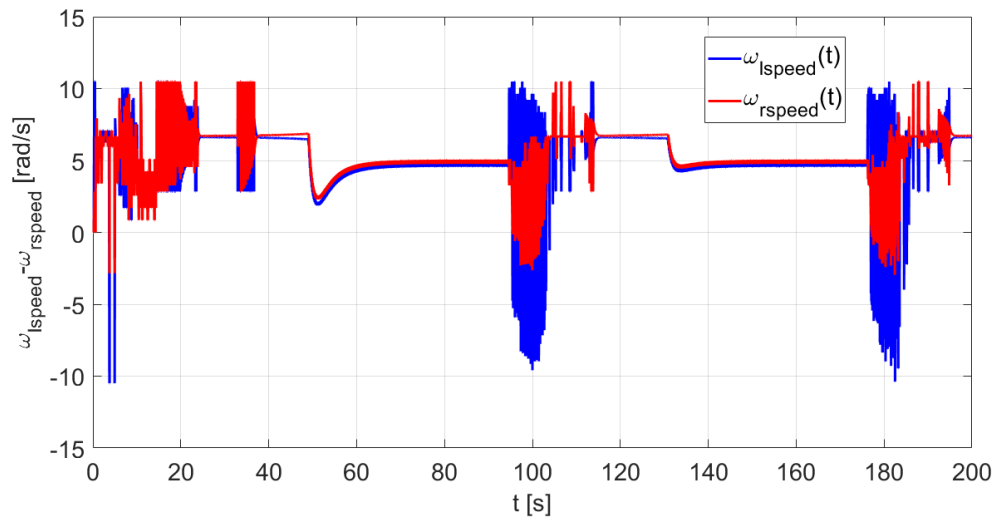


Figura 4.19: Velocidades de rueda $\omega_{l_{speed}}$ y $\omega_{r_{speed}}$. Robot 1.

la trayectoria deseada, la velocidad de la rueda derecha es ligeramente mayor que la velocidad de la rueda izquierda para que el robot móvil genere la trayectoria circular. Otra forma de apreciar el resultado que aporta el control híbrido es viendo las Figuras 4.20a y 4.20b que corresponden a los errores de posición. En la Figura 4.20a se nota como los errores no convergen a cero al comienzo de la simulación, esto es debido a que el robot 1 se encuentra

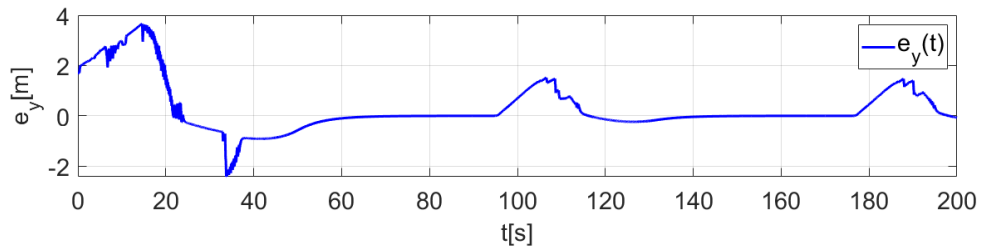
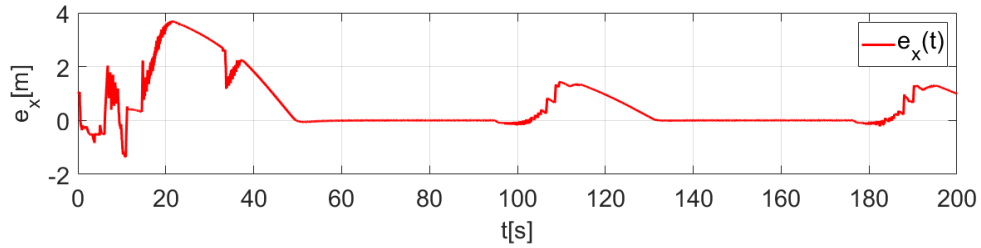
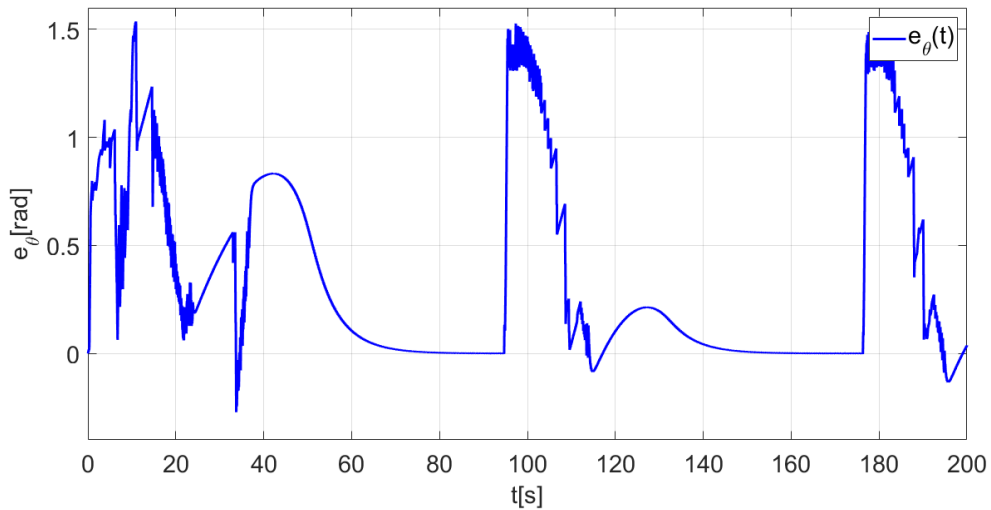
(a) Errores de posición en x e y . Robot 1.(b) Error de posición en θ . Robot 1.

Figura 4.20: Errores de posición. Robot 1.

realizando acciones de evasión dentro del rectángulo central junto al robot 2. Después de que el robot 1 consigue salir del rectángulo central comienza a tender a la trayectoria deseada (los errores comienzan a tender a cero), hasta que en un instante de tiempo cercano a 35[s] detecta y evade al obstáculo estático, lo cual ocasiona una pequeña variación en los errores de posición. Posterior a ello, los errores de posición tienden a cero y es en un tiempo

CAPÍTULO 4. RESULTADOS DE SIMULACIÓN NUMÉRICA

cercano a los 95[s] cuando el robot 1 vuelve a detectar al obstáculo estático y se puede apreciar de mejor forma la variación en los errores de posición, ilustrando así las acciones de evasión que toma el robot 1 para evadir al obstáculo estático (otra evasión similar ocurre en un tiempo cercano a los 175[s]). En la Figura 4.20b se ilustra como el robot 1 varia su orientación para generar las acciones de evasión. Una forma más clara de notar la efectividad del control

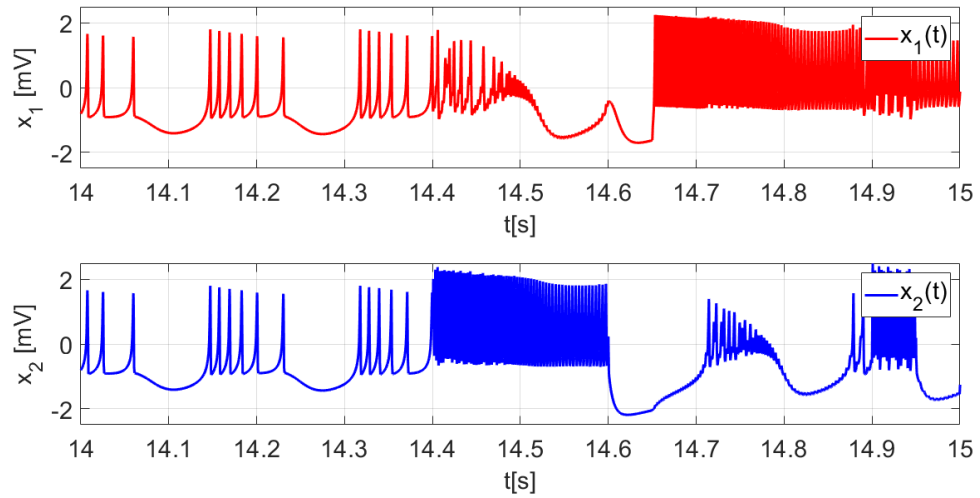


Figura 4.21: Zoom en la evolución de los potenciales de membrana x_1, x_2 , que actúan en el robot 1.

basado en el modelo neuronal, es observando los gráficos de la Figura 4.21, que corresponden a los potenciales de membrana x_1 y x_2 , en estos se nota como en el comienzo se tiene una sincronización en las señales x_1 y x_2 , esto quiere decir que el robot móvil posee la misma velocidad en ambas ruedas y esto le permite desplazarse en línea recta. Por otra parte, hay intervalos de tiempo donde se nota como hay un mayor número de espigues en un potencial de membrana que en otro, lo cual significa que una velocidad de rueda es mayor que la otra y se efectúa un giro, que puede ser de evasión ó, debido a la trayectoria deseada. En la Figura 4.22 se muestran los potenciales de membrana x_3 y x_4 del robot 1, en ellos se ve como sus comportamientos a lo largo de toda la simulación son del tipo espigues, esto quiere decir que, el robot 1 en este caso, no realizó ninguna evasión frontal, sino que todas las evasiones realizadas fueron laterales. Este aspecto se puede notar también en la misma Figura 4.22, donde se muestra el comportamiento de x_{4A} , este a lo largo de toda la simulación tiene un valor unitario positivo, lo cual indica que el sentido de giro de la rueda derecha siempre es el mismo que el de la

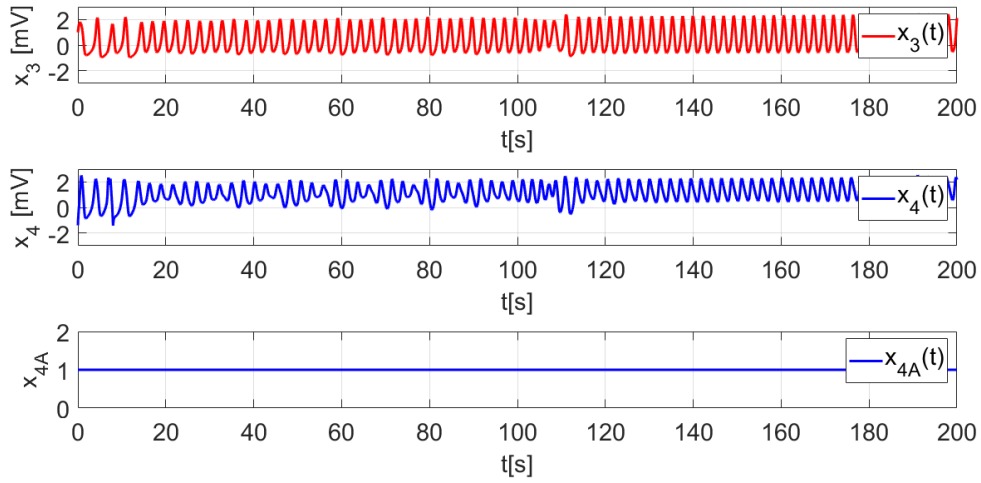


Figura 4.22: Evolución de los potenciales de membrana x_3 , x_4 y el potencial de membrana x_{4A} , que actúan en el robot 1.

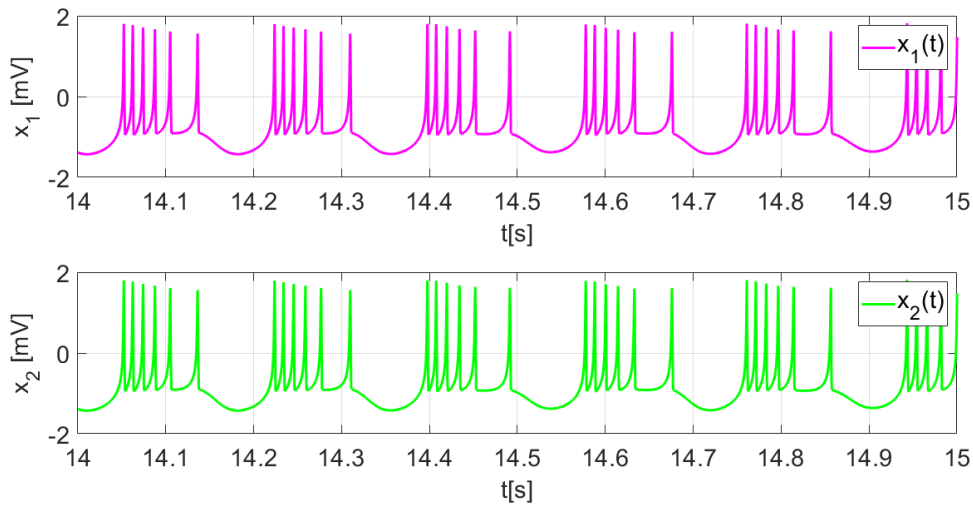


Figura 4.23: Zoom en la evolución de los potenciales de membrana x_1 , x_2 , que actúan en el robot 2.

rueda izquierda.

En la Figura 4.23 se puede apreciar un zoom en la evolución de los potenciales de membrana x_1 y x_2 . Se puede notar como en ese instante x_1 y x_2 se encuentran sincronizadas, esto significa que el robot 2 posee la misma

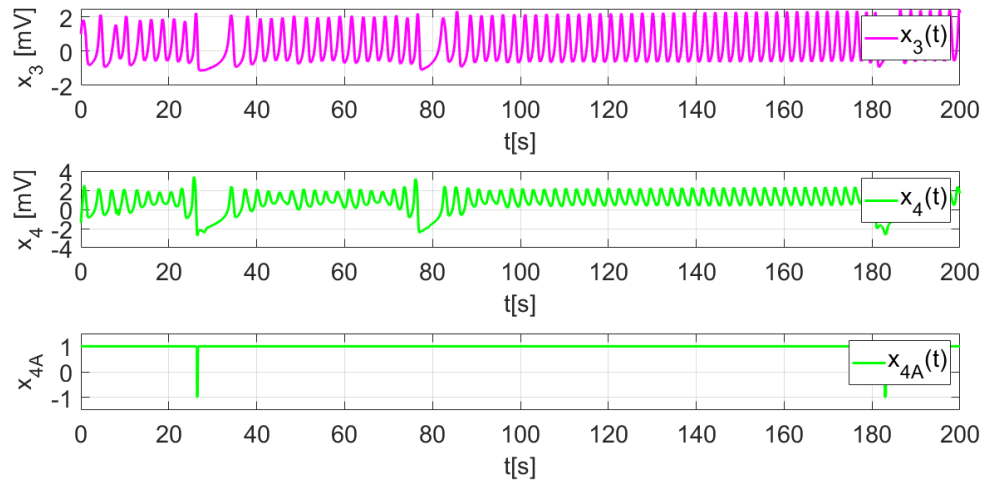


Figura 4.24: Evolución de los potenciales de membrana x_3 , x_4 y el potencial de membrana x_{4A} , que actúan en el robot 2.

velocidad en sus ruedas izquierda y derecha y se desplaza en línea recta. La Figura 4.24 muestra los potenciales de membrana x_3 , x_4 y el potencial de membrana acondicionado x_{4A} del robot 2. En ellos se aprecia como x_3 y x_4 pasan de un comportamiento tipo espiguelo a un comportamiento tipo relajación completa en un breve periodo de tiempo (tiempo cercano a los 25[s] y otro similar ocurre a los 185[s]), esto ocasiona que x_{4A} tome un valor unitario negativo y cambie el sentido de giro de la rueda derecha para efectuar una evasión frontal (cabe mencionar que a lo largo de esta simulación, el robot 2 realizó dos evasiones frontales).

4.6. Simulación 4

Esta simulación se realizó en el ambiente de Gazebo mediante la co-simulación entre Gazebo y Simulink. La simulación consiste en colocar a un robot móvil programado únicamente con el control basado en el modelo neuronal, en un espacio cerrado conocido como “turtlebot3 plaza”, ver Figura 4.25. La tarea es navegar de forma autónoma alrededor de todo el mapa sin colisionar. Con este experimento se busca poner a prueba el esquema de evasión basado en el modelo neuronal, ante un entorno más realista donde se consideran factores como gravedad y fricción.

La condición inicial del robot, así como los parámetros usados se muestran en las Tablas 4.9 y 4.10, respectivamente. Además, se considera un tiempo

de muestreo de 0.001[s].

Condición inicial			
	$x(0)$	$y(0)$	$\theta(0)$
robot	2.0[m]	0.0[m]	0.0[rad]

Tabla 4.9: Condición inicial del robot para la simulación 4.

Parámetros del control basado en el modelo neuronal						
a_1	b_1	c_1	I_{min}	I_{max}	x_{min}	x_{max}
2.2	1.4	0.9	3.25	10	0.4[m]	5.0[m]
a	b	c	d	s	x_0	r
1.0	3.0	1.0	5.0	4.0	-1.6	0.005
ϵ	$\gamma_1 = \alpha$	$\gamma_2 = k$	I	x_{fmin}	p	r_r
0.5	1.0	10.0	3.281	0.5[m]	1.7	33.0[mm]
b_r	V_{max}	W_{max}	r_d			
160.0[mm]	0.22[m/s]	2.84[rad/s]	-0021			

Tabla 4.10: Sintonización de parámetros del control basado en el modelo neuronal para el robot. Simulación 4.

La sintonización de los parámetros es obtenida de forma similar como se hizo en la simulación 2.

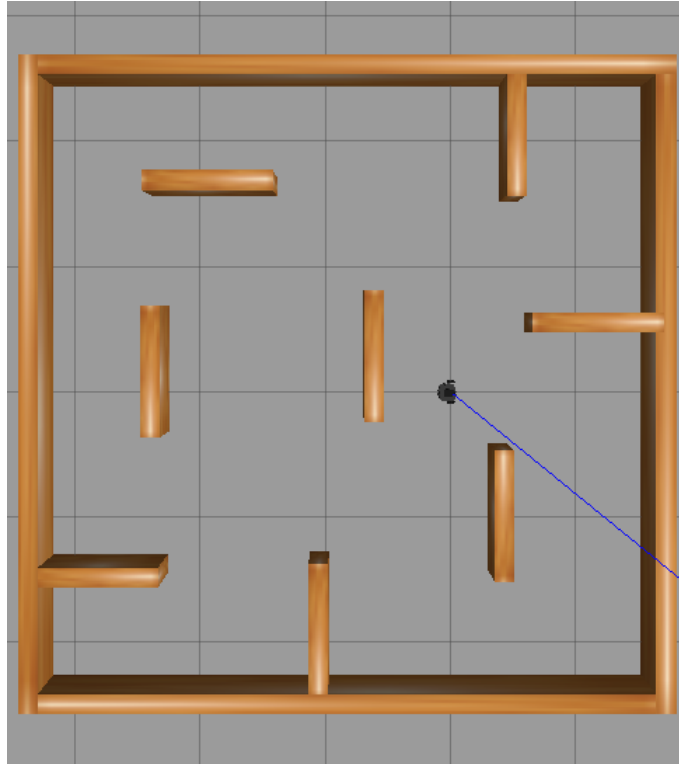


Figura 4.25: Trayectoria del robot móvil.

4.6.1. Resultados de la simulación 4

En la Figura 4.26 se puede observar la evolución de la trayectoria del robot móvil (trayectoria color azul), cuya animación se presenta en [37].

Las señales de las velocidades de rueda ω_{l_speed} y ω_{r_speed} se muestran en la Figura 4.27, donde se puede observar como en la mayor parte del tiempo, existe una variación entre ellas, es decir, una velocidad de rueda suele ser mayor que la otra. Esta diferencia entre las velocidades de rueda izquierda y derecha puede ser grande o pequeña y depende de la detección de las paredes, la cual suele ser constante debido a la estructura del mapa. En algunos pequeños intervalos de tiempo, se puede apreciar como la velocidad en ambas ruedas es la misma y esto le permite al robot móvil desplazarse en línea recta. Los gráficos de la Figura 4.28 corresponden a un zoom de los potenciales de membrana x_1 y x_2 , se puede notar como no hay sincronización en la mayor parte de ese intervalo de tiempo, lo cual coincide con la respuesta de las velocidades de rueda, ya que al existir diferencia entre los potenciales de membrana ocasiona que una velocidad de rueda sea mayor que la otra, esto con el fin de tomar acciones de evasión. Por otro lado, en la Figura

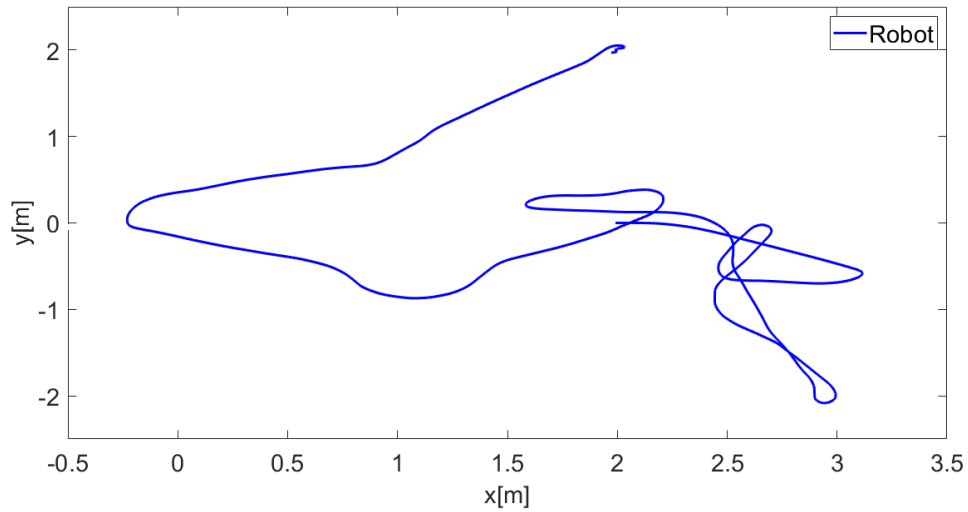


Figura 4.26: Trayectoria del robot móvil.

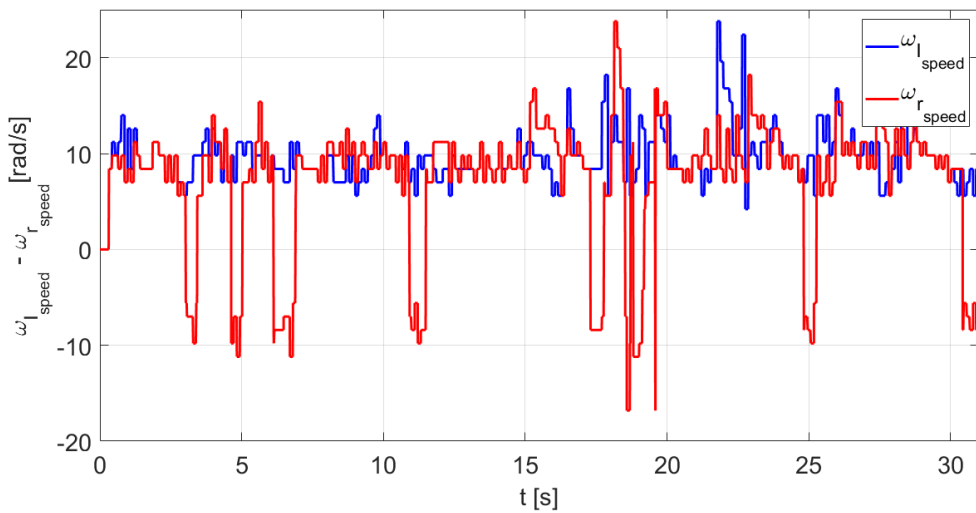


Figura 4.27: Velocidades de rueda $\omega_{l_{speed}}$ y $\omega_{r_{speed}}$.

4.29 se muestran los potenciales de membrana x_3 y x_4 , en ellos se ve como sus comportamientos aparentan ser de relajación completa, sin embargo, no lo son, estos comportamientos reflejan como comienza la transición de un comportamiento de espiguelo a relajación completa y viceversa, y se mantiene así ya que en este experimento se tienen varias detecciones frontales casi continuas (o que la distancia captada por el sensor frontal es cercana al

CAPÍTULO 4. RESULTADOS DE SIMULACIÓN NUMÉRICA

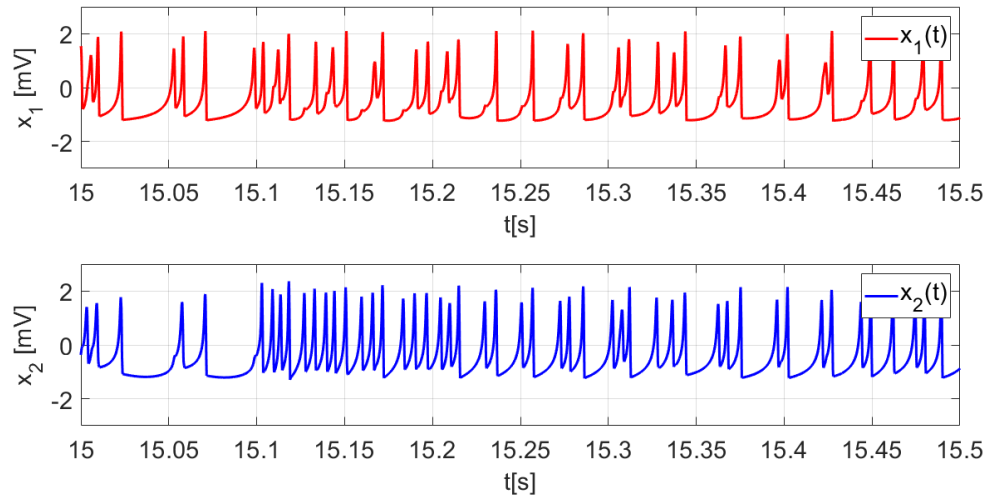


Figura 4.28: Zoom en la evolución de los potenciales de membrana x_1, x_2 , que actúan en el robot móvil.

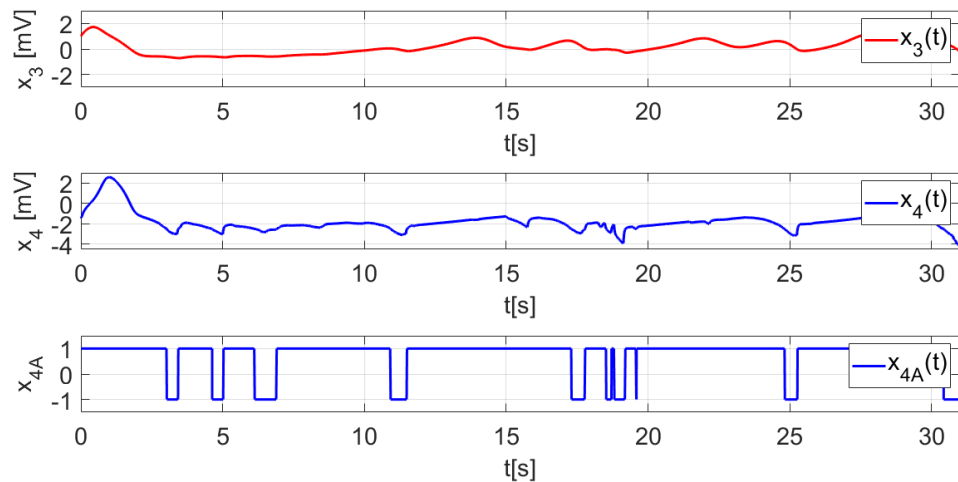


Figura 4.29: Evolución de los potenciales de membrana x_3, x_4 y el potencial de membrana x_{4A} , que actúan en el robot móvil.

de la distancia umbral x_{fmin}). De igual forma estas detecciones frontales influyen en el comportamiento de x_{4A} y por consecuencia, en el sentido del giro de la rueda derecha para realizar evasiones. En general se aprecia como el robot móvil realizó alrededor de diez evasiones frontales a lo largo de toda

la simulación.

4.7. Simulación 5

Esta simulación se realizó en el ambiente de Gazebo mediante la co-simulación entre Gazebo y Simulink. La simulación consiste en colocar a un robot móvil programado con el control híbrido, en un espacio abierto. Su tarea es navegar de forma autónoma, desarrollar una trayectoria deseada (en este caso un círculo de radio 2.0[m]) y efectuar una acción de evasión ante un obstáculo estático que se ubica sobre la trayectoria deseada. Con este experimento se busca ilustrar el desempeño del esquema de control híbrido, pero en un entorno más realista donde se consideran factores como gravedad y fricción.

Para conseguir la trayectoria circular, se recurre a las ecuaciones paramétricas del círculo, ver Ec.(4.2). Los parámetros usados, así como la condición inicial del robot se muestran en las Tablas 4.11, 4.12 y 4.13. Además, se considera un tiempo de muestreo de 0.001[s].

Parámetros del control de trayectoria deseada para el robot				
r_r	b_r	r_c	ω_{sc}	$k_1 = k_2 = k_3$
33.0[mm]	160.0[mm]	2[m]	0.0770	0.4

Tabla 4.11: Sintonización de parámetros para el control de seguimiento de trayectoria del robot. Simulación 5.

Parámetros del control basado en el modelo neuronal						
a_1	b_1	c_1	I_{min}	I_{max}	x_{min}	x_{max}
2.2	1.4	0.9	3.25	10	0.4[m]	5.0[m]
a	b	c	d	s	x_0	r
1.0	3.0	1.0	5.0	4.0	-1.6	0.005
ϵ	$\gamma_1 = \alpha$	$\gamma_2 = k$	I	x_{fmin}	p	r_d
0.5	1.0	10.0	3.281	0.5[m]	1.7	-0021
V_{max}	W_{max}					
0.22[m/s]	2.84[rad/s]					

Tabla 4.12: Sintonización de parámetros del control basado en el modelo neuronal para los robots. Simulación 5.

Condición inicial			
	$x(0)$	$y(0)$	$\theta(0)$
robot	2.0[m]	0.0[m]	$\frac{\pi}{2}$ [rad]

Tabla 4.13: Condición inicial del robot para la simulación 5.

La sintonización de los parámetros es obtenida de forma similar como se hizo en la simulación 2.

4.7.1. Resultados de la simulación 5

En la Figura 4.30 se puede observar la evolución de la trayectoria del robot móvil (trayectoria color azul). Se puede apreciar que posterior a la evasión del obstáculo estático, el robot móvil tiende a la trayectoria deseada (trayectoria color rojo). En [38] se añade un hipervínculo, el cual dirige hacia un video que muestra de forma animada, la evolución de la trayectoria del robot móvil. Las señales de las velocidades de rueda ω_{l_speed} y ω_{r_speed} se muestran en la

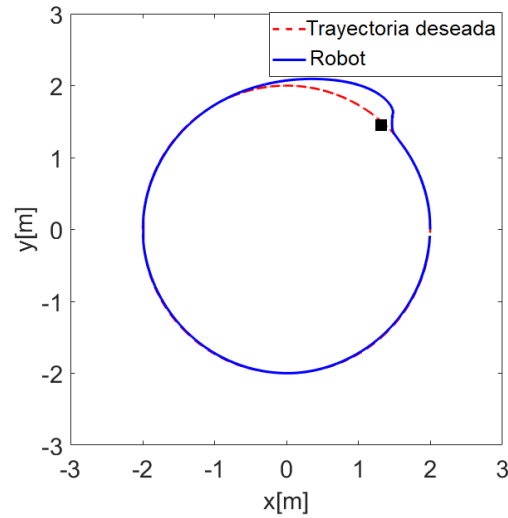


Figura 4.30: Trayectoria del robot móvil.

Figura 4.31, donde se puede observar como las velocidades de las ruedas izquierda y derecha son similares, esto significa que el robot es operado por el control de seguimiento de trayectoria y tiende a la trayectoria deseada, la velocidad de la rueda derecha es ligeramente mayor que la velocidad de la rueda izquierda para que el robot móvil genere la trayectoria circular. Sin

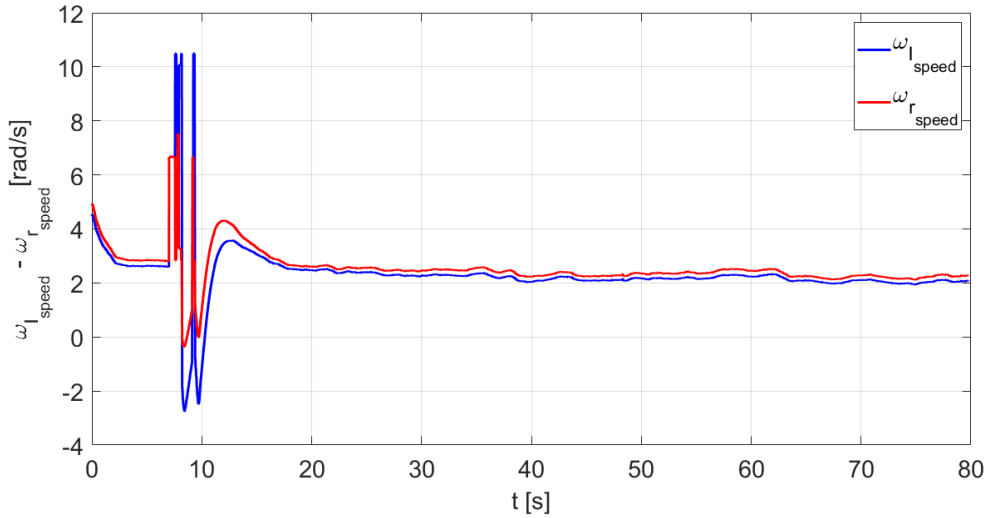
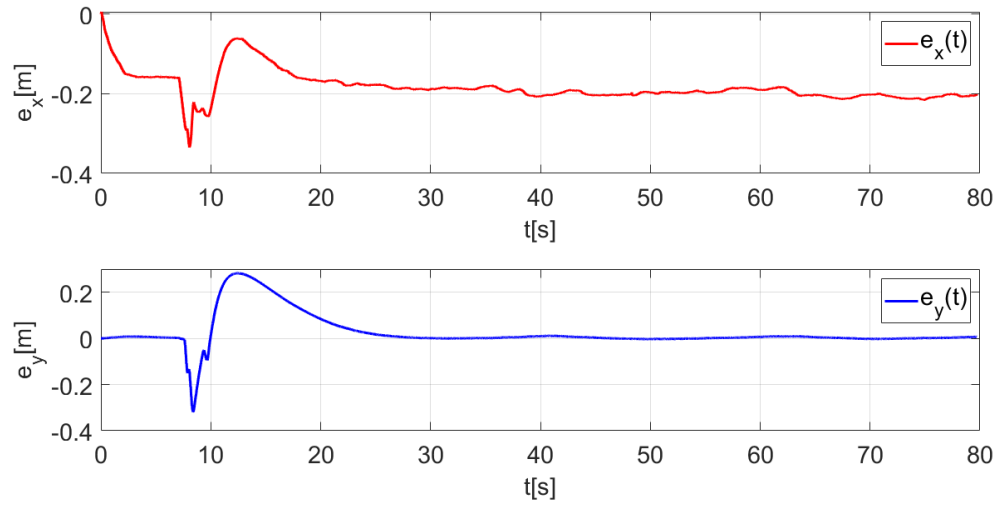
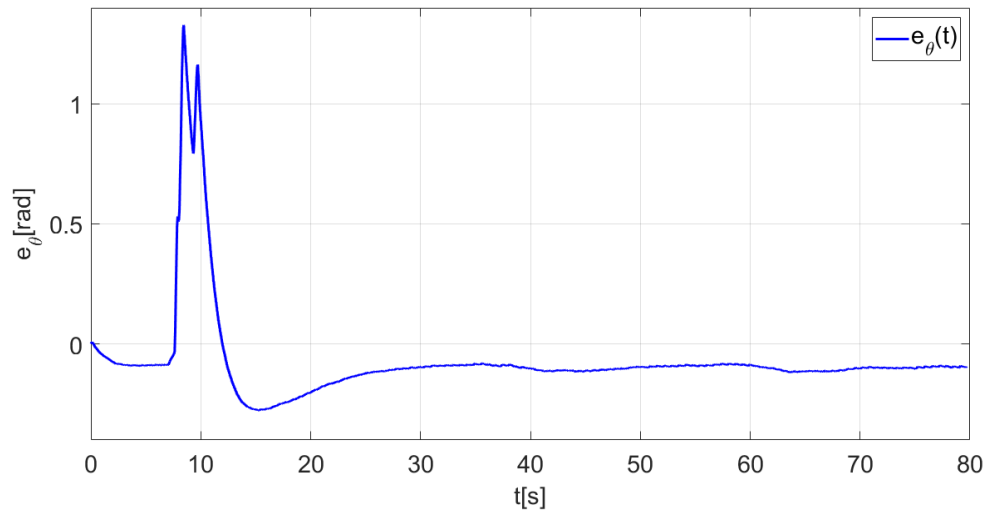


Figura 4.31: Velocidades de rueda ω_{l_speed} y ω_{r_speed} .

embargo, cuando el robot móvil detecta el obstáculo estático, las velocidades de rueda comienzan a variar, siendo una mayor que la otra para efectuar giros de evasión, es decir el robot es operado con la estrategia de evasión basada en el modelo neuronal y no tiende a la trayectoria deseada. Posterior a la evasión del obstáculo estático, se nota como el comportamiento de las velocidades de rueda vuelve a ser como el que se tuvo en el comienzo. Los gráficos de la Figura 4.32a corresponden a los errores de posición en x e y , se puede notar como ambos tienden al valor cero, especialmente, el error en x se aleja un poco más y posee un error de aproximadamente $0.2[m]$, sin embargo, el error es relativamente pequeño tal que, no impide que el robot móvil desarrolle la trayectoria deseada. El error de posición en θ mostrado en la Figura 4.32b, también tiende a cero en el comienzo, esto indica que el robot móvil tiende a su trayectoria deseada. El seguimiento de trayectoria deseada continua hasta que el robot móvil detecta al obstáculo estático en un tiempo aproximado a $9[s]$, esto ocasiona que los errores de posición difieran de cero, indicando que se están tomando acciones de evasión. Posterior a la evasión del obstáculo, los errores vuelven a tender a cero. Una forma más clara de notar los resultados aportados por el control basado en el modelo neuronal, es observando los gráficos de la Figura 4.33, que corresponden a los potenciales de membrana x_1 y x_2 , en estos se aprecia como en el comienzo se tiene una sincronización en las señales x_1 y x_2 , esto quiere decir que el robot móvil posee la misma velocidad en ambas ruedas y le permite desplazarse en línea recta. Por otra parte, hay intervalos de tiempo donde se nota como desaparece la sincronización



(a) Errores de posición en x e y .



(b) Error de posición en θ .

Figura 4.32: Error de posición.

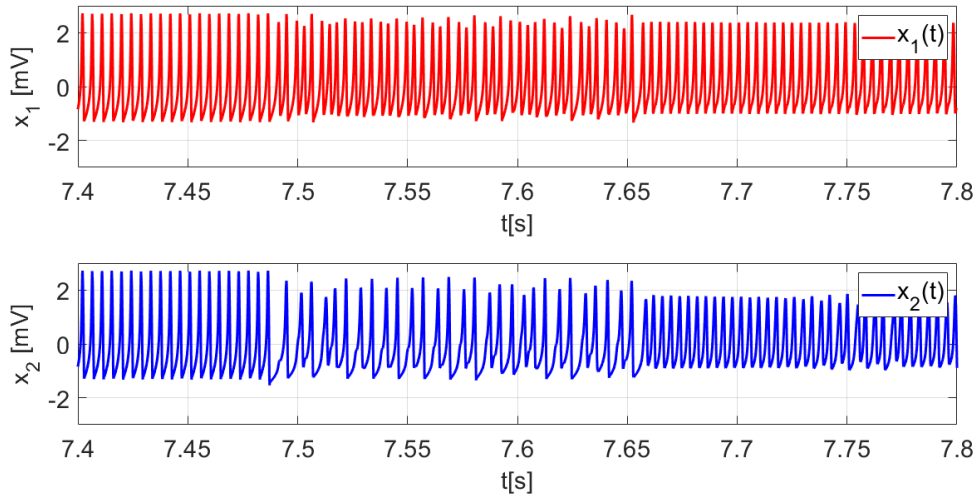


Figura 4.33: Zoom en la evolución de los potenciales de membrana x_1, x_2 , que actúan en el robot móvil.

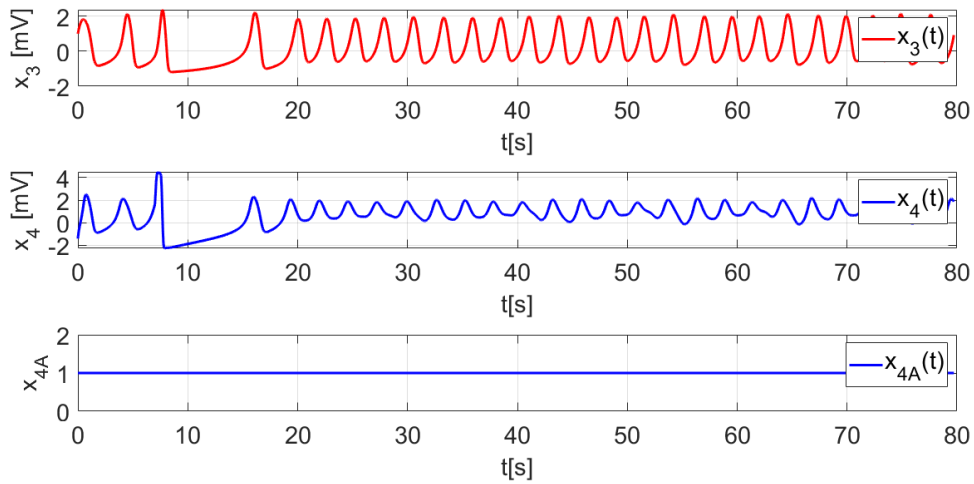


Figura 4.34: Evolución de los potenciales de membrana x_3, x_4 y el potencial de membrana x_{4A} , que actúan en el robot móvil.

CAPÍTULO 4. RESULTADOS DE SIMULACIÓN NUMÉRICA

y hay un mayor número de espigues en un potencial de membrana que en otro, lo cual significa que una velocidad de rueda es mayor que la otra y se efectúa un giro, que puede ser de evasión o debido a la trayectoria deseada. En la Figura 4.34 se muestran los potenciales de membrana x_3 y x_4 , en ellos se ve como sus comportamientos a lo largo de toda la simulación son del tipo espigues, esto implica que, x_{4A} mantiene un valor unitario positivo, y el sentido de giro de la rueda derecha siempre es el mismo que el de la rueda izquierda, es decir, el robot no realizó ninguna evasión frontal a lo largo de toda la simulación.

Capítulo 5

Conclusiones y trabajo futuro

Tanto el control híbrido como el control de evasión basado en el modelo neuronal, han mostrado un resultado aceptable. Otros controles requieren un conocimiento completo o parcial del entorno para generar una planificación previa de una ruta, y así conseguir una navegación autónoma sin colisiones (donde solo se consideran obstáculos estáticos), sin embargo, el esquema tratado en este trabajo le permite al robot móvil navegar de forma autónoma mientras evade obstáculos estáticos o dinámicos sin el conocimiento del entorno. Otro aspecto es que, en otros esquemas de este tipo, el robot móvil se detiene cuando detecta un obstáculo frontal, pero una ventaja de este trabajo es la estrategia de evasión frontal que permite al robot móvil navegar continuamente.

Los resultados de este trabajo dependen de una correcta sintonización de los parámetros del modelo. Además, el diseño de la estrategia de evasión se ha realizado considerando que las neuronas son idénticas. La distancia umbral x_{fmin} permite al robot móvil detectar obstáculos frontales, si este valor se reduce, entonces, el robot móvil puede explorar más área del entorno, pero, probablemente, el robot móvil requiere una mayor velocidad para evitar los obstáculos. Hablando de la distancia de umbral d_c , esta permite conmutar los controles cuando se detecta un obstáculo, por lo que, si este valor se disminuye, entonces, el robot móvil también requiere una mayor velocidad para evitar los obstáculos. De lo contrario, si el valor d_c se incrementa, entonces, el robot móvil puede detectar fácilmente obstáculos y áreas rectangulares para poder evitarlos (esto es útil cuando el robot móvil sólo está programado con el control basado en el modelo neuronal). Sin embargo, si el robot móvil tiene que seguir una trayectoria deseada, con un valor d_c mayor el robot móvil deja

CAPÍTULO 5. CONCLUSIONES Y TRABAJO FUTURO

la trayectoria deseada más fácil si detecta un obstáculo que tiene que evitar.

Las ganancias del control de seguimiento de trayectoria k_1 , k_2 y k_3 permiten obtener una convergencia más rápida o más lenta a la trayectoria deseada. En particular, si las ganancias de k_1 , k_2 y k_3 son pequeñas, el robot móvil podrá converger a la trayectoria deseada lentamente, sin embargo, con esto puede evitar los obstáculos fácilmente.

Acerca de la elección de la ventana de tiempo, esta se realiza con base en el periodo mínimo que hay entre dos picos sucesivos, y tal que la entrega de la medición del número de picos sea lo suficientemente rápida para que el robot móvil cambie su velocidad y así pueda efectuar una evasión.

Por otro lado, al comparar los resultados de simulación numérica obtenidos en los ambientes de Matlab-Simulink y Gazebo, se puede notar un cambio en las respuestas del robot móvil. La diferencia más clara se presenta en la respuesta de las velocidades de rueda, ya que las obtenidas en el ambiente de Matlab-Simulink se aprecian con menor ruido o perturbaciones, además, los errores de posición en el control híbrido son relativamente menores que los obtenidos en el ambiente de Gazebo, esto se debe a los factores que se consideran en el entorno de Gazebo como lo son la gravedad, la fricción y la dinámica no modelada.

Como trabajo futuro se plantea implementar el mismo esquema de evasión de manera experimental. Por otro lado, se quiere probar la implementación del modelo neuronal propuesto por Izhikevich y comparar sus resultados con el modelo Hindmarsh-Rose que actualmente se tiene implementado. Además, se propone emplear solo una neurona para establecer la estrategia de evasión frontal, tomando a su corriente externa como el parámetro que gobierne esta estrategia de evasión, y comparar sus resultados con la propuesta actual.

Bibliografía

- [1] J. A. J. Lora y A. R. Angeles, “Formación autónoma bio-inspirada para robots diferenciales,” *Congreso Nacional de Control Automático CNCA*, pp. 546–551, 2019.
- [2] A. O. Baturone, *Robótica: manipuladores y robots móviles*. Marcombo, 2005.
- [3] R. Siegwart, I. R. Nourbakhsh, y D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.
- [4] V. R. B. Sotelo, J. R. G. Sánchez, y R. S. Ortigoza, “Robots móviles: Evolución y estado del arte,” *Polibits*, no. 35, pp. 12–17, 2007.
- [5] D. D. y M. J. L. Bruemmer, David J y Dudenhoeffer, “Dynamic-autonomy for urban search and rescue.” en *AAAI mobile robot competition*. Menlo Park, CA, 2002, pp. 33–37.
- [6] B. y B. G. de Wit, Carlos Canudas y Siciliano, *Theory of robot control*. Springer Science & Business Media, 2012.
- [7] R. Rincon-Jara, R. Ambrosio, y J. Mireles, “Análisis y caracterización de un acelerómetro capacitivo fabricado con tecnología polymump’s,” *Superficies y vacío*, vol. 23, no. 3, pp. 26–31, 2010.
- [8] A. H. Martha, I. Joaquín, y A. Bautista Miguel, “Diseño y control de robots móviles,” *Nuevo Laredo: Instituto Tecnológico de Nuevo Laredo*, 2007.
- [9] J. Cubillos Martínez, *Modelos de neuronas artificiales en software para su uso en preparaciones de electrofisiología*. Tesis de Maestría, Universidad Autónoma de Madrid, Escuela Politécnica Superior, Departamento de Ingeniería Informática, 2016.

BIBLIOGRAFÍA

- [10] Á. F. J. Fernández, *Diseño y evaluación de sistemas de control y procesamiento de señales basadas en modelos neuronales pulsantes*. Tesis Doctoral, Universidad de Sevilla, Escuela Técnica Superior de Ingeniería Informática, 2010.
- [11] Y. Marín Mas, *Determinación de parámetros de modelos de neuronas*. Tesis de Maestría, Universidad de las Islas Baleares, Instituto de Física Interdisciplinar y Sistemas Complejos (IFISC), Departamento de Física, 2018.
- [12] L. Carrillo-Reid y J. Bargas, “Codificación de estados funcionales en redes neuronales biológicas,” *TIP Revista Especializada en Ciencias Químico-Biológicas*, vol. 11, no. 1, pp. 52–59, 2008.
- [13] G. G. Velasco Equihua y J. P. Ramirez, “Synchronization of Hindmarsh-Rose neurons via Huygens-like coupling,” *IFAC-PapersOnLine*, vol. 51, no. 33, pp. 186–191, 2018, 5th IFAC Conference on Analysis and Control of Chaotic Systems CHAOS 2018. [En línea]. Disponible: <https://www.sciencedirect.com/science/article/pii/S2405896318336590>
- [14] T. Steur, Erik y Vromen y H. Nijmeijer, “Adaptive training of neural networks for control of autonomous mobile robots,” en *Sensing and Control for Autonomous Vehicles*. Springer, 2017, pp. 387–405.
- [15] E. Vromen, TGM y Steur y H. Nijmeijer, “Training a network of electronic neurons for control of a mobile robot,” *International Journal of Bifurcation and Chaos*, vol. 26, no. 12, p. 1650196, 2016.
- [16] M. Hank y M. Haddad, “A hybrid approach for autonomous navigation of mobile robots in partially-known environments,” *Robotics and Autonomous Systems*, vol. 86, pp. 113–127, 2016. [En línea]. Disponible: <https://www.sciencedirect.com/science/article/pii/S0921889015302992>
- [17] J. Velagic, B. Lacevic, y B. Perunicic, “A 3-level autonomous mobile robot navigation system designed by using reasoning/search approaches,” *Robotics and Autonomous Systems*, vol. 54, no. 12, pp. 989–1004, 2006. [En línea]. Disponible: <https://www.sciencedirect.com/science/article/pii/S0921889006000947>
- [18] M. S. Gharajeh y H. B. Jond, “Hybrid global positioning system-adaptive neuro-fuzzy inference system based autonomous mobile robot navigation,” *Robotics and Autonomous Systems*, vol. 134, p. 103669, 2020. [En línea]. Disponible: <https://www.sciencedirect.com/science/article/pii/S0921889020305091>

- [19] E. Baklouti, N. B. Amor, y M. Jallouli, “Reactive control architecture for mobile robot autonomous navigation,” *Robotics and Autonomous Systems*, vol. 89, pp. 9–14, 2017. [En línea]. Disponible: <https://www.sciencedirect.com/science/article/pii/S0921889016305413>
- [20] L. MUTLU y E. UYAR, “Control and navigation of an autonomous mobile robot with dynamic obstacle detection and adaptive path finding algorithm,” *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 355–360, 2012, 10th IFAC Symposium on Robot Control. [En línea]. Disponible: <https://www.sciencedirect.com/science/article/pii/S1474667016336369>
- [21] R. Chatterjee y F. Matsuno, “Use of single side reflex for autonomous navigation of mobile robots in unknown environments,” *Robotics and Autonomous Systems*, vol. 35, no. 2, pp. 77–96, 2001. [En línea]. Disponible: <https://www.sciencedirect.com/science/article/pii/S092188900000124X>
- [22] P. K. Panigrahi y S. K. Bisoy, “Localization strategies for autonomous mobile robots: A review,” *Journal of King Saud University - Computer and Information Sciences*, 2021. [En línea]. Disponible: <https://www.sciencedirect.com/science/article/pii/S1319157821000550>
- [23] G. Alvarez y O. Flor, “Desempeño en métodos de navegación autónoma para robots móviles,” *Minerva*, vol. 1, no. 2, pp. 19–29, 2020.
- [24] D. López, F. Gómez-Bravo, F. Cuesta, y A. Ollero, “Planificación de trayectorias con el algoritmo rrt. aplicación a robots no holónomos,” *Revista Iberoamericana de Automática e Informática Industrial*, vol. 3, no. 3, pp. 56–67, 2006.
- [25] L. Zhang, Y. Zhang, y Y. Li, “Path planning for indoor mobile robot based on deep learning,” *Optik*, vol. 219, p. 165096, 2020.
- [26] M. Mora y J. Tornero, “Planificación de movimientos mediante la propagación de campos potenciales artificiales,” en *8º Congreso Iberoamericano de Ingeniería Mecánica, e-book*, 2007.
- [27] F. A. Cosío y M. P. Castañeda, “Autonomous robot navigation using adaptive potential fields,” *Mathematical and computer modelling*, vol. 40, no. 9-10, pp. 1141–1156, 2004.
- [28] A. Lazarowska, “Discrete artificial potential field approach to mobile robot path planning,” *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 277–282, 2019.

BIBLIOGRAFÍA

- [29] A. Melingui, R. Merzouki, J. B. Mbede, y T. Chettibi, “A novel approach to integrate artificial potential field and fuzzy logic into a common framework for robots autonomous navigation,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 228, no. 10, pp. 787–801, 2014.
- [30] F. M. Caballero Flores, “Sincronización de modelos matemáticos de neuronas mediante acoplamientos basados en umbrales,” Tesis de maestría, 2019.
- [31] D. Fan y Q. Wang, “Synchronization and bursting transition of the coupled hindmarsh-rose systems with asymmetrical time-delays,” *Science China Technological Sciences*, vol. 60, no. 7, pp. 1019–1031, 2017.
- [32] D. Hrg, “Synchronization of two Hindmarsh–Rose neurons with unidirectional coupling,” *Neural Networks*, vol. 40, pp. 73–79, 2013.
- [33] Std. [En línea]. Disponible: <https://emanual.robotis.com/docs/en/platform/turtlebot3/features/>
- [34] Std. [En línea]. Disponible: <https://youtu.be/LEuaK2trGRY>
- [35] Std. [En línea]. Disponible: <https://youtu.be/z2dJJwt1FTU>
- [36] Std. [En línea]. Disponible: <https://youtu.be/fEByEFvW8ec>
- [37] Std. [En línea]. Disponible: <https://youtu.be/clpdSUpeLjY>
- [38] Std. [En línea]. Disponible: <https://youtu.be/pomPbigH6G4>

Anexos

Navegación autónoma de un robot móvil en un ambiente desconocido usando una red de neuronas tipo Hindmarsh-Rose

Enrique Martinez Sanchez
Electrical Engineering Department
CINVESTAV-IPN
Mexico City, Mexico
e-mail: enrique.mtz@cinvestav.mx

Jonatan Pena Ramirez
Electronics and Telecommunications Department
CICESE
Ensenada, Mexico
e-mail: jpena@cicese.mx

Alejandro Rodriguez Angeles
Electrical Engineering Department
CINVESTAV-IPN
Mexico City, Mexico
e-mail: aangeles@cinvestav.mx

Resumen—En este trabajo se propone un control híbrido para la navegación autónoma de un robot móvil en un espacio con obstáculos estáticos y dinámicos. Este control híbrido está conformado por la conmutación entre un control que permite el seguimiento de trayectoria y un control de evasión basado en neuronas tipo Hindmarsh-Rose (HR), interconectadas mediante un acoplamiento difuso. La conmutación se da en función de la lectura de los sensores integrados al robot y se opera a nivel de velocidad de rueda. Para que se dé la evasión de obstáculos, se utiliza una fuerza de acoplamiento pequeña, lo cual permite aprovechar la desincronización de los potenciales de membrana ante entradas de corriente externas diferentes para cada neurona, de esta manera se obtiene una mayor sensibilidad ante alguna detección. Por otra parte, se hace uso de un acoplamiento dinámico para establecer una estrategia de evasión frontal. Para mostrar el desempeño del control se presentan resultados de simulación.

Index Terms—Neuronas Hindmarsh-Rose (HR), acoplamiento difuso, acoplamiento dinámico, control basado en el modelo neuronal, espigues

I. INTRODUCCIÓN

Existen diversas aplicaciones que pueden desempeñar los robots móviles autónomos: el transporte de material, limpieza, vigilancia, localización de objetos, exploración no tripulada, entre otros [1]–[11]. La navegación autónoma de los robots móviles puede realizarse por medio de la planificación previa de una ruta, esto suele realizarse en ambientes totalmente conocidos y que no cambian con el paso del tiempo, en estas condiciones se puede probar la existencia de un camino libre de colisiones y posiblemente optimo hacia la meta (en términos de distancia, tiempo y energía empleada) [5]–[7]. Desafortunadamente, el caso anterior es poco común en muchas aplicaciones donde se considera un entorno real, en las que el ambiente es desconocido. Este hecho requiere que un robot móvil sea capaz de identificar y evadir obstáculos (estáticos y dinámicos), realizar autolocalización, tomar decisiones, etc. [1], [2], [4], [11]. Dentro de la amplia gama de controles que abordan la problemática de la navegación autónoma de un robot móvil, se hallan controles con enfoques del tipo híbrido, que son conformados por controladores para la navegación reactiva y controladores para un seguimiento de

trayectoria [5]–[9]. Ejemplo de ello, son los controladores de sistemas de interferencia difusa que requieren de un entorno parcialmente conocido y se conmutan de la forma más suave posible, para elaborar una fase de preprocesamiento dependiente a una tarea específica y así poder construir una ruta de referencia óptima en tiempo [5]. Por otra parte, está la combinación entre controles PI, y controladores de lógica difusa con algoritmos de retroceso que actúan de forma simultánea, y se encargan de establecer una fusión de datos obtenidos de sensores, construcción de mapas y tareas de planificación de rutas [6]. Para imitar comportamientos más naturales, se han desarrollado algunos controladores bio-inspirados los cuales poseen un radio de confort y un lazo externo que contiene fuerzas de evasión [4], además, están aquellos controles que imitan el manejo de automóviles impulsados por una respuesta humana basados en lógica difusa, acciones reflexivas y toma de decisiones de alto nivel [10]. También, se hallan controles que poseen una intercomunicación inalámbrica entre una computadora móvil (que se incorpora al robot y se encarga de recopilar datos del entorno mediante sensores laser), y una computadora estacionaria que se encarga de elaborar un algoritmo de control para la evasión de objetos dinámicos y la búsqueda de una ruta adaptativa [9]. Otro aspecto importante a resaltar, es la variedad de sensores que se pueden integrar, ya que estos proporcionan al robot móvil la información necesaria para poder realizar sus tareas, entre más sofisticados sean estos sensores, mejor percepción tendrá el robot de su entorno, sin embargo, estos sensores también deben seleccionarse de acuerdo con el tipo de controlador a utilizar (infrarrojos, Lidar, ultrasonicos, Bluetooth, cámaras, módulos GPS, módulos ZigBee, entre otros) [1]–[11]. Actualmente, una de las propuestas más atractivas de control para la navegación autónoma de robots móviles es el uso de un “cerebro artificial”, el cual está formado por redes de neuronas. Los candidatos para estos cerebros artificiales son las redes de neuronas tipo spiking (espigues), interconectadas entre sí, y cuya respuesta en los potenciales de membrana de cada una de ellas es del tipo spiking, imitando de esa forma la

arquitectura de un cerebro biológico [1], [2]. Lo interesante en un modelo neuronal y la forma en que se interconecta, es la variedad de comportamientos y patrones de espiguo que se pueden generar en su respuesta, estos patrones se pueden interpretar y asociar con una tarea o acción específica (tres comportamientos principales que se pueden obtener son conocidos como spiking, bursting y chaotic bursting) [1]–[3].

En este trabajo, se propone utilizar el modelo neuronal Hindmarsh-Rose (HR), puesto que produce una respuesta adecuada para la idea que se tiene como control de evasión, además, su implementación en electrónica analógica es relativamente simple. Se contemplan cuatro neuronas tipo HR, dos de ellas son interconectadas mediante un acoplamiento difuso [2], y las dos restantes interactúan a través de un acoplamiento dinámico, el cual ofrece una mejor sincronización y permite operar en un modo de relajación completa [3].

Las contribuciones que presenta este artículo consisten en reducir el número de neuronas que permiten la evasión lateral, y en agregar tres parámetros que permiten ajustar de mejor forma las corrientes externas del modelo neuronal que modulan la frecuencia del espiguo en los potenciales de membrana, con el fin de hacer más preciso el ajuste del giro de evasión. Además, se propone una estrategia de evasión frontal utilizando dos neuronas interconectadas a través de un acoplamiento dinámico, que favorece un modo de relajación completa. Por otra parte, se añade la implementación de un control de seguimiento de trayectoria que junto con las estrategias de evasión, le permite al robot móvil establecer una navegación autónoma, en donde sigue una trayectoria deseada mientras evade obstáculos.

La Sección II de este artículo expone el planteamiento del problema y muestra el modelo cinemático del robot móvil a utilizar. En la Sección III se implementa una estrategia de control para el seguimiento de trayectoria. La Sección IV muestra el control de evasión basado en neuronas tipo HR. En la Sección V se expone el control híbrido. La Sección VI muestra los resultados de una simulación numérica, así como una discusión de los mismos. Finalmente, en la Sección VII se presentan las conclusiones y trabajo a futuro.

II. MODELO Y PLANTEAMIENTO DEL PROBLEMA

II-A. Planteamiento del problema

Se quiere hacer navegar a un robot móvil tipo 2.0 (robot diferencial) de forma autónoma en un ambiente desconocido. En este trabajo se considera como ambiente desconocido, a un mapa plano (entorno 2D) ya sea abierto o cerrado, en el cual se encuentran obstáculos del tipo estáticos y dinámicos (otros robots móviles). Adicional a ello, se quiere que el robot realice un seguimiento de trayectoria deseada, y si este se encuentra con algún obstáculo en su camino, sea capaz de evadirlo. Para abordar el problema, se pretende implementar

un control híbrido conformado por un control de seguimiento de trayectoria y un control de evasión basado en neuronas tipo HR.

II-B. Modelo cinemático del robot móvil

Considere el robot móvil tipo 2.0 mostrado en la Figura 1, cuyo modelo cinemático está dado por

$$\begin{aligned}\dot{x} &= V \cos(\theta), \\ \dot{y} &= V \sin(\theta), \\ \dot{\theta} &= W,\end{aligned}\quad (1)$$

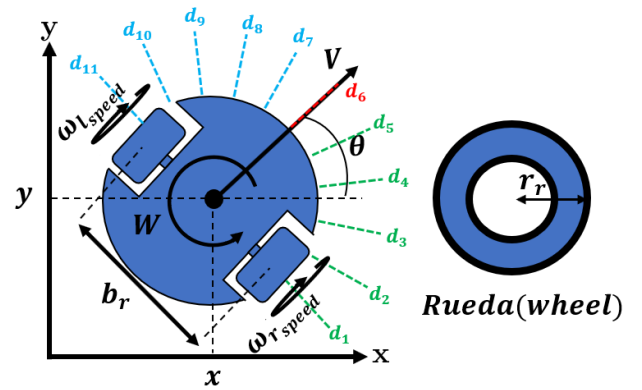


Figura 1. Robot móvil tipo 2.0 y orientación de los sensores.

donde (x, y) es el punto que representa el centro y posición del robot, el ángulo θ es su orientación con respecto al eje horizontal, mientras que V y W son las velocidades traslacional y rotacional, respectivamente, y las posiciones radiales de los sensores están espaciadas por $\frac{\pi}{10}$.

III. ESTRATEGIA DE CONTROL DE SEGUIMIENTO DE TRAYECTORIA

Para poder establecer el seguimiento de trayectoria se opta por la estrategia de un robot virtual de referencia el cual genera la trayectoria deseada, de esta forma el robot móvil se encargará de seguir al robot virtual de referencia.

III-A. Procedimiento de la estrategia de control

De acuerdo con [12], se debe determinar V y W que actúan como entradas de control para el robot móvil.

1) Se establecen las ecuaciones paramétricas de la posición deseada x_d y y_d , sus primeras y segundas derivadas \dot{x}_d, \dot{y}_d y \ddot{x}_d, \ddot{y}_d , respectivamente.

2) Se calculan las velocidades V_r, W_r , y la posición angular θ_r del robot virtual

$$\begin{aligned} V_r &= \sqrt{\dot{x}_d^2 + \dot{y}_d^2}, \\ W_r &= \frac{\dot{x}_d \ddot{y}_d - \dot{y}_d \ddot{x}_d}{V_r^2}, \\ \theta_r &= \tan^{-1} \left(\frac{\dot{y}_d}{\dot{x}_d} \right). \end{aligned} \quad (2)$$

3) Se formula el error de seguimiento

$$e = \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_d - x \\ y_d - y \\ \theta_r - \theta \end{pmatrix}. \quad (3)$$

4) Se propone el control

$$\begin{aligned} u_1 &= -k_1 e_1, \\ u_2 &= -k_2 \tanh(10V_r) e_2 - k_3 e_3. \end{aligned} \quad (4)$$

5) Se sintonizan las ganancias k_1, k_2, k_3 del control

6) Se determina el valor de V y W

$$\begin{aligned} V &= -u_1 + V_r \cos(e_3), \\ W &= W_r - u_2. \end{aligned} \quad (5)$$

7) Finalmente, se obtienen las velocidades angulares de la rueda derecha ω_{rT} e izquierda ω_{lT}

$$\omega_{rT} = \frac{1}{r_r} \left(V + \frac{b_r W}{2} \right), \quad (6)$$

$$\omega_{lT} = \frac{1}{r_r} \left(V - \frac{b_r W}{2} \right), \quad (7)$$

donde r_r es el radio de las ruedas del robot, b_r es la separación entre las ruedas. Note que las Ecs. (6) y (7) permiten operar a nivel de velocidad de rueda.

IV. ESTRATEGIA DE CONTROL DE EVASIÓN BASADA EN NEURONAS TIPO HR

En esta estrategia de evasión se aprovecha la variación de espiguelo en los potenciales de membrana de un primer par de neuronas (HR1 y HR2), que se asocian a las velocidades de cada rueda del robot móvil. También, se contempla una comunicación entre otro par de neuronas (HR3 y HR4), y se utiliza la respuesta del potencial de membrana en HR4 para establecer una evasión frontal. El procedimiento está inspirado en [2] y se muestra a continuación.

1) Se establecen los modelos neuronales [1]–[3]

$$HRi : \begin{cases} \dot{x}_i &= y_i + b x_i^2 - a x_i^3 - z_i + I_i + v_i, \\ \dot{y}_i &= c - d x_i^2 - y_i, \\ \dot{z}_i &= r_i [s(x_i - x_0) - z_i], \end{cases} \quad (8)$$

donde $i = 1, 2, 3, 4$, $x_i \in R$ es el potencial de membrana, $y_i \in R$ es la corriente de recuperación, $z_i \in R$ es la corriente de adaptación, $a, b, c, d, s, r_1 = r_2 = r, r_3 = r_4 = r_d, x_0$

son parámetros constantes. $a \in R$ permite conmutar entre comportamientos de bursting y spiking, además determina la frecuencia del spiking. Por otra parte $r \in R$ gobierna el comportamiento bursting, $s \in R$ controla la adaptación, si este posee valores pequeños (alrededor de 1) provocan un comportamiento spiking rápido. $x_0 \in R$ es el potencial de reposo del sistema que sirve como potencial de referencia, y $b, d, c \in R^+$ son parámetros positivos. I_1, I_2 e $I_3 = I_4 = I \in R$ son las corrientes externas (entradas del modelo), al tener una correcta configuración de los parámetros, se puede variar el valor de la corriente externa, entre mayor sea esta corriente, mayor número de picos se reproducirán en el potencial de membrana y viceversa.

Las neuronas HR1 y HR2, interactúan a través de un acoplamiento difuso:

$$\begin{cases} v_1 &= \varepsilon(y_2 - y_1), \\ v_2 &= \varepsilon(y_1 - y_2), \end{cases} \quad (9)$$

con fuerza de acoplamiento $\varepsilon > 0$.

Las neuronas HR3 y HR4, están interconectadas mediante un acoplamiento dinámico:

$$\begin{cases} v_3 &= h_2, \\ v_4 &= -G h_2, \end{cases} \quad (10)$$

$$\begin{cases} \dot{h}_1 &= -\alpha h_1 + h_2, \\ \dot{h}_2 &= -\gamma_1 h_1 - \gamma_2 h_2 - k(x_3 - x_4), \end{cases} \quad (11)$$

con fuerza de acoplamiento k y parámetros $\alpha, \gamma_1, \gamma_2$.

Finalmente G es una expresión propuesta, del tipo exponencial que está en función del sensor frontal.

$$G = -11e^{-3(d_{f_{ron}} - x_{f_{min}})} + 1. \quad (12)$$

Con la expresión G se obtienen valores que le permiten a x_3 y x_4 cambiar de un comportamiento del tipo espiguelo a un comportamiento de relajación completa y viceversa. $x_{f_{min}}$ es una distancia de umbral que cuando es alcanzada por la lectura del sensor frontal concede el cambio de un comportamiento a otro (esto se detalla de una mejor forma en el paso cinco del procedimiento).

2) Se distribuye equidistantemente la posición radial de 11 sensores tipo lidar, véase la Figura 1.

3) Se obtienen las distancias mínimas captadas por todos los sensores (d), por el grupo de sensores del lado izquierdo (d_{izq}), por el grupo del lado derecho (d_{der}) y la lectura del sensor frontal ($d_{f_{ron}}$).

$$\begin{aligned} d_{der} &= \min(d_1, d_2, d_3, d_4, d_5), \\ d_{izq} &= \min(d_7, d_8, d_9, d_{10}, d_{11}), \\ d_{f_{ron}} &= d_6, \\ d &= \min(d_{izq}, d_{der}, d_{f_{ron}}). \end{aligned} \quad (13)$$

4) Se establecen las expresiones para las corrientes externas I_1, I_2 (entradas de HR1-HR2) en función de la distancia del grupo de sensores derecho e izquierdo.

$$I_j = \begin{cases} a_1 I_{min} & d_j \leq x_{min} \\ b_1 I_{min} \left(\frac{I_{max}}{I_{min}} \right) \frac{d_j - x_{min}}{x_{max} - x_{min}} & x_{min} < d_j < x_{max} \\ c_1 I_{max} & d_j \geq x_{max} \end{cases} \quad (14)$$

donde $j \in \{der, izq\}$, $I_1 = I_{der}$, $I_2 = I_{izq}$. I_{min} e I_{max} son las corrientes externas mínima y máxima, respectivamente, x_{min} y x_{max} son las distancias mínima y máxima, respectivamente, que puede leer el sensor. Los parámetros a_1, b_1, c_1 , se utilizan para un mejor ajuste en la caracterización de las neuronas.

5) Se realiza un conteo de espigas en x_1, x_2 durante una ventana de tiempo (en este trabajo se consideró una ventana de 100.0[ms]) y se obtienen las velocidades de cada rueda por medio de la siguiente expresión.

$$\omega_{l_s} = n_1 p; \quad \omega_{r_s} = x_{4A} n_2 p \quad (15)$$

donde ω_{l_s} y ω_{r_s} son las velocidades de las ruedas izquierda y derecha, respectivamente, $j = 1, 2$, n_j es el número de espigas contadas en el potencial de membrana x_j , y p una ganancia de sintonización para regular la velocidad del robot móvil. El potencial de membrana x_4 puede entrar en un comportamiento tipo espiguelo, o en un modo de relajación completa, esto dependiendo del valor que tome G , ver Ec. (12). La señal x_{4A} es la señal x_4 acondicionada, x_{4A} toma un valor unitario positivo ($x_{4A} = 1.0$) cuando x_4 entra en modo espiguelo, sin embargo, x_{4A} toma un valor unitario negativo ($x_{4A} = -1.0$) cuando x_4 entra en el modo de relajación completa. De acuerdo con una adecuada sintonización en los parámetros de HR3 y HR4, y la propuesta en (12), x_4 entra en modo de relajación completa cuando $d_{fron} \leq x_{fmin}$, si esto no es así, x_4 tendrá un modo espiguelo. De esta forma al detectarse un obstáculo frontal a una distancia de x_{fmin} , se realiza un cambio en el sentido del giro en la rueda derecha del robot móvil, permitiendo establecer un giro de evasión.

V. CONTROL HÍBRIDO

El control que permite seguir una trayectoria deseada y a su vez la evasión de obstáculos, es un control híbrido que surge de la combinación de los dos controles antes mencionados. La condición que se establece para operar la conmutación de estos dos controles es la siguiente:

$$\left. \begin{aligned} \omega_{l_{speed}} &= \omega_{l_s} \\ \omega_{r_{speed}} &= \omega_{r_s} \end{aligned} \right\} d \leq d_c \quad (16)$$

$$\left. \begin{aligned} \omega_{l_{speed}} &= \omega_{l_T} \\ \omega_{r_{speed}} &= \omega_{r_T} \end{aligned} \right\} d > d_c \quad (17)$$

$\omega_{l_{speed}}$ y $\omega_{r_{speed}}$ son las velocidades de la rueda izquierda y derecha, respectivamente, que se entregan al robot, y d_c es

un valor umbral.

La prueba de estabilidad del control híbrido consiste en el análisis por separado de cada estrategia, para el caso del control del seguimiento de trayectoria, dicha prueba se puede consultar en [11]. Por otro lado, para el caso de los modelos neuronales, la prueba de estabilidad consiste en garantizar la sincronización entre las neuronas [1]–[3].

VI. SIMULACIÓN Y RESULTADOS

Para la simulación del robot se utiliza el software Matlab-Simulink 2020b con herramientas Robotics System Toolbox y Mobile Robotics Simulation Toolbox. La simulación consiste en colocar a dos robots móviles dentro de un espacio confinado, el robot 1 tiene implementado el control híbrido mientras que el robot 2 únicamente contiene la estrategia de evasión. La tarea es que ambos salgan de un área con una sola salida y posterior a ello naveguen de forma autónoma (sin colisionar) alrededor del resto del mapa, en especial el robot 1 tendrá que desarrollar una trayectoria deseada (en este caso un círculo de radio 2.0[m]) y efectuar una acción de evasión ante un obstáculo estático que se ubica sobre la trayectoria deseada. Por otra parte, el robot 2 navegará de manera autónoma. A continuación, se muestra la sintonización de los parámetros utilizada, aclarando que en este trabajo se emplearon datos del robot móvil Turtlebot3 (r_r radio de las ruedas, b_r separación entre las ruedas, V_{max} velocidad traslacional máxima, y W_{max} velocidad rotacional máxima [13]), ya que se piensa realizar la implementación en un robot móvil de ese tipo.

Los parámetros usados son: $r_r = 33.0[mm]$, $b_r = 89.0[mm]$, $V_{max} = 0.22[m/s]$, $W_{max} = 2.84[rad/s]$, $r_{ciculo} = 2[m]$, $k_1 = k_2 = k_3 = 1.2$, $a_1 = 0.95$; $b_1 = 1.35$, $c_1 = 0.85$, $I_{min} = 3.25$, $I_{max} = 10$, $x_{min} = 0.4[m]$, $x_{max} = 5.0[m]$, $a = 1.0$, $b = 3.0$, $c = 1.0$, $d = 5.0$, $s = 4.0$, $x_0 = -1.6$, $r = 0.005$, $r_d = -0.0021$, $\epsilon = 0.5$, $\gamma_1 = \alpha = 1.0$, $\gamma_2 = k = 10.0$, $I = 3.281$, $x_{fmin} = 0.5[m]$, $p = 0.7$. Las condiciones iniciales son: $x_1(0) = 3.7[m]$, $y_1(0) = -1.0[m]$, $\theta_1(0) = \frac{\pi}{2}[rad]$, y $x_2(0) = 6.0[m]$, $y_2(0) = 0.0[m]$, $\theta_2(0) = \pi[rad]$. Además, se considera un tiempo de muestreo de 0.05[s].

VI-A. Resultados

En la Figura 2 se puede observar cómo fue la evolución de las trayectorias deseadas del robot 1 (trayectoria color magenta) y el robot 2 (trayectoria color azul), es importante notar que posterior a la salida del rectángulo central por parte del robot 1, este trata de converger a la trayectoria deseada (trayectoria color rojo), en la cual se monta después de evadir el obstáculo estático que se colocó en medio con el fin de poner a prueba el control híbrido. En [14] se halla un hipervínculo, el cual dirige hacia un video que muestra de forma animada, la evolución de las trayectorias de los robots móviles que se consideran en la simulación numérica. De esta manera, se aprecian las evasiones realizadas por parte de ambos robots.

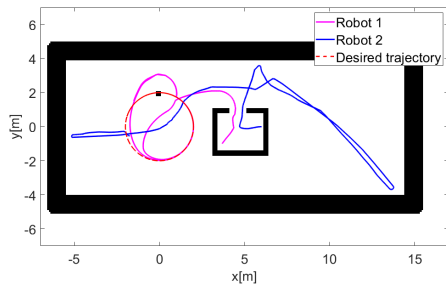


Figura 2. Trayectorias de los robots móviles.

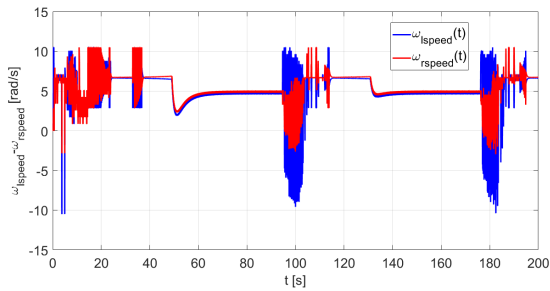
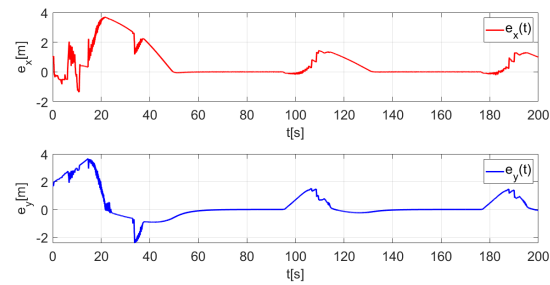
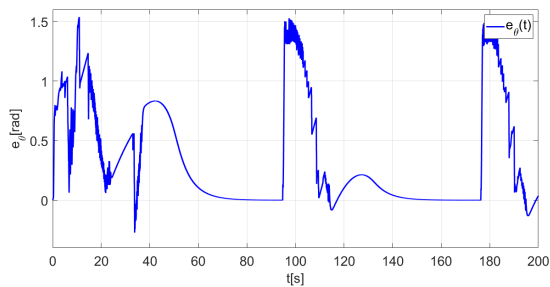


Figura 3. Velocidades de las ruedas $\omega_{l_{speed}}$, $\omega_{r_{speed}}$ del robot 1.



(a) Errores de posición en x e y . Robot 1.



(b) Error de posición en θ . Robot 1.

Figura 4. Errores de posición. Robot 1.

Las señales de las velocidades de rueda $\omega_{l_{speed}}$ y $\omega_{r_{speed}}$ del robot 1 se muestran en la Figura 3, donde se puede observar como en un primer momento varían, siendo una mayor que la otra para efectuar giros de evasión, es decir el robot 1 es operado con la estrategia de evasión basada en

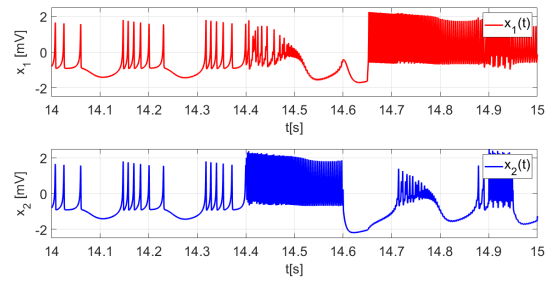


Figura 5. Zoom en la evolución de los potenciales de membrana x_1 , x_2 , que actúan en el control híbrido. Robot 1.

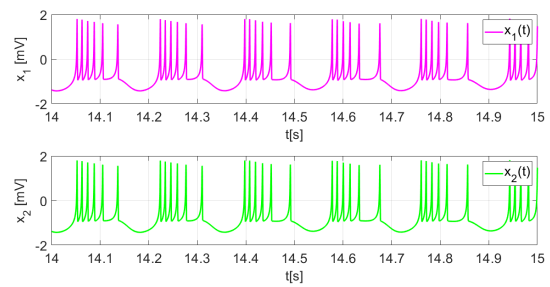


Figura 6. Zoom en la evolución de los potenciales de membrana x_1 , x_2 , que actúan en el robot 2.

el modelo neuronal y no tiende a la trayectoria deseada, sin embargo, después de que el robot 1 sale del área rectangular, es evidente como las velocidades de las ruedas izquierda y derecha son similares y constantes, esto significa que el robot 1 es operado por el control de seguimiento de trayectoria y tiende a la trayectoria deseada, la velocidad de la rueda derecha es ligeramente mayor que la velocidad de la rueda izquierda para que el robot móvil genere la trayectoria circular. Otra forma de apreciar el resultado que aporta el control híbrido es viendo las Figuras 4a y 4b que corresponden a los errores de posición. En la Figura 4a de igual forma, se nota como no se tiende a una convergencia directamente debido a que el robot 1 se encuentra dentro del rectángulo central junto al robot 2, posterior a ello se ve como en un instante de tiempo cercano a 95[s] se evade al obstáculo estático (otra evasión similar ocurre en un tiempo cercano a los 175[s]). En la Figura 4b es evidente como el robot 1 varía su orientación para generar las acciones de evasión.

Una forma más clara de notar los resultados aportados por el control basado en el modelo neuronal, es observando los gráficos de la Figura 5, que corresponden a los potenciales de membrana x_1 y x_2 , en estos se nota como en el comienzo se tiene una sincronización en las señales x_1 y x_2 , esto quiere decir que el robot móvil posee la misma velocidad en ambas ruedas y esto le permite desplazarse en línea recta, por otra parte, hay intervalos de tiempo donde se nota como hay un mayor número de espigues en un potencial de membrana que en otro, lo cual significa que una velocidad de rueda es mayor que la otra y se efectúa un giro, que puede ser de

evasión ó, debido a la trayectoria deseada.

En la Figura 6 se puede apreciar un zoom en la evolución de sus potenciales de membrana x_1 y x_2 , aquí se puede notar como en ese instante x_1 y x_2 se encuentran en sincronización, esto significa que el robot 2 posee la misma velocidad en sus ruedas izquierda y derecha y se desplaza en línea recta.

VII. CONCLUSIONES Y TRABAJO A FUTURO

Tanto el control híbrido como el control de evasión basado en el modelo neuronal, han mostrado un resultado aceptable. Otros controles requieren un conocimiento completo o parcial del entorno para generar una planificación previa de una ruta, y así conseguir una navegación autónoma sin colisiones (donde solo se consideran obstáculos estáticos), sin embargo, el esquema tratado en este trabajo le permite al robot móvil navegar de forma autónoma mientras evade obstáculos estáticos o dinámicos sin el conocimiento del entorno. Otro aspecto es que, en otros esquemas de este tipo el robot móvil se detiene cuando detecta un obstáculo frontal, pero un plus de este trabajo es la estrategia de evasión frontal que permite al robot móvil navegar continuamente.

Los resultados de este trabajo dependen de una correcta sintonización de los parámetros del modelo, así como, el primer y segundo par de neuronas deben ser idénticos (configuración ideal). La distancia umbral x_{fmin} permite al robot móvil detectar obstáculos frontales, si este valor se reduce, entonces, el robot móvil puede explorar más área del entorno, pero, probablemente, el robot móvil requiere una mayor velocidad para evitar los obstáculos (recuerde, se utilizan datos de turtlebot3 y se tienen velocidades máximas como límites para la evasión). Hablando de la distancia umbral d_c , esta permite conmutar los controles cuando se detecta un obstáculo, por lo que, si este valor se disminuye, entonces, el robot móvil también requiere una mayor velocidad para evitar los obstáculos. De lo contrario, si el valor d_c se incrementa, entonces, el robot móvil puede detectar fácilmente obstáculos y áreas rectangulares para poder evitarlos (esto es útil cuando el robot móvil sólo está programado con el control basado en el modelo neuronal). Sin embargo, si el robot móvil tiene que seguir una trayectoria deseada, con un valor d_c incrementado, entonces, el robot móvil deja la trayectoria deseada más fácil si detecta un obstáculo que tiene que evitar.

Las ganancias del control de seguimiento de trayectoria k_1, k_2 y k_3 permiten obtener una convergencia más rápida o más lenta a la trayectoria deseada. En particular, si las ganancias de k_1, k_2 y k_3 son pequeñas, el robot móvil podrá converger a la trayectoria deseada lentamente, sin embargo, con esto puede evitar los obstáculos fácilmente.

Acerca de la elección de la ventana de tiempo, esta se realiza con base en el periodo mínimo que hay entre dos picos sucesivos, y tal que la entrega de la medición del número de picos sea lo suficientemente rápida para que el robot móvil cambie a tiempo su velocidad y así pueda efectuar una evasión.

Como trabajo a futuro, se plantea implementar el mismo esquema de evasión en un ambiente de simulación más real como lo ofrece el software de Gazebo, de igual forma, se buscará implementar de manera experimental.

VIII. AGRADECIMIENTOS

Los autores agradecen el apoyo del CONACYT a través del proyecto de ciencia básica A1-S-26123, "Análisis, control y sincronización de sistemas complejos con interconexiones dinámicas y acoplamientos flexibles."

REFERENCIAS

- [1] E. Steur, T. Vromen, and H. Nijmeijer, "Adaptive training of neural networks for control of autonomous mobile robots," in *Sensing and Control for Autonomous Vehicles*. Springer, 2017, pp. 387–405.
- [2] T. Vromen, E. Steur, and H. Nijmeijer, "Training a network of electronic neurons for control of a mobile robot," *International Journal of Bifurcation and Chaos*, vol. 26, no. 12, p. 1650196, 2016.
- [3] G. G. Velasco Equihua and J. P. Ramirez, "Synchronization of hindmarsh-rose neurons via huygens-like coupling," *IFAC-PapersOnLine*, vol. 51, no. 33, pp. 186–191, 2018, 5th IFAC Conference on Analysis and Control of Chaotic Systems CHAOS 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896318336590>
- [4] J. A. J. Lora and A. R. Angeles, "Formación autónoma bio-inspirada para robots diferenciales," *National Congress of Automatic Control CNCA*, pp. 546–551, 2019.
- [5] M. Hank and M. Haddad, "A hybrid approach for autonomous navigation of mobile robots in partially-known environments," *Robotics and Autonomous Systems*, vol. 86, pp. 113–127, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889015302992>
- [6] J. Velagic, B. Lacevic, and B. Perunicic, "A 3-level autonomous mobile robot navigation system designed by using reasoning/search approaches," *Robotics and Autonomous Systems*, vol. 54, no. 12, pp. 989–1004, 2006. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889006000947>
- [7] M. S. Gharajeh and H. B. Jond, "Hybrid global positioning system-adaptive neuro-fuzzy inference system based autonomous mobile robot navigation," *Robotics and Autonomous Systems*, vol. 134, p. 103669, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889020305091>
- [8] E. Baklouti, N. B. Amor, and M. Jallouli, "Reactive control architecture for mobile robot autonomous navigation," *Robotics and Autonomous Systems*, vol. 89, pp. 9–14, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889016305413>
- [9] L. MUTLU and E. UYAR, "Control and navigation of an autonomous mobile robot with dynamic obstacle detection and adaptive path finding algorithm," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 355–360, 2012, 10th IFAC Symposium on Robot Control. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474667016336369>
- [10] R. Chatterjee and F. Matsuno, "Use of single side reflex for autonomous navigation of mobile robots in unknown environments," *Robotics and Autonomous Systems*, vol. 35, no. 2, pp. 77–96, 2001. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S09218890000124X>
- [11] P. K. Panigrahi and S. K. Bisoy, "Localization strategies for autonomous mobile robots: A review," *Journal of King Saud University - Computer and Information Sciences*, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1319157821000550>
- [12] C. C. de Wit, B. Siciliano, and G. Bastin, *Theory of robot control*. Springer Science & Business Media, 2012.
- [13] Std. [Online]. Available: <https://manual.robotis.com/docs/en/platform/turtlebot3/features/>
- [14] Std. [Online]. Available: <https://youtu.be/fEBYEFvW8ec>

Autonomous navigation of a mobile robot using a network of Hindmarsh-Rose (HR) neurons

Enrique Martinez Sanchez
Electrical Engineering Department *Electronics and Telecommunications Department* *Electrical Engineering Department*
CINVESTAV-IPN
Mexico City, Mexico
e-mail: enrique.mtz@cinvestav.mx

Jonatan Pena Ramirez
CICESE
Ensenada, Mexico
e-mail: jpena@cicese.mx

Alejandro Rodriguez Angeles
Electrical Engineering Department
CINVESTAV-IPN
Mexico City, Mexico
e-mail: aangeles@cinvestav.mx

Abstract—In this paper, a hybrid control is proposed for autonomous navigation of a mobile robot in an unknown space with static and dynamic obstacles. This hybrid control is composed by the commutation between an avoidance control based on Hindmarsh-Rose (HR) neurons, which are interconnected by a diffusive coupling, and a posture tracking control. The commutation is in terms of the distance sensors that are incorporated on the frontal hemisphere of the mobile robot. The mobile robot control is implemented at wheel speeds level. A small coupling force is used to take advantage of the desynchronization on the membrane potentials when different inputs (external currents) are applied to each neuron. In this way, a greater sensitivity to any detection is obtained and a strategy of lateral avoidance is generated. On the other hand, a dynamic coupling is used to establish a strategy of frontal avoidance. Additionally, the performance of the proposed controller is numerically illustrated by means of computer simulations.

Index Terms—Hindmarsh-Rose (HR) neurons, diffusive coupling, dynamic coupling, control based on neuronal model, spiking

I. INTRODUCTION

Nowadays, there are many applications of mobile robots: transporting objects, cleaning, surveillance, unmanned exploration of dangerous regions, etc. [1]–[11]. Autonomous navigation of mobile robots can be realized by a prior path planning, that requires a complete knowledge of the environment to prove the existence of a feasible path (in terms of distance, time and energy used) without collisions [5]–[7]. Unfortunately, the previous case is not common in many real applications where an unknown environment is considered. This fact requires that the mobile robots must be more sophisticated to recognize and to avoid obstacles (static and dynamic), to do self-location, to take decisions, among others, [1], [2], [4], [11]. There are many controls to satisfy the problem of autonomous navigation of a mobile robot, some of them are of hybrid type approaches, which are composed by controllers for the reactive navigation and controllers for a posture tracking [5]–[9]. An example of the latter are the controllers of fuzzy interference systems that require a partially known environment, and are switched as smooth as possible to elaborate a pre-processing phase in accordance with a specific task, to generate an optimal reference path in terms of time [5]. On the other hand, there is the combination between PI controllers and fuzzy logic controllers with backstepping algorithms that

act simultaneously, and they establish a data fusion obtained by sensors to construct maps and tasks of prior path planning [6].

To imitate natural behaviours, some bio-inspired controllers that have a confort radius and an external feedback loop that contains avoidance forces are developed [4], also, there are controllers that imitate automobile driving by human response based on fuzzy logic, reflex actions and higher level logical decisions [10]. In addition, controllers that have a wireless intercommunication between a mobile computer and a stationary computer are also proposed. The mobile computer is used to collect laser measurement data and provides the communication between vehicle and stationary pc. On the other hand, the stationary computer is in charge to generate a control algorithm for the avoidance of obstacles and the finding of a path planning [9]. Another important aspect is the variety of sensors that can be implemented, because they provide all necessary data to the mobile robots to make their tasks, if the sensors are more sophisticated then the mobile robot has a better view of the environment, however, these sensors must be selected according to the type of controller to be implemented (infrared, Lidar, ultrasonic, Bluetooth, cameras, GPS modules, ZigBee modules, etc.) [1]–[11].

At present time, the use of an “artificial brain” composed by networks of neurons is an interesting propose for the autonomous navigation of mobile robots. Candidates for these artificial brains are spiking neuronal networks, which are interconnected and their responses on the membrane potentials of each neuron are spiking type, in this way, the architecture of a natural brain is imitated [1], [2]. Generally, there are different types of neuronal models (Hindmarsh-Rose, FitzHugh-Nagumo, Izhikevich, Morris-Lecar, Hodgkin-Huxley, among others) [3], and different ways to interconnect them (dynamic couplings, diffusive couplings, chain networks, ring networks, among others) [1]–[3]. The interest on a neuronal model and the interconnection way, is the variety of spiking modes and patterns that can be generated on its response, these patterns can be interpreted and associated with a specific task (three main modes can be obtained: spiking, bursting and chaotic bursting) [1]–[3].

In this work, the neuronal model of Hindmarsh-Rose (HR) is used, because it can produce a proper response for the scheme

of avoidance strategy, also, the electronic implementation is relatively simple. Four HR neurons are considered, the first two neurons are interconnected by a diffusive coupling [2], and the other two neurons are interconnected by a dynamic coupling, which allows operating on a relaxing mode [3].

The outline of this paper is as follows. Section II presents the problem statement and shows the kinematic model of the mobile robot. In Section III, a strategy of posture tracking control is presented. Next, Section IV shows the avoidance control based on Hindmarsh-Rose neurons. Then, in Section V, the hybrid control is introduced. Section VI shows the results of a numerical simulation as well as a discussion. Finally, in Section VII, the conclusions and future work are provided.

II. KINEMATIC MODEL AND PROBLEM STATEMENT

A. Kinematic model

Consider a type 2.0 mobile robot which is shown in Figure 1. The corresponding kinematic model is given by

$$\begin{aligned}\dot{x} &= V \cos(\theta), \\ \dot{y} &= V \sin(\theta), \\ \dot{\theta} &= W,\end{aligned}\quad (1)$$

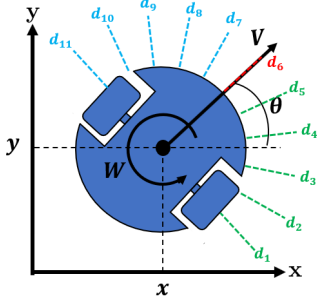


Figure 1. Type 2.0 differential mobile robot and sensor orientation.

where (x, y) is the point that represents the center and position of the mobile robot, the angle θ is the orientation that is relative to the horizontal axis, V and W are the translational and rotational velocities, respectively, and the radial position of the sensors are indicated by d_1 to d_{11} .

B. Problem statement

The autonomous navigation of a type 2.0 differential mobile robot on an unknown environment is here investigated. In particular, an unknown environment is considered as a plane map (2D environment) where there are static and dynamic (other mobile robots) obstacles. In addition, it is required that the mobile robot follows a desired trajectory and if it detects obstacles at a threshold distance, the mobile robot is able to avoid them by switching to an avoidance strategy. To tackle this problem, a hybrid control composed by a posture tracking control and an avoidance control based on HR neurons is implemented.

III. POSTURE TRACKING CONTROL STRATEGY

A strategy of a reference virtual robot that generates the desired trajectory is implemented for the posture tracking, in this way, the mobile robot will follow the reference virtual robot.

A. Procedure of the control strategy

According to [12], V and W are the control inputs of the kinematic model and they must be determined. The control procedure is as follow.

1) The parametric equations of the desired trajectory x_d and y_d are established, as well as, their first and second derivatives \dot{x}_d, \dot{y}_d and \ddot{x}_d, \ddot{y}_d , respectively.

2) The V_r and W_r velocities, and the angle θ_r of the virtual robot are calculated

$$\begin{aligned}V_r &= \sqrt{\dot{x}_d^2 + \dot{y}_d^2}, \\ W_r &= \frac{\dot{x}_d \ddot{y}_d - \dot{y}_d \ddot{x}_d}{V_r^2}, \\ \theta_r &= \tan^{-1}\left(\frac{\dot{y}_d}{\dot{x}_d}\right).\end{aligned}\quad (2)$$

3) The tracking error is defined

$$e = \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_d - x \\ y_d - y \\ \theta_r - \theta \end{pmatrix}. \quad (3)$$

4) The control is proposed

$$\begin{aligned}u_1 &= -k_1 e_1, \\ u_2 &= -k_2 \tanh(10V_r) e_2 - k_3 e_3.\end{aligned}\quad (4)$$

5) The control gains k_1, k_2, k_3 are tuned.

6) The control inputs V and W are calculated

$$\begin{aligned}V &= -u_1 + V_r \cos(e_3), \\ W &= W_r - u_2.\end{aligned}\quad (5)$$

7) Finally, the wheel speeds are determined

$$\omega_{rT} = \frac{1}{r_r} \left(V + \frac{b_r W}{2} \right), \quad (6)$$

$$\omega_{lT} = \frac{1}{r_r} \left(V - \frac{b_r W}{2} \right), \quad (7)$$

where ω_{rT} and ω_{lT} are the right and left wheel speeds, respectively, r_r is the wheel radius, and b_r is the distance between wheels. Notice that the Eqs. (6) and (7) allow operating at wheel speed level.

This posture tracking strategy is chosen because it provides a good performance and the procedure is relatively simple to implement, furthermore, the only modification was to use a $\tanh()$ function instead of $\text{sign}()$ function on the control this is to have a continuous controller and to avoid chattering, see (4), as well as tuning on the gains.

IV. COLLISION AVOIDANCE STRATEGY BASED ON HINDMARSH-ROSE NEURONS

In this strategy, the spiking variation on the membrane potentials of two neurons (HR1 and HR2), that are associated with the wheel speeds of the mobile robot, is used. Also, another two interconnected neurons (HR3 and HR4) are considered, and the membrane potential on HR4 is used to establish a strategy of frontal avoidance. The procedure is inspired on [2] and it is shown below.

1) The neuronal models are established [1]–[3]

$$HR1 : \begin{cases} \dot{x}_1 = y_1 + bx_1^2 - ax_1^3 - z_1 + I_1 + u_1, \\ \dot{y}_1 = c - dx_1^2 - y_1, \\ \dot{z}_1 = r[s(x_1 - x_0) - z_1], \end{cases} \quad (8)$$

$$HR2 : \begin{cases} \dot{x}_2 = y_2 + bx_2^2 - ax_2^3 - z_2 + I_2 + u_2, \\ \dot{y}_2 = c - dx_2^2 - y_2, \\ \dot{z}_2 = r[s(x_2 - x_0) - z_2], \end{cases} \quad (9)$$

$$HR3 : \begin{cases} \dot{x}_3 = y_3 + bx_3^2 - ax_3^3 - z_3 + I + h_2, \\ \dot{y}_3 = c - dx_3^2 - y_3, \\ \dot{z}_3 = r_d[s(x_3 - x_0) - z_3], \end{cases} \quad (10)$$

$$HR4 : \begin{cases} \dot{x}_4 = y_4 + bx_4^2 - ax_4^3 - z_4 + I - Gh_2, \\ \dot{y}_4 = c - dx_4^2 - y_4, \\ \dot{z}_4 = r_d[s(x_4 - x_0) - z_4], \end{cases} \quad (11)$$

where $i = 1, 2, 3, 4$, $x_i \in R$ is the membrane potential, $y_i \in R$ is the recovery current, $z_i \in R$ is the adaptation current, $a, b, c, d, s, r, r_d, x_0$ are constant parameters, I and $I_1, I_2 \in R$ are the external currents (model inputs). G is an exponential function in terms of the frontal sensor and is given by

$$G = -11e^{-3(d_{fron} - x_{fmin})} + 1. \quad (12)$$

This function allows to x_3 and x_4 changing from a spiking mode to a complete relaxing mode and vice-versa, and x_{fmin} is a threshold distance. If the value of the frontal sensor reading is equal to x_{fmin} , then the membrane potentials x_3 and x_4 change from a spiking mode to a complete relaxing mode (this is better explained at the step five of the procedure.)

The HR1 and HR2 neurons are interconnected by a diffusive coupling:

$$\begin{cases} u_1 = \varepsilon(y_2 - y_1), \\ u_2 = \varepsilon(y_1 - y_2), \end{cases} \quad (13)$$

with a coupling strength $\varepsilon > 0$.

The HR3 and HR4 neurons are interconnected by the coupling signal h_2 , which is dynamically generated by:

$$\begin{cases} \dot{h}_1 = -\alpha h_1 + h_2, \\ \dot{h}_2 = -\gamma_1 h_1 - \gamma_2 h_2 - k(x_3 - x_4), \end{cases} \quad (14)$$

with a coupling strength k and positive parameters $\alpha, \gamma_1, \gamma_2$.

2) The radial position of 11 sensors is equidistantly distributed, see Figure 1.

3) The minimum distances of the right sensors group (d_{right}), of the left sensors group (d_{left}), of the frontal sensor (d_{fron}), and the minimum distance of all sensors (d) are obtained as follows

$$\begin{aligned} d_{right} &= \min(d_1, d_2, d_3, d_4, d_5), \\ d_{left} &= \min(d_7, d_8, d_9, d_{10}, d_{11}), \\ d_{fron} &= d_6, \\ d &= \min(d_{left}, d_{right}, d_{fron}). \end{aligned} \quad (15)$$

4) The expressions for the external currents I_1, I_2 (inputs of HR1-HR2) are established in terms of the minimum distances of right and left sensor groups, according to

$$I_1 = \begin{cases} a_1 I_{min} & d_{right} \leq x_{min} \\ b_1 I_{min} \left(\frac{I_{max}}{I_{min}} \right) \frac{d_{right} - x_{min}}{x_{max} - x_{min}} & x_{min} < d_{right} < x_{max} \\ c_1 I_{max} & d_{right} \geq x_{max} \end{cases} \quad (16)$$

$$I_2 = \begin{cases} a_1 I_{min} & d_{left} \leq x_{min} \\ b_1 I_{min} \left(\frac{I_{max}}{I_{min}} \right) \frac{d_{left} - x_{min}}{x_{max} - x_{min}} & x_{min} < d_{left} < x_{max} \\ c_1 I_{max} & d_{left} \geq x_{max} \end{cases} \quad (17)$$

where I_{min} and I_{max} are the minimum and maximum external currents, respectively, x_{min} and x_{max} are the minimum and maximum distances, respectively, that the sensor can read. The parameters a_1, b_1, c_1 are positive constants.

5) A count of spikes on x_1, x_2 within a time window is realized (in this work a time window of 100.0ms is considered), and the wheel speeds are calculated by the expression

$$\omega_{l_s} = n_1 p; \quad \omega_{r_s} = x_{4A} n_2 p, \quad (18)$$

where ω_{l_s} and ω_{r_s} are the left and right wheel speeds, respectively, n_j is the number of spikes counted on the membrane potential x_j , $j = 1, 2$, and p is a tuning gain to regulate the velocity of the mobile robot. The membrane potential x_4 can be on a spiking mode or a complete relaxing mode, it depends of the value of G , see Eq. (12). The x_{4A} signal is the conditioned membrane potential x_4 , that is, if the membrane potential x_4 is on a spiking mode, then, the value of x_{4A} is set to one ($x_{4A} = 1.0$), however, if the membrane potential is on a complete relaxing mode, then, the value of x_{4A} is set to minus one ($x_{4A} = -1.0$). Accordingly with a correct tuning of the parameters on HR3 and HR4, and the function G proposed in (12), x_4 is on a complete relaxing mode when $d_{fron} \leq x_{fmin}$, otherwise, x_4 is on a spiking mode. In this way, if the mobile robot recognizes a frontal obstacle at a distance of x_{fmin} , then, the right wheel speed

has opposite sign to the left wheel speed, so, the mobile robot turns to avoid the obstacle.

V. HYBRID CONTROL

The hybrid control is the commutation between the control based on the neuronal model, see (18), and the posture tracking control, see (6)-(7). This allows following a desired trajectory and to avoid obstacles. The condition to operate the commutation is in terms of d , see (15), and is established as follows:

$$\left. \begin{aligned} \omega_{l_{speed}} &= \omega_{l_S} \\ \omega_{r_{speed}} &= \omega_{r_S} \end{aligned} \right\} d \leq d_c, \quad (19)$$

$$\left. \begin{aligned} \omega_{l_{speed}} &= \omega_{l_T} \\ \omega_{r_{speed}} &= \omega_{r_T} \end{aligned} \right\} d > d_c, \quad (20)$$

where $\omega_{l_{speed}}$ and $\omega_{r_{speed}}$ are the left and right wheel speeds, respectively, that are given to the mobile robot, and d_c is a threshold value. It is important to mention that the priority is to generate wheel speeds that allow obstacles avoidance. On the other hand, it is not intended to ensure smooth wheel speed changes.

VI. SIMULATION AND RESULTS

A numerical simulation on the Matlab-Simulink 2020b environment is realized, using the Robotics System Toolbox and Mobile Robotics Simulation Toolbox. The simulation consists in two mobile robots inside of a rectangle area that has only one way in and out. The robot 1 is magenta and is programmed with the hybrid control (19)-(20). The robot 2 is blue and it is programmed only with the avoidance strategy (control based on the neuronal model) see Eq. (18). The task is that the mobile robots have to exit the rectangle area. After this, the robot 1 has to follow a desired trajectory (in this case the desired trajectory is a circle with a radius of 2.0[m]), also, a static obstacle is on the desired trajectory and the robot 1 has to avoid it. On the other hand, the robot 2 navigates autonomously without collisions. The tuning of the parameters is shown below. In this work, the data of the Turtlebot3 are used (r_r wheels radius, b_r distance between wheels, V_{max} maximum traslational velocity, and W_{max} maximum rotational velocity.)

Posture tracking control parameters:

$$r_r = 33.0[mm]; b_r = 89.0[mm]; V_{max} = 0.22[m/s]; W_{max} = 2.84[rad/s]; r_{circle} = 2[m]; k_1 = k_2 = k_3 = 1.2.$$

Parameters of the control based on the neuronal model:

$$a_1 = 0.95; b_1 = 1.35; c_1 = 0.85; I_{min} = 3.25; I_{max} = 10; x_{min} = 0.4[m]; x_{max} = 5[m]; a = 1.0; b = 3.0; c = 1.0; d = 5.0; s = 4.0; x_0 = -1.6; r = 0.005; r_d = -0.0021; \epsilon = 0.5; \gamma_1 = \alpha = 1.0; \gamma_2 = k = 10.0; I = 3.281; x_{fmin} = 0.5[m]; d_c = 1.0[m]; p = 0.7.$$

$$\text{Initial conditions robot1 } x_1(0) = 3.7[m], y_1(0) = -1.0[m], \theta_1(0) = \frac{\pi}{2}[rad].$$

$$\text{Initial conditions robot2 } x_2(0) = 6.0[m], y_2(0) = 0.0[m], \theta_2(0) = \pi[rad].$$

A sampling time of 0.05[s] is considered.

A. Numerical Results

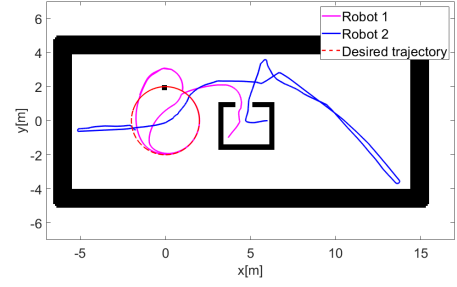


Figure 2. Trajectories of mobile robots.

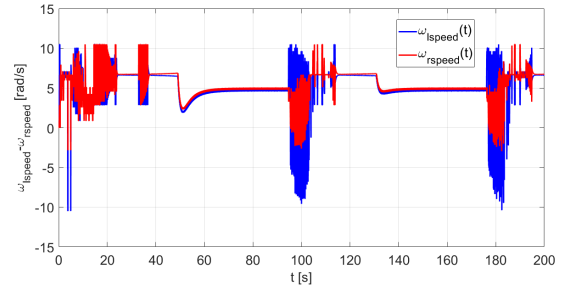
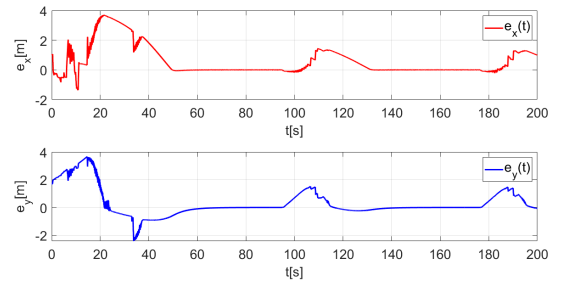
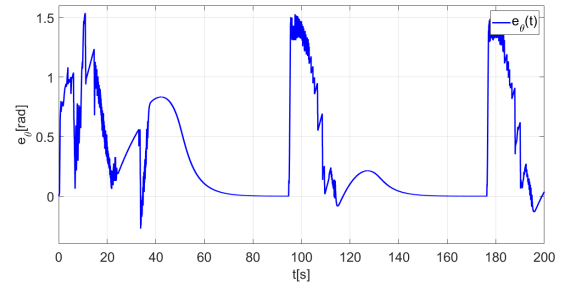


Figure 3. Wheel speeds $\omega_{l_{speed}}, \omega_{r_{speed}}$ for robot 1.



(a) Position errors on x and y . Robot 1.



(b) Position error on θ . Robot 1.

Figure 4. Position errors for robot 1.

Figure 2 shows the evolution of the robot trajectories where, after the mobile robots leave the rectangle area, it can be seen

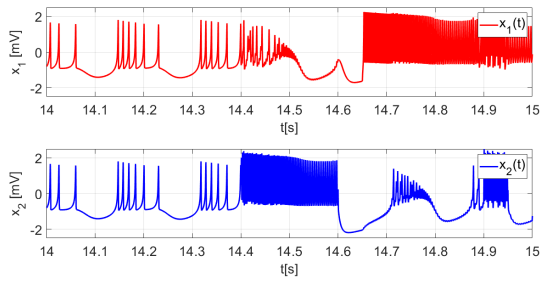


Figure 5. Zoom on the evolution of the membrane potentials x_1, x_2 that act on the hybrid control for robot 1.

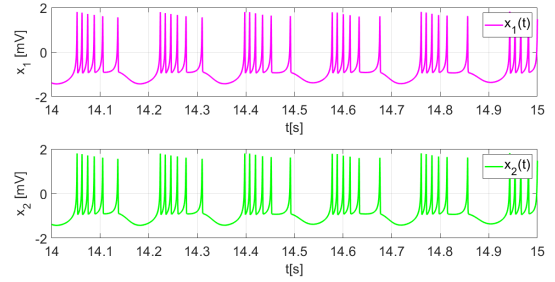


Figure 7. Zoom on the evolution of the membrane potentials x_1, x_2 that act on robot 2.

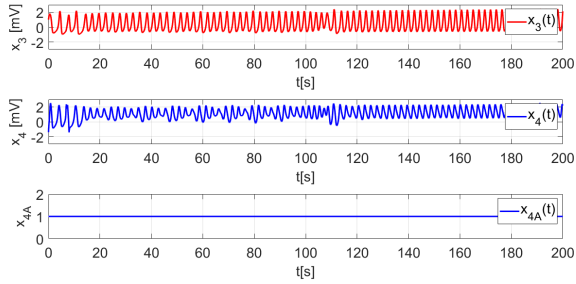


Figure 6. Zoom on the evolution of the membrane potentials x_3, x_4 and the conditioned membrane potential x_{4A} that act on the hybrid control for robot 1.

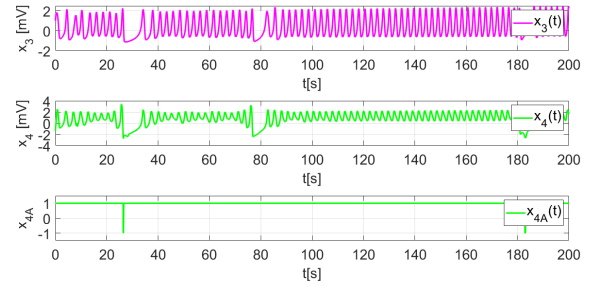


Figure 8. Zoom on the evolution of the membrane potentials x_3, x_4 and the conditioned membrane potential x_{4A} that act on robot 2.

how the robot 1 detects the static obstacle and avoids it. After this evasion, robot 1 follows the desired trajectory (red trajectory). On the other hand, the robot 2 navigates autonomously without collisions. In [13], there is a hyperlink that shows a video about the animation of the numerical simulation. The wheel speed signals ω_{l_speed} and ω_{r_speed} of robot 1 are shown in Figure 3, where it can be noted how in a first moment the wheel speeds are varying, i.e. robot 1 is operated with the avoidance strategy based on the neuronal model and does not converge to the desired trajectory, however, once the robot 1 leaves the rectangle area, it is evident how the left and right wheel speeds are similar and constant, meaning that the robot 1 is operated by the posture tracking control and tends to the desired trajectory. The right wheel speed is slightly larger than the left wheel speed such that the mobile robot is able to generate the circular trajectory, otherwise, if the wheel speeds are the equal, the mobile robot moves in a straight line. Another way to see the results of the hybrid control is to look at Figures 4a and 4b, which represent the position errors. Figure 4a shows how at the beginning robot 1 does not tend to the desired trajectory (the position errors do not tend to zero), and after this, at about 95 s the robot 1 avoids the static obstacle (another similar evasion occurs at about 175 s). From Figure 4b it becomes evident, how robot 1 is varying its orientation to generate an avoidance.

A way to note the results provided by the control based on the neuronal model is by looking at the graphs, in Figure 5,

that represent the membrane potentials x_1 and x_2 . There, it can be seen how in a first moment the membrane potential x_1 and the membrane potential x_2 are synchronized, it means that the wheel speeds of the mobile robot are the same and the robot moves in straight line. On the other hand, there are time intervals at which the membrane potentials x_1 and x_2 are different, so they are not synchronized, and the right and left wheel speeds are different, this means that the mobile robot turns. In Figure 6 the membrane potentials x_3 and x_4 of the robot 1 are shown. It can be seen how the behaviors on x_3 and x_4 are spiking modes at all time, this means that the robot 1 did not realize any frontal avoidance (all evasions were lateral). This aspect is noted in the same Figure 6, where the conditioned membrane potential x_{4A} is shown. The value of x_{4A} signal is one at all time, this means that the direction of the right wheel speed are the same that the direction of the left wheel speed.

Another aspect to highlight, is that the robot 2 navigated autonomously without collisions because it is programmed only with the control based on the neuronal model. Figure 7 shows a zoom on the evolution of the membrane potentials x_1 and x_2 that act on robot 2. There, it can be noted how in this time interval x_1 and x_2 are synchronized which means that the wheel speeds of robot 2 are the same and the mobile robot moves in straight line. Figure 8 shows the membrane potentials x_3, x_4 and the conditioned membrane potential x_{4A} of robot 2. It can be seen how x_3 and x_4 change from a spiking mode to relaxing mode in a small interval of time (at about 25 s

and another similar occurs at about 185 s). Consequently, the value of x_{4A} is minus one, this means that the direction of the right wheel speed is opposite to the direction of the left wheel speed, so, robot 2 makes a frontal avoidance.

VII. CONCLUSIONS AND FUTURE WORK

The hybrid control and the control based on the neuronal model showed a good performance. Other controls require a complete or partial knowledge of the environment to generate a path planning for an autonomous navigation without collisions (where only static obstacles are considered), however, the scheme of this work allows the mobile robot to navigate autonomously while avoiding static or dynamic obstacles, without knowledge of the environment. In neuron application literature is common to use a coupling strength relatively large to get an easy synchronization, it is also common to have a network composed for a large number of neurons, however, in this work the number of neurons is decreased and a relatively small coupling strength is used to have an acceptable sensibility for the evasions. Another aspect is that, in other schemes of this type the mobile robot stops when it detects a frontal obstacle, but a plus of this work is the frontal avoidance strategy which allows the mobile robot to navigate continuously. The results of this work depend of a correct tuning of model parameters, as well as, the first and second pair of neurons must be identical (ideal setting). The threshold distance x_{fmin} allows the mobile robot to detect frontal obstacles, if this value is decreased, then, the mobile robot can explore more area of the environment, but, probably, the mobile robot requires a higher velocity to avoid obstacles. Talking about threshold distance d_c , it allows to commute the controls when an obstacle is detected, so, if this value is decreased, then, the mobile robot also requires a higher velocity to avoid obstacles. Otherwise, if the d_c value is increased, then, the mobile robot can detect obstacles and rectangular areas ("rooms") easily and it avoids them (this is useful when mobile robot is only programmed with the control based on neuronal model). However, if the mobile robot has to follow a desired trajectory with an increased d_c value, then, the mobile robot leaves the desired trajectory easily if it detects an obstacle. The parameters a_1, b_1, c_1 and p originally were set to one (in [2]), however, they help to obtain a faster or slower turn for avoidance, and allow to increase or decrease the mobile robot velocity. The gains of the posture tracking control k_1, k_2 and k_3 allow to obtain a faster or slower convergence to the desired trajectory, if these gains are large, then, a faster convergence is obtained, but if these gains are very large, then, the mobile robot could not avoid obstacles, because the priority of the mobile robot is to follow the desired trajectory and it can move in reverse if is necessary- However, while moving in reverse, the mobile robot can not detect and avoid obstacles. On the other hand, if k_1, k_2 and k_3 gains are small, then, the mobile robot may converge to the desired trajectory slowly, however, it can avoid obstacles easier. So, with the tuning proposed in this work, a good performance is obtained for hybrid control and control based

on the neuronal model, and the performance is illustrated with several numerical simulations where the mobile robot navigates autonomously in an unknown environment without collisions. Importantly, avoidances and tracking of desired trajectory are decided by perception of the mobile robot.

For future work, an experimental validation is desired.

VIII. ACKNOWLEDGEMENTS

The authors acknowledge the support from the SEP-CONACYT project: A1-S-26123, "Análisis, control y sincronización de sistemas complejos con interconexiones dinámicas y acoplamientos flexibles."

REFERENCES

- [1] E. Steur, T. Vromen, and H. Nijmeijer, "Adaptive training of neural networks for control of autonomous mobile robots," in *Sensing and Control for Autonomous Vehicles*. Springer, 2017, pp. 387–405.
- [2] T. Vromen, E. Steur, and H. Nijmeijer, "Training a network of electronic neurons for control of a mobile robot," *International Journal of Bifurcation and Chaos*, vol. 26, no. 12, p. 1650196, 2016.
- [3] G. G. Velasco Equihua and J. P. Ramirez, "Synchronization of hindmarsh-rose neurons via huygens-like coupling," *IFAC-PapersOnLine*, vol. 51, no. 33, pp. 186–191, 2018, 5th IFAC Conference on Analysis and Control of Chaotic Systems CHAOS 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896318336590>
- [4] J. A. J. Lora and A. R. Angeles, "Formación autónoma bio-inspirada para robots diferenciales," *National Congress of Automatic Control CNCA*, pp. 546–551, 2019.
- [5] M. Hank and M. Haddad, "A hybrid approach for autonomous navigation of mobile robots in partially-known environments," *Robotics and Autonomous Systems*, vol. 86, pp. 113–127, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889015302992>
- [6] J. Velagic, B. Lacevic, and B. Perunicic, "A 3-level autonomous mobile robot navigation system designed by using reasoning/search approaches," *Robotics and Autonomous Systems*, vol. 54, no. 12, pp. 989–1004, 2006. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889006000947>
- [7] M. S. Gharajeh and H. B. Jond, "Hybrid global positioning system-adaptive neuro-fuzzy inference system based autonomous mobile robot navigation," *Robotics and Autonomous Systems*, vol. 134, p. 103669, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889020305091>
- [8] E. Baklouti, N. B. Amor, and M. Jallouli, "Reactive control architecture for mobile robot autonomous navigation," *Robotics and Autonomous Systems*, vol. 89, pp. 9–14, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889016305413>
- [9] L. MUTLU and E. UYAR, "Control and navigation of an autonomous mobile robot with dynamic obstacle detection and adaptive path finding algorithm," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 355–360, 2012, 10th IFAC Symposium on Robot Control. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474667016336369>
- [10] R. Chatterjee and F. Matsuno, "Use of single side reflex for autonomous navigation of mobile robots in unknown environments," *Robotics and Autonomous Systems*, vol. 35, no. 2, pp. 77–96, 2001. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S09218890000124X>
- [11] P. K. Panigrahi and S. K. Bisoy, "Localization strategies for autonomous mobile robots: A review," *Journal of King Saud University - Computer and Information Sciences*, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1319157821000550>
- [12] C. C. de Wit, B. Siciliano, and G. Bastin, *Theory of robot control*. Springer Science & Business Media, 2012.
- [13] Std. [Online]. Available: <https://youtu.be/fEBYEFvW8ec>